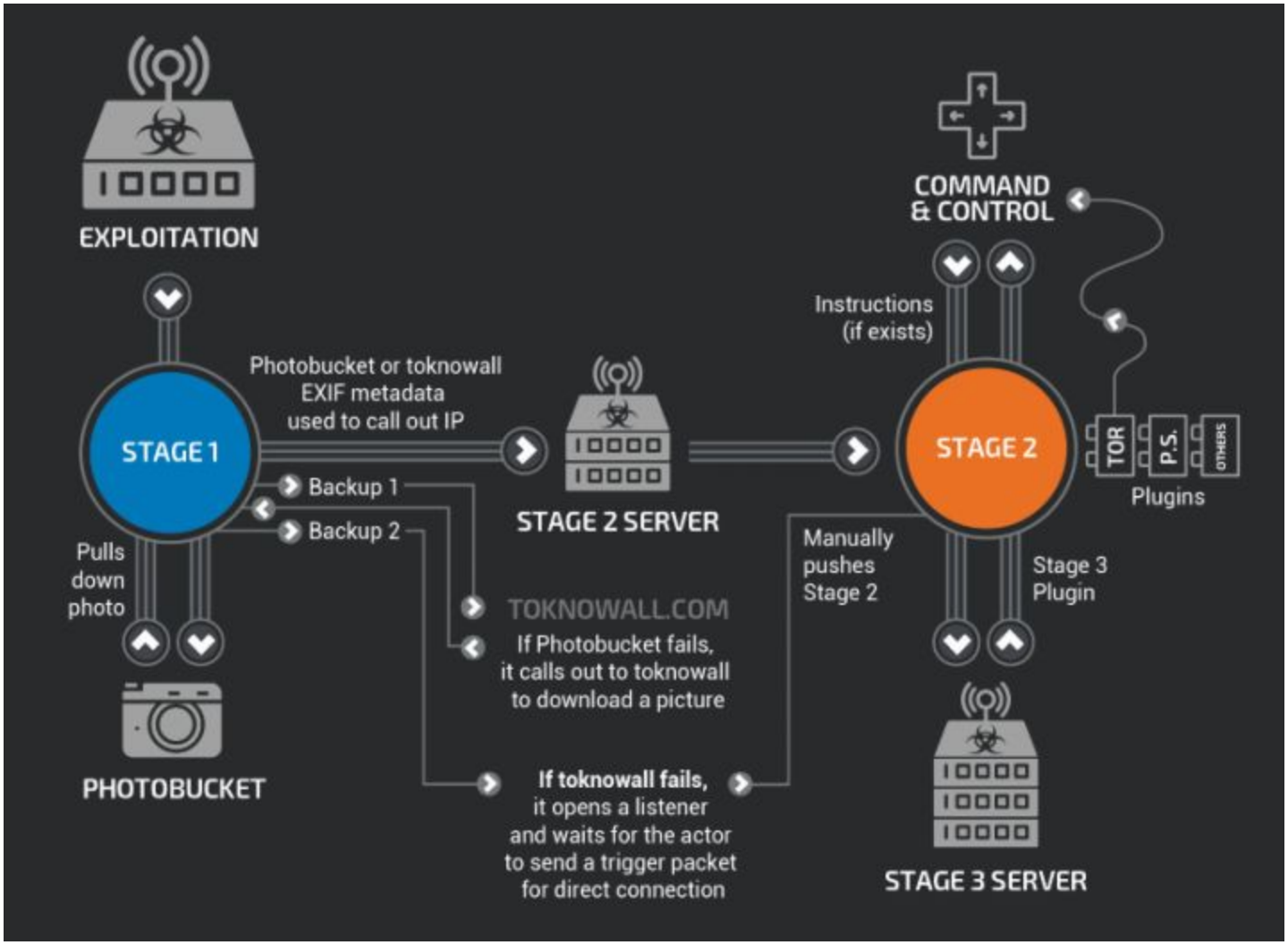
Security Now! #665 - 05-29-18

VPNFilter

This week on Security Now!

This week we discuss Oracle's planned end of serialization, Ghostery's GDPR faux paus, the emergence of a clever new banking Trojan, Amazon Echo and the case of the Fuzzy Match, more welcome movement from Mozilla, yet another steganographic hideout, an actual real-world appearance of HTTP Error 418 (I'm a Teapot!), the hype over Z-Wave's Z-Shave, and a deep dive into the half a million strong VPNFilter botnet.

Our Picture of the Week



Security News

Oracle plans to drop Java's built-in serialization mechanism
Mark Reinhold, chief architect of the Java platform group at Oracle called Java's Object serialization "A horrible mistake made in 1997" which has been a constant thorn in Oracle's side and has been responsible for nearly half of Java's many security troubles through the years... including the infamous Apache Struts vulnerability which bit Equifax after they failed to patch in a timely fashion.

Remember: A deserializer is a potentially complex interpreter... and one of this podcasts recurring observations is that interpreters are quite difficult to make perfect.

By leveraging Java's error-prone deserialization, an enterprise web app server, in its default install, allowed unauthenticated remote code execution via a single web request, allowing attackers to remotely run code on the web server and access files and bypass all security controls.

Mark Reinhold said that serialization, overall, is brittle but holds the appeal of being easy to use in simple use cases.

So... to replace Java's current serialization technology, a small serialization framework would be placed in the platform once records, the Java version of data classes, are supported. The framework could support a graph of records, and developers could plug in a serialization engine of their choice, supporting formats such as JSON or XML, enabling serialization of records in a safe way. But Reinhold cannot yet say which release of Java will have the records capability.

https://www.infoworld.com/article/3275924/java/oracle-plans-to-dump-risky-java-serialization.html

Ghostery: How NOT to be GDPR compliant!
So you're Ghostery...

In a recent tweet dated May 25th, 2018 you declare: "We at @Ghostery hold ourselves to a high standard when it comes to users' privacy"

And you have recently decided to take your previously outsourced eMail list management in-house in order to carefully manage the privacy of your users and take full responsibility for their care.

In the wake of the new GDPR regulations you decide to send an update eMail to each of your registered users containing a privacy policy update.

Unfortunately, the person who is put in charge of this task is apparently insufficiently familiar with the operation of the new in-house eMail system. And is sent out to users, against all reason, expectation and understanding are eMail's containing -- I kid you not -- a series of eMails, each containing a To: header stuffed to the brim with a list of 500 of Ghostery's privately registered user email addresses.

GDPR anyone??

Twitter had some fun with this, with tweets such as:

Matt: "you cannot make this up: the privacy driven Chrome extension @Ghostery for blocking third-party trackers has done its GDPR mass email exposing all(?) of its users in the email TO: field!!

Daniel Tsekhman: Hey @Ghostery. You sent your privacy policy update email without blind CCing the contacts. Now I'm on an email chain with a whole bunch of randoms replying to your no-reply email address.

Dan Barker: Weird move from @Ghostery :
1. Accidentally share thousands of email addresses with users. (in a GDPR email)
2. Apologize for it on Twitter.
3. Delete the apology tweet.

Last Friday: ESET Security posted about "BackSwap"
https://www.welivesecurity.com/2018/05/25/backswap-malware-empty-bank-accounts/
“BackSwap malware finds innovative ways to empty bank accounts” - ESET researchers have discovered a piece of banking malware that employs a new technique to bypass dedicated browser protection measures

Last year, 33% of all phishing campaigns were attempts to deliver banking Trojans.

Just the top 10 are: Zeus (aka Zbot - and after its source code leaked, Citadel, Atmos, FlokiBot), Neverquest (aka Vawtrak aka Snifula), Gozi (aka Ursnif), Dridex (aka Bugat aka Cridex), Ramnit, GozNym, Tinba, Gootkitm Qadars, Ronvix

<quote> Banking malware (also referred to as banker) has been decreasing in popularity among cybercrooks for a few years now, one of the reasons being that both anti-malware companies and web browser developers are continuously widening the scope of their protection mechanisms against banking Trojan attacks. This results in conventional banking malware fraud becoming more complicated to pull off every day, resulting in malware authors shifting their time and resources into developing easier-to-make and more profitable types of malware like ransomware, cryptominers, and cryptocurrency stealers.

BackSwap passively monitors the user's activity watching for banking activity. When observed it uses two tricks to get its own interception code to execute: It quickly launches the browser's own JavaScript developer console or types the JavaScript directly into the browser's URL field.

The URL scheme "javascript://" is

The malicious payload is initially delivered as a modified version of a legitimate application that's been partially overwritten. The application used as the target for the modification is being changed regularly but some examples of previous apps misused include TPVCGateway, SQLMon, DbgView, WinRAR Uninstaller, 7Zip, OllyDbg, FileZilla Server.

The malware copies itself into the Windows startup folder to obtain persistence.

Complications afflicting traditional banking Trojans:

To inject into the browser's process, 32 or 64 bitness. Browsers and A/V are becoming very good about catching and stopping process injection.

Entrypoint "hooks" must be found to grab text before encryption and after decryption. The functions vary from browser to browser, from version to version, from OS to OS. This makes "hooking" extremely problematic. And Chromium-based browsers have these functions deeply embedded in their binary code, making their discovery very difficult. And even IF the proper hooking locations can be found they may still be thwarted by add-on security solutions.

What's a poor banking Trojan to do????

Get a load of this: The recently discovered "BankSwap" Trojan handles EVERYTHING by working with the Windows GUI elements and by simulating user input.

Think of it as "malicious keystroke macros" running on the user's desktop.

This allows the malware to remain completely agnostic to the browser's inner depths since it no longer needs to interact with the browser's process at all. This, in turn, means that it does not require any privileges and bypasses any 3rd-party hardening of the browser. And it no longer depends upon the bitness or architecture of the browser. or on its version. One solution runs universally.

Using the Windows accessibility interface, the malware monitors the user's activities from afar -- in the background -- watching and URL events which match known patterns. The malware will then look for bank-specific URLs and window titles in the browser that indicate that the victim is about to make a wire transfer.

Once identified, the banking malware loads malicious JavaScript appropriate for the corresponding bank from its resources and injects it into the browser. The script injection is also done in a simple, yet effective way.

In older samples, the malware originally inserted the malicious script into the clipboard and simulated pressing the key combination for opening the developer’s console (CTRL+SHIFT+J in Google Chrome, CTRL+SHIFT+K in Mozilla Firefox) followed by CTRL+V, which pastes the content of the clipboard and then sends ENTER to execute the contents of the console. The malware then sends the console key combination again to close the console. The browser window is also made invisible during this process – to regular users it might seem as if their browser simply froze for a moment.

This approach has been upgraded in newer variants of the malware: Instead of interacting with the developer’s console, the malicious script is executed directly from the address bar, via JavaScript protocol URLs... which is a little-used feature supported by most browsers. The malware simply simulates pressing CTRL+L to select the address bar followed by the DELETE key to clear the field, then “types” in “javascript:” by calling SendMessageA in a loop, and then

pastes the malicious script with the CTRL+V combination. It then executes the script by sending the ENTER key. At the end of the process, the address bar is cleared to remove any signs of compromise.

And get a load of this... Chrome, Firefox and IE incorporate a protective feature designed as a countermeasure against Self-XSS attacks: When users attempt to paste text starting with "javascript:" into the address bar, the protocol prefix is removed. Users need to enter it into the address bar manually. So, BackSwap bypasses this countermeasure by simulating the typing of the prefix into the address bar, letter by letter, before pasting in the malicious script.

Our takeaway here is: Banking Trojans are not gone. And we need to be very very careful about performing high-value monetary transactions on our computers. If that is something that's being done a lot, with computers being as inexpensive as they are, perhaps consider setting up an older machine with Debian Linux and using it only for banking.

GRC's SQRL client uses a secondary, darkened desktop when it prompts the user to identify themselves. This was primarily done to prevent web browsers from spoofing SQRL's prompt for the user's one SQRL password since no web browser can darken the whole screen outside of its own borders. But this also inherently disconnects any keyboard hooks that might be in the other desktop. During our testing some of our testers complained that their password managers no longer worked -- and that's exactly the point!

Amazon Echo and the Case of the Fuzzy Match

KIRO, Channel 7 in Seattle carried a story last week about a misbehaving Echo.

A Portland family contacted Amazon to investigate after they say a private conversation in their home was recorded by Amazon's Echo, and that the recorded audio was sent to the phone of a random person in Seattle, who was in the family's contact list.

Danielle, who did not want her last name revealed, said: "My husband and I would joke and say I'd bet these devices are listening to what we're saying."

Every room in her family home was wired with the Amazon devices to control her home's heat, lights and security system.

But Danielle said two weeks ago their love for the Echo changed with an alarming phone call. "The person on the other line said, 'unplug your Echo devices right now. 'You're being hacked.'"

That person was one of her husband's employees, calling from Seattle.

"We unplugged all of them and he proceeded to tell us that he had received audio files of recordings from inside our house," she said. "At first, my husband was, like, 'no you didn't!' And the (recipient of the message) said 'You sat there talking about hardwood floors.' And we said, 'oh gosh, you really did hear us.'"

Danielle listened to the conversation when it was sent back to her, and she couldn't believe someone 176 miles away heard it too.

"I felt invaded," she said. "A total privacy invasion. Immediately I said, 'I'm never plugging that device in again, because I can't trust it.'"

Danielle says she unplugged all the devices, and she repeatedly called Amazon. She says an Alexa engineer investigated.

"They said 'our engineers went through your logs, and they saw exactly what you told us, they saw exactly what you said happened, and we're sorry.' He apologized like 15 times in a matter of 30 minutes and he said we really appreciate you bringing this to our attention, this is something we need to fix!"

But Danielle says the engineer did not provide specifics about why it happened, or if it's a widespread issue.

"He told us that the device just guessed what we were saying," she said. Danielle said the device did not audibly advise her it was preparing to send the recording, something it's programmed to do.

When KIRO 7 asked Amazon questions, they sent this response:

"Amazon takes privacy very seriously. We investigated what happened and determined this was an extremely rare occurrence. We are taking steps to avoid this from happening in the future."

Thursday afternoon, Amazon spokeswoman Shelby Lichliter sent this statement:

"Echo woke up due to a word in background conversation sounding like 'Alexa.' Then, the subsequent conversation was heard as a "send message" request. At which point, Alexa said out loud 'To whom?' At which point, the background conversation was interpreted as a name in the customer's contact list. Alexa then asked out loud, '[contact name], right?' Alexa then interpreted background conversation as 'right.' As unlikely as this string of events is, we are evaluating options to make this case even less likely."

So... this was all unlikely as hell, but possible. With voice control we have an inherently soft-matching best-guess system which is inherently prone to failure.

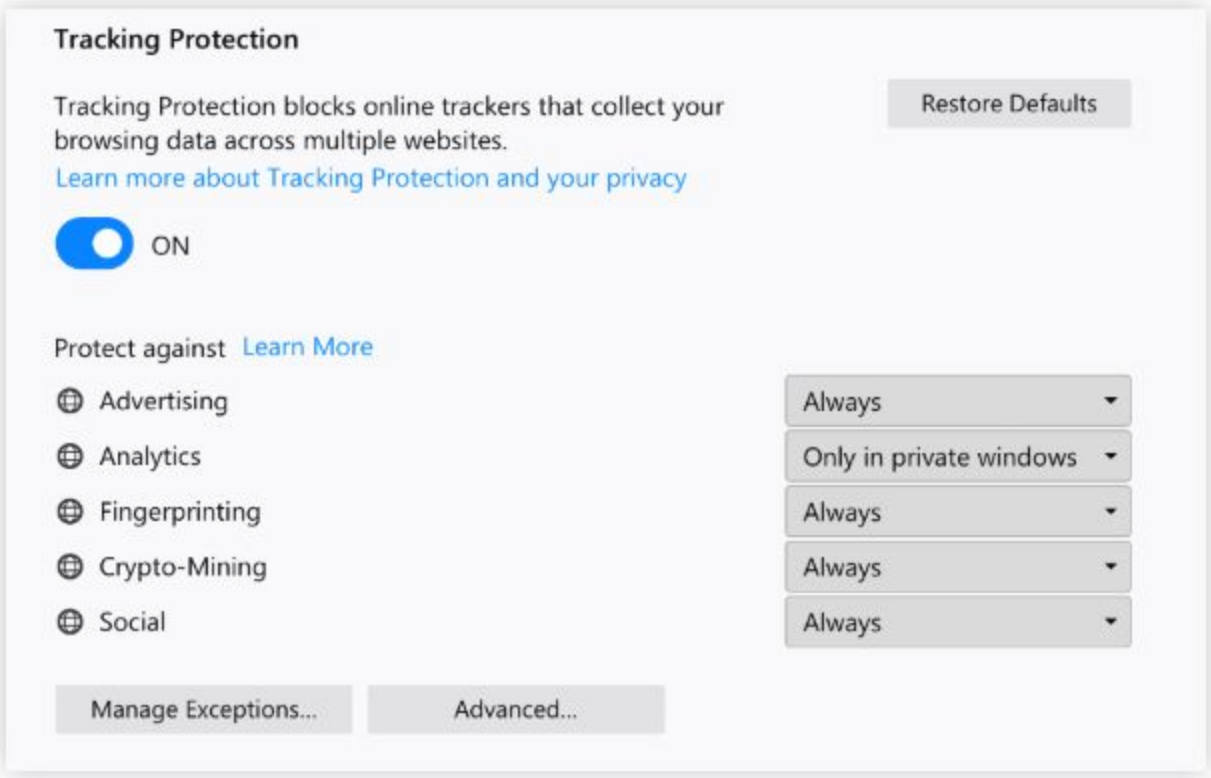
Another of these podcast's enduring and often-examined memes is the inherent tradeoff between security and convenience. Voice control represents a classic, massive tradeoff between security and convenience. Voice control is not secure. It is too prone to failure.

In this instance, Amazon clearly did everything they could reasonably do to use voice control. The only way to make such things secure will be to selectively require a button-press on the device to confirm the user's request.

Firefox 63 will be blocking In-Browser "MalMining" (we're currently at 60)
(I still like that term. It seems clearer than CryptoJacking.)
Remember that by default FF only enables tracking protection within its Private Browsing window. But under Tools / Options / Privacy 7 Security: Tracking Protection "Always" to enable it globally. And this noticeably speeds up web browsing by eliminating a large download burden.

First of all, TP is so useful that future versions of FF will make it more accessible, such as from the main drop-down menu at the upper right.

The term "Tracking Protection" is being expanded with granular and additional options:



Tracking Protection has always blocked advertising, analytics, and social sharing. FFv63 adds protection from web fingerprinting technologies and also web mining scripts. And, each f those can be set to "always", "only in private windows" or "never."

And... per-site exceptions are also supported.

A specific site's cookies and other browser-side storage will be easily clearable from a menu which drops down under the URL lock icon.

And, finally... over the next few weeks, Mozilla will be gradually rolling out OTP-style one-time-passwords for Firefox Mozilla account login. It will be appearing under user account "Preferences" in the coming weeks.

A new steganographic target: Font Glyphs

"Steganography" is the act of hiding in plain sight.

Hiding a secret message within an ordinary message.

"FontCode: Embedding Information in Text Documents using Glyph Perturbation"
http://www.cs.columbia.edu/cg/fontcode/fontcode.pdf
https://arxiv.org/pdf/1707.09418.pdf

Photo LSB tweaking.

Music tone timing and slight detuning.

VERY impressive work.

Posting secret messages in the classifieds.

We've actually had a real world sighting of "ERR! 418 I'm a teapot"!
For seven hours yesterday, many developers using NPM, the Node.js package manger for JavaScript, were receiving an error message we have discussed here previously, and which confused and amused: "ERR! 418 I'm a teapot"

The trouble was seen by those teams operating behind a proxy which was appending the port specifier ":443" to the domain name. This was confusing the NPM registry's servers which resulted in the "I'm a teapot" declaration. Somewhere, someone, decided to return that error for unforeseen conditions.

As we've discussed before, this all hails from the HTCPCP protocol which was once submitted to the IETF -- the Internet Engineering Task Force -- as a geeky joke. HTCPCP stood for the "Hyper Text Coffee Pot Control Protocol"... and it has an official RFC (#2324) dated April 1st, 1998:

https://tools.ietf.org/html/rfc2324

The RFC begins... "Rationale and Scope

There is coffee all over the world. Increasingly, in a world in which computing is ubiquitous, the computists want to make coffee. Coffee brewing is an art, but the distributed intelligence of the web-connected world transcends art. Thus, there is a strong, dark, rich requirement for a protocol designed espressoly for the brewing of coffee. Coffee is brewed using coffee pots. Networked coffee pots require a control protocol if they are to be controlled.

Z-Shave

This one made lots of headlines of the form:
"Z-Shave attack could impact 100 million IoT devices."
"Z-Wave Downgrade Attack Left Over 100 Million IoT Devices Open to Hackers"
"Z-Shave lets you open Z-wave based locks"
"More than 100 Million IoT devices potentially exposed to Z-Shave Z-Wave attack"

We've covered Z-Wave in great detail in the past.

Z-Wave can have no encryption security, S0 level or S2.

If either S0 or S2 security is in place the environment's secret key is known to all Z-Wave devices and all communications is securely encrypted.

But the challenge is in initializing a new device into the network so that it can obtain that key. VERY very unfortunately, Silicon Lab's -- owners of Z-Wave -- did not do this securely the first time... and so now that legacy of weak security is coming back to bite them.

S0 level is weak and has been abandoned because it uses a key of all 0's for its initial network key exchange. This allows any passive eavesdropper who may be listening in on =any= current or future device association or re-association to obtain the secret key for the home's or office's entire network.

They properly solved this initial device pairing problem with their updated S2 security. It uses an elliptic curve Diffie-Helman ECDH key exchange where the counter parties each generate and exchange a random nonce which allows them to establish a key agreement in full view of any eavesdropper without that 3rd-party learning what it now a shared secret. The master station then encrypts the house-master-secret key with the ephemeral shared key and the newly added device is able to jump onto the network without risk.

The problems, however, are still many...

For this S2 system to function, naturally both parties must by S2 capable. Yet S2-capable hubs remain rare and are the exception. Samsung's most popular hub is still not S2-capable. So the initial interaction between devices will necessarily be downgraded to S0 unless both ends understand S2.

But what's worse -- and this is the bit that made the news -- an active attacker can interfere with even as S2-to-S2 pairing in a classic downgrade attack to lead the endpoints to believe that the other end is only S0 capable... and the home's master encryption key can be obtained.

This is a complex attack. It requires a persistent attacker and a pairing-interception. Yet it does work and it is possible.

SpinRite

Wouter (pronounced wow-ter, but you may use Walter instead ;-))
Location: Netherlands
Subject: Replacing hard drives
Date: 18 May 2018 00:26:45

Hi Steve, and Leo,
Thanks for talking about the difference between broken hard drive sectors and failing hard drives. If the described problems and errors do not indicate a failing hard drive, what does?

I once read a general recommendation to replace spinning hard drives every 3 years, but that sounds pretty costly. I do have some older hard drives in use for Time Machine backups.

Should I worry? Some general information on (a) how to spot failing hard drives and (b) when to replace a hard drive, would be greatly appreciated.

Regards, Wouter

Google: SpinRite SMART
https://www.grc.com/sr/smart.htm

VPNFilter

Last week we had an evolving story regarding concerns surrounding a massive botnet of more than 500,000 routers. Yes -- more than half a million routers commandeered into a single botnet!

https://blog.talosintelligence.com/2018/05/VPNFilter.html

This is where the answer to the question: "What could it do?" is a chilling "Anything it wants."

Forensics evidence strongly implicates Russia as the botnet's creator and it is believed to be intended as a network for attacking Ukraine.

Cisco found more than 500,000 instances of this VPNFilter malware on routers manufactured by Linksys, MikroTik, NETGEAR, and TP-Link and also some QNAP NAS devices.

Cisco said that no zero-days were used to create this botnet, just older public vulnerabilities.

Symantec's list of infected routers is:

- Linksys E1200
- Linksys E2500
- Linksys WRVS4400N
- Mikrotik RouterOS for Cloud Core Routers: Versions 1016, 1036, and 1072
- Netgear DGN2200
- Netgear R6400
- Netgear R7000
- Netgear R8000
- Netgear WNR1000
- Netgear WNR2000
- QNAP TS251
- QNAP TS439 Pro
- Other QNAP NAS devices running QTS software
- TP-Link R600VPN

VPNFilter is a highly sophisticated and complex botnet with many moving parts and a plug-in architecture.

First of all, it is only the second router malware known to be persistent across reboots. Earlier this month we discussed the first of these, the "Hide & Seek" botnet which is able to write some of itself into the router's FLASH memory. So in the case of VPNFilter, as with Hide & Seek, simply rebooting one's router is not guaranteed to remove the underlying infection.

The VPNFilter malware consists of three interdependent stages:

Stage One is lightweight and small. Its only role is to infect the device and to obtain boot persistence. If you think about that, this is ideal, since small means it's much more likely to find a nook or cranny inside the router's space-limited file system.

Stage Two and Three do not survive reboots, but they don't need to since the router's Internet connectivity is a given, and Stage One can simply reload another copy of the latest to live in RAM. And also note that our routers are among the least rebooted components of our networks.

So, Stage Two has two functions: First, it provides the hooks for a plug-in architecture used by various Stage Three plug-in modules. So far plug-in modules have been found for sniffing network traffic, detecting the presence of SCADA industrial control network traffic, and communicating with the botnet's command and control servers through the TOR network.

What's Stage Two's second function? It's a FLASH memory wiper. It contains a self-destruct function that overwrites critical portions of the device's firmware with pseudo random noise data, then reboots the device to permanently kill it.

It's one thing to DDoS a network. But imagine the scope of the outage that would result from irreversibly destroying a massive number of routers within a target/victim country!

Okay... so that was the situation earlier last week...

The VPNFilter malware communicates in two different ways:

Once the malware has completed initialization, it attempts to download an image from Photobucket.com, a popular image-sharing site. The malware downloads the first image from the gallery the URL is referencing, then extracts the C&C server's IP address from six integer values encoded into the GPS latitude and longitude of the photo's EXIF information.

If this process fails for any reason, Stage One then looks to a backup domain, toknowall.com from which it downloads an image and attempts the same process. All of this is done over the TOR network or over TLS-encrypted connections.

Secondly, each infected router also establishes a listening port to accept incoming C&C traffic at its public IP address.

So... later last week the US Federal Bureau of Investigation obtained a court order compelling Verisign, the registrar for ToKnowAll.com, to turn over control of the domain to law enforcement.

The FBI had determined that the VPNFilter botnet represented an existential threat to the world after security experts declared VPNFilter to be incredibly dangerous, not only because it was the work of a nation-state cyber group with nefarious purpose, but because it also included functions to intercept network traffic, search for SCADA equipment, and wipe firmware to temporarily brick devices.

As Cisco's Talos group wrote of this threat:
We assess with high confidence that this malware is used to create an expansive, hard-to-attribute infrastructure that can be used to serve multiple operational needs of the threat actor. Since the affected devices are legitimately owned by businesses or individuals, malicious activity conducted from infected devices could be mistakenly attributed to those who were actually victims of the actor. The capabilities built into the various stages and plugins of the malware are extremely versatile and would enable the actor to take advantage of devices in multiple ways.

The Ukrainian Secret Service believed an attack was planned to take place on Saturday when the Ukrainian capital of Kiev would be hosting the UEFA Champions League soccer final.

The FBI confirmed that the botnet was created and was under the control of the famous Russian state-sponsored cyber-espionage unit known under different names including APT28, Sednit, Fancy Bear, Pawn Storm, Sofacy, Grizzly Steppe, STRONTIUM, Tsar Team, and others.

An Estonian Foreign Intelligence Service identifies APT28 as a unit of the Russian Military's Main Intelligence Directorate -- their GRU.

So... with the C&C domain under its control, the FBI has asked everyone in the world who own any of those possibly-infected devices to reboot their routers.

As we know, this alone will not cleanse any affected router of the Stage One infection. But with ToKnowAll.com now under the FBI's control, it will keep the network from obtaining the Stage

Two and Three components.

During this time, the FBI will be monitoring all incoming connections to ToKnowAll.com and thus recording the IP addresses of all routers that phone home. This will give it a strong sense for the network's status and would also allow the IPs to be passed to the relevant ISPs to allow them to, in turn, inform their customers.

What can go wrong?

The Botnet's owners MAY also have a recent list of the infected routers and those routers are also listening for incoming C&C traffic. There is no reporting on what can be done from incoming commands to that network... but it seems unlikely that some provision was not made for recovering from the loss of their C&C domain.

OUR TAKEWAY:

If you own one of the affected devices: Rebooting it is insufficient. Restoring to factory defaults could be done, but reflashing the firmware -- take this opportunity to get the latest -- would be best. And while you're there, be sure to disable all WAN-side management, disable UPnP if you can, and setup a seriously strong username and password.

~30~