Security Now! #674 - 07-31-18

Attacking Bluetooth Pairing

This week on Security Now!

This week we examine still another new Spectre processor speculation attack, we look at the new "Death Botnet", the security of the US DoD websites, lots of Google Chrome news, pushes by the US Senate toward more security, the emergence and threat of clone websites in other TLDs, more cryptocurrency mining bans, Google's Titan hardware security dongles, and we finish by examining the recently discovered flaw in the Bluetooth protocol which has device manufacturers and OS makers scrambling. (But do they really need to?)

Security News

NetSpectre: Read Arbitrary Memory over Network
https://misc0110.net/web/files/netspectre.pdf

"Throwhammer: Rowhammer Attacks over the Network and Defenses"
Authors: VU Amsterdam - Herbert Bos and his team.

Using high-speed network NIC DMA to pound on DRAM to induce selective bit-flips.

Now we have true Spectre Variant 1 exploitation over network connections.

ABSTRACT:
Speculative execution is a crucial cornerstone to the performance of modern processors. During speculative execution, the processor may perform operations the program usually would not perform. While the architectural effects and results of such operations are discarded if the speculative execution is aborted, microarchitectural side effects may remain. The recently published Spectre attacks exploit these side effects to read memory contents of other programs. However, Spectre attacks require some form of local code execution on the target system. Hence, systems where an attacker cannot run any code at all were, until now, thought to be safe.

In this paper, we present NetSpectre, a generic remote Spectre variant 1 attack. For this purpose, we demonstrate the first accessdriven remote Evict+Reload cache attack over network, leaking 15 bits per hour. Beyond retrofitting existing attacks to a network scenario, we also demonstrate the first Spectre attack which does not use a cache covert channel. Instead, we present a novel highperformance AVX-based covert channel that we use in our cachefree Spectre attack. We show that in particular remote Spectre attacks perform significantly better with the AVX-based covert channel, leaking 60 bits per hour from the target system.We verified that our NetSpectre attacks work in local-area networks as well as between virtual machines in the Google cloud.

NetSpectre marks a paradigm shift from local attacks, to remote attacks, exposing a much wider range and larger number of devices to Spectre attacks. Spectre attacks now must also be considered on devices which do not run any potentially attacker-controlled code at all. We show that especially in this remote scenario, attacks based on weaker gadgets which do not leak actual data, are still very powerful to break address-space layout randomization remotely. Several of the Spectre gadgets we discuss are more versatile than anticipated. In particular, value-thresholding is a technique we devise, which leaks a secret value without the typical bit selection mechanisms. We outline challenges for future research on Spectre attacks and Spectre mitigations.

////////

Spectre attacks have so far been demonstrated in JavaScript and in native code, but it is likely that any environment allowing sufficiently accurate timing measurements and some form of code execution enables these attacks. Attacks on Intel SGX (software guard extensions) enclaves

showed that enclaves are also vulnerable to Spectre attacks. However, there are billions of devices which never
run any attacker-controlled code, i.e., no JavaScript, no native code, and no other form of code execution on the target system. Until now, these systems were believed to be safe against such attacks. In fact, vendors are convinced that these systems are still safe and recommended to not take any action on these devices.

In this paper, we present NetSpectre, a new attack based on Spectre variant 1, requiring no attacker-controlled code on the target device, thus affecting billions of devices. Similar to a local Spectre attack, our remote attack requires the presence of a Spectre gadget in the code of the target. We show that systems containing the required Spectre gadgets in an exposed network interface or API can be attacked with our generic remote Spectre attack, allowing to read arbitrary memory over the network. The attacker only sends a series of crafted requests to the victim and measures the response time to leak a secret value from the victim's memory.

We show that memory access latency, in general, can be reflected in the latency of network requests. Hence, we demonstrate that it is possible for an attacker to distinguish cache hits and misses on specific cache lines remotely, by measuring and averaging over a larger number of measurements.

////////

They develop a powerful and effective remotely exploitable ASLR bypass.

////////

To complete the attack, the attacker measures the response time for every secret bit to leak. As the difference in the response time is in the range of nanoseconds, the attacker needs to average over a large number of measurements to obtain the secret value with acceptable confidence. Indeed, our experiments show that the difference in the microarchitectural state becomes visible when performing a large number of measurements. Hence, an attacker can first measure the two corner cases (i.e., cached and uncached) and afterward, to extract a real secret bit, perform as many measurements as necessary to distinguish which case it is with sufficient confidence, e.g., using a threshold or a Bayes classifier.

////////

Attacks on the kernel are pretty much a given since all moden OSes have their network drivers in the kernel for performance.

////////

Leakage:
To evaluate NetSpectre on the different devices, we constructed a victim program which contains the same leak gadget and transmit gadget on all test platforms (cf. Section 3). We leaked known values from the victim to verify that our attack was successful and to determine how many measurements are necessary. Except for the cloud setup, all evaluations were done in a local lab environment. We used Spectre variant 1 for all evaluations, however, other Spectre variants can

be used in the same manner.

6.1.1 Desktop and Laptop Computers. In contrast to local Spectre attacks, where a single measurement can already be sufficient, NetSpectre attacks require a large number of measurements to distinguish bits with a certain confidence. Even on a local network, around 100 000 measurements are required to reduce the noise to a level where a difference between bits can be clearly seen. By repeating the attack, the noise is reduced, making it easier to distinguish the bits.

For our local attack we had a gigabit connection between the victim and the attacker, a typical scenario in local networks but also for network connections of dedicated servers and virtual servers. We measured a standard deviation of the network latency of 15.6 ìs.

////////

Performance:
6.2.1 Local Network. Attacks on the local network achieve the best performance, as the variance in network latency is significantly smaller than the variance over the internet (cf. Section 6.1.3). In our lab setup, we repeat the measurement 1 000 000 times per bit to be able to reliably leak bytes from the victim. On average, leaking one byte takes 30 min, which amounts to approximately 4 min per bit. Using the AVX covert channel instead of the cache reduces the required time to leak an entire byte to only 8 min.

To break ASLR, we require the cache covert channel. On average, this allows breaking the randomization remotely within 2 h.

6.2.2 Cloud Network. We evaluated the performance in the cloud using two virtual machines instances on the Google Cloud. These virtual machines have a fast network connection. We configured the two instances to each use 2 virtual CPUs, which enabled a 4 Gbit/s connection [23]. In this setup, we repeat the measurement 20 000 000 times per bit to get an error-free leakage of bytes. On average, leaking one byte takes 8 h for the cache covert channel, and 3 h for the AVX covert channel.

While this is comparably slow, it shows that remote Spectre attacks are feasible between independent instances in the public cloud. In particular, APTs typically run for several weeks or months. Such an extended time frame is clearly sufficient to leak sensitive data, such as encryption keys or passwords, using the NetSpectre attack in a cloud environment.

///////

So what does that all mean?

On this one, the popular press coverage mostly got this wrong. Not because they were overhyping it for click bait, but because they misunderstood it and were underhyping it. They believed that the slow information exfiltration speed made NetSpectre of little use to attackers. But in a network attack scenario, obtaining a 256-bit secret key on a local network at the rate of 4 minutes per bit, or 1024 minutes or 17 hours is a bargain for any attacker who can get into one machine inside an enterprise and then begin sucking private keys out of the enterprise's servers.

When mitigating the Spectre threat there has been a "fix it where we must" approach since fixing it is generally a performance-killing pain. What this NetSpectre attack means is that the universe of places that Spectre-based information leakage must be fixed -- for this class of slow-dribble attack -- just got must larger.

Here comes the "Death" Botnet

Last week we covered the work of a security researcher, Ankit Anabhav, with NewSky Security, tracking the sudden emergence of a "flash Botnet" consisting of 18,000 Huawei routers within 24 hours... solely by leveraging a long-known vulnerability which had gone unpatched.

Now Ankit has uncovered the emergence of the "Death Botnet" which is being built upon a 2-year old and also long since patched AVTech device flaw.

The firmware in several AVTech-based devices including DVRs, NVRs, IP cameras, and more, both exposes the device's passwords in plaintext, but also permits an unauthenticated attacker to add users to existing devices.

The hacker, going by the handle "EliteLands" is adding new users to AVTech devices, but using shell commands as the passwords for these accounts. Why??

Ankit tweets:
This exploit is hilarious. In very simple words, if you create a new user with any:

username "bleepingcomputer"
password: "reboot"

The device will really reboot ! Of course the threat actor is using this to run a botnet, not simply reboot it. #iot #exploit #Hacking

The problem here is that these older AVTech devices are also vulnerable to a command injection flaw that makes the device read the password as a shell command, allowing the hacker to take over these devices.

Ankit told BleepingComputer: "If I put reboot as password, the AVTech system gets rebooted. Of course, the Death botnet is doing much more than just rebooting."

Ankit says the hacker has been experimenting with different payloads for use in the password field and has recently started using these payloads for building a botnet to which he refers to as Death.

In the latest version of this payload, Anubhav says EliteLands is adding accounts with a lifespan of only five minutes that execute his payload and then disapear from the infected device.

"This is like a burner account," Ankit said, "Usually people don't make new user accounts with access of only 5 minutes."

The size of this botnet is unknown, but the way it was built required minimal effort. Ankit says he identified over 1,200 AVTech devices that can be hijacked in this way using an IoT search engine.

AVTech has patched the issues exploited by this Death Botnet more than a year and a half ago, back at the beginning of 2017... but we know that most devices will never be updated.

https://www.bleepingcomputer.com/news/security/malware-author-building-death-botnet-using-old-avtech-flaw/

The US DOD and HTTPS: "Moving at the speed of government"
Two months ago:
https://assets.documentcloud.org/documents/4620887/Wyden-DOD-Letter-HTTPS.pdf

The DOD finally responded to Ron Wyden's letter stating that "we've been working on it for two years and we plan to have this done by the end of the year."

Chrome & HTTPS:
Meanwhile, Chrome 68 (and beyond) executes on its intention of flagging non HTTPS sites as "Not Secure"

According to Cloudflare's telemetry, 542,605 from the top 1 million sites do not use, or do not redirect users to, an HTTPS version of the HTTP URL. This means that a large number of users will probably see a "Not Secure" indicator next to most of the sites they visit after they update to Chrome 68.

Our browsers really should start attempting an HTTPS connection when not otherwise forced to HTTP.

"Not Secure" http://www.foxnews.com/
https://www.foxnews.com/
But then, even so: Your connection to this site is not fully secure. You should not enter any sensitive information on this site (for example, passwords or credit cards), because it could be stolen by attackers.

Perfectly valid (non-EV) certificate issued by DigiCert.

<GOOGLE> Hi All -- The reason why https://www.foxnews.com is getting marked "not fully secure" is because it includes a search form that sends data to http://www.foxnews.com (note the insecure "HTTP"). We currently count that as "not fully secure" and downgrade the security indicator icon to the "i" symbol rather than the lock. A secure HTTPS site should load all resources over HTTPS and have all forms submit to HTTPS. If you have any questions about this policy or our current implementation of it, feel free to follow up in this thread.

Even if some people are squawking, this transition cannot have caught anyone by surprise. Google deliberately signaled its plans well in advance to give website admins ample time to migrate their sites to fully secure connections.

And we also know that no one changes anything unless they are forced to. Having sites marked as "Not Secure" ought to finally provide the required motivation.

But wait... there's more! Other cool new features in Chrome 68:
https://blog.chromium.org/2018/06/chrome-68-beta-add-to-home-screen.html

Many other new feature improvements as Chrome continues moving forward.

Support for the Web payments API for unifying the online purchase experience.

Many other web-developer improvements.

For me... Still no strict site isolation under Chrome 68.
chrome://flags/   and search for "isol"

And speaking of Ron Wyden...
Tech Savvy Senator Ron Wyden asks US Government to Remove Flash From Federal Sites and Computers
https://www.bleepingcomputer.com/news/government/senator-asks-us-government-to-remove-flash-from-federal-sites-computers/

Last week, US Senator Ron Wyden sent a letter to the US National Institute of Standards and Technology (NIST), the National Security Agency (NSA), and the Department of Homeland Security (DHS). That letter asked those government officials to find solutions and procedures to mandate the removal of Adobe Flash content from all US government websites by August 1, 2019.

As we know, Adobe has formally announced FLASH's end-of-life at the end of 2020, after which Adobe has said it would cease to provide any technical support for the software. (God help us!)

In his letter, Wyden wrote: "The federal government has too often failed to transition away from software that has been decommissioned." Wyden points out Flash's "serious, largely unfixable cybersecurity issues" as one of the reasons US government sites should take proactive steps to permanently remove this technology from all of its sites, well before the cut-off date, after which Adobe won't provide any security fixes at all.

Wyden asked officials to:

- Mandate that government agencies not deploy new Flash-based content on any federal website, effective within 60 days.
- Require federal agencies to remove all Flash-based content from their websites by August 1, 2019.
- Require agencies to remove Flash from desktop computers by August 1, 2019.

And since HTML5 is able to do everything FLASH can, it is in serious decline:

In February, Google's Director of Engineering said that the percentage of daily Chrome users who've loaded at least one page containing Flash content per day has gone down from around 80% in 2014 to under 8% by early this year.

And according to web technology survey site W3Techs, only 4.9% of today's websites utilize Flash code, a number that has plummeted from a 28.5% market share recorded at the start of 2011.

Firefox is also moving toward strict site isolation
https://www.bleepingcomputer.com/news/software/mozilla-is-working-on-a-chrome-like-site-isolation-feature-for-firefox/
https://wiki.mozilla.org/Project_Fission

Remember: Site isolation provides much more strict inter-domain isolation by sandboxing all page rendering for a single domain into its own process.

Google believes that the isolation will be effective enough that it will be able to relax or remove its current Spectre mitigations.

An important reminder about spoofed websites:
https://www.bleepingcomputer.com/news/security/fake-websites-for-keepass-7zip-audacity-others-found-pushing-adware/

French and Spanish language versions of the legitimate sites:

keepass.fr
7zip.fr
inkscape.fr
gparted.fr
clonezilla.fr
paintnet.fr
greenshot.fr
scribus.fr
audacity.es
stellarium.fr
celestia.fr
celestia.es
azureus.es
clonezilla.es
inkscape.es
paintnet.es
handbrake.es
gimp.es
thunderbird.es

unetbootin.org
unetbootin.net
notepad2.com
audacity.fr
filezilla.fr
truecrypt.fr
blender3d.fr
grooveshark.fr
adblock.fr

Can be HTTPS & TLS and "secure"

They have been pushing legitimate versions of the applications... but with added AdWare, for which the site receives payment for every copy installed.

Google formally bans cryptocurrency mining from the Play Store

Google has updated the Play Store policy page this week to ban apps that mine cryptocurrencies on users' devices.

New Section of the "Developer Policy Center":

> Financial Instruments
>
> We don't allow apps that expose users to deceptive or harmful financial instruments.
> Binary Options
>
> We do not allow apps that provide users with the ability to trade binary options.
> Cryptocurrencies
>
> We don't allow apps that mine cryptocurrency on devices. We permit apps that remotely manage the mining of cryptocurrency.

Previous:
https://web.archive.org/web/20180102092726/https://play.google.com/about/developer-content-policy-print/

Updated: https://play.google.com/about/developer-content-policy-print/

Google will begin removing any app from the official Play Store that uses a device's CPU or GPU for cryptocurrency mining operations. Mining apps which are used to control cryptocurrency mining operations on remote devices will still be allowed.

Before this, voluntary mining apps were permitted when the user clearly gave permission.

Cryptocurrency mining apps HAVE damaged devices. Smartphones have been overheated and batteries have been destroyed... being caused to deforms, leaks, or catch fire.

This policy change was not publicly pre-announced, so developers who had apps removed were complaining on Reddit. Other policy changes were also made, such as the decision to ban apps with repetitive content.

Apple similarly banned all mining from the iOS App Store last month.

So, mobile mining is a thing of the past. And just as well. Smartphones were not designed for that sort of extended power consumption.

Also See:
https://thehackernews.com/2018/07/android-cryptocurrency-mining.html
https://www.bleepingcomputer.com/news/google/google-bans-cryptocurrency-mining-apps-from-the-play-store/

And speaking of Cryptocurrency mining... another supply-chain attack:
We're beginning to encounter more supply-chain attacks.

Notably, the popular CCleaner site suffered a supply-chain attack.

https://www.bleepingcomputer.com/news/security/microsoft-discovers-supply-chain-attack-at-unnamed-maker-of-pdf-software/

https://cloudblogs.microsoft.com/microsoftsecure/2018/07/26/attack-inception-compromised-supply-chain-within-a-supply-chain-poses-new-risks/

BleepingComputer:

Microsoft said today that hackers compromised a font package installed by a PDF editor app and used it to deploy a cryptocurrency miner on users' computers.

The OS maker discovered the incident after its staff received alerts via the Windows Defender ATP, the commercial version of the Windows Defender antivirus.

Microsoft employees say they investigated the alerts and determined that hackers breached the cloud server infrastructure of a software company providing font packages as MSI files. These MSI files were offered to other software companies.

One of these downstream companies was using these font packages for its PDF editor app, which would download the MSI files from the original company's cloud servers during the editor's installation routine.

But... Hackers created a copy of the company's cloud servers

Microsoft's security researchers said that "Attackers recreated the [first company's] infrastructure on a replica server that the attackers owned and controlled. They copied and hosted all MSI files, including font packages, all clean and digitally signed, in the replica server."

Then "The attackers decompiled and modified one MSI file, an Asian fonts pack, to add the malicious payload with the coin mining code."

"Using an unspecified weakness (which does not appear to be MITM or DNS hijack), the attackers were able to influence the download parameters used by the [PDF editor] app. The parameters included a new download link that pointed to the attacker server."

Users who downloaded and ran the PDF editor app would unknowingly install the font packages, including the malicious one, from the hackers' cloned server. Because the PDF editor app was installed under SYSTEM privileges, the malicious coinminer code hidden inside would receive full access to a user's system.

The malicious miner would create its own process named xbox-service.exe under which it would mine for cryptocurrencies using victims' computers.

Microsoft said Windows Defender ATP detected mining-specific behavior from this process. Investigators then tracked down the origin of this process to the PDF editor app installer and the MSI font packages.

Security researchers said it was easy to identify which MSI font package was the malicious one because all other MSI files were signed by the original software company, except one file, which lost its authenticity when crooks injected the coinminer code inside it.

This malicious miner also stood out to investigators because it also tried to modify the Windows hosts file in a poor attempt at sinkholing update operations for various security apps. Tinkering with the Windows hosts file is a big no-no, and most antivirus software will mark this operation as suspicious or malicious.

Microsoft did not reveal the names of the two software companies involved in this incident. The OS maker says the compromise lasted between January and March 2018, and affected only a small number of users, suggesting the hacked companies aren't big names on the PDF software market.

Titan Security Keys – Google launches its own USB-based FIDO U2F Keys
https://thehackernews.com/2018/07/google-titan-security-key-fido.html
https://www.blog.google/products/google-cloud/building-on-our-cloud-security-leadership-to-help-keep-businesses-protected/

Titan Security Keys are based on the FIDO (Fast IDentity Online) Alliance, U2F (universal 2nd factor) protocol and incorporates a secure element and firmware developed by Google that verifies the integrity of security keys.

2nd factor vs secure One factor.

SpinRite

Joshua Montgomery in Springfield, IL
Subject: SpinRite Saved one of our ATM's
Date: 26 Jul 2018 05:43:20
:
Steve,

I have been an avid listener since the beginning of Security Now. Thanks to you and Leo for all of the hard work. I Wanted to thank you for making this splendid product. It saved my neck. We had a very high traffic ATM hard drive fail. The tech said that they could not make it for 48 hours. I pulled the hard drive from the ATM and ran SpinRite on it. It fix enough of the drive that I could clone it to an SSD within 3 hours. Now we have one the fastest ATM machines in the North. Keep up the good work!

"Breaking the Bluetooth Pairing Fixed Coordinate Invalid Curve Attack"

https://www.cs.technion.ac.il/~biham/BT/bt-fixed-coordinate-invalid-curve-attack.pdf

As covered in the pop/tech media:
https://arstechnica.com/information-technology/2018/07/decade-old-bluetooth-flaw-lets-hackers-steal-data-passing-between-devices/
https://thehackernews.com/2018/07/bluetooth-hack-vulnerability.html
https://www.bleepingcomputer.com/news/security/many-bluetooth-implementations-and-os-drivers-affected-by-crypto-bug/

https://www.bleepingcomputer.com/news/security/many-bluetooth-implementations-and-os-drivers-affected-by-crypto-bug/

Bluetooth is a widely deployed platform for wireless communications between mobile devices. It uses authenticated Elliptic Curve Diffie-Hellman for its key exchange. In this paper we show that the authentication provided by the Bluetooth pairing protocols is insufficient and does not provide the promised MitM protection. We present a new variant of an Invalid Curve Attack that preserves the x-coordinate of the public keys. The attack compromises the encryption keys of all of the current Bluetooth authenticated pairing protocols, provided both paired devices are vulnerable. Specifically, it successfully compromises the encryption keys of 50% of the Bluetooth pairing attempts, while in the other 50% the pairing of the victims is terminated. The affected vendors are currently working to patch their products.

The Elliptic Curve Diffie-Hellman (ECDH) protocol, introduced in the 1980s, is a variant of the original Diffie-Hellman key exchange protocol. It utilizes the algebraic structure of elliptic curves over finite
fields in order to exchange cryptographic symmetric keys over a public compromised channel.

An EC key pair is a private scalar value and a point (X,Y)

Secure Simple Pairing (SSP)

This confinement implies that the attack succeeds only if both private keys SKa and SKb are even. In
this case, which occurs with probability 0.25, we get DHKeya = DHKeyb = ¥.

This manipulation is not detected since the x-coordinates are left unchanged.
The semi-passive attack is thus as follows:

1. Eavesdrop both parties throughout the pairing protocol.

2. Let both parties perform the first phase of the pairing (feature exchange).

3. Let the parties transmit their ECDH public keys.

4. Modify the y-coordinate of both public keys to zero during the transmission.

5. Do not intervene with rest the of the pairing protocol.

6. Observe if the pairing succeeded, otherwise, quit.

7. Derive the symmetric session keys (LTK and MacKey) using the expected DHKey = ¥ and the public parameters.

8. After the pairing is finished, forge or passively decrypt packets sent between the participating devices using the derived keys.

Since LE SC is implemented in the host, the vulnerability is found in the host's operating system, regardless of the Bluetooth adapter. We found the Android Bluetooth stack, "Bluedroid", to be vulnerable2. On the other hand, Microsoft Windows does not support LE SC.

The vulnerability in SSP depends on the Bluetooth chip firmware implementation, since unlike LE SC, the key exchange is performed by the chip, rather than by the host. During our research we found that devices of most major chip vendors are affected. In particular, Qualcomm's, Broadcom's and Intel's implementations3 are vulnerable, which together constitutes most of the Bluetooth chips market. We stress that every device (e.g., mobile phone, laptop or car) that uses such a chip is vulnerable.