Tsunami

Description: This week we look at Mozilla's surprise suspension of their Firefox Send service, Zoom's latest remote code exploit vulnerability, the latest revision of the U.S. Congress's EARN IT Act legislation, the growing tension with stalkerware apps, a Chinese Internet equipment vendor in the hot seat, the challenge of geolocating illegal drone operators, Fraunhofer's report of rampant router vulnerabilities, and SpinRite's move toward increased political correctness. Then we wrap up by looking at Tsunami, Google's latest and extremely useful-looking contribution to the open source community.

High quality (64 kbps) mp3 audio file URL: http://media.GRC.com/sn/SN-775.mp3
Quarter size (16 kbps) mp3 audio file URL: http://media.GRC.com/sn/sn-775-lq.mp3

SHOW TEASE: It's time for Security Now!. Steve Gibson is here. Yet another Chinese Internet equipment manufacturer showing some vulnerabilities, or maybe they're intentional. We'll tell you why you shouldn't worry too much about that recent Fraunhofer report on vulnerabilities in Internet routers. And then it's a look at a new Google vulnerability scanning tool they're giving away. It's all coming up next on Security Now!.

Leo Laporte: This is Security Now! with Steve Gibson, Episode 775, recorded Tuesday, July 14th, 2020: Tsunami.

It's time for Security Now!, the show where we cover the latest news about security, how things work, how to keep yourself safe with this cat right here, our security guru, Steve Gibson from GRC.com. Steve?

Steve Gibson: Also known as Steverino, or this cat.

Leo: This cat right here, man.

Steve: Or the best we can do on any given week.

Leo: No, that's not true. It is the best we can do, yes.

Steve: So we have Episode 775 titled "Tsunami," after Google's latest and extremely useful-looking contribution to the open source community. So we're going to wrap up talking about that. I think that will probably be of interest to a lot of our listeners,

especially in larger enterprises, which is what it's aimed at. But we've got a bunch of stuff to talk about.

We're going to look at Mozilla's surprise suspension of their Firefox Send service; Zoom's latest remote code exploit vulnerability; the latest revision of the U.S. Congress's EARN IT Act legislation still working its way; the growing tension with stalkerware apps; a Chinese Internet equipment vendor in the hot seat; the challenge of geolocating illegal drone operators; Fraunhofer's report of rampant router vulnerabilities; and also, sort of on a lighter note, SpinRite's move toward increasing political correctness for itself. And then we're going to wrap up, as I mentioned, by looking at Tsunami, which looks like another really useful contribution from Google. So a bunch of fun stuff to talk about this week.

Leo: Cool, very cool. I'm sad to hear about Firefox Send.

Steve: Well, it's going to be back. But it's interesting what happened to it, and that's really where the story is. So, yeah, be talking about that.

Leo: Steve?

Steve: And lest we forget the Picture of the Week, thank you somebody on Twitter for sending this to me. It's just kind of fun. I'm not sure what country this is where they have...

Leo: Looks like a college campus, actually. I bet you that's what it is.

Steve: Oh. That does look kind of like Harvard in the background maybe.

Leo: Yeah, yeah. And it's - or Colorado because I see the snow stick attached to the fire hydrant.

Steve: I was thinking that was a snow stick, yes, exactly. Although it's not on - I thought snow sticks were normally along the side of the road; right? This looks like it's marking...

Leo: Well, it's on the path. You also need to know where the hydrant is if it's buried in the snow.

Steve: Like a sidewalk. Anyway...

Leo: You may be able to find the hydrant, but don't try to use the emergency phone.

Steve: So there's a phone stuck over, looks like an empty hole in this kind of very modern-looking structure. And it just says, as if you wouldn't know, says "Emergency phone not installed." And then the lower half of the note: "Please do not have an emergency at this location."

Leo: Definitely campus humor. Definitely.

Steve: Yeah. That does seem apropos.

Leo: Yeah, yeah.

Steve: So if someone were to go to send.firefox.com at the moment, they would be greeted by the little screenshot that I have at the top of the security news of our show notes, which reads, "Firefox Send is temporarily unavailable while we work on product improvements. We appreciate your patience while we make the Firefox Send experience better." And Leo, I think you did just go there, didn't you.

Leo: I did, yes.

Steve: And there it is.

Leo: That's too bad because I really love Firefox Send. It's such a good idea.

Steve: Oh, it's my go-to, yes, it's become my go-to file sharing service. The good news, it'll be back. The bad news is why it went away. Just to remind our listeners, we've talked about it before. It allows files of up to 1GB if you're not signed in, or 2.5GB if you are, to be locally encrypted in the browser, optionally password protected so that only the recipient is able to retrieve and decrypt a sent file. You get retention controls allowing the sender to set the time, that is, the duration and/or a download count, after which that content will expire from the Firefox Send cloud and be removed.

Unfortunately, as with anything that is simple, free, and effective, like think email, it's also subject to abuse by nefarious forces. The bad guys also love Firefox Send because it lets them generate short-term links based on good-looking trusted domains for sharing arbitrary evilware to unwitting victims. So thanks to Firefox Send, the bad guys don't need to set up their own file sharing server and try to get a legitimate-looking URL domain. They don't have to worry about making sure that URLs expire automatically. Mozilla does that for them. And links that only work once create an extra challenge for security researchers because, even if the malicious URL is captured in a log, by then it's probably been used. So it's not possible to go back and obtain the original because it's been removed already. And of course, since the IP is one of Mozilla's servers, it's not one that anyone wants to just put a blanket block on. And I was going to say they wouldn't want to blacklist it, but I'm working to be better.

Over the past few months, Firefox Send, it turns out, has been used increasingly to store payloads for all sorts of cybercrime operations, from ransomware through financial crime, banking trojans, spyware, and used to target human rights defenders. FIN7; REvil, also known as Sodinokibi; Ursnif, which is also the Dreambot network; and ZLoader are just some of the malware gangs and strains that have been seen hosting payloads using Firefox Send.

As a consequence, the cybersecurity industry has finally tipped its collective hat to Mozilla for suspending what has become a widely used and unfortunately now widely abused service. Mozilla didn't just say, oh, you know, we recognize there's a problem.

We'll be considering some changes in the future. They just shut it down. They said, okay, we're just going to stop making this available until we can upgrade it.

Cybersecurity researchers have suggested various changes to strengthen the service. One is to add a Report Abuse button so that flagging or killing malicious links could be made much more quick and easy. What Mozilla said in their statement about this, they said: "Before relaunching, we will be adding an abuse reporting mechanism to augment the existing Feedback form, and we will require all users wishing to share content using Firefox Send to sign in with a Firefox Account."

So it's sad that once again we see the Internet's inherent anonymity being abused and then having to be restricted. It was cool to be able to send up to 1GB with a 24-hour expiration without needing an account with Mozilla, even though I have one. But just as Zoom was forced to limit what they would allow to be done with full anonymity, so too now has Mozilla. And as we know, even requiring an account is not a very high bar. And my sense is it's not going to be very effective, but at least it will help a bit. And it will help Mozilla to say, hey, you know, we've done all we can. We're doing all that we can do.

But, you know, in light of all this, of everything we see going around us, I will make a prediction that in some future - not tomorrow, not even soon, maybe not during our lifetime. But I'll bet that some set of abuse-prone services, maybe anything that generates rather than consumes content, will end up requiring some form of affirmative, probably government-issued, verifiable identity. In that possible, and I would argue probable eventual future, it will be permissible to browse and read and consume content anonymously. But any public content will probably be traceable back to its source. We know we're not there today, and it won't happen soon. But I'll bet it happens eventually.

Or maybe, as a sort of an interim step, content will be flagged as being from a verified source or not. Sort of like verified Twitter identities that would allow users to choose what level of veracity and/or risk they wish to accept. Or make it just very, very visible that something is from a verified source or not. We have that with HTTP versus HTTPS. We have the lock icon saying that this was secure. Initially it was to protect users from submitting content over an insecure link. Then later it became an aspect of reputation. It was the reputation was better.

Then of course, as we know, security became just a given for all websites so that now if a site doesn't have a TLS certificate, it's like it's said to be insecure. But in any event, they had to pull down, you know, Mozilla had to pull it down. They're going to change the way it works and then bring it back up. Maybe they'll do more things. We don't really know. But it had to go away because it was being abused. So it's sort of sad to see that happen, and to sort of see the dream of an anonymous Internet sort of slowly disappearing, being chipped away at.

And speaking of Zoom, they fixed a new remote code execution flaw that turned out to be affecting Windows 7 and earlier systems. And it was a bit of an odd chain of events. A private researcher, who wishes to remain anonymous and still is, quietly reported his discovery of what was at the time an unknown remote code execution vulnerability in the Zoom client, which affects all versions of Windows up to and including 7 and probably its likely matching Windows Server 2008 R2.

He did not report it to Zoom. For some reason he reported to Acros Security, who are those 0patch guys, the guys that create the little, quickie, really very small patches of components of the Windows operating system and do so immediately, often beating Microsoft to an official patch by weeks, sometimes even longer. So Acros, the 0patch guys, knew about this. They confirmed and reproduced the problem, including creating a

proof-of-concept demo video of this. Then they privately and responsibly disclosed the problem to Zoom.

So last Thursday, July 9th, 0patch blogged about the discovery. They said: "Earlier this week a security researcher shared a remote code execution 'zero-day' vulnerability" - and I have "zero-day" in quotes for a reason I'll come back to in a moment. They said: "...a zero-day vulnerability in the Zoom Client for Windows with our team. The vulnerability allows a remote attacker to execute arbitrary code on a victim's computer where the Zoom Client for Windows," and, they said, "any currently supported version is installed by getting the user to perform some typical action such as opening a document. No security warning is shown to the user in the course of attack.

"The researcher, who wants to keep their identity private, stated that they did not report the vulnerability to Zoom either directly or through a broker, but would not object to us reporting it to Zoom. We analyzed the issue and determined it to only be exploitable on Windows 7 and older systems. While Microsoft's official support for Windows 7 ended this January, there are still millions of home and corporate users prolonging its useful service life with Microsoft's extended security updates or with 0patch," which of course is their product.

They said: "We documented the issue, along with several attack scenarios, and reported it to Zoom earlier today along with a working proof of concept and recommendations for fixing. Should a bug bounty be awarded by Zoom, it shall be waived in favor of a charity of the researcher's choice. On the micropatching side," they said, "we were able to quickly create a micropatch that removes the vulnerability in four different places in the code. The micropatch was then ported from the latest version of the Zoom Client for Windows," which is 5.1.2, "to previous five versions back to 5.0.3 released on May 17th of 2020.

"The Zoom Client features a fairly persistent auto-update functionality that is likely to keep home users updated unless they really don't want to be. However, enterprise admins often like to keep control of updates and may stay a couple of versions behind, especially if no security bugs were fixed in the latest versions." And they said, parens, "(which is currently the case)."

They finished: "Our micropatches have already been released and distributed to all online 0patch Agents. Zoom users with 0patch installed are therefore no longer affected by this issue. According to our guidelines, we're providing these micropatches to everyone for free until Zoom has fixed the issue or made a decision not to fix it. To minimize the risk of exploitation on systems without 0patch, we're not publishing details on this vulnerability until Zoom has fixed the issue, or made a decision not to fix it, or until such details have become public knowledge in any other way."

And then, in an update to that blog posting yesterday, Monday, they amended the posting to add: "Update 7/13. Zoom only took one day to issue a new version of the Client for Windows that fixes this vulnerability, which is remarkable. We've reviewed their fix and can confirm that it efficiently resolves the vulnerability. With an official vendor fix available to all users, we have made our micropatches for this PRO-only according to our guidelines." And "PRO-only" meaning that paid for 0patch.

"Meanwhile, after issuing micropatches for this issue targeted at Zoom clients for Windows versions 5.03 to 5.12, we noticed a lot of our users being on all of these versions despite Zoom's highly persistent update mechanism." Which, as an aside, I think it's interesting that for whatever reason people are not staying current, despite the fact that Zoom is pushing people. Although the oldest one, 5.03, is only May, so it's not like it's really old, but still. They said: "We had expected most users to be on version 5.1.2, but this indicates many users may still be on even older Zoom Client versions. We

therefore ported our micropatch to the remaining supported versions of the Zoom Client back to 5.0.0, .0.1, and .0.2." They said: "We're now covering all vulnerable supported clients."

So for our listeners we want to make sure that they're keeping their Zoom clients current. I'm not a Zoom user, so I haven't experienced what's going on with updating. But for what it's worth, you want to be on 5.1.3, I think, or later, since Zoom has an aggressive auto-update policy in place and a facility. I don't understand why people aren't already clear. The 0patch guys have a bunch of Q&A on their blog posting about the vulnerability, and I put a link to the blog in the show notes. So anyone who's interested can check out the link.

But I'll also note that, as an industry, we appear to be suffering a bit of definition drift. This was covered by the tech press as a zero-day. Now, maybe that's because the tech press just covered what the 0patch people said, and I guess the 0patch wants to sort of call things a zero-day. But a zero-day is defined as a vulnerability that is first discovered as a result of its being actively exploited in the wild. It's considered of the highest possible importance and priority specifically because it's already loose.

And by the time it's first seen, it's under exploitation. And so these 0patch guys came along because they wanted to fix it immediately, which is cool. They were zero-patching zero-days. But now they're patching things that are not zero-days, and they're calling them zero-days, which doesn't make them zero-days. So I'm seeing this in the tech press, like that here, as far as we know, a vulnerability which was identified but has never been abused even once, is being called a zero-day. But it's not. It's just a discovery, responsibly reported and remediated before, as far as we know, it was ever abused. That's the way all regular vulnerabilities are discovered and reported responsibly; right? They're not zero-days.

So that's the best case, and it's happening all the time. That's what HackerOne has made their business model. So since the term "zero-day" has a very specific and important meaning, I'm going to work not to promulgate this misuse of the definition. And I hope that as an industry we will keep from being over, like, overreactive in our responses to this because it's a useful and good definition that I think we need to hold to.

We have a revision of the EARN IT Act bill, take two. We'll recall that the EARN IT is the rather tortured acronym for Eliminating Abusive and Rampant Neglect of Interactive Technologies Act of 2020. It's been revised just a couple days ago in an attempt to collect presumably the votes necessary to get it passed through Congress. Although it's not clear to me how it achieves that. I struggled through a bunch of legalese and mumbo jumbo in an attempt to distill what has changed about the bill in its amendment. And as near as I can see, what the bill has changed is definitely not for the better.

The original bill would have impaneled a 19-person federal commission, predominantly seeded with law enforcement representatives who would have had the power to determine whether this or that website was "in compliance." And of course that phrase is chilling. And what they would be determining was whether the website was in compliance with the committee's determination of best practices, whatever that was - which, if they were in compliance, if the site was in compliance, would entitle them to retain the legal protections that all websites currently enjoy and depend upon under Section 230 for hosting user-provided content. But that was before. Now, this 19-person federal commission has been abandoned. With the new bill, handing over - oh, my goodness.

Leo: Let me guess. It's just Bill Barr now. Just Bill Barr. That's all.

Steve: Well, actually he's in the picture still, but it's even worse. It hands over the power to the 50 individual states of our union. Lord help us.

Leo: Are they trying to make this more unworkable? Is that what they're trying to do?

Steve: Oh, Leo. And I'll just note that we've seen how well allowing each state to determine its own course of action has worked with what has become of the COVID-19 catastrophe, fiasco, mess. So yes, let's just let every state decide to do what it wants. Under this newly revised bill, should EARN IT ever become law, individual state lawmakers will be able to create new laws allowing private lawsuits and criminal prosecutions against Internet platforms as long as they say that their purpose is to stop crimes against children.

The EFF summarizes the intent of Section 230 as follows. The EFF wrote: "The whole idea behind Section 230 is to make sure that you are responsible for your own speech online, not someone else's. Currently," they said, "if a state prosecutor wants to bring a criminal case related to something said or done online, or a private lawyer wants to sue, in nearly all cases the prosecutor has to seek out the actual speaker. They can't just haul a website owner into court because of their user's actions. But that will change if EARN IT passes."

They said: "Section 230 protections enable the Internet as we know it. Despite the politicized attacks on Section 230 from both left and right, the law actually works fine. It's not a shield for Big Tech. It's a shield for everyone who hosts online conversations. It protects small messaging and email services, and every blog's comments section. Once websites lose Section 230 protections, they'll take drastic measures to mitigate their legal exposure. That will limit free speech across the Internet. They'll shut down forums and comment sections, and cave to bogus claims that particular users are violating the rules, without doing a proper investigation."

They finish, the EFF says: "We've seen false accusations succeed in silencing users time and again in the copyright space, and even used to harass innocent users. If EARN IT passes, the range of possibilities for false accusations and censorship will expand."

They also said, later they said: "When we say the original EARN IT was a threat to encryption, we're not guessing. We know that a commission controlled by Attorney General William Barr will try to ban encryption because Barr has said many times that he thinks encrypted services should be compelled to create backdoors for police. The Manager's Amendment, approved by the Committee today, doesn't eliminate this problem. It just empowers over 50 jurisdictions to follow Barr's lead in banning encryption."

So with the amended bill, it will only take one state to inspire a wave of prosecutions and lawsuits against online platforms. And just as some federal law enforcement agencies have declared they're opposed to encryption, so have some state and local police. The previous version of the bill suggested that if online platforms want to keep their Section 230 immunity, they would need to "earn it," as we know, by following the dictates of an unelected government commission. But the new text doesn't even give them a chance.

The bill's sponsors simply dropped the "earn" from EARN IT. Website owners, especially those that enable encryption, will no longer be able to "earn" their immunity from liability for user content under the new bill. They'll have to defend themselves in court, as soon as a single state-sponsored prosecutor, or even just a lawyer in private practice, decides

that offering end-to-end encryption was a sign of indifference towards crimes against children. And that's the way it's probably going to play out. So we'll see what happens.

It turns out that stalkerware is becoming more of a thing and is on more people's radar, which actually is probably good. Stalkerware has grown to become enough of a problem that Google, Apple, AV makers and even the Federal Trade Commission have been pushing back on this growing application segment. The term "stalkerware" refers to spyware apps designed to allow an abusive partner in a relationship to spy on their significant other by installing spyware onto the other's smartphone without their knowledge or consent. Sadly, stalkerware is also sometimes referred to as "spouseware."

It turns out that, although it's not highly popular, its use has been steadily increasing this last decade, largely enabled by the proliferation of smartphones that are able to host said malware, or grayware at least, in another person's phone. It allows partners to keep tabs on their partners at all times by tracking the phone. And the ready availability of these stalkerware products in official app stores has increased the visibility of them, opening them up to millions of potential users. According to statistics gathered by Kaspersky, the number of users who had stalkerware-like apps installed on their Android devices increased from 40,386 known devices in 2018 to more than 67,500 one year later. So better than a 50% increase in 2019. So the market's not massive, but it is slimy.

According to independent antivirus testing lab AV-Comparatives and the EFF, detection rates for stalkerware applications on Android and Windows devices have slowly improved as the issue is gaining more press coverage, and security vendors are moving in to address the growing risk that this is creating. And we've touched on this in the past. The apps are not technically malware. They sort of fall into that gray zone where they're benign, but they're slimeware, I guess you could call it.

So against this backdrop we have Google's announcement just last week of an important coming update to what they call their "Enabling Dishonest Behavior" policy for advertising across apparently all of their properties. And they basically created a one-month notice or notification. They said: "In August 2020" - so next month - "the Google Ads Enabling Dishonest Behavior policy" - is that an acronym for - GAEDB. No, I guess they didn't even try. "Google Ads Enabling Dishonest Behavior policy will be updated to clarify restrictions on advertising for spyware and surveillance technology. The updated policy will prohibit the promotion of products or services that are marketed or targeted with the express purpose of tracking or monitoring another person or their activities without their authorization. This policy will apply globally, and we will begin enforcing this policy update on August 11th, 2020.

"Spyware and technology used for intimate partner surveillance, including but not limited to spyware and malware that can be used to monitor texts, phone calls, or browsing history; GPS trackers specifically marketed to spy or track someone without their consent; promotion of surveillance equipment (cameras, audio recorders, dash cams, nanny cams) marketed with the express purpose of spying is covered. This does not include private investigation services, or products or services designed for parents to track or monitor their underage children. Violations of this policy will not lead to immediate account suspension without prior warning. A warning will be issued at least seven days prior to any suspension of an account."

They said: "Please review this policy update to determine whether or not any of your ads fall in scope of the policy; and, if so, remove those ads before August 11th, 2020." So I guess they're not saying they're going to remove the apps themselves, but they're just going to say no to any advertising of these sorts of things globally, across all of their products. So I say yay to that.

Leo: On we go with the show.

Steve: So I don't know what to think about Internet appliances from China that are having a problem. We have a Chinese Internet equipment vendor very much in the hot seat. And it's not as if it's not possible, just as possible, for domestic vendors in the U.S. to deliberately plant backdoors in their products. As we know, the security industry remains quite suspicious of our own American investigative services, the NSA and the CIA. We have quite strong circumstantial evidence that both organizations are developers of powerful computer subversion and surveillance technology. I think all large nation-states have such.

And to this day we suspect that the NSA may have influenced the design of the default source of entropy used in RSA's earlier BSAFE crypto API. And they didn't influence it in a way that made it any stronger, by the way. So it's certainly not fair, I think, to paint all of China and anything that comes out of China with a broad red brush, when we discover that a particular Chinese vendor is apparently shipping very high-end networking equipment containing multiple apparently deliberate backdoors.

Leo: And it's not Huawei.

Steve: And it's not, exactly, it's not Huawei.

Leo: What?

Steve: Although now I guess they may be, too. But this one we have absolute proof of. This is not, like, whoops. So without drawing any conclusion or rendering any judgment, here's the story. Last week two security researchers, Pierre Kim and Alexandre Torres, very clearly documented their discover of seven vulnerabilities in the firmware of what's known as FTTH OLT devices - I'll explain that in a second - manufactured by the Chinese equipment vendor C-Data, C hyphen Data. And these are not consumer products, so it's not a company name known to us. But apparently it's very popular stuff. There's a lot of their equipment around.

FTTH stands for Fiber to the Home, and OLT stands for Optical Line Termination. So taken together, FTTH OLT refers to networking equipment that allows ISPs to bring fiber optic cables as close to the end users as possible, transiting that so-called "last mile" to our doorstep. These devices are the termination on a fiber optics network. They convert data from an optical line into an Ethernet cable connection that's plugged in at the end user's home, in datacenters, or business centers. And they appear all over an ISP's network due to their crucial role. So this makes them one of today's most widespread types of networking devices at the high end. And they're situated in millions of network termination endpoints all over the globe, not necessarily C-Data's, but these sorts of things, these FTTH OLT devices.

So Pierre and Alexandre confirmed the vulnerabilities by performing a static code analysis of the latest firmware running on two of C-Data's devices. But due to the common internal architecture shared by the entire similar family of devices, they believe that the same vulnerabilities impact 27 other of the company's FTTH OLT models, which run the same or very similar firmware. So a total of 29 different model devices. Their full report is up now on GitHub, and I've got a link in the show notes for anyone who wants more.

But the seven vulnerabilities are about as bad as it gets. By far the worst and most disturbing of the seven is the presence of telnet backdoor accounts hardcoded into the firmware. Yeah, you heard that right: backdoor telnet accounts hardcoded into the firmware. Or as they say, that's not a bug, it's a feature. And yes, it gets worse. The telnet accounts are on the WAN side interface. They allow anyone who knows the hardcoded credentials to connect to the device using any telnet client running on the device's WAN interface, that is, a telnet client on the Internet that is able to access the WAN interface.

Pierre and Alexandre said that the accounts granted intruders full administrator command line interface (CLI) access. Through their static analysis of the firmware of the two devices, they uncovered four username/password combinations hidden in the C-Data firmware. Those are, and I'm sorry to be, like, these are known because, as we'll get to a minute, they did not - they have not disclosed this responsibly, which makes this a whole lot worse. So username suma123, password panger (P-A-N-G-E-R) 123; username debug, password debug124; username root, password root126; username guest, with an empty password, believe it or not. And this initial backdoor command line interface then allows access to exploit additional vulnerabilities.

For example, they said that an intruder could also exploit a second bug to list credentials in cleartext through the telnet command line for all other device administrators, that is, the ones that are officially created. The credentials could then be used at a later point in case the backdoor account was removed in a subsequent firmware update. A third vulnerability allowed the attacker, or just like the interloper because it's not much of an attack when you log in with an open telnet port, to execute shell commands with root privileges from any command line account.

A fourth bug was discovered in the same telnet server, running on the WAN interface. The researchers said that this server could be abused; that using this additional bug, the server could be abused to crash the FTTH OLT device. Since the server was running by default on the WAN interface, this bug could be used to sabotage an ISP's network if they're not filtering and blocking incoming traffic upstream of the FTTH OLT devices.

But the devices also run a web server - of course they do - that's included to power the device's management web interface. Here, they found a fifth bug. By downloading six text files from this web server, an attacker could get their hands on cleartext account credentials for the device's web interface, the telnet server, and the SNMP management interface. And in case any of the passwords are in an "encrypted," and let's put that in air quotes, format, you'll see in a minute that's not a problem, either, because all credentials are secured by XORing them against a fixed known string. That string is in the show notes: *j7a(L and so on. It's in the show notes. So if you were to retrieve an encrypted password, not a problem, just XOR it with this string, and you will see what the user originally typed in.

And last, but not least, the two researchers pointed out that all management interfaces on the tested devices ran in cleartext mode. In other words, HTTP, not HTTPS; telnet instead of SSH, and so on. They said this opened devices and the ISPs that used them to easy man-in-the-middle attacks. Of course, because anybody who was able to sniff any traffic would see everything going back and forth, unencrypted telnet logons, even if they were being used.

So on top of all this, which is really bad news for anyone who has purchased any of these devices anywhere in the world, as we know, responsible disclosure is the norm. But Pierre and Alexandre are sure that they cannot explain what they have uncovered as inadvertent mistakes. And frankly, I would have to agree with them. How do you possibly explain this? So they published their complete and detailed findings without notifying the vendor because the nature of what they have found, they felt, could only be explained as

a deliberate backdoor functionality, intentionally placed into the firmware by the vendor. This is not something that is, like, on and can be turned off. These telnet strings are embedded in the firmware.

They also noted that identifying all vulnerable devices may be a problem for ISPs, as some of the vulnerable equipment appears to have been sold as white-labeled product under different brands, including OptiLink, V-SOL CN, and BLIY, and perhaps others. Now, I sympathize with their presumed outrage and their feelings. But doing what they did is still hugely irresponsible. Disclosure is not made responsibly to protect the vendor of the malfunctioning software or hardware. Disclosure is made responsibly to protect the users of the malfunctioning product.

So no matter whether or not this company's products may have been deliberately backdoored, and I would have to agree it sure does look like they have been, the only responsible thing to do is to inform them in private of what has now been found and give them a fixed hard deadline to update their products' firmware, remove all discovered problems, and send out urgent notices to all their customers and their OEMs, their resellers, informing them of the urgent need to update the firmware. And then, whether or not the company complies by the deadline, only then, after having given them a reasonable chance of fixing this and getting the patches out to all of their customers who are vulnerable, only then go public with the details.

Instead, what they have done really does create a huge mess. Now we have presumably, who knows how many tens of thousands of pieces of high-end networking gear spread around the world, located in crucial networking positions, that cannot be readily taken offline. And if there are 29 different makes and models of these things, there are probably some that are very old. So who knows how much management they're even getting any longer? But still, I mean, this was a bad original sin. But going public with this is, no matter the outrage, can never be responsible. It's really irresponsible.

So now everyone, all the bad guys in the world know exactly how to exploit these devices. And the devices, it doesn't take any cleverness. They're all immediately exploitable. I imagine there will be scans across the Internet for telnet ports using these newly released credentials to see what that is publicly available can be found. And if they're not exposed, if some of these things are not exposed to the public, then they're on internal LANs, and there will be new scans for these usernames and passwords. You know, these guys could have been heroes, but that's not the path that they chose.

So this sort of does bring us to an important point, which is how do purchasers of networking equipment know what they are getting? It is a huge temptation to purchase an inexpensive piece of equipment from a foreign supplier who may be offering a lot of functionality at a very attractive price. But first of all, I would be a little skeptical of the fact that it isn't using HTTPS, and it isn't using SSH. I mean, it does have the feeling of a bargain basement piece of equipment.

Again, I don't have any sense of scale for how widely deployed these 29 different model number devices may be. But, boy, you know, buyer beware in this instance. Wow. And again, yes, it's a Chinese vendor. Maybe factor that in as a signal in a complex decision about where you want to buy your equipment. As you said, Leo, it's not Huawei. But it's C-Data, and we can now prop them up against Huawei and say, well, okay. Well, I don't really remember whether Huawei has been found definitively doing something wrong like this, or just having a strong suspicion of being a bad actor. But wow.

Leo: It depends who you ask. A lot of governments say Huawei is as bad as this. But yeah, depends who you ask. But the problem - so are you better if you buy American made? And can you even get American made? Where do you get that?

Steve: That's just it. Cisco's been having a huge bunch of problems recently. They've been having remote code executions and emergency patches. Nothing this egregious. I mean, this is just in-your-face awful.

Leo: Well, and that's the question. Is it error or intentional; right? It's hard to...

Steve: This has to be. This has to be deliberate.

Leo: This looks intentional, yeah.

Steve: You don't embed telnet access, undocumented telnet usernames and passwords in the firmware without it, I mean, without it. And they could say, oh, it was just meant for, you know, we shipped debugging kernels by mistake. We never meant to do that. Okay.

Leo: I mean, there'd be more subtle ways to do it if you really wanted to do it than putting a telnet in there. But I don't know. I don't know. I don't know. You know what, encrypt everything. Trust no one. Encrypt everything that goes over the network. Zero knowledge. Zero trust.

Steve: We all know what a problem locating the sources of remotely piloted consumer drones can be when they're operating illegally in the vicinity of a commercial airport. Those little guys, those little drones can and have caused serious disruptions in aviation because a drone cannot be allowed to strike the intake of a jet engine. That would be very bad. Birds are bad. Drones are worse.

So as with anonymous postings on the Internet, drone operators have been able to operate under the assumption that they cannot be caught because they cannot be located. They're able to operate a good distance away. And with today's camera-equipped drones, they no longer even need line of sight to their little vehicles. But if they could be reliably located, the word of mouth would spread, and the problem would be largely resolved. A few highly publicized arrests and the game would change.

So far, the only solution that has been applied is sort of the obvious one. Surround locations where drones pose a real hazard with radio receiving stations tied back to a central control point, and attempt to use the radio frequency emissions from a drone operator's transmitter to geolocate that device. Now, in an empty field in the middle of Nebraska, that might work well. But a busy commercial airport in any modern urban center turns out to be a very different problem. The air is hugely contaminated, not only with avionic radio frequencies, including high-power radar, but also the typical ground clutter that you have with cellular phones, WiFi, Bluetooth, IoT, and all manner of other radio frequency generating equipment. And to further complicate matters, drone operation radio signals can have short duration, their frequency usually hops over most of the band, and they have relatively low power.

So Leo, to address this problem, our industrious academics from Israel's Ben-Gurion University of the Negev, who we're often referring to, have been performing some research, and have just publicized a new paper titled, "Can the operator of a drone be located by following the drone's path?" For anyone who's curious, I have a link in the show notes.

They said that, well, so far - I did read enough of it to get a sense for it. I won't go into great detail, but I'll summarize that their work so far has been with simulations. Their underlying supposition appears to be holding. The path of a drone being remotely controlled by someone from a distance will inherently contain clues about the location from which that person is viewing the drone. Just a simple example is that the drone's motion along the line of sight to the viewer will be much less obvious to that viewer than any motion perpendicular to their line of sight to the drone. So it's reasonable to suppose that this might influence the drone's path.

Unfortunately, this system assumes that the drone is being piloted visually from a remote location and not using an FPV drone-mounted camera. So the application might be somewhat limited in practice. But they have trained up neural networks and have been able to predict with 78% accuracy the operator's remote location knowing only the drone's flight path. So I thought that was pretty cool. I just wanted to share this since I thought it was interesting and clever and not at all immediately obvious, and also since it was new work from our industrious academics at the Ben-Gurion University of the Negev.

We have a report from Fraunhofer Institute.

Leo: Oh, yeah, this thing was terrifying.

Steve: I know. And ridiculous, unfortunately.

Leo: Oh. Oh, good.

Steve: Yes, yes, that's the good news.

Leo: Okay.

Steve: So the report's title was "Rampant Router Insecurities." And it's not what it appears. The tech press has been hyperventilating over a 25-page report from, as I said, the Fraunhofer Institute "Home Router Security Report 2020." So just the executive summary. It's pretty quick, gives you a taste for this. And this is, of course, what the tech press jumped on.

They wrote: "This report analyzes 127 different routers for private use developed by seven different large vendors selling their products in Europe. An automated approach was used to check the routers' most recent firmware versions for five security-related aspects. We were able to extract completely 117 of the 127 firmware images. Four firmware images could be extracted partly, and six firmware images could not be extracted at all. 116 of 127, that is, 91% of the devices are powered by Linux. One was powered by ThreadX and another by eCos.

"The security aspects addressed in this report are: When were the devices updated the last time? Which operating system versions are used, and how many known critical vulnerabilities affect these operating system versions? Which exploit mitigation techniques do the vendors use, and how often do they activate these techniques? Do the firmware images contain private cryptographic key material? Are there any hard-coded credentials?"

And they said: "Our results are alarming. There is no router without flaws. 46 routers did not get any security update within the last year. Many routers are affected by hundreds of known vulnerabilities. Even if the routers got recent updates, many of these known vulnerabilities are not fixed. What makes matters worse is that exploit mitigation techniques are used rarely. Some routers have easily crackable or even well-known passwords that cannot be changed by the user. Most firmware images provide private cryptographic key material. This means whatever they try to secure with a public-private crypto mechanism is not actually secure at all.

"Nonetheless," they wrote, "vendors seem to prioritize security differently. Especially AVM" - that must be a European manufacturer - "does a better job than the other vendors regarding most of the security aspects. However, AVM routers are not flawless. ASUS and Netgear do a better job on some aspects than D-Link, Linksys, TP-Link, and Zyxel.

"To sum it up," they said, "much more effort is needed to make home routers as secure as current desktop or server systems. Additionally, our evaluation showed that large-scale automated security analysis of embedded devices is possible today. We used the Firmware Analysis and Comparison Tool (FACT), and it worked very well for almost all firmware images analyzed during this study. FACT is an open source software available on GitHub." And I should mention that it was developed by Fraunhofer. The "F" of what is now Firmware Analysis used to stand for Fraunhofer before they decided to open it to the public.

So what does this mean? Is it good? No. Is it the end of the world as we know it? Also no. Everyone knows that my blanket recommendation would be to use a current build of pfSense, which is built on top of state-of-the-art FreeBSD, and load it onto a little fanless multiport appliance PC. You get a super solid platform that's not being continually updated because there are no known problems. If there were, they'd be fixed. And the darn thing will do anything you might wish for.

Okay. So first of all, the comparison with a consumer PC is not fair because a Windows or a Mac or a Linux, it has to be secure against all the random application crap that people run on that OS. Thank goodness a router is only running its own fixed firmware. It is not a host to randomly added consumer software, or it would be the end of the world as we know it. But for what it's worth, it at least is an appliance. Okay, so it's not pfSense running FreeBSD. Everyone already has something that they like. They've got a router that they're using.

So what I'll note here is that the one thing Fraunhofer failed to mention, unless any of these 127 routers is misconfigured to expose vulnerable open ports and services to the public Internet, which could happen, so let's assume that isn't the case. And that isn't what they're looking at anyway in this study. All of these vulnerabilities are internal and accessible only from the LAN side of the router, not the WAN side. Okay. So a router has a hard-coded private key. That's not as cool as if it generated a random key the first time it was turned on so that then every router of that make and model and firmware edition would be uniquely keyed. But it's not the end of the world, either.

That's not to say that this is good. But remember, we were recently talking about a router vulnerability that was so egregious that hostile code running on a web page in the user's browser could be used to reach out to the router and cause some trouble. But in general, once you have hostile code running loose inside your network, things are already bad. Note that all the vulnerabilities cited by Fraunhofer require access to the router.

So one way to secure the router would be to block access to the LAN side management services except from, for example, one specific LAN IP. With pfSense, that's trivially done

through its web-based UI. I don't recall seeing that generally available in consumer routers. But if it were, use it. If you were to block management to a specific IP, no system on the LAN could access the router's management, period. You could have a separate pokey old laptop that's no longer useful for much else as your router management PC. Its IP would be hard-coded, and it would be your router management laptop that's normally turned off and in a closet.

With pfSense or some other advanced router, you could even give the LAN interface a second IP on a different private network, like a 10-dot. That way the router's management would not even be on the same network. It wouldn't be in the same Ethernet broadcast domain as any of the other services on the LAN. Or you could use VLAN tagging and place a cheap little VLAN-aware smart switch in front of the router to block any access to the management traffic's VLAN that wasn't tagged with the proper VLAN tag. And of course that would only work for wire traffic. Oh, I guess, you know, you could do VLAN over WiFi, come to think of it. So there's another solution. Bind the management to a wired port or to a VLAN WiFi that cannot be seen by management.

So anyway, you get the idea. Many solutions are available, depending upon your network and its configuration. It's true that consumer routers are a problem. That's worth keeping in mind. Older routers like older smartphones may no longer be receiving updates, and they may be more of a problem, especially if problems become known that are never going to be fixed in your router's firmware. But the problem is not existential, and a bit of thought given to strictly limiting all inside access, in the same way that the WAN side is strictly limited or it'd be the end of the world, that might just pay off - if nothing else, for your peace of mind.

So it's worth looking at, you know, poking at your router's UI features and see, for example, if you can put a filter on the router's port 80, or hopefully port 443, the TLS connection port. Although I don't know if I've seen a router that supports HTTPS. I think I've seen some that actually do allow themselves to get a cert over the ACME protocol from Let's Encrypt.

Leo: It's a self-signed - it's often a self-signed cert. Yeah. Sometimes they just do a self-signed cert.

Steve: Yeah. And so you get warning notices, and oh my god.

Leo: They say, yes, I am trying to contact this router. So let me in.

Steve: Right.

Leo: And then you've exchanged certs. So from then on it's a secure conversation. You just have to be careful that first time; right?

Steve: Right, right. You say yes, I will trust something from this self-signed cert, the device, yeah.

Leo: I've seen that.

Steve: So a little bit of miscellany, Leo. I just wanted to mention I was grumbling about the price hike of YouTube TV? Well, I uncanceled my subscription to it.

Leo: Oh, really. Sling, you mean.

Steve: Yeah, I was going to move over...

Leo: Oh, you uncanceled YouTube because you were going to move to Sling; right.

Steve: Right. I was going to move to Sling.

Leo: It's not very good, is it.

Steve: And I didn't make it very far, since I really, really, really need the recorded content scrubber to work correctly.

Leo: Right.

Steve: "Working correctly" means that you can pause the content and then jump forward or backward by some amount of time, and you get to see a thumbnail of where you now are. It's critical for the way I watch content. And YouTube TV works exactly like that on my Roku. I get to pause it, and then I can jump forward and backward, typically over the commercial until I see that the show has started again. Then I'll back up one frame and unpause. It's perfect.

Sling TV works in that annoying progressively faster and faster if you push the fast-forward or fast-reverse button either way. But it's totally blind. You can't see where you are. So anyway, no fixed-time jumping, and no, thank you. I just said, well, it would have saved me some money, but it's worth it for me to pay the extra monthly fee in order to get movement through pre-recorded shows the way I think it should be done.

Leo: And a much better selection of channels. I have to say Sling is kind of scant in its offerings.

Steve: Yeah.

Leo: We're just stuck, unfortunately.

Steve: And as you said, it's the standard bundling. It's just we have moved from cable delivery over to streaming delivery.

Leo: It's still cable, basically. It's the same price model, yeah.

Steve: Yeah, maybe someday somebody will challenge the bundlers. Although I guess I don't know how that happens because bundles are bundles.

Leo: Right. It's the content guys who really determine it in the long run. And that's what they want, yeah.

Steve: Yeah. So Leo?

Leo: Yes?

Steve: I was reading a couple days ago about how the Linux team has approved new terminology banning, officially banning terms like "blacklist" and "slave."

Leo: Right.

Steve: And of course we've been touching on this topic of insensitive language and terminology in our technology. And I thought, why not do my part, too? As I've been working on SpinRite's forthcoming technology, I've been thinking about our new "woke" awareness of the challenges faced by those in any highly heterogeneous culture and environment. Even when we're all created equal, we're all still created differently.

So in that spirit, in a spirit of accepting these differences, I realized that labeling a sector as "bad" is really quite harsh. I mean, it's not really a "bad" sector. At the moment it's just checksum challenged. It's error non-correcting, or just having a weak bit. It's not bad, it's just different from its neighboring sectors. And so SpinRite's newly enlightened job will be to have a gentle conversation with the sector. SpinRite will check in with it to see how it's feeling. SpinRite will work with the sector to bring about the change that will be in everyone's long-term best interest. And you know, 4,096 bits is a lot of bits. Oh, my goodness.

So if it should turn out that this particular, otherwise beautiful and perfect little sector just can't get the hang of holding onto every one of those oh, so many bits, well, then SpinRite will find a nice place for it to go so that it just doesn't need to worry about all that any longer. It's a lot, after all. So it will be able to just rest and relax and live out the rest of its drive's life in peace. It will still be there, but a fresh and brand new sector will be taking over, handling all of its bits for it so that it just doesn't need to worry about all of that anymore.

So in the future SpinRite we're not going to have any bad sectors. No. We're just going to have some that SpinRite decides have already worked as hard as they need to, and it's time to just give them a rest and give some brand new sectors that have been waiting all this time their own chance to hold onto all of those bits for their owner. It's 2020, after all, and this really feels much better.

So it's not a Technado, Leo, it's a Tsunami.

Leo: Don's show is a Technado; but this, a Tsunami.

Steve: This is a Tsunami.

Leo: Which is probably just as bad.

Steve: This I think is going to be very interesting, of interest to our listeners and another really, you know, another really good thing that Google has done. I mean, look at Chromium and what that has meant for the industry's browsers. Google has open sourced a vulnerability scanner for large-scale enterprise networks. Which is not to say that it can't be used for smaller networks. But I'll explain why this is, like, well, it's what they've been using. It's for large-scale enterprise networks consisting of thousands or even millions of Internet-connected things. I've got two links in the show notes, both GitHub links because, yes, that's where this thing lives.

Google has named it "Tsunami." They've been using it themselves internally at Google, and it has recently been made available to us all. And I'm sure before long we'll have, like, prepackaged binaries for all of our popular OSes. It's not going to be a Google-branded product; but it will be maintained, further developed, and extended by the open source community, very much in the way Google first made Kubernetes available. And that just became a staple in the industry.

So of course, as we know, hundreds of other commercial or open source vulnerability scanners already exist in the world. The difference here, what Tsunami is doing, is that Google built the scanner with very large enterprise deployment in mind, like its own. I mean, Google uses it. And so this is not to suggest, as I said, that it would not be just as useful for smaller environments, but that it's inherently designed to scale well. And this is critical. It is explicitly designed to absolutely minimize the production of false positive detections, which can be a, well, which unfortunately scale just as the network scales. It turns out that false positives are the bane of IT personnel when traditional scanners are let loose across a large enterprise.

So Tsunami was designed to run inside giant networks where even the slightest false positive findings could result in sending incorrect patches to hundreds or thousands of devices, possibly resulting in crashes, network crashes, countless wasted work hours. So Tsunami is designed instead to gracefully tackle networks which may include hundreds of thousands of servers, workstations, networking equipment, and IoT devices that are connected to the Internet and visible to the scanner, all while providing the highest possible scanning accuracy with an eye toward minimizing false positives.

They said that they designed Tsunami with the ability to adapt to extremely diverse and extremely large networks without the normal need to run different scanners for each device type, as is typically done and is too often necessary. They did this by splitting Tsunami into two logically and operationally separate pieces. The first component is the scanner. They call that the "reconnaissance module." It scans a company's network for open ports and tests each port, attempting to identify the protocols and services running on each to prevent mislabeling ports and testing devices for the wrong vulnerabilities. Its port fingerprinting module was derived from NMAP, you know, the legendary network mapping engine. But then they added to that a bunch of their own new code to the NMAP core.

The second component is the more complex of the two. It takes the first component, the scanner/mapper's output as its input. It takes each located device and its exposed port, selects from a list of vulnerabilities to test, and runs benign exploits to determine whether the device is actually vulnerable to that attack. This is the vulnerability verification module, the primary means through which Tsunami may be extended because it uses a flexible plugin architecture to allow the entire vulnerability verification module to grow over time and evolve through straightforward community extension, and

it will allow security teams to add new attack vectors and vulnerabilities to check inside their own networks.

So you can imagine, for example, when news of this Chinese C-Data fiber terminator vulnerabilities became public, someone might quickly add a vulnerability verifier to Tsunami in order to see whether there are any of those exposed and vulnerable, and to locate any that are. So, for example, you would immediately, out of that public disclosure, you would add those four new username/password pairs, and I don't know even whether it's open on telnet 23 or some other port. But whichever port, you would add that, quickly create a Tsunami module, probably taking an existing telnet vulnerability scanner and just adding a few more lines to it, and then turn it loose. And you may have already, for example, run reconnaissance on your network, so you would have a global network map already established. So you would just then run that new vulnerability module driven by the reconnaissance input against your network in order to immediately locate and start dealing with a newfound problem.

So at the moment, at release time, Tsunami comes with plugins which check for exposed sensitive interfaces. Applications such as Jenkins, Jupyter, and Hadoop Yarn ship with UIs that allow a user to schedule workloads or to execute system commands. If these systems were exposed to the Internet without authentication, attackers could leverage the functionality of the application to execute malicious commands. So that's an example of something that you could use this to make sure you're safe against. And also it checks for weak credentials. It uses other open source tools such as ncrack to detect weak passwords used by protocols and tools including SSH, FTP, RDP, and MySQL.

So with this public release of Tsunami, Google is not by any means stepping away from the project. They plan to continue enhancing it for their own use with new plugins to detect an ever-widening variety of exploits, and those will be contributed back to the community. There'll be a separate Tsunami plug-in repository that they create. And Google said that it will focus on meeting the goals of high-end enterprise clients like itself, and the conditions found in these types of large and diverse multidevice networks.

So anyway, I wanted to put this on our listeners' radar. It looks like a very cool piece of new technology donated by Google to the IT users of the world, for our use. And I haven't had any chance to play with it. But I'm sure we will start seeing Tsunami binaries that'll make it easy for people to play with.

Leo: Cool. Nice.

Steve: Very cool. And thank you once again, Google.

Leo: Google does good stuff. Those security guys especially do good stuff.

Steve: Yeah.

Leo: Steve, we've come to the end of another thrilling, gripping edition of Security Now!.

Steve: 775 podcasts in the can.

Leo: Holy-moly. Only 225 to go. Well, 224 to go. I'm going to talk you into doing an Episode 1000, just without any numbers or something.

Steve: Well, or zero. We never did zero.

Leo: Got to do the zero. It resets to zero. Then we start all over again. We do this show every Tuesday, right after MacBreak Weekly. That's about 1:30 usually. It's going to be a little later often. But 1:30 Pacific is the goal, 4:30 Eastern time. That's 20:30 UTC. Here's the deal. You can watch us do it live. If you go to TWiT.tv/live, there's audio and video streams there.

But you can also get on-demand versions of the show. Steve has them at his website. He actually has some unique versions of it, a 16Kb audio version, for instance, along with a 64Kb, the normal audio. He also has a transcript of every show, which is a really useful tool, either for reading along while you listen, or for searching because you can search right into any show by searching the transcript of that. That's all at GRC.com.

While you're there, take a look at SpinRite, SpinRite 6, the current version. 6.1 on its way. If you buy now, you'll be participating, if you wish, in the beta of 6.1, and you can weigh in on various features. There's a lot of freebies associated with that as he spins off concepts and ideas. That's all at GRC.com. He's also on Twitter. And if you want to leave a question for Mr. G, you see that Twitter handle right there, @SGgrc. He accepts DMs from anybody.

You can also get copies of the show at our site, TWiT.tv/sn. We have video, as well as audio, TWiT.tv/sn. And of course we always encourage you, if you can, to subscribe in your favorite podcast application. That way you don't even think about it. Just whenever you're in the mood for Security Now!, there it will be, ready, queued up to listen to.

Thanks, Steve. Have a great week, and we'll see you next time on Security Now!.

Steve: Thanks, Leo.