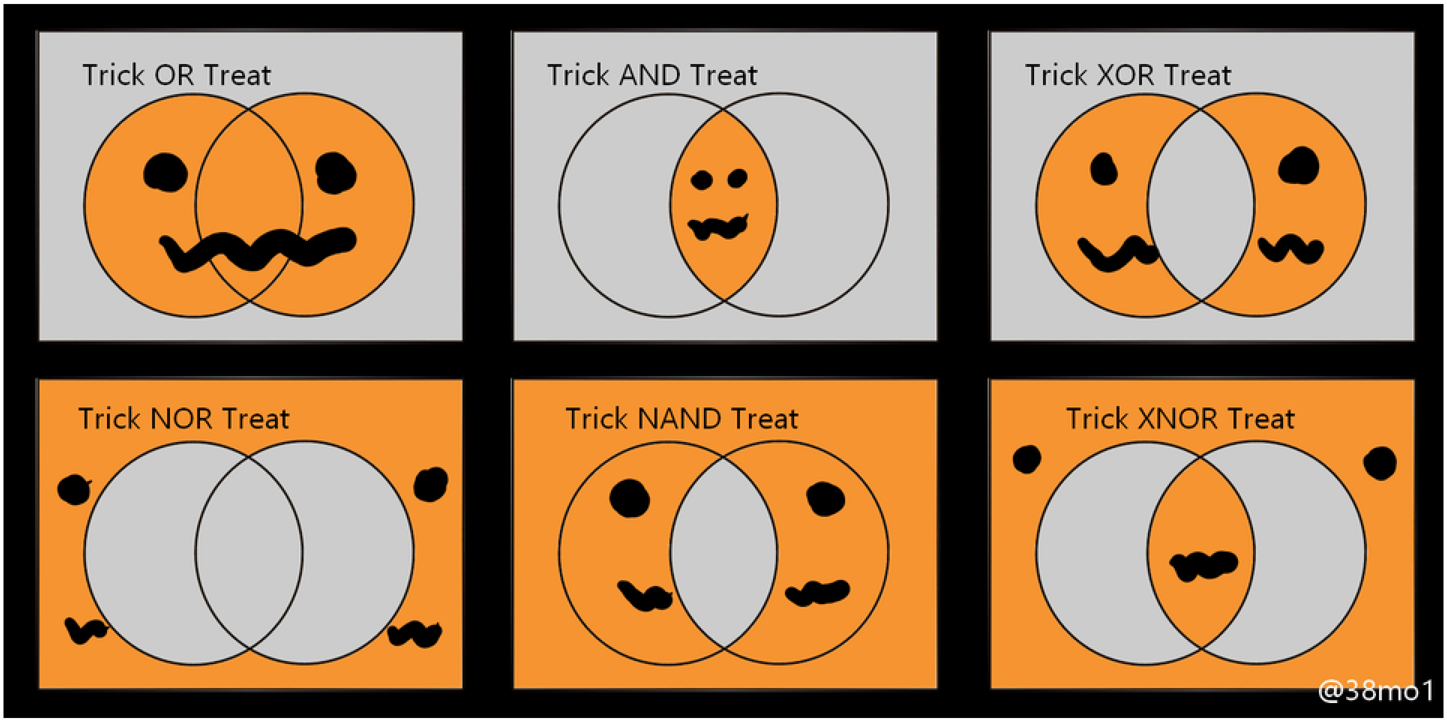
Security Now! #794 - 11-24-20

Cicada

This week on Security Now!

This week we have a bunch of news on both the Chrome and Firefox fronts with patches, updates and new features. We have a comical bit of news from the ransomware front, more troubling ongoing WordPress attack specifics, including a weird eCommerce site spoofing attack. We look at the future consequences of ongoing vulnerability announcements coupled with their very incomplete patching, and Android's bold move right into the middle of the unbreakable end-to-end encryption controversy. And then we'll conclude with a look at a large, multi-year (as in 11-year) advanced very-persistent threat state-based attack perpetrator.



We missed Halloween... but still, kinda fun.

Browser News

Chrome moves to release 87

Fixes 33 security vulnerabilities, 10 being high-severity, 11 medium and two low.

To provide some sense for the nature of the high severity flaws fixed, we had a: Use after free in payments, Inappropriate implementation in filesystem, Inappropriate implementation in cryptohome, Race in ImageBurner, Insufficient policy enforcement in networking, Insufficient data validation in WASM, Use after free in PPAPI, Use after free in WebCodecs, Heap buffer overflow in UI, and a Heap buffer overflow in clipboard. The "Use after free in WebCodecs" was awarded a $15K bounty split between its two researchers. The other nine are all $TBD.

I wonder what the awarding logic is for these, since most of the medium and low criticality vulnerabilities have bounties set ranging between a high of $7500 and a low of $500. One thing is clear, Google is not just paying some fixed bounty. They apparently really look at what went into it and what having the vulnerability repaired means, and they pay accordingly.

Interestingly, among those 10 high-severity vulnerabilities is Samy Kamkar's CVE-2020-16022, which Google called "Insufficient policy enforcement in networking." This is the "NAT Slipstream" attack we covered in detail two weeks ago. Although it can perhaps best be leveraged remotely with a browser, it was never a fault of any browser, and it was enabled by all. So it's nice that Chrome has now blocked connections to those remote SIP ports 5060 and 5061 from being targeted by any browser code. And as we noted two weeks ago, all browsers are sure to eventually follow. And from this day forward web servers will be unable to accept incoming connections from web browsers. This is becoming an industry full of warts and exceptions.

And as for the other things fixed, Google is keeping their details private "until the majority of users are updated with a fix" ... which, in practice, actually means "until no one really cares anymore and stops asking." And they're right... we really don't are. The details are useful for spotting trends and, as we often do here, drawing broader and more useful general conclusions. But problems that are fixed on any platform that's able to push their updates are mostly useful for what they might be able to teach us about how to avoid such vulnerabilities in the future. Because until they are discovered and fixed, they are exploitable. And I think that's really where we are at this point in the industry. We really do need to stop all this nonsense and start getting serious about figuring out how to not make these mistakes in the first place.

There were a handful of interesting new Web developer features for cookie and fonts, and this release 87 increased Chrome's somewhat weird deprecation from the 1% of users of release 86 to 50% of its users in last week's 87. The plan is for January's release of 88 to finally kill off FTP completely.

Chrome 87 also comes with some user-facing performance improvements. 87 now has "page occlusion" tracking so that the browser can determine what's actually visible to the user and will throttle pages and tabs that are not visible. The other new feature to have finally landed, initially for Chrome in Android is the so-called "Back-forward cache" or "bfcache." I say "to finally land in Chrome" because both Firefox and Safari have had a bfcache for years across both desktop and mobile. The technology is a local cache optimizing for the use of the back and forward arrows. A great deal of work is required to setup a page, especially when a lot of JavaScript is going on behind the scenes. Firefox and Safari have long been very smart about holding a page's entire state — including its expensive-to-establish JavaScript state — when the user leaves a page so that if they should hit the back arrow, that page can be instantly restored from RAM-based history. Chrome has previously been moving toward this goal with bits and pieces of this. But it's finally here in Chrome 87 for Android. The mainstream desktops are expected to receive it in the future.

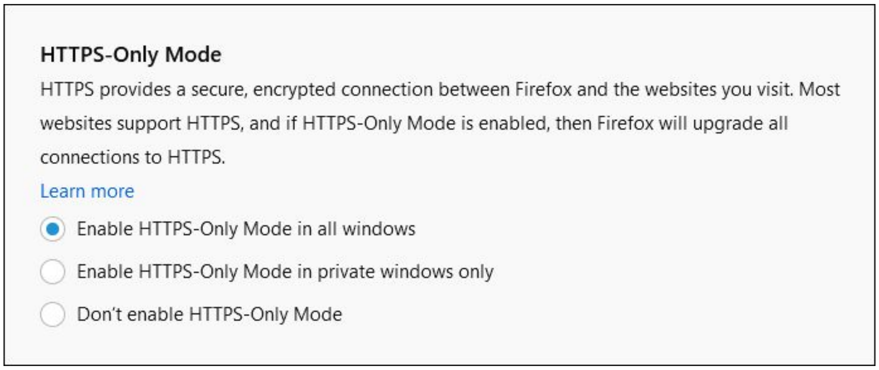
Explicit Publication of Privacy Practices

During last June's Apple WWDC 2020 developer conference Apple announced that all App Store app listings would soon be required to include an explicit privacy prompt (label) that lists all the data apps collect from their users and which are used to track users across apps. Those Apple privacy labels are scheduled to go live next month on December 8th.

And last Wednesday, Google announced that starting January 18th, 2021, they would also be adding a new section to the Chrome Web Store where extension developers would be able to disclose what user data they're collecting and what they plan to do with the information. After January 18th, a "Privacy practices" button will appear on each extension's Web Store listing. To get ready for the change, Google added a new section to their web store developer dashboard last week to enable extension developers to indicate what they're collecting, and why.

Firefox 83 gets HTTPS-Only Mode (disabled by default)

Currently, Firefox's new “HTTPS-Only Mode” is disabled by default. But security-conscious users using Firefox 83 or later may go to “about:preferences#privacy” then scroll down to the bottom of the page, where a new section has just been added:



HTTPS-Only Mode

HTTPS provides a secure, encrypted connection between Firefox and the websites you visit. Most websites support HTTPS, and if HTTPS-Only Mode is enabled, then Firefox will upgrade all connections to HTTPS.

Learn more

● Enable HTTPS-Only Mode in all windows

○ Enable HTTPS-Only Mode in private windows only

○ Don’t enable HTTPS-Only Mode

While this is enabled, Firefox will attempt to connect to the HTTPS versions of websites when a non-explicit URL is provided, or to attempt to upgrade and prefer an HTTPS connection whenever possible. Mozilla explained that the feature works by attempting to find the HTTPS version of any website, even if the user has accessed the site by typing or clicking on an HTTP link. And with HTTP-Only Mode enabled, if Firefox is unable to auto-upgrade a site to an HTTPS connection, the browser will show an error to the user and ask them to click a button to confirm they want to access a website via an older HTTP connection. So that’s slick. It’s exactly what we would want.

I did immediately worry that one possible **huge** problem with doing this is that it might force all

connections to the Localhost to also be HTTPS.
In the first place, that's moronic. Localhost is simply the system's own local stack. Have you ever done a netstat on a UNIX machine? It's full of network servers on the local machine's IP stack.
It turns out that using IP and local sockets is one of the more elegant IPC (inter-process communication) links available on any networked OS. And, yes, tiresomely, it IS possible for this IPC mechanism to be abused if something malicious somehow gets into someone's machine. But in no way does Locahost in and of itself create any vulnerability. Microsoft has never been secure from any local attack. I run with a nifty on-the-fly spell checker that watches everything I type. If it were malicious it could be functioning as a keystroke logger. And anything else in the system could be too, without any announcement. If something evil gets into your machine it's game over. So protecting the browser from something that's setup shop on Localhost is nuts. Despite that, Microsoft has been threatening for years to shut down its browser's local access to the Localhost domain. But every time they try, the world of actual users explodes, because doing so breaks too many things. Having Localhost is so handy, reliable and elegant that it's being quietly used by many systems. And I'll note that SQRL is among them. SQRL clients run a little HTTP web server (notice that I said HTTP and not HTTPS) on port 25519. There's ZERO need for TLS on the local stack. It makes no sense. The IP connection is an abstraction. The fact that the Internet also uses IP packets doesn't somehow poison local usage. There's no point in encryption for privacy because nothing ever leaves the machine — the virtual connection is to the operating system's IP stack. And there's no need for authenticating the localhost because it cannot be anything else. It's just another local API.

The good news is, all of this has already occurred to Mozilla (and to Google) and they're treating Localhost as a "presumed trusted" exception to the HTTPS always rule. And they really had no choice.

Mozilla also fixed 21 vulnerabilities in Firefox 83, including that bug in its use of the Freetype library that had been used in attacks against Chrome. We noted at the time that everything that used Freetype and allowed attacker-provided font glyphs was a potential target. Web browsers were perhaps the biggest target of all since they fit all the requirements out of the box and were inherently exposing themselves. So, "CVE-2020-15999: Heap buffer overflow in freetype" is now also closed in Firefox.

A few other honorable mentions about this release of Firefox 83. Mozilla wrote...

- Firefox keeps getting faster as a result of significant updates to SpiderMonkey, our JavaScript engine, you will now experience improved page load performance by up to 15%, page responsiveness by up to 12%, and reduced memory usage by up to 8%. We have replaced part of the JavaScript engine that helps to compile and display websites for you, improving security and maintainability of the engine at the same time.

- Pinch zooming will now be supported for our users with Windows touchscreen devices and touchpads on Mac devices. Firefox users may now use pinch to zoom on touch-capable devices to zoom in and out of webpages.

- Picture-in-Picture now supports keyboard shortcuts for fast forwarding and rewinding videos: use the arrow keys to move forward and back 15 seconds, along with volume controls. When you are presenting your screen on a video conference in Firefox, you will see our improved user interface that makes it clearer which devices or displays are being shared.

- For the recently released Apple devices built with Apple Silicon CPUs, you can use Firefox 83 and future releases without any change. This release (83) will support emulation under Apple’s Rosetta 2 that ships with macOS Big Sur. We are working toward Firefox being natively-compiled for these CPUs in a future release.

- This is a major release for WebRender as we roll out to more Firefox users on Windows 7 and 8 as well as on macOS 10.12 to 10.15.

And before we leave Firefox...
I appreciated ZDNet’s headline: “Fearing drama, Mozilla opens public consultation before worldwide Firefox DoH rollout.”

Mozilla wants to enable DNS-over-HTTPS (DoH) in Firefox for all users worldwide, but once burned, twice shy, Mozilla wants to creep forward a bit more cautiously, hearing from ISPs, governments, and companies before they flip the switch.

So, last Thursday, they initiated a period for public comment and consultation about the ways they could safely enable support for the previously controversial DNS-over-HTTPS (DoH) protocol.

My reference to “once burned” was to the backlash of criticism they encountered last year in the UK over their stated plan to support DoH inside Firefox. As we'll recall, UK government officials, law enforcement agencies, and local Internet service providers criticized Mozilla — using some quite over-the-top language of "Internet Villain" — over Mozilla's desire to roll out DoH. Those opposed, claimed that it would help bad guys bypass enterprise firewalls and parental controls blocklists. Oh! The villainy!!

As we'll recall, once of the concerns was that DNS was meant to be distributed and aiming all DoH connections to a single point would lose the benefits of DNS's historical distributed architecture. But since then, as expected, many US ISPs have established their own DoH servers.

So, at the time Mozilla chose to back off of their timeline and agreed to delay deploying DoH inside the UK. Yet Mozilla did deploy DoH for all Firefox users in the US, where it's been under use at scale since February. And Mozilla has always planned to roll out DoH to all of its users across the world — barring the UK, where it promised not to deploy it.

So now, the current consultation period exists as a way to give "stakeholders" — governments and ISPs — an opportunity to speak up or forever hold their peace. This consultation period runs from last Thursday through January 4th, 2021. And Mozilla has promised to consider every reasonable and practical suggestion, whatever it might be.

However, since the ouchy “Internet Villain” backlash last year Mozilla has addressed most of the DoH criticism. They have added a "canary" domain that can be queried on managed networks to force Firefox to disable DoH support and defer to local enterprise policies for DNS management.

They added additional support for DoH providers besides Cloudflare which was the sole provider last year, and they made DoH settings management simpler and easier.
Apple, Google and Microsoft, for their part, having also announced plans to support the DoH protocol and having watched Mozilla stumble last year, have all deployed enterprise-friendly DoH implementations from the start. As we know, Google's DoH support in Chrome already went live.

Ransomware News

I have one bit of entertaining ransomware news:
Aside from just noting that devastating ransomware attacks continue, I wanted to share the news of a new tactic being employed by the Egregor Ransomware to announce its dastardly deeds and presence within an enterprise's network. The ransomware gangs know that many businesses will attempt to cover up a ransomware attack to keep it from becoming public. They'll just claim some generic network outage. And in sufficiently large organizations this coverup extends to include their own employees for fear of a news leak tanking their stock price and incurring reputation damage.

So now, in a move clearly designed to increase the pressure and increase the likelihood of a very public leak, after an attack the Egregor operation locates all of the available hardcopy printing devices within a network and uses them to repeatedly print the announcement of their successful attack and their ransom demands.

You just gotta shake your head. I'd be tempted to ask "What next?" ... but I'm sure we're destined to find out.

Word*de*Press

WordFence, the WordPress web application firewall company, posted news of the latest in an ongoing and escalating attack on WordPress sites:
https://www.wordfence.com/blog/2020/11/large-scale-attacks-target-epsilon-framework-themes/

Their posting was titled: "Large-Scale Attacks Target Epsilon Framework Themes" and they wrote:

"On November 17, 2020, our Threat Intelligence team noticed a large-scale wave of attacks against recently reported Function Injection vulnerabilities in themes using the Epsilon Framework, which we estimate are installed on over 150,000 sites. So far today, we have seen a surge of more than 7.5 million attacks against more than 1.5 million sites targeting these vulnerabilities, coming from over 18,000 IP addresses. While we occasionally see attacks targeting a large number of sites, most of them target older vulnerabilities. This wave of attacks is targeting vulnerabilities that have only been patched in the last few months."

They then proceed to list the 15 WordPress themes that are known vulnerable to these attacks.

At this point, the attacks appear only to be probing, looking for vulnerable WordPress instances.

Presumably accruing a master list of vulnerable targets. This makes sense, since the use of 18,000 source IPs sounds like a Botnet being used as the scanning phase of a future targeted attack. The WordFence folks noted that both the probing attacks and the later exploitation use POST queries to “admin-ajax.php” and will not leave distinct log entries. They have also confirmed that an exploit chain exists to permit full Remote Code Execution enabling a full site takeover. Obviously, someone else is aware of this as well.

Jayant Shukla, the CTO and co-founder of K2 Cyber Security told ThreatPost in an eMail: “WordPress powers as much as a third of all websites on the internet, including some of the most highly trafficked sites and a large percentage of e-commerce sites. So WordPress security should be of top concern to organizations. This latest attack on a recently patched injection vulnerability on WordPress sites which use the Epsilon Framework themes, is looking for sites that have neglected to install the latest updates. As we know from past research, as many as 60 percent of successful attacks are on vulnerabilities that already have a patch to prevent its exploit. Organizations need to take the security of their WordPress sites more seriously, starting with keeping the plugins and software up-to-date and patched.”

No one listening to this podcast doubts any of that for a moment. But we also know how much more easily this is said than done.

And as if all that wasn't enough…

A new cybercrime gang has been found to be taking over vulnerable WordPress sites to install hidden e-commerce stores with the purpose of hijacking the original site's search engine ranking and reputation and promoting online scams. Sheesh.

The attacks were discovered earlier this month when a WordPress honeypot that had been set up was targeted. The honeypot was managed by Larry Cashdollar, a security researcher for the Akamai security team.

The attackers initially leveraged brute-force attacks to gain access to the site's admin account, after which they overwrote the WordPress site's main index file, appending their own malicious code. Although the malicious code was heavily obfuscated, Larry indicated that the malware's primary function was to act as a proxy to redirect all incoming traffic to a remote command-and-control (C&C) server under the attacker’s control. This server hosted the actual attack logic.

The way the attack worked, a user would visit a hacked WordPress site. The hacked code on the WordPress site would then redirect the user's request to the malicious C&C server. If a user met certain criteria (it’s unclear what that would be) the C&C server would instruct the original WordPress site to reply with an HTML file containing an online store which was peddling a wide variety of mundane objects. Presumably, this criteria would be designed to avoid detection by the site’s actual admin and owner.

The Akamai researcher indicated that during the time the hackers had access to his honeypot, they hosted more than 7,000 e-commerce stores that they intended to serve to incoming visitors. So, weird as this seems, it’s no small operation. It’s clearly intended as a long-lived scam.

In addition, the hackers generated XML sitemaps for the hacked WordPress sites that contained entries for the fake online stores together with the site's authentic pages. The attackers generated the sitemaps, submitted them to Google's search engine, then deleted the sitemap to avoid detection. Although this procedure seemed harmless, it had a lasting and very negative impact on the WordPress site because by poisoning its keywords with unrelated scam entries, the site’s search engine results page ranking was significantly reduced.

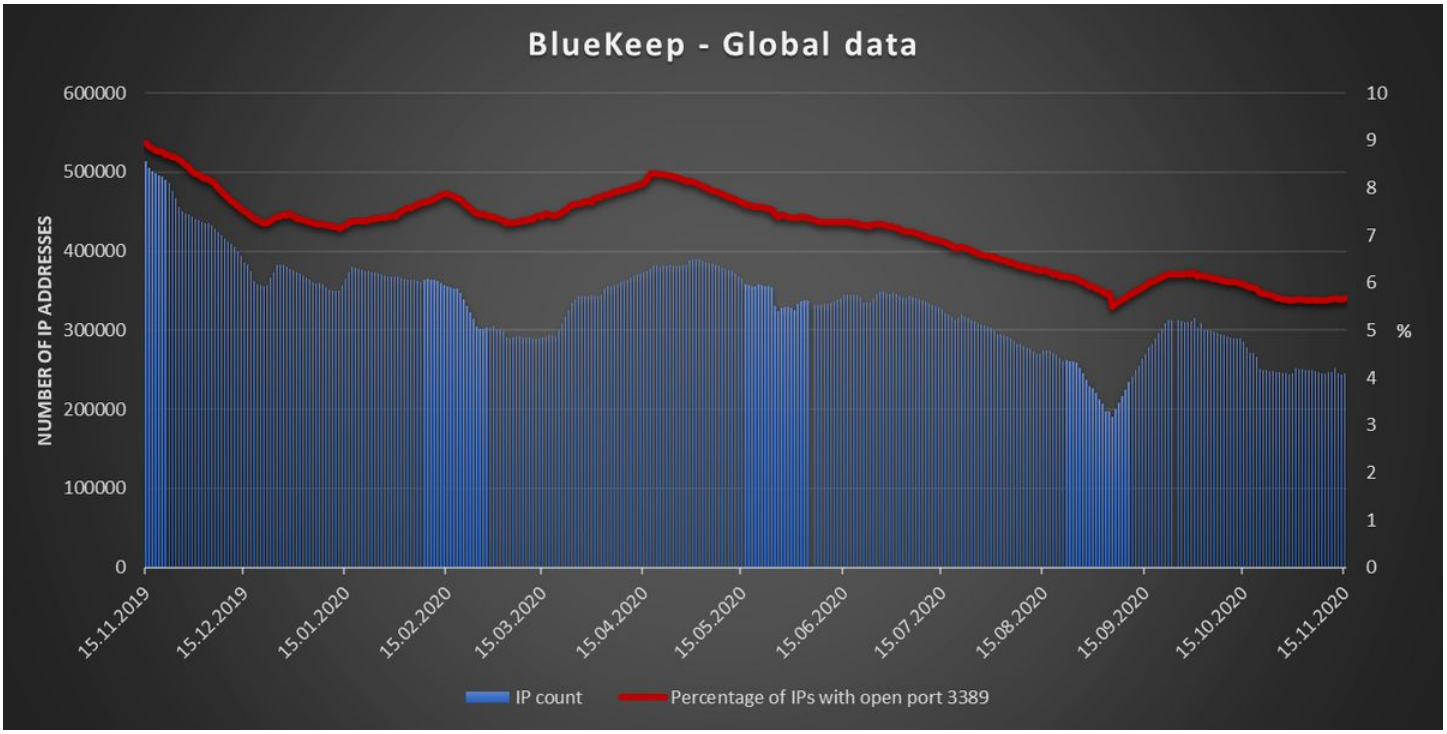
The more we learn the more WordPress appears to be quite out of control.

Security News

“More than 245,000 Windows systems still remain vulnerable to BlueKeep RDP bug”
Let me repeat that: today, more than 245,000 Windows systems are still vulnerable to the BlueKeep complete authorization bypass remote desktop protocol bug. HOW is that possible?

It's clearly the case that the #1 biggest scandal on the security side of the IT industry today must be the lack of timely — or ever — patching of highly public, widespread, remotely accessible vulnerabilities in today's computer networks. We're going to wrap up today's podcast by looking at some details of APT10, a well known advanced persistent threat group. The details are interesting and sobering. But the question of how such threat actors gain access to their victims... well, is it really much of a question?

This chart made me do a double-take:



The chart is a timeline of exactly one year, from November 15th of 2019 to November 15th of 2020. During that time the number of open port 3389 dropped from 9% to just under 6%.

Now, in fairness, the work-from-home response to the novel Coronavirus did put sudden pressure upon the whole terrain of remote access. And we can clearly see its effect in the chart's rise through March and April. But, regardless, right now, today, 245,000 IPs answering on port 3389 remain vulnerable to BlueKeep. That's about 25% of the 950,000 Windows machines that were initially discovered to be vulnerable to BlueKeep during a first scan last May, 2019.

But that's not all. Today, more than 103,000 Windows systems also remain vulnerable to the SMBGhost vulnerability in the Server Message Block v3 (SMB) protocol that was discovered and patched back in March, 2020. Either of these vulnerabilities allow attackers to take over Windows systems remotely and are considered some of the most severe bugs discovered — and patched — in Windows over the past few years. But despite their severity, an incredible number of Windows systems have remained unpatched and thus vulnerable to remote takeover. Is it any wonder that I've had to stop enumerating each week's ransomware attacks for fear of boring our listeners?

And in fairness it's not just Windows. Just look at this table which was recently compiled by a Czech security researcher:

CVE	**PRODUCT**	**UNPATCHED SYSTEMS**	**CVSSv3**
CVE-2019-0211	Apache web server	3,357,835	7.8
CVE-2019-12525	Squid	1,219,716	9.8
CVE-2015-1635	Microsoft IIS	374,113	10
CVE-2019-13917	Exim	268,409	9.8
CVE-2019-10149	Exim	264,655	9.8
CVE-2019-0708	(BlueKeep) Windows RDP	246,869	9.8
CVE-2014-0160	(Heartbleed) OpenSSL	204,878	7.5
CVE-2020-0796	(SMBGhost) Windows SMB	103,000	10
CVE-2019-9787	WordPress	83,951	8.8
CVE-2019-12815	ProFTPD	80,434	9.8
CVE-2018-6789	Exim	76,344	9.8

Now, we might all say "Yeah, that's not good, but that's other people's machines and networks." But one thing to remain cognizant of, is the potential for cross-network contagion. We've seen this earlier in Managed Service Providers where, for example, an MSP gets themselves infected, and the bad guys realize that they've hit the motherlode because they now have access to all of that MSP's client's networks. We are becoming more deeply interconnected. And the trend is clearly one of increasing our outsourcing of various ancillary and non-mainline business functions such as order fulfillment, payroll management, and an increasing number of network and cloud-based infrastructure functions.

The upshot is that while the networks under our own control may be battened down and secure, we're increasingly needing to implicitly trust that the growing number of external entities we're connecting to are also similarly taking their own security seriously. When queried, they will all profess to be totally safe and secure. Who wouldn't? But that means that most, if not every one of the companies represented in that table above, would proclaim exactly the same thing.

Microsoft vividly demonstrates for us every single month that things are **not** getting any better. It's not as if the “critically buggy code problem” has been solved. So they're leaving a lengthening trail of critical vulnerabilities in their wake. All of the evidence shows that a significant percentage of these things are never going to be patched and the table above shows just how devastating that ever-lengthening trail of critical vulnerabilities will be to our future.

Three digits may not be sufficient for this podcast after all. :)

Google moves forward on E2EE encryption for Android's Messages

As all of our regular listeners know, the politically-charged and supremely interesting topic of unbreakable End-to-End Encryption (E2EE) is a frequent focus of this podcast. So it's of interest that Google has begun rolling out end-to-end encryption for their long-awaited SMS-replacement Rich Communication Services.

Unlike iMessage, which is a proprietary Apple-only facility, RCS — Rich Communication Services — is an open industry standard. Wikipedia has this to say about it:

> Rich Communication Services (RCS) is a communication protocol between mobile telephone carriers and between phone and carrier, aiming at replacing SMS messages with a text-message system that is richer, provides phonebook polling (for service discovery), and can transmit in-call multimedia. It is part of broader IP Multimedia Subsystem. It's also marketed as Advanced Messaging, Chat, joyn, SMSoIP, Message+, and SMS+. In early 2020, it was estimated that RCS is available from 88 operators throughout 59 countries in the world. There are approximately 390 million users per month and the business is expected to be worth $71 billion by 2021.

Despite its specification and ratification back in 2007, adoption of RCS has been lackluster. But the system offers a number of new and useful features including typing indicators, presence information, location sharing, longer messages, and better media support. This, in turn, provides better-quality photos and videos, chat over Wi-Fi, knowing when a message is read, sharing reactions, and better capabilities for group chats.

So, last Thursday Google said that they have completed their worldwide rollout of RCS and are moving into a new phase: adding native end-to-end encryption. So Android's native messaging platform would potentially be able to offer the privacy and authentication features of Threema, Signal, WhatsApp, and so on.

At the moment, end-to-end encryption in Android Messages is only available to those using the beta version of the app. And of course, it requires a beta version user to be at each end. Google's rollout is expected to continue into next year and the best news of all is that Google wisely chose not to roll their own solution. Yay!! End-to-end encryption has been added to the “solved problems” list, thus no need to do it again. Google has adopted the very well-designed and already time-tested Signal protocol.

https://www.gstatic.com/messages/papers/messages_e2ee.pdf

This means that Signal goes mainstream in Android moving forward. And this is a big deal and a big day for end-to-end encryption. With “Signal for RCS” built into Android’s Messages app, Google said, “eligible conversations will automatically upgrade to be end-to-end encrypted.”

Given Android’s spread and reach at three quarters of the entire mobile OS platform globally, and that no 3rd-party app will be needed for automatic unbreakable encryption to be provided to is users in the future, this represents a big poke in the eye to law enforcement and intelligence services worldwide. I have a feeling that we haven’t heard the end of this intriguing encryption debate. The ante was just upped.

Cicada

The group’s use of the ZeroLogon vulnerability was what first brought them to me attention. But the more I dug into the background of their existence the more interesting they became. They are Chinese, state sponsored and sanctioned, and a highly capable advanced persistent threat (APT) group known by several names including Cicada, APT10, Stone Panda and Cloud Hopper. Cicada has been involved in cyber espionage operations since 2009.

Their cyber intrusions are generally aimed at large international Japanese companies with attack presence having been spotted in 17 geographical regions and multiple business sectors. They are an advanced persistent threat because they burrow into a large company's network and remain performing long-term intelligence gathering. The business sectors they focus upon include automotive, pharmaceutical, and engineering sectors, as well as managed service providers (MSPs).

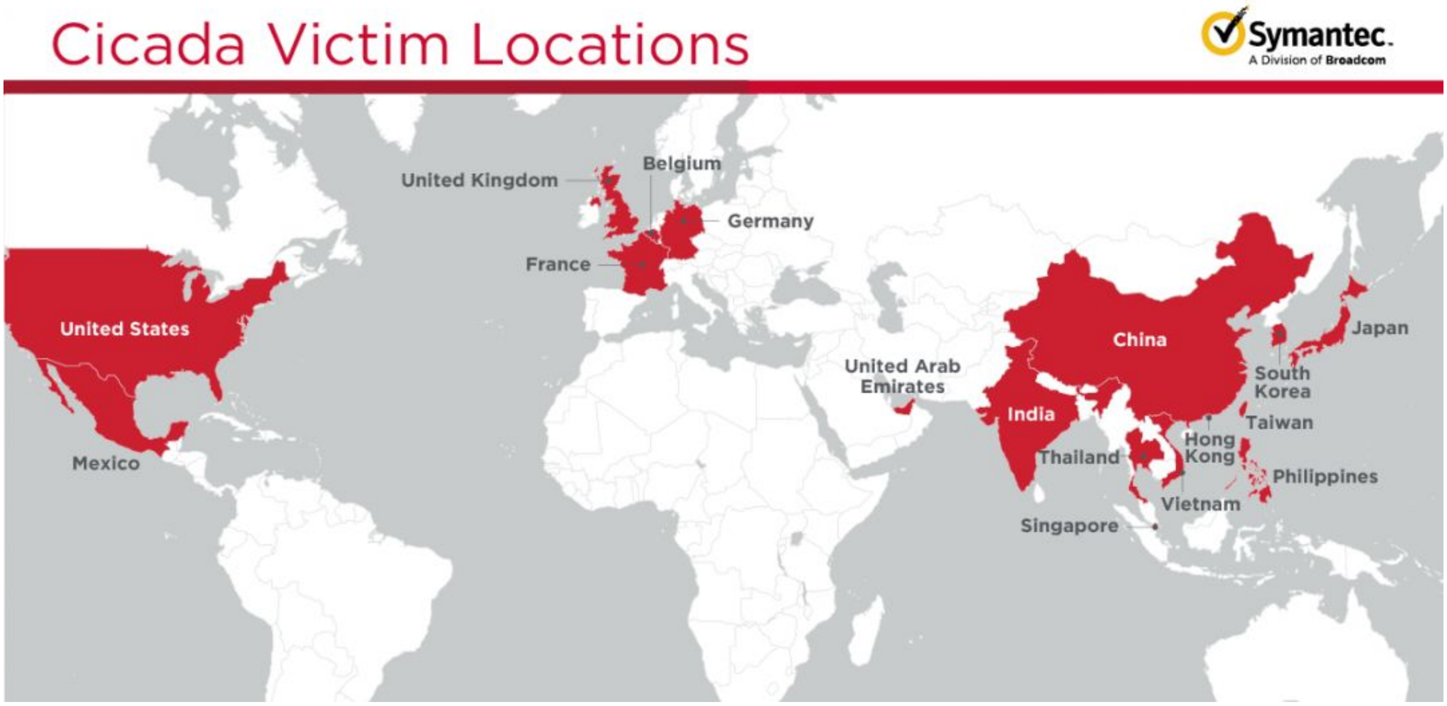
The group uses what has become known as “living-off-the-land tools” — using what's available on a network — as well as custom malware in this attack campaign, including a new custom malware “Backdoor.Hartip” which Symantec has not seen being used by the group before. Among the machines compromised during this attack campaign were domain controllers and file servers, and there was evidence of files being exfiltrated from some of the compromised machines.

The attackers extensively use DLL side-loading in this campaign, and were also seen leveraging the ZeroLogon vulnerability that was patched in August 2020. DLL side-loading takes advantage of the Windows so-called Side-by-side or WinSxS system to trick Windows into loading a malicious DLL for an application when it is run. DLL side-loading is becoming an increasingly popular means of sneaking malicious code into Windows processes.

So it should be no surprise that the present campaign was first discovered when Symantec observed suspicious DLL side-loading activity on one of their customer’s networks. That triggered an alert in their Cloud Analytics technology, which is part of the Symantec Endpoint Security Complete (SESC) offering. The activity was brought to the attention of their analysts then passed on their investigations team for further analysis.

Since Symantec's instrumentation is widely spread, once the IOC (indication of compromise) was identified, they were able to apply that across their entire coverage space to identify other previously unknown victim networks and enterprises.

And as for persistence, once something like this is discovered, it's possible to scroll back through prior activity logs to reexamine behavior that wasn't appreciated for what it was at the time, and to carefully examine the timestamps on files that were not previously suspect, to determine how far back an intrusion took place. Although the Cicada group is known to have existed since 2009, in this case, this campaign has been ongoing since at least mid-October 2019 through the beginning of last month, October 2020. The attacking group has been active on the networks of some of its victims for a year. The campaign is very wide-ranging, with victims in a large number of regions worldwide.



Compromised corporate networks have been found in the United States, the UK, France, Belgium, Germany, the UAE, India, China, Hong Kong, Singapore, Vietnam, the Philippines, Taiwan, South Korea and Japan.

Although it's unusual to see a Chinese-government-linked group attacking companies within their own borders, like many of the companies targeted in this campaign, the target is a subsidiary of a large Japanese organization.

Symantec's research discovered similar DLL side-loading malware on EVERY victim network and, as I noted above, the attacks employed a wide variety of so-called "living-off-the-land", dual-use, and publicly available tools and techniques in these attacks, including:

- RAR archiving – files are transferred to staging servers before exfiltration. They may be encrypted or compressed, to make them easier to extract.

- Certutil – a command-line utility that can be exploited and used for various malicious purposes, such as to decode information, to download files, and to install browser root certificates.

- ADfind – a command-line tool that can be used to perform Active Directory queries.
- Csvde – can be used to extract Active Directory files and data.
- Ntdsutil – can be used as a credential-dumping tool.
- WMIExec – used for lateral movement and to execute commands remotely.
- PowerShell - yes... bringing power not only to admins but also to attackers.

And get this... they upload their ill gotten goods to legitimate Internet cloud file-hosting services for exfiltration. That makes sense since such services are unlikely to be blocked from local use.

Observing the activity patterns, the amount of time attackers spent on specific networks varied widely with the attackers spending a significant amount of time — as in many months — on the networks of some victims and just a few days on other victim networks. And, in some cases, the attackers would spend time on a network, activity would cease, then resume months later.

So, I wanted to take a moment to just switch our perspective a bit.

When you sit back to think about this, it's a bit astonishing and quite sobering. And it should be clear to us all that this is not the only such operation ongoing. For example, we know with absolute certainty thanks to Wikileaks and Edward Snowdon that U.S. based and inherently state sponsored, intelligence services have developed and are developing powerful tools for doing the same.

We tend to focus upon the big splashy “Oh my God” ransomware attacks which make headlines, which take out ads on Facebook, expose Matthew McConaughey's private contracts and spit out ransom demands on all of a company's printers. But there's another very much different, entirely covert and arguably far more insidious reality that goes largely unremarked... because it doesn't want to be found — ever. It wants to borough into many enterprise networks, spread far and wide, to establish a covert observation post from which it's able to sneak around, steal and exfiltrate highly sensitive corporate secrets.

If you're a nation state like China, you couldn't care less about a ransom payment. What you want is the detailed production flow for a highly effective COVID-19 vaccine.