Cicada

Description: 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, and 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, multiyear (as in 11-year) advanced very-persistent threat state-based attack perpetrator known as "Cicada."

High quality (64 kbps) mp3 audio file URL: http://media.GRC.com/sn/SN-794.mp3
Quarter size (16 kbps) mp3 audio file URL: http://media.GRC.com/sn/sn-794-lq.mp3

SHOW TEASE: It's time for Security Now! with Steve Gibson. I'm Jason Howell, filling in for Leo this week. Steve's going to cover the many patches and updates that are hitting Chrome and Firefox right now, a whole bunch of news on that front. More attacks on WordPress. RCS finally gets end-to-end encryption; I'm pretty excited about that. And an 11-year threat called Cicada. That, and so much more, next with Steve Gibson, breaking it all down for you next on Security Now!.

JASON HOWELL: This is Security Now! with Steve Gibson, Episode 794, recorded Tuesday, November 24th, 2020: Cicada.

It's time for Security Now!, the show where we talk about all the latest security news with none other than Steve Gibson. I'm not Leo Laporte. I'm Jason Howell, filling in for Leo. But Steve, you are the man of the hour. How are you doing today?

Steve Gibson: Well, Jason, it's great to be with you once again. You are an experienced Security Now! cohost, so we know how this goes. You know how to do the show open and the close and so forth. Forgot to ask you, do we have three sponsors today as usual?

JASON: We do indeed. We have three. It's a jam-packed show. Yes, "experienced" in the sense that I can sit here and be amazed at your incredible security knowledge throughout the course of the entire show, along with everybody that's watching and listening. And then I throw in some ads from time to time. That's my experience in the world of security.

Steve: You're embarrassing me. We have Episode 794 for, what is this, oh, this is the last Tuesday of November. We're going to be in December next time. This episode is titled "Cicada," C-I-C-A-D-A, after the insect. Which is kind of appropriate. It turns out that's the name that's been given to a very pernicious, very advanced, very persistent threat actor, which we're going to talk about because Symantec stumbled upon them a

little over a year ago and has been watching what they've been doing. And just it's a different sort of profile than where we've been spending a lot of time talking, for example, about overt public attacks, especially those made by ransomware groups, which live to be public. We have a little bit of news about that, too.

But we've got a bunch of news on the browser front, both from Chrome and the Firefox projects - patches, updates, and features. A little bit of comical news, strangely enough, from the ransomware front. More troubling ongoing WordPress attack specifics. WordPress has really been under, as our listeners know, under attack a lot recently. We've got a weird ecommerce site spoofing attack. Also I want to take a look at the future consequences of ongoing vulnerability announcements, coupled with their very incomplete patching, which is a theme that we've been seeing.

We've also got - and I was thinking of you, Jason, because this is about Android - Google's just-announced bold move right into the middle of the unbreakable end-to-end encryption controversy with an upcoming update to Android Messages, which is now in beta. And I'm very impressed and pleased with the direction that Google has taken for implementing this. And then, as I said, we'll wrap up with a look at this large multiyear, like 11-year at least, advanced, very persistent threat, state-based attack perpetrator. And of course we always have a Picture of the Week. This one is a little late because it has a Halloween theme. But it was just too fun. So I thought, well, we'll pick it up anyway.

JASON: All right, Steve. I love this Picture of the Week that you selected. But tell us about it. I think it's great.

Steve: So it just - it popped up on Twitter. Thanks to one of my followers for posting it. It's, as I said, sort of a Halloween theme. It uses Venn diagrams to illuminate or illustrate various logical conjunctions. We have the first one is Trick OR Treat, showing the OR of the two Venns. And of course then we sort of have a carved pumpkin face there. And then Trick AND Treat, showing just where the two circles overlap, and so we've squeezed the pumpkin face in there. Then of course Trick XOR Treat, which is XOR of course being either of them, but not both, so that one is shown. And then for all of those three - OR, AND, and XOR - the second line is the logical inverse, the "not" of that. So NOR, NAND, and XNOR.

So anyway, just it wasn't specifically security related, but just a little fun thing for, as I said, a little late for Halloween, but still a nice picture for the podcast, a techie-oriented podcast.

JASON: A little late for Halloween, but still I'm sure someone out there is willing, like up for the challenge of making this an actual reality and actually carving a pumpkin in this way.

Steve: And we're also, you know, we're on this side of Thanksgiving still, so it's all sort of that holiday thing.

JASON: Totally.

Steve: You know, pumpkin pie of course is a big thing.

JASON: Pumpkins are still appropriate. It's cool.

Steve: Yeah, exactly. So Chrome has moved to release 87. In the process, it fixed 33 security vulnerabilities - 10 which were high severity, 11 medium, and two were rated low. And to provide some sense for the nature of the high-severity flaws which were

fixed, we had a use-after-free in payments, inappropriate implementation in filesystem, inappropriate implementation in cryptohome, a race condition in the ImageBurner, insufficient policy enforcement in networking. And that's - we're going to come back to that in a second because that struck me as an odd - "insufficient policy enforcement" was the term they used for something that we just talked about two weeks ago.

Also insufficient data validation in WASM, that's the WebAssem component. We had a use-after-free in PPAPI, a use-after-free in WebCodecs, a heap buffer overflow in the UI, a heap buffer overflow in clipboard management. Also use-after-free in Web Codecs was awarded a $15,000 bounty, which was split between its two researchers who discovered it. The other nine were all to-be-determined awards, which I thought was interesting. So all of those were regarded as high severity. And as we know, things like a heap buffer overflow in the UI, that's going to be exposed to the 'Net, potentially. Clipboard, same thing. Certainly WebCodecs are Internet-exposed.

So it's certainly - it's understandable that those 10 would be high severity. Looking through the list of 33, though, it wasn't clear what was going on with this assignment of bounties. And I don't know what the awarding logic is for those since most of the medium and low criticality vulnerabilities did have bounties set and awarded. And they ranged also between 7,500 on the high end down to $500 on the low end. So one thing is very clear, which is Google is not paying a fixed award amount for the bounties. They apparently really do look at what went into it, maybe what having the vulnerability repaired means in terms of enhanced security, and they pay accordingly.

But what was interesting, I mentioned one of those 10, that so-called "insufficient policy enforcement in networking." Well, that was referring to Samy Kamkar's CVE-2020-16022, which we talked about two weeks ago. That was the NAT slipstreaming vulnerability. And although I guess it can perhaps best be leveraged remotely with a browser - I'm sure it could best be leveraged with a browser, that's the obvious thing to do - it was never really the fault of any browser. And it was enabled when it was discovered by all of the browsers. So it wasn't Chrome's fault that they weren't already blocking SIP ports 5060 and 5061 from being targeted by browser code. And I don't know that I would even call that a critical vulnerability. I mean, yes, it was easily exploitable. But it was, you know, okay, sort of a theoretical issue.

Anyway, what we got last Tuesday with update 87 is now outgoing connections to remote SIP ports 5060 and 5061 are now blocked. And of course we can expect all other web browsers to be following suit before long. And there were a bunch of other things fixed. Google is keeping their details private, as they do, until the majority of users have updated their browsers. Which it's interesting, too, because I had just fired up a machine. I started Chrome, and I was still on 86 from the previous week. I had to go into About Chrome and sort of wake it up. And then it said, oh, hold on a second, and then it began, you know, I got a little spinning wheel, and I got updated to 87. But it's interesting that even starting a new instance, it's not like I had a system running and Chrome had been sitting there for days. It needed me to ask for it in order to get updated.

So anyway, so they're saying that they're not going to tell us anything more about what's going on until the majority of users are updated with the fix. Which I think, you know, in practice actually means until no one really cares anymore and stops asking because those are no longer newsy. And of course they're right. We really don't care. The details, I think, are useful for spotting sort of broader trends in what's going on and, as we often do hear, generating some conclusions about the nature of the problems that are being fixed and whether there's anything we can do moving forward to keep these things from continuing to happen. And certainly they do teach us like how to avoid such vulnerabilities. We've seen, for example, the generic problems with codecs being

interpreters of a compression technology. We're able to generalize the nature of these problems.

So I think that's where we are at this point in the industry. Somehow, and we'll come back to this a little bit later when we talk about some Windows problems, we really do need to stop all of this nonsense and start getting serious about figuring out how to not make these mistakes in the first place because, even though Chrome is really good about quickly responding to problems and fixing them, we'll be talking, as I said, a little bit later about that that's really a rarity in the industry.

And I know Android is where you spend a lot of your time, Jason. As we know, one third of Android devices are no longer, they're like pre-7.1. They're not getting updated any longer. And they're never going to be. They'll eventually die is the way those problems get resolved. In fact, it's very clear that's the only way that Windows 7 is going to stop being the second most popular desktop OS is that those Windows 7 machines will eventually just die. And then anything that replaces them will be Windows 10. Not that people wouldn't want to still have Windows 7, apparently; but you just won't be able to get it any longer.

So there were a handful of interesting new developer features for cookies and fonts added to Chrome with this release 87. And they continued their sort of weird deprecation in Chrome of FTP. Remember that when we went to Chrome 86, they only blocked FTP for 1% of Chrome's users. Now, of course, 1% of by far the most popular web browser on the planet, that's a lot of people. I guess they were just wanting to see, well, okay, let's see what happens if we turn off FTP, File Transfer Protocol, for 1% of our users. Well, apparently the world didn't end. Maybe no one even noticed.

So now, as of last week with 87, they're blocking half, just half, randomly chosen half of Chrome 87's users can't use FTP. I don't remember the last time I tried. And I think that's probably the case for virtually all of current web users. So their plan is for the January release of 88 to finally kill off FTP completely. There is a switch you can turn on, if you happened to be deprecated for FTP, and you need it for some reason. But, boy, just go get a client. Get an FTP client. You should not be using your browser for FTP, and before you long you won't be able to. So it's time.

They also, in 87, have some user-facing performance improvements. 87 now has what they call "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 currently visible. And the other new feature that has finally landed for Chrome in Android is the so-called "back-forward cache" or "bfcache." And I say to "finally land in Chrome" because both Firefox and Safari have had the equivalent of bfcache for years across both desktop and mobile platforms. The technology is a local cache which is optimized for the use of the back and forward arrows. When a page is initially loaded and set up, a lot of work goes into getting it ready, especially when there's a lot of JavaScript going on behind the scenes. We know that there's a JIT (Just In Time) compilation of JavaScript, and pages are not JavaScript "light" typically anymore. They are JavaScript "heavy."

Well, it turns out that Firefox and Safari have for years been very smart about holding a page's entire state, including its expensive-to-establish JavaScript state, such that when a user leaves a page, like by clicking on a link, if they should hit the back arrow to return to the page that they had previously left, that page can be instantly restored from a RAM-based history. Maybe it's because Chrome has been doing so much plumbing work on page operation with WebAssem and with a V8 engine, they just didn't get around to working on caching the state when a user left. Or maybe they just figured, well, we're going to be so quick to reestablish a state because we're so good that we're not going to take this approach.

Anyway, they've decided, no, we're going to use something called a "bfcache" in order basically to allow users to navigate backwards and forwards through a chain of pages and make that page-changing instantaneous. So it is here in Chrome 87 for Android, and mainstream desktop versions of Chrome are expected to be receiving it in the future. So I expect that Android users may notice a performance improvement. We're saying "instantaneous." I saw some reports that said a 20% improvement. So, okay, it probably just takes some amount of time just to switch the page, just to get it on the screen and show it to the user. Anyway, Safari and Firefox have been doing it for a while. Chrome 87 now adds it for Android.

During last June's Apple Worldwide 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 that apps collect from their users and what they're doing with it, how that data is being used to track users across apps. Those Apple labels that were announced in June are scheduled to go live next month on December 8th. And I bring this up because last Wednesday Google also announced that, starting on January 18th of next year, 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 are collecting and what they plan to do with that information.

After the 18th of January, a new Privacy Practices button will appear on each extension's Web Store listing. And to get ready for the change, Google has already added a new section to their Web Store Developer Dashboard to enable extension developers to indicate what they're collecting and why. So at this point, if we have any extension developers within earshot of the podcast, it would probably be a good thing, you've got, what, about a month and a half to go over, fill out that page for your web extensions so that when it appears on the user-facing side of the store January 18th, it'll be there. I imagine in the future it'll be a requirement moving forward, and just nice to see that we're getting this improvement in explicit privacy disclosure.

So that's Google stuff and Chrome stuff. Firefox 83 gets an HTTPS-Only Mode. It is currently disabled by default, but this happened last week. Security-conscious users using Firefox 83 or later will be able to go to, and here's where it is, in the URL about:preferences#privacy. So about:preferences takes you to a page with a bunch of different goodies, and then the #privacy takes you to the privacy tab. That's a long page. So you then need to scroll all the way down to the bottom.

When you get there, there's something new: HTTPS-Only Mode. And Firefox describes it as: "HTTPS, as we know, provides a secure encrypted connection between Firefox and the websites you visit," they write. "Most websites support HTTPS. And if HTTPS-Only Mode is enabled, then Firefox will upgrade all connections to HTTPS." So there's three settings. And actually I have a screenshot in the show notes, which is onscreen right now in the podcast, because that's what I immediately turned on, that is, "Enable HTTPS-Only Mode in all windows." The middle setting is "Enable HTTPS-Only Mode in private windows only." And then the default setting is "Don't enable HTTPS-Only Mode."

So while this is enabled, as it is for me, Firefox will attempt to connect to the HTTPS versions of websites when a non-explicit URL is provided. And in fact I've talked about this many times. It's like, it seemed quite a while ago browsers, for example, if you just put in GRC.com, browsers have for historical reasons tried, assumed that when the scheme HTTP whatever, S or not, was absent, to default to old school http://GRC.com. That has seemed to be wrong for a long time. That's like, we know, like the vast majority of sites have now switched to HTTPS. So it seemed to me that specifying nothing ought to imply HTTPS. And then maybe if that failed, then the browser should fall back to HTTP.

Anyway, what this change does is it finally changes that logic explicitly so that, if you don't specify, Firefox will go to https://. And now, with HTTPS-Only Mode, it will always

try HTTPS. And if that fails, you will then be given a dialog explaining that we were unable to get to this in HTTPS. You have manually enabled HTTPS-Only Mode, so you want to make an exception. Which is like, this is exactly what we would like to have happen. At this point, it seems to me they ought to just turn this on. And, if you go to a site where there's a probably, you can't get there with HTTPS, then you should explicitly say, okay, fine, let me through HTTP.

I did, however, immediately worry that one possible huge problem with doing this might be that it would force all connections also to the localhost to be HTTPS. This has been a concern for some time. Interestingly, it should not be a concern. It's sort of a moronic issue, that is, the idea of asking for a secure connection to your own machine. Localhost, as a lot of our listeners I'm sure know, is the technical term for 127.0.0.1, which is a subnet defined as your own machine, that is, the machine you're running on. It's the system's own local stack. And if anyone has ever done a netstat on a Unix machine, you know that it is 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 it's gotten very popular on Windows, too. And yes, Microsoft, it is possible for this IPC mechanism to be abused if something malicious were to somehow get into someone's machine. But in no way does localhost in and of itself create any vulnerability. Microsoft has, you know, Windows has never been a secure platform. It's never been secure against local attack.

I run with a nifty on-the-fly spellchecker 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. That's the nature of Windows. So if something evil gets into your machine, it's already game over. So protecting the browser from something that's set up shop on localhost is crazy. Yet despite that, Microsoft has been threatening for years to shut down its browser's local access to the localhost domain. And every time they try, the world of actual users explodes because doing so breaks too many things. Having localhost available is handy. It's reliable. And it's elegant. Which is why it's being used quietly by so many systems.

And I'll also note that SQRL is among them. SQRL clients run a little HTTP web server, and that's HTTP, not HTTPS, which listens on port 25519, for the browser's connection to the machine, which is a way that the authenticated URL is passed from the authenticating server to the SQRL client and then over to the browser to cut out any possible man-in-the-middle interception. And it's a way that SQRL provides a high level of anti-hacking provision. And localhost is just used widely. So there's zero need for TLS on the local stack. That makes no sense. If the IP connection is an abstraction, you are connecting to the operating system. And so there's no need for privacy, like encryption privacy, because nothing ever leaves the machine. And there's no need for authentication because it's the OS. It's the local systems.

Anyway, the good news is all of this, even though it hasn't really gotten through to Microsoft, this has already occurred to Mozilla and Google, and they're treating localhost as a presumed trusted exception to this HTTPS-Always rule. So they're not going to break anything, and they really have no choice. In addition to this cool HTTPS-Only Mode, this latest update also fixed 21 vulnerabilities in Firefox 83, including that bug that existed in, well, not in it anymore than it existed in Chrome. But its use of the FreeType library that had been used in attacks - you remember the zero-day attacks against Chrome that also leveraged a problem in Windows. We noted at the time that everything that used FreeType, which also allowed attacker-provided font glyphs, would be a potential target. So web browsers were perhaps the biggest target of all since they fit all the requirements out of the box and were inherently exposing themselves.

That heap buffer overflow in FreeType is now also closed in Firefox as it was closed in Chromium and Chrome and presumably in all the Chromium derivative browsers, as well. There was like a handful of other goodies that Firefox 83 brought to us. Mozilla wrote that Firefox, they said, keeps getting faster as a result of significant updates to SpiderMonkey, which is the JavaScript engine. They said we would experience improved page load performance by up to 15%, and page responsiveness by up to 12%, and reduced memory usage by up to 8%. They said: "We have replaced part of the JavaScript engine that helps to compile and display websites, improving security and maintainability of the engine at the same time."

They also said that pinch zooming will now be supported for Firefox users of Windows touchscreen devices and touchpads on Mac devices. They said Firefox users may now use pinch to zoom on touch-capable devices. Picture-in-Picture now supports keyboard shortcuts. Let's see, what else. Oh, and they said for the recently released Apple devices built with Apple Silicon CPUs, they said we can use Firefox 83 and future releases without any change. They said this release, 83, will support emulation under Apple's Rosetta 2 that ships with macOS Big Sur. And they said: "We're working toward Firefox being natively compiled for these CPUs in a future release." So that sounds like a good thing. And they just concluded saying it's a major release of WebRender as they roll out Firefox also to users on Windows 7 and 8, as well as macOS 10.12 and 10.15.

JASON: And we have just a little bit more in the realm of Firefox right now; right? Big Firefox news week.

Steve: Yeah, I appreciated ZDNet's headline. They said: "Fearing drama, Mozilla opens public consultation before worldwide Firefox DoH rollout." Of course, as we know, Mozilla wants to enable DNS over HTTPS, so-called DoH, in Firefox for all their users worldwide. Although maybe the U.K. will remain an exception. We'll have to see how that goes. But once burned, twice shy. Mozilla wants to creep forward a bit more cautiously this time, soliciting input from ISPs, governments, and companies before they flip that switch.

Last Thursday they initiated a period for public comment and consultation about the ways they could safely enable support for what was previously their controversial rollout of DNS-over-HTTPS. My reference to "once burned" was to the backlash of criticism they encountered, and our listeners will recall last year, mostly in the U.K. over their stated plan to support DoH inside Firefox. As we'll recall, U.K. government officials, law enforcement agencies, and even local Internet service providers, actually the service provider organization, criticized Mozilla using some quite over-the-top language, including declaring them the "Internet Villain of the Year," just because Mozilla wanted to protect their users from DNS games being played by those ISPs. Those opposed to this claimed that it would help bad guys bypass enterprise firewalls and parental control blacklists. You know, yes, villainy.

At the time, Mozilla chose to back off of their timeline as a result of all of this pushback, and agreed to delay deploying DoH inside the U.K. But they did deploy it, as we've talked about since, for all Firefox users in the U.S. And it's been under use at scale since earlier this year, like since February. And Mozilla has always planned to roll out DOH to all of its users across the world, although they may still hold off in the U.K. because they have promised not to deploy it there after their first attempt. So now the current consultation period exists as a way to give what they're calling stakeholders, you know, governments and ISPs, an opportunity to speak up or forever hold their peace.

This consultation period runs from last Thursday through January 4th of 2021. So several months now to allow them, anybody who's opposed, to say, okay, look, this is what we want you to do. Mozilla has promised to consider every reasonable and practical suggestion, whatever it might be. But Mozilla has already addressed most of the DoH criticism that they received. They have added a so-called "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. So that problem has been solved.

A lot of enterprise users were upset because they did not want the Firefox instances within their network to be setting up an encrypted tunnel that they didn't have any oversight over. So now there is a mechanism, well established, for Firefox to back off of that in enterprise environments. And also they've added support for more providers than just Cloudflare. Again, at the first announcement there was this concern over the development of a monoculture, that everybody, all Mozilla users would be using Cloudflare as the sole provider. Now you've got multiple choices. And so Apple, Google, and Microsoft have also announced plans to support DoH protocol. And having watched Mozilla stumble last year, they've all deployed enterprise-friendly DoH implementations from the start. And of course Google's DoH support in Chrome has already gone live.

So anyway, it's like this kind of change upsets people. It gives users more privacy and security at the cost of some other entities that may have been enjoying the ability to monitor DNS queries and see what users were doing. That's not going to work in the long term. And ISPs have also, as we expected, deployed their own DoH servers so they can say hey, you know, you're able to use DoH if you want to, and still stick with your own ISP. So that's been a good thing.

I had an interesting little bit of 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, which now announces 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 being public. The enterprises will just claim like a generic network outage. And oftentimes we're left guessing. You know, the news reporting media is like, well, was that a ransomware attack, or did somebody trip over a cord? What happened? And in sufficiently large organizations, the cover-up of what actually happened typically is extended to include their own employees for fear of a news leak tanking their stock price and incurring their reputation damage.

So now, in a move clearly designed to increase the pressure and increase the likelihood of just exactly that kind of public leak, after an attack, the Egregor operation locates all of the available hard copy printing devices within a network and uses them to repeatedly print the announcement of their successful attack and ransom demands, for any employee to see on a printer. They're announcing that this enterprise and its network has been taken over by ransomware, and here's the demand letter, and please send your bitcoin payment to the following address. So, yeah. That's happening now, too. And if it ends up being an effective tactic, you can imagine that the other ransomware gangs will adopt it before long.

The Wordfence WordPress web application firewall company posted news of the latest in their ongoing and escalating attack on WordPress sites. Their posting was titled "Large-Scale Attacks Target Epsilon Framework Themes." 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" - that is, WordPress themes - "using the Epsilon Framework." They said they estimate there are over 150,000 sites using this framework.

They said: "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." They said: "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."

In their announcement they then proceed to list the 15 WordPress Epsilon Framework themes that are known vulnerable to these attacks. And at this point the attacks appear to only be probing, looking for vulnerable WordPress instances. Presumably what's happening is they are accruing a master list of vulnerable targets. And this makes sense since the use of 18,000 source IPs sounds, well, it must be 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, you know, HTTP POST queries to admin-ajax.php; and, as a consequence, do not leave distinct log entries.

They've also confirmed that an exploit chain does exist to permit full remote code execution enabling a full-site takeover. So obviously someone else is aware of this, as well: 7.5 million attacks against 1.5 million sites targeting the known install base of 150,000 sites where these vulnerable themes are being used. So, boy, I mean, again, just another big problem for WordPress. They have confirmed that a full exploit chain exists. So this is going to end up in site takeovers as soon as they switch from aggregating their targets and probing to attack.

Jayant Shukla, who's the CTO and cofounder of K2 Cyber Security, told Threatpost in an email, he said: "WordPress" - and we know this - "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 ecommerce sites," which we're going to get to in a second, in the next story here. He said: "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." And that's of course not surprising.

"As we know from past research," he said, "as many as 60% of successful attacks are on vulnerabilities that already have a patch to prevent its exploit. Organizations," he said, "need to take the security of their WordPress sites more seriously, starting with keeping the plugins and software up-to-date and patched." So, yeah, no surprise there. No one listening to this podcast doubts that for a moment. But we also know, apparently, how much easier that is said than done.

Okay. So get this next issue. A new cybercrime gang has been found to be taking over vulnerable WordPress sites to install hidden ecommerce stores with the purpose of hijacking the original site's search engine rank and reputation, and promoting online scams. 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. And we've looked at some of Larry's work before. 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 to the end of that index.

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 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 command-and-control server. If a user met certain criteria, and it was unclear what that criteria would be, but I have a hunch, the command-and-control server would instruct the original WordPress site to reply to the user with an HTML file containing an online store, which was pedaling a wide variety of mundane objects.

Presumably, this criteria would be designed to avoid detection by the site's actual admin and owner. You know, if you went to your own site and saw some weird ecommerce store instead of what you expected, that's not what you would, you know, you would obviously immediately know you had been taken over, compromised, and you'd go about

fixing that. So the point is that some subset of visitors, instead of getting the site they expected, would get an ecommerce site with what were described as "mundane objects" for sale. And presumably, if you actually ordered something, that must be the nature of this scam. It's not like you're going to get your order fulfilled; right? They would take your money and say thank you very much, and you would never get your KN95 masks or whatever was on sale.

Larry, the Akamai researcher, indicated that - get this. During the time the hackers had access to his honeypot, which he set up, they hosted more than 7,000 ecommerce stores. So what must be going on is that, based upon where you're coming from, like where you're located geographically based upon your IP address, they on-the-fly select the store for you to be displayed at this hacked WordPress presence. 7,000 ecommerce stores that they intended to serve to incoming visitors. Which suggests that, as weird as this seems, it's no small operation. It would take some effort to set up 7,000 different ecommerce presences. And clearly it's intended to be a long-lived scam across a large base of WordPress sites.

Now, if all of this seems odd, like okay, this just seems bizarre, here's what's going on. In addition, the hackers generated XML sitemaps for the hacked WordPress sites that contained entries for the fake online stores mixed in with the site's authentic pages. And the attackers, after generating the sitemaps, submitted them to Google's search engine, then deleted the sitemap to avoid its detection. So although this whole procedure seems harmless, it's clear that what they were going for in hacking WordPress sites with good search engine rankings was they were trying to leverage the ranking of the site in order to get this collection of bogus ecommerce sites visited.

And of course the problem is that by doing this to the site, they dramatically reduced the site's search engine ranking over time because, you know, basically it became a scammy spammy site that Google was not going to rank highly in the long term. So this just seems weird and bizarre. But it's what's going on because WordPress has pretty much gotten out of control from a security standpoint.

Okay. So we have to talk about the fact that Windows and arguably some other high-profile services on the 'Net are just not being patched over the long term. Here we are today, more than a year downstream of the BlueKeep RDP bug. Remember the BlueKeep was described as a "preauthorization attack," meaning you didn't need authorization in order to attack an instance of remote desktop protocol. Today, more than 245,000 Windows systems are still vulnerable to BlueKeep's complete authorization bypass on remote desktop protocol - 245,000 instances. It's clearly the case that overall the biggest scandal on the security side of the IT industry today has to be the lack of timely or ever patching of highly public widespread remotely accessible vulnerabilities in our computer networks.

We will wrap up today's podcast by looking at some details of a group known by many names, one of which is APT10, which is, you know, they're a well-known Advanced Persistent Threat group. Those details are interesting and sobering. But the question of how such threat actors gain access to their victims is hardly a question in a world where, after more than a year, nearly a quarter million servers remain vulnerable to a now well-known, easily exploited problem.

I looked at a chart of BlueKeep, and I had to do a double-take. It's onscreen right now, on the video podcast. This chart is a timeline of exactly one year, from November 15th of 2019 to November 15th of 2020, basically last week. During that time, the number of open port 3389 dropped from 9% to just 6%. Okay, now, in fairness, this was COVID-19 year; right? So the work from home response to the novel coronavirus did put a sudden upward pressure on the whole terrain of remote access.

And we can clearly see its effect in the chart's rise through March and April. That is, the incidence of open Port 3389 was dropping. Then it kind of leveled off. Then it went back up in March and April, and then it's been sort of declining, although it hit another little bump in August. But still, we're talking about we're now at just shy of 6% of all IPs, 4.3 billion IPs; 6% of the Internet's IPs have 3389 open on them, which is just stunning to me. We're now at 25% of the original 950,000 Windows machines that were initially discovered to be vulnerable to BlueKeep during that first scan in May of 2019. So we were just shy of a million then. Today we are just shy of a quarter million, nearly a year and a half later.

Okay. Which, again, this is a problem for the security industry. Today, in addition, separately, more than 103,000 Windows systems also remain vulnerable to the SMB Ghost vulnerability. That's the, remember, Server Message Block v3 protocol problem that was discovered and patched back in March of this year. Yet here we are in November, and 103,000 Windows machines still have SMB exposed, and they're still vulnerable to this exploit. Either of these vulnerabilities allow attackers to take over Windows systems remotely and are considered some of the most severe bugs discovered and having been patched in Windows over the last few years.

But despite their severity, an incredible number of Windows systems have remained unpatched and are thus vulnerable to remote takeover. So is it any wonder that I finally had to stop, like, enumerating each week's ransomware attacks, for fear of boring our listeners? I mean, it's like, okay, yeah, fine. Here's who's been attacked in the last week. It's just it's crazy how many systems are vulnerable.

And in fairness, it's not just Windows. A Czech researcher just compiled a list of outstanding vulnerabilities when they were discovered and fixed, what they were in, how many systems are unpatched against it, and the CVSS vulnerability score. I've got the table in the show notes. For example, Apache web server has a CVE back from 2019, it's 0211. At the moment 3.357 million Apache web servers remain unpatched. It's not super critical. It's got a CVSS of 7.8. But still, Apache web server. Update Apache web server. But no.

Squid, also a CVE from 2019, 1.2 million unpatched systems. That one's 9.8 severity. Microsoft IIS, 374,000. That's got a CVSS of 10 out of 10. Also, okay, get this one. This has a CVE from 2015, 2015-1635, affecting Microsoft IIS. Again, 374,000 with a severity of 10. We know the Exim, the mail server, has been having problems. Two of them, both from 2019, in one case 268,000, another one 264,000, both with CVSSes of 9.8. We talked about that one previously. Of course BlueKeep, we were just talking about the Windows RDP problem, 246,000. That's got a CVSS of 9.8.

Heartbleed. Remember Heartbleed? That was 2014. Guess how many OpenSSL instances are still online, susceptible to Heartbleed? More than 200,000 - 204,878. Now, yes, not readily exploitable. We know it's very difficult to exploit. You can, if you succeed, you can potentially obtain useful private keys. So it's got a CVSS of only 7.5. But still, it's six years ago, and 200,000 OpenSSL instances are still there with that. SMB Ghost we were just talking about, the Windows SMB problem, 103,000, CVSS 10. Top of the charts, 10. WordPress, there's a problem from last year, nearly 84,000 instances out there with a CVSS of 8.8. ProFTPD, 80,000, with a severity of 9.8. And one other Exim from two years ago, from 2018, 76,000, more than 76,000 instances with a CVSS of 9.8.

So the point is this is serious. This is really a problem. And it's possible to say, okay, yeah, that's not good. But that's other people's machines and other people's networks. But it's important to remain cognizant of the fact that there is a potential for cross-network contagion. We've seen this earlier in the managed service providers where, for example, an MSP gets themselves infected, and the bad guys realize that they've hit the mother lode because now they have access to all the MSP clients' networks.

Remember when all of those dental services were getting hit with ransomware because they were all, all of these dental services had outsourced some aspect, and I think it was their health records management, to some managed service provider that was providing that as an outsourced service. And suddenly all of those dentist offices got zapped because a common single managed service provider got themselves compromised. 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.

So I would argue that 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 connected to are also similarly taking their own security seriously. Now, when queried, they'll all profess to be totally safe and secure. Who wouldn't? But that means that most, if not every one of those companies represented in that table above would proclaim exactly the same thing, despite the fact that they're sitting there with systems that haven't been patched for years.

And Microsoft vividly demonstrates for us every single month that things are not getting any better. It's not as if the critical buggy code problem has been solved. So Microsoft, just Microsoft is leaving a lengthening trail of critical vulnerabilities in their wake. These things are not getting fixed, yet more and more of them are being revealed every month. Some percentage of each of them endures over time, creating this long tail, potentially a never-ending tail of trouble. All the evidence shows that it's not until those machines eventually die that these problems are going to get patched. That table above just proves it. So maybe our listeners are right: Three digits is not enough for this podcast.

JASON: Now, before we get to the main event, this last story, I'm excited to see this news. I was wondering if it was going to happen because I remember when Google initially launched RCS, it seemed like end-to-end encryption might not actually happen, like, oh, this is not part of this. But the news sounds like is sounding positive for that.

Steve: Yeah. And the way they did it is just exactly right. The listeners of this podcast know that end-to-end encryption is politically charged and a super hot topic. The question is, how do we resolve this fundamental tension between individuals' desire and right to privacy, and law enforcement's, depending upon what country you're in. In the U.S. we have a Constitution that protects us from unwarranted search and seizure. But if you get a court order from a judge, then law enforcement has a warrant in order to search, the argument being that that's for the general public betterment.

Anyway, the problem, of course, happens when you have encryption that cannot be cracked. So as we know, exactly as you were saying, Jason, traditionally Google has been using SMS. And back in '07, so, what, 13 years ago, the replacement for it was technically designed, finished, and ratified - and we talked about it way back then - known as RCS, Rich Communication Services. It's an open industry standard.

I thought it would be good to give it sort of a setting. So I snipped the first paragraph from Wikipedia. Wikipedia says: "Rich Communication Services is a communications protocol between mobile telephone carriers and between phone and carrier, aiming at replacing SMS messages with a text message system that is richer, provides phone book polling for service discovery, and can transmit in-call multimedia. It is part of a broader IP multimedia subsystem." They said: "It's also marketed as Advanced Messaging, Chat, joyn" - spelled J-O-Y-N - "SMSoIP, Message+, and SMS+." They said: "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 next year, 2021."

So anyway, despite its specification and ratification way back in '07, the adoption of RCS has been somewhat lackluster. But, you know, it does offer a number of new and useful features. You can get typing indicators, meaning you know when someone at the other end is typing. You know, iMessage has that, and that's sometimes interesting. Presence information, location sharing, longer messages, and better media support. So you get better quality photos and videos, chat over WiFi, knowing when a message has been read, sharing reactions, and better capabilities for group chats. So it's like, you know, it's a set of capabilities whose time has certainly come.

And what's most significant, I think, from a political standpoint, is that this isn't all by virtue of some add-on. It will be in the base Android OS. And that's significant. Last Thursday, Google said that they've completed their worldwide rollout of RCS and are moving into a new phase. And here it comes: adding native end-to-end encryption. So Android's native messaging platform would potentially be able to offer the privacy and authentication features that we're all familiar with, from like Threema and Signal and WhatsApp and so on. At the moment, end-to-end encryption in Android Messages, which is the Android app, 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. Their 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. At this point, end-to-end encryption has been added to the solved problems list, thus there is no need to do it again. Google has adopted the very well-designed and already well time-tested Signal protocol. So that just could not be cooler. This means that Signal goes mainstream in Android moving forward. And this is a big deal, and a big day, I would argue, for end-to-end encryption. Essentially it means we have Signal for RCS built into Android's native Messages app.

Google said: "Eligible conversations will automatically upgrade to be end-to-end encrypted." Which means, as this happens, as existing devices update to having the latest Android Messages app, and certainly for all new Android devices moving forward, you just get automatic end-to-end encryption. And given Android's spread and reach, they're currently three quarters of the entire mobile OS platform globally, with iOS being the other one quarter, and a few other also-ran OS platforms. And given that no third-party app will be needed for automatic unbreakable encryption to be provided to its users in the future, there's no way to see this other than 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 has just been upped.

So I just think it's very cool. Not only that Messages now is fully onboard with RCS, but the idea that it just bundles Signal in. Boy, I mean, law enforcement and governments keep grumbling about end-to-end encryption. It is a problem without a solution. There is no way we know for there to be some sort of weakening that allows for selective encryption without breaking the value of encryption. So I don't know how we're going to solve this problem. I just - I don't think we do. I think you get access to the data before it goes into the encrypted tunnel or after it comes out at the other end. But you just have to give up on decrypting it in any way in transit. And of course there's also the "data at rest" encryption problem of these devices really being resistant to anyone decrypting them without the unlocking key. So yay for Google. We'll see what this creates.

Okay. So Cicada. It was the group's use of the Zerologon vulnerability that first brought them to my 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 - we know that's APT - group known by several names: Cicada, APT10, Stone Panda, and Cloud Hopper. They've been involved in cyberespionage operations since at least 2009. Their cyber intrusions are generally aimed

at large international Japanese companies with attack presence, having been spotted in 17 geographical regions and across multiple business sectors.

And it's not just Japanese companies because affiliates or subcontractors which may well not be Japanese, but are connected to Japanese parents, are also targets. They're an advanced persistent threat because they burrow into a large company's network and remain in place performing long-term intelligence gathering. The business sectors they focus upon seem to primarily be automotive, pharmaceutical, and engineering sectors, as well as managed service providers. The group uses what has become known as "living off the land" tools, meaning using what's available on a network. And they do have custom malware which has been observed in their attack campaigns, including a custom malware that Symantec has named "Backdoor.Hartip," H-A-R-T-I-P. They have not seen it being used by the group previously, but now they're seeing it everywhere, so it's a recent addition to their arsenal.

Among the machines compromised during the 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 a newer technique known as "DLL side-loading" in the campaigns that Symantec has observed. And they were also seen leveraging the Zerologon vulnerability that was patched in August of 2020. So as we know, a few months ago it's been patched. But lots of corporations haven't updated their internal machines. Zerologon is beautifully suited for moving laterally within an organization, and we recently did a deep dive into one of the ransomware attacks which exactly and specifically use Zerologon for that purpose.

This 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 that application is run. DLL side-loading is becoming an increasingly popular means for sneaking malicious code inside of Windows operating processes, getting it into RAM. And it's very effective in avoiding antiviral systems.

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 customers' networks. That triggered an alert in Symantec's Cloud Analytics technology which is part of their endpoint security, the Symantec Endpoint Security Offering. The activity was brought to the attention of their analysts to verify that it wasn't a false positive. Then it was passed up to their investigations team for further analysis.

Since Symantec's instrumentation is widely spread throughout the large customer base, once the so-called IOC (Indication of Compromise) was identified, they were then able to apply that across their entire coverage space to identify other previously unknown victim networks and enterprises and begin through the visibility that they had to get some sense for just how pernicious and widespread this thing was. Also, once something like this is discovered, it's possible to scroll back through prior activity logs to reexamine previous behavior that wasn't appreciated for what it was at the time, and to carefully examine the timestamps on files that they now know to look for, which were not previously suspect. This allows them 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 of 2019 and up through the beginning of last month, October of 2020. The attacking group has been active on the networks of some of its victims for a full year. The campaign is very wide-ranging, with victims in a large number of regions worldwide. They have found compromised corporate networks in the U.S., U.K., France, Belgium, Germany, the UAE, India, China, Hong Kong, Singapore, Vietnam, the Philippines, Taiwan, South Korea, and Japan.

So this one group is widespread. And although it's unusual to see a Chinese government-linked group attacking companies within their own borders, that is, you noted that China was among those that I cited, many of the companies have Chinese subsidiaries of a larger Japanese organization. So that explains why they got swept up in this. It's because, yup, they're dealing with a Japanese organization. And it may be a way of getting into the larger organization. Here again, I was talking about having to trust all the companies that you are connected to.

Anyway, Symantec's research discovered similar DLL side-loading malware on every victim network. And as I noted, the attacks employed a wide variety of the so-called "living off the land" dual-use and publicly available tools and techniques. They use RAR for archiving. Files are transferred to staging servers and RARed before exfiltration. They are often encrypted and compressed, making them easier to extract and not set off any IDS systems, Intrusion Detection Systems.

A certutil is a command line utility in Windows that can be exploited and used for various malicious purposes such as to decode information, to download files, and to install browser root certs. So again, living off the land, using things that are already present for malicious purposes. And then there's ADfind, which is the command-line tool that can be used to perform Active Directory queries. Csvde can be used to extra Active Directory files and data.

Ntdsutil can be used as a credential-dumping tool. And WMIExec is used for lateral movement within a network and to execute commands remotely on other systems. And of course we have PowerShell, bringing power not only to admins, but also to attackers. And they upload their ill-gotten goods to legitimate Internet cloud file hosting services for exfiltration. This, of course, makes sense, since such services are likely not to be blocked, and they're accessible for local use.

Observing the activity patterns, the amount of time attackers spent on specific networks varied widely, with the attacker spending a significant amount of time, in some cases being active for months on some networks of some victims, sometimes just a few days on other victim networks. And in some cases the attackers would spend some time on a network, probably looking around to see what they've got, then would go away, having their attention directed elsewhere, only to resume activity on one of those networks at a later time.

So it's just, if we step back for a moment to switch our perspective away from ransomware to the nature of this or think about it, it's a bit astonishing and sobering. Here we have a Chinese state-sponsored entity, a group, who have - and we didn't get from Symantec a sense of number, how many organizations this is. But it's extensive. They have infiltrated these networks. They've established a long-term ongoing persistence in their networks. They are able to connect anytime they want, move around hidden, unseen for years, exfiltrating what they're interested in.

It just seems to me, it's clear to us, that this one operation is not the only such thing that is going on. We know with absolute certainty, thanks to WikiLeaks and Edward Snowden, that U.S.-based and thus inherently state-sponsored intelligence services in the U.S. have developed and are developing powerful tools for doing much the same thing. We tend to focus upon the big, splashy, oh-my-god ransomware attacks which make headlines. We were just talking about how now they're taking out ads on Facebook, in one case exposing Matthew McConaughey's private contracts, and now spitting out ransom demands on all of a company's printers. But there's another very different, entirely covert, and I would argue far more insidious reality that goes largely unremarked because it doesn't want to be found, ever. It wants to burrow into many enterprise networks, spread far and wide, establish covert observation posts from which they're able to sneak around, jump between networks. If that company is connected to

somebody else, whoops, well, that's how they get into somebody else's network - steal and exfiltrate highly sensitive corporate secrets.

And you know, 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. And this is the way those sorts of secrets escape. And we have heard our own domestic health vaccine researchers saying that China and Russia have vaccines that bear the signatures of U.S. work. So this is not just fiction and fantasy. This is really going on. It's the world that we're in today, and it is quite sobering.

JASON: Well, this year, especially, has been quite sobering, in combination with all of this, all of the myriad different directions that this stuff is going. And this is just one aspect of that.

Steve: Yup. So you have the big flashy ransomware attack, and then the flipside is APTs, Advanced Persistent Threats. These things burrow in, they set up shop, and they stay as long as they can. Wow.

JASON: They get comfortable. They take off their shoes for a while. They make themselves at home.

Steve: They're cicadas, yeah.

JASON: Exactly. That's how they roll in 2020. Steve, thank you so much. Awesome stuff, as always. Definitely want to remind people to go to GRC.com, where you can find everything you need to know about what Steve is working on, what he's writing about, podcasting about. Everything is there. SpinRite, of course, is there, Steve's hard drive recovery and maintenance tool. You can actually get your copy there, also information about SQRL, of course. You can find audio and video of this show. Transcripts, that's the only place that you can find transcripts of Security Now! is at GRC.com. So make sure and go there and check it out.

If you want to find this show on our site, you can do that, as well: TWiT.tv/sn for Security Now!. You go there, you'll find all the ways that you can subscribe to this show, audio and video formats, a link out to YouTube if that's where you like to watch the show. Everything is there that you need to know, including the information about when we record. We record live every Tuesday, starting at around 1:30 p.m. Pacific, 4:30 p.m. Eastern, 21:30 UTC. And so you can go to TWiT.tv/live, actually, if you want to be part of the live show and see it recorded in real-time. But we appreciate it when you subscribe. That's the most important thing. Subscribe, and you won't miss a single episode with Steve, and also with Leo when he's back next week.

Steve, thank you so much for another episode of Security Now!. Always appreciated. And thanks for inviting me to be here alongside you today.

Steve: Thanks for holding the fort down, Jason. It was a pleasure working with you. And till next time.

JASON: All right, until next time. We'll see you then on Security Now!.

Steve: Bye.