Security Now! #642 - 12-19-17

BGP

This week on Security Now!

This week we examine how Estonia handled the Infineon crypto bug, two additional consequences of the pressure to maliciously mine crypto currency, 0-day exploits in the popular vBulletin forum system, Mozilla in the doghouse over Mr. Robot, Win10's insecure password manager mistake, when legacy protocol come back to bite us, hold to bulk-steal any Chrome user's entire stored password vault... and we finally know where and why the uber-potent Mirai botnet was created, and by whom. We also have a bit of errata and some fun miscellany.. then we're going to take a look at BGP, another creaky yet crucial -- and vulnerable -- protocol that glues the global Internet together.

Our Picture of the Week

Security News

Cybernetica Case Study: Solving the Estonian ID-card Case
https://cyber.ee/en/news/cybernetica-case-study-solving-the-estonian-id-card-case/

As we know, due to the shortcuts taken by the crypto system within the widely used Infineon crypto processor embedded in ID cards, the private key buried within the RSA private key was not sufficiently hidden and could be feasibly extracted.

<quote> The Estonian ID-card is used nation-wide for both governmental and private sector services. Several critical processes rely on the operability of the digital identity infrastructure, while some of the systems support the ID-card exclusively (in Estonia, mobile-ID is also available for authentication and digital signatures in many, but not all services). "Shutting down" all ID-cards would have had a severe impact on the entire country, including the economic impact to the businesses, and probably resulted in the deployment of substitute measures with lesser security standards while making several services more costly and time-consuming.

Replacing all the cards physically would have taken a long time, given the necessary steps for creating an entirely new card: choosing the chip, programming and testing the application, acquiring the necessary certification and procuring the new cards equipped with the new chips. Only after these steps would the actual replacement procedure take place, limited by the low amount of available personnel in the Police and Border Guard service points. According to our estimates the process would have taken at least a year, if not more, to complete.

The alternative was to create a solution that would bypass the vulnerability by updating the existing cards. There is a requirement that keys must be generated on-card and never leave the card. This is required in order to be able to use the ID-card to give legally binding digital signatures. The vulnerability that we had to bypass was found in the on-chip RSA key generation procedure. We had to abandon using RSA altogether. Thankfully, the ID-card chip also supported elliptic curve cryptography which was not affected by the found vulnerability. The solution was to update the cards to use elliptic curve cryptography instead of RSA. We analyzed the possibilities to continue with RSA by using alternative key lengths, but ruled them out for several considerations (e.g there was no capability of generating 3k keys within the chip). Moreover, the decisions had to be made before the article on the vulnerability was published. In that situation, migrating to ECC was the most viable decision.

Devising the concept for the solution itself was done rather quickly, mainly due to the lack of alternatives. Most of the time was spent on the development and testing of the ID-card base software and card application, retuning the remote update system, updating the service provider systems to support the elliptic curves.

The two core functions of the Estonian ID-card are authentication and digital signatures. Legally binding digital signatures in Estonia have always been time-stamped, retaining the authenticity of digitally signed documents even throughout this situation, as it is possible to provide evidence that the signature was given before the information about the vulnerability became available.

<< Two interesting consequences of the pressure to mine Cryptocurrencies >>

Believe it or not: Android phone-destroying malware via cryptocurrency mining!
https://www.bleepingcomputer.com/news/security/android-malware-will-destroy-your-phone-no-ifs-and-buts-about-it/

Kaspersky Labs has discovered an evolved version of an older Android malware which was known as Podec. The updated strain is known as Loapi. It has been found in a large number of apps hosted by third-party (non-Google Play) repositories masquerading as adult-themed and bogus antiviral utilities.

However, if users don't remove it from their Android devices, when left to its own means, Loapi will download a Monero cryptocurrency miner that will overheat and overwork the phone's components, ignore throttling power management, make the battery bulge, deform the phone's cover... or worse.

The original Podec malware was used to bypass Advice of Charge (AoC) and CAPTCHAs to subscribe victims to premium-rate SMS services. But today's evolved Loapi malware is far more advanced. Kaspersky experts call it a "jack of all trades," since Loapi has a highly advanced modular structure and components for all sorts of nasty operations including:

- Mine Monero
- Install a proxy to relay traffic
- Inject ads in notification area
- Show ads in other apps
- Open URLs in browsers, also used to show ads
- Download and install other apps
- Launch DDoS attacks
- Interact with the phone's SMS function
- Crawl web pages (to subscribe users to premium SMS services), and more.

Loapi has NOT been detected within the apps offered by the Google play store. So this should be yet another reason to be very selective about what is downloaded into our mobile devices, and from where.

We need to remember that if security is imperfect that means it's porous. And this suggests that the more pressure there is to subvert, the more subversion will occur. Now that malware has picked up on the idea of cycle-stealing for cryptomining -- stealing other people's available CPU power -- there is new reason and pressure to get malicious code running persistently in unwitting peoples' devices.

And... to that end: "Zealot" Campaign Uses NSA Exploits to Mine Monero on Windows and Linux Servers
https://www.bleepingcomputer.com/news/security/-zealot-campaign-uses-nsa-exploits-to-mine-monero-on-windows-and-linux-servers/

Researchers at F5 Networks have found evidence of an aggressive and sophisticated malware campaign which is currently underway, targeting Linux and Windows servers with an assortment of exploits whose common goal is installing Monero cryptocurrency mining malware. The malware was dubbed "Zealot" after one of the files (zealot.zip) that is dropped onto the target servers.

The F5 guys noted that the attackers may be fans of StarCraft since many of the names found in the code and files of the campaign are taken from the game, including Zealot, Observer, Overlord, Raven, and others.

The attackers are scanning the Internet for servers vulnerable to the Apache Struts (CVE-2017-5638) flaw which was used in the Equifax breach and also for the DotNetNuke ASP.NET Content Management System (CVE-2017-9822) vulnerability in Windows servers, in order to obtain a foothold on those unpatched machines.

And, when they are able to get inside a Windows server, their code also leverages the EternalBlue and EternalSynergy NSA exploits to allow other systems within the server's internal network to also be targeted for infection.

Finally... Windows PowerShell is used to download and install the final-stage malware, which, for this campaign, is a Monero cryptocurrency miner.

And over on the Linux side, if the Apache Struts flaw allows them in, the attackers use Python scripts, apparently lifted from the EmpireProject post-exploitation framework, to also install the same Monero miner.

This server compromise is a bit clever in that Internet-facing servers can not only be readily scanned for, located and probed... but they also tend to have more RAM and much beefier processors, enabling more competitive and effective cryptocurrency mining.

Two Critical 0-Day Remote Exploits for vBulletin Forum Disclosed Publicly
https://thehackernews.com/2017/12/vbulletin-forum-hacking.html

Web forum systems have long been difficult to secure. Since they inherently allow remote users to submit posts that are then stored, parsed by the server and displayed on everyone else's browsers, there has always been ample opportunity for miscreants to discover clever ways to compromise those systems. (This is why GRC's forthcoming web forums were built on a physically separate machine with its own firewall and independent network to the Internet. I cannot risk anything getting loose from GRC's forum software and into the rest of GRC's internal network. So there is full physical and network isolation in place.)

The situation on the online forum front have become much better over time, so that major flaws are becoming more rare as modern forum software has become far more careful about sanitizing what it accepts from unknown posters. But as we know, mistakes still happen, and vBulletin, one of the granddaddies of the forum systems, has recently had a pair of 0-Day vulnerabilities disclosed publicly.

The public 0-Day disclosures were deliberately made by researchers out of frustration after the vBulletin maintainers failed to acknowledge any communications for nearly a month. So this means that a huge number of vBulletin sites are CURRENTLY vulnerable and, as we know, that even after vBulletin is updated, many sites are likely to remain vulnerable.

The first vulnerability discovered in vBulletin is a file inclusion issue that enables remote code execution. A remote attacker is able to include any file from the vBulletin server and execute arbitrary PHP code against any file installed on Windows OS. The disclosure includes working Proof-of-Concept (PoC) exploit code to show the exploitation of the vulnerability. A Common Vulnerabilities and Exposures (CVE) number has not been assigned to this particular vulnerability.

The second vulnerability has been assigned CVE-2017-17672 and described as a deserialization issue that an unauthenticated attacker can exploit to delete arbitrary files and even execute malicious code "under certain circumstances." The vulnerability arises from the unsafe usage of PHP's unserialize() function on user-supplied input, which allows an unauthenticated hacker to delete arbitrary files and possibly execute arbitrary code on a vBulletin installation. And here, too, the advisory includes Proof-of-Concept (PoC) exploit code to demonstrate the severity of this vulnerability.

Not only is server compromise immediately possible, but anything else running on the same system might also be at risk and all visitors to and user of the affected forums might become victims as well.

Mozilla Angers Firefox Users After Force-Installing Mr. Robot Promo Add-On
https://www.bleepingcomputer.com/news/software/mozilla-angers-firefox-users-after-force-installing-mr-robot-promo-add-on/

Mozilla and Firefox stumbled into the doghouse last week when they used their "Firefox Studies" system, which is present in all recent versions of Firefox and is enabled by default, to download and install an unwanted -- apparently promotional -- browser add-on without its users' knowledge of consent. And, bizarrely, it was apparently tie-in with the finale of the 3rd season of Mr. Robot.

The "Firefox Studies" facility allows the Mozilla Firefox developers to update their users' browsers with additional code for whatever purpose they have. In this case the code took the form of an add-on named "Looking Glass v1.0.3." Firefox Studies CAN be disabled by opening settings under Privacy and Security and turning off "Allow Firefox to install and run studies."

What was this about?
https://support.mozilla.org/en-US/kb/lookingglass

> "Through the Looking Glass"
>
> What's happening?
>
> Are you a fan of Mr Robot? Are you trying to solve one of the many puzzles that the Mr

> Robot team has built? You’re on the right track. Firefox and Mr Robot have collaborated on a shared experience to further your immersion into the Mr Robot universe, also known as an Alternate Reality Game (ARG). The effects you’re seeing are a part of this shared experience.
>
> No changes will be made to Firefox unless you have opted in to this Alternate Reality Game.
>
> How do I opt in or out?
>
> To participate, install the Looking Glass add-on from https://addons.mozilla.org/firefox/addon/looking-glass/. This add-on is available only in the U.S. in English.
>
> If you no longer wish to participate in this shared world experience, enter about:addons into your address bar and remove Looking Glass.
>
> Looking Glass was previously delivered as a Shield study, so you might see "looking-glass-2" and "pug-experience" in your past studies in about:studies. It has already been removed as a study and moved to an add-on so you do not need to take any further action.

Win10 bundles an insecure password manager?
https://bugs.chromium.org/p/project-zero/issues/detail?id=1481&desc=3

For 8 days Windows bundled a password manager with a critical plugin flaw
https://arstechnica.com/information-technology/2017/12/microsoft-is-forcing-users-to-install-a-critically-flawed-password-manager/

Tavis <quote>: Title: "keeper: privileged UI injected into pages (again)"

I recently created a fresh Windows 10 VM with a pristine image from MSDN, and found that a password manager called "Keeper" is now installed by default. I'm not the only person who has noticed this:

https://www.reddit.com/r/Windows10/comments/6dpj78/keeper_password_manager_comes_preinstalled_now/

I assume this is some bundling deal with Microsoft. I've heard of Keeper, I remember filing a bug a while ago about how they were injecting privileged UI into pages (issue 917). I checked and, they're doing the same thing again with this version. I think I'm being generous considering this a new issue that qualifies for a ninety day disclosure, as the same attack [still] works.

Nevertheless, this is a complete compromise of Keeper security, allowing any website to steal any password. Here is a working demo that steals your twitter password:

https://lock.cmpxchg8b.com/keepertest.html

The tech press coverage of this had headlines like "For 8 days Windows bundled a password manager with a critical plugin flaw." ... which in this case appears not to be overblown.

Unlike other stuff bundled with Win10, users do not get a link to click and install. Keeper comes pre-installed.

The Keeper folks responded this time to Tavis, and had a patch available and pushed out within 24 hours. So that's good.

https://blog.keepersecurity.com/2017/12/15/update-for-keeper-browser-extension-v11-4/

All customers running Keeper’s browser extension on Edge, Chrome and Firefox have already received Version 11.4.4 (or newer version) through their respective web browser extension update process. Customers using the Safari extension can manually update to version 11.4.4 (or newer) by visiting Keeper’s download page.

aPAColypse now: Exploiting Windows 10 in a Local Network with WPAD/PAC & JScript
https://googleprojectzero.blogspot.com/2017/12/apacolypse-now-exploiting-windows-10-in_18.html

Google's Project Zero

Many widely-deployed technologies, viewed through 20/20 hindsight, seem like an odd or unnecessarily risky idea. Engineering decisions in IT are often made with imperfect information and under time pressure, and some oddities of the IT stack can best be explained with “it seemed like a good idea at the time”. In the personal view of some of the authors of this post, WPAD (“Web Proxy Auto Discovery Protocol” - and more specifically “Proxy Auto-Config”), is one of these oddities.

At some point in the very early days of the Internet - prior to 1996 - engineers at Netscape decided that JavaScript was a good language to write configuration files in. The result was PAC - a configuration file format that works as follows: The browser connects to a pre-configured server, downloads the PAC file, and executes a particular Javascript function to determine proper proxy configuration. Why not? It certainly is more expressive and less verbose than (let’s say) XML, and seems a reasonable way to provide configurations to many clients.

PAC itself was coupled with a protocol called WPAD - a protocol that makes it unnecessary for the browser to have a pre-configured server to connect to. Instead, WPAD allows the computer to query the local network to determine the server from which to load the PAC file.

Somehow this technology ended up being an IETF draft which expired in 1999, and now, in 2017, every Windows machine will ask the local network: “Hey, where can I find a Javascript file to execute?”. This can happen via a number of mechanisms: DNS, WINS, but - perhaps most interestingly - DHCP.

PAC = Proxy Auto-Config

WPAD protocol = Query DHCP and then DNS and others.

The returned argument is an http://.... file containing JavaScript that the Windows old JScript engine interprets.

Multiple vulnerabilities were found and a full highly-reliable PoC exploit was developed to run CMD with full SYSTEM privileges. That would allow other remote code to be retrieved and run within the caller's SYSTEM privileges.

How a Dorm Room Minecraft Scam Brought Down the Internet
https://www.wired.com/story/mirai-botnet-minecraft-scam-brought-down-the-internet/
https://www.techdirt.com/articles/20171214/18385638815/how-minecraft-led-to-mirai-botnet.shtml

We now know the whole backstory behind Mirai, the most sophisticated botnet to date which was responsible for originating the many largest, most potent and devastating DDoS bandwidth attacks ever seen:

It was three MineCraft playing college kids running a profitable MineCraft server who wanted to blast competing MineCraft hosting servers off the Net in order to obtain their competitors' user bases.

They just pleaded guilty in an Alaskan court.

First servers, then DDoS mitigation services.

Then posting the Mirai source to deflect attention.

But the FBI had already zeroed-in on the leader.

The publication of the full source code was further devastating since then others snatched it up and used it.

Filed under "That's not a bug... it's a feature!": Anyone can steal all of chrome saved passwords, form fields, bookmarks, history
https://medium.com/@liormarga/anyone-can-steal-all-of-chrome-saved-passwords-form-fields-bookmarks-history-ab2da3b4853e

First of all... we know it's possible to simply view the passwords saved by Chrome without providing any authentication.

Lior Margalit writing in Medium:

<quote> "Anyone can steal all of chrome saved passwords, form fields, bookmarks, history"

"You can try it with your friends at work or with anyone [who] gives you access to a computer… it's really funny but dangerous.
I reported this issue to Google and their response was: "Yes, given unrestricted access to a user's account, you can steal data from it … Status:WontFix"

That's true but still it's really easy so i'll show you how

1. Goto chrome://settings/manageProfile

2. click on the Edit person or chrome://settings/people

3. Sign Out.

4. Click "Sign in to Chrome." ... and use another (the attacker's) Gmail account.

5. Chrome helpfully notes that another user (provides their gmail account) was previously using Chrome and you choose between "This Wasn't Me" or "This Was Me" Choose "This Was Me."

"This Was Me" says: Add my Bookmarks, History, Passwords, and other settings to _____ attacker's gmail account.

So Click continue.

Lior writes: "BOOM you just stole chrome all saved passwords, form fields, bookmarks, history without knowing their password.

open any other computer

Sign in with your gmail account
browse to chrome://settings/?search=password

All of that user's passwords are now captured and available.

Errata

Drew Green @agreenbhm
@SGgrc been listening to previous episodes and heard you and Leo discussing Google's Advanced Protection Program. A Bluetooth 2FA device isn't necessary; I found it works with an NFC Yubikey in place of a Bluetooth device, with the other device being a different USB-only Yubikey

Miscellany

"Be a Coder" Humble Bundle
https://www.humblebundle.com/books/be-a-coder-books
13 days remaining

zerohedge @zerohedge (via Simon Zerafa)
The @FBI sold the 144,336 bitcoins its confiscated from Silk Road for $48 million. They would now be worth $2.4 billion

Tavis Ormandy @taviso
Everyone wants there to be simple answers in security, but sometimes there are no simple answers.

Apple makes iOS 11.2.1 and tvOS 11.2.1 available, fixing HomeKit vulnerability and restoring remote access for shared users in Homekit (@apollozac / 9to5Mac)

A Bitcoin transaction WOULD be a capital gain.... so only 15% since held for many years.

SpinRite

Philipp / @pimiddy
@SGgrc #spinrite saves the day! SSD drive was failing. Had everything backed up...except my private GPG key. Damn! A level 2 on the drive took 26 hours, but the drive was mountable again and I rescued the key. Thanks for a great product!

Grégory Paul / @paulgreg
My 2 USB keys booting my freenas NAS failed both around the same time. Thanks to @SGgrc for SpinRite, which bring them back online the time to migrate from USB keys to a SSD ! #freenas #spinrite

Closing The Loop

Gerry Martinez / @gerrymartinez
@SGgrc Hi Steve, is it normal for a VPN to dramatically slow your connection? I decided to purchase TunnelBear & my connection goes from ~170Mbps to less than 5Mbps. I'm connecting to US sites & my ISP is Cox.

James Parsons / @jparsons_net
@SGgrc The new #FirefoxQuantum is amazing! The old Firefox was getting so bad I was close to reluctantly switching browsers. I still prefer the look of the old UI, though.

Steve Whisenhant / @stevewhisenhant
.@SGgrc regarding the question on SN641 about using user's name as password salt. I consider it best practice (and sometimes mandated by regulation) to encrypt PII such as customer name while at rest. This would either prevent or unacceptably complicate password validation IMO.

Good point!: So let's hash the username and use that as the salt. :)

John Helt / @FunWithACamera
@SGgrc In reference to this week's SecurityNow podcast, using the UID as the salt for a password hash gives an attacker who downloads the file with UID and hashed passwords an advantage when attempting to brute force cleartext passwords. Random salt stored separately seems better

James Parsons / @jparsons_net
@SGgrc Disagree with you about using a user's name as the salt. If multiple sites follow the same practice, it could indicate that the user reuses the same password on multiple sites.

BGP

BGP Mon purchased by OpenDNS, which was purchased by Cisco
https://bgpmon.net/popular-destinations-rerouted-to-russia/

Popular Destinations rerouted to Russia
Posted by Andree Toonk - December 12, 2017
Early this morning (UTC) our systems detected a suspicious event where many prefixes for high profile destinations were being announced by an unused Russian Autonomous System.

Starting at 04:43 (UTC) 80 prefixes normally announced by organizations such Google, Apple, Facebook, Microsoft, Twitch, NTT Communications and Riot Games were now detected in the global BGP routing tables with an Origin AS of 39523 (DV-LINK-AS), out of Russia.

Looking at timeline we can see two event windows of about three minutes each. The first one started at 04:43 UTC and ended at around 04:46 UTC. The second event started 07:07 UTC and finished at 07:10 UTC.
Even though these events were relatively short lived, they were significant because it was picked up by a large number of peers and because of several new more specific prefixes that are not normally seen on the Internet. So let's dig a little deeper.

One of the interesting things about this incident is the prefixes that were affected are all network prefixes for well known and high traffic internet organizations. The other odd thing is that the Origin AS 39523 (DV-LINK-AS) hasn't been seen announcing any prefixes for many years (with one exception below), so why does it all of sudden appear and announce prefixes for networks such as Google?

Sunday, February 24, 2008:
For two hours viewers across most of the world were unable to reach YouTube. The root cause of this outage was a flaw in the Border Gateway Protocol (BGP), which governs how routing data propagates across the Internet. It began the previous Friday when the Pakistan Telecommunications Authority ordered the nation's Internet service providers to black out a YouTube video it feared would trigger riots. Pakistan Telecommunication Company Ltd. (PTCL), Pakistan's largest Internet provider, responded by blocking the entire YouTube site through a quirk in BGP that's readily exploitable by anyone who controls a BGP-enabled router (border router) of an autonomous system (AS) such as PTCL. PTCL connects to the Internet solely through another AS, PCCW Ltd., a Hong Kong telecommunications company. PTCL technicians failed to warn PCCW to block the bogus route. Consequently, it spread through PCCW into the rest of the Internet.

In the summer of 2014:
Attackers cleverly misused the Bitcoin stratum protocol. By hijacking IP addresses of the pool server IP addresses, the attacker stole 83,000 US dollars worth of Bitcoins. This was an advanced attack, not just from the Bitcoin perspective but also from the BGP perspective. In an attempt to hide the attack, the originator used AS path prepending with a range of Autonomous systems.

In September of 2014:
A number of spamming campaigns relied on IP squatting to send spam. IP squatting is a technique where folks find un-announced address space, announce that IP space for a brief period, effectively stealing those IP addresses from the rightful owner and then use it, for example to send spam. In examining these events, each time a small group of prefixes was announced for a few hours only to quickly be withdrawn and a new group of prefixes announced again. This method was used by spammers because it solves the problem of IP reputation (spam) list. By the time these addresses are added to the various lists, the spammer has moved on to other addresses from a previously unused ranged. Tracking the actual origin of the spammers is also harder as whois data will only show who owns the space, not necessarily who announced it.

Routing errors can be hard to detect because Internet connectivity changes often. According to a Sprint technical report, "there is continuous BGP 'noise' (around 50–200 updates/minute) interspersed with high 'churn' periods (9000 updates/minute)." And it's difficult to determine whether a routing change is legitimate. Most ASes try to prefer a route provided by an AS for addresses within itself, versus a route advertised by an outside source.

However, BGP makes it difficult to detect routes created by outsiders, and does nothing to prevent their creation, despite the likelihood that these routes will be bogus.

And the Internet's growth aggravates these problems. For example, from 1997 through 2005, routing tables grew from 3,000 to more than 17,000 ASes, and from 50,000 to more than 180,000 routing prefixes.

The sparsity of IPv4 addresses has further fractured and exploded routing tables.

Routing Tables.