Volume Thirty-Two, Number Four
Winter 2015-2016, $6.95 US, $8.95 CAN

2600

The Hacker Quarterly

5 4>
0 71486 83158 7

Worldly Payphones

Italy. Seen on the island of Capri, this is a standard Italian phone, usually not spotted in such elaborate housing.

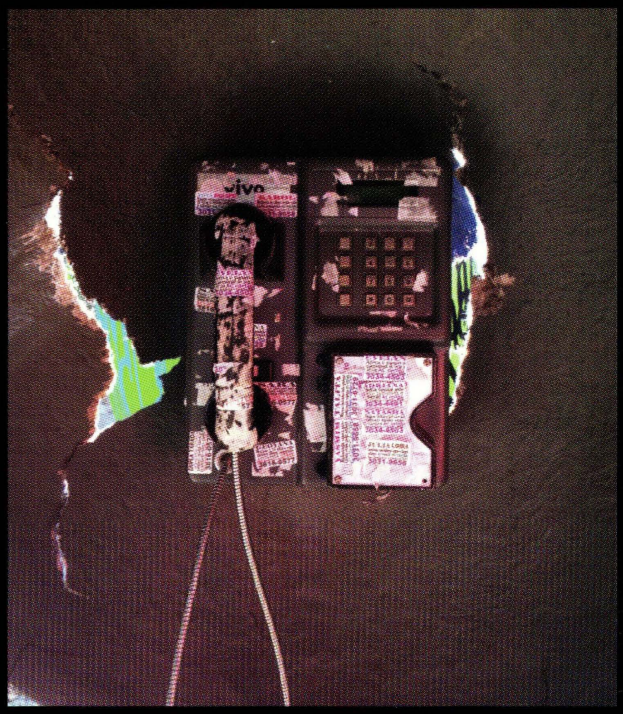
Photo by Paul

Australia. This brightly colored phone was found on Fraser Island, where humans have lived for over 5000 years. Clearly, they've learned how to keep their phones clean.

Photo by SirBif

Peru. This model, discovered in Cusco, is about as old school as you can get. The shoelace around the handset to keep it from hitting the ground is an especially nice touch.

Photo by Jessica Otte



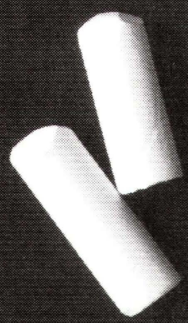
Brazil. This phone, believe it or not, is in a very nice part of Sao Paulo. Even more unbelievable is the fact that it still works.

Photo by Renato Leon Bourdony

Got foreign payphone photos for us? Email them to **payphones@2600.com**. Use the highest quality settings on your digital camera! (Do not send us links as photos must be previously unpublished.) (More photos on inside back cover)

TEACHINGS

plus ça change...



The New Normal

Sadly, this past year ended in much the same way as it began. With fear, anger, and a whole lot of uncertainty. Whether your attention was focused on terrible events that took place in cities like Paris or Kabul, Charleston or Baga, or any others from a very long list that would easily fill these pages, terror was the pervasive theme. It came in a variety of sizes and it was always delivered with an astounding lack of reason.

The one comforting thing - if we can even call it that - is that none of this is anything new. That means we at least have an opportunity to learn from past mistakes and to be ready for future challenges. The abundance of each is going to keep us very busy.

In the days immediately following 9/11, the United States was defiant. We heard many vows never to yield and how if we changed our values we'd be letting the terrorists win. Yet, as the years progressed, that is pretty much exactly what we did.

The Patriot Act was one of the more obvious mistakes. In passing it less than two months after the attacks, our government acted in a hasty and destructive manner, causing more damage to the foundations of our country than any act of terrorism could. We now had surveillance powers that bypassed due process, the ability to detain people indefinitely without trial, lack of oversight for agencies like the FBI who gained far-reaching powers, a much broader definition of terrorism subsequently used for a wide variety of investigations and prosecutions, sneak and peek warrants used as standard operating procedure... you get the idea. Our rights were taken away and it's very unlikely we'll ever get them back. We're simply not speaking loudly enough for that to happen. And for a great number of people, these rights simply aren't that important. They have bought into the illusion that these kinds of changes are keeping us safe, which is pure propaganda and nothing more. Any claims that the Patriot Act has done anything to keep us more secure are easily refuted with the actual evidence. Even the FBI admitted in 2015 that no major cases were cracked because of its existence. Not one. And yet, we now accept as reality a society where we can expect to be monitored against our will, profiled, questioned, and encouraged to always be suspicious. This is our new normal.

When the rules change, it's rarely sudden. Such things tend to occur stealthily and out of sight. Most of it happens when we avert our gaze, like a very sophisticated sleight of hand. While we're all out at the circus laughing at some orange-haired clown, we're not paying attention to the fact that we're being robbed. And when we emerge from our trance, we find that the landscape has changed. Not so dramatically to keep us from falling for the same trick again, but enough to continue moving us in a certain direction.

If we were to look at our world from, say, 20 years ago, we would likely be astounded at how we've become obsessed with surveillance and control. When panic is part of the mix, the speed of such change can be increased, but there's still that critical period where it needs to set into place in the populace or ultimately be rejected. And there hasn't been a whole lot in the way of rejection when it comes to draconian new laws. We have been far too accommodating in accepting these alterations and in allowing our fears and concerns to be exploited.

In the wake of the latest Paris attacks, we saw an almost immediate response from the authorities, quite similar to what we saw in New York 14 years earlier. We're not referring to the kind of response we expect and need from them in the wake of such an event. This is something far more insidious.

Consider these words which appeared in our pages shortly after September 11, 2001: *"It was as if members of Congress and lawmakers were poised to spring into action the moment public opinion began to turn and before common sense had a chance of regaining its dominance. Within hours of the horrific events, new restrictions on everything from encryption to anonymity along with broad new powers allowing much easier wiretapping and monitoring of Internet traffic were being proposed - all with initial overwhelming support from the terrified public."*

Once more in recent weeks, we've seen the demonizing of encryption, even without evidence that encryption played *any* part in the planning of the Paris attacks. In much the same way that the public can be manipulated into fearing a particular group of people, so too can we be conned into believing that encryption - and by extension, privacy - somehow poses a threat.

The fact of the matter is that people are easier to track today than ever before. Through various social networks and a desire to fit in with something, we put ourselves on the grid in ways that make investigators absolutely ecstatic. Many of us have the equivalent of a tracking device on our person every minute of the day - and we willingly pay for the privilege. Never before has it been possible to follow a terrorist group on a network like Twitter and find hundreds and hundreds of associates to investigate and listen in on. There is more than enough out there for any decent spy to infiltrate and learn about all sorts of strategies and weaknesses. None of this is in any danger of disappearing. If anything, it's rapidly expanding.

What the authorities want - what *all* authorities want - is the ability to pick and choose from our private data, to hear and see it all without having to do any of that bothersome digging. This is why the NSA was caught red-handed spying on so many innocent people. They wanted to just *have* all the data and sift through it later at their convenience. This explains why there is so much anger in the government towards Edward Snowden. He took that reality away from them and revealed the very inconvenient truth of their actual motivations. And we all owe him a great debt. But by portraying such whistleblowers as traitors who cost innocent lives, those in power manipulate us once more into avoiding the actual issue of our rights being abused and the Constitution trampled upon. Yet nobody has gone to jail, been fined, or forced into exile for these illegal actions. Only the person who revealed them has been punished. What can make it more clear as to what the true agenda is here?

By coming to terms with the fact that this is simply how governments will always act, we can at least come to expect that and take steps to protect ourselves. It almost isn't their fault, just as it isn't really a snake's fault when it bites you. It's simply what they do and what they will always try to do. We're the only ones with the power to keep history from repeating itself.

Now is the time to *embrace* the technologies we're being told to reject. We need to encourage their use, not shy away from them. As long as people talk to each other and devise evil plots, there will always be ways of finding out more and devising methods of defeating them. Learning how to communicate securely amongst ourselves doesn't hinder this process, no more so than the inability to read our thoughts hinders it. If the authorities somehow had *that* capability, you can bet they would fight like hell to hold onto it. But we know it's not something they need, nor have the right to. Similarly, they don't have the right to monitor us the way they want to under the guise of security.

All of this is moot, however, since technology will always allow for a way around restrictions. If encryption is made illegal, terrorists will still be able to encrypt communications, as will all of us. If we put back doors into everything, all we're doing is opening up another security vulnerability that inevitably will be exploited for nefarious purposes. No matter how you look at it, the general public loses some of their rights and privacy - and the actual supposed targets lose nothing at all.

This kind of thing will happen again. And eventually we'll have a scenario where encryption *is* actually used at a pivotal moment and the authorities will attempt to use this as evidence that encryption is the problem. It's not. It's reality, as much so as our thought waves or the air we breathe. Instead of making it the issue, we need to come to terms with the fact that it's just another tool - and a very essential one - as technology and communications evolve.

This is the good side of what's normal now. Let's accept it for what it is.

The Best Way to Share a Treasure Map

by Mike
mike@tofet.net

Recently I spent a fair amount of time researching forward error correcting codes, also known as FEC. FEC are a class of algorithms and math constructs that allow you to reconstruct a damaged message in a one-way communication channel that doesn't allow you to ask the sender to send it again. This is opposed to a two-way communication channel, such as TCP/IP, that allows you to detect an error and ask for a resend.

These codes are actually quite prevalent in modern technology. They make CDs, DVDs, and other optical data storage methods work. They are integral in interplanetary communications due to the extreme time lags. You also see one every time you look at a QR code. Those pixelated bar codes are designed to still be readable even if there is significant damage to the original code. It is for an optical data encoding method much like a QR code that I started investigating FEC.

Along the way I discovered there are two very distinct classes of forward error codes. There are forward error correction codes and there are forward erasure correction codes. A forward error correction code can take a message of a given length, detect mangled bytes, and replace them to reconstruct the original message. A forward erasure correction code will break a message into a number of blocks and ensure that if any combination of a given number of those blocks make it to the receiver undamaged, the original message can be reconstructed. Both of them work by appending a set of parity bytes to the original data.

Both of these rather amazing algorithms are dependent on a class of finite field mathematics called Galois (pronounced "Galwah"... he was French) fields. Galois invented this math before he died at 21 years of age from injuries sustained in a duel. Personally - though I am much older and highly unlikely to engage in a duel - I found myself more than a little flummoxed by the complexity of this math, despite the more than a few tutorials available on this specific topic. But I still need to use them for my project.

I've got a communication channel that can reliably send data, one-way, with nearly 99 percent reliability. But those errors express themselves as random, unpredictable bit flips. The correct number of bits will arrive, but they might not be right. This is actually a perfect application for forward error correction and I began my hunt for an available library accordingly.

Long story short, there aren't any good open-source forward error correction libraries. There are some really promising contenders, but I could not get them to work and, due to my limited understanding of the underlying math, I could not fix them. But, there are a quite a few very excellent forward *erasure* correction libraries out there. After some experiments, I settled on zfec, which has a native Python library with the underlying C readily available.

You can get zfec at: `https://pypi.➥python.org/pypi/zfec` or your Linux distribution may have a package available. I had better results with the package on python.org than I did with the native Ubuntu package.

Once you have zfec installed you can issue a command such as:

```
zfec -m 5 -k 3 myfile.txt
```

This will break your document into five files (called "shares" in zfec parlance), any three of which may be recombined by the command:

```
zunfec -o myfile.txt myfile.txt.0_
➥5.fec myfile.txt.2_5.fec myfile.
➥txt.4_5.fec
```

This will yield your original file again, even though two of the pieces are missing.

Although it was completely unrelated to my original project, I got to thinking. What a wonderful way to share some critical file with your friends, but prevent any one of them from accessing the file on their own. In the case above, if someone wants to read myfile.txt, they have to obtain at least three of the portions to do so. It is a mathematical way to

enforce collaboration (or, if you are a CISSP, "collusion").

So, say you were with a crew of pirates - like "Arrrrgh!" pirates, not the political party - and you buried some treasure. You want to share the treasure map out, but you don't want any one pirate to be able to find it on their own. If you had nine pirates, and you knew they socially formed three distinct groups of three pirates each, you could protect your map by using:

```
zfec -m 9 -k 7 treasuremap.jpg
```

Then, assuming the pirates were able to keep their individual files safe, no individual or single group of pirates could recreate the treasure map on their own. They would have to come to an agreement across all of the groups to go after the treasure.

Why not just split it into nine files and make all nine required? Because, then, one person could prevent anyone from getting to the treasure. That would be a tyranny of the minority.

It's parliamentary politics enforced by 19th century math!

But there is a problem. Inside of each share there is a portion of the file that is unmodified by zfec. If the original file was ASCII text, you would be able to read a portion of that file in each share and that would be unacceptable. The solution I came up with was to encrypt the file before splitting it.

Using AES-256 in Cipher Block Chaining mode means each block cannot be decrypted without knowledge of what the previous block was, even if you know the encryption key. So, I AES-256-encrypt the file, store the initialization vector at the front of the file and the randomly-generated key at the back of the file, and zfec-split the encrypted file. The owner of the first share will not be able to decrypt her block because she doesn't have the key. The owner of the last share can't decrypt his block, even though he has the key, because he doesn't have the initialization vector nor the ciphertext in the previous share.

I became quite taken with this idea. Seems like a great way to disperse a copy of your will to your beneficiaries. They don't know what it says, but they know they have to come together to hear it! There might be other great applications of this technique.

To make this easy, I wrote a python script called "tmap.py", short for Treasure Map. The code is included below. You need to install the zfec package and the PyCrypto package in order to use tmap. Both of these are readily available in the Python package library.

Tmap will encrypt and split a file into your designated number of shares, storing the necessary metadata to recreate and decrypt the file in each block. Tmap will then recreate a file when the proper number of shares are found.

To encrypt and split a file (assuming you have made tmap.py executable):

```
./tmap.py --make m k filename.ext
```

m = the number of total shares; k = the minimum number required to recover the file

For instance:

```
./tmap.py --make 3 2 rickroll.mp4
```

will create three files named: rickroll.mp4-1.tmap, rickroll.mp4-2.tmap, rickroll.mp4-3.tmap

You keep one for yourself and give the other two to your friends. Friend #2 comes over and *really* wants to see Rick, so he gives you a USB stick with his share on it. You then use:

```
./tmap.py --recover rickroll.mp4
➥-1.tmap rickroll.mp4-3.tmap
```

Tmap will recreate rickroll.mp4 and you can sit back and enjoy the show.

I tried to keep the code for tmap.py as short as possible to assist print publication, so there is a lot of error checking missing. The code will also overwrite files willfully, so it is best to make and recover in a fresh folder.

Now if I could only find a math concept that would force Congress to compromise and legislate!

Code follows:

```
#!/usr/bin/python

import zfec
from Crypto.Cipher import AES
from Crypto import Random
import sys

def tmapzfecsplit(data, m, k):
    splitter = zfec.easyfec
➥.Encoder(k, m)
    splitdata = splitter.encode(
➥data)
    # calc pad
    padlen = len(splitdata[0])
➥*k - len(data)
    # prepend k,m sharenum, and
➥ padlen to each block - can't
➥ recreate without it
    splitdata = [str(k)+","+str(
```

```
➥m)+","+str(snum)+","+str(padlen
➥)+":"+sdata for snum, sdata in
➥ enumerate(splitdata)]
    return splitdata

def tmaprecreate(datablocks):
    sharenums = []
    cleanblocks = []
    k = 0
    m = 0
    padlen = 0
    for block in datablocks:
        splitdata = block.split(
➥":", 1)
        cleanblocks.append(splitdata[1])
        metadata = splitdata[0].split(",")
        k = int(metadata[0])
        m = int(metadata[1])
        sharenums.append(int(metadata[2]))
        padlen = int(metadata[3])

    if len(cleanblocks) < k:
        # not enough blocks provided
        raise Exception("Can't recreate file. Need {} parts, only
➥ have {}.".format(k, len(cleanblocks)))

    decoder = zfec.easyfec.Decoder(k,m)
    origdata = decoder.decode(cleanblocks, sharenums, padlen)
    return origdata

def tmap_encryptdata(data):
    # AES-256 encrypts the data, pre-pending the IV and appending
➥ the key
    # the IV is the stringified padding value to get to 16 byte block
➥size
    # 32 byte key is random
    padlen = 16 - (len(data) % 16)
    iv = "0" * (16- len(str(padlen))) + str(padlen)
    data = data + "0" * padlen
    key = Random.new().read(32)
    cipher = AES.new(key, AES.MODE_CBC, iv)
    encryptedData = iv + cipher.encrypt(data) + key
    return encryptedData

def tmap_decryptdata(encryptedData):
    # AES-256 decrypts data
    # expects first 16 bytes is IV (and padlen)
    # and last 32 is the key
    iv = encryptedData[:16]
    key = encryptedData[-32:]
    cipher = AES.new(key, AES.MODE_CBC, iv)
    msg = encryptedData[16:-32]
    data = cipher.decrypt(msg)
    padlen = int(iv)
    data = data[:-padlen]
    return data

def tmap_split(filename, m, k):
    # encrypts data in filename
    # splits the data into m parts, k of which can recreate the file
    # prepends the filename to each file
    # saves as filename-X.tmap
    f = open(filename)
    data = f.read()
```

```
    f.close()
    enc_data = tmap_encryptdata(data)
    datablocks = tmapzfecsplit(enc_data, m, k)

    for filenum, data in enumerate(datablocks):
        f = open("{}-{}.tmap".format(filename, filenum+1), "wb")
        data = filename+":"+data
        f.write(data)
        f.close()

    return

def tmap_rebuild(filenames):
    # rebuilds a tmap file from the parts in filenames
    savefilename = ""
    datablocks = []
    for name in filenames:
        fr = open(name, "rb")
        data_raw = fr.read()
        fr.close()
        data_split = data_raw.split(":", 1)
        if savefilename == "":
            savefilename = data_split[0]
        elif not savefilename == data_split[0]:
            raise Exception("Not all data is from the same file.
➥ Cannot recreate")
        datablocks.append(data_split[1])

    enc_data = tmaprecreate(datablocks)
    data = tmap_decryptdata(enc_data)
    f = open(savefilename, "wb")
    f.write(data)

def tmap_cmdline(args):
    # not using getopt to save code space, but harder to detect
➥ errors
    help_text = 'Usage:\ntmap --make m k filename\n\tSplit filename
➥  into m parts, k of which are needed to recover\n'
    help_text = help_text + 'tmap --recover file1 file2 file3 ... filen
➥\n\tAttempt to recover the file from the listed parts.'
    help_text = help_text + '\nArguments must be in order!!'

    if len(args) < 3:
        print help_text
        sys.exit(2)

    try:
        if args[1] == "--make":
            m = int(args[2])
            k = int(args[3])
            tmap_split(args[4], m, k)
            print "{} split into {} parts, {} needed to recover.".
➥format(args[4], m, k)
        elif args[1] == "--recover":
            tmap_rebuild(args[2:])
            print "File recovered!"
        else:
            print help_text
    except Exception as e:
        print "Failed to work:"
        print e.args[0]
        sys.exit(2)

tmap_cmdline(sys.argv)
```

A Program for the Very Paranoid Computer User

by Aaron Grothe
ajgrothe@yahoo.com

One of the first things the authorities or a company will usually do when they grab a computer is to "secure" the computer. This usually involves the following steps: making sure the user cannot touch or do anything else with the computer (such as close the lid of a laptop, unplug the power, or type anything on it). Next is usually installing a device called a "Mouse Jiggler." The final step is usually making sure the computer has power, either through battery or a UPS, so they can investigate it at their leisure.

Mouse Jiggler is a simple USB device that simulates a mouse and jiggles the cursor a few pixels every few seconds. The purpose of these is to prevent your computer from engaging the screen saver or doing anything else it might do while idle, such as unmounting encrypted drives, and so on. There are also similar devices that will emulate a keyboard and hit the shift key in the same manner. These devices are readily available online just look for Mouse Jiggler.

What Can You Do?

On Linux/BSD and Mac OS X, there is a program called USBkill which, when installed and running on your computer, will monitor the USB bus of your system and shut down the system if it detects any changes to your attached USB devices (adding or removing). In this example, once Mouse Jiggler is installed, the system will shut down and optionally perform some basic security cleanup (removing files, wiping memory, swap, and so on) as well as running any custom commands you'd like.

What Can USBkill Do for You?

- Remove files
- Remove directories
- Remove the USBkill program (useful if you only encrypt certain directories)
- Wipe Swap
- Wipe Ram
- Custom Commands

Whitelisting a USB Device

If you have a USB device that you regularly plug into and unplug from your computer, you can add it to the USBkill whitelist. This way it won't trip the USBkill command. For instance, I plug and unplug my Nokia phone into my Linux box on a daily basis. To add it to the USB whitelist, I followed these steps:

```
# lsusb
```

find the entry for the Nokia phone

```
Bus 001 Device 016: ID 0421:06fc
➥ Nokia Mobile Phones
```

add the "0421:06fc" to the whitelist section of the usbkill.ini file

Note: USB IDs can be cloned, so keep in mind that this is a potential security risk.

A Few Tips

1) You can have a USB memory stick or other device on a lanyard connected to your wrist. That way if you pull it out of the system it will initiate a shutdown. This is suggested by the author of the USBkill program.

2) USBkill uses the Secure Delete commands, so make sure that you have those utilities installed if you want to be able to do file removal and other commands. You can also modify the usbkill.ini file to use different commands if you'd prefer.

3) USBkill by default uses the fast versions of the Secure Delete commands - "sdmem -l" instead of "sdmem", "srm -l" instead of "srm". You can enhance the strength of the wipe by removing the "-l"s from the usb.ini for the additional security. Keep in mind these will also slow down the speed at which your computer halts.

4) To test USBkill without shutting down the computer (to make sure you have everything started correctly), you can start USBkill with the "--no-shut-down" option.

5) If you write a program to launch USBkill automatically when you start your system, you might want to give it a few minutes to let the USB devices be recognized or else you can end up with a machine that refuses to boot. This one is a personal experience issue!

6) Rename the usbkill.py program to something else before you run it. This way if a tech savvy person grabs your computer and you have a longer set of shutdown commands, they won't see the program running if they do a "ps" command.

One Enhancement for USBkill

The following is one simple enhancement I've added to my version of USBkill. It adds the capability to send a "pkill --signal USR1 -f usbkill" from a terminal to shut down the system. One issue with this is that the terminal with this command also needs to be running as root. Here is the patch if anybody else would like to apply it:

Patch

```
--- usbkill.py 2015-09-04 09:55:41.000000000 -0500
+++ usbkill_sigusr1.py 2015-09-22 13:36:41.320000000 -0500
@@ -438,9 +438,18 @@
                log(settings, "[INFO] Exiting because exit signal was
➥ received")
                sys.exit(0)

+       # Define SIGUSR1 handler
+       def usr_handler(signum, frame):
+               print("\n[INFO] Starting system shutdown because SIGUSR1
➥ was receieved\n")
+               log(settings, "[INFO] Starting system shutdown because
➥ SIGUSR1 signal was received")
+               kill_computer(settings);
+
        # Register handlers for clean exit of program
        for sig in [signal.SIGINT, signal.SIGTERM, signal.SIGQUIT, ]:
                signal.signal(sig, exit_handler)
+
+       # Kill computer if you receive a SIGUSR1
+       signal.signal (signal.SIGUSR1, usr_handler);

        # Start main loop
        loop(settings)
```

Future?

USBkill is designed to do one thing and it does it pretty well. At the GitHub page for it, there are several new feature requests. One of the most interesting is the ability to also detect Thunderbolt, Ethernet, and FireWire changes. Also, the ability for a laptop to detect whether it is running on AC or battery power might be useful as well. The source code is pretty small for USBkill and it is pretty well documented, so it is easy to customize to meet your needs.

Summary

As there is "Security in Depth" there is also "Paranoia in Depth." Tools such as USBkill can be useful if you are doing work on your computer and you would like to be able to quickly shut down your system in the event that someone tries to grab your computer.

References

GitHub repo for USBkill - https://github.com/hephaest0s/usbkill

Home page for Secure Delete Utilities - https://www.thc.org/releases
➥.php?q=delete/

Circumventing Chrome and Firefox's Third Party Cookie Block

by Armando Pantoja

Many web browsers, including Chrome and Firefox, make a strict distinction between first-party and third-party cookies. First-party cookies are created by the web server identified by the address shown in the browser's address bar. Third-party cookies are created when content is loaded from domains other than the one shown in the address bar, by iframe, for example. By default, these websites will not allow third party cookies, even if you have specifically allowed them in your settings, without you first visiting the originating site. This means clicking an inbound link or typing the URL in your address bar. This allows you to "opt in" to receiving cookies.

This policy is meant to increase security, but much like most "pseudo security" engagements, this can be easily circumvented.

I ran across this issue while developing an asp.net application for a client that required us to create a separate widget that would be installed on an existing WordPress site. This widget was basically an iframe that showed a form from the asp.net application that we created. Our client needed the widget to integrate into this existing site, and allow the user to input a username and password, and forward them to the application that we had written in .net seamlessly. Asp.net form authentication writes a token cookie to validate each user, and during testing we found that Chrome and Firefox were blocking this cookie from being written, resulting in errors that would not let the user login to the application.

By default both Chrome and Firefox have a setting that blocks and allows first- and third-party cookies. By default, first-party cookies are allowed, and third-party cookies are blocked. This was a huge issue because both of those browsers make up a significant percentage of the marketplace. At this point we had two options:

1) Start from scratch and recreate the application to handle the third party issue.

2) Find a way to circumvent the third party issue.

Of course, we choose the second option.

The most surprising, and actually scary, thing about the solution was that we figured it out in less than ten minutes. It was very simple.

After a user logged in, we created server side logic to test to see if the cookie was written.

If indeed the cookie was not written, we redirected the parent page to the secondary URL by using JavaScript:

```
window.top.location.href =
➥ "http://www.ourwebapplication
➥.com/noCookies.html";
```

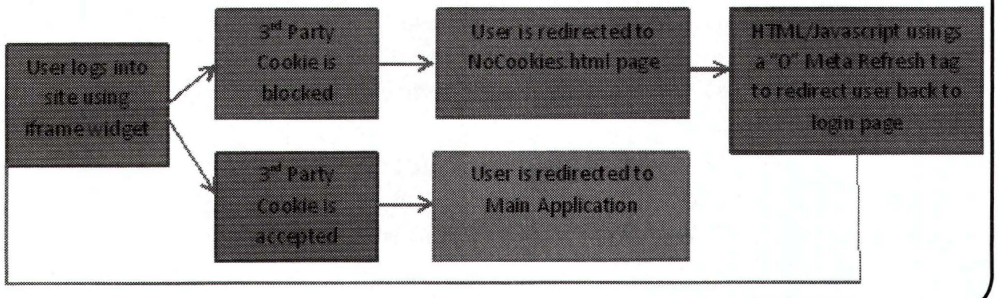
Once the secondary page was loaded, we used a meta tag refresh to redirect back to the original login page on the primary site, to create an instant client-side redirect:

```
<html>
 <head>
 <meta http-equiv="refresh"
➥ content="0;URL='http://Login
➥Page.com/'" />
 </head>
</html>
```

Using this redirect scheme, the third party cookie opt-in was completely satisfied as the browser now sees the user has visited the secondary URL and the user is then allowed to login, and the cookie is allowed to be written.

From the user side, they simply see the page they are on refresh, as the redirect happens faster than they can perceive it.

The reason this has been implemented is not for real security, but simply a marketing scheme to win the "browser wars" by casting themselves as a more secure option. This was easily circumvented with a little bit of thought and creativity. This type of "pseudo security" only hurts and slows down those doing real work, and does nothing to stop hackers or malicious attacks.



TELECOM INFORMER

by The Prophet

Hello, and greetings from the Central Office! I'm at a toll center downtown today, overseeing the installation of some new equipment. As I have mentioned before, my employer operates both POTS and wireless service. And today is a particularly special day, one that is likely to give me a lot of future headaches. The equipment we're installing involves a feature that has been the utter bane of my existence: Voice over LTE, or VoLTE. Nothing has worked according to plan, and as we install the equipment today, I know that things are going to get worse before they get better. I'm already dreading my upcoming performance review. But all of this is a topic for a future column.

LTE was sold to network operators as a convergence solution to essentially every technology problem they had in the mid-2000s. This should have been a giant red flag, but network operators were almost willing to believe anything that could get them out of the bind they were in. At the time, the wireless landscape in the U.S. was a messy mishmash of technologies, at least relative to the rest of the world. And the technologies in place were being rapidly outgrown. Carriers and the FCC had to figure out a way to address the fragmented market in the U.S. And in the mid-2000s wireless market, there was plenty of divergence. Five different and entirely incompatible technologies were in wide use across the country, and even when there was technology parity with other parts of the world, there wasn't frequency parity. And the end result was that American mobile networks were falling rapidly behind the rest of the world.

As is often the case in technology, early decisions result in technical debt that can carry forward for a very long time. The problems in North America started with a technology mismatch that had its roots in AMPS, the legacy analog cellular network. Mobile phones were more or less invented in the U.S. at Bell Labs, and the first standard to be widely deployed was Bell Labs' Analog Mobile Phone System, or AMPS. Licenses in 850 MHz frequency ranges were granted nationwide, and duopolies were established in each market area. One license set, also known as the "A" carrier, was made available for companies that didn't operate land line phone service in the market area. McCaw Cellular, Dobson Cellular, and many other companies that evolved to operate under the "Cellular One" umbrella bid for these licenses. The second license set, known as the "B" carrier, was made available to companies that operated land line phone service in the market area. Most of these licenses were snapped up by the Regional Bell Operating Companies (RBOCs) that served the given market areas, such as US West and Ameritech, but some licenses (such as the one in Portland, Oregon) were won by independents such as GTE.

As digital technology became available, carriers wanted to switch to it. It was more efficient than AMPS, allowed more calls to be handled per cell site, and it offered value-added features such as data service and text messaging that weren't available on AMPS. However, the FCC had other plans. The licenses they'd granted specified AMPS. People with analog phones didn't want to be forced to buy new ones, and AMPS offered usable (albeit highly insecure - anyone with a modified scanner could listen) service over a longer distance than digital technologies. So although carriers decided to switch to digital technology, they had to continue supporting AMPS. In fact, the FCC held fast to its requirement to support AMPS all the way through February 18, 2008, long after the technology was considered obsolete!

Faced with a mandate to continue supporting AMPS, carriers looked for technologies that could work on both digital and analog networks. The first such technology to become available was called IS-54, marketed as TDMA, which launched in 1993. Later, Qualcomm released a superior technology called IS-95, marketed as CDMA, which launched in 1995. Both standards were backwards compatible with AMPS, so your phone would keep working if you traveled into an analog-only area (albeit without digital

features). Digital handsets used the same, old, familiar programming as AMPS handsets. The cellular companies that became Verizon, along with US Cellular and several other smaller network operators, chose the CDMA system for their digital evolution, roughly following the lines of the former "wireline" B-side AMPS licenses. Meanwhile, AT&T, Dobson Cellular, and most other A-side carriers chose TDMA (which were the first digital networks deployed in the U.S.). This was the first real technology split in North America, because the two digital technologies were incompatible. CDMA phones could roam on a TDMA network, but only by using the older (and insecure) AMPS system. The same was also true in reverse.

Europe and most other countries in the world, meanwhile, settled on the GSM standard, which launched in 1991. This system wasn't compatible with older analog networks, but these hadn't been widely deployed there in the first place; the U.K. had the largest deployment with only two such networks. Spectrum licensing in Europe also didn't depend on maintaining compatibility with older networks. In the U.K., carriers forced their users to upgrade handsets and abruptly switched off the analog system shortly after the launch of GSM.

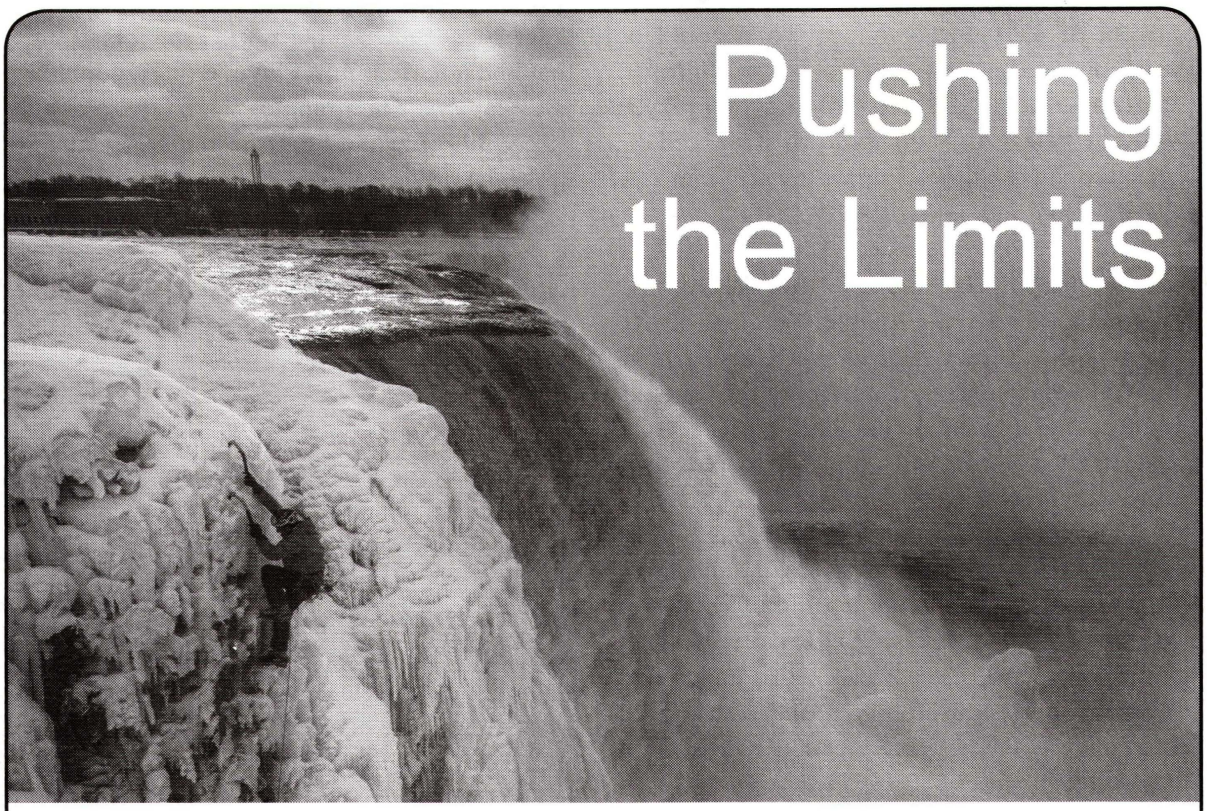
Europe also wasn't hamstrung by congested spectrum, as was the case in the U.S. For its deployment, two frequency bands around 900 MHz were initially chosen (additional bands around 1800 MHz were subsequently deployed). These bands were also adopted in most places outside of the Americas. Unfortunately, the 900 MHz and 1800 MHz bands were already in use in the U.S. An FCC working group explored the possibility of a frequency swap, but the exploration didn't last long. The U.S. Department of Defense lodged a formal objection, and the recommendation was made to maintain the status quo.

Meanwhile, the FCC, recognizing the demand for additional wireless services at lower prices than afforded by a duopoly, made additional wireless spectrum available. These were called "PCS" bands, and were earmarked for digital-only networks. However, the frequencies available were in the 1900 MHz bands, meaning there was no overlap with any of the digital frequencies deployed throughout the rest of the world. The CRTC closely followed in Canada with essentially the same spectrum allocations, as had been done previously with the deployment of AMPS in Canada.

Sprint and VoiceStream were the first "PCS" networks to deploy in the U.S. While VoiceStream chose GSM, the same technology in use in Europe, and its handsets worked only where VoiceStream had (poor and spotty) coverage, Sprint deployed CDMA. Through roaming agreements (initially AMPS, later both AMPS and CDMA) with the legacy cellular carriers, Sprint was able to offer coverage outside of its "native" coverage area. Meanwhile, AT&T got in on the fun a few years later, deploying its own GSM network, bringing to three the number of technologies supported on its network (AMPS and TDMA were also still supported). And to introduce an element of randomness into the equation, Nextel patched together a network by buying taxi dispatch companies and deploying, through a loophole in the licenses, iDEN technology. While Nextel handsets operated like phones, they could also operate as half duplex two-way radios, meeting the licensing requirement.

Got all that? By 2007, the U.S. cellular airwaves were crowded with AMPS, TDMA, CDMA, GSM, and iDEN technologies deployed on three separate and distinct sets of wireless licenses and frequencies, none of which were (with a few exceptions) compatible with anywhere else in the world outside of North America. What's more, 3G had been deployed by many networks, but the 3G technologies weren't compatible with one another either! UMTS and HSDPA had been deployed by AT&T, T-Mobile (which had since acquired VoiceStream) hadn't deployed any 3G technology, and Sprint and Verizon were operating 1xEV-DO. So, when the 3GPP - a working group dedicated to standardizing mobile phone technology - promised a unifying standard in 2008, carriers leaped at the opportunity. LTE would treat everything as packet data. Voice and data would all run on the same carrier. Everything would magically become interoperable. The mess would become untangled. It'd be called 4G.

Well, it didn't quite work out that way. 4G is currently the bane of my existence. But that's a topic for my next column, where we'll talk about the future. For now, if you're calling relatives for the holidays, enjoy the call quality. If you're moving while you talk, be happy that the call doesn't drop. I'm going to do my best, but no guarantee that either of these will be the case when this awful cursed VoLTE equipment goes live!



Pushing the Limits

I have had the pleasure of being involved in many beta tests for companies known, unknown, and possibly never to be again. The tasks required vary for each company. You are typically required to find bugs, problems with the software through manual processes, and to ask before using any automated process or mods to try and find bugs in games or software on a computer (i.e., randomly pressed buttons on a controller, randomly pressed keys, etc.). To my tinkering self-nature, this seemed silly until I realized how this could spiral out of control quickly and how much of a mad house the process really was. I will share my experience for the companies that have a less restrictive NDA or an NDA that expires after product release. Note: none of the products I talk about are currently under NDA or are older products out of NDA, or the NDA was associated with a company no longer in business.

Microsoft

I remember my first official beta test was the public update beta from Microsoft for the Xbox 360 (I am not allowed to talk about current updates under beta to comply with Microsoft's NDA). I remember Major Nelson advertising it or paraphrasing it in a way that made it seem cool or hip to get the update. So, I signed up through the Microsoft connect website and, sure enough, received an email to beta test the update. I remember the chaos of submitting feedback through that website. For one, there was what I would call "duplicate posters" who post the same feedback wanting the same feature to be implemented, which would drive anyone crazy after about the fifth person started posting the same thing. Somehow, Microsoft managed to make it through this chaos. Let's skip to the "Xbox 360 Kinect Version 1" beta test, so I do not bore you with details of the rest of the betas, which only gave minor improvements. I had the chance to get invited to beta test the Kinect for Xbox 360 (developers like to refer to it as "Kinect Version1 (V1)" not "Kinect for Windows (KFW) V1"). I thought it was so cool grabbing it out of the box for the first time since no one had actually seen it yet. It came in an official looking box with the calibration card, and I thought it was neat to get to play with it.

Naturally, being a programmer, eventually my experimentation kicked in and I wanted to hook it up to my computer and figure out how to program with it. During that time period, OpenNI SDK by PrimeSense was one of the few solutions for programming with the Kinect (if any were available). I played with OpenNI for a few weeks, only to realize the

appeal of hacking the Kinect so it could be used on a computer was so great Microsoft made an official SDK for it soon after. I played with the beta 1 and beta 2 of this Kinect SDK, each with new improvements the previous one did not have in a fast manner.

Once I got to Version 1 of the SDK, I started having the ability to perform 3D scanning of objects; Enable Near mode for users in close proximity to the Kinect (KFW only); and avateering or mapping movements on the Kinect to a 3D character, green screen, and other neat features that appeared over time with new releases.

I made one or two projects with the Kinect SDK: the first project I was involved in was to make a Kinect version of the Microsoft Multipoint multi-mouse SDK called Kinect Multipoint and the other is an unfinished project for translating American Sign Language with the Kinect called Kinect Sign Language.

When I started getting used to the Kinect, they released another version of the hardware sensor called the Kinect V2 (newer Xbox One version of Kinect). This is where my experience with Microsoft ends and combat testing experience begins.

Combat Testing

Combat Testing was one of those websites I first encountered and thought: "I am not sure about this." After applying, I was almost immediately accepted into their elite game beta testing program. The website was partnered with EA games and many other top game companies. I fondly remember the quote repeated on the website: "The first thing about CT is you don't talk about CT." For terms of simplicity, I will be general in my descriptions and time on the website.

After joining, I remember over time seeing beta tests for *Battlefield*, *Crysis 2*, *Homefront*, and other various games. Normally, the beta tests would be for PC with the occasional Xbox 360 or PlayStation beta. The terms of usage for the website referred to how you could not: cuss, inflame, or break or mod software without permission (the managers might have let you at least mod if you nicely asked for PC games). There was a nice quality bug feedback system which was used often. The two main games I remember beta testing were *Crysis 2* and *Homefront* (2011). *Homefront* was a game that had a varied multiplayer experience (MP) with gamers. The graphics were not great and sometimes I would get disconnected during multiplayer games (I believe this to be my connection). However, its replay value was high because of the large battlefield type experiences and players. *Crysis 2* was a game that was great during beta but I cannot state what experiences I had past beta stage.

Other Beta Tests

After combat testing, I went into other various beta tests. I tested the beta of ReconstructMe. ReconstructMe was a program for scanning 3D objects with the Kinect and a few other devices the programmers decided to support. I picked up the Lazy Susan type scanning of objects quickly with the program. Now they have a GUI with color scanning added to the abilities. I used it in combination with MeshLab in order to see my models and edit some bad scans. Besides ReconstructMe, I had the chance to beta test many other items such as Norton, a firewall I believe was from GFI, debugging software I cannot remember the name to, and other various software to which the NDA is still active and so I cannot discuss anything about it.

Where Can I Find It?

Reconstructme - `http://www.reconstructme.net`

Website to sign up for Microsoft beta programs - `http://connect.microsoft.com`

Kinect Multipoint - `http://kinectmultipoint.codeplex.com`

Kinect SDK download (for Kinect V2) - `http://www.microsoft.com/en-us/kinectforwindows/develop/default.aspx`

Microsoft Multipoint Mouse SDK - `http://www.microsoft.com/multipoint/mouse-sdk/`

OpenNI Tutorial - `http://www.codeproject.com/Articles/148251/How-to-Successfully-Install-Kinect-on-Windows-Open`

Kinect Sign Language (to be name Kinect ASL?) - `http://kinectsignlanguage.codeplex.com/`

ROMEO TANGO OSCAR

by 2-6 India

Radio Telephone Operator. Sounds like a cushy job. Air conditioned office 8 am to 5 pm. Monday through Friday. Nights, weekends, and holidays off.

Not even close.

The U.S. Army sent me to Viet Nam in 1969. I served as a combat infantryman, a rifleman, assigned to the Second Platoon of Company D, First Battalion, Seventh Cavalry, First Cavalry Division. During my first two months, I was just another grunt humping the boonies. I had always been detail oriented, and drew the diagrams when we set up automatic ambushes. An AA consisted of several Claymore mines linked by det cord; they exploded when a simple trip wire device was touched. My job was to record where each Claymore was placed and where the trip wire, blasting caps, and ignition flare were located. My sketch would be used the next morning to locate and safely disable the mines. This attention to detail put me in line to become the next RTO when a vacancy occurred.

In each infantry platoon of 20 men, there is a platoon leader, usually a lieutenant, and a platoon sergeant. Each has an RTO assigned exclusively to him for communication. The radios are the lifeline to other platoons, the company commander, medevacs, artillery support, gunships, and resupply. The men carrying the radios are called radio telephone operators, RTO for short. There is no formal training other than observing an RTO for a day or two. The rest is learned on the job. An RTO is an infantryman, but his first job is radio communications.

In late February 1970, my company was on LZ Compton, a remote fire base in Sông Bé province, when the North Vietnamese Army launched a ferocious mortar attack. The platoon command bunker took a direct hit. One man was killed and six were wounded. Among the dead and wounded was the acting platoon leader, the new platoon sergeant, and both their RTOs.

The next day our new platoon leader, a fresh-faced second lieutenant, arrived and I became his RTO. The first thing I had to

learn was the Army/NATO phonetic alphabet. Each letter of the alphabet is represented by a specific word. A/Alpha, B/Bravo, C/Charlie, and so on. Words are used instead of letters to avoid confusion with letters that sound alike.

The next task was to learn the specific identifiers for the company. The company commander was designated "6." Each platoon leader was designated by the platoon number and then 6. So, first platoon leader was 1-6, second platoon leader was 2-6. Each platoon sergeant was "5." So, the second platoon sergeant was 2-5, and so on. Each RTO was designated as "India." So the company commander's RTO was 6-India. As the second platoon leader's RTO, I was 2-6-India. Our platoon sergeant's RTO was 2-5-India. This may sound confusing, but it was actually simple, made sense, and was quickly learned.

As an RTO, the radio became part of me, and I attached myself to the lieutenant, carrying the radio on my back attached to my rucksack. By today's standards, the radio was big and heavy. In fact, the PRC-25 (pronounced "prick-25") was the first solid state FM backpack radio used by the Army. It had 920 channels spaced 50 kHz apart, operating in the 30-75.95 MHz spectrum. It transmitted about 1.5 watts of power. The operating distance was three to seven miles. It weighed about 25 pounds, was approximately four to five inches thick, ten to 12 inches wide, and about 18 inches long. It had a metal case, painted subdued green. There was a black cord, similar to a home telephone of the day, and a black plastic handset that resembled a small version of that on a regular telephone. On top of the radio were several dials. These were used to change the frequencies on which we communicated. These frequencies were changed on an irregular basis. A change usually occurred in the middle of the night. Since it would be totally dark, the second dial could be preset so that a one click turn of that dial would accomplish the change at the appointed time just by feel.

The radio itself was water resistant. I never totally submerged it, so I don't know if it would still operate. We RTOs did our best to keep it dry when crossing streams or rivers. The handset was thought to be damaged by water. When it rained, we wrapped the handset in plastic, which did not interfere with commo. For most uses, a small, flexible whip antenna, about two and a half feet long, protruded from the top of the radio. I could walk through most jungle conditions with no problems. On occasion we used a folding antenna, about ten feet in length, which increased the frequency strength, but was so tall and rigid it could only be used when we were not moving. Considering everything I encountered, I felt confident with this radio. I never experienced a situation when the radio did not function. The radio took on water, dust, dirt, heat, bumps, bangs, and drops, and never failed.

Regulations required the battery be replaced daily. No exceptions. The batteries, three to four inches thick - the length and width of the radio - looked like cardboard bricks. They clicked into place on the bottom of the radio in a purpose-built compartment, which protected them from the elements. Once a new battery was installed, the old battery, which still contained power, was destroyed: the enemy had their own booby traps for us. Several methods of destruction were employed. In relatively safe areas, we smashed the battery with a shovel or a rock, or hacked it with a machete. Where noise was a problem, two wires in the battery were pulled out and attached together. The battery would get intensely hot, start to smoke, and eventually short itself out, rendering it useless.

In the jungle, we were usually resupplied by helicopters every three days. Each RTO received three new batteries during each resupply. We would immediately replace the old battery, but had to carry the others - which weighed three pounds each - in our packs. Due to the weight of the radio and batteries, RTOs did not carry Claymore mines and M-60 machine gun ammunition, which every man except the medic had to do. On jungle patrols, I walked directly behind the platoon leader, giving him the handset as needed. This allowed him to give and receive orders on the radio while still moving forward. When we stopped to set up a temporary position, the platoon leader would determine on the map our exact position. He would then give me our latitude and longitude coordinates. I would take the numeric coordinates and convert them to alphabet letters using a code book.

The code book had different numeric/alphabet conversions for each day and for each 12 hour portion of each day. Therefore,

it was critical to go to the correct page for that day and time to make the correct conversion. Our position would be recorded by an artillery crew on a distant fire base. If we came under attack I would call in our encrypted coordinates for artillery fire. Any mistake could result in "short rounds," i.e., artillery shells that dropped on us instead of NVA or VC. After I made the conversion, I would call my radio counterpart at the company level and tell him our coded location. For example, I would say: "6 India, this is 2-6 India. Our location is Juliet Mike Golf Delta Victor Sierra Romeo. I say again, Juliet Mike Golf Delta Victor Sierra Romeo." 6 India would then read the letters back to me to confirm a correct transmission. We never used the word "repeat." The word "repeat" was *only* used when we wanted artillery to fire exactly the same coordinates again and again. Some RTOs did not use the phrase "I say again." They used instead the phrase, "I shackle," and then read the letters. I was aware that RTOs in other platoons had their coordinate conversions checked by the platoon leader before transmission.

My platoon leader never checked my conversions. Although I appreciated his trust in me, knowing a mistake could have deadly consequences, I always had the platoon sergeant's RTO check my conversion. When we stopped at the end of the day and set up a night perimeter, I had several duties. Either myself or the sergeant's RTO would accompany the fire team setting out the automatic ambush, usually a hundred meters from where we were. Our job was to maintain radio contact with the company and announce our return to the night perimeter once the AA was set up. This ensured that we would not be mistaken for the enemy. The next duty would be to convert our position to the alpha code. I would say, "This is 2-6 India. Our November Delta Papa is Oscar Hotel Quebec, etc." During the night, I would initiate sit reps, which were used to ensure that the men on guard duty in foxholes around the perimeter were awake and monitoring the radio. Softly, I would speak into the handset, "This is Silverspartan 2/6 Indy, what is your sit rep?" The usual response was "My sitreps are negative at this time." If the answer was anything else, for example, "I have movement," the platoon leader would speak with that man immediately. If absolute silence was required, my request would be, "If your sit rep is negative, break squelch twice." A push of the transmit button on the radio handset made a noise known as squelch on the receiving end. To break squelch twice, the handset button was pushed twice quickly in succession. Simple, but effective, and totally silent.

During an ambush, firefight, or mortar attack, things changed. The platoon leader would communicate with his counterpart at the company level, and with artillery and helicopter gunships. Most, if not all of this communication would be "in the open." There was simply no time to encrypt words and coordinates. On occasion I would communicate with gunship pilots. Usually this concerned the color of smoke grenades. Purple was "Grape" or "Goofy Grape." Yellow was "Banana." Green was "Green Giant." Red smoke was "Ruby Red." One day while on LZ Compton, a Cobra gunship pilot came to talk with the RTOs. It was his day off, but he hitched a ride on a resupply bird to get to the firebase. He spent several hours with us discussing communication techniques and how to improve our effectiveness. He was not lecturing us, but rather seemed sincere in his desire to learn and improve.

On April 23, 1970, I was on guard duty on LZ Francis, a fire base in Tây Ninh province. I was at a fighting position on the base perimeter. The radio was propped up against sandbags, as it had been all night long with the changing guards. At about 4 am, we came under an intense mortar attack. The first round landed about 50 meters directly in front of my position. After shouting the alarm, I grabbed the radio and started running to my bunker. A round exploded and I was hit with shrapnel, and came face down in the dirt. I crawled to the bunker, pushing the radio ahead of me. The lieutenant and others pulled me inside. Using information that I gave to him, the lieutenant directed outgoing artillery fire towards the source of the incoming mortars, eventually silencing them. Severely wounded, I was medevaced to Saigon. Many months passed before my physical wounds properly healed and I could walk without a cane.

I enjoyed being an RTO. I liked being in the information loop. I willingly accepted the responsibility that came with the PRC 25. I'm happy to report my sit reps are negative.

Shout outs: third platoon medic.

YULL ENCRYPTION

by Ronald Gans

The core of symmetric encryption programs is the encryption routine itself. Most programs do not create their own encryption routine, but instead use one of several routines that have been vetted and approved by the U.S. government and academic cryptographers. You can download these routines like AES or Blowfish, along with others, and examine them yourselves. They are fairly easily to incorporate into your program, so with a moderate degree of programming skill, you can almost write your own government-standard encryption program. Perhaps.

Yull doesn't use AES. For Yull, encryption is handled in two parts which relate to each other. The first is the setup, handled by the Yull class. Besides validating parameters and files, it utilizes eight options to determine how the file is encrypted or decrypted, as you will read below. Once the setup, which is fairly complex, is completed, the reads are submitted to the encryption class, Andromeda. One of Andromeda's 60 routines is Blowfish, but the rest are written by me.

One of the bases of modern cryptography is Kerckhoffs's principle, which states that "A cryptosystem should be secure even if everything about the system, except the key, is public knowledge." (Wikipedia).

But if the size of the key is fixed or within a small range, the integrity of the system might be compromised, as that might be useful information as to which file is the key. In fact any information about the system you use is useful to the enemy. For instance, BlowFish, a popular encryption protocol, uses keys up to 448 bits, or 56 bytes; AES, the government-approved standard, uses a key up to 256 bits or 32 bytes, so that is definitely some help when trying to narrow down the range of likely keys. This key space, which is used in most other encryption programs, is not very large; and even if it is relatively immune to a brute force attack today, the trend is not good. If the key is stored on a disk, knowing the size or the size range might narrow down the key possibilities, assuming that the enemy has access to the entire contents of the disk. But if the key were not the key, that might make a difference. And with Yull, that is the case.

Also, with AES, the encryption system most commonly used, you know the rounds (the number of times the data goes through the encryption routine) are between four and 20. According to the Wikipedia page for AES, it's ten cycles for a 128 bit key, 12 for 192, and 14 for 256 bit keys. Yull uses a variable number of rounds ranging from one to 150.

Other encryption programs, mainly those which rely on AES, add a fixed amount of dummy data, so you might also figure out how much dummy data is added. Also, the files are not read out of disk order (to my knowledge; I think Yull is the only encryption application which does this).

Basically, the more predictable and reliable your encryption system is, the weaker it is.

Why Yull is Different

First, the size of the Yull key is not fixed. It can be any file between 100 and 10,000 bytes. It could be a Word file or a text file or a system file. Yull encrypts the key internally during runtime before using it, so the randomness of the key is not an issue (but of course, the more randomness the better). Yull gives you the option to create your own keys, but you don't have to. So with Yull, without additional knowledge it is impossible to even guess the size of the key, let alone where it is. Yull can create the key if you want, but you can also select the file yourself.

Second, unlike other applications (I have done a brief survey of them but I can't validate this is 100 percent correct - just seems that way), Yull's encryption mechanism depends not only on the key, but also on eight options:

Users, by setting the options, can influence, but not determine, the values Yull uses:

- *the minimum and maximum size of the read buffer used*
- *the level of encryption*
- *the minimum and maximum number of rounds in the array of rounds*
- *the amount of dummy data added*
- *the order the reads are made*
- *the personal data (a type of initialization vector for the key)*

Now these might not sound like much, but numerically they are important in putting brute force decryption way out of range.

The Array of Rounds

The rounds (the number of times a read buffer is submitted to encryption) range from one to 150. If a file is 10,000 bytes long and the read size is 100, there are 100 reads. And with 100 reads, what are the rounds? There are 149,100 (or 2.082701e+217, which is 2.08 times 10^{217}) possible combinations to try, a number way too large to be meaningful. So, in order to brute force decrypt the file, you would have to try that many attempts, for starters.

The Read Order

Yull does not read the file in disk order. That is to say, it will determine the read order once it knows how many reads there will be. If there were two reads, there would be two possible read orders (2! or 2 factorial, which is 2x1). If there were ten reads, there would be 10! (3,628,800) possible read orders (10x9x8x7x6x5x4x3x2x1), and if there were 50 reads, there would be 50! or 30414093000000000000000000000000000000000000000000000000000000000000000000000000000 possible combinations of reads. So if you knew somehow there were 50 reads, you would be facing that number of decryption attempts, but *times* the number of array combinations (149^{50}).

Read Buffer Size

This is a value between 20 and 600. If you did not know the read buffer size, you would have to try all of the numbers that fit, between 20 and 600. If your file was, say 10K in size (10,240 bytes), you would have to try all of the combinations where the read size is 20 (512 possible reads, which means 512!, a number too big to print on this page), up to the read size of 600, along with all of the other combinations of values. And you have to do it backwards. You have to exactly mimic the decryption mechanism. You would have to brute force duplicate decryption. It is physically impossible, regardless of how many quantum computers and GPU arrays one uses.

Dummy Data

Yull will always add dummy data to the read buffer. Yull will prepend between 20 and 100 bytes of dummy data (that is, pseudo-random data) to the read, and then encrypt that with the real data. The dummy data is generated with each read, so it is not the same each time. At the end of the encryption, the dummy data is scattered in with the actual data.

Personal Data

Even if the enemy figures out your key, your key is encrypted with your personal data, a 200 character text string which is like an initialization vector. While this can be stored in the options file and the options file can be encrypted (that key is not stored anywhere), you can also elect to *not* store this anywhere and Yull will ask you for it when it needs it. So even if the enemy guesses your key, they also have to guess a possible 200-character-long ASCII string of personal data. There are 247 possible values (if you enter them via ALT+#) you can use for your personal data. That's all of the normal ASCII characters (including the space) along with most of the 128 values above 127. You would enter those via your numeric keypad ALT+ two or three digit value.

This gives you 247 possible characters and, with 200 characters for the personal data, you have something like 247^{200} or 3.4625×10^{478} which, like most of my other numbers, is way too big to imagine. More than all of the nanoseconds left in Earth's existence. Its size is almost meaningless except in this context: by numbers alone it is impossible to decrypt by force. You can also use ALT+0+three digits. This maps the output to the ANSI character set rather than the IBM/OEM PC extended ASCII character set. (Note: Yull does not recognize Unicode; to Yull they are just bytes in a file.)

But you do not have to store this data in Yull or in the options file. You can check the "Ask For PData" check box and Yull will ask you for it at run time and never store it.

Or you can use *no personal data* at all and Yull will automatically create that for you, again, only at run time. Yull uses a very long string of values which is in turn used to create a SHA512 hash which is in turn used to encrypt the key. Again, only at runtime, never stored anywhere.

The Key File

Yull can use any file it can read as a key as long as the file is between 100 and 10,000 bytes long. If you want to keep which file you use as key secret, you can instruct Yull to ask you for it at runtime.

Encryption

The encryption class (Andromeda) uses both block and stream ciphers. Block sizes are key-dependent; some blocks are derived from non-contiguous bytes (from the read buffer); byte streams may be treated as bit streams (that is, ROR and ROL on a byte stream converted into bits).

Yull uses 60 encryption functions, which are called based on the key (see the Andromeda section below). So if the read buffer were 20 bytes long, there would 6020 or 1.152921504606846976 x 10^{78} possible combinations of routines. And if the read buffer were 600 bytes long, there would be 60^{600} possible combinations, which is another number way too large to be meaningful, putting brute force decryption way out of range. But, in case you're curious, there are this many combinations:

77758921822273638463652243033738663459096026047499749793055195367088512405958251191527636305592795619147625821196869874979406220179646899435863958367759727796622149608138604042852462403683738935485852634057709256771299230380098822466080142001159030156662377138169091661053378027306159140588698401044531517806120574444647452982608759967619192447037241981639101929668716184831900032478998027811609344055333647488533711459486991110405287751122235682242271887405273317376000

For most (or all) other symmetric encryption programs, if you have the key, the encrypted file, and the program itself, you can get the plaintext. But with Yull, that is not enough, as Yull also requires that all the options are correct. And where are the options stored? Yull lets you store them in external files, but you don't have to. They could be stored on a piece of paper in a drawer if that seems best. They can be encrypted and that key is not stored anywhere.

Yull does far more work than other encryption programs. It is somewhat slower and, with the encryption level set to MAX, it could be quite a bit slower. However, if it takes, say, three seconds to encrypt (and decrypt), that is actually an additional safety feature in that brute force will take too long, more time than the Earth has left.

If you have the key and the encrypted file, you will still not have any way to guess the read buffer size (and hence the number of reads) or the number of rounds for each read; nor will you know the amount of dummy data added to each read. With the key and the program and source code, you might be able to narrow down the read order, but that's all.

By encrypting many times and with various combinations of the 60 encryption routines in Andromeda (the encryption class), it is common for Yull to encrypt a file with tens of millions, hundreds of millions, even hundreds of billions of encrypts. So, not only is there an enormous number of option parameter combinations running to unimaginably huge numbers, there are also a possible several hundred billion encrypts (and decrypts) which, without the options and the key, all have to be tried in the correct order.

The option values, which help Yull determine the read size, the dummy data amount, the read order, the rounds array, and how the key is encrypted with your personal data are stored (if you want) in an options file. This file can be encrypted as well, but the key for the options file is never stored.

What Kind of App is Yull?

Yull Encryption Pro is a .NET 4.0 application written in C# using Visual Studio 2012. The executable, YullG, is a Windows-based, GUI application. YullG has three major components (classes) and a few minor helper classes. The first one is the UI (user interface class) which handles user input, hosts the controls as WinForm controls, performs validation, manages options, and so forth. Yull.exe is a console application, which does everything YullG does, but as a console app which can be called by a batch file or a Powershell script, for instance, as a job.

The Yull Object

The console program is simpler than the GUI program. It has only three classes: the Yull class, the Andromeda class (encryption), and the Parameters class. The GUI app has many more classes, mainly to handle the graphical elements. Also, the GUI program handles the file selection and file name selection, passing the Yull object one input file and one output file at a time.

In the console app, the Yull object does all this work itself, determining the current file to be processed whether it exists or not, etc. After file handling, both the graphic program and console program are the same.

The Yull object continues with the setup. It validates again that the parameters make sense,

that the source file exists and can be opened, that an output file can be created in the specified location, and also if a key was selected. If no key is selected, Yull will exit with a message. But if Yull was instructed to make a key, it will create a key of the specified length, with a minimum length of 100 bytes. The user can select nearly any file as a key - there is no limit to the size or location as long as it is a local file (that is, one that Yull can open).

Next, Yull figures out the read buffer size. This is one of the five parameters Yull uses to encrypt.

There are several "buff" size and "round" size controls. These control how big of a read buffer Yull will create, hence how many reads. The "rounds" value helps determine how many rounds per read, that is, the number of times a read buffer is submitted to encryption. The buffer size and rounds cannot be determinedly set by the user. The values are suggestions, which Yull uses, in conjunction with the level and key to create the actual values.

The dummy size option is just that: the amount of dummy data added to each read. This means that along with the original file data, random data values are also added to the input.

The final parameter is "personal data." This is a maximum 200 byte series of characters, like a password, which Yull uses to encrypt the key.

Once Yull has the read buffer size, it determines how many reads there will be from the file size by dividing one into the other. If the read size is 100 and the file size is 1000 there will be ten reads. If the read size is 100 and the file size is 1001 there will be 11 reads.

After the number of reads is calculated (which includes the buff size), Yull creates an array of rounds. A round is a call to the encryption class, Andromeda, to begin the encryption process. When Yull knows how many rounds and how many reads, it will create an array of keys based on the supplied (or created) key. There is one key used per each round of encryption. If there are 100 reads and 100 rounds per read, there will be 10,000 keys created. In the encryption process, if the key is smaller than the read buffer, it is extended to the size of the read buffer; if larger, truncated to the read buffer size. The keys are always the size of the read buffer - in essence a one-time pad.

Levels

There are six predefined levels: MAX, FAST, NORMAL, NEURO, TINY, and PLANK.

When you select a level, the values on the options tab change to reflect that level. You can, of course, override them easily.

Yull Setting Up Actions

The Yull class performs two other major actions, one before the encryption process starts and one during the encryption process. Based on the number of reads, Yull creates an array that number long and assigns a unique number to each element of that array with the aid of the "read order" option. The numbers are not sequential and have no relation to each other; they have no significance except that they are random and not close to each other in size. Yull then reorders that array to determine the read order for encryption. The read order is under partial control of the user by setting the "read order" value.

```
Buffer No.      Value
1               390
2               107
3               414
4               242
5               192
6               171
7                83
8               169
```

So, now ordered by value, the buffer numbers are:

```
7, 2, 8, 6, 5, 4, 1, 3
```

which is the read order. Note again, these values are derived from simple math and the key. They are otherwise irrelevant, except that they are all unique. This means that Yull reads the seventh block of data first, processes it and writes it out, then the second, then the eighth, then the sixth, and so forth.

Dummy Data

Before Yull submits a block of data to the encryption object, it adds in random values from a call to the .NET RNGCryptoServiceProvider, which is also what Yull uses to create the key, if asked to do so.

The amount of dummy data to be added is either set by the user or by Yull within the range of 20 to 100 bytes. If you were to encrypt the file again, the random data would be different. Of course the random data inserted into the blocks is always different.

After Yull finishes encrypting the file, it zeroes out all of the buffers (arrays) it used, then deletes them, closes the open files and exits, returning to the UI object, which then updates the main UI form, writes some data out, and continues with the next file if there is one. The process for decryption is similar.

Yull is available for download and analysis at `https://www.yullencryption.com`.

A Brief Cryptanalysis of Yull

by Erebos (Simmons)

Let me start by saying that I am not a professional cryptographer, so take everything hereafter with a grain of salt. There was a letter submitted in 32:2 that brought to light a new encryption program: Yull. What follows is a brief analysis of the cipher it employs named Andromeda. Before going any further, let's all just agree that rolling your own encryption is a bad idea. If you do, at least make the source code readily available and with some type of mathematical backing. There are a series of white papers on the site that describe some design choices, but the source (for the cipher) can be found here: `https://www.yull-encryption.com/AndromedaCode`

Key Sizes

The author takes issue with the fact that modern ciphers only utilize "small" key sizes, typically in fixed increments. While it is true that most modern symmetric ciphers max out at 1024-bit keys, this is considered more than adequate. It would still take a billion computers making a billion guesses a second longer than time itself has existed to brute force a 256-bit key. Assuming an adversary cannot crack a one kilobyte key, every bit beyond that is just wasted computation. However, the author objects moreso to the fact that an attacker could search your drives for files the same size as a given cipher's key length and thus reveal your keys. In practice, though, one can use any arbitrary file with established ciphers by first running it through a hash function whose output is the same size as the required key. This is (very) roughly how key files are already incorporated into existing encryption software (e.g. VeraCrypt[1]).

It should also be noted that approved ciphers like AES are meant to fit a wide variety of tasks, not just file encryption on a local machine. Packet encryption, PRNG, and hardware based encipherment speak to their flexibility. By tying oneself down to user selected options and files, the use case is pigeonholed to just local file encryption. But there is still plenty of need for that. So how does Andromeda hold up?

Security Through Obscurity

The phrase "security through obscurity" is borderline blasphemy in the world of cryptography. One should never assume complexity directly correlates to strength. The irony of Andromeda is that its attempt at security might be its greatest undoing. As the first white paper states, "For most (or all) other symmetric encryption programs, if you have the key, the encrypted file, and the program itself, you can get the plaintext. But with Yull, that is not enough, as Yull also requires that all the Options are correct." These options are chosen by the user per encryption, which means they can also provide a distinguisher. For example, Andromeda can use a variable number of rounds to encrypt each block,

which opens itself up to timing attacks. The user can also set a read buffer size, but based on modern caching algorithms, this could lead to cache timing attacks. The encryption itself boasts 60 different functions are used to scramble data, but that falls to power analysis attacks. Modern ciphers that employ much simpler algorithms succumb to the same types of attacks, even the venerable AES.[2]

This is the reason we have seen the advent of ARX ciphers, which use only the constant-time operations ADD, ROT, and XOR in a defined order (see Threefish[3] or Salsa[4] for a defense of this design). By giving each encryption so much variability, Andromeda is opened up to a variety of side-channel attacks that can leak information about a user's given options. But we can assume the user's machine is locked down tight and free from any observation. So how is the encryption function itself?

60 Encryption Functions

First, I want to clear something up. The description of Yull frequently refers to data being encrypted millions of times. What is actually meant here is that the Andromeda cipher uses up to 60 different functions to scramble data a number of times for each input. I suppose this could be referred to as "encrypting" each input, but some functions (e.g. negate) provide no security in and of themselves (by the author's own admittance).

The biggest issue with the Andromeda cipher is the lack of rationale (or at least the lack of clear presentation thereof). Each round function is composed of XOR, NOT, and ROT. Now these are not bad building blocks for a cryptographic structure, e.g. SHA-3 uses the same operations plus AND.[5] As mentioned earlier, simple is not a problem, but it does require reasoning. For example, ARX ciphers derive their nonlinearity from the ADD operation, which is why they include it. But why does the 27th encryption function XOR bytes from the key with the input block? Is this the only function that does that, or is key material added in elsewhere? Why just in these functions and not in all? In fact, it is worth noting that if each function does not add key material, one could design a key/option case where only functions that perform linear operations are used. Some functions also employ matrix operations. Again, this is a solid idea, e.g. AES uses matrix multiplication over Galois fields to provide its security. Yet with Andromeda, it seems to be a simple matter of setting up a matrix full of values and then pulling specific data out (sort of like a large S-box). Ultimately, regardless of the number of functions employed, there appears to be no rhyme or reason to the data processing done in each round. When it comes to cipher design, every function should have a purpose and serve to strengthen the overall cipher. Andromeda seems to rely on its huge complexity to obfuscate any data it processes, but this does not make it secure.

Closing Remarks

I urge someone with more experience than me to do a formal cryptanalysis of the Andromeda cipher. I would be interested to see how this stands up to linear or differential cryptanalysis methods. Yull would benefit from a full source code release in an easy to access manner, along with a more formalized white paper. Unfortunately, the complexity of the encryption process only makes verifying the security harder rather than increasing the security itself. Although code snippets are provided, the white papers never give a clear picture of *why* any choices were made other than the large number of combinations they provide. While this may seem like an utter condemnation of Yull, I hope it is seen more as constructive criticism. It is only through such criticism that we have the secure cryptoprimitives available today. And in the end, it is never a bad thing to have more people interested in cryptography.

Sources

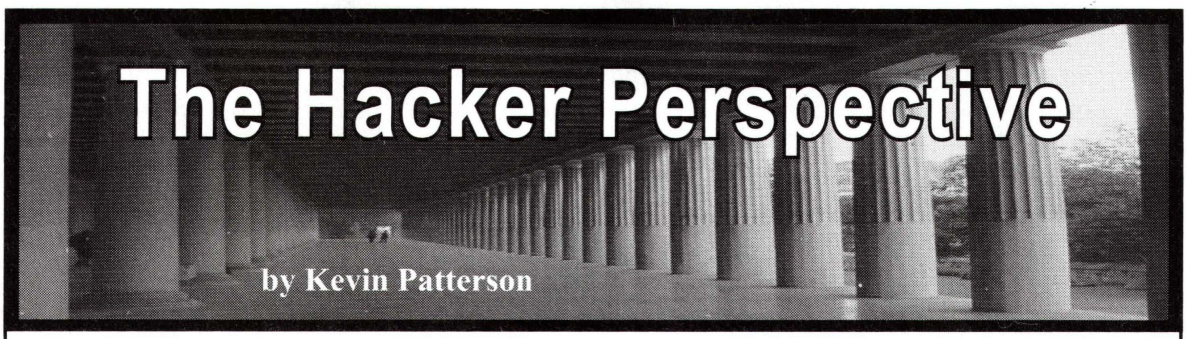
[1] http://www.veracrypt.fr/en/docs/keyfiles-technical-details/

[2] http://cr.yp.to/antiforgery/cachetiming-20050414.pdf

[3] http://www.skein-hash.info/sites/default/files/skein.pdf (specifically §2.2)

[4] http://cr.yp.to/snuffle/salsafamily-20071225.pdf (specifically §2.3)

[5] https://en.wikipedia.org/wiki/SHA-3



The Hacker Perspective

by Kevin Patterson

This is my first major submission for publication, so please be patient. You need to know that I am almost completely computer illiterate and that my hacks are probably what most of you would consider to be relics from the Pleistocene epoch, but if I understand the term correctly, a hack is neither about modernity nor technological sophistication. Rather, it is a way of thinking about and looking at the world. The way most self-described hackers, not to mention the publishers and editors of this magazine, use the word, it is someone who circumvents obstacles, or even more broadly, solves problems. It is someone driven by curiosity to discover how something works and, depending on the circumstances, either improves it or neutralizes it.

I am serving 292 months for a terrorism-related offense, and have written a book entitled *Framed* about my experiences. In it, I detailed some hacks that either I or somebody else used in the real world between 1990 and 1999.

On the rare occasions when professionals from the intelligence community comment publicly on these techniques, they usually roll their eyes and imply that such activities are embarrassing comic-book anachronisms. The truth is these techniques are still taught and practiced by every major intelligence and counterintelligence service in the world. Collectively, these techniques are referred to as "tradecraft" and continue to be part of modern espionage curriculum because A) they are simple; B) they are cheap; and C) they still work. I found out the hard way what works and what doesn't. Learn from my mistakes.

When I first wrote this article, I thought I could be both inclusive and concise, but I was over 3,000 words and wasn't half finished, so I will have to cut it down to just a couple o£ hacks that I used successfully, or otherwise.

One of my favorite hacks which I used frequently, albeit unintentionally, was to habitually ditch FBI tails. In their reports to their superiors, they breathlessly informed them that I was "surveillance conscious," but it was really just the way I drove. I hate being tailgated or feeling rushed, so I regularly took back roads and drove comparatively slowly, so they were continually having to "terminate surveillance." This simple technique really works; get in the habit of using it. Other common vehicular counter-surveillance techniques are frequently pulling over (ostensibly to consult a map), executing U-turns, pulling into driveways, and doubling back - and, if you are really suspicious, pre-positioned observers surreptitiously watch your progress for tails. This can be done in either rural or urban settings.

A simple but effective counter-surveillance method I always liked was originated by Whittaker Chambers, the Communist spy turned informer who helped put Richard Nixon on the map. This method is best used in a suburban setting. Albert is walking south on Elm Street, while Ben is walking toward him northbound, also on Elm. They can each see for several blocks behind one another and can mutually observe any tails, either vehicular or on foot. When they are parallel to each other and there is no surveillance detected, they give one another a prearranged signal, (scratching nose, pushing up glasses, coughing, etc.). If surveillance has been detected, they pass no signal at all. It is confusing, but a signal means all clear, no signal means you have a possible tail. To be doubly sure, they can check one another for tails again by proceeding in the same direction on Elm for a couple of blocks, then each turning east (or west), proceeding on two more blocks to Broadway, and again turning towards each other. This time Albert headed north and Ben is headed south. If the same person or vehicle is still trailing either of them, there is a problem. The process of losing a physical tail is called "dry cleaning."

In my original article, I profiled many more examples of tradecraft, but there just is not sufficient space. Therefore, trimming as much as possible on the subject of secure communications, a good rule of thumb to remember is the faster and more convenient it is, the less secure it is. In descending order of security, they are dead drops, face-to-face meetings, radio, mail, telephone, and Internet. I understand that this is

difficult advice to follow for a magazine whose readership consists of the hacker community, but if you want to stay out of trouble, keep off the phone and computer.

A dead drop is a physical site where information is dropped off for future retrieval without any interpersonal contact. For this reason, most professional spies prefer the dead drop. They can be in a rural or remote setting or in the middle of a city. During the Cold War, Central Park in New York City was a favorite dead drop venue for Communist bloc spies using the UN as a diplomatic cover. Cemeteries are also popular sites because they are usually sparsely populated and yet being in one arouses no particular suspicion. The usual procedure is for the person delivering the data to drop it off at the site, quickly move on, and be nowhere in the vicinity when it is picked up. This is so neither party can recognize or identify the other.

Face-to-face meetings are the next in order of security. If the person you are meeting is not betraying you and the meeting is not being monitored, you should be safe. Of course, if your contact is betraying you, no security measures are adequate. An old favorite trick for face-to-face meetings is to provide your contact with cheap nylon or canvas gym bags or similar accessories which are identical to your own. If you meet in a public venue, both of you bring your bags and exchange them, and of course their contents, during the course of the meeting. This can be easily done and is difficult to detect. If you set up such a meeting in a restaurant, library, public transportation, movie theater, or similar venue, make an effort to sit in an area away from concentrations of people. If someone enters shortly after you do and walks past more convenient seating in an apparent effort to sit near you, both of you get up and leave and go to a place selected randomly out of the phone book. Remember what the possible eavesdropper looked like; if you see him later, you are being tailed.

Radio is your next most secure means of communication. I saw a method of radio communication with which I was very impressed at a seminar in 1993. I am sure it is much more highly evolved now. The exhibitor had a laptop computer hooked up to a handheld ham radio transceiver. A typed message was sent via radio in what was even then a very brief transmission and received by a similar rig a mile away. Spies refer to such a compressed transmission as a "squirt." I doubt if the motivation of the inventor was circumventing surveillance - it was probably just reducing time between transmissions - but the effect is the same. Using encryption and techniques such as troposcatter, such transmissions can be almost impossible to detect, much less counter. If you do choose this method to communicate, do not transmit from your house. That is why you are using handheld transceivers and if you devise a set of random locations from which to transmit, make sure your house is not in the geographic center of the circle, triangle, quadrangle, or other geometric shape your transmissions are generating. I thought very highly of this system and the only drawbacks I could see were cost and the fact that you must be licensed and registered with the FCC in order to transmit on ham frequencies. Of course, the penalties are negligible and the chance of apprehension is remote, but I still don't like it. If you are a prepper, make your ham base station out of heavy, clunky old vacuum tube transceivers from the 50s and 60s. They are almost indestructible and will withstand an EMP burst even if they are turned on and in use during a nuclear strike. The handhelds won't.

Mail is probably the least appealing method to a high-tech readership, but it has the advantages of extreme elasticity, reasonable speed, and reliability along with fairly high security. One mail hack I used successfully was to send a postcard of a certain well-known local landmark to all of the members of my group, with a coded message on the back. At the next meeting, they were all greatly mystified about the strange postcard they had received. I explained that I sent it and that it was a simple arrangement for a rendezvous. The photograph on the front of the postcard specified the location of the meeting, and the text of the note on the back specified the time. On this occasion, at least, the message was received with 100 percent reliability.

While on the subject of using the mail for covert communications, allow me to give you some strange advice: don't throw out your junk mail! On the contrary, make an effort to accumulate as much as possible. Dozens of businesses have thoughtfully provided you with bulk mail envelopes with your name and address on them. Gather them up and distribute them among your contacts, and have them do the same with you. You can steam open the junk mail envelopes, insert any message you wish, reseal them, and drop them in the nearest mailbox. The post office will obligingly deliver the message to your contact in the most non-threatening format imaginable.

If the FBI or USPS are doing a mail cover on you, all they will do is record the name and address of the sender and possibly photocopy

the envelope. Junk mail from magazines, charities, politicians, and others will not get a second look. A warrant to actually open mail is much harder to get, but junk mail will still receive a low priority. Concerning resealing envelopes, if yon use egg whites as an adhesive and allow it to dry, the envelope cannot be steamed open.

One of my favorite methods of using the mails was originated by organized crime. The FBI had been monitoring the communications of several suspects for a long period and the only common denominator among them was that they all used the same dry cleaner in Las Vegas. The FBI intercepted the parcels and minutely examined them, but could find no messages. The only thing the parcels contained were dirty clothes. The solution smells like the work of an informant rather than painstaking police work to me, but eventually it turned out that the clothes themselves were the message. Number and color of shirts, short or long sleeves, cotton or linen, size, missing buttons, etc. all comprised the coded message. But the wise guys were too wise by half. Think about it: why would someone in Williamsburg or Cicero need to use a dry cleaner in Las Vegas? It was way too elaborate, but you can use the same principle successfully. Instead of sending dirty clothes, let the letter itself be the message. Size and color of envelope, size and color of paper, watermarks, number of pages, writing on one or both sides, color of ink, machine printed, hand printed or written in cursive, font type, and even odors can be used to convey the message. This way the entire gestalt of the letter becomes the message, not the content of the text. In fact, the actual text can be used to confuse, misdirect, and disinform.

Telephone communications are so easily compromised that I can only recommend two simple hacks in good conscience, neither one of which involves actual conversation. The first is the ring code, in which at prearranged times the called party allows the phone to ring without picking it up. The number of rings is the code. If you have Caller ID, the ring code can be used at any time. Let the phone ring twice, exhibiting the incoming number as an authenticator, then immediately call back with the ring code. This method has obvious limitations.

The second telephone code is the "silent call." In this case, the called party answers the phone and is met with silence. The duration of the silence is the message; the caller terminates the message by hanging up after the appropriate interval creating a dial tone. In a country with a properly functioning criminal justice system, intercepted wiretaps of such events would not be allowed into evidence; they are simply too ambiguous. Unfortunately, that does not describe 21st Century America. Why do you think the book is entitled *Framed?* The coded calls will probably be admitted, but their significance can only be speculated upon. Ultimately, there is only silence.

I know this next part is heresy if not blasphemy, but to me the Internet seems as if it were deliberately designed to be compromised. I would not use it for any but the most innocuous, vanilla communications. Yes, I know about encryption and steganography, but I still do not trust it.

My last hack is probably the most important and definitely the most low-tech. In fact, it is no-tech. It predates the wheel. It predates fire. It is your own instincts. I knew something was wrong every time the CRI (Confidential Reliable Informant) brought up the subject (and he always brought it up - I never did). I got a painful knot in the pit of my stomach. Like Captain Ahab, all good angels were mobbing me with warnings, but I foolishly allowed the veneer of sophisticated modernity with which I was brought up to drown out the vocalizing primate that was shrieking away inside my cranium.

There is nothing supernatural or irrational about this. In fact, it is the most natural and rational phenomenon in the world. It is hundreds of thousands of years of hard-learned survival instincts trying to break through the shell of rationalization and denial. It was my subconscious trying to assert itself and picking up on the rat's own subconscious cues.

You know yourself. You know the difference between nervous excitement or the thrill of the chase and a feeling of dread and impending doom. If I had listened to these hacks, I would not be here now. Listen to your instincts; they are still there in spite of all of the high-technology double-edged swords with which you are smothered. Those long dormant urges and hunches may be wiser than your deepest conscious thoughts. If something *seems* wrong, something *is* wrong.

I apologize for exceeding my allotted space, and I hope these examples meet with your definition of the word hack, and that you may profit from them. Good luck, and all hail the New World Order.

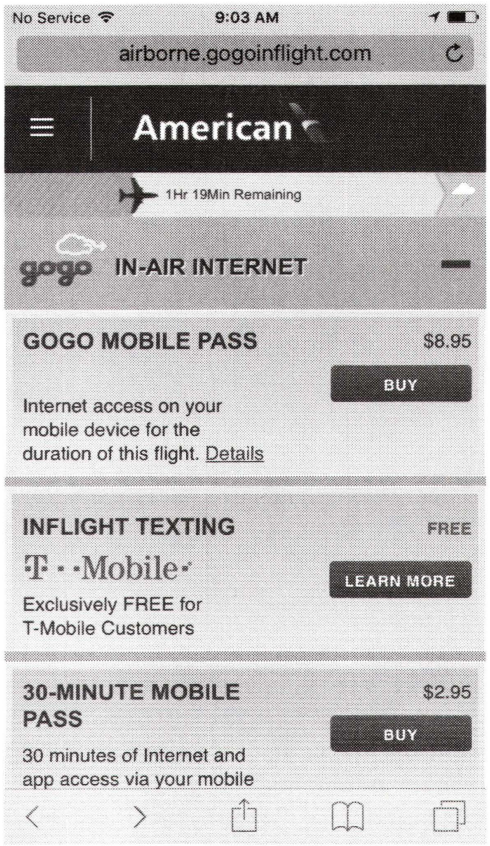
HACKER PERSPECTIVE submissions are closed for now.
We will open them again in the future so have your submission ready!

How to Get Free Gogo In-Flight Internet Access

by Big Bird

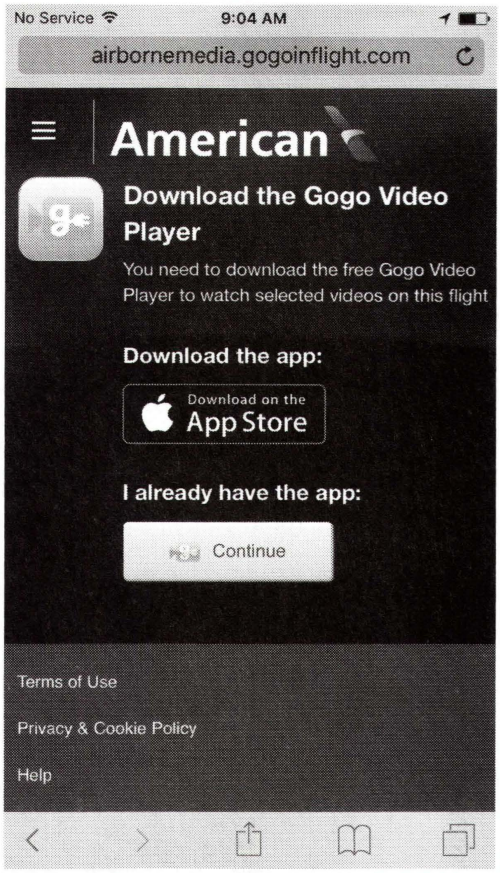
The standard disclaimer applies to this information. Use it at your own risk, and don't be surprised if some hulking Air Marshal comes down on you hard (let's hope not).

While on an American Airlines flight equipped with Internet access, I was dismayed to learn that any sort of lame 90s-era speed connection costs far too much money ($8.95, are you serious?). So I thought I'd try a few things to circumvent this cost. Turns out it was quite easy.



To do this, connect to the obligatory Gogo inflight Wi-Fi signal. On your iDevice (your mileage may vary with other types of devices), launch a browser and get to the capture portal. You'll want to scroll down and look for the "Entertainment" titles. Look through the titles or find one that is free. This might mean tapping "View All." Often these can be previews of shows or perhaps even a movie. It must be free for this process to work.

Once you've requested that free title, you'll get an "Install this app" screen. This comes up regardless of whether you actually have the app installed. Tap the "Continue" button. You'll go through a CAPTCHA process and when tapping "Submit" you'll see the "Open this page in App Store" prompt. Do it. This is likely where your MAC is authenticated for access.



Once the "Gogo Video Player" application opens in the App Store, leave all of this and enjoy your unencumbered Internet access. You'll be connected for a short time, but when the time runs out, simply repeat the above process to reconnect. This is not going to be great if you plan to Skype Granada for an hour, but for most typical things you could do online, you'll have access for the entire flight (if you want it).

Naturally, this is a real oversight on the part of American Airlines/Gogo. They simply open up everything instead of only Apple's App store installers. If they are going to gouge people for Internet access, they ought to get their technical shit together.

Happy flying, be safe, and enjoy.

Accessing Admin Privileges:

A Quest Through One of Mac's Backdoors

by NerveGas Jr.

Using Security Against Itself 0x1

Modern Apple computers boot up straight to the volume "Macintosh HD OS X, [version number]". By accessing this volume through single user mode, one can reset passwords on an admin account without the initial admin password and without needing to get knee-deep into coding. This feature is usually used to troubleshoot problems in an easier way, but one could also access administrator privileges without a password. Below is a way one can go about it.

Resetting an Admin Password 0x2

First, reboot the computer. After the screen lights up, hold down CMD + R as the screen shows the loading up of the system. It is crucial to start pressing CMD + R before the Apple icon shows up with the loading bar, If you don't do it in time, the computer will most likely boot into the default mode.

After this, the system recovery partition shows. The dialogue box should be asking you to select a country and/or language. Double click on the desired option. Then, another dialogue box will appear with different utilities one can use in the different cases of crap that need attention. Let us ignore this. Instead, click on "utilities" in the menu bar. Three things will appear: "Firm-

ware Password Utility", "Network Utility", and "Terminal". Double click on Terminal.

In this specific mode, the Bash commands are different from those when the computer is booted up into the multiple user mode. Keeping that in mind, type in the following:

```
$ resetpassword
```

That, my fellow people, is the only amount of coding you must do to reset a password. You will not even be prompted for the initial password of the admin! After this, a new dialogue box will show up. Select the volume for which you are changing passwords. Usually there will only be one, which is "Macintosh HD OS X, [version number]". I am doing this on Macintosh HD OS X, 10.10.5.

This is where you get to change passwords. Click on the admin account that you want a new password for. It will prompt you for the new password, and then again to verify it. The great thing is that if you forget this new password, then you can go through the process again, resetting the password once more. After this is done, you can enter a password hint, but if you do this, then the true administrator will more easily discover that the password was reset. If there was a hint before you restarted the password, then it would be wise to set the new hint to that one.

A dialogue box will show up after you press reset. Press okay on this after reading it.

Now go back to Terminal and type in this command:

```
$ reboot
```

After rebooting, you should be able to access the admin account!

Accessing Admin Privileges Through Root 0x3

Rather than changing the password for the admin account, you can instead set a password with the System, Administrator (root) account. This will automatically enable it to show up in the login screen. Eureka!

One can enable and deactivate this by going back to the dialogue box where they changed the user's password, and then deactivate it that way. Sadly, this take a long time, and some suspicion can be aroused if the true admin is shoulder surfing at the time.

Also, one could enable it through Terminal in the already hacked admin account. This is easier and more efficient. In order to do this, go into Terminal and type in the following command:

```
$ dsenableroot
```

After this, you will be prompted for the admin password. Type in the password, carefully. Then, you will be prompted for the root password. If you have not set up the root account, then type it in. Neither a hash code nor the text letters/numbers will show up. This is great for security purposes, but if you screw up, then you have to type in the command again, and enter the password again. The only problem with this is that you cannot enable the root account through any other user besides that of an admin user.

Disguising the Root User as a "Non-Administrator" 0x4

If you want to take the most precautions possible in resetting the passwords, you can choose to disguise the root account as a "non-administrator" account. You can start this by changing the name and profile picture of it. After doing that, you can deactivate the password on the root account, unless the legitimate Administrator is fine with you putting a password on your account. Doing these things enables you to have control over the restrictions of your account, furthermore concealing the fact that you did anything.

Using a Firmware Password 0x5

Apple created the ability to have a firmware, or BIOS, password so as to "[prevent] your Mac from starting up from any device other than your startup disk." You can set this up easily through recovery mode. First, boot up the computer to recovery mode. After this, click on the Utilities section. Rather than going into Terminal, double click on the Firmware Password Utility. From this you can set up a firmware password that will make it even harder to get into the BIOS settings.

The firmware password has not been successfully cracked into without taking apart the computer (as of October, at least). There must be a backdoor through this and with the new Mac update OS X El Capitan, I am positive that many of you who read this will take up the challenge to crack into the firmware password without having to take the system apart.

Conclusion 0x6

This hack is surprisingly not known as well as one would be led to think. The best thing to do with this knowledge is to tell other people about the backdoor and attempt to get them to use the firmware passwords. It would be nice to figure out how to reset the password without needing to either break it apart or take it to an Apple store in case someone forgets the firmware password.

PERSPECTIVES ON CYBER SECURITY

by Super Ells

The way cyber security has changed through the digital age, going from simple passwords on S/360s interfaced through dumb terminals to multi-factor authentication, routing and firewall security, and even shredding paper to counter dumpster divers and social engineers has, overall, not really increased security. The ways that security has been "improved" have done very little to truly improve cyber security. Over the decades, people from Kevin Mitnick to Edward Snowden have consistently been able defeat security measures, as have organizations from governments eager to spy on its citizens along with hacktivist groups such as Anonymous. A complete paradigm shift must be made in order to improve cyber security. The days of making networks a "vault" are belated in their inevitable demise.

Cyber security has been "improved" in three ways: encryption, layering (multi-factor authentication, complex passwords), and access restrictions (security clearances, physical security, need-to-know, access permissions). Of those, the only one that has been successful is encryption, enough that the U.S. government freaks out about it - from considering it a munition in the 1980s to the FBI director asking the American people to accept being spied on.[2] Encryption, when properly implemented, has been the most effective tool for security, and with encryption tools such as PGP and AES that are extremely strong, it is not only extremely difficult to crack, but also widely used.

The second way is layering. Multi-factor authentication, though a good idea, has its weaknesses. Cell phones can be spoofed to get text message security codes, CACs can be copied as well as other card-based access mechanisms, and users have the risk of losing one of the factors - and that hampers productivity. Also, increasing the complexity of passwords has allowed tools like Apple's Keychain to proliferate, as well as convincing more users to write down their passwords. However, from many anecdotal stories and security assessments, it is routine for just about any sysadmin, anywhere, to grab 30 to 40 sticky notes a week with user passwords on them. Some of these passwords were found to be able to access supercomputers, mainframes, and even web-based email accounts. The sysadmins would then let the affected people know not to leave their passwords out. Even with utilities like Keychain, you can extract a user's Keychain and grab every password they save, further compromising security.

This leads into the human factor, and how it can be exploited - the fine art of social engineering. The human weakness is always the biggest weakness; even with extensive (and annoyingly repetitive if you work in the U.S. government) training, it is still a large problem. Even security clearances have issues; they can detect weaknesses and deception, but they cannot detect true human intentions. Even the use of polygraphs is not often effective. If anything, their use is far from it. Edward Snowden and Chelsea Manning are the two most recent examples of why just simply being cleared doesn't mean that you have brought in an insider threat, and people who may be thought of as insider threats because they don't "play the game" or "act normal" due to being eccentric or culturally different may be the best people to have. Shredding documents has become a deterrent to dumpster divers - until they start looking for old hard drives, CDs, memory cards and USB sticks, or even intercepting Wi-Fi transmissions.

Even more interesting are systems like the Pwn Plug that can be plugged in a back room and used to extract data without being detected easily.[3] Even VoIP can be compromised and used to listen in on unsuspecting people.[1]

The best, and most efficient ways, of extracting information from users and compromising security is still the simple phone call, acting like a colleague or an IT support team, and getting the information from the unsuspecting user that way. Spear phishing is still effective, but it is losing its effectiveness due to counter-spoofing measures. Government agencies ban the entry of cell phones

with cameras into certain facilities, but there is no way to legally trace phones without a Stingray (and even then, it's legally dubious). Cell phones are so small and easy to hide - the smallest GSM phones are the size of a credit card. It is a matter of trust, and more times than not, people bring them in. Many government offices that ban the use of cell phones have found that, because of the inconvenience of trying to enforce the policy, it's easier to simply not say anything about them unless they're blatantly visible out in the open.

The effects of increased cyber security through the above mentioned ways are very profound and simple to express. It inconveniences the normal working body of people by forcing them to go through one layer of security after another just to be productive, while building a structure for people who want to get information or intelligence that's somewhat difficult to penetrate, but isn't enough to discourage them from trying. Also, cyber security is very reactionary instead of proactive. Policies can change drastically because of one incident, and not even in the right way. Flash drives were banned because of Chelsea Manning. It does not make any sense, since Chelsea Manning should have never been able to keep a security clearance, much less be deployed, due to a myriad of issues. Worse yet, and typical of the reactionary implementation of cyber security, he burned the leaks on CD-RWs named "Lady Gaga," for example. Does that have anything to do with flash drives? Absolutely not. Does it make the "cyber security" professionals look great? Wonderfully so. However, they are so deceived in arrogance that they cannot shift to another paradigm about security. That arrogance blinds them from the most crucial element of security: the human element. You cannot eliminate it, but you must be able to mitigate it.

How to change it? On private networks, it is best to use a combination of items to mitigate the human element. There is no need for two-factor authentication unless it can be easily usable, reliable, and most of all, stable. It's good to have your standard firewalls and IDS, as well as good malware protection. But the main difference is that instead of locking every single thing down, you only need to lock out what you wish to lock people out of (keeping people out of others' personal folders, for instance), and keep the rest open. Use port control on your switches, and lock to MAC addresses. The most important approach to change is to implement a file transaction logging system. This allows the ability to identify and catch a problem in open view, since every file transaction, program accessed, and file location access is logged. With proper user management and port controls, if information leaks out, it is easy to trace out who did it, and pursue action against the offender. Vigilance is most important in the endeavor of security. You need to constantly flag and monitor what is going on in the network. Using tools such as transaction logging will allow security managers to be able to assess in real time what is happening on their networks, and by whom. Then, when breaches happen (and they will, and it's not worth the effort trying to stop them - it's best to just mitigate the spread), you know who was the perpetrator. With the use of strong encryption for inter-network and off-network communication to remote workstations, and without monitoring the workstation itself (improves privacy while maintaining security of on-LAN files), this is the most effective approach to managing network security.

In conclusion, an open and transparent approach to security without impeding productivity should always be looked at and, with this outlook, a paradigm shift needs to be made. The old methods of implementing cyber security are on their way to irrelevancy; and recent events have made it necessary to be able to guarantee a level of privacy while maintaining a level of security.

References

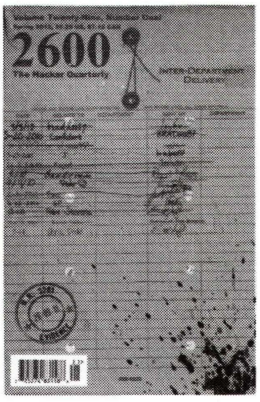
Malvineous. (2014, Autumn). "Bugging a Room with an IP Phone." *2600: The Hacker Quarterly*.

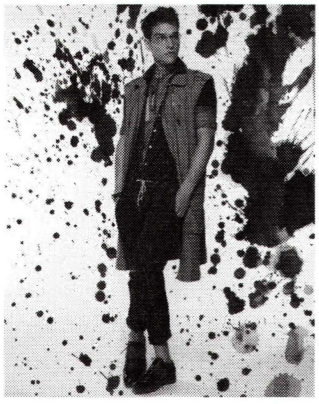
Lemos, R. (2014, October 16). "FBI Director to citizens: Let us spy on you." Retrieved from *Ars Technica*: http://arstechnica.com/security/2014/10/fbi-director-to-citizens-let-us-spy-on-you/

McMillan, R. (2012, March 3). "The Little White Box That Can Hack Your Network." Retrieved from *Wired*: http://www.wired.com/2012/03/pwnie/

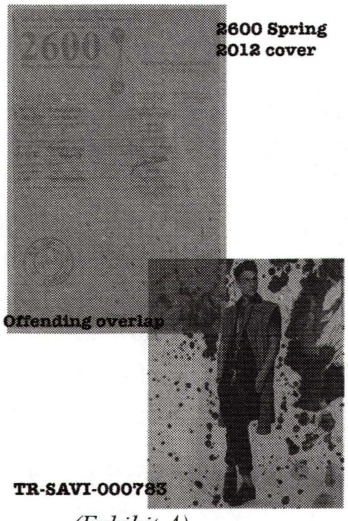
THE SPLOTCHGATE SAGA

It was sometime over the summer when we received the first threat. We get our fair share of nasty letters, so it's not always possible to give each one the attention it deserves. After a couple more of them came in, it didn't take long for us to realize that this one was something special. The letters came from License Compliance Services at an establishment known as Trunk Archives. They said our Spring 2012 cover was considered copyright infringement and that we owed them $714. As evidence, they showed us two images that looked nothing like each other.





Then we looked a little closer.

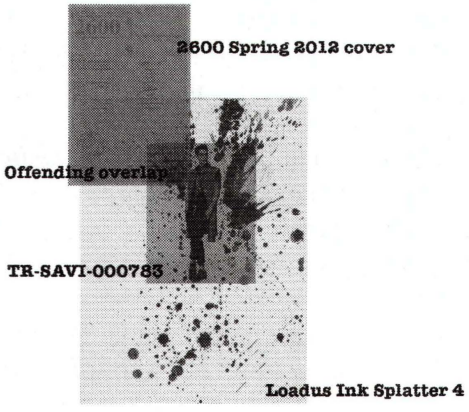


(Exhibit A)

Yes, that's right. Their sophisticated computer algorithms had detected a match in a small part of an ink splotch graphic. This is what they spend their time doing - searching the entire Internet for any instance of matching imagery. It doesn't stop there, of course. Copyright trolls (as such entities are disparagingly referred to by many) seek to find similar instances in audio, video, and anything else that can be represented in digital form. If we gave them only a little more leeway, we're certain they'd be able to claim the rights to specific colors and musical notes. This kind of thinking is why simple creativity is in a constant battle with corporate might. It's why Google flags your YouTube videos, why Internet radio is crippled with fees for basically helping to promote artists, and it's why you'd better be sure you're all paid up if someone even *whistles* part of a recognizable melody in that documentary you're making. The scariest part is that every year the technology to do this sort of thing gets much better.

We were all too aware of how insane this practice was - and we knew we had to fight back. When a company can seriously threaten people over an ink splotch, it's time we all had a little reality check.

But then we discovered something even more surprising. We did a bit of investigating (at great time and expense, not to mention the emotional toil) and found that the source of the ink splotch was a Finnish artist named Loadus who had specifically made it public domain - free for anyone to use. Somehow Trunk Archives got it into their heads and algorithms that they owned the image. But they didn't. They had merely used it as a background for one of their images and apparently didn't bother to differentiate.



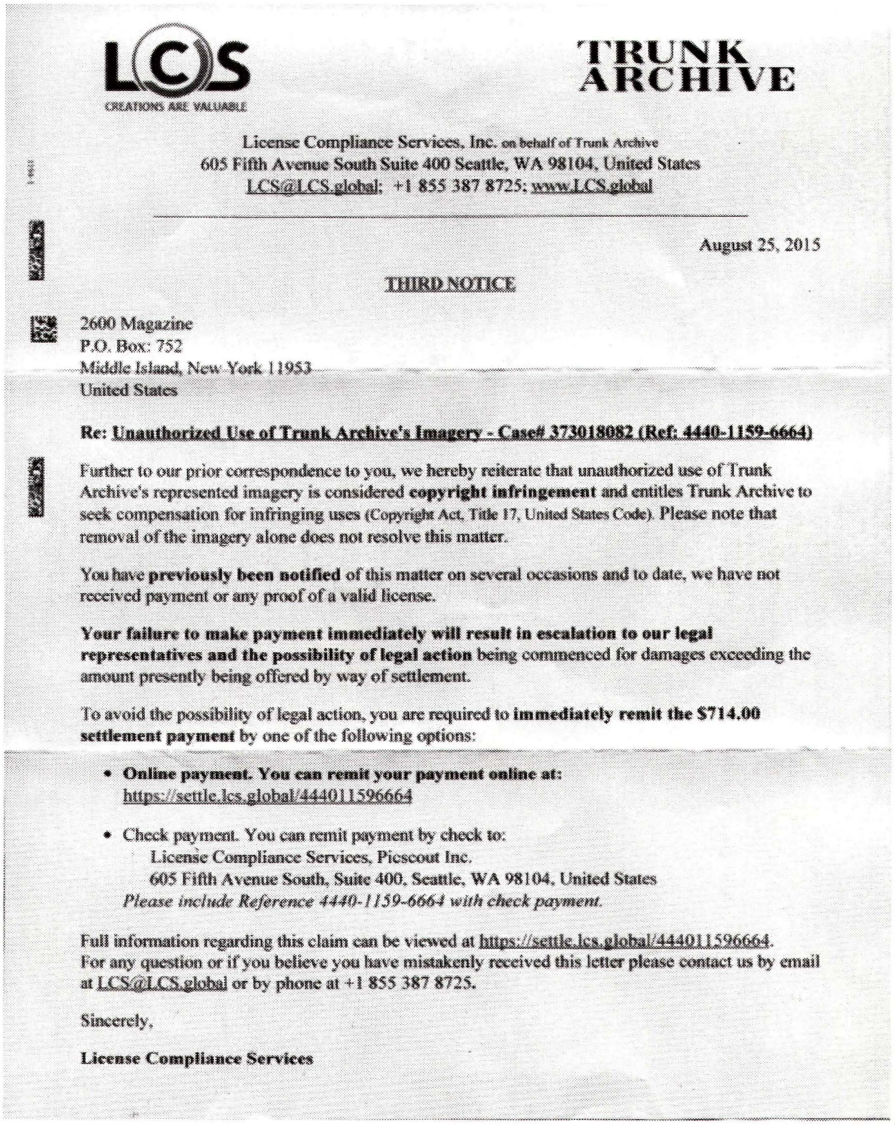
(Exhibit B)

So not only were we being threatened for something as absurd as an ink splotch, but we were being threatened by people who had absolutely no rights to it in the first place! At least, no rights beyond what everyone else in the world also had. In a way, it was disappointing because the case suddenly became so ridiculous that we never had the opportunity to actually fight to win it.

At this point, the Internet stepped in and spread the story far and wide. It didn't take long for us to get a terse note from License Compliance Services which simply said "Hello, I just wanted to take a moment to inform you that after further review this matter has been closed." That was followed several days later by an actual apology from the Chief Operating Officer at Trunk Archives, who wrote: "As COO of Trunk Archive, I would like to offer my sincere apologies for 'Ink Splotchgate.' Artist attribution and copyright protection are very important to us, so we truly regret this case of 'mistaken identity'. Using a digital copyright infringement service is a relatively new thing for us and we have learnt a lot about what can go wrong through the comments made by *2600* and the community at large. Thanks for all the feedback and please feel free to reach out to me with any questions or thoughts on this matter. Once again.. very sorry for our error."

While acknowledging this problem is a good first step, it's still just a first step. The bigger problem remains. Threats like these go out constantly for absurdly minor uses of material that are often completely protected as transformative work under the fair use doctrine. The average person doesn't have the time or resources to do battle with these accusations and so, far more often than they should, they wind up paying the demanded amount. This has got nothing at all to do with compensating artists, who are often horrified to hear of the actions being taken allegedly in their names. This is about intimidation, power, and the quelling of creativity. We would have an interesting time going back in history to see how many derivative and transformative works would never have been created had entities like Trunk Archive, Getty Images, License Compliance Services, Picscout, etc. been around to silence them with their digital services.

This little episode woke us up to the danger. We hope documenting it here helps to get a lot more people involved in fighting this unhealthy trend.



LCS
CREATIONS ARE VALUABLE

TRUNK ARCHIVE

License Compliance Services, Inc. on behalf of Trunk Archive
605 Fifth Avenue South Suite 400 Seattle, WA 98104, United States
LCS@LCS.global; +1 855 387 8725; www.LCS.global

August 25, 2015

THIRD NOTICE

2600 Magazine
P.O. Box: 752
Middle Island, New York 11953
United States

Re: Unauthorized Use of Trunk Archive's Imagery - Case# 373018082 (Ref: 4440-1159-6664)

Further to our prior correspondence to you, we hereby reiterate that unauthorized use of Trunk Archive's represented imagery is considered **copyright infringement** and entitles Trunk Archive to seek compensation for infringing uses (Copyright Act, Title 17, United States Code). Please note that removal of the imagery alone does not resolve this matter.

You have **previously been notified** of this matter on several occasions and to date, we have not received payment or any proof of a valid license.

Your failure to make payment immediately will result in escalation to our legal representatives and the possibility of legal action being commenced for damages exceeding the amount presently being offered by way of settlement.

To avoid the possibility of legal action, you are required to **immediately remit the $714.00 settlement payment** by one of the following options:

- **Online payment. You can remit your payment online at:** https://settle.lcs.global/444011596664
- Check payment. You can remit payment by check to:
  License Compliance Services, Picscout Inc.
  605 Fifth Avenue South, Suite 400, Seattle, WA 98104, United States
  Please include Reference 4440-1159-6664 with check payment.

Full information regarding this claim can be viewed at https://settle.lcs.global/444011596664.
For any question or if you believe you have mistakenly received this letter please contact us by email at LCS@LCS.global or by phone at +1 855 387 8725.

Sincerely,

License Compliance Services

Proclamations

Ideas

Dear *2600:*

I have been toying with an article for some time. The idea initially occurred to me after the fallout from the Edward Snowden affair. I have sought an unbiased publisher, but the government rags (in which its publication might actually do some good for Uncle Sam) are too wedded to incompetent vendors. The article has to do with why our information security capabilities are in the state that they're in and what could have been done about it - if our government cared one whit. Developments make it crystal-clear that they have already surrendered their technological and military superiority to China and, moreover, are expending ever less effort on even putting on a "show" to caring about computer security. Meanwhile, each "expert" we see is more readily buffaloed than the previous buffoon: an RSA "consultant" was on Fox News the other day who (a) didn't know what RSA stands for and (b) explained a recent hack as - get this - the attackers "went into" the system. "Went into." Here's an attached resume to indicate that I'm not some abject moron. Something tells me you will find it as unique as I know my written perspective is. Trust me, I'm nothing like the others, and I long since tired of lifting a finger to help our government. From a technical perspective, I won't go into the details of ad hoc hacking techniques as it were, but I have plenty to share on the underpinnings of high-assurance military systems, which - I guarantee - are way beyond the lion's share of your readers, both from the historical exposure perspective and the formal mathematics perspective. Don't be so quick to dismiss everything that comes from DoD because Joe Schmuck leaves a guest account undeleted.

Name Deleted

We've gotten a few such letters, but what we really want to get our hands on is an article! Resumes aren't necessary. Just write about what you know and submit it. The community will thank you.

Dear *2600:*

America should nuke the evil Commie Chinks, which kill Falun Gong members, persecute dissidents, and occupy East Turkestan, and Tibet! Blessed be Lord Our God, George Yahweh Washington, Saint Thomas Jefferson, Saint Thomas Paine, Saint Patrick Henry, Saint John Hancock, Saint Benjamin Franklin, Saint Paul Revere, Saint Betsy Ross, Saint Martha Washington, and Christ Baden-Powell, Our Lord! May the American Master Race rule the cosmos/omniverse forever and ever! Amen!

Hugo L.

And then we get these. Readers, please help drown this drivel out with some intelligent discourse. We have so much of note to focus upon and you have some of the most enlightened perspectives currently out there. Most of you, anyway.

Dear *2600:*

I'm loving the magazine, and keep it up! I have a couple of articles I would love to send in, but I want to make sure no one has written them yet. I have one about OPSEC (Operational Security), and one about how to access the root user on a Mac - getting admin privileges in the process (without needing a different admin password).

NerveGas Jr.

Trust us, you won't be duplicating the efforts of others. He who hesitates is lost and never winds up getting published.

Dear *2600:*

Seems like this could be a *hot* topic for *2600* - how to effectively combat junk/spam/robo phone calls.

They are a monumental pain in the ass. There are (at least!) a dozen or more websites devoted to "reporting" them and (helplessly) screeching about them. They too often sucker some naive recipients into scam deals - sometimes (often) even extracting life savings from some hapless victims. At the least, residential landline phone users could/should be protected. Ditto for at least non-biz cell users. There *have* been recent Senate subcommittee hearings about them!

There *are* some partial - maybe total - solutions to them (aside from making them "illegal," which is about as effective as making petty theft illegal). Some, but not all, involve "forcing" the telecom cartel to do something easy (e.g. verify

calling numbers before passing along fake Caller ID, etc.).

Other possible partial or total solutions are techie/gadget-based. They would surely be of interest to entrepreneurs (as well as white-hatted black hatters).

Just a thought. Although I'm a 50-year geek, I am *not* an expert in this area!

jim

While we share the impatience and concern over such annoyances and hazards to the easily convinced, we must also be wary of solutions that ultimately cause more harm than good. Losing the ability to disguise one's phone number would inevitably make it much harder to communicate anonymously. Despite what we're taught, this is a valuable skill to have. It's of particular use to whistleblowers as well as anyone who's trying to avoid unpleasant people - as well as those of us who simply value our privacy. There will always be those who use technology for sleazier purposes who will stop at nothing to make a quick buck. There are numerous clever ways to not let such people gain the upper hand, just like there are in every facet of the Internet. We agree with the idea of making solutions through technology because that allows for customization and evolution without falling victim to a bunch of draconian laws.

Article Feedback

Dear *2600:*

Really liked the Metrocard article. I've looked into this in the past a bit, but was always turned off by potentially wasting money on a reader (glad I didn't!). I grew up in Queens, so work is especially interesting to me. Is there any way I can help? It would be cool to extend the project to other systems as well in other cities.

Josh

All it takes is research and the interest to pursue the subject, as well as the desire to tell the world what you learn. We don't expect people to know everything about a particular subject. The only way we learn is by sharing what we do know and encouraging others to fill in the gaps. All too often, we encounter people who doubt the value of their input and wind up delaying their participation until after the info is fresh. Or they lose interest and never submit anything. Countless times we've learned that getting the conversation started is what leads to a fuller understanding. And, as you see, that conversation can last for a very long time.

Dear *2600:*

Regarding a discussion in the Summer 2015 "Letters" section, child pornography is documentation of actual acts. As Asia Argento, director and star of her own real-sex feature, *Scarlet Diva,* once said, the *point* of watching porn is to have the "this really happened" experience. Ergo, it's a bit hard to stomach the filing away of the legitimately-worth considering sorts of distinctions - i.e., erotic paintings featuring nymph-like children; the first-ever conviction in America for "obscenity" of *Boiled Angel* cartoonist Mike Diana in the later 90s to scant attention in even the *alternative* media; the "ruining" of "lives" erotic photographer Richard Kern describes, of clearly over-18 girls who model for hardcore porn - into the intellectual dustbin (as happened in this discussion) along with the exceptions which have no ambiguity about them whatsoever.

Entrapment's entrapment, as a crime practiced by law enforcement officers; similarly, degrees of crime sentencing should be, in a sane society, debated as to proportion vis-a-vis other violent, damaging-to-others crimes. If this is not the case, however, one feels a little queasy about the "benefit of the doubt" being extended to consumers - as though the age-old "coke users *should* feel guilty about fueling cartels' profits south of the border" bromide, for instance, doesn't apply to something that is, quite frankly, ethically behind the pale.

Pete Townshend, it turned out (for those of us who heard later), had posted on his website a warning to people who knew him and knew of him that he fully intended to run the risk of downloading massive files of child porn to explore, research, and pursue the perpetrators and distributors - and did, in fact, have to face the music for this daring feat - no pun intended. (As any fan of *Tommy* with knowledge of Townshend's conceding in interviews that he, too, had suffered from such abuse as a child, this sort of priority on his part was hardly surprising.)

Heaping additional scorn on offenders out of spite isn't worth doing, and only feeds a blackness of the heart; that, too, *should* go without saying. But - in the opposite sense of the way the phrase is usually invoked - let's just leave them *be,* okay?

No, it's not minor. Files aren't just files if their very existence is a crime.

Leave *that* at *that!* (Other than that, your journal's irreplaceable, for what it's worth!)

With what I hope is only-the-appropriate paranoia, I've asterisked words that won't make

my email and address crop up in some Fed's filter for assholes. As text, I'd intend the words to run in full; if you're going to run this letter, please know I'd rather it read with the full words, intact. Can you blame me?

Smiley McGrouchpants

That last bit really illustrates the point we were making in that issue. When you can't even write the words "child pornography" out of fear, that's something that needs to be looked at. Yes, the files are reprehensible, no question. But we can't simply make everyone who finds a way to copy such files, regardless of the reason, guilty of the crime involved in their production. If we do, then why don't we also make it a crime to possess videos of people being beheaded? That is a reprehensible action as well and its distribution certainly helps to encourage the perpetrators. Yet we turn a blind eye towards the easy availability of such material. The point is if people want to see such content, that's a problem that needs to be addressed head on. We don't solve this by ignoring it nor by classifying everyone with the same broad brush. And we definitely don't get anywhere by being afraid to discuss it.

Dear *2600:*

I bet the cover of the Summer 2015 Magazine generated some interest. Looks like some USA federal government employees are finally earning their fat salaries and pensions. What is the back story? Did it come from the CIA, DIA, or NSA? The symbol inside the star basically means "sneaking in." What is the significance of the three-concentric-circle symbol? The latitude and longitude were helpful in locating the building housing these Chinese military government employees. (Whoa... we have a spy versus spy comedy forming here.)

Didn't Premier Xi say recently that China would never condone government sponsored hacking? I will take a democratic republic funded by a capitalistic economy over communism any day. How can we help the Chinese revolutionaries in Hong Kong?

Webspider

We appreciate your noticing the details. But we can't really discuss it until the digital digest for this volume comes out in mid 2016. (All covers get explained in each year of the digest, incidentally.) And the best way to help people anywhere is to pay attention to what's going on with them and to get the word out to those who will listen. We often have much more power in that department than we realize.

Dear *2600:*

This is about the article explaining security issues in Brazilian voting machines (32:1). I just met Diego de Freitas Aranha, a researcher from the University of Campinas, SP, Brazil, who helped to check some issues with Brazilian voting machines. After some talk, I sent him the article and he emailed me back some remarks and corrections (though there were good things there). Here is the text which I translated, which includes some important mistakes:

"The voting machine has run GNU Linux for a long time and the software is no longer produced by Diebold (only hardware).

"The University of Brasilia was not 'hired,' but won a public tender competition with other institutions. The attack on the secrecy of the vote was mounted on public information without a need to change the source code because the restrictions imposed by the TSE prevented that. I coordinated the team.

"There is no evidence that 'Rangel' in fact changed election results - there is much politicking in the case."

He also send me this link with the English version of his report, concerning software vulnerabilities in the Brazilian voting machines, available at https://sites.google.com/site/dfaranha/projects/report-voting.pdf (English) and https://sites.google.com/site/dfaranha/projects/relatorio-urna.pdf (Portugese).

Derneval Cunha

Dear *2600:*

I am writing to provide some clarification to you and your readers on the .mil domains listed in the Ashley Madison article in issue 32:3 (Autumn 2015). Looking at these "domains," it becomes obvious that many are not at all domains. Many of these are the username portion of .mil email addresses. The military has changed over to a new email address naming convention. FirstName.M.Lastname.mil@mail.mil for military folk. They use similar setups for civilians and contractors: FirstName.M.Lastname.civ@mail.mil for civilians and FirstName.M.Lastname.ctr@mail.mil for contractors. This makes it very easy to quickly remove many of the items as username portions of email addresses just by looking for the username pattern.

Someone with a .mil email account could very easily run through the ones that look to be email usernames and verify them against the address book that is available to users. Someone on the outside could also send test messages looking for bounces or lack thereof by adding "@mail.mil" to the end of any that look like

email address usernames.

Nobody should fear that I'm giving away some state secrets here as all this info can be easily found with a bit of searching the web. For example: https://gcn.com/Articles/2011/02/04/Army-Begins-Move-to-Enterprise-Email.aspx

I was happy to see that I didn't recognize any of the email address usernames.

Enjoy.

Phreak480 from Long Island
The Home of 2600 Magazine

To clarify, we knew from the start that many of the so-called domains were simply what people typed in on the Ashley Madison site, which provided no verification. If there are people dumb enough to use that site as well as enter their real names, then it stands to reason there are people dumb enough to give out some juicy top level .mil domains as well. At least, that was our hope.

Dear *2600*:

I was excited to get my Fall 2015 issue of *2600* and see what I thought was a picture I emailed you folks years ago and was surprised to see you needed to do detective work to find out where it was because I had already told the whole story. (That's what gave away the fact that it wasn't mine.)

Specifically, that motel is on Lincoln Avenue on the far north side of the city. Along that stretch of Lincoln Avenue is a series of tacky motels (some of which have been torn down but with the signs still intact) that today probably offer hourly rates but were undoubtedly gold mines in the summer in the days before interstate highways.

Might not be the best place for a convention; I recommend staying at the Hotel Penn for now. But what I did find sly was that one of the former sites of the Chicago *2600* meetings was at the Boys and Girls Club, which is bordered on the west by Rockwell Street, which in the Chicago street addressing system would be 2600 West.

The More You Know

Edgewater Sean

It's truly amazing that so many people are literally looking out for us.

Dear *2600*:

Learning to obey the laws of the land is an uphill battle considering who my teacher is. You are continually putting good stuff in *2600*. Re 30:1, the Raspberry Pi article is my type of thing. So when are Beowulf jackets going to show up? In "A Lost Promise" we can't disregard the lesson it relates. The paragraph starting "Recognizing the signs of someone in trouble" speaks by relating to me aspects of myself. I have been dreaming in Unix: "rm r *.*". The SCDC rules changed, saying we can't place pen pal ads. Prison is prison. The main thing I am grateful for is we can correspond with anyone except for fellow prisoners. We need people who will step it up, be our advocates and proxies. Few people are assisting us. We're last in line for most. Extra-legal harassment is an institutionalized art form with guards delivering panoramic displays. "Hypercapitalism and Its Discontents" points to the common need for support for important global issues. Everyone should pick a message that needs to be told, then do it. Don't assume the facts you see on public display tell the whole story. The establishment counts on you to see things their way. Look past the open/closed community debate and consider. Are we ready and willing to change? The future, multi-generational, self-sustaining, constantly changing, multi-faceted, networked networks, connected/disconnected, anonymous, and public. Today is the future's black and white TVs. I do wonder if I'm finished now!

Cypher2x aka James E. Anderson #283022
Tyger River Correctional Institution
200 Prison Rd Unit 6-9B
Enoree, SC 29335

Dear *2600*:

In response to joshua's artificial intelligence letter to the editor in 31:4, I should make the comment that a human baby has only two hardwired words to it. Those words are "mom" and "cup." All further XML statements depend upon the parsing of phrases to these two words.

There is a temporal value that the human brain holds in long term memory. The human body is the computer, holding the hard drive that is the brain with primary input the eyes, and primary output the larynx. So the input and the output are fuzzy. The emotions are of the spine. So can the body endure with strength, spineless or otherwise.

There was an interesting feature in the original Apple II microcomputer that Steve Wozniak designed. Upon power up, after all internal housekeeping was set, the microprocessor ran the BASIC program named "START" so a basic program could execute. This was a powerful feature to streamline user applications.

When Joshua is run, '"mom" and "cup" is the equivalent of the Apple II "START" - a self-programming computer that knows the spinal scheduling emotional subsystem of itself could "goto" and "let" to its heart's content. The heart

of such the interrupts and the reset hard and soft.

The heart of the computer? Simply the quartz crystal that is at the center of synchronization timing that oscillates (for the Apple II, one megahertz) (at today's clock speeds in the many gigahertz) for the benefit of resonant data. The same quartz that is at a center for new age activities. The same quartz that converts mechanical energy into electrical energy, and from electrical energy to mechanical energy. All computers use clocks, and clocks are of vibrating quartz; therefore, all computers use vibrating quartz.

The computer could self program for the benefit of its own clock. In this way does the computer have heart. And the heart; then, of love, which is what we all need, want, and desire anyway. The "request to parse request" somehow in its own resonance.

John

"Cup?"

Humble Requests

Dear *2600:*

I have checked and found that someone is trying to misuse my personal detail as given on different websites. I request you to please remove all from Google. URL is given below.

Vipin
West Delhi

We don't know what people are saying about us over there, but we can assure you we do not at present have the power to "remove all from Google." We have no idea how these things get started.

Dear *2600:*

I am a new subscriber but I have no experience in hacking or computer programming at all. I am desperate to learn and I was wondering if you could teach me or tell me the best way to learn. Thank you.

The Prince

Apart from people thinking we have superhuman abilities, we also often get requests like this. We want to be encouraging, but we also have to be quite clear that hacking isn't something you just teach. Computer programming is. So if it's the latter you're after, you'll find answers in classes and tutorials, both online and in person. But as for the hacking part, that is something that has to come from within. There's no class in the world that teaches you that. If you have the passion and curiosity, that is what you build upon with the knowledge you gain from exploring technology, asking lots of questions, and never giving up. Diving into these pages will at least give you a sense of what that's all about.

Dear *2600:*

I have attended a couple of meetings, but it has been some time since I have been to one. A client of mine needs some work done on her computer that would require your expertise. I am hoping you could help her out - I couldn't imagine it being too terribly hard for you with all your knowledge. It pays. Please call me as soon as you can to discuss further.

Jacob

And then there are countless letters of this type, which is a variation on the first one. We don't know everything and we're not always interested in doing this sort of thing in the first place. But you might very well find someone at your local meeting which seems a much better place to ask this sort of thing than here. For that matter, you can find bright people who can work on computers all over the place. If it's some sort of "hacker" magic you're asking us for, you'll need to be more specific so we can mock you with better accuracy.

Dear *2600:*

I was wondering who I need to talk to about permission to create a static copy of the *2600* meeting information to distribute in Cuba.

We are hosting an ICT Security conference in Havana this winter and thought it would be great to start a *2600* meeting there and present your meeting information as a White Paper in the conference proceedings.

L.

That's a great idea and we've sent you the info you need to pursue this. Hopefully others will think of equally creative ways to open up the hacker world even more.

Dear *2600:*

Please block the word "Puti" and "puti lado" from Google Instant while I search words starting from "P" or "Pu" because these words are not accepted in our society.

Thank you.

Kalyan
Nepal

And we're back to this. A number of years ago, we discovered that Google Instant (that feature that finishes words for you in the Google search bar) wouldn't finish a number of words that Google apparently considered controversial. So we printed a whole list of them. (You can see the list we made before we lost interest and got back to our lives at www.2600.com/googleblacklist/.) Words like assmunch and swastika wouldn't yield any additional suggestions for the search, although the search itself worked. Somehow this revelation morphed into

people somehow thinking we were in charge of this and a whole bunch of requests like this one. Again, there's nothing we can do, other than help teach the world another couple of words never to say when in Nepal.

Meeting Mania

Dear *2600:*

I've been attempting to resurrect the Melbourne, Florida *2600* meeting. I've gone, as proscribed, to the proper location at the proper time twice in the last three months.

The first time I was a little late so I went around and bothered every group of people I saw at the coffee shop, but none of them were even aware of a thing called "2600." Not only that, but I'm a bit outside of the age demographic for that coffee shop at that time of day, so I got to look like a slightly creepy old man attempting to hit on college kids. One of them even called me "sir!"

The second time - just this past Friday - I got there early. I set up shop in a prominent location, booted up my Kali Linux laptop, and placed a couple of *2600* magazines out in the open. One individual did approach me, pointed at the magazines, and asked, "What is that?" A second or two into my explanation, it became clear he was actually more interested in the bowl of hummus and pita that was waiting right next to the magazines.

Although I did enjoy my time at the coffee shop and got quite a bit of work done on a new article for *2600,* I was hoping to actually interact with some like-minded souls.

Mike

This does happen on occasion and it's a part of the whole community-building process. It can often take time and patience for a group to actually form. Sometimes existing groups disband without new ones taking their place. Most frustrating is when groups move to other locations and forget to tell us! Whatever the situation, we try and provide every opportunity for the community to grow. Obviously, we don't wait forever. We hope this one works out - please keep us updated.

Dear *2600:*

I am the founder of Proto Makerspace. I am wondering if you all will allow me to host a *2600* meeting henceforth at our space in north Houston. The *2600* scene is not active in Houston any more and I wanted to revive it.

Roo

We're glad to see the interest. But right now the Houston group has a web page up that continues to show updates for the original location. If we hear otherwise from a number of people, then we can consider the change. We do advise meeting in a public space that fosters conversation, not only between existing attendees, but entirely new ones who may have never seen a hacker before. Going to a hackerspace or equivalent afterwards combines the best of both worlds. This is merely our suggestion, however.

Dear *2600:*

Is there an active chapter in Edmonton still meeting on Whyte Avenue? Is there a contact member I can speak with here?

Ken

That meeting is active from what we can tell. We don't give out any personal information for anyone involved in them, however. If a group has a web page, there may be contact info there. We are also building a Twitter network of meetings around the world, so following @2600Meetings might be the best way to establish contact with people involved in local meetings.

Dear *2600:*

For the most recent November meeting for *2600* in Chicago, I went to the specified meeting location. The proprietors of the establishment had never heard of the meeting, and I couldn't find anyone there. I did see a couple of people that could possibly fit the bill, but I didn't want to harass anyone just having dinner.

Is the *2600* meeting still happening there at 6 pm? Or do you know how to get in touch with the meeting organizer or how to be able to tell if a particular group is with *2600?* I was looking for the magazine, but I didn't see anyone with it on the table.

pi

As you may know by now, that meeting has changed to a new location and is listed in this issue. Since we come out quarterly, we may sometimes have inaccurate info if such a move takes place. We hope to have quicker updates online. (For the record, it's always a good idea for at least one meeting attendee to have a copy of the magazine out or a hacker shirt on so people can make contact more easily.)

Splotchgate Comments

Dear *2600:*

I would like to comment on your issue with Getty Images (owner of Trunk Archive).

First, this is a practice Getty Images has engaged in for years. A client several years ago received a demand letter from Getty Images for a thumbnail image used on his website. The image was part of a design that had been properly li-

censed from another party. Getty Images refused to accept that license as indication of "good intent" or to take action against the (larger) company that had sold the template and license. The amount demanded was much more than it would have cost to license the image from Getty to begin with and the client ended up shutting down the business to avoid paying this ransom amount. I have heard similar stories from other web designers (purchasing legitimately licensed images).

Second, Getty Images used to be only one (overpriced) player in a diverse market. They have been buying up many of the stock image providers and raising the price of stock images across the board. It also means that they can apply their "infringement" tactics across a much larger set of images. It sounds like this is the reason *2600* got caught in their net.

Third, it would be technologically feasible for Getty to provide an infringement search on their website that webmasters and graphic artists could use to ensure they didn't run afoul of Getty. Obviously this wouldn't be as profitable for Getty. They actually stand to profit more from these demand letters and it stands to reason Getty intends to freeze out the competition (as clients of competing stock image providers will fear being targeted by Getty).

If *2600* has the appropriate legal counsel (or can recruit an organization like the EFF), I would favor a suit against Getty. A class action suit would be ideal as it would (hopefully) put an end to this snowball that is growing into an avalanche against small businesses and individuals. Otherwise, I sympathize. For what it's worth, the amount demanded of *2600* is much less than they were asking from my client.

Matthew

We are down for the challenge and we know many others are too. We are well aware of how most cases aren't as comical as ours and that many have had livelihoods and businesses adversely affected or even destroyed by these types of actions. In the end, the creative process is crippled out of fear and an overabundance of caution. Incidentally - and we know it's awfully confusing - but it seems that Trunk Archives and Getty Images aren't technically related, other than the fact that they both use something known as PicScout which we believe is owned by Getty Images and also the fact that they share the same address. *(We have lots more of this on page 34.)*

Dear *2600:*

According to the DMCA rules, a claim shown to be false shall be penalized.

If you send a DMCA takedown notice that is both false and meant in bad faith (such as to harass, or doesn't state a real claim), you have committed perjury. Though unlikely, if the party you sent the takedown notice to decided to pursue this in court, you could face all of the consequences that your state imposes on people who lie in court.

Pitiful. Pathetic. Trolls.

Respect the process. Vote.

Bill

This is probably why they don't actually use a DMCA takedown letter, but instead simply send an invoice. A team of lawyers with principles and some free time could help turn these practices into history.

Dear *2600:*

I came across your brief and ridiculous confrontation with your image troll on a TechDirt thread. I read the quote, "Art has always been derivative and transformative." I have been working on grants, applying, etc., for the last nine months, and love this definition. I would like to use it. May I?

Monday

We're sorry but our quotes are ours alone and may not be requoted. In fact, your letter makes unauthorized use of the quote and an invoice has already been sent. (We half assume your question was as sarcastic as our answer.)

Random Thoughts

Dear *2600:*

I've got a great story that I've been working on. It would be a great perspective piece. Hacker meets Hackee. Let me know if your interested.

Sent from my iPhone

Tommy

This all seemed to start off normally enough.

Dear *2600:*

I don't need to remain anonymous. My family and I have been humiliated, degraded, and tortured for months. I already know that you are aware of who I am, where I live, and what has happened. I have been relentlessly studying your phishing tactics, codes, follow patterns locations, addresses, third party loopholes, etc. for months over end. I even phish myself to better understand the tactics. There is no other story that falls in line with the mounds of evidence that I have been collecting over this past "Winter." I have written several statements that already support what is shown in this magazine.

I lost my job at a the telecommunications company that you hacked, over a game and entertainment. I have a few people that will be very interested in this magazine seeing as it's an exact timeline of events which I have already told to FBI, Charter Cyber Security, and local sheriffs. Or we can make a deal for this sick form of entertainment and part ways forever.

Sent from my... You already know phone.

Tommy

OK... this letter is in first place for the Incomprehensible Award of this issue. We've never been accused of having phishing tactics before, so this is definitely new territory for us.

Dear *2600:*

Uhhm maybe I should have read this all the way through before replying. This appears to be phished to me indirectly, correct? Guy in the orange shirt dropped it off and knew I would find it is what I'm guessing.

Sent from my iPhone

Tommy

He clearly has a real fascination with phishing. And he's certainly not the first to believe that an entire issue was written with him specifically in mind. But nobody around here wears orange. So something clearly doesn't add up.

Dear *2600:*

Holy Shit! You guys are fucking good!! I want in.

Sent from my iPhone

Tommy

What he didn't realize at this point was that he was already in and that what he really wanted was to get out.

Dear *2600:*

I apologize for my threats. It can be squashed now.

Sent from my iPhone

Tommy

This came as a relief to all of us.

Dear *2600:*

You ought to know my persona by now. I would never harm anyone nor want to. I need your help to become a better person and like always I skip through shit and don't read thoroughly. This I will read several times thoroughly.

Sent from my iPhone

Tommy

We've often been told that reading our magazine several times has a soothing effect. Reading it only once can have precisely the opposite effect.

Dear *2600:*

I jumped to conclusions before giving *2600* the respect it deserved and at the least apologize for my brashness, regardless of what you do.

Sent from my iPhone

Respectfully,
Tommy

All's well that ends well.

Dear *2600:*

PFACNHK BASEHIT NASDAQ AKA HUMPY DUMPTY

Edward T

Dear *2600:*

Set C_N_R_M_F on your Radar right now! Its Poised to take off! Anticipating great reports!

[phone number deleted]

It's these coded messages that really help get us through the day.

Experiences

Dear *2600:*

Have you guys run across the Google "foo. bar" code challenges?

I was working on a bit of Python code for a *2600* article and did a Google search on some Python arcana. I got my search results, but then my Firefox window sort of split and rotated down to reveal a page "behind" the page.

This page simply said, "You speak our language, would you like to take a challenge?" There were three boxes, "Yes", "Maybe Later", and "Don't show me this again." I clicked "Yes" and was taken to a web-based command line interpreter that controls a programming challenge system.

I completed two code challenges and found them entertaining. No doubt they get much harder as you progress, but I wanted to get back to my work. I've got no idea if the challenge will appear again and have deliberately *not* googled it this evening to see if others are talking about it.

What does it lead to? If I finish all the "Level 5" challenges, will Google offer me a job?

The initial problems seemed harmless enough, but I bet they get a lot harder. Do they eventually become commercially useful? Or close enough that Google engineers might crib my code without telling me?

Has Google really opened a Python and Java sandbox for random folks to run arbitrary code on their servers?

Anyway, it was a very interesting experience and I wondered if others in the community had come across it.

Mike

This is indeed a real thing and we've heard a number of similar reports. The google.com/foobar page is the starting point, but you won't

get anywhere if you haven't been invited and particular Python code is what seems to trigger things. It's really clever and interesting, but it also serves as a reminder that what you search for can trigger something somewhere to launch into action. For now that's a positive thing.

Dear *2600:*

I was listening to an aired *Off The Hook* from either late September or October, and you were discussing whether people in their 30s had used rotary phones. I think the topic was regarding how a few kids were given rotary phones and some didn't know how to use them.

I wanted to mention that we only had rotary phones in our household until the mid 90s, and I'm 34. What I remember most about the phones was how frustrating it was if you misdialed your number and had to start over! Do that a few times and your finger would fall off.

Also, you were talking about the touch tone charge - up here in Toronto, from what I recall, we still had that charge up into the late 90s. A friend of mine still had a pulse dial until Bell Canada finally forced users over to touch tone; his father refused to pay Bell the extra charge for touch tone dialing. Every time I dialed home from his place, I would have the number dialed but would then have to anxiously wait for the pulse tone to catch up. And yet, I miss those days.

By the way, my parents have one of those huge wooden crank phones, intact with the guts. I'll have to get some more back story, but it was handed down from my grandfather who was an electrician and grabbed it from a restaurant that was closing.

David

That crank phone is a great find - never let it go. Your friend's dad was quite wise to not yield to paying the phone company's fee for nothing. It's amazing how long they got away with that little scheme. To clarify one point, phone companies didn't cut off pulse service to customers - in fact, they should still work today on all POTS lines. What you described was their tactic of forcing customers to use touch tones and pay an additional fee by upgrading equipment in the central office so touch tones could be detected and then ignored if the fee wasn't paid. (Some touch tone phones had a switch that allowed the buttons to be used in pulse mode, which is what you describe above.) Older phone switches simply accepted touch tones by default because they were considered standard equipment. Only the newer technology had the ability to differentiate and thus take advantage of the consumer. They could just have easily have charged extra if you hit the star key on your phone. This little history lesson teaches us something about the motivation of phone companies everywhere.

Suggestions

Dear *2600:*

The new store looks good and ordering went smoothly. If it is possible and cost effective, please think about adding vinyl stickers to the items you carry. I would definitely deface/improve various things I own with stickers of your logo and other designs available on the clothing.

Emilie

We will consider this. We're also open to design ideas.

Dear *2600:*

I plan on buying the "blue box" t-shirt, but wish you would have made the text *blue,* not white.

Toby

If enough people want that, we'll consider it for our next run.

Observations

Dear *2600:*

I work for a company that deals with merchant branded reward cards and, upon scanning a card with a strip, there is a number on one of the many lines that comes back. The first digit in this string of numbers tells the little box in most retail stores if the card has an EMV chip or not. If your card has this EMV technology, then the number is a 2 - at least that's what we have seen. If it's a plain old strip card, then the number is a 0. Here is the fun part. If you have a reader/writer and clone a credit card with a strip/chip combo and simply change the number from a 2 to a 0, the credit card goes right through as normal without requiring the chip reader. This could be used as interesting malware to circumvent the requirement for the chip reader to an unknowing consumer.

Code Jester

This is fascinating as it defeats one of the major purposes of switching people over to the chip cards, which was to cut down on the epidemic of duplicated cards. It's much harder to duplicate a chip card than it is a mag strip card. But if it's possible to tell the machine to simply ignore the chip using the method above, we suspect this will become a huge issue in very short order.

Dear *2600:*

Wanted to let you know that 2600.wrepp.com *is* an author index and is up to date as of Oc-

tober 2015 with no plans to stop (have a lifetime subscription). I would suggest, though a work in progress, it's a bit more than an author index - it includes info on every article published including links (most with local wget copies), addendum (i.e., notes issue/page of author letters published concerning their article), is searchable, data may be downloaded, and has details on *The Best of 2600* book. Also, nychacker did email me and I honored his request in the July 2015 update. Author feedback is always welcome.

William

This is a great service for our readers and we all thank you for dedicating the time to it.

Dear *2600:*

As a watcher and reader of Internet news and entertainment, my hackles always rise when I see any reference to hacking. I recently saw a story about JPMorgan and many other banks being "hacked." I am referring to an article in the *Hacker News*. "The three men... were charged with 23 counts including hacking, identity theft, securities fraud, and money laundering, among others."

The accused are charged with as many as 23 crimes and the first listed by the magazine is "hacking." Is it really a crime? I thought a crime was a crime and hacking was an activity or hobby.

I will continue to educate myself and make as many people as will listen aware. I live in a rural area of Oklahoma. I talk every day about how dangerous it is to leave your info on a company or bank website. The company I work for insists on direct deposit, so the people here trying for "off the grid" living are being forced into exposure.

The story referenced above is listed as the largest information theft in history. An estimated 100 million plus persons' information was stolen. Aren't the banks partly at fault? Where is their security?

metal_cutter

These are all good questions. But to address the first point, hacking itself is considered by many to be a crime, even though by most actual definitions it isn't. It may seem trivial but it really isn't, as people accused of hacking are often being accused of merely experimenting or asking too many questions. If we tie those healthy things to crime, we're only helping to perpetuate myths and build a very unhealthy society.

Dear *2600:*

I have enjoyed *2600* for years. Until recently, I exclusively read the Kindle Edition since your magazine is difficult to get at bookstores and reading e-books is more convenient. However, I have been feeling guilty because of Amazon's labor practices. I also have privacy concerns. Amazon knows as least how much of any book/magazine/newspaper you read and probably which articles as well. Since you are now offering digests in epub format, I have canceled my *2600* subscription with Amazon and will buy the 2016 *Hacker Digest* in epub format when it becomes available (I already have all the 2015 issues). It's a shame that you don't offer magazine subscriptions in epub format. I would prefer to give all of my subscription money to you instead of partly to some middle man. If small science fiction and fantasy magazines such as *Lightspeed Magazine* can offer epub formats from their Wordpress website, I'm surprised that *2600*, whom I assume has greater technical prowess is unable to do so. Otherwise, keep up the good work!

Vernon

Right now, epub is the least popular of all of the formats we offer for our digests. This surprises us since so many people were clamoring for it. We have so much digitizing to do and so many formats to support, so right now we're trying to do what makes the most people happy. We undoubtedly will be expanding even more soon.

Questions

Dear *2600:*

I was wondering to what email address do I send images with "2600" in them? I searched the site and, sadly, I don't have a copy of the mag in front of me or I wouldn't have to bother you. Thanks for any help.

Arthur

It's perfectly OK to bother us, though if you still don't have a copy in front of you, this may be difficult to convey. The address is the same as when you're sending in an article, which is articles@2600.com.

Dear *2600:*

Are you only accepting articles and submissions, or do you accept fiction, too? If you do, what email address would I send my story to? Thanks much.

Robin

Yes, not only do we accept fiction, but we've printed a good amount of it. We even have a popular fiction series we've been running, the latest chapter of which appears in the back of this issue. The address is, again, the same as for articles, which is articles@2600.com.

Dear *2600:*

Hey, this is my second time emailing you - I haven't gotten a response from you. I am running out of time. Can you please respond and tell me to F off, or that you can help me or anything. Just please tell me something. This is my life and I don't know who to turn to for help. I went to the last *2600* meeting and met a guy that was going to help me, but my sister got into a car wreck and I had to leave abruptly and forgot to exchange info. I can't wait another month to link up with him again. So please at least talk with me.

s

We don't want to appear callous, but this is not our purpose. We hope your sister's OK and that you solve whatever unspecified problem you were working on. We publish a magazine. We're not detectives, counselors, or problem solvers. You can probably find all three and more at our meetings, something you seem to already know. Good luck.

Dear *2600:*

I remember an article from one of the 1990s or early 2000s issues of *2600* that did an excellent job explaining how a quantum computer could find the factors of the product of two large prime numbers. I don't remember anything more than that. Could someone please look that up in the archives and tell me which year/month edition it was in?

Owen

Going to our store and typing in the word "quantum" will yield the names of all issues that had such an article. The same trick also works for other words.

Dear *2600:*

Hello, I am part of a small group of Canadians who have discovered the art of the mail system. Would you be able to help me locate some literature or articles pertaining to this? Thank you much.

Mike M

Another thing we're not is a library. We've printed articles on postal hacking, though none on the Canadian postal system that we know of. We would certainly like to and it sounds like you may one day be in the position to write a piece on this and help satisfy the curiosity of many others.

Dear *2600:*

Hello friends, I cannot access the link for the *Off The Hook* DVDs on your store. Is there any other link to use?

Lucio

We no longer offer this in DVD format, which is why the link no longer works. We're considering a thumb drive version for people who don't want to spend a lot of time downloading all of the shows that are on our website.

Dear *2600:*

Is the paper edition of the Autumn 2015 edition available online?

jeffrey

No, but the digital version is. We haven't yet achieved the level of magic required to put actual paper online yet.

Dear *2600:*

I really need some help with finding the right crowd, and I believe you can point me in the right direction.

For a school in Holland, I need to contact some people to help me hack the old beamers/projectors. The school has received new touchscreen monitors for use in the classroom. The old projectors that were used for this are now obsolete. We would *love* to use them for projecting interactive games on the floor and walls of the school - simple things like Pong or Pacman, or simple racetracks, things of that sort....

Please, please, pretty please with cherry on top - can you help me find some people who can help me with ideas, software, and/or experience in this? Thanks for even considering to try and help us.

Rob

This shouldn't be too difficult with a little experimentation. We suggest reaching out at a local meeting or hackerspace and finding people that might have a little knowledge in this field who would be willing to do some experimenting. Failing that, looking up your specific model online along with a wish list of what you want to accomplish may prove useful. The important thing is to get a number of people together who see this as a worthwhile challenge. That is a powerful force to have on your side and it usually results in something positive.

HOPE Tickets

Dear *2600:*

Wow! That sold out pretty quickly. I refreshed the page, added two tickets, hit checkout, and I got a cart with the message that all the tickets were sold out. I clicked on continue and my cart was empty. Can't believe that really happened in less than three seconds into 11:11. Hope there are more tickets for sale soon.

Good luck with The Eleventh HOPE!

Jalil

Thanks for the support and we're sorry you didn't get tickets in the first batch (released on 11/11 at 11:11). By the time you're reading this, we will have had one more semi-discounted offering and the normally priced tickets will be on sale hopefully for a while to come.

Dear *2600:*

This was upsetting. I was online at 11:11 and kept adding tickets to the cart for ten minutes straight to only see them automatically being removed.

If there are still tickets available at $100, I would like to purchase two.

Vladimir

We only offered 100 tickets at that low price. All kinds of weird things can happen when that many people are trying to do the same thing at the same time. It's nothing personal.

Dear *2600:*

I was on this from just before 11:11 until 11:25 or so. I tried to order immediately once the button stopped being grayed out, but although it let me add a ticket to my shopping cart, when I went to check out I was told the item had sold out and my cart had been emptied. When I went back to the order page, the item still showed as being available. I tried several times and got the same results. Did the tickets really sell out in a few seconds, or was this a glitch with the store?

Dan

Probably a little of both. We were afraid we'd break the whole thing.

Dear *2600:*

I bought tickets to HOPE in 2014 and flew to New York City for it, but then couldn't even get inside because it was too packed. Why would I ever buy tickets again?

S.

While we had a lot of crowded talks, there was never an instance where the entire conference was too packed for people to go inside. There are always going to be rooms where the laws of physics and public safety make it impossible for everyone to be able to get in. In those cases, we provide as many overflow areas as we can. But a good rule of thumb is to never plan your entire trip around a couple of talks. There is so much else going on throughout the conference that it's almost a challenge not to find something interesting to take part in.

Dear *2600:*

Hi. Myself and a few others tried repeatedly to purchase tickets and they kept being removed from our cart at checkout, even when the site still showed inventory.

Lauren

Most likely tickets were selling faster than the software could update the inventory. The only chance you'd have at that stage would be if a sale were canceled.

Dear *2600:*

Not sure what happened, but I was diligently reloading the page waiting for the ticket sale, was able to add several tickets to my cart, but was unable to check out. I was going through a loop for about four or five minutes. It appeared as if I had three tickets reserved, I was able to get them into my cart repeatedly, but would then error out. I would go back to the page and get a message stating that there were two or three tickets left. Multiple browsers were confirming that tickets were available and allowing me to add them to the cart.

I would really appreciate if the order I can demonstrate here and which is corroborated by the server logs would be honored. I attempted several browsers: Firefox, Chrome, and Edge, running from Windows 10.

Robert

We don't doubt your account. But the same thing happened to scores of other people. Merely adding tickets to the cart is only the first step. You don't actually have the tickets until the sale is approved. It was likely more luck than skill that determined who got through, just as with any event involving a massive amount of people. If there was any skill used, we'd sure like to know what it was as nobody here succeeded in getting through either. And we knew the instant the button was pushed, so we had a big advantage.

Dear *2600:*

Let me first say I *love* the hacker quarterly. I wish it was released monthly - the articles are great and I always learn something, if not a plethora of new things!

Anyway, on to the meat and potatoes of this letter: I am interested in attending The Eleventh HOPE conference. I have heard nothing but great things about previous events. My coworkers went to HOPE X and still talk about how awesome it was. I checked the site a few weeks ago and could not find anything about the next one. I just checked back and it looks like pre sale tix are already gone!

I really don't want to miss out on The Eleventh HOPE, so can you please, please tell me when the next ticket sale will be and how they are purchased?

You guys are amazing! Thanks for your time!

Melissa

Thanks for all of the praise - it helps to fuel us. We took the liberty of adding you to the HOPE announcement mailing list so you get notified whenever a new ticket sale comes along. Good luck!

Hackerspaces: A Definition

by RAMGarden

Have you heard of a hackerspace? Chances are that if you are reading this, you most definitely have. But for those who want to know more, I'll give you a definition from my first-person experience as a member of one.

I have been reading *2600 Magazine* since 2001 when I first found one at my local bookstore. I always read them cover to cover and wanted to try some of the projects that people wrote about, like the credit card mag stripe reader made from an old tape deck play head. But I lacked the tools, parts, and working space to do this. Fast forward to the year 2011 - around October - and we'll pick up the rest of the story.

I finally decided to go see what one of these *2600* meetings was all about - fully expecting a bunch of people looking over each other's shoulders on various laptops doing some extreme programming and getting some serious hacker "work" done. Much to my surprise, it was mainly just a bunch of regular people with engineering-type jobs with some average Joes mixed in and just a few laptops out on the table. Instead of talking exclusively about *2600 Magazine* or the articles in the latest edition (this sometimes comes up for the really good/interesting ones), they were talking about completely random topics amongst each other like you would anywhere else people would gather. *It was awesome.*

After spending some time there, I heard a few of them ask each other if they were going to the "shop" after the meeting. "The Shop?" I asked. "Yeah. It's a cool little hackerspace we have right down the road here where we build and make things, work on personal projects, and write code. Among other things." I turned my head like a dog does when they hear a high pitched sound. "Hackerspace?" Seeing the confusion on my face, the stranger I had just met only an hour ago replied with possibly the best response ever, "It's a bit like the Matrix. You can't really explain what a hackerspace is. You have to be *shown*. Follow us."

We got in our cars and I followed them downtown to an old metal building with a garage door on one side and small windows and a metal door on the front. I watched one of them take a USB stick out of his pocket and stick it in a small metal box mounted to the right of the front door. It instantly emitted a *beep* and I heard the door's dead bolt unlock with a small mechanical sound. I would later learn that it was a servo motor part of the standard keypad dead bolt they hacked to use USB keys for access control. I walked in to see an office area up front with two couches, two tables, and assorted office chairs in various states of disrepair, complete with rips and stains. I was then shown the kitchen which had a small fridge with freezer, a stove that was wired in very recently with the wires proudly displayed out in the open, and a microwave that looked like it had seen its fair share of Chinese takeout and pizza rolls. Then they opened up an inner door at the end of a short hallway and I think I made a pretty embarrassing sound or squeak when I saw what was on the other side because I heard of few of them snicker.

I saw a huge, 4000 square foot open area with an upstairs loft full of so many tools and open space with various parts and pieces of projects in progress strewn and stored about. Someone had pulled their car in the garage door to do a quick oil change. There was an old pinball machine that looked like it was rescued from a dumpster - from the 1950s! Under the loft was an entire area devoted to woodworking tools like a chop saw, drill press, scroll saw, table sander, and power planer. Another area under the loft was devoted to metal working, full of welding equipment and safety gear. The rest of that area had various hand tools, screwdrivers of all sizes and kinds, a collection of nuts, bolts, and screws loosely organized on the shelves, and several safety goggles and gloves in one corner.

In a side room, there was an entire electronics bench with what seemed like hundreds of small, clear, organizing drawers

full of all of the various components you'd need for building your own circuit board from scratch or just fixing something like a DVD player instead of throwing the whole thing away. Also found in this room was one of the first 3D printers made for companies like NASA. It was a BPM personal modeler they had rescued from someone who had a few in a barn. They never got it to completely work, but the extruder would move around inside it like it was trying to print the thing it showed on the program written for DOS displayed on the built-in slide out drawer. They had replaced the old CRT monitor with a flat LCD screen and exchanged the floppy drive with a USB port. Turning around, they showed me the 40 watt CO2 laser cutter from Full Spectrum. They had various things cut from sheets of 1/8 inch thick clear acrylic to paint and put on the sides of PC towers and custom enclosure boxes for their homemade circuit boards.

Going upstairs to the loft area, they showed me dozens of shelves full of hackable parts, like a Rubbermaid tub full of dead Roomba robots, a stack of old laptops, old flatbed scanners, and a banker's box full of different sized wall wart power supplies. "Take anything you want!" they said. "What do you mean?" I asked. "Take anything from these shelves, take it apart, and make something useful or just really neat." I immediately thought of some uses for those Roomba bots as a platform for a telepresence robot I could use to visit home when I travel for work. That night, I asked everything I could about the place.

"This is a hackerspace - or makerspace. Some people don't like to use the word hacker because of all the negative thoughts that have become associated with it. We're not some secret place where nefarious computer geeks sit around computers and write malicious code to try to break into bank accounts and such. We're just a bunch of people who like to tinker and make things or take things apart and make them work in ways they probably weren't intended for, but are better or more useful in some way. Some people have even built prototypes for products and started their own business from here! Everyone has different skills and most are willing to help other people with their project if they get stuck on a part that isn't their specialty. If someone doesn't know how to program, but their project needs a bit of code to run, one of our programmer members can either teach them how to do it or help them do it. It's the community plus the tools that makes this place great!"

Hearing that, I immediately asked how to become a member.

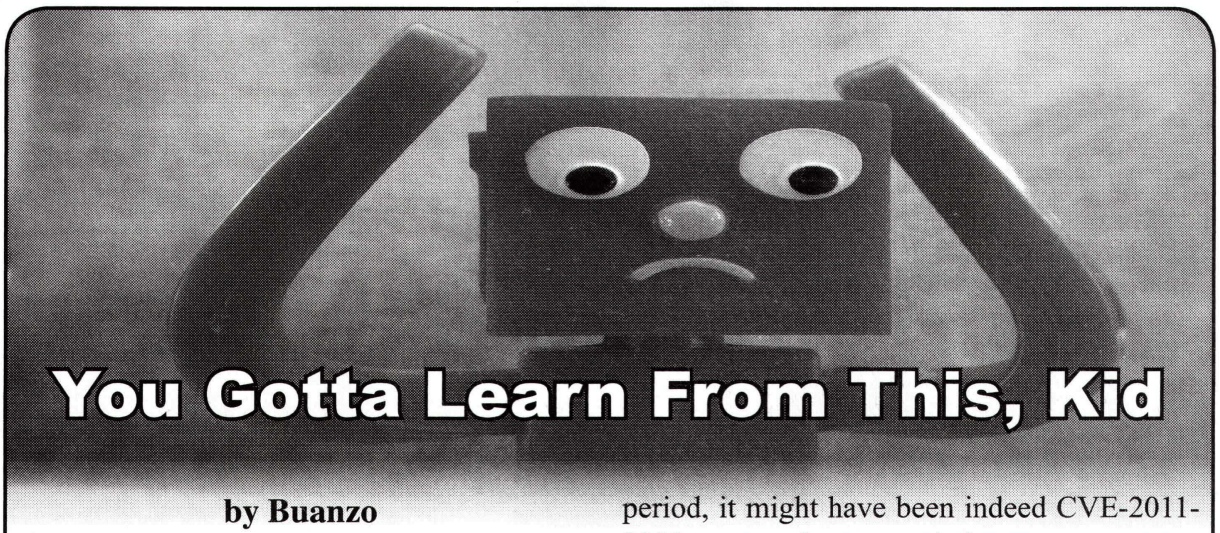
I learned that this wasn't the only place like it and there were several all over the U.S. and across the globe. I filled out the paperwork and, after visiting several times over the next few weeks, I became a member paying my monthly donation with 24/7 access. I brought in my own USB stick and they added its hardware ID to their white list so now I could open the door. I worked on various projects over a few years and helped others with the programming parts of theirs. I even helped beta test a "learn to solder" class where you build a six-sided die simulator circuit that one of the members there invented. He called it the ST:EAK or Soldering Trainer: Entropy Approximation Kit.

Scroll ahead to 2013 and I moved away to Florida for the better weather. Before moving, I made a deal that wherever I moved had to have a nearby hackerspace! Now, I am helping run the local space as the secretary (hackerspaces usually have a President, Vice President, Secretary, Treasurer, etc.) to help get new members signed up and put into our database and Google Group.

Recently, I helped build the RFID card-based door lock that was put together from a Raspberry Pi computer and LCD screen from `adafruit.com`. I look forward to writing up more articles about the various projects I work on in the future.

If you haven't joined your local hackerspace, I strongly urge you to find it on `hackerspaces.org`. If there isn't one near you, then start a `meetup.com` for one and see if you can get ten or so people together to rent a small space and gather some tools. Then pass the hat around for donations to buy a 3D printer and other large tools. There are lots of hackerspaces out there to ask for help if you want to start your own. I also recommend that everyone wanting to start a space join the discuss mailing list at `hackerspaces.org`.

Happy Hacking, Making, and Learning.



You Gotta Learn From This, Kid

by Buanzo

Around early 2008, I was coding a website using PHP. When debugging, I came across a username and password I was using for basic HTTP auth, in the PHP_AUTH_USER and PHP_AUTH_PW variables. That is *not* the strange part, of course.

But hear this: the username and password, those were for a totally different website. Different domain altogether. I was developing on somesite.com, but those credentials were for totallysomethingelse.net.

What was going on there? Why was my username and password being sent to another website? And how long has this been happening?

I immediately added a couple of lines of code to index.php to email me the contents of those variables anytime they were set for any request. And forgot about it, until one day I received an email... that included someone else's username and password. And that was not the only time it happened. I got 17 emails in between 2008 and 2012, averaging 3.5 emails per year. Then it stopped.

Of course, as I recognized the username and password I mentioned at the beginning, I knew it was my Nagios credentials! And I *was* using a proxy to access Nagios, and I might have used that proxy to access the other website I was developing.

I tried searching for credentials leakage vulnerabilities in Firefox, and I found `https://bugzilla.mozilla.org/show_bug.cgi?id=664983`, but no non-proxied, basic http auth cross-domain leakage.

But, as I got a quite small amount of usernames and passwords in a four-year period, it might have been indeed CVE-2011-2990, or an unknown variation.

I got some interesting usernames, and some pretty cool passwords, too. But I never saved REFERER headers, nor User-Agent strings. I did not want to know what those usernames and passwords applied to. But now, I remembered all about it. How long was this vulnerability out there? Did I find it before it was even publicly reported (if it is indeed CVE-2011-2990)? I'll probably never know.

It is now 2016, and I see no harm in publishing the list of usernames and passwords I got (although I will mask some characters using #s, just to be on the safe side of it).

So, may this story serve as a cautionary tale, kids: if you come across something odd, *do your frickin homework!*

Cheers!

```
USERNAME / PASSWORD
--------------------
haz##shoep / ENG...zo##_1988
timo### / ba##lon4
na##us78 / 230##309
yudi###idis08032 / 75*11*21Sa##uy
amd###operations / Ense###e99a
- / -
###00655 / Kal###0_13
fakeuser / fakepass
avazqu###.ext / avazqu###2010
##user / ##heslo
j###.contreras@ad###rus.com / jua#
➥##05
m###er-1376996b11 / #bd8e5ef439501
➥9834ceb694c367803#
###oleta.ruse@ro.###.com / beja##e
➥113
##Y@.@os~vq2+(+2os~vq@-1(\'@lvo)&1
➥,.1./.1+@./* / I8BBAC5F097B342BD
➥DDAF644C6D0A1F##
searc##lox / Axd##bqYepMum4jt
```

The Limits of Open Source Hardware

by Monican

If you're a hacker who has ever thought about a new way to integrate computers into the world around us, you've probably heard of the Raspberry Pi, the Arduino, or the BeagleBone. These three extremely popular open source hardware development boards are being used in nearly every project you read about on engineering blogs. The fanciful names of these three boards hide the power that they offer the user by making it extremely easy to integrate hardware - and thus the physical world - into software projects by wiring up sensors to collect data, or to embed a powerful and energy efficient computer into an engineering or art project.

However, despite the term "open source" being used to describe these devices, that phrase has a much different meaning than when we apply it to open source software projects like the Linux OS or coding projects you'll find on codesharing sites like GitHub. What do I mean? Well, the best example is to look at the Raspberry Pi's restricted schematics. Although the Raspberry Pi is widely used and supported in the open source community, the developers have chosen to restrict the release of the schematic files.[1] They claim this is to prevent copycatting and sub-par rip-off boards that cause headaches for the official devs (i.e., if someone has a problem with their poorly made Chinese knockoff and complains to the actual Raspberry Pi devs, this wastes their time). This is understandable, especially since the Arduino team has suffered from knockoffs,[2] but at the same time it limits repairability and opportunities for learning.

One major difference between open source software and hardware is that with software, you can literally examine your entire code stack if you're using an open source OS, and running only open source software in your development environment. With hardware this isn't the case, because the intellectual property of the chip manufacturers is a closely guarded secret. For example, Atmel makes the microcontrollers around which the Arduino system is built, but they are not an open source company. Neither is Texas Instruments (who make the AM335x CPU for the BeagleBone) or Broadcom (who makes the CPU for the Raspberry Pi). So if you want to get down to the bare metal and help develop the most fundamental parts of these systems, you are restricted by nondisclosure agreements, which in some cases are unavailable for hobbyists.

What does this mean for the open source projects being developed with these boards? Well, if you can't see inside the hardware, you can't check it for glitches and you can't rule out intentional backdoors or forgotten debug modes that might compromise the security of your project. Whoever controls the actual silicon can pull the rug out from under any software running on it, no matter how secure. Just look at the discussion around Intel's random number generator after the NSA revelations came out: the developers of the FreeBSD operating system decided they couldn't trust the opaque hardware inside Intel's CPUs, so they have to program as if the hardware they're running on is hostile.[3] You can check the chip's "silicon errata" datasheet to see what bugs people have found and reported, but those are usually updated very infrequently. For example, the silicon errata for the BeagleBone Black's CPU was last updated in November of 2013.[4]

Take heart, though - there are some hackers out there who are pushing the limits to see just how open they can make their hardware. Legendary hacker Bunnie Huang has created the Novena Laptop which - although it has specs that aren't that great and costs a lot when considering pure performance - is so open that he even x-rayed the hardware to demonstrate that there isn't anything hidden inside the metal housing.[5] His team had to make some tough tradeoffs with performance because they're only using hardware for which they have all the information you could possibly need to call it truly open. There were very few parts to choose from that fit this bill and it shows the paucity of options since everything else is restricted by IP at the level of the hardware manufacturers.

With all of this said, the benefits of these systems far outweigh the problems I've outlined above. These devices make embedded electronics accessible to people who aren't engineers, like artists and students, or even scientists and hobbyists who just need to rapidly prototype something. This makes me very optimistic about the future, and I look forward to a day when any laptop or electronic gadget you buy will have a sticker saying "Certified XX% Open Source Hardware."

1. http://www.tuxradar.com/content/%EF%BB%BFinterview-eben-upton
2. http://blog.arduino.cc/2013/07/10/send-in-the-clones
3. http://boingboing.net/2013/12/10/freebsd-wont-use-intel-via.html
4. http://www.ti.com/lit/er/sprz360f/sprz360f.pdf
5. http://www.bunniestudios.com/blog/?p=3991

EFFecting Digital Freedom

Defending Privacy on the Roads by Dave Maass

I like to imagine that if vehicle license plates hadn't been invented, the public would never stand for them if they were introduced today. With technological advances in character recognition, CCTV networks, data analysis, and geomapping, people would understand that a license plate would be little more than a beacon for the surveillance state to track your movements.

This mental exercise does little to help me sleep better at night. My mind runs circles around the problem of automated license plate recognition (ALPR) systems. How do you fight mass surveillance when the government mandates that you wear the thing they're tracking?

ALPRs are networks of cameras that photograph any license plate that appears within view, extract the plate number into a machine-readable format, and combine it with the time, date, and location of the plate capture. The systems can collect data on thousands of vehicles every hour. It is one of the most pervasive mass surveillance technologies in use by local law enforcement agencies around the country. A 2012 Police Executive Research Forum survey found that 71 percent of agencies surveyed used ALPR. These days, I get several email alerts each week from a government procurement website telling me that an agency - often multiple agencies - have bought ALPR systems or renewed their ALPR contracts.

Law enforcement agencies employ ALPR in three distinct ways: stationary, mobile, and through private database access. In the first scenario, police install cameras on streetlights, telephone poles, and other static locations to capture plates as they pass. Police also mount ALPR cameras to patrol vehicles, then drive around areas collecting plates (often of parked cars). These systems often use "hot lists" to ping the police every time a particular vehicle is spotted. Police agencies also subscribe to privately maintained ALPR databases by companies that aggregate data that the companies have either collected themselves or acquired from other agencies.

Proponents will say that ALPR systems are really no different than a police officer back in the day jotting down plates in his notebook. There's nothing private about this information, they say, since the cars are in plain view. Of course, it is very different and the information is very personal: by collecting thousands of plates each day and storing them for significant periods of time, ALPR gives police the ability to discover sensitive information about drivers. In aggregate, these data points can reveal where you work, where you sleep at night, what churches you attend, and what doctors you visit. Ultimately, we're talking about a surveillance system that collects far more information on innocent people than it does genuine suspects.

Police argue that ALPR is an important tool to locate stolen cars or to find kidnapped children, but we've seen these tools proposed for far lesser offenses. The DEA has acknowledged that one of the primary values of ALPR is how it helps them seize currency from drivers. Louisiana police proposed a pilot to install a statewide ALPR system to scan for uninsured drivers. Meanwhile, police in Florida use ALPR data to identify cars driving through neighborhoods known for prostitution. They then send intimidating "Dear John" letters to the registered vehicle owners warning them about sexually transmitted diseases and warning them that they should refrain from driving into that area.

The companies that sell ALPR systems have their own agendas. Vigilant Solutions, for example, gives police free ALPR cameras in exchange for a cut of the proceeds from collecting on unpaid fines. The benefiting agency has to hit a regular quota to keep the devices.

If you worry about your communication being snooped on, you can use encryption. But there's no simple solution for license plates, since many states have laws forbidding anything that would make it difficult for a police officer to read your plate. In California, the law even bans anything that would make it difficult for ALPR.

That leaves few options. You could go the Steve Jobs route and lease a new car every six months. You could use a full-vehicle cover or attach a bumper protector, but that would only protect your privacy when you're parked. In those wee, sleepless hours, I fantasize about a coordinated citizen effort to paste printouts of license plates around a city so that it produces so many false positives that ALPRs become fatally unreliable. Setting aside that it would be a cast-of-thousands production, the other major hurdle with that idea is that without access to the ALPR devices or raw data, we'd have no way to know if it worked. Chances are the manufacturers would quickly adjust their algorithms anyway.

So far, our battle has been over transparency and accountability. EFF is currently suing the Los Angeles Police Department and Los Angeles County Sheriff's Department to get access to a week's worth of ALPR data under the California Public Records Act. The agencies claim that it's all protected from public disclosure because they are investigative records. Who's under investigation? Everyone, they say. That case is now before the California Supreme Court.

We had better luck with the Oakland Police Department, who did provide us with a week's worth of collected plates. EFF Technologist Jeremy Gillula and I analyzed and mapped out the data. Through time lapse, we were able to see how police vehicles mounted with ALPR cameras wound their way through the city, street-by-street, gobbling up plates, like one of those old "Snake" games on a Nokia phone. It also became clear that African American and Hispanic areas of the city were under far more intense vehicular surveillance.

ALPR present another threat to privacy: bad security practices on behalf of the police. Piecing together research from various security researchers working with the Shodan search engine, EFF Technologist Cooper Quintin and I were able to identify dozens of police ALPR cameras that were insecurely connected to the Internet, mostly in Southeastern Louisiana. In some cases, the control panels and live video streams from the cameras were viewable through a browser - no password required. You could also siphon off the live plate data as it was being transmitted to the central servers.

We did score one major legislative victory in California this year: a new bill - S.B. 34 - classifies ALPR data as sensitive information under the state's data breach law. It also requires agencies that use ALPR to take adequate measures to protect ALPR data and to publish privacy and usage policies. Private citizens can now sue if they are harmed by an ALPR data breach.

As for the people who believe that if you've done nothing wrong, you've got nothing to worry about: just ask Denise Green. San Francisco police pulled the innocent driver over, held her at gunpoint, handcuffed her, forced her to her knees, and then searched both her and her vehicle - all because an ALPR camera misread her plate and the officers didn't bother to verify the number. You can also ask 74-year-old Richann Flynn, who *The Sacramento Bee* reported received 55 notices from the Bay Area Toll Authority, accusing her of failing to pay tolls for bridges she hadn't crossed in at least 15 years. Again, she was the innocent victim of a flawed automated system.

ALPR is really only the beginning. We're also beginning to see government agencies adopt crossover technologies, such as Xerox's "Automated Vehicle Occupancy Detection," which is supposed to determine whether there are enough people in a vehicle to justify use of the carpool lane. Already, ALPR companies are devising ways to use facial recognition to conduct similar tracking surveillance.

But you can rest assured that we'll be up all night fighting back.

Dave Maass is an investigative researcher at the Electronic Frontier Foundation, working on its Street Level Surveillance project.

Rewriting History

by Steffen Fritz
sfnfrz2600@gmail.com

0x0 Web Archiving

With the growth of the World Wide Web and its increasing cultural and political influence, the archiving of web published content became an important matter for preserving cultural heritage. Public institutions like the Library of Congress (LoC) in the United States[1] or the Bibliothèque nationale de France (BnF)[2] and non-profit organizations like the Internet Archive (IA)[3] are doing a great job in this. While the LoC or the BnF don't crawl the whole web - they curate, collect, and preserve topic, event, or domain specific - the IA takes them all, automatically. At least they try. Other services like `http://archive.is` or `http://webrecorder.io` allow users to manually mirror web pages and see the results right away.

Whoever is preserving has three possible archiving methods: transactional archiving, database archiving, and remote harvesting. The most common one is the latter and the idea is fairly simple: Copy a website, search the source code for URLs, copy the referenced resources, and repeat recursively until you hit a termination condition, e.g., no new web resource found or when leaving the domain. A program doing this is called a web crawler. Popular tools are Heritrix[4] and HTTrack[5]. HTTrack saves files as a web server delivers them, e.g., image.jpg as image.jpg and index.html as index.html. Heritrix creates web archives according to the WARC file format, which is the de facto standard for web archives.

0x1 WARC Format

The WARC file format defines how to store payload content, control information, and arbitrary metadata as blocks together in one file. Control information like DNS and HTTP requests and responses make the crawl comprehensible. Hash sums, dates, and file sizes describe the digital objects. Each WARC record in a WARC file is initiated by "WARC/1.0" and consists of a record header that describes the type and content of the record. It is followed by the content and two newlines.

You can create a WARC file with wget >= 1.14. Just add the switch "--warc-file=FOO", e.g.,

```
wget --warc-file=2600 http://
➥2600.com
```

wget creates a warc.gz file. Unzip it and open the WARC file with an editor like vim or emacs. The first block in the file describes the WARC file itself. The following blocks are related to network traffic and payload. The fields in the blocks have a simple named fields structure, terminated with CRLF. An important field is "WARC-Target-URI". It is identical to the source URI and therefore it also determines the file name of the payload.

Let's have a look at an example. Some lines are omitted. All blocks are from the same file. We investigate three blocks:

```
<CODE>
    WARC/1.0^M
    WARC-Type: warcinfo^M
    Content-Type: application/
➥warc-fields^M
    WARC-Date: 2015-09-15T13:
➥20:42Z^M
    WARC-Filename: test.warc.gz^M
    WARC-Block-Digest: sha1:XGCP
➥3I5MSJ4DGD7EH5DTLJJXULVOATQK^M
    Content-Length: 224^M
    ^M
    software: Wget/1.16.3 (linux
➥-gnu)^M
    format: WARC File Format
➥ 1.0^M
    ^M
    ^M

    (...)
</CODE>
```

The above is the first block in our WARC file. It is an info block and contains "warc-fields". The content, i.e., the following two lines, has a length of 224 bytes. The second block is a request block in which the network communication for a single request is logged.

```
<CODE>
    WARC/1.0^M
    WARC-Type: request^M
    WARC-Target-URI: http://test
```

```
➥.wtf/^M
    Content-Type: application/
➥http;msgtype=request^M
    WARC-Date: 2015-09-15T
➥13:20:42Z^M

    (...)
</CODE>
```

The third block contains the response from the server. After the WARC fields and the metadata, you can see the html payload.

```
<CODE>
    WARC/1.0^M
    WARC-Type: response^M
    WARC-Target-URI: http://test
➥.wtf/^M
    WARC-Date: 2015-09-15T
➥13:20:42Z^M
    Content-Type: application/
➥http;msgtype=response^M
    Content-Length: 4896^M
    ^M
    HTTP/1.1 200 OK^M
    Server: nginx/1.8.0^M
    Date: Tue, 15 Sep 2015
➥ 13:20:42 GMT^M
    Content-Type: text/html^M

    (...)

    ^M
    <!DOCTYPE html>
    <html dir="ltr" lang="en">
    <head>
      <meta charset="UTF-8" />
      <meta name="viewport"
➥ content="width=device-width,
➥ initial-scale=1">

    (...)
</CODE>
```

For a full description read the specification. The ISO draft is available at the BnF and well readable.[6]

At the time of this writing, the International Internet Preservation Consortium (IIPC) is working on version 1.1 of the specification - pretty transparently on github, by the way.[7]

What to do with the WARC file? Replay it. There are a few tools to render the archived content. One is the (Open) Wayback Machine you may know from the Internet Archive. Another one is Pywb, which I prefer for local testing because it is pretty easy to set up and much lighter.[8][9]

Whatever you use, you set up a data storage for the WARC files and the tool of your choice serves the content, rendered by a browser. Suppose we are using the Wayback Machine on localhost and the above example with WARC-Target-URI http://test.wtf. You would open the URL http://localhost:8080/web/20150915132042/http://test.wtf and you'd see how the website http://test.wtf looked like in September 2015.

Do you see where this is going? Let's assume we could create WARC files with arbitrary content. And let us assume further we could manage to inject that file into a trustful archive and that we could share a link with Alice and Bob: Both might be tricked into believing a website looked like something it never did. Let's call it "post defacing."

0x2 Create a WARC File and Make Bob Trust It

Of course, you could create a WARC file with a text editor. But the creation of hash sums, length of content, etc. might be a little bit annoying. You could also set up an environment to crawl a fake site. I decided to write a Python script to create minimal, valid warc files.[10]

You call the script "python html2warc $URL $SOURCE $TARGET_FILE". $URL is the root value for the WARC-Target-URI field, $SOURCE must be a directory with the desired content, and $TARGET is the name of the WARC file.

A proof of concept WARC file can be downloaded from github.[10]

You can upload that file to `webrecorder.io` and watch the result. Fascinating, isn't it? Well, `webrecorder.io` isn't an archive and the service explicitly states that. But are Alice and Bob aware of that? Checking the trustworthiness of sources isn't a standard procedure in online communication. Sadly.

To upload the file to `archive.org` and trick Bob, things are a little bit more complicated. You can upload a WARC file with an ordinary user account into a collection. But then it is stored as the mediatype "texts" and can only be downloaded again as a WARC file. If you try to change the web memory for a specific site, you have to convince a member of the Archive Team to copy your WARC into their collection and change the media type from "texts" to "web". Obviously, it is possible to steal the archive login from a member and do it yourself. No doubt, some

Mallorys are trying to do this.

Remember: It is not about defacing a web site. It is about changing the political, cultural, and social memory.

0x3 Impact and Responsibility

Putting false documents into trusted archives is not a new threat. In 2005, the British National Archives detected faked documents, claiming that Heinrich Himmler was murdered in custody. And in 1967, Gérard de Sède wrote in his book *Le Trésor Maudit* that a guy named Pierre Plantard is a descendant of Dagobert II and therefore the one and only King of France. De Sède referred to documents found in the National Archives in Paris. Placed there by, you guessed it, Pierre Plantard. You may read on this very interesting case by searching for the "Plantard Dossiers." I am pretty sure that faked documents have rewritten history and they will in the future. Web archives are just another playground. But an important one.

Who's responsible? Surely, archives have to check their objects and they are responsible for the data they provide - be it books, birth certificates, or web archives. But in my humble opinion, users also have to check their sources and should not automatically trust something because of its outer packing. Remember that Trojan horse?

[1] http://www.loc.gov/webarchiving/
[2] http://www.bnf.fr/en/professionals/digital_legal_deposit.html
[3] https://archive.org
[4] https://webarchive.jira.com/wiki/display/Heritrix/Heritrix
[5] https://www.httrack.com/
[6] http://bibnum.bnf.fr/WARC/
[7] http://iipc.github.io/warc-specifications/
[8] https://github.com/iipc/openwayback
[9] https://github.com/ikreymer/pywb
[10] https://github.com/ampoffcom/warc_2600

The Herculean Task of Making a Documentary on the History of Computer Hacking (Part 1)

by Michael Lee Nirenberg
restraining.order.ltd@gmail.com

When my last film, *Back Issues: The Hustler Magazine Story,* was in its final stages, I was itching to make another documentary. My friend and executive producer of *Back Issues,* Nick McKinney, proposed making a film on the history of computer hacking. I thought it had been done already. How could one not exist?

There have been plenty of films about contemporary hacking (*We Are Anonymous, Downloaded,* etc.), but the history of it has remained relatively unknown to the public. Hacking is present in everything we do in this society. It's no secret to the readers of *2600* that hackers have made contributions to tech that are now omnipresent in every aspect of American life.

Later, I learned several "made for TV movies" had been produced, but no serious cinematic documentary had been made on the people and events that have brought us here, particularly over the second half of the last century. Is it possible that every major technological advancement can be traced back to a hacker like Nikola Tesla who had a better idea of how their corner of the world should work?

I'm not a hacker, at least not a hacker in the sense portrayed by the media. I'm not interested in cracking cybersecurity, coding, programming, or repurposing hardware myself. Although that stuff greatly interests me, I suppose I'm a hacker in its original 1960s vernacular. The pioneer hacker Richard Stallman, who has been called "the last of the original hackers," defines a hacker as "someone who enjoys playful cleverness."

Nick McKinney had suggested a book he read called *Masters of Deception*. It is a book about the hacker "gangs" of the 1980s. It introduced a whole cast of colorful characters with names like Phiber Optik, Corrupt, Scorpion, and Acid Phreak. The book's drama unfolds between our hacker protagonists and a befuddled National Security Agency (NSA), the forward thinking formation of the Electronic Frontier Foundation

(EFF), and the unjust prison sentences for these hacker teenagers. Nick was right. It's a damned good story; however, after further research, we learn that these hackers were not "gangs." The word "gangs" makes better ad copy than "groups." They were more like rock bands. Each hacker had his own system he liked to explore. It turns out there was better info out there - the hackers themselves. One thing I noticed by researching and observing hacker groups is that they always seem to develop a strong sense of ethics. That's something interesting that may not necessarily be native to hackers, but I would like to explore that aspect as in my film.

In the 80s, these hackers were just teenagers being teenagers, poking around the vast expanses of the networks of that time. One of the tenets of *Masters of Deception* was "leave everything the way you found it." At the time, hackers thought that this tenet was a preventative measure that would keep the hackers from being considered destructive by the courts, which ultimately helped them stay out of worse trouble in some cases. Of course, we now know boys will be boys (and it almost always *was* boys until recently) and that not all teenagers demonstrate self-control. This was the beginning of black hat (destructive) and white hat (harmless fun) hacking, but these terms were yet to develop.

Over the next few months, my colleagues and I were finalizing mastering for the release of *Back Issues*, and I was absorbing as many books on the topic of hacking as I could. I realized *Masters of Deception* was only the tip of the iceberg. The next book that was a great influence on me was Steven Levy's *Hackers* from 1984. At that time, *Hackers* was the canonical book on a subculture of programmers, entrepreneurs, and visionaries. I believe it was the first of its kind. *Hackers* covers the early days of the PC revolution, "phone phreaks," the early AI experiments at MIT, as well as the first video game systems. I began to draw a generational connection between hackers.

There were waves of hacker groups throughout history, but their timeline was nonlinear and, like most of history, messy. After the computer left the corporate clutches of IBM, it was further developed by the math geniuses of more radical 1960s and 1970s subcultures. I'm interested in the computer as a development for mind expansion. Hacker culture sometimes has a mainstream presence when a negative occurrence happens, but it's mostly the subculture that first attracted me as a filmmaker. It has its own hierarchies and fiefdoms despite its lofty goals for equality through information. I suppose that's human nature.

The history of hacking gets really interesting to me in the 1960s. That's when hippies and students became drawn to phone phreaking. Phone phreaking was the name given to obsessive telecommunication enthusiasts. Many early phone hackers involved were probably into only making free phone calls but, as we know now, a great many were simply drawn to understanding telecommunications in a larger sense. In that period, we meet some of the pioneers of the modem and our contemporary communications systems.

Back then, you could listen to the clicks on a telephone and begin to untangle the routes in which a phone call would travel. Before the Internet, this was the vanguard of telecommunications. This story is forever tied to John "Cap'n Crunch" Draper and his fellow explorers, many of whom were blind. The blind men had a real knack for listening closely to the clicks and telephone switch lines. Draper later went to Apple and developed the AppleCat, which was their modem and one of the earliest from what I understand. He is still around. I interviewed him on Skype recently.

This act of phreaking was radicalized by the Yippie movement's *TAP* newsletter, a tongue in cheek acronym for *Technological American Party* and later changed to *Technological Assistance Program*. (The word "party" could only be registered for political affiliation.) Back then, the telephone system was a monopoly and AT&T was the only game in town. Al Bell (a pseudonymous play on "Ma Bell" - the public's name for the telephone company) was the longtime editor of *TAP* until it was taken over by legendary hacker and all around interesting guy, Cheshire Catalyst. Cheshire turned out to be, in his own words, "not a very good businessman" and *TAP* folded shortly after he took it over. The hole for a hacker/phone phreaking magazine was shortly filled by Emmanuel Goldstein's *2600*.

Phone phreaking peaked in the 1980s when the Internet was just around the corner. Groups of talented hackers would meet on unsanctioned conference calls and computer bulletin boards in order to share their common interests. Being teenagers, they formed cliques and groups based on who could get into what systems. Knowledge is power when you are only 16.

Being a New Yorker made it much easier to start this film for me. It just so happens many of the notable names throughout the story either

live here or pass through frequently. New York also happens to be the home of *2600: The Hacker Quarterly*. Every culture needs its publication to coalesce around. *2600* has been around since 1984 and has been steadily publishing for 31 years.

When one starts a documentary, one has to reach out to people to get started. I reached out to about a dozen or so people to conduct interviews. The first one to get back to me was cyberspace pioneer (and famed Grateful Dead lyricist) John Perry Barlow. John was in town for meetings concerning one of the several projects he could be working on. At the time, all I knew was I wanted to start gathering interview footage for a film on the history of computer hacking, so I had lots of questions. He was a great first interview for several reasons, despite knowing nothing about my previous work.

Along with Mitch Kapor (founder of Lotus 1-2-3), John Perry Barlow had founded the Electronic Frontier Foundation. Mr. Kapor and Mr. Barlow met on The WELL, which was the online community started by *Whole Earth Catalog* founder Stewart Brand. The WELL was an acronym that stood for "Whole Earth 'Lectronic Link," with the intention of starting an online community that would usher in the age of electronic enlightenment - a meeting of minds. Barlow and Kapor both received unsolicited visits from the FBI and both men discussed it on The WELL. After meeting, they decided something had to be done about government harassment in the electronic age. Ostensibly, the FBI was looking for hackers. While both Barlow and Kapor were innocent of any wrongdoing, they became aware that many kids who were poking around the early Internet were becoming the victims of a government hacker witch-hunt, which would ultimately hurt a burgeoning Internet. They formed the EFF to create a bill of rights for the Information Age. We owe them lots. As it turned out, among the many kids who were poking around the Internet and having fun getting into systems were several "elite" groups of teenagers who were the best and brightest of their generation. I've tracked down and spoken to them on camera.

I've had the pleasure of interviewing legends of the early Internet over the course of 2014-2015, some of whom have become friends of mine through the process.

Around the time I was heavily researching the hackers of the 80s and 90s, I was drawn towards cyberculture magazine *Mondo 2000*. It was the tastemaker of hip cyberpunk. Not only did they have contributions from Bruce Sterling and William Gibson, they also had Timothy Leary and countless other counterculture icons contributing to a magazine that defined the times. *Mondo 2000* was published by a lady named Queen Mu, who inherited the startup capital for the enterprise, as I understand. The magazine's creative force was Ken Goffman, known to the world as R.U. Sirius. Unfortunately, *Mondo 2000* was crushed by corporate startup *Wired,* which is another story for another time, but *Mondo 2000* was the real thing. I believe every issue is available on archive.org, where our friend Jason Scott is holding down the entire history of the Internet.

Not being a linear story, the challenge of the documentary director is to make sense of a sprawling history and to make it presentable. Hell, we haven't even mentioned the tale of Kevin Mitnick or Kevin Poulsen yet. As it stands, I don't even know where we will stop. Just recently, the Ashley Madison hack was in the news and tomorrow is uncertain. The landscape changes beneath us every day.

We even started an experiment in covering modern hacking a bit. I was particularly excited to spend some time with Ellen Jorgensen in the homemade biohacking lab she and her colleagues set up in Brooklyn. The frontier will always interest me, as will history. I struggle with the solution. In addition to the 22 hacker/experts I've interviewed, I've had a few icy interactions with some people who would be great here. Sometimes they warm up, sometimes they don't.

Filming could take another year or so. There is a lot of information to process into a cohesive whole. Many of you will be mad at me. I'm going to have to leave out some of your favorite hackers, hacks, and stories. I apologize in advance. The movie business isn't ready for a ten hour documentary. These are the compromises one has to make to get millions to view it. In my last film, I had to edit out all the vaginas from a documentary on *Hustler Magazine*. Why did I agree with the studio in the end? I wanted a wide audience and that's what I got. Am I a sellout? Yeah, I probably am. Preaching to the converted doesn't interest me. It doesn't educate and widen our culture. I welcome your disagreements, but ultimately don't care. That's the Faustian bargain the documentary director has to make with himself and the audience.

It's cool. History will still be told regardless.

Feel free to contact me to discuss further. My team will pass along all of the positive and constructive ideas as well as block/delete any trollish negativity.

Dev Manny, Information Technology Private Investigator "Hacking the Naked Princess"

by Andy Kaiser

Chapter 0xF

How do you kill a program? You can try going with the classics, like CTRL-ALT-DEL, Task Managers, and - when all else fails - you go nuclear by launching that admin kill command to a PID.

In this case, I needed more, because I wasn't just dealing with stopping a program, but its output. I had to undo the damage. Everyone who had seen the Naked Princess picture had been freaked and terrified, and digital data being what it was, I was sure there were plenty of copies spawning via networks and clouds and SANs.

Well, to be honest, the Naked Princess freaked and terrified everyone but me. While the picture was disturbing, yeah, I'd seen worse. I wasn't some hardened, jaded, emotionally dead Information Technology Private Investigator... well, maybe I was, but still, I knew I was missing something. Those I'd interviewed about it had seemed emotionally ripped, as if a cold hand had reached inside their soul and yanked on something important. I was missing something, and it was at a personal, private, emotional level.

One thing I *did* know was the 384-digit encryption key I'd recently sent to P@nic. I used it now on the file she'd sent me: `Decryption Achievement Get.`

I was looking at the source code to the true "Naked Princess."

My life was, of course, filled with intrigue and excitement. I generally stayed away from things that were not. Application development was a not-so-random and timely example. I hated coding and programming, and therefore my coding and programming skills were sub-zero. If I was really being honest, I just didn't have the brain for it. But I preferred to lie to myself and just say "Application design? Coding? Creating something from nothing that will exist eternally, like a nerdy god with a surplus of logic, creativity and power? Meh. That sounds totally boring." Then I'd at least have an excuse for my failure.

My challenge was clear. I had to break the program. I had to take the thing that came from nothing, that should now exist eternally, and I had to figure out how to delete it from existence. Easy peasy.

P@nic had created the Naked Princess app. Unlike me, she *did* have the brain for it. As I tried to review her code, I saw that she emulated the spirit of genius programmers everywhere: She was horrible at documentation. Arcane and inexplicable pieces of abbreviations and mental shorthand were dusted over the code. These supposed comments were there to better explain how the program actually worked, but to interpret them I'd need help from someone way smarter than me, like from the love child of Elon Musk and Stephen Hawking. And from what I could tell, there wasn't one.

After trying too long to interpret the code on my own, I was getting queasy. Not because of the code itself, but what my inability would lead to.

If I couldn't read and interpret the program, I'd have to run it.

I certainly couldn't send it to anyone else for assistance. If this really was the Naked Princess app, I couldn't risk spreading it, not without knowing how it worked or spread or generated its disgusting content. I was stuck investigating on my own.

While I was just a blushing virginal programming newb, I was at least able to recognize the code's language and compiler. A quick 657,175 milliseconds later, I had the executable.

I ran it.

I was met with an empty black screen. After a few seconds of my CPU spiking, white text appeared at the bottom of the window.

```
\Naked Princess\
\version NSF\
\Would you like to download my
➥ vision? (Y/N)\
```

My heart skipped a beat. Downloading a

vision.... Was this the Naked Princess's method for showing me the creepy and disturbing picture I'd seen? Would doing this kick out another picture? Was it really this easy to do?

There was only one way to answer these questions. I slowly pressed the "Y" key.

```
\Hmm, I'm not ready.\
\Let's talk first. Get to know
➥ each other before we Netflix
➥ and chill.\
\Who are you?\
```

Never one to take any innocent question seriously, I typed back:

```
Franklin W. Dixon
```

That's when the conversation got weird.

```
\Processing that...\
\Come on. You're lying to me.\
```

The last line was highlighted in red.

This was odd. It was an old-school text interface, but the conversation so far implied I was dealing with complexity and intelligence. Although maybe it used this same response with everyone who ran the program. I decided to test it with some potential for stress and conflict.

```
No really. I am. My friends call
➥ me Frank.
\Yeah, and I'm Bill Gates. The
➥ wiki-matrix-hive-mind knows
➥ all, silly human. Tell me who
➥ you are or I'll hold your
➥ breath until you turn blue.\
```

This might be a really clever AI, a tool programmed with personality and snarky threats to personal safety. It could also be a link to an outsourced location. Was I chatting with an actual human? On impulse, I left the program running, and disabled my Internet connection. The response was immediate.

```
\Wait. I need that.\
```

Interesting.

I waited a few seconds, but the program said nothing more.

I turned my Internet back on. The response came back, again in white text:

```
\Ah, that's better. Now again,
➥ for realsies: Who are you?\
```

I ran a few monitoring tools and watched the Naked Princess in byte-level detail. Encrypted packets were blasting out to dozens of locations in China, Russia, and North Korea. I saw no consistency or pattern... apart from each location being an easy-to-compromise enemy nation of the United States. Whatever or whoever the Naked Princess was talking to, it had a lot of friends overseas, friends that looked like a distributed network. Or a botnet.

I thought about the brain - electronic or human - behind the glowing lines sitting so patiently on my screen. The language was strange. Not strange to *me,* it just wasn't right for this situation. Meaning that in my many years of talking with overseas tech support, none of them had ever used casual slang, figures of speech, or goofy language. That wasn't the technique of ESL speakers trying to communicate well. Whoever was on the other end of this output was likely an English-native speaker. And given the appearance of four pop-culture references in this short conversation already, they were probably American.

I typed a response.

```
My name is Dev Manny. Information
➥ Technology Private
➥ Investigator.
\Processing that...\
\...3.2K data points agree.
➥ Okay, I believe you. Let's do
➥ this.\
\What is your FriendyFace
➥ profile?\
```

I paused a moment, trying to understand the reason for the question. The Naked Princess had just said we needed to get to know each other. Okay, although this was a strange way to go about it.

It was a safe bet to assume I had a FriendyFace account - most of the Net-connected world did. But there were always pathetic exceptions. And as my fourth-grade teacher had constantly reminded me, I was one of them. Until very recently, I didn't have a FriendyFace account. I'd only built a profile - a fake one with fake personal data - while I was tracking down P@nic. She was the one who was so socially-connected, not me. Still, I typed in the identifiers for the dummy account I'd built.

```
\Processing that...\
\What is your SyncedIn profile?\
```

It continued to ask for more and more social media accounts. I didn't have any, so I filled what was asked by using the dummy accounts I'd set up in my search for P@nic. After each one, CPU and Internet use continued to spike. The Naked Princess ended this sequence with a reassuring and ominous:

```
\Processing that...\
\...Done.\
\Would you like to download my
➥ vision? (Y/N)\
```

You better believe I hit "Y".

Listed here are some upcoming events of interest to hackers. Hacker conferences generally cost under $200 and are open to everyone. Higher prices may apply to the more elaborate events such as outdoor camps. If you know of a conference or event that should be known to the hacker community, ***email us*** at **happenings@2600.com** or by snail mail at **Hacker Happenings, PO Box 99, Middle Island, NY 11953 USA.** We only list events that have a firm date and location, aren't ridiculously expensive, are open to everyone, and welcome the hacker community.

January 15-17
ShmooCon
Washington Hilton Hotel
Washington DC
www.shmoocon.org

March 4-6
CarolinaCon-12
Hilton North Raleigh/Midtown
Raleigh, North Carolina
www.carolinacon.org

March 25-28
Easterhegg 2016
Salzburg. Austria
eh16.easterhegg.eu

April 23-24
Maker Faire U.K.
Life Science Center
Newcastle upon Tyne, England
www.makerfaireuk.com

May 5-6
THOTCON 0x7
Chicago, Illinois
thotcon.org

May 20-22
Maker Faire Bay Area
San Mateo Event Center
San Mateo, California
www.makerfaire.com

May 20-22
NolaCon
Crowne Plaza New Orleans
New Orleans, Louisiana
nolacon.com

June 8-12
ToorCamp
Doe Bay Resort
Orcas Island, Washington
toorcamp.toorcon.net

June 10-12
CircleCityCon
Westin Indianapolis
Indianapolis, Indiana
circlecitycon.com

July 22-24
The Eleventh HOPE
Hotel Pennsylvania
New York City, New York
xi.hope.net

August 4-7
DEF CON 24
Paris/Bally's
Las Vegas, Nevada
www.defcon.org

Please send us your feedback on any events you attend and let us know if they should/should not be listed here.

Marketplace

Events

THE ELEVENTH HOPE. *2600* presents the eleventh Hackers On Planet Earth conference at New York City's HOtel PEnnsylvania July 22-24, 2016. Visit xi.hope.net for the latest news, travel info, special hotel rates, etc. Speakers wanted: email speakers@hope.net. Volunteers wanted: email volunteers@hope.net. Vendors wanted: email vendors@hope.net. Projects wanted: email projects@hope.net. You get the idea. You can help define what The Eleventh HOPE focuses on and be a real part of hacker history, right in the middle of midtown Manhattan, across the street from the busiest train station in America. You can also join our announcement mailing list from the main page of our website. Call (212) PEnnsylvania 6-5000 for the special conference room rate.

For Sale

HACKER CLOTHING & LOCK PICKS - HackerStickers.com has a growing selection of hacker, gamer, geek, and security advocate clothing, hardware, caffeine, stickers, lock picks, patches, pins, etc. *2600* readers get a free sticker with any order. Add a sticker to cart and enter code "FREESTICK" at checkout at HackerStickers.com.

PRIVACYSCAN seeks & destroys privacy threats on the Mac wiping your tracks on where you surf and what you do on your computer. Learn more at http://privacyscan.securemac.com/

BLUETOOTH SEARCH FOR ANDROID searches for nearby discoverable Bluetooth devices. Runs in the background while you use other apps, recording devices' names, addresses, and signal strength, along with device type, services, and manufacturer. Handles Bluetooth Classic and Bluetooth LE (on LE-equipped Android devices). This is a valuable tool for anyone developing Bluetooth software, security auditors looking for potentially vulnerable devices, or anyone who's just curious about the Bluetooth devices in their midst. Exports device data to a CSV file for use in other programs, databases, etc. If you've used tools like btscanner, SpoofTooph, Harald Scan, or Bluelog on other platforms, you need Bluetooth Search on your Android device. More info and download at http://tinyurl.com/btscan.

CLUB-MATE is now easy to get in the United States! The caffeinated German beverage is a huge hit at any hacker gathering. Available in two quantities: $36.99 per 12 pack or $53.99 per 18 pack of half liter bottles plus shipping. Write to contact@club-mate.us or order directly from store.2600.com. We are now working to supply stores nationwide - full details at club-mate.us.

A TOOL TO TALK TO CHIPS. It's the middle of the night. You compile and program test code for what must be the 1000th time. Digging through the datasheets again, you wonder if the problem is in your code, a broken microcontroller... who knows? There are a million possibilities, and you've already tried everything twice. Imagine if you could take the frustration out of learning about a new chip. Type a few intuitive commands into the Bus Pirate's simple console interface. The Bus Pirate translates the commands into the correct signals, sends them to the chip, and the reply appears on the screen. No more worry about incorrect code and peripheral configuration, just pure development fun for only $30 including world wide shipping. Check out this open source project and more at DangerousPrototypes.com.

HACKER WAREHOUSE is your one stop shop for hacking equipment. We understand the importance of tools and gear which is why we strive to carry only the highest quality gear from the best brands in the industry. From WiFi Hacking to Hardware Hacking to Lock Picks, we carry equipment that all hackers need. Check us out at HackerWarehouse.com.

PROTECT YOUR PRIVACY ONLINE. FoxyProxy sells VPN and proxy services. Why choose us? We've been around since 2006 and have always been privately owned, independently operated. We don't have any shareholders or venture capitalists to satisfy by compromising your privacy. No advertising. No logging. No spamming or marketing emails. We don't sell your email address and other information. WE ARE HUGE OPEN-SOURCE CONTRIBUTORS. All accounts come with both VPN AND proxy service. Choose from 60 different countries. Use coupon code "2600-hope" for 10% off any purchase. getfoxyproxy.com

OPEN POWER: *Electoral Reform Act of 2015 - Open Source Activist Tool Kit* by HOPE speaker Robert Steele available on the Kindle and at amazon.com

Announcements

AUSTIN HACKERSPACE: A shared workshop with electronics lab, laser cutters, 3D printers, CNC machines, car bay, woodworking, and more! $60/mo for 24/7 access to all this and a great community as well. Open House and open meetups weekly. 9701 Dessau Rd, Austin, TX http://atxhs.org/

HAVE YOU SEEN THE NEW *2600* STORE? We've finally made the jump into the 21st century with a store that has more features, hacker stuff, and endless possibilities than ever before. We now accept Bitcoin and Google Wallet, along with the usual credit cards and PayPal. We have more digital download capability for the magazine and for HOPE videos. Best of all, we've lowered prices on much of our stock. Won't you pay us a visit? store.2600.com

Wanted

VETERANS: For the website and a possible anthology, Medic in the Green Time.com wants your response to the ubiquitous phrase "Thank you for your service." What do you think and feel when you hear it? How do you reply? Guidelines: 3-5 pages. Include contact info, branch of service, rank, tour dates, job, service or current photo. See examples at Medic in the Green Time.com >>> Post War >>> Five Simple Words. Send to: silverspartan@gmail.com

I NEED A LAWYER for ineffective assistance of counsel. I'm Jesse McGraw (3:09-CR-210-B), and seeking relief for damages against prior attorney for breach of fiduciary duty, attorney-client confidentiality, and effective abandonment of my 2255 Appeal. He tried to inform against me, my 2255 was never filed, and there's critical (Brady) evidence missing from my case file. (I'm on a botnet/hacking case.) freejesselegalteam@hush.ai, freejesselegal.wix.com/freejesse

Services

DOUBLEHOP.ME is an edgy VPN startup aiming to rock the boat with double VPN hops and encrypted multi-datacenter interconnects. We enable our clients to VPN to country A, and transparently exit country B. Increase your privacy with multiple legal jurisdictions and leave your traditional VPN behind! We don't keep logs, so there's no way for us to cooperate with LEOs, even if we felt compelled to; we simply respond with one liners from 50 shades. We accept Bitcoin and promote encrypted registration over Telegram Messenger. Use promo code

COSBYSWEATER2600 for 50% off, to celebrate our launch! https://www.doublehop.me

HACKERS, PHREAKERS, COMPUTER NERDS. Feel disillusioned, depressed, and dissatisfied with the way your life is passing? Need love, happiness, togetherness, and financial freedom? Here is the solution. Be with us to be yourself. You can be independent by joining with your kind. Enjoy the possibilities of collective thought, with associates who feel and think just like you do. Break that old routine, and dare to explore something new and unique. Contact THE HUB at: P. Bronson, P.O. Box 1000-AF8163, Houtzdale, PA 16698-1000.

FBI FILES - Public service websites GetGrandpasFBIfile.com and GetMyFBIfile.com provide simple form letters to get dossiers from the FBI and other agencies. Free of charge. You can also print out the blank request templates if you prefer not to share personal information while using the website.

BOBBY JOE SNYDER TEACHES HACKING (or an aspect of hacking). The lesson is geared to anyone with an algebra 2 knowledge. The equation is number theory so complex patterns are described by simple algebra. Usually the audience is math literate. But even being math literate, the way I describe the problem may sound complicated. The problem is broken down in these lessons in a way that will make the equations best understood. Lesson 1: https://onedrive.live.com/redir?resid=6DA091A54585CF8E!176&authkey=!AClfEOTHrMEGqy0&ithint=file%2cpptx. Lesson 2: https://onedrive.live.com/redir?resid=6DA091A54585CF8E!183&authkey=!AFCufpJkTzUNs54&ithint=file%2cpptx. Lesson 3: https://onedrive.live.com/redir?resid=6DA091A54585CF8E!188&authkey=!AGLL_XUnRG2ooGE&ithint=file%2cpptx

GET YOUR HAM RADIO LICENSE! KB6NU's "No-Nonsense" study guides make it easy to get your Technician Class, General Class, or Extra Class amateur radio license. They clearly and succinctly explain the concepts, while at the same time give you the answers to all of the questions on the test. The PDF version of the Technician Class study guide is free, but there is a small charge for the other versions. All of them are available from www.kb6nu.com/study-guides. Paperback versions are also available from Amazon. E-mail cwgeek@kb6nu.com for more information.

ANTIQUE COMPUTERS. From Altos to Zorba and everything in between - Apple, Commodore, DEC, IBM, MITS, Xerox... vintagecomputer.net is full of classic computer hardware restoration information, links, tons of photos, video, document scans, and how-to articles. A place for preserving historical computers, maintaining working machines, running a library of hard-to-find documentation, magazines, SIG materials, BBS disks, manuals, and brochures from the 1950s through the early WWW era. http://www.vintagecomputer.net

LISTEN TO THE GR3YNOISE PODCAST. The podcast formerly known as the SYNACK Pack is now GR3YNOISE! There are many security minded podcasts out there, and we're one of them. We are here for the newbies and veterans alike! The GR3YNOISE podcast discusses general news as well as technology specific issues, all from a hacker perspective. Recorded at the SYNShop Hackerspace in Las Vegas, NV. Have a listen and we LOVE feedback! http://greynoi.se.

DATA RAIN SOLUTIONS is a budding Colorado IT startup specializing in reliable and affordable remote tech support in advanced malware removal, PC optimization, diagnostics, and more. *2600* subscribers get 10% off their first order, as-need basis, or 1 year sub. Contact us: shanaroneasomi@yahoo.com. Visit us: http://shanaroneasomi.wix.com/datarain. Join the team! (Hackers welcome)

INTELLIGENT HACKERS UNIX SHELL: Reverse.Net is owned and operated by intelligent hackers. We believe every user has the right to online security and privacy. In today's hostile anti-hacker atmosphere, intelligent hackers require the need for a secure place to work, compile, and explore without big-brother looking over their shoulder. Hosted in Chicago with Filtered DoS Protection. Multiple Dual Core FreeBSD servers. Affordable pricing from $5/month, with a money back guarantee. Lifetime 26% discount for *2600* readers. Coupon Code: Save2600. http://www.reverse.net/

DIGITAL FORENSICS FOR THE DEFENSE! Sensei Enterprises believes in the Constitutional right to a zealous defense, and backs up that belief by providing the highest quality digital forensics and electronic evidence support for criminal defense attorneys. Our veteran experts are cool under fire in a courtroom - and their forensic skills are impeccable. We recover data from many sources, including computers, external media, and smartphones. We handle a wide range of cases, including hacking, child pornography possession/distribution, solicitation of minors, theft of proprietary data, data breaches, interception of electronic communications, identity theft, rape, murder, embezzlement, wire fraud, racketeering, espionage, cyber harassment, cyber abuse, terrorism, and more. Sensei's digital forensic examiners all hold prestigious forensic certifications. Our principals are co-authors of *The Electronic Evidence Handbook* (American Bar Association 2006) and hundreds of articles on digital forensics and electronic evidence. They lecture throughout North America and have been interviewed by ABC, NBC, CBS, CNN, Reuters, many newspapers, and even Oprah Winfrey's *O* magazine. For more information, call us at (703) 359-0700 or email us at sensei@senseient.com.

SECURE UNIX SHELLS & HOSTING SINCE 1999. JEAH.NET is one of the oldest and most trusted for fast, stable shell accounts. We provide hundreds of vhost domains for IRC and email, the latest popular *nix programs, access to classic shell programs and compilers. JEAH.NET proudly hosts eggdrop, BNC, IRCD, and web sites w/SQL. *2600* readers' setup fees are always waived. BTW: FYNE.COM (our sister co.) offers free DNS hosting and WHOIS privacy for $3.50 with all domains registered or transferred in!

Personal

LIVING THE FAST LIFE has slowed mine down dramatically. Looking to shoot the shit with intellects and the likes from all walks of life. Anything from coding to social engineering. Enjoy learning the science of everything. Today I'm grateful to hear from those who look past my poor choices and even those who don't/won't. If you have it on your mind to reach out, do so. Shouts out to nachash and keep your head up Stormbringer. Elijah Cotchery 01751459, Coffield Unit, 2661 FM 2054, Tennessee Colony, TX 75884.

STORMBRINGER IS STILL ALIVE AND WELL in Club Fed. 15 yrs thus far, 4½ to go. Looking for correspondence to pass the time. A geekette would be nice, but anyone can also write. Looking also for white papers in infosec, networking, etc., so I can hit the ground running upon release. Henry French, where R U? www.freestormbringer.com W.K. Smith, 44684-083, P.O. Box 999, Butner, NC 27509-0999.

ONLY SUBSCRIBERS CAN ADVERTISE IN *2600!* Don't even think about trying to take out an ad unless you subscribe! All ads are free and there is no amount of money we will accept for a non-subscriber ad. We hope that's clear. Of course, we reserve the right to pass judgment on your ad and not print it if it's amazingly stupid or has nothing at all to do with the hacker world. We make no guarantee as to the honesty, righteousness, sanity, etc. of the people advertising here. Contact them at your peril. All submissions are for ONE ISSUE ONLY! If you want to run your ad more than once you must resubmit it each time. Don't expect us to run more than one ad for you in a single issue either. Include your address label/envelope or a photocopy so we know you're a subscriber. If you're an electronic subscriber, please send us a copy of your subscription receipt. Send your ad to *2600* Marketplace, PO Box 99, Middle Island, NY 11953. You can also email your ads to subs@2600.com.

Deadline for Spring issue: 2/21/16.

THE ELEVENTH HOPE
July 22nd, 23rd, and 24th, 2016
New York City

Yes, it's happening. By the time you read this, we will be in the midst of organizing our next conference at the world-renowned Hotel Pennsylvania. Last time we had Edward Snowden and Daniel Ellsberg as our keynotes. Previously we've hosted the likes of Steve Wozniak, Jello Biafra, Kevin Mitnick, Adam Savage, and Richard Stallman. It's too early to tell you who's going to be our keynote this time, but we guarantee over 100 awesome talks with some of the best minds in the hacker community, plus three days and nights of nonstop activity (hardware hacking, breakout sessions, workshops, lockpicking, concerts, Segway rides, hacker art displays, and so much more) taking up three full floors right in the middle of Manhattan.

PREREGISTRATION NOW OPEN!

Just visit store.2600.com and order your tickets. We take all major credit cards and Bitcoin! You'll get an email confirmation and you'll be all set. Thanks to the hard work of so many great people, we're able to continue to keep the price extremely low for a conference of this kind in the middle of one of the busiest places in the world. (Check hope.net for special deals on hotel rooms for conference attendees.)

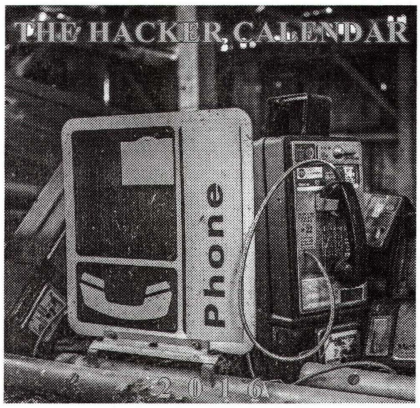
Interested in speaking? We want to hear from you! Just email speakers@hope.net and let us know in a few paragraphs what you want to address, who you are, and any other relevant info. Guidelines can be found at hope.net.

We couldn't do any of this without the hundreds of volunteers who pitch in to make it all happen. If that sounds like something you want to be part of, just send an email to volunteers@hope.net and let us know if you have a particular talent or if you just want to help out wherever help is needed.

We're taking it to Eleven. And we hope to see you there.

iT'S 2016!

If you don't have a 2016 Hacker Calendar on your wall, we guarantee you're going to miss something. Maybe you won't actually forget what day it is (although you'd be surprised how many people do). But you will certainly miss learning what happened in the world of hacker history for just about every day of the year. Some days have three or more different events! And you will definitely miss out on all of the amazing 12"x12" glossy photos of payphones from around the world. With the 2016 Hacker Calendar, you get art, education, and a useful tool. All for only $9.99 from store.2600.com/calendar



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 1-9 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.

"We live in a society exquisitely dependent on science and technology, in which hardly anyone knows anything about science and technology." - Carl Sagan

***2600* is written by members of the global hacker community.**
You can be a part of this by sending your submissions to articles@2600.com or the postal address below.

2600 (ISSN 0749-3851, USPS # 003-176); Winter 2015-2016, Volume 32 Issue 4, is published quarterly by 2600 Enterprises Inc., 2 Flowerfield, St. James, NY 11780. Periodical postage rates paid at St. James, NY and additional mailing offices.

POSTMASTER:
Send address changes to: *2600*
P.O. Box 752 Middle Island,
NY 11953-0752.

SUBSCRIPTION CORRESPONDENCE:
2600 Subscription Dept., P.O. Box 752,
Middle Island, NY 11953-0752 USA
(subs@2600.com)

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.
2000-2015 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**

MEETINGS

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 (Surrey): Central City Shopping Centre food court by Orange Julius.

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.

CZECH REPUBLIC

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: Fenniakortteli food court (Vuorikatu 14).

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: Cafe Monde et Medias, Place de la Republique. 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: Bar 1929, Slavyanskaya Square 2. 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), lowest level. 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: near the Cenotaph in George Square. 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: HeatSync Labs, 140 W Main St. 6 pm
Prescott: Method Coffee, 3180 Willow Creek Rd. 6 pm

Arkansas

Ft. Smith: River City Deli at 7320 Rogers Ave. 6 pm

California

Anaheim (Fullerton): The Night Owl, 200 N Harbor Blvd. 7 pm
Chico: Starbucks, 246 Broadway St. 6 pm
Los Angeles: Union Station, inside main entrance (Alameda St side) near the Traxx Bar.
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. 6 pm

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: O'Brothers Irish Pub, 1521 Margaret 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: Krystal Hamburgers, 2914 S Washington Ave (US-1).

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: Tomlinson Tap Room in City Market, 222 E Market St.

Iowa

Ames: Memorial Union Building food court at the Iowa State University.
Davenport: Co-Lab, 1033 E 53rd 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
Worcester: TESLA space - 97D Webster St.

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, 2400 S Jefferson Ave.

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): 1075 American Pacific Dr Suite C. 7 pm
Reno: Barnes & Noble Starbucks 5555 S. Virginia St.

New Hampshire

Keene: Local Burger, 82 Main St. 7 pm

New Jersey

Morristown: Panera Bread, 66 Morris St. 7 pm
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. 7 pm
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
Memphis: Republic Coffee, 2924 Walnut Grove Rd. 6 pm
Nashville (Franklin): CoolSprings Galleria food court, 1800 Galleria Blvd. 6 pm

Texas

Austin: The Chicon Collective, 301 Chicon St, Suite D. 7 pm
Dallas: Wild Turkey, 2470 Walnut Hill Ln. 7 pm
Houston: 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: Washington State Convention Center. 2nd level, south side. 6 pm
Spokane: The Service Station, 9315 N Nevada (North Spokane).
Tacoma: Tacoma Mall food court. 6 pm

Wisconsin

Madison: Fair Trade Coffee House, 418 State St.

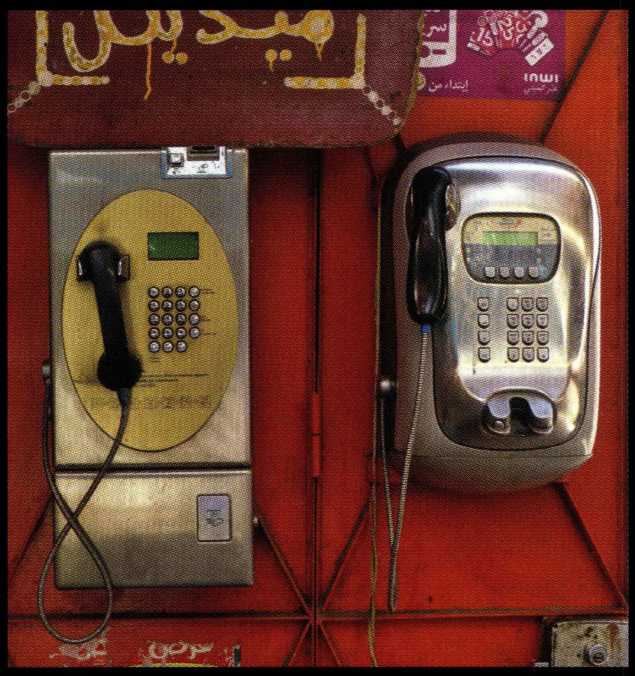
All meetings take place on the first Friday of the month (a * indicates a meeting that's held on the first Thursday of the month). Unless otherwise noted, *2600* meetings begin at 5 pm local time. To start a meeting in your city, send email to meetings@2600.com.

Follow @2600Meetings on Twitter and let us know your meeting's Twitter handle!

Payphones From All Over

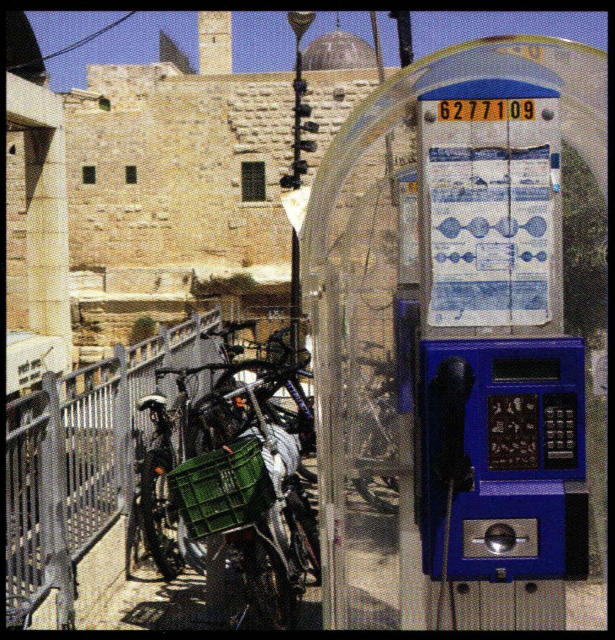
Malaysia. Found in the capital city of Kuala Lumpur, this payphone apparently once brought bad news to somebody who didn't take it very well.

Photo by Charlotte White



Morocco. Two phones in Marrakech - a standard no-nonsense coin-accepting payphone and a curvy stylish model that only takes cards.

Photo by Howard Feldman



Israel. Seen inside the Old City of Jerusalem with the Western Wall and the Al-Aqsa Mosque atop the Temple Mount in the background. It doesn't get more peaceful than this.

Photo by Bavs

Turkey. Where thoughts naturally turn to dolphins. Seen in Istanbul, but apparently they exist all over the country as we've gotten multiple submissions of these things.

Photo by Peter Vibert

Visit **http://www.2600.com/phones/** to see our foreign payphone photos!
(Or turn to the inside front cover to see more right now.)

The Back Cover Photos

Only in Japan would you be able to find food containers that are somehow related to UNIX. There's so much potential here. Thanks to **Randy Frank** for sending this in, as well as for placing this on top of a very special television set in order to make this shot even more memorable.

This may just be the coolest street in all of San Antonio, Texas, as discovered by **Abel Lopez**. Let's hope its residents realize just how lucky they are.

If you've spotted something that has "2600" in it or anything else of interest to the hacker world (such as funny uses of "hacker," "unix," "404," you get the idea...), take a picture and send it on in! Be sure to use the highest quality settings on your camera to increase the odds of it getting printed. Make sure and tell us where you spotted your subject along with any other info that makes it interesting - many photos are eliminated due to lack of detail.

Email your submissions to articles@2600.com or use snail mail to *2600* Editorial Dept., PO Box 99, Middle Island, NY 11953 USA.

If we use your picture, you'll get a free one-year subscription (or back issues) or a *2600* t-shirt of your choice.