SolarWinds

Description: This week is crammed with news leading up to our holiday break. Chrome is throttling ads. There's new cross-browser as insertion malware. We have a new term in the ransomware world. We have last week's Patch Tuesday, a jaw-dropping policy leak from Microsoft, trouble for Cisco's Jabber, an embarrassing vulnerability in many D-Link VPN servers, the brief Google outage, more horrific news of IoT network stack vulnerabilities, another WordPress mess, the 2020 Pwnie Awards, the welcome end-of-life of Flash, JavaScript's 25th birthday and free instruction classes, a bit of closing the loop, and SpinRite news. Then we take a full reconnaissance dive into what happened with the monumental and in so many ways horrific SolarWinds supply chain security breach.

High quality (64 kbps) mp3 audio file URL: http://media.GRC.com/sn/SN-797.mp3
Quarter size (16 kbps) mp3 audio file URL: http://media.GRC.com/sn/sn-797-lq.mp3

SHOW TEASE: It's time for Security Now!. Steve Gibson is here. We're going to talk about Patch Tuesday, a retrospective; and a new classification system Microsoft's using that might be just a little bit, oh, I don't know, self-serving? The Pwnie Awards are in. I don't know if this is good or bad. And then Steve's going to break down this massive hack from Russia, the SolarWinds attack, what we know so far. Coming up next on Security Now!.

Leo Laporte: This is Security Now! with Steve Gibson, Episode 797, recorded Tuesday, December 15th, 2020: SolarWinds.

It's time for Security Now!, the show where we protect your security and privacy online, holiday edition. Steve Gibson is here. Hello, Steve.

Steve Gibson: Yo, Leo. Great to be with you for, well, this is not the last podcast of the year. We're actually...

Leo: Almost. We got one more.

Steve: We're off next week, but we're back the podcast before New Year's.

Leo: And you are kept warm today by the MITS Altair.

Steve: Ah, very nice.

Leo: It's an Altair 8800. This is that kit that you ordered, and we ordered. It's the Altair-Duino by Chris Davis. And it's a really great simulation running, oddly enough, on an Arduino, which is probably faster than the original Altair was. But thank you to Burke McQuinn, who is a master with a soldering iron and was able to solder this hundreds of little tiny pieces together to make that. It's beautiful.

Steve: Very cool.

Leo: It's a nice replica. Did you use the original Altair ever?

Steve: No. I was...

Leo: You had mini computers. You didn't need a micro.

Steve: That was after, yeah, after the DEC PDP stuff. And I think I was busy. I was over on the Apple side, although I did end up using all of the software for the light pen, the LPS II for the Apple, I wrote under CPM, so on a CPM-based machine, with a cross-assembler, which cross-assembled...

Leo: Holy cow.

Steve: ...from that over to the 6502.

Leo: Did you not have an Apple that you could do that with?

Steve: Oh, no, I absolutely did.

Leo: You just liked the tooling on CPM?

Steve: Yeah, exactly. The tooling, the editor, the compilers. I had a hard drive, the Corvus hard drive.

Leo: Oh, I remember that, yeah.

Steve: Yeah. And so there was, like, none of that was ready or really there yet over on the Apple side because CPM was, you know, we had the S100 bus, and it was a very mature system. And actually I did the programming on - it was called, I think it was called a SoftCard, an Apple II SoftCard.

Leo: Oh, that's right, that's right.

Steve: That had an 8088 and allowed you to run CPM on your - and sort of like dual boot, run CPM on your Apple.

Leo: Your Apple, yeah.

Steve: So you used the motherboard, the power supply, the keyboard monitors and things; but it was actually a CPM environment. So, yeah, it was a - I miss those days where, you know, you had some idea what was going on with your computer. I tried bringing up the Blob Opera on Chrome, and it just like literally brought my system to its knees. I don't have any fancy soundcard or anything about sound, and it did mention at the bottom that it needs some advanced sound stuff. I'll try it on Windows 10 tonight. I'm on a Win7 machine. So maybe Chrome with Win7 it doesn't like, or maybe it's some conflict. Anyway, my point is that things have gotten so far advanced from where we were back then.

Leo: Well, this thing is running an Arduino, but it says it's about the same speed as the original, emulating 64K of RAM, which would have been horrifically expensive back in the Altair day. Can emulate an i8080 or a Z80. Z80 runs at a reduced clock speed of 2.6 MHz. It's got Altair extended BASIC. I remember it was Bill Gates who wrote the original BASIC for Altair. That's what got Microsoft all started.

Steve: Yeah.

Leo: It's kind of cool. Kind of amazing. And it was only a couple hundred bucks and about $3,000 worth of Burke's time. And so...

Steve: Still a bargain.

Leo: Yes, still a bargain. What are we talking about today? I can guess, but I'm curious.

Steve: We're absolutely going to talk about the big, I mean, the sort of "monumental" I don't think is too big a word, in fact it's the word I used in the show notes, this SolarWinds attack, basically a supply chain attack. My original working title for the podcast was Supply Chain Risks because there was one other one also. We have more problems in TCP/IP stacks for IoT devices. But this week is crammed with news leading up to, as I mentioned, to our holiday break next week, although we'll have a best-of show on.

We've got Chrome throttling ads. We've got a new cross-browser ad insertion malware. A new term being used now in the ransomware world. We've got last week's Patch Tuesday retrospective, a jaw-dropping policy change/leak, I guess, from Microsoft, that I'll be anxious to have you ask Paul and Mary Jo about. We've got trouble for Cisco's Jabber; an embarrassing vulnerability in many D-Link VPN routers. Just a quick note about the brief Google outage that happened, I guess it was yesterday, early morning Pacific time. More horrific news of IoT network stack vulnerabilities, as I referred to. Another WordPress mess. The 2020 Pwnie Awards.

The welcome end-of-life of Flash, oh my god it's finally here. JavaScript's 25th birthday with some free instruction classes available, one per week throughout the rest of the year. I wanted just to put it on our listeners' radar because JavaScript is the language now. We have a bit of closing-the-loop and SpinRite news. And then we're going to take what I called in my notes a full reconnaissance dive into what happened with this monumental and in so many ways horrific SolarWinds supply chain security breach where we still don't know everybody who was affected because mostly probably because they don't want us to know.

Leo: Yeah. Lot of companies use this SolarWinds package.

Steve: Yes. They've confirmed that 30,000 of their 300,000 customers had it, and that it affected at least 18,000 of those customers.

Leo: Wow.

Steve: And among them is like the Who's Who of government and commercial business. So anyway, we're going to get to that. We've got a fun Picture of the Week. So I think a great podcast, jam-packed for our listeners.

Leo: Jam-packed. Full of good stuff. All right. I was hoping you would do the SolarWinds thing. I'm really interested in what happened there. It looks like that might be the most significant hack in a long time. And we may never know.

Steve: Yeah, that's how it's being regarded. And the way it was found, how it might have been missed, how long it's been there, what it is, and the interesting macro key that they've got, macro keyboard combination that they've got programmed at the Russian Embassy, where they just hit a particular key, and it spits something out. So anyway, lots of fun to talk...

Leo: In Soviet Union, keyboard types for you. Picture of the Week, Mr. Gibson.

Steve: So this one, I don't know, maybe it's tangentially related to security. But I found it in my Twitter feed. One of our listeners sent it to me. And I showed it to Lorrie, and she thought it was just adorable. So I thought, oh, okay, fine. So it's been in the stack, and it has popped to the top of the stack.

Leo: It's cute. It's kind of a dad joke, but it's cute.

Steve: Yeah. So it's a four-frame cartoon. The first frame is this odd sort of thing, you're not sure what you're looking at, but it looks like something stacked in a hat and coat. Oh, and the sleeve is not fully filled. And this thing is at a ticket counter, says, "One adult ticket, please." And the ticket guy frowns and says, "I can tell you're three sheep in a trenchcoat." And the top sheep of the stack says, "Are you sure?" And the ticket guy says, "Yes. Look. One, two, three." And then in the next frame...

Leo: [Snoring]

Steve: ...he's fallen asleep because of course he was counting sheep, so that's going to happen. And our strange stack of three sheep has decided to go into the movie and not be harassed by the ticket agent. So anyway, like I said, I'm not sure what that's about, like why that's on the podcast, but it got Lorrie's endorsement, so I thought, what the heck.

Leo: Identity and access management, right there in a nutshell.

Steve: Ah. There it is.

Leo: See?

Steve: Thank you, Leo. I needed to put a caption on the picture, yeah. And spoofing your identity.

Leo: Yes.

Steve: So we talked about this approaching some time ago. And I remember it was interesting because of the heuristics that the browser, Chrome, was going to be applying. It has begun to roll out and has been spotted in the wild. And that's the so-called "Heavy Ad Intervention" which is now present in all of our Chromes, since we're all on 87. So it affects both, nicely, third-party ads and Google's own AdSense, so they're not biasing, I mean, they'd probably get in trouble if they were. They're not biasing ad treatment. It's being, as are many of these things Chrome does, rolled out gradually.

I think we're, what, are we at the 50% FTP mark? Or maybe we're at the - I've forgotten where we left off. But they're doing the same thing with FTP. It's like, okay, now these people don't get to use it. Now these people don't get to use it. Now almost no one gets to use it, like they're waiting for someone to complain. Well, I don't think anybody's going to complain if heavy ads, that is, ads which Chrome decides are abusing their users' experience are blocked.

But I definitely wanted mine turned on. And so I know that a lot of our users will, although already I'm running with uBlock Origin, which makes my experience so much more tolerable than what most people see. But you can go to chrome://flags. And if you put into the search term, if you put in "heavy ad," that will return two choices. The first one is Heavy Ad Intervention, and the second one is Heavy Ad Privacy Mitigations. So I enabled the first one.

They were both set to default, which gives Chrome control over this stochastic process of deciding who's going to get it and who isn't. Anyway, mine's enabled now. And the Privacy Mitigations is disabled. What I'm assuming that does, the description says: "Enables privacy mitigations for the heavy ad insertion. Disabling this makes the intervention deterministic. Defaults to enabled." That's, you know, someday. So what that must be is they recognize that it would be possible for someone to deliberately create a false-positive heavy ad in order to detect whether your browser was deterministically blocking that.

So in other words, it would be one bit worth of tracking signal. And so it's just sort of refreshing to see that the world is like so worked up now over the issue of privacy that even something like this, where it's like, yes, I want heavy ad mitigation, darn it, they're like, well, yeah. But you know, if you say that, we that would help us know who you are. It's like, okay. So I'm only going to say it sometimes. Which is of course what this second privacy mitigation does. But anyway, I just wanted to point out that it's there. Anybody who wants it in Chrome can turn it on, and then bad ads won't show.

Speaking of bad ads, Microsoft 365 Defender Research Team posted a blog titled "Widespread malware campaign seeks to silently inject ads into search results and affects multiple browsers." They named this thing, I don't know why, Adrozek, you know, "Ad" as in advertising, A-D-R-O-Z-E-K. Really rolls off the tongue. So one of the things that makes this malware noteworthy is that it is widely cross-family and multibrowser. It affects Edge, Chrome, Yandex, and Fox uniformly. And although its most prominent feature is unwanted ad injection, it not only injects ads, but the malware also exfiltrates any of the browser's stored credentials that it may have access to. You know, and that's when you tell your browser rather than your password manager that you want it to save things.

Unfortunately, if you're up in a browser's business, you can figure out what credentials it has saved because in order for it to send them, they cannot be encrypted. So that's a problem. So of course that could cause - exporting your credentials could cause significantly more harm than some unwanted ads injected into search results. I saw some pictures of it. And, yeah, it shows you what you would normally see would be Google results on a Google search, and instead like the whole first page is these bogus ads stuck in, tied to keywords. So it's trying to get ad payment benefits.

Microsoft said: "We call this family of browser modifiers Adrozek. If not detected and blocked, Adrozek adds browser extensions, modifies a specific DLL per target browser, and changes browser settings to insert unauthorized ads into web pages, often on top of legitimate ads from search engines. The intended effect is for users, searching for certain keywords, to inadvertently click on these malware-inserted ads, which lead to affiliate pages. The attackers earn through affiliate advertising programs, which pay by amount of traffic referred to sponsored affiliate pages. Cybercriminals abusing affiliate programs is not new," Microsoft wrote. "Browser modifiers are some of the oldest types of threats." In fact, remember, it was that Adaware thing that I found in my browser, one of those old BHOs, remember Browser Helper Objects, that caused me to write the first antispyware because it was spyware. It had gotten in unannounced and unasked for.

Anyway, Microsoft said: "However, the fact that this campaign utilizes a piece of malware that affects multiple browsers is an indication of how this threat type continues to be increasingly sophisticated. In addition, the malware maintains persistence and exfiltrates website credentials exposing affected devices to additional risks."

So anyway, it was surprisingly sophisticated. Disables browser updates to prevent its configuration modifications from being reversed. And it even establishes a Windows service to gain persistence over the long term. So that if you did something that tried to get rid of it, the service would keep running and say, hmm, and then put itself back. So it's a serious issue. Microsoft was tracking this thing's compromise of more than 30,000 PCs per day.

So the good news is I'm sure that Microsoft's security people became aware of these threats, and shortly thereafter so did Windows Defender Protection Suite. So it might be a good idea for some piece of mind just to ask Defender to perform a full scan of your various Windows systems from time to time. And that takes some time. There's no way around that. As the author of SpinRite, I'm all too aware that actually reading everything

on your system will take some time. So I just thought while I was putting the show together I would do that.

So I started up a full scan on my Win10 machine. And Defender says go ahead, use your machine while it scans in the background. But I did notice that it aggressively throttles its scanning so as not to interfere with your use of the computer in the foreground. So it's probably better to choose a time when you are about to be away from your machine and fire it up at that point. So when it was all done, while I was assembling the notes, I noted that it took 90 minutes, and it scanned 5,797,899 files.

Leo: That always blows me away when I see that.

Steve: And Leo, after the scan was finished, I just stared at that number. I remembered fondly, and this takes us back to our Altair days, I remember when our hard drives had seven files on them.

Leo: Yeah, yeah.

Steve: Now the number of files has seven digits.

Leo: It's amazing; isn't it?

Steve: You know, I do miss those days.

Leo: Yeah, yeah.

Steve: So an interesting new term of art. I just thought I would add this to our vocabulary. It's termed "Double Extortion" on the ransomware front. It's being used with increasing regularity, so I thought I would add that to everyone's lexis. The term is "double extortion." It originated with Check Point last April, referring to the double threat of encryption plus public exposure of proprietary data if the victim should choose not to pay up. As we know, some companies will be extremely sensitive to the reputation damage they might suffer, not to mention maybe even the potential liability if the news of their breach should become widely known. So henceforth that embarrassment strategy will be known as double extortion.

This is the third Tuesday of the month. The first one was right on the 1st. So this was one of those months where the second Tuesday was the earliest it could possibly be. And of course your first thought upon hearing that last week Microsoft patched 58 known vulnerabilities across their various products might be to think, wow, only 58 this month. That's way fewer than the more than 100 we've been beaten down into accepting as normal this year. But then, when you stop and look closer, you realize that more than one third of those 58, 22 in total, are all remote code execution vulnerabilities. Wow.

So, and because several of them are on Exchange Server and in SharePoint, both which have natural exposures to the Internet, I hope that everyone has by now made time to get those updated, although none are zero days, meaning that none are known to be under exploitation at the time of their discovery. A total of nine of those 22, all of which are remote code execution, are rated critical, and some are not difficult to exploit once

they become known. We know that bad guys rejoice now every month, whether it's the holidays or not, and quickly work to reverse engineer Microsoft updates in hope of working out an effective exploit before hapless Windows users update their vulnerable machines for those now known problems.

And of course this is especially true when they're enterprises running servers that they would like to avoid rebooting and having offline during a patch cycle. Maybe that explains the reticence to do that. Well, and of course sometimes updates actually do tank a machine so that they don't want to do it without, like, making a full image so they're able to roll back if they need to and so forth. Anyway, among these 58 that Microsoft fixed this month, last week, was a bug in Microsoft's Hyper-V virtualization technology. It was exploitable via a malicious SMB packet and would allow remote attackers to compromise virtualized sandbox environments, which of course Hyper-V was designed to protect from, but not so much. So yes, as always, don't wait to update for long.

Oh, and, okay. Speaking of Microsoft updates, here's a little bit of tid that caught my eye. The news was about a zero-click wormable vulnerability in Microsoft Teams. Before this was fixed, it would have allowed an adversarial attacker to remotely compromise a target's machine simply by sending them a specially crafted chat message. The reception of the message would have enabled zero-click remote code execution on that system. This discovery was reported to Microsoft at the end of August, on the 31st, by Oskars Vegeris, a security engineer with Evolution Gaming. Microsoft addressed the issue at the end of October.

In Oskar's write-up he said: "No user interaction is required; exploit executes upon seeing the chat message. The result is a complete loss of confidentiality and integrity for end users access to private chats, files, internal network, private keys, and personal data outside MS Teams." So, like, really, really bad. And it's cross-platform. It affects Microsoft Teams for Windows, for Linux, for macOS, and even the web version. And as I said, as a consequence of the zero-clickness, it can be made wormable so that upon receiving a chat, that machine executes with the user needing to do nothing and is able to then send chats out to, for example, maybe everybody that they have a chat history with, and it could explode. So very much like the iOS problem that we talked about last week.

So here's what just brought me up short. This is obviously quite serious. It's zero-click. It's wormable. It's a remote code execution vulnerability. And it was assigned no CVE designation by Microsoft. Why? Microsoft said, and I quote: "It is currently Microsoft's policy not to issue CVEs for flaws in products that automatically update without user interaction." What? So I haven't tracked down the policy, but this seems like a change. It would be very interesting if you were to ask Paul and Mary Jo about this tomorrow.

Leo: Okay.

Steve: So could Microsoft's solution to the embarrassment of hundreds of CVEs being patched every month to simply redefine problems by whether or not they will automatically be repaired? If so, it's a whole new ballgame. This would suggest that anything that auto updates like Windows would no longer have any actual vulnerabilities. After all...

Leo: It's going to be fixed.

Steve: ...Windows is now, by this definition, a continually moving target that's always in flux. So Leo, those are not actually vulnerabilities at all. They're just some miscellaneous things, like remotely taking over a Microsoft Teams user by sending them a chat message, that haven't been finalized yet. But don't worry, we're working on it. It's not worth bothering yourself about. Okay. So, right, we're not going to assign those any vulnerability numbers. Don't worry about it. No CVEs here. We fixed it. Because, after all, it updated, and it's not a problem anymore. Right.

Cisco is Jabbering. They've been having recurring trouble keeping their chat system secure. They have again attempted to patch their Jabber conferencing and messaging application against a critical vulnerability that made it possible for attackers to execute malicious code that would spread from computer to computer, once again with no user interaction required. You know, we're going to have to see one of these happen at some point because we keep finding them before the bad guys do, and at some point some bad guy's going to say, oh, this is going to be fun because everyone keeps talking about these, but no one ever does it. So here goes.

The discoverers of the trouble, Watchcom Security, explained what's been going on. They said in their TL;DR: "Three months ago, Watchcom disclosed four high-severity vulnerabilities in Cisco Jabber. One of the vulnerabilities allowed Remote Code Execution by sending specially crafted chat messages a problem that everyone seems to be having. The vulnerabilities were reported to Cisco, and a patch was issued. Shortly after, one of Watchcom's clients requested a verification audit of the patch to ensure that the vulnerabilities had in fact been sufficiently mitigated." Whoops.

Three of the four vulnerabilities Watchcom disclosed in September have not been sufficiently mitigated. Hello, Cisco? Are you listening? Is anyone home? Watchcom reported that Cisco released a patch that fixed the injection points they had reported, but the underlying problem was not fixed. And consequently, Watchcom was able to find new injection points that could be used to exploit the same underlying vulnerabilities. All currently supported versions of Cisco Jabber client from 12.1 through 12.9 are affected.

So for the sake of clarity they assigned three newish vulnerabilities CVE numbers to distinguish them from the original similar vulnerabilities which were disclosed to Cisco in September, and Cisco just sort of said, oh, whoops, and put a little putty on them or, you know, I don't know, a piece of Scotch tape. They explained that these newfound vulnerabilities have the same impact as the original and range in severity from medium through critical. As such, two of the vulnerabilities can be used to gain remote code execution. And Cisco, as I said, didn't really bother to fix them.

The most severe vulnerability is a cross-site scripting vulnerability that can be used to achieve remote code execution by escaping from the Chromium Embedded Framework, the CEF sandbox. This vulnerability does not require user interaction and, again, is wormable. Since the payload is delivered via an instant message, this means it can be used to automatically spread malware without any user interaction.

The second vulnerability can be exploited to collect NTLM - you know, that's NT LanMan, LAN Manager - password hashes from unsuspecting users. In a very clever hack, by sending a message that contains a malicious image tag, an attacker can induce the victim's Cisco Jabber client, which is apparently based on the Chromium Embedded Framework. They can get the Jabber client to interact with a file share under the attacker's control. If the file share requires authentication, the victim's NTLM password hash will be sent to authenticate the victim, and thus the bad guy can capture the LanMan hash. That's actually been done in the past, but not in this context. So these are guys who know what they're doing and said, hey, you know, here's how we could get a Jabber client to give us some more useful information.

The third vulnerability involves the custom protocol handlers used by Cisco Jabber. These protocol handlers are vulnerable to command injection because they failed to consider URLs that contain spaces. By including a space in the URL, an attacker can inject arbitrary command-line flags that will be passed to the app. Since the app uses, again, CEF, the Chromium Embedded Framework, and accepts Chromium command-line flags, several flags that can be used to execute arbitrary commands or load arbitrary DLLs exist.

So again, a clever hack. Cisco apparently asleep at the switch on this. You know, the first patch round that Cisco implemented filtered some of these. Just, I mean, they didn't actually fix the problem. They just said, oh, okay, we'll prevent that bad thing that Watchcom found from happening. But they didn't, as I said, get to it. So hopefully this time.

Watchcom wrote, in their conclusion of their disclosure, something that I thought was worthy of the whole story. They said: "The continued existence of these vulnerabilities, even after the first patch, highlight the complexity of modern software and the challenges developers face when trying to secure it. When choosing to use frameworks such as CEF, it is important to consider their security implications. Security should also be considered in every step of the development process, both in the initial planning stages as well as during implementation and maintenance."

They said: "This also serves as a reminder that software acquired from external vendors also pose a risk to organizations' IT security. It's important to be aware of these risks and take steps to mitigate them. Watchcom recommends regular audits of third-party software for security vulnerabilities." And of course amen to all that. This is one of the things that we've been talking about a lot recently is there have been serious issues where external libraries like this Chromium Embedded Framework in this case are just dropped in.

It's like, oh, look, we just got a browser. Like in our Jabber. How cool. It's like, uh, yes, except you've got the whole browser. You've got a browser that will accept command-line options, and you're allowing spaces in URLs to get past to it. So bad guys can send it something that will cause it to execute whatever they want. That's now under their control. So, I mean, it's easy to recognize that this is kind of a thing that you can just miss in oversight, but it's going to cause problems.

D-Link VPN servers have some embarrassing vulnerabilities, three of them. They were discovered by the guys at Digital Defense and were subsequently responsibly disclosed to D-Link four months ago on August 11th. D-Link finally confirmed the issue in an advisory on the 1st of December, so two weeks ago, adding that the patches were under development for two of the three flaws, which have now been released to the public. The flaws are high-risk and, as I mentioned, a little embarrassing. They are security vulnerabilities affecting D-Link's widely sold VPN router models DSR-150, 250, 500, and the DSR-1000AC, and other VPN models running the same DSR family. It affects current firmware versions 3.14 and 3.17.

Even if the devices, these VPN servers, are secured with strong passwords, the vulnerabilities have left millions, because these things are widely used, millions of home and business networks open to attack. And we know, just because D-Link has said, oh, new versions of the firmware are on our website, okay, of millions of home and business networks, how many are going to update?

Leo: Zero? One percent?

Steve: I know. It's just, I mean, Leo, it just seems like it's out of control. You can hardly blame Microsoft for saying, okay, wait a minute, we're going to redefine "vulnerability."

Leo: Yeah, not a vulnerability.

Steve: Because they provide these vulnerabilities, a full authentication bypass allowing remote attackers to execute arbitrary commands on those devices through specially crafted requests. And did I mention that these were particularly embarrassing? The flaws originate from the fact that the web management interface, exposed publicly, uses Lua CGI, which is fully accessible without authentication and lacks any server-side filtering. This makes it possible for an unauthenticated attacker to inject malicious commands that will be executed with root privileges. And this works over the Internet-facing WAN interface. So as always, the takeaway for our listeners, if you or your enterprise are using any of these quite popular D-Link VPNs, be very sure to obtain an update to the most recent firmware with some priority because these are being attacked in the wild.

Leo: If you turn off WAN administration, that would also work; right? You have to have...

Steve: Yes.

Leo: You have to be WAN accessible.

Steve: Yes.

Leo: And we do recommend everybody do that on every router unless you absolutely need it.

Steve: Absolutely.

Leo: Yeah.

Steve: Yeah. We'll actually talk in a minute about an issue of that relating to WordPress. I just did want to really briefly mention that Google suffered an outage. Nothing to see here. The conspiracy folks stepped into a brief, multi-hour, I mean, it was only a few-hour vacuum, with various attack theories and nonsense. But Google quickly dispelled those. Google first acknowledged the trouble at 4:20 in the morning our time, Pacific time, on the West Coast. They posted: "We're aware of a problem with Gmail affecting a majority of users. The affected users are unable to access Gmail. We will provide an update by December 14, 2020" - so that's yesterday morning. So at 4:20 - oh. So this was at 4:12 Pacific time - "detailing when we expect to resolve the problem. Please note that this resolution time is an estimate."

Then three hours later, at 7:20 a.m., they explained. Well, kind of. They said: "Today, at 3:47 in the morning Pacific time, Google experienced [what they called] an authentication system outage for approximately 45 minutes due to an internal storage quota issue. This was resolved at 4:32 a.m. Pacific time, and all services are now

restored." So unusual as that was for Google, it was not an attack, nothing untoward. Just whatever an authentication system outage is. Sounds like it's a backend thing that authenticates people's Gmail access. And if that's down, then everything that it authenticates would be down. So, yeah.

Last Wednesday, during Black Hat Europe 2020, researchers from Forescout Technologies presented their paper titled: "How Embedded TCP/IP Stacks Breed Critical Vulnerabilities." In their teaser synopsis for the conference they said: "In the past few years, there's been a rise in critical vulnerabilities affecting embedded TCP/IP stacks which had remained undiscovered for over a decade. The direct, unauthenticated, and sometimes cross-perimeter network exposure of these stacks, the often privileged portions of the system they run in, and their position at the top of opaque supply chains complicating vulnerability management efforts make for a highly dangerous mix resulting in periodic waves of critical vulnerabilities affecting billions [with a 'b'] of devices across industry verticals. But contrary to what many assume, the fragility of these fundamental components isn't limited to specific vendors or older, closed-source stacks alone.

"In this talk, we will present over a dozen" - and actually well over - "over a dozen new vulnerabilities" - and they of course mean newly discovered vulnerabilities - "in multiple widely used embedded TCP/IP stacks deployed in everything from networking equipment and medical devices to industrial control systems. We will discuss the nuances in their exploitability and potential impact and demonstrate a proof of concept against a yet-to-be-disclosed high-profile target. In addition, we will present the first quantitative and qualitative study into vulnerabilities affecting embedded TCP/IP stacks showing a clear pattern to the affected components and features, as well as the root causes of the vulnerabilities that affect them. Finally, we will provide concrete advice on how to mitigate and manage vulnerabilities affecting billions of devices in the absence of centralized patching and notification efforts."

And of course the absence of centralized patching and notification is one of our big hobbyhorses on the podcast because it is such a problem. So needless to say, their presentation is quite a meal. That's the introduction to their 47-page paper. I've got a link to it in the show notes. But stepping back a bit, they coined the name Amnesia:33 because they uncovered a set of 33 vulnerabilities collectively impacting four different open source TCP/IP protocol stacks, one known as Micro IP; the second, FNET; the third is picoTCP; and the fourth is Nut/Net. They are commonly used in IoT (Internet of Things) and embedded devices.

As a consequence of improper memory management, successful exploitation of these flaws could cause memory corruption allowing attackers to compromise devices, execute malicious code, perform denial-of-service attacks, steal sensitive information, and even poison DNS cache memory. In real-world scenarios, the attacks could play out in various ways, disrupting the functioning, for example, of a power station to result in a blackout, or taking smoke alarm and temperature monitor systems offline by using the DoS vulnerabilities, meaning it's, as we know, trivial to crash these things.

And so if you've got an embedded device that doesn't have a so-called "watchdog timer," which is able to notice that the device hung and then perform a physical restart, you know, well-designed systems tend to; really inexpensive things just don't. They just - it doesn't occur to them. That would cost an extra penny to put into the hardware, so we'll save that. Many millions of devices from an estimated 158 different vendors are vulnerable collectively to these Amnesia:33 discoveries, with a possibility of remote code execution allowing an adversary to take complete control of a device and using it as an entry point on a network of IoT devices to then move laterally.

And of course we've also been talking a lot about lateral movement, thanks to the Zerologon flaw that allows somebody that gets into an enterprise to easily compromise

the Active Directory system and move laterally throughout the network. This allows them to establish persistence, co-opt their compromised systems without any outward appearance of compromise, thus setting up shop. And of course our topic for today is, as we'll see, is big on that. So if we imagine that nation-state actors are greedily mopping up all available exploits everywhere they appear, then this research from Forescout was likely greeted with a great deal of mopping because IoT devices are exploding in number.

Forescout said that: "Amnesia:33 affects multiple open source TCP/IP stacks" - yup, those four - "that are not owned by a single company. This means that a single vulnerability will exist across multiple codebases, multiple development teams, multiple companies and products, which presents significant challenges to patch management." You know, there's not one person to notify. And oftentimes these things are forked. They'll take, they'll thank you very much, take the source code repository in-house and then do their own modifications from there, thereby never getting the benefit of the original repository's fixes, which these guys would be happy to provide.

So they say: "Because these vulnerabilities span a complex IoT supply chain, Forescout cautioned it's as challenging to determine which devices are affected as they are hard to eradicate." Right? Because they're just embedded. You don't know which TCP/IP stack is in that smart thermometer or in that smart plug or in the remote webcam. That just, like, comes with it. It's embedded. So they stem from out-of-bound writes, buffer overflows, lack of input validation. You know, basically running the gamut of a well-meaning but casually designed, oh, look, it works. You're welcome to use it.

And unfortunately it's got lots of problems. These guys found 33 different ones. So there are critical remote code vulnerabilities in this Micro IP, picoTCP, and Nut/Net. They have been disclosed. They're publicly known. Each of them has a CVSS score of 9.8. So yeah, remote code vulnerability, trivial to exploit. Some of the vendors who do utilize these stacks are being responsible. Vendors such as the very well-known Microchip Technology, they're using this open source stack. And Siemens, whose products are affected by these vulnerabilities, have released security advisories about them. But again, that's the exception to the rule. There's two companies. That leaves 156 others. As Forescout put it: "Embedded systems such as IoT and operational technology devices tend to have long vulnerability lifespans" - okay, there's a new term to coin, "long vulnerability lifespans," meaning, right, they never get patched.

They said: "...resulting from a combination of patching issues, long support lifecycles, and vulnerabilities trickling down, highly complex and opaque supply chains, or sometimes not trickling. As a result, vulnerabilities in embedded TCP/IP stacks have the potential to affect millions, if not billions, of devices across vertical markets and tend to remain a problem for a very long time." The problems disclosed were severe enough for the CISA, you know, our U.S. CISA, to get involved and to urge awareness. But that didn't appear to have much impact when they urged companies to update against the Microsoft Zerologon vulnerability earlier this year. Asking IoT vendors, random vendors, probably most in China, to patch their unpatchable devices seems a clearly doomed exercise in futility.

So what's our course of action? What's our takeaway? My feeling is that we must treat our IoT gadgets with the assumption that they are compromised and rigorously relegate them to their own isolated networks. If you're able to access your various IoT devices from outside your home, then it's clear that you and they do not need to share a common network. Your untrusted IoT LAN should coexist with your trusted LAN. But they should not have any contact with one another. And they don't need it. You know, I'm not suggesting that we're seeing like broad-based intrusions. We're not at this point.

But I would not be surprised if there isn't some sort of IoT-pocalypse at some point where these things, they're just like, there's such a critical mass of them, all connected

out of this country, that they just don't represent such low-hanging fruit that it will be impossible for them not to be abused. Let them attack each other on their own LAN segment. Don't let them get to your main operating LAN where you've got your PCs. And as I said, if you can change your home temperature when you're out roaming around away from the house, or turn plugs on and off and things, you're already on a separate network. That demonstrates that your trusted LAN does not need to have contact with your untrusted IoT LAN. And this need for network segmentation I would argue has never been greater.

Then we have another WordPress mess. I got a kick out of the subhead that ZDNet chose for their coverage of this. Their headline was "Zero-day in WordPress SMTP plugin abused to reset admin account passwords." And their subhead was "A patch was released earlier this week; but many WordPress sites remain unpatched, as usual." So, yes, the word is getting around that the fact that patches are released, well, that's good. But it's necessary, but not sufficient.

So first off, as we know, the term "zero-day" has unfortunately become synonymous with "bug." The press is tending to call everything a zero-day because it sounds a lot more serious. It was meant to sound more serious. But referring to everything as a zero-day will ultimately render the term worthless. In this case, refreshingly, it really is a zero-day, although that's not good for the people whose WordPress sites have been hacked. Hackers have been using a design mistake, coupled with a dumb configuration setting, of a popular WordPress add-on to easily reset the admin passwords on WordPress sites. The add-on is considered popular because it's installed on more than half a million sites. The hacking has been underway for some weeks, and the patch for the design error was made available last Monday. Thus it is a true zero-day vulnerability.

The add-on in question is called "Easy WP SMTP." Of course we know that's Easy WordPress Mail, right, email, SMTP, Simple Mail Transfer Protocol. Obviously a plugin that lets site owners do something, in this case configure their SMTP settings for their site's outgoing emails and add features to it. One of the features it boasts is the option to enable debug logging to see if the emails are getting sent out successfully or not. That feature causes the system to log all email headers and the email body that is set. And that email log is located in the plugin's well known installation directory: /wp-content/plugins/easy-wp-smtp. Thus that's no mystery.

But the team at Ninja Technologies Network (NinTechNet) discovered that, although Easy WP SMTP v1.4.2 and earlier, which was current before last week's update, although it gives the log a fancy random name, like "5fcdb91308506_debug_log.txt," oh, nobody is ever going to figure that out; right? The plugins folder where this default directory, where this log is, lacks any index.html file.

So when the site is being hosted on servers with directory listing enabled, hackers can view the directory, see the fancy named email log, and view its contents. Now they can see everything being sent out. Then they cause the blogging site to send its admin a password reset email, refresh the view of the sent email log, capture the password recovery link, and take over the site. So no rocket science here, folks. Just, whoops, a problem with a plugin, more than half a million of these things out there. And sites are being compromised right and left.

I mentioned before that while I was hosting my own WordPress blog, I was horrified by the idea of the site's admin login form being public. The idea that anyone in the world could enter the well-known URL of the admin login and be looking at a prompt for a username and password to log in as me was appalling. So one of the first belt-and-suspenders things I did was to completely block access to any admin-related pages first and foremost the front door from any remote IPs other than mine. As we know, the public IPs we're assigned by an ISP are relatively static, so it's just as simple as using

a .htaccess file; or, in my case, a web-config file because I'm using IIS to filter incoming page requests.

If my IP did happen to change so that I would also be locked out of my own admin page, then I would need to log onto the server which is hosting the site, which I would do using a certificate-tied SSH client obviously not merely a username and password. Then I would update the access control with my new remote IP and again be able to get to the page. So my point is I'll never know what attacks that bit of superstition might have thwarted. But the idea of exposing my WordPress login page to the world just made me shiver, as I hope it would any of our listeners. And again, it's easy to be a little proactive and just keep that front door closed.

And always the advice is really, really, really minimize your use of third-party plugins for WordPress. It must be that it is coming under additional scrutiny. Maybe that and a combination of over time it's gotten more popular. Lots of people are setting up sites. Bad guys want to just create, you know, we talked about it the other day, some weird - they were compromising WordPress-based ecommerce sites to put up weird bogus shopping sites, some crazy stuff. But apparently there's money in it, so that creates pressure to create the compromises.

Also last week at Black Hat Europe was the 2020 Pwnie Awards. As we know, the Pwnies are to our cybersecurity industry what the Oscars and the Razzies are to the movie industry, you know, both the best things and the worst things in their category. Every year cybersecurity researchers are invited to nominate and vote for both the best and the worst in our industry. This includes selecting the best and most ingenious vulnerabilities discovered during the past year and also the worst vendor responses and epic fails that put their users at risk. Traditionally, the Pwnies have taken place in August, you know, the summer in Las Vegas, during the Black Hat Las Vegas. But this year with COVID-19 pandemic virtualizing conferences, it was decided that the Pwnie Awards would be moved to Europe's Black Hat conference.

So among the results are many things we've talked about during the year. The best server-side bug went to BraveStarr, a remote code exploit in the Telnet daemon on Fedora 31 servers. The best client-side bug went to a zero-click MMS attack on Samsung phones. The bug was discovered by Google's Project Zero team, so that one they kind of kept in-house. But that was the best client-side. The best privilege escalation bug, of course, that's Checkm8, the unpatchable hardware jailbreak for seven generations of Apple silicon.

Leo: I want to see the thank you speeches for this, the acceptance speech. On behalf of all of my collaborators, I want to say [laughter].

Steve: The best crypto attack went to Zerologon, which as we know is a bug in Microsoft's Netlogon authentication protocol that can be performed by adding a bunch of zero characters in certain Netlogon authentication parameters.

Leo: These are moving targets. There's a worse one tomorrow.

Steve: Yes. Actually there was a worse one two days ago.

Leo: Yeah, right.

Steve: This horrible, well, it's the topic of our podcast.

Leo: Right, we'll get to it.

Steve: Then we had the most innovative research went to TRRespass, with two R's. That was bypassing the TRR protections on modern RAM cards to carry out Rowhammer attacks on them. The most epic fail was Microsoft for CurveBall, a bug in how the company implemented elliptic curve signatures on Windows, allowing for easy spoofing of HTTPS sites and legitimate apps. And, finally, the epic achievement award went to Guang Gong, a known Chinese bug hunter, for discovering three different bugs that allowed remote takeovers of Android Pixel devices. So the Annual Pwnie Awards brought back some of the things that we've covered in the last year, not surprisingly.

Also, not a flash in the pan. Adobe's infamous Flash player was anything but a flash in the pan, as we know. It was released 24, Leo, 24 years ago. Wow. In January of 1996. And of course back then web pages just laid there. They didn't do anything. They were predominantly static HTML. JavaScript, it turns out, was just beginning to happen. But it didn't have any of the new browser features to drive with scripting, so its application back then was very limited. But Flash added a complete, self-contained, content-authoring, local interaction and animation facility. You know, you could write working games in Flash, and many developers did.

And because it was a world unto itself, it was inherently browser agnostic. If a browser had a Flash plugin, the content would run, period. Didn't matter which browser, although there weren't that many back then, either. It really was quite something for the era. And it was immediately adopted by developers to create interactive content for the web. As we know all too well, however, Flash's Achilles heel was that it was originally written, like most of the software of that era, with virtually no regard for security. And it was never really able to recover from that lack of security legacy. It was much like the Internet back then. The fact that it ran at all was regarded as a miracle. Security wasn't even a thought, let alone an afterthought.

But thanks to the incredible progress made in turning our browsers into fully programmable web application hosting containers that they are today, driven by JavaScript, there are more than 1,444,231 add-on JavaScript function libraries. And we now have a very mature and formalized DOM, the Document Object Model, that allows a web page's presentation to be fully accessible and manipulatable by JavaScript. Consequently, the only thing that has kept Flash alive has been the residual inertia still remaining from its once total dominance over that aspect of what it offered. So against that backdrop, last week Adobe released their final update ever, which includes a kill switch, which will go into effect next month.

Leo: Wow.

Steve: And reminded the world that Flash is finally once and for all being extinguished forever. And it's not as if no one has received notice. It was way back in 2017 that Adobe, Microsoft, Google, Apple, and Mozilla made a joint announcement that they would be retiring support for Adobe Flash Player at the end of 2020. Well, that day has come, finally. In their final Flash player release notes, Adobe said: "We want to take a moment to thank all of our customers and developers who have used and created amazing Flash Player content over the last two decades. We are proud that Flash had a crucial role in evolving web content across animation, interactivity, audio, and video."

And indeed it did once. But beginning next month, Chrome, Safari, Firefox, Edge, IE11, and other Chromium-based browsers, which is like everybody else, will remove Flash from their bodies, and it will become impossible to put it back. So long and farewell, and phew. Wow. And, you know, it has been fading. We haven't, you know, we used to - Flash breaches were just a constant source of, unfortunately, of content for the podcast.

Leo: Oh, yeah, yeah.

Steve: And it's been fading. Believe it or not, it's not gone. There are going to be some enterprise things that die next month. It's just amazing. But there are probably things that are - they're still working. The source has been lost. Wouldn't even matter now if you had the source because it's Flash. And so it's just going to stop working next month.

Leo: I get a lot of calls. I've been getting calls on the radio show from people saying, "What do I do now?" And I say, "What do you mean, what do you do now? Nothing."

Steve: No kidding. No kidding.

Leo: Well, I don't think - I think they think there are sites that are using Flash that have long ago probably switched to other HTML5 technologies. Like YouTube hasn't used Flash in years. So I think they think they're using it.

Steve: And often when you see a site that tells you you have to install it, it's malware.

Leo: Yeah, don't, yeah.

Steve: It's telling you, it's like, oh, here, click this link.

Leo: You need Flash.

Steve: You need Flash in order to view the site content.

Leo: Yeah, right.

Steve: It's like, what?

Leo: No website that wants visitors would still be running Flash. So I told them don't do anything. Don't worry about it. Get rid of it. You'll be fine.

Steve: Yes, yes. And speaking of its formal welcome replacement, while I'm on the topic of browser coding and automation, as I mentioned at the top, JavaScript is celebrating its 25th birthday, and we're in the second week of free courses being offered, one per week, at JavaScript.com. So the site says - as you would imagine, it's www.javascript.com. And

the site says: "To celebrate one of the most popular languages in the world, we're making five of our expert-authored JavaScript courses free each week in December."

And I apologize for not catching it last week. That was the first week of the free courses. The second week is available, and there's still three, four, and five. And I looked over the summary of them, and they look wonderful. So I'm not vouching for them, but just based on the topics that each one covers, if you're interested in JavaScript, if you want to buff up on your JavaScript, JavaScript.com. You can sign up and get one free course. There are four weeks of them starting this week for the remainder of the month. So looks like it might be useful.

Leo: Yeah.

Steve: Two bits of closing the loop. David P. Vallee said: "Hi Steve. Listened to the Amazon Sidewalk podcast." That was last week. "Sounds like Amazon did everything imaginable to protect customers. The question that occurs to me is how often does the Internet connectivity of a single home go down? If the carrier drops, everyone's IOT devices will go out in a large radius, way beyond the range of Sidewalk. Since IoTs use a small amount of bandwidth, for a home to need this, they would need a complete failure of either their router or modem. When's the last time yours, or a friend of yours, had a router crash? Thanks again for a great podcast."

Well, okay. I probably used a poor example. And I was reaching for, in the example of the Genie garage door, the garage door that would, if your network went down, it would just use your neighbor's, I mean, it would. Mostly the concept here is to create a WAN; right? The example they gave of their Amazon employees who just all took some gadget home, and all automatically L.A. Basin, the Los Angeles Basin, was completely covered with Sidewalk access, 900 MHz connectivity. So that's really the point and the whole goal. And I think it's very cool.

Skynet, who tweets from @fairlane32, said: "I'm excited about Amazon Sidewalk in that if you could get pets that are microchipped onto the Sidewalk network, it may be possible, providing a lot of people participate, to locate them if they're ever lost. Imagine being able to find those dogs and cats that get loose. And with all the established social media groups using the network, you could possibly be able to find lost pets within hours, not days or weeks." He says: "But are those microchips transmitting on the 900 MHz spectrum?"

So I did want to clarify. The microchips that I think he's talking about are not the tags, which are collar-based tags. The microchips are little RFIDs. They need to be pinged with energy. They don't contain a battery in them. It's just basically for serializing things. So you could have a lost pet, and if it went to a vet or got to a vet who had a chip reader, then that would tell them who the pet belonged to potentially if you were registered.

But in the Sidewalk example that Amazon gave, and this is working technology, it's a tag like the size of a square square; right? You know, like an inch on a side and not very thick. But it contains probably a 2032 or a 3032 mercury cell that'll last for some length of time. And then, yes, that thing is on the 900 MHz network. And also probably not persistently. It probably pings the network to sort of save battery and periodically announce its location, which would allow then some finding knowing which part of the network was receiving the information. And that would go to Amazon and then to the application server that would then be able to close the loop. So anyway, just two interesting bits from our listeners.

InitDisk, the free USB thumb drive formatting tool, is now at Release 4. I mentioned it had been updated to 2 last week. It's had two more releases. We are overall at the release candidate stage for this project, the ReadSpeed project, where we've been for a couple weeks. But a diminishing number of minor things are still popping up here and there. My feeling is that it's much better to deal with them in our currently relatively quiet setting with people who've become very familiar with the project, than ignoring those edge cases and putting them off until a much broader public release. So as very anxious as I am to get all of this work into the hands of a much wider audience, it makes more sense to first get it as right as possible.

And so to give you a sense, last week we found a system whose BIOS did not like the Master Boot Record which I was using on the original couple InitDisk releases. And that was the MBR, the Master Boot Record, from Windows 2000, which is a highly regarded way to boot. And I deliberately chose it for its maturity and its assumed compatibility. But that system did like the MBR from Windows 7 because remember the Master Boot Record is a little table of four entries describing the partitions that follow. But it's actually executable code. So it matters what's in there. It's not just the table.

The BIOS doesn't know about an MBR. It just knows to jump to the beginning of the first sector of the media, which then is a series of instructions that takes control, scans the table, looks at the entry, and then jumps out to the proper partition and runs its first sector in order to begin booting the OS. Anyway, we're now, as a consequence of having discovered that there was a system that didn't like the Win2K, we're now using Win7 as the Master Boot Record. And so far nobody's had a problem with it. We got an increase in compatibility, which of course is my goal.

Also somebody had a USB thumb drive that just refused to work on his Win10 machine. He was able to bring up a temporary Win7 PE system where it would work. And since my clear goal has been to create a single USB formatting utility that easily and always works everywhere for everyone, I needed to figure out what was going on. We struggled with that one for a couple days, trying all kinds of different things, until I realized that something about the history of that particular USB stick and his particular Windows system had to be the problem. So now InitDisk explicitly deletes any prior drive mounting history for that device as it's formatting it. And it completely cured his problem. So again, who knows how many other people would have been hit by that. Now they won't be.

Oh, and then a couple of people have dying drives that are in really bad shape, so bad that they barely benchmark. But I was able to improve the error handling on the benchmark so that it would keep looking for a nearby block of storage where it might be able to succeed. That's now much more robust. So of course the upshot of all this is that our already public InitDisk utility is now at Release 4, and it's better than ever. And this is the technology that the ReadSpeed Benchmark Windows prep utility will be using, as will the next SpinRite. So because we're going to be off next week with a Christmas flashback best-of edition, that's two weeks before you're going to hear from me again. And I'm sure I'm going to be telling you that it's ready for you to all play with. So I'm excited about that.

Leo: Oh, that would be exciting. So you're going to start working hard on Christmas.

Steve: I'll start off the New Year with that good announcement, yup.

Leo: All right. Let's talk about SolarWinds. First of all, what is SolarWinds? It's a security tool?

Steve: No, SolarWinds is a company that produces an extremely popular network management system.

Leo: Network management, okay.

Steve: Yeah.

Leo: It's like a dashboard or something.

Steve: Oh, no, no. I mean, it's a physical appliance.

Leo: Oh, it's an appliance, oh.

Steve: Like for example it contains, unfortunately, all of the credentials on the network. And like it specializes in managing them for large organizations and enterprises.

Leo: Okay.

Steve: So, I mean, it's not something that you want to have compromised.

Leo: Absolutely not.

Steve: Okay. So the story begins with last Tuesday, a week ago, last Tuesday's news and admission from FireEye that they were hacked. Now, they're a $3.5 billion security company, one of the largest of its kind in the world. It's 16 years old, founded in 2004. FireEye has more than 8,500 customers spread across 103 countries, and more than 3,200 employees worldwide. So I mean, this is like a serious big security firm. We've talked about FireEye often in the past. They've not been on our radar for a while, but I remember their name came up a lot years ago.

So in his disclosure of this event, their CEO, Kevin Mandia, explained at that time what they knew. So this is one week ago today, so it was on the 8th. He said: "FireEye is on the front lines defending companies and critical infrastructure globally from cyber threats. We witness the growing threat firsthand, and we know that cyber threats are always evolving. Recently, we were attacked by a highly sophisticated threat actor, one whose discipline, operational security, and techniques lead us to believe it was a state-sponsored attack. Our number one priority is working to strengthen the security of our customers and the broader community. We hope that by sharing the details of our investigation, the entire community will be better equipped to fight and defeat cyberattacks."

He said: "Based on my 25 years in cybersecurity" - this is the FireEye CEO - "and responding to incidents, I've concluded that we are witnessing an attack by a nation with top-tier offensive capabilities. This attack is different from the tens of thousands of incidents we have responded to throughout the years. The attackers tailored their world-class capabilities specifically to target and attack FireEye. They are highly trained in

operational security and executed with discipline and focus. They operated clandestinely, using methods that counter security tools and forensic examination. They used a novel combination of techniques not witnessed by us or our partners in the past.

"We are actively investigating in coordination with the Federal Bureau of Investigation and other key partners, including Microsoft. Their initial analysis supports our conclusion that this was the work of a highly sophisticated, state-sponsored attacker utilizing novel techniques.

"During our investigation to date, we have found that the attacker targeted and accessed certain Red Team assessment tools that we use to test our customers' security. These tools mimic the behavior of many cyberthreat actors and enable FireEye to provide essential diagnostic security services to our customers. None of the tools contain zero-day exploits. Consistent with our goal to protect the community, we are proactively releasing methods and means to detect the use of our stolen Red Team tools.

"We are not sure if the attacker intends to use our Red Team tools or to publicly disclose them. Nevertheless, out of an abundance of caution we have developed more than 300 countermeasures for our customers and the community at large to use in order to minimize the potential impact of the theft of these tools.

"Consistent with a nation-state cyberespionage effort, the attacker primarily sought information related to certain government customers. While the attacker was able to access some of our internal systems, at this point in our investigation we have seen no evidence that the attacker exfiltrated data from our primary systems that store customer information from our incident response or consulting engagements, or the metadata collected by our products in our dynamic threat intelligence systems. If we discover that customer information was taken, we will contact them directly."

Okay. So that was exactly one week ago today. Then, two days ago, on Sunday, the other big and startling shoe dropped. We'll stay with Kevin for the moment. His Sunday update posting was titled: "Global Intrusion Campaign Leverages Software Supply Chain Compromise." They had discovered, FireEye discovered the bad guys' point of entry.

Kevin wrote: "In our announcement on December 8th we stated we would provide updates as we discovered additional information, in order to ensure that the broader community is aware of the evolving threats we all face. As part of that commitment, we want to provide you with the following update on our investigation.

"We've identified a global campaign that introduces a compromise into the networks of public and private organizations through the software supply chain. This compromise is delivered through updates to a widely used IT infrastructure management software, the Orion network monitoring product from SolarWinds. The campaign demonstrates top-tier operational tradecraft and resourcing consistent with state-sponsored threat actors.

"Based on our analysis, the attacks that we believe have been conducted as part of this campaign share certain common elements. First, use of malicious SolarWinds update, inserting malicious code into legitimate software updates for the Orion software that allow an attacker remote access into the victim's environment; a light malware footprint, using limited malware to accomplish the mission while avoiding detection; prioritization of stealth, going to significant lengths to observe and blend into normal network activity; and high OPSEC, operational security, patiently conducting reconnaissance, consistently covering their tracks, and using difficult-to-attribute tools."

He said: "Based on our analysis, we have now identified multiple organizations where we see indications of compromise dating back to the spring of 2020, and we are in the process of notifying those organizations. Our analysis indicates that these compromises

are not self-propagating. Each of the attacks require meticulous planning and manual interaction. Our ongoing investigation uncovered this campaign, and we are sharing this information consistent with our standard practice.

"We have been in close coordination with SolarWinds, the Federal Bureau of Investigation, and other key partners. We believe it is critical to notify all our customers and the security community about this threat so organizations can take appropriate steps. As this activity is the subject of an ongoing FBI investigation, there are limits to the information we're able to share at this time. We have already updated our products to detect the known altered SolarWinds binaries. We are also scanning for any traces of activity by this actor and reaching out to both customers and non-customers if we see potential indicators.

"FireEye's mission is to make our customers and the broader community safer. We are methodically uncovering and exposing this campaign piece by piece and working to prevent future attacks. It will require coordinated action by public and private organizations to fully expose and mitigate this threat, and we intend to continue our efforts."

So then the interesting technical details. They begin by setting the stage, writing: "FireEye has uncovered a widespread campaign that we are tracking as UNC2452. The actors behind this campaign gained access to numerous public and private organizations around the world. They gained access to victims via trojanized updates to SolarWind's Orion IT monitoring and management system. This campaign may have begun as early as spring 2020 and is currently ongoing. Post-compromise activity following this supply chain compromise has included lateral movement and data theft. The campaign is the work of a highly skilled actor, and the operation was conducted with significant operational security." Then they get into the details.

"'SolarWinds.Orion.Core.BusinessLayer.dll' is a SolarWinds digitally signed component of the Orion software framework that contains a backdoor that communicates via HTTP to third-party servers. We are tracking the trojanized version of this SolarWinds Orion plugin as Sunburst." So that's their name for the malware. And I'll just take a moment to note that this Trojan was digitally signed by SolarWinds, I've seen a picture of the cert, on March 24th of this year. And once SolarWinds customers updated their systems to incorporate this malicious, though signed, properly signed by SolarWinds component, those customers were trojanized.

So the tech goes on: "After an initial dormant period of up to two weeks, it retrieves" - this now malicious DLL - "retrieves and executes commands, called 'Jobs,' that include the ability to transfer files, execute files, profile the system, reboot the machine, and disable system services. The malware masquerades its network traffic as the Orion Improvement Program (OIP) protocol and stores reconnaissance results within legitimate plugin configuration files, allowing it to blend in with legitimate SolarWinds activity."

I mean, this thing is deeply stealthed. I mean, and even the DLL, they had to first get the DLL, reverse engineer it so that they could add their malware to it, and create a new DLL that did everything the old one, the original SolarWinds DLL did, plus add their trojan functionality, and then get that into the right place at SolarWinds so that, when they did the next update, it would get updated. Or maybe they actually did, they got the source code repository - that probably makes more sense in retrospect - added their stuff to the source code repository, and then when the next DLL was built from the master source, along with it went the Trojan. It was signed legitimately starting on March 24th of this year. And every one of their customers, they have 30,000 users of this particular Orion technology, and they know that 18,000 of those 30,000 updated and installed this DLL, and it went live.

"The backdoor," they write, "uses multiple obfuscated block lists to identify forensic and antivirus tools running as processes, services, and drivers." So, okay. That means that this malware's deep knowledge of SolarWinds application software and network operation means that it was not a coincidental intrusion into SolarWinds, either. So we started talking about FireEye, who found this in their network, and that set off the alarms for the industry.

But SolarWinds was breached at some time before that as part of the setup for this. The bad guys would have had to first see whether they could somehow arrange to get their malware merged into SolarWinds' codebase so that it would then be signed and sent along with the next update and subsequent updates. They targeted SolarWinds because updates to this company's products would successfully spread any inserted compromise far and wide. This was clearly a huge effort.

The tech note says multiple trojanized updates were digitally signed from March through May of 2020 and posted to the SolarWinds updates website. And I have a link to one of them in the show notes - although HTTPS is turned into HXXPS, and there's brackets around the dot of the dotcom so that it's not an active link - for anyone who's interested. That trojanized update file is a standard Windows Installer Patch file that includes compressed resources associated with the update, including the trojanized SolarWinds.Orion.Core.BusinessLayer.dll component.

Once the update is installed, that malicious DLL will be loaded by the legitimate SolarWinds.BusinessLayerHost.exe or SolarWinds.BusinessLayerHostx64.exe, depending upon system configuration. After a dormant period of up to two weeks, the malware will attempt to resolve a subdomain of avsvmcloud.com. The DNS response will return a CNAME record that points to a command-and-control domain. The command-and-control traffic to the malicious domains is designed to mimic normal SolarWinds API communications. Of course, this would allow the malicious traffic to slip through malware-aware intrusion detection systems. The list of known malicious infrastructure is available on GitHub's web page.

"FireEye," they wrote, "has detected this activity at multiple entities worldwide. The victims have included government, consulting, technology, telecom, and extractive entities in North America, Europe, Asia, and the Middle East. We anticipate there are additional victims in other countries and verticals. FireEye has notified all entities we are aware of being affected.

"We're currently tracking the software supply chain compromise and related post-intrusion activity as UNC2452. After gaining initial access, this group uses a variety of techniques to disguise their operations while they move laterally. This actor prefers to maintain a light malware footprint, instead preferring legitimate credentials and remote access for access to a victim's environment. This section will detail a few of the notable techniques and outline potential opportunities." So there's a lot more in this that gets us down into the weeds.

So these folks were really clever. The more details that emerge, the more you realize how much time and attention the malware authors put into this campaign. For example, the IP addresses used for the campaign were obfuscated by VPN servers which were deliberately located in the same country as the victim so that the victim was using near country IPs, not for example some IP in Russia or China or somewhere that might itself have raised suspicion. Local VPN server endpoints were set up just for this purpose.

So what we know: SolarWinds networking and security products are used, as I said, by more than 300,000 customers, including Fortune 500 companies, government agencies, education institutions. They used to have a page, SolarWinds did, bragging about all their customers by name. But gee, I wonder what happened? It's disappeared from the

Internet. And I looked on the Internet archive, and it had been removed from there, too. So they're not at the moment bragging about their customer base, since that would provide a list of targets.

They also serve several major U.S. telecommunications companies, all five branches of the U.S. military, and other prominent government organizations such as the Pentagon, the State Department, NASA, the NSA, the Postal Service, NOAA, Department of Justice, the Office of the President of the United States. So it's not difficult to imagine the frantic scurrying that has been going on over the past several days. We don't know when exactly FireEye learned what we now know they know. So there may have been like early access within the government before they made their Sunday posting. The U.S. Department of Homeland Security, the U.S. Treasury, and the U.S. NTIA are all confirmed victims.

In an SEC filing, SolarWinds confirmed that the trojanized updates were installed by more than 18,000 of their customers. They're a major government contractor, with regular customers including CISA, U.S. Cyber Command, the DOD, the FBI, the Department of Homeland Security, the VA, and many others. And imagine being the U.S. NSA and learning that since March an extremely well-designed and carefully used spying agent has without doubt been operating within your network. We don't know what protections the NSA might have. But we know that they were unaware of it because this thing has been out in networks, 18,000 of them, since March of 2020 worldwide. Maybe the NSA has fully separate networks, that this Orion SolarWinds product was only on the administrative staff network and not on the internal networks. But there's no way this is not a huge event.

Citing industry sources, Reuters reported that despite a broad install base for the Orion platform, the attackers appear to have focused only on a small number of high-value targets, leaving most Orion customers unaffected. And, of course, this is exactly how you would act if you had just landed the "golden goose" of all intrusions, arranging to have itself installed into many ultra-high-value targets. You would want to protect that. You would not want to be attacking low-value targets where the opportunity of being discovered was too high, not worth what you'd get in return.

Other people familiar with the situation told the Wall Street Journal that the Russian foreign intelligence service is believed to be behind the attacks and that "hundreds of thousands of government and corporate networks" have been opened to potential risk, making it a notable attack that goes far beyond the garden variety espionage attempt. Now, hundreds of thousands of government and corporate networks, that seems like a bit of an hysterical exaggeration. But it's true that any networks that were accessible by any of the compromised networks would have been put at risk through lateral motion.

And Leo, at the Russian Embassy in Washington, D.C., someone pressed a keyboard macro which apparently automatically typed the response: "The reports are unfounded attempts of the U.S. Fake Media to blame Russia."

Leo: Of course.

Steve: Anyway, yeah.

Leo: It is hard to attribute these things. How is it that they seem so confident that it is Russia and Fancy Bear and all that?

Steve: Maybe we will learn.

Leo: Kind of the fingerprint, in a way, of how these guys operate?

Steve: Again, because this is an ongoing investigation, they're not saying at this point. But they are saying that it is Fancy Bear. If SolarWinds knows how the bad guys gained that first foothold within their enterprise network, they're not saying yet publicly. And of course the enduring trouble is that networks are networks of computers. We all know that once a single computer has been infected with malware, the nature of malware means that it's never possible to fully trust that machine again. You and I, Leo, have been talking about this for years. You just don't know that you've got it all. And this of course would be especially true for this presumed ultra highly sophisticated attacker.

But what makes matters so much worse is that it wasn't a single machine that was attacked, but a highly connected network management system which itself in turn had access to any number of other machines on the network. We know how good these perpetrators are, so they may well have planned ahead for the day that their hack was discovered by planting entirely different malware in targets they don't want to lose access to after the way they got in has been closed. I'd be surprised if they hadn't.

And to make matters worse, it turns out that Orion had been, and I'm using the past tense, a highly trusted component which by design was trusted to hold and deploy credentials, including the Domain Admin, Cisco/Router/Software root and enable credentials, ESXi and vCenter credentials, AWS/Azure/Cloud root API keys, and much more. If you had the malicious Orion component on your network, all of those credentials must be considered to have been compromised now.

And let's not forget that these cretins had been crawling around since the end of March of this year. The mistake they made and they probably didn't make it until they had already done a lot more damage elsewhere was in attempting to crawl around within FireEye's network.

Something tipped off FireEye to the presence of this extremely stealthful intrusion. If it weren't for FireEye's detection, SolarWinds would still be unwittingly distributing malicious components, and this probably Russian espionage campaign would still be going strong. So it was just the fact that a week ago something happened that FireEye noticed and said, what, wait, whoa, whoa, what is that? And that's the only reason we know about this. It's been since March 24th that that cert was signed containing the first update that was malicious. So since then, some group believed to be Russian had access. And we know from FireEye's forensics that these guys were really, really, really good. They did not want to get caught. They wanted to stay.

Leo: Wow. It's an amazing story. And I think we'll never really know the whole story. I mean, this is not the kind of thing that [crosstalk].

Steve: True. It'll blow over. And maybe the next presidential administration will put some sanctions on Russia for this behavior, if attribution can be made sufficiently clear. We'll see.

Leo: And the tools that they got, the FireEye tools, do we know much about them?

Steve: Yes, actually. Because FireEye is so embarrassed, they have published a whole bunch of stuff on their GitHub page. So there is, like, they now no longer feel that these tools are proprietary.

Leo: No point in keeping them secret, yeah.

Steve: Yup.

Leo: This kind of reminds me a little bit about when the NSA hacking tools were breached, and they caused endless problems as they were reused in malware attacks.

Steve: Right.

Leo: So, yeah, it's bad when these things get out. Among other things. All right, Steve. Well, I was hoping you would cover this because this is - the story broke on Sunday, right before TWiT, and there wasn't much to say at that time. I've been watching it with interest. And now we know so much more, even in a couple of days. So thank you for doing that update.

If you listen to this show, there are a few ways you can get it. The first place to go is Steve's site, GRC.com. He has 16Kb audio versions. Sounds a little tinny, but it is a small file. That's the advantage of that. Even smaller, not much probably, is the text version of it.

Steve: It's searchable.

Leo: It's searchable, which is nice. And of course thanks to Elaine Farris for doing that. He also has 64Kb audio. That's all at GRC.com. As soon as you get there, if you don't already have a copy of SpinRite, the world's best hard drive maintenance and recovery utility, the world's best SSD maintenance and recovery utility we now know - at least when you get 6.1 out it will be; right? That's very exciting. If you buy it now, you get 6.0, you will get 6.1, and you'll also be able to participate in the development of it, which is rapidly coming to a close. So this would be a good time to pick up SpinRite: GRC.com. There's lots of other free stuff there, including of course the world famous ShieldsUP!, which gave the phrase "stealthed" to - even Apple uses "stealth."

Steve: Yeah, it's important.

Leo: Yeah, everybody - it became the phrase.

Steve: Yeah, it became the jargon, yeah.

Leo: Thanks to Steve. We have 64Kb audio and video at our website, TWiT.tv/sn. You can also of course get it on YouTube. There's a YouTube channel for Security

Now!. You can subscribe in your favorite podcast player. That's a good way to get it, so you get it automatically. If you have questions for Steve, couple of ways you can reach him. His Twitter account is open to direct messages from all. That's @SGgrc. He also takes feedback at GRC.com/feedback. We have a forum, twit.community. Actually, go to www.twit.community if you want to sign up for that. That uses SQRL, by the way. If you have a SQRL account, you can sign up there easily. We also have a live chat that's going on now and all throughout the day and night at irc.twit.tv.

All right, Steve. We shall talk again in two weeks. Remember next week's the best-of. But we'll be back on the 29th with a look back at the year.

Steve: Right. And we should remind our listeners that I will be on the Sunday show with you.

Leo: The 20th, yes, this Sunday. A very special OG TWiTsters episode with Steve, Jeff Jarvis, Paul Thurrott, and me. That'll be a lot of fun. I always look forward to these holiday episodes. Thanks, Steve. We'll see you next week, I mean in two weeks, on Security Now!. Bye-bye.

Steve: Yeah, bye.

Copyright (c) 2014 by Steve Gibson and Leo Laporte. SOME RIGHTS RESERVED

This work is licensed for the good of the Internet Community under the Creative Commons License v2.5. See the following Web page for details: http://creativecommons.org/licenses/by-nc-sa/2.5/