Security Now! #676 - 08-14-18

The Mega FaxSploit

This week on Security Now!

This week we cover lots of discoveries revealed during last week's Black Hat 2018 and DEF CON 26 Las Vegas security conferences. Among them, 47 vulnerabilities across 25 Android smartphones, Android "Disk-In-The-Middle" attacks, Google tracking when asked not to, more Brazilian DLink router hijack hijinks, a backdoor found in VIA C3 processors, a trusted-client attack on WhatsApp, a macOS 0-day, a tasty new feature for Win10 Enterprise, a new Signal-based secure eMail service, Facebook's FIZZ TLS v1.3 library, another Let's Encrypt milestone, and then "FaxSploit" the most significant nightmare in recent history (FAR worse, I think, than any of the theoretical Spectre & Meltdown attacks).

The dialog all HP Combo Fax/Printer users should immediately see:

NOTE:

This is the FINAL PODCAST of year 13. Our first 18-minute podcast was August 19th, 2005. Next week will be August 20th, so we will have lapped ourselves by one day and started into the 14th year of this weekly security-focused podcast.

Security News

47 vulnerabilities disclosed across 25 Android smartphones
https://www.kryptowire.com/portal/android-firmware-defcon-2018/

During last week's Las Vegas DEF CON security conference, researchers with the US mobile and IoT security firm, Kryptowire, revealed the findings from their research which was conducted as part of a drant awarded by the Department of Homeland Security (DHS).

They presented the details of 47 vulnerabilities in the firmware and default apps of 25 Android smartphone models, 11 of which have a presence in the US market.

This is one of those distrassing cases where the vulnerabilities are too numerous to tick off one by one, so I've included a link to Kryptowire's announcement:

https://www.kryptowire.com/portal/android-firmware-defcon-2018/

The vulnerabilities discovered on devices offered by the major US carriers are the following: arbitrary command execution as the system user, obtaining the modem logs and logcat logs, wiping all user data from a device (performing a factory reset), reading and modifying a user’s text messages, sending arbitrary text messages, getting the phone numbers of the user’s contacts, and more. All of these the aforementioned capabilities are obtained outside of the normal Android permission model.

Major brands here are Alcatel, Asus, LG, Nokia, Sony and ZTE.

Lesser brands: Coolpad, Doogee, Essential, Leagoo, MXQ, Oppo, Orbic, Plum, SKY and Vivo.

Note that Google is not present in this list.

What's best: Monoculture or Heterogeneous spread?

What's the real danger here?

Might you be targeted?

CheckPoint: Man-in-the-Disk Attacks
https://blog.checkpoint.com/2018/08/12/man-in-the-disk-a-new-attack-surface-for-android-apps/

Check Point's researchers examined a shortcoming in the way Android apps use storage resources. Careless use of External Storage by applications may open the door to attacks resulting in any number of undesired outcomes, such as silent installation of unrequested, potentially malicious, apps to the user's phone, denial of service for legitimate apps, and even cause applications to crash, opening the door to possible code injection that would then run in the privileged context of the attacked application.

To understand the security deficiency in Android's design, we need to look at the storage resources on an Android device.

Within the Android OS there are two types of storage: Internal Storage, which each application uses separately and is segregated by the Android Sandbox, and External Storage, often over an SD card or a logical partition within the device's storage, which is shared by all applications. The External Storage is often used to deliberately share files between applications or with a PC. So it is a sandbox bypass -- by design.

For example, for a messaging app to share a photo from the phone's gallery, the application needs to have access to media files held in the External Storage.

There are other reasons for an app developer to choose to use External Storage rather than the sandboxed Internal one. Perhaps a lack of sufficient capacity in the internal storage, backwards compatibility considerations with older devices, not wanting the app to appear to use too much space, or even just lack of concern on the developer's part.

Whatever the reason may be, when using the External Storage, certain precautions are necessary... and Google understands this, even if they do not yet enforce it: According to Google's Android documentation, application developers are advised about their use of the External Storage in their apps. These guidelines include:

- "Perform input validation when handling data from external storage"
- "Do not store executables or class files on External Storage"
- "External Storage files should be signed and cryptographically verified prior to dynamic loading"

Man-in-the-Disk attacks are made possible when applications are careless about their use of External Storage. Failing to employ security precautions leaves applications vulnerable to the risks of malicious data manipulation. However, apps from major OEMs, and even Google, do not always follow these guidelines.

During Check Point's research they found instances where an app was downloaded, updated or received data from the app provider's server, which passed through the External Storage before being sent on to the app itself. Such practice offered an opportunity for an adversary to manipulate the data held in the External Storage before the app reads it again.

Meddling with the data can be performed by a seemingly innocent application, e.g. a fake flashlight app, within which holds the attacker's exploit script. The user is persuaded by the attacker to download this innocent looking app, which in turn asks for the user's permission to access the External Storage, a common and innocent-appearing request which many apps request, and so is unlikely to raise suspicion from the user. From that point on, the attacker is able to monitor data transferred between =ANY= other app on the user's device and the External Storage, and overwrite or modify it with their own data on-the-fly.

The results of the attacks can vary, depending on the attacker's intent and expertise. Check Point's research demonstrated the ability to install an attacker's application in the background, without the user's permission. They demonstrated the ability to crash the attacked application, causing it a denial of service. Once crashed and with the app's defenses down, the attacker could then potentially carry out a code injection to hijack the permissions granted to the attacked application and escalate his own privileges to access other parts of the user's device, such as the camera, the microphone, contacts list and so forth.

Applications that were tested for this new attack surface included Google Translate, Yandex Translate, Google Voice Typing, Google Text-to-Speech, the Xiaomi (Shaou mee) Browser and various other applications. After referring to the advice given within Google's guidelines, Check Point compared the advice to what was actually the case.

The Xiaomi Browser was found to be using External Storage as a staging resource for application updates. Check Point was able to execute a successful attack by which the application's update code was replaced, resulting in the installation of an alternative, undesired application instead of the legitimate update.

In the case of Google Translate, Yandex Translate and Google Voice Typing, Check Point found that the developers failed to validate the integrity of data read from the External Storage. Check Point was therefore able to compromise certain files required by those apps, resulting in the crash of each of these applications... which, with additional work, might lead to external takeover.

After discovery and verification of these application vulnerabilities, Check Point contacted Google, Xiaomi and vendors of other vulnerable applications to update them and request their response. A fix to the applications of Google was released shortly after, additional vulnerable applications are being updated and will be disclosed once the patch is made available to their users, while Xiaomi chose not to address it at all.

Check Point summarizes the problems and what they view as shortcomings of Android as follows:

- An Android device's External Storage is a public area which can be observed or modified by any other application on the same device.
- Android does not provide built-in protections for the data held in the External Storage. It only offers developers guidelines on proper use of this resource.
- Developers anywhere are not always versed in the need for security and the potential risks, nor do they always follow guidelines.

- Some of the pre-installed and popularly used apps ignore the Android guidelines and hold sensitive data in the unprotected External Storage.
- This can lead to a Man-in-the-Disk attack, resulting in the manipulation and/or abuse of unprotected sensitive data.
- Modification to the data can lead to unwelcome results on the user's device.

Google Tracks Android, iPhone Users Even With 'Location History' Turned Off
https://thehackernews.com/2018/08/google-mobile-location-tracking.html

The Hacker News reported on some research and reporting by the Associate Press which found that disabling Google's tracking of users using its Android and iPhone apps was trickier then turning off the obvious "Location History" function.

They led with the somewhat inflammatory click-bait statement: "Google tracks you everywhere, even if you explicitly tell it not to." Unfortunately, there's some truth to that.

Here's what was written:

Every time a service like Google Maps wants to use your location, Google asks your permission to allow access to your location if you want to use it for navigating, but a new investigation shows that the company does track you anyway.

An investigation by Associated Press revealed that many Google services on Android and iPhone devices store records of your location data even when you have paused "Location History" on your mobile devices.

Disabling "Location History" in the privacy settings of Google applications should prevent Google from keeping track of your every movement, as its own support page states: "You can turn off Location History at any time. With Location History off, the places you go are no longer stored."

The AP explains: "For example, Google stores a snapshot of where you are when you merely open its Maps app. Automatic daily weather updates on Android phones pinpoint roughly where you are. And some searches that have nothing to do with location, like "chocolate chip cookies," or "kids science kits," pinpoint your precise latitude and longitude—accurate to the square foot—and save it to your Google account."

To demonstrate the threat of this Google practice, the AP created a visual map of the movements of Princeton postdoctoral researcher Gunes Acar, who carried an Android smartphone with 'Location History' switched off to prevent location data collection.

However, the researchers discovered that the map includes records of Dr. Acar's train commute on two trips to New York and visits to the High Line park, Chelsea Market, Hell's Kitchen, Central Park and Harlem.

To protect the privacy of Dr. Acar, the publication did not plot the most telling and frequent marker on the map... which includes Acar's home address.

According to the researchers, this privacy issue affects around two billion Android users and hundreds of millions of iPhone users across the world who rely on Google for maps or search.

In response to the APs investigation, Googled issued the following statement:

GOOGLE: "There are a number of different ways that Google may use location to improve people's experience, including Location History, Web, and App Activity, and through device-level Location Services. We provide clear descriptions of these tools, and robust controls so people can turn them on or off, and delete their histories at any time."

Jonathan Mayer, a Princeton researcher and former chief technologist for the FCC's enforcement bureau, argued: "If you're going to allow users to turn off something called 'Location History,' then all the places where you maintain location history should be turned off. That seems like a pretty straightforward position to have."

To stop Google from saving time-stamped location markers, users need to turn off another setting, called "Web and App Activity"—a setting which is enabled by default and stores a variety of information from Google apps and sites to your Google account.

Once disabled, it will not only stop Google from storing location markers, but also prevents the company from storing information generated by searches and other activities.

- For Any device:
  Open your web browser, go to myactivity.google.com, select "Activity Controls" from the upper left drop-down menu, and now turn off both "Web & App Activity" and "Location History."

- For Android Devices:
  Head on straight to the "Security & location" setting, scroll down to "Privacy", and tap "Location." Now you can toggle it off for the entire device. You can also use "App-level permissions" to disable access to various apps.

- For iOS Devices:
  If you use Google Maps, Go to Settings ? Privacy Location Services and adjust your location setting to 'While Using' the app. This will prevent the app from accessing your location when it is not active.

The Brazilian DLink routers are being attacked again...
https://security.radware.com/ddos-threats-attacks/threat-advisories-attack-reports/dns-hijacking-brazil-banks/

This time the exploit is DNS Hijacking.

Radware's Threat Research Center has identified a hijacking campaign aimed at Brazilian bank customers via their network routers which is attempting to obtain their banking credentials.

The research center has been tracking malicious activity targeting DLink DSL modem routers in

Brazil since June 8th. Once again leveraging old and long-since-patched exploits dating from 2015, a malicious agent is modifing the DNS server settings in the routers of Brazilian residents, redirecting all their DNS requests to a malicious DNS server. The malicious DNS server is hijacking requests for the hostname of Banco de Brasil (www.bb.com.br), redirecting visitors to a fake, cloned website hosted on the same malicious DNS server which has no connection to the legitimate Banco de Brasil website.

Another Brazilian financial institution Itau Unibanco (hostname www.itau.com.br), is also being redirected, although not backed by a cloned website at the moment. For all other DNS requests, the malicious server works as a forwarder and resolves just as an ISP DNS server would.

So ... what do we know about the effectiveness of this and other such attacks?

Unless the attackers are able to obtain valid TLS certs for the domains, it won't be possible for users to avoid seeing a warning or some sort.

If user links are for HTTP (if the banks offer services -- even redirection services -- over HTTP, then a security downgrade COULD be performed.

HSTS would prevent this.

A backdoor revealed in VIA C3 x86 Processors
https://media.defcon.org/DEF%20CON%2026/DEF%20CON%2026%20presentations/Christopher%20Domas/DEFCON-26-Christopher-Domas-GOD-MODE-%20UNLOCKED-hardware-backdoors-in-x86-CPUs.pdf

This year, in a DEF CON 2018 presentation titled "GOD MODE unlocked: Hardware backdoors in x86 CPUs" Christopher Domas, an annual contributor at DEF CON, revealed the presence of a hidden "God mode" present and undocumented in VIA C3 x86-based CPUs at least back from 2001 through 2003. This feature was present in VIA C3 Nehemiah chips, but Chris said that all other C3 chipsets are bound to feature a similar mechanism.

This so-called God mode, which Chris named the "Rosenbridge Backdoor" allows an attacker to elevate the execution level of malicious code from kernel ring 3 (user mode) to kernel ring 0 (OS kernel).

If "Rosenbridge" sounds familiar that's because the official name for a spacetime wormhole is an Einstein-Rosen bridge which suggests the possibility of short-circuiting spacetime if a non-trivial solution to Einstein's field equations hold.

Anyway...

Domas says that this backdoor mechanism is a RISC (Reduced Instruction Set Computer) co-processor that sits alongside the main C3 processor. By using a launch-instruction (.byte 0x0f, 0x3f) a register control bit can be flipped to enable this additional coprocessor, which Chris says doesn't enforce the same security protections as the main C3 chipset.

Other experts have argued that since it was finally documented in 2004 it's not really a backdoor. According to VIA's document (page 82), the hidden RISC coprocessor's purpose is to provide an "alternate instruction set" that offers hardware vendors (OEMs) more control over the CPU.

"This alternate instruction set includes an extended set of integer, MMX, floating-point, and 3DNow! instructions along with additional registers and some more powerful instruction forms over the x86 instruction architecture."

The VIA document also mentions that the additional instruction set is specifically meant for testing, debugging, or other special conditions, hence the reason it is not "documented for general usage."

But... it is present nevertheless, and too potent to be left available because any instructions sent to this additional coprocessor are run under ring 0 -- bypassing all security and privilege barriers -- rather than under the normal ring 3 level.

The good news is that this controversial "backdoor" —as Chris himself explains— "should require kernel level access to activate."

Nevertheless, Domas also points out that the Rosenbridge backdoor mechanism "has been observed to be enabled by default on some systems, allowing any unprivileged code to modify the kernel" without any prior exploitation. In these scenarios, the attacker only needs to send the specially-crafted instructions to the additional RISC processors, which will be ready to read and execute them.

Chris has published a GitHub repository containing tools to identify if VIA C3 x86 CPUs contain the Rosenbridge "backdoor" mechanism, and close it to prevent any possible intentional or accidental exploitation.

https://media.defcon.org/DEF%20CON%2026/DEF%20CON%2026%20presentations/Christopher%20Domas/DEFCON-26-Christopher-Domas-GOD-MODE-%20UNLOCKED-hardware-backdoors-in-x86-CPUs.pdf

A client-side hack of WhatsApp

- https://www.bleepingcomputer.com/news/security/whatsapp-vulnerability-allows-attackers-to-alter-messages-in-chats/
- https://thehackernews.com/2018/08/whatsapp-modify-chat-fake-news.html

Check Point researchers (who have apparently been working overtime recently) revealed that someone who is already participating in a two- or more- way WhatsApp chat can alter messages to spoof their content and sender.

The coverage of this suggested that "flaws in WhatsApp" take advantage of a loophole in WhatsApp's security protocols to change the content of the messages, allowing malicious users to create and spread misinformation or fake news from "what appear to be trusted sources."

The vulnerabilities could allow hackers to misuse the 'quote' feature in a WhatsApp group conversation to change the identity of the sender, or alter the content of someone else's reply to a group chat, or even send private messages to one of the group participants (but invisible to other members) disguised as a group message for all.

In an example, the researchers were able to change a WhatsApp chat entry that said "Great!"—sent by one member of a group—to read "I'm going to die, in a hospital right now!"

By decrypting the protocol received by someone who is IN the group, the Check Point researchers were able to develop three "attacks":

1. Changing a Correspondent's Reply To Put Words in Their Mouth
2. Change the Identity of a Sender in a Group Chat, Even If They Are Not a Member
3. Send a Private Message in a Chat Group But When The Recipient Replies, The Whole Group Sees It

When the researchers reported the flaws to the WhatsApp security team, the company argued that since these messages do not break the fundamental functionality of the end-to-end encryption, users "always have the option of blocking a sender who tries to spoof messages and they can report problematic content to us."

WhatApp replied: "These are known design trade-offs that have been previously raised in public, including by Signal in a 2014 blog post, and we do not intend to make any change to WhatsApp at this time."

So... WhatsApp users might be usefully cautioned against this possibility for abuse depending upon the nature of any large group.

macOS Zero-Day Flaw Lets Hackers Bypass Security Using Invisible Mouse-Clicks
https://thehackernews.com/2018/08/macos-mouse-click-hack.html

High Sierra is vulnerable but Mojave is not.

Patrick Wardle, an ex-NSA hacker who is not the Chief Research Officer of Digita Security, discovered and revealed a critical zero-day vulnerability in the High Sierra macOS operating system that could allow a malicious application installed in the targeted system to virtually "click" objects without any user interaction or consent.

The problem arrises from "synthetic mouse clicks" which macOS code deliberately offers as part of its accessibility features for disabled people to interact with the system interface in non-traditional ways.

But, due to its potential for abuse, Apple has added some limitations to block malware from abusing these programmed clicks.

Normal mouse click events are a "down click" event, perhaps followed by mouse move events,

and eventually terminated by an "up click" event. What Patrick accidentally discovered was that High Sierra incorrectly interprets two consecutive synthetic mouse "down" events as a legitimate click. This allows attackers to programmatically interact with security warnings which ask users to choose between "allow" or "deny" and access sensitive data or features.

Naturally, these warnings were intended to be exempt from automated clicking. But Patrick's discovery bypasses the exception.

Although he has not yet published technical details of the flaw, Patrick says the vulnerability can be exploited to dump all passwords from the keychain or load malicious kernel extensions by virtually clicking "allow" on the security prompt to gain full control of a target machine.

He said that he found this loophole accidentally when copying and pasting the code and that just two lines of code were sufficient to break this security mechanism.

And, unlike with earlier findings, Patrick did not report to Apple about his discovery but instead to publicly reveal details of the zero-day bug at last week's DefCon hacker conference.

Again, Apple's next version of macOS, Mojave, has already mitigated the threat by blocking all synthetic events.

Windows 10 Enterprise Getting \"InPrivate Desktop\" Sandboxed Execution Feature
https://www.bleepingcomputer.com/news/microsoft/windows-10-enterprise-getting-inprivate-desktop-sandboxed-execution-feature/

Lawrence Abrams published on his BleepingComputer site that Windows 10 Enterprise would apparently soon be getting a new "InPrivate Desktop" feature. This feature would allow administrators to run untrusted executables in a secure sandbox without fear that it might or could make any changes to the operating system or system's files.

Lawrence quotes the Windows 10 Insider Feedback Hub as saying: "InPrivate Desktop (Preview) provides admins a way to launch a throwaway sandbox for secure, one-time execution of untrusted software. This is basically an in-box, speedy VM that is recycled when you close the app!"

The quoted quest is no longer available in the Feedback Hub, but according to it's description, this feature is being targeted at Windows 10 Enterprise and requires at least 4 GB of RAM, 5 GB of free disk space, 2 CPU cores, and CPU virtualization enabled in the BIOS. It does not indicate if Hyper-V needs to be installed or not, but as the app requires admin privileges to install some features, it could be that Hyper-V will be enabled.

Lawrence wrote: "When the quest was live, I had attempted to install the InPrivate Desktop (Preview) app, but it was not accessible from the Microsoft Store as described. Furthermore, a wiki link in the quest description brought me to a page asking me to login to my Microsoft account. When I logged in with my account, I received a message that indicates that I need to be part of the Azure Active Directory (Azure AD) tenant for "Microsoft".

"It is too bad that I was unable to test this feature as it looks to be an interesting way to execute untrusted software without fear of permanent file modification, program installation, or configuration changes. This will also provide a new security boundary that Microsoft will need to protect and that researchers will be hammering for bug bounties."

It would be nice is this were to migrate down to other non-Enterprise builds.

New "Criptext" eMail service using the Signal protocol.
https://www.criptext.com/

<quote> Criptext is an encrypted email service that guarantees security, privacy and control over all your email communications. We don't have access to your emails nor do we store them in our servers. You're in control now.

- End-to-End Encryption
  All your emails are locked with a unique key that's stored on your device alone, which means only you and your intended recipient can read the emails you send.

- Signal Protocol
  The Criptext email service utilizes the open source Signal Protocol library, which protects your privacy and security throughout your entire Criptext experience.

- Open Source
  Criptext's source code is open to the entire privacy community to see. We actively work with our open source community to improve on the software in order to provide the best email experience.

- No Cloud Storage
  Criptext doesn't store any emails in its servers. All your emails are stored on your device alone, which means you're in control of your data at all times.

We know that privacy isn't just about encryption, which is why Criptext doesn't store any emails in our servers.

Available on Web & Mobile: Mac, Windows, Linux, iOC & Android

(Thanks to BleepingComputer for the pointer to this.)

Facebook Open Sources their "Fizz" TLS v1.3 Library For Speed and Security
https://thehackernews.com/2018/08/fizz-tls-ssl-library.html

Facebook has open sourced Fizz—a library designed to help developers implement TLS 1.3 protocol with all recommended security and performance related configurations.

As we know, TLS 1.3 incorporates a number of new features such as encrypting handshake messages to keep certificates private, redesigning the way secret keys are derived, and a zero

round-trip (0-RTT) connection setup to make some requests faster than TLS 1.2.

Written in C++ 14, Fizz is a reliable and highly performant TLS library that supports all major handshake modes, robust encryption algorithms, and performance optimizations aim to transfer data securely over 10 percent higher speed.

Facebook writes: "With zero copy encryption and decryption, tight integration with other parts of our infrastructure, and other optimizations, we see a reduced usage of memory and CPU with Fizz," Facebook says in a blog post announcing that it's open sourcing the library.

"In addition to the enhancements that come with TLS 1.3, Fizz offers an improved solution for middlebox handshake failures, supports asynchronous I/O by default, and can handle scatter/gather I/O to eliminate the need for extra copies of data."

Facebook has already replaced its older custom protocol, called Zero, with Fizz which is now responsible for securing trillions of connections every day at Facebook.

The social media giant says it has "deployed Fizz and TLS 1.3 globally in our mobile apps, Proxygen, our load balancers, our internal services, and even our QUIC library, mvfst. More than 50 percent of our internet traffic is now secured with TLS 1.3."

By open-sourcing Fizz, Facebook is sharing this technology with the world and helping drive deployments of the latest TLS 1.3 protocol across the Internet, making apps and services faster and more secure than ever.

Fizz is available on GitHub, and anyone can access and use it.

https://github.com/facebookincubator/fizz

Let's Encrypt Is Now Officially Trusted by All Major Root Programs
https://letsencrypt.org/2018/08/06/trusted-by-all-major-root-programs.html

Let's Encrypt Root Trusted By All Major Root Programs

Aug 6, 2018 • Josh Aas, ISRG Executive Director

As of the end of July 2018, the Let’s Encrypt root, ISRG Root X1, is directly trusted by Microsoft products. Our root is now trusted by all major root programs, including Microsoft, Google, Apple, Mozilla, Oracle, and Blackberry.

Today’s announcement that we’re trusted by all major root programs represents a major milestone for us, but it’s not the conclusion of our journey towards being directly trusted everywhere.

Certificates from Let's Encrypt have been widely trusted since our first issuance because of a cross-signature from another CA called IdenTrust. Browsers and operating systems have not, by default, directly trusted Let's Encrypt certificates, but they trust IdenTrust, and IdenTrust trusts us, so we are trusted indirectly. IdenTrust is a critical partner in our effort to secure the Web, as they have allowed us to provide widely trusted certificates from day one.

While Let's Encrypt is now directly trusted by almost all newer versions of operating systems, browsers, and devices, there are still many older versions in the world that do not directly trust Let's Encrypt. Some of those older systems will eventually be updated to trust Let's Encrypt directly. Some will not, and we'll need to wait for the vast majority of those to cycle out of the Web ecosystem. We expect this will take at least five more years, so we plan to use a cross signature until then.

As a subscriber of Let's Encrypt, today's milestone does not require any action on your part. Just continue to use best practices, including making sure that your ACME client (e.g. Certbot or an alternative) is regularly receiving software updates.

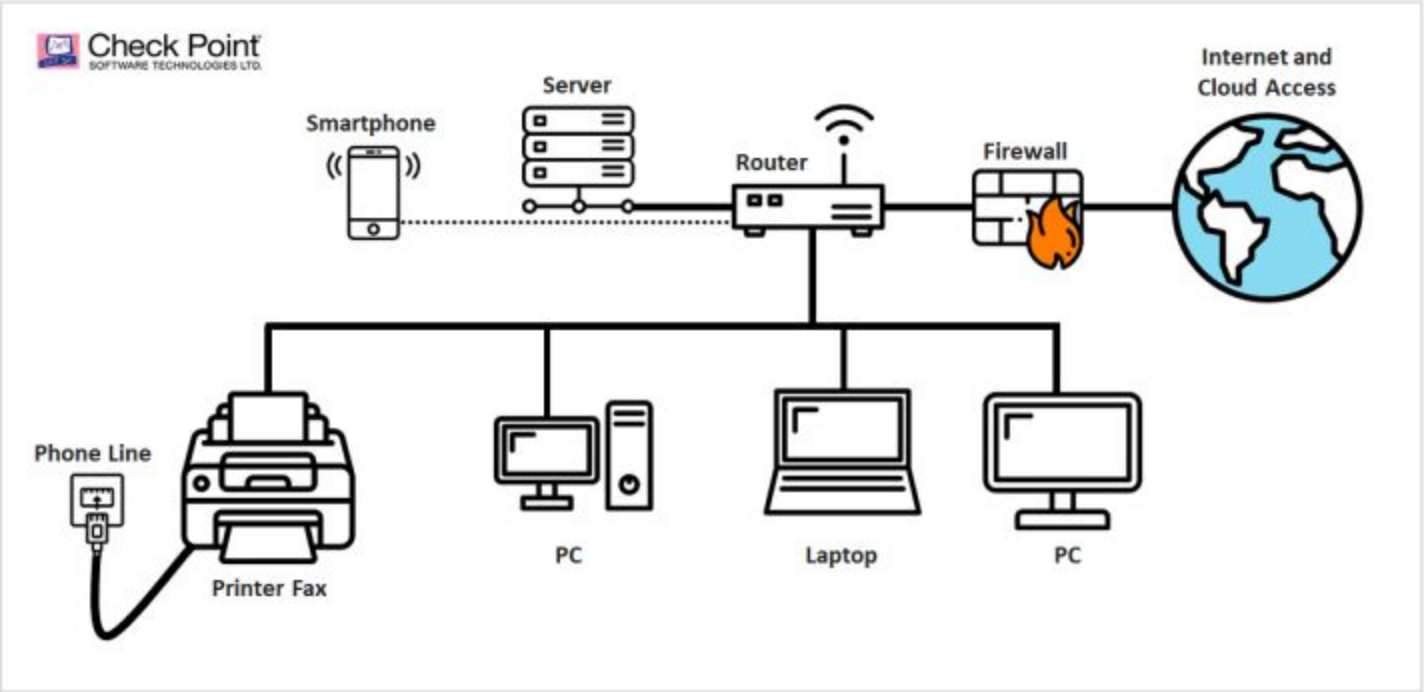
Let's Encrypt is currently providing certificates for more than 115 million websites. We look forward to being able to serve even more websites as efforts like this make deploying HTTPS with Let's Encrypt even easier. If you're as excited about the potential for a 100% HTTPS Web as we are, please consider getting involved, making a donation, or sponsoring Let's Encrypt.

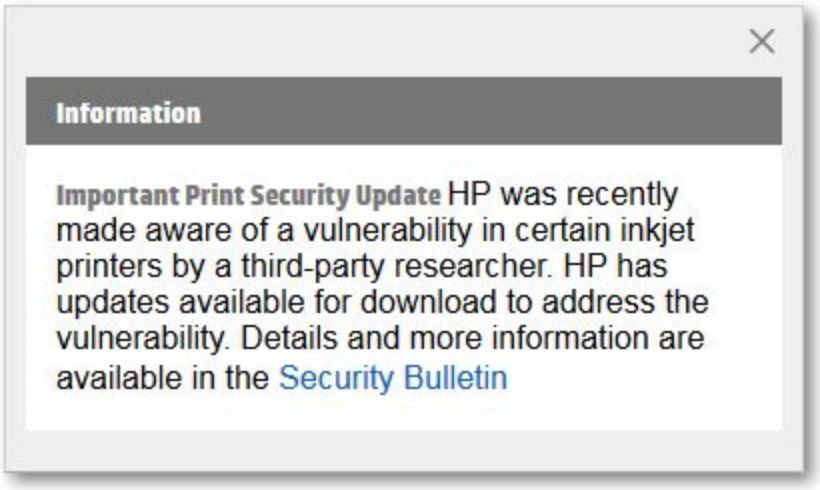
SpinRite:

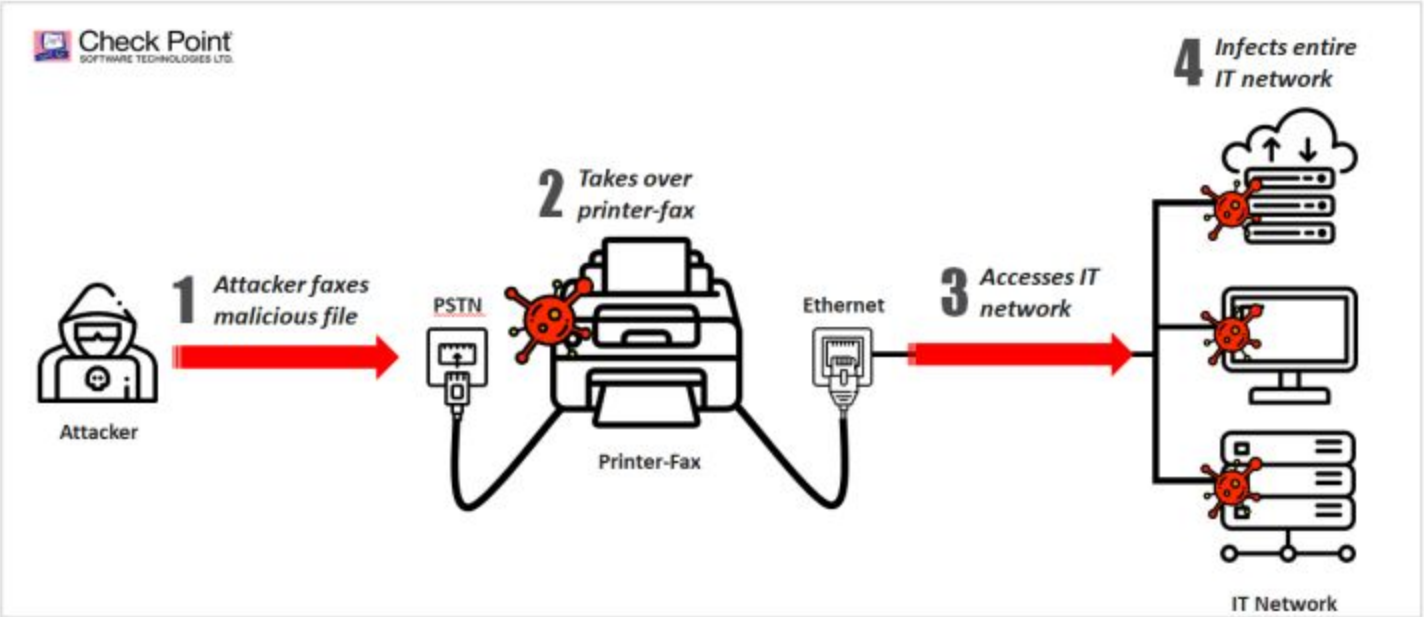
Jim Berry @jamesmberry / Monday, 6:58pm
Testimonial: I have a HP Z200 used in a high priority security context at my job that was experiencing the "click of death" and it wasn't booting up at all. I was able to see the drive in Linux but Windows was a NO GO. I ran SpinRite on Level 4 over the weekend (500 GB drive) and it still clicks (getting it replaced ASAP) but now it boots up. So the drive is being imaged onto a replacement and the machine will be re-deployed with the hearty thanks from my customer (Warden).

The Mega FaxSploit

https://research.checkpoint.com/sending-fax-back-to-the-dark-ages/
https://blog.checkpoint.com/2018/08/12/faxploit-hp-printer-fax-exploit/







Recon Phase
The first step in reverse engineering the firmware, once we loaded it to IDA, was to figure what is being executed, and in what environment. After a quick recon phase, we found out these details:

Architecture
The firmware is loaded to and executed by an ARM 32bit CPU, running in Big Endian mode. The main CPU uses a shared memory region to communicate with an MCU that controls the LCD screen.

Operating System
The Operating System is a ThreadX-based [ref. 3] real-time Operating System by Green Hills [ref. 4]. It uses a flat memory model in which there are many tasks that run in Kernel-Mode, all sharing the same virtual address space. Since this is a flat memory model, we would expect the tasks to communicate with each other over a message queue (a FIFO). In addition, the virtual address space is fixed, and no ASLR-based mechanisms are deployed.

Spreading Throughout the Network
Simply taking over a printer would be nice, but we wanted to do more. Indeed, if we could take over the entire computer network that the printer is part of, we could achieve a much bigger impact. So, knowing that one of the members in our Vulnerability Research team knows Eternal Blue quite well [ref.14] and that our Malware Research team did a similar research on Double Pulsar [ref.15], we decided to implement both NSA tools by using our file-based Turing Machine. And so, our payload implemented the following features:

1. Taking over the printer's LCD screen – demonstrating full control over the printer itself.
2. Checking if the printer's network cable is connected.
3. Using Eternal Blue and Double Pulsar to attack a victim computer in the network, taking full control over it.

To our knowledge, we now had the first (publicly documented) printer capable of using Eternal Blue and Double Pulsar to autonomously spread an attacker's payload over a computer network.

Questions and Answers:

Q: What is this research about?
A: Check Point Research has uncovered critical vulnerabilities in the fax protocol. These vulnerabilities allow an attacker with mere access to a phone line, and a fax number to attack its victim's all-in-one printer – allowing him full control over the all-in-one printer and possibly the entire network it is connected to.

Q: Does this apply only to all-in-one printers?
A: No. We conducted our research on all-in-one fax printers; however similar vulnerabilities are likely to be found in other fax implementations, such as fax-to-mail services, standalone fax machines, etc.

Q: Who uses fax anyway?
A: Surprisingly fax is still used by many industries, governments and individuals around the world. These include the healthcare industry, legal, banking and commercial – some of which are governed by regulations, and others simply for legacy reasons.

Q: What is the severity level of this vulnerability?
A: HP classified this vulnerability as 'Critical'. Full details can be found here.

Q: How does this affect organizations and consumers?
A: Once an all-in-one printer has been compromised, anything is possible. It could be used to infiltrate an organization's or consumer's internal network, steal printed documents, mine Bitcoin, or practically anything.

Q: Does this apply to all fax machines?
A: Our research was done on HP Officejet all-in-one printers though this was merely a test-case. We strongly believe that similar vulnerabilities apply to other fax vendors too as this research concerns the fax communication protocols in general.

Q: Is it widespread?
A: By our estimates, there are currently hundreds of millions of fax machines still in use around the world. Financial reports from Wall Street indicate that tens of millions of all-in-one printers are sold worldwide each year.

Q: Has it been fixed?
A: We worked closely with HP to fix the vulnerability and, following the process of responsible disclosure, they managed to release a patch before this publication. In fact, if your device is already configured to auto-update then the patch has likely already been applied. This patch, however, only applies to HP all-in-one printers and the vulnerability may well still apply to devices from other manufacturers as well.

Q: What should I do to protect myself?

A: If you own an HP Officejet all-in-one printer then follow the instructions from HP here. In addition, you should implement segmentation policies, software patching and proper IT hygiene – please see our 'Recommendations' section in the above blog post. Also, if you are no longer actually using the fax functionality in your all-in-one printer then we recommend you to disconnect the PSTN line.

Q: Has it been seen it the wild?
A: Not yet. Our research was intended to highlight a potential security risk.