TLS PRIVATE KEY LEAKAGE

Description: This week we look back at last week's Patch Tuesday to learn how much better Microsoft various products are as a result. We look at Facebook's announced intention to creep further toward end-to-end encryption in Messenger, and at the puzzling result of a recent scan of the Internet for completely exposed VNC servers. I want to take a few minutes to talk about the importance of planning ahead for a domain name's future, share my tip for a terrific website cloning tool, and a few more updates. Then, after sharing some feedback from our ever-attentive listeners, we're going to address the question: "Can a remote server's TLS private key be derived simply by monitoring a sufficient number of its connections?" What?! We all know that everything has been designed so that's not possible. But edge cases turn out to be a surprising problem, and the details of this research are quite interesting.

High quality (64 kbps) mp3 audio file URL: http://media.GRC.com/sn/SN-884.mp3
Quarter size (16 kbps) mp3 audio file URL: http://media.GRC.com/sn/sn-884-lq.mp3

SHOW TEASE: It's time for Security Now!. Steve Gibson is here. We conclude our 17th year of podcasting with this episode. We start Year 18 next week. We'll look back at Patch Tuesday. How many years have we been doing that? Yes, more Microsoft stuff. We look at Facebook's intention to increase end-to-end encryption in Messenger. It's about time. And then Steve answers a very interesting and puzzling question: Can a remote server's private key be derived simply by monitoring a sufficient number of connections? The answer might surprise you. It's all coming up next on Security Now!.

Leo Laporte: This is Security Now! with Steve Gibson, Episode 884, recorded August 16th, 2022: TLS Private Key Leakage.

It's time for Security Now!, the time to talk about all those things that protect you and your loved ones on Windows and Mac. And there he is, living long and prospering, the great Steve Gibson. Hello, Steve.

Steve Gibson: Yo Leo. Great to be with you. I frightened you a little bit before we began recording when I said that this was the final podcast.

Leo: Yes, don't say that.

Steve: And I needed to follow that up with "of our 17th year."

Leo: Oh, whew. What a relief.

Steve: We have now finished 17 years.

Leo: That's amazing. That is amazing.

Steve: And next week we're going to start to get serious.

Leo: Yes.

Steve: Because we've figured out how to do this.

Leo: Enough of this silliness. Let's do a show, yes.

Steve: No more fooling around.

Leo: Unh-unh.

Steve: So we've only done 883. This will make 884 of these little things.

Leo: That's amazing.

Steve: I know.

Leo: Neither of us had any idea it would go on this long.

Steve: This was all your fault, too, of course.

Leo: I take full blame.

Steve: Leaning against the set up in Toronto, you said, "What would you think about doing a weekly podcast?"

Leo: Hey, Steve. What about this?

Steve: And I remember thinking, oh, I hope he doesn't bring this up again.

Leo: Really? You really didn't want to do it.

Steve: I just didn't know how fantastic it was going to be. It's the best thing that ever happened.

Leo: Oh, okay. So you're glad now in hindsight.

Steve: Oh, I am so glad.

Leo: Oh, I'm relieved, okay.

Steve: Absolutely. Absolutely.

Leo: Because you never told me you didn't want to do it. It's a little late now.

Steve: We're going into 18 years. As many of my various exes have said, oh, he doesn't ever do anything he doesn't want to do. So...

Leo: Eh? That's good. Actually, I like that because then I know I'm not imposing on you because if you didn't want to be here, you wouldn't be. So I appreciate that.

Steve: It is truly the best decision that I didn't make. You made it, and I'm really glad.

Leo: Well, and I agree, I mean, you're the bulwark of the whole podcast network, so thank you. I appreciate it.

Steve: Okay. So a really, really interesting thing. I noticed on MacBreak you covered some of the...

Leo: Oh, man.

Steve: ...security issues that I skipped over.

Leo: Yeah.

Steve: Because they were like, okay, you know, like Zoom fixed their remote code execution vulnerability, and the whole Mac security layers bypass has been resolved. And there was - so of course this is the first podcast after the big Vegas summer - it's considered hacker summer camp, that whole week in Las Vegas with Def Con and BSides and all the other stuff.

Leo: Black Hat, yeah, yeah.

Steve: Black Hat of course. So but the big news came out of the USENIX Security Symposium, I felt. We've talked about several stories already. There was another one which ended up replacing what I sort of thought we might talk about, which was Bernstein's suing the NSA over a Freedom of Information Act because he wants to know where they came up with some of the decisions that they've made. And I said, you know, if something bigger than that happened, we would talk about it. Well, a group of four researchers at UC San Diego and in Boulder, Colorado, two from each University, did some watching of the Internet. And what they found really needs to get our attention.

But first we're going to talk about last week's Patch Tuesday, which also needs to get our attention, to learn how much better Microsoft's various products are as a result. Actually it didn't turn out to be so bad; but, boy, was there a lot that happened. We're also going to look at Facebook's announced intention to creep further, carefully, apparently, with a lot of caution, toward end-to-end encryption by default in their built-in native Messenger app. Also at the puzzling result of a recent scan of the Internet for completely exposed VNC servers, not something I think we've ever spoken of before.

I also want to take a couple of minutes to talk about the importance, and I'm sure you're going to have fun chiming in, Leo, the importance of planning ahead for a domain name's future. Several things have happened in my off-the-podcast life that sort of brought this to my attention, and I thought I just should take a minute to talk about that. Also I want to share my tip for a terrific website cloning tool and, additionally, a few more updates. Then we've got some feedback from our ever-attentive listeners.

And we're going to wrap by addressing the question, believe it or not, can a remote server's TLS private key, which is the thing they guard more than anything else, I mean, how many hours of podcast time have we talked about the issues surrounding protecting the private key, warning people if it leaks out, how do we revoke it, I mean, just like nothing more important. Can a remote server's TLS private key be derived simply by monitoring a sufficient number of its connections? What? We all know that everything has been designed around that not being possible. But edge cases turn out to be a surprising problem, and the details of this research are quite interesting.

Leo: Ah.

Steve: So today's topic is TLS Private Key Leakage, and we do have a fun Picture of the Week. Somebody sent this who knows some of my gripes about graphing. So I think another great podcast for our listeners.

Leo: Yeah, we've griped about graphs with no origin and all, you know, every Apple event I have to gripe again about those. So good. It'll be fun. I look forward to it. And Mr. G. and the hard work he does to keep us save and secure. And laughing with our Picture of the Week.

Steve: Yes, indeed. So we've often complained. My biggest gripe is when something is trying to show like how big something has gotten or small something has gotten, like the amount of change. And it looks like, oh, like a lot. And then you realize, wait a minute, the scale goes from 111 at the bottom to 112 at the top. And it's like, what? So if you put it on a zero-based scale, you couldn't tell the difference between the two bars.

Leo: Exactly, yes.

Steve: Which is just, you know, it's cheating. Anyway, this is just a beautiful piece. And is this xkcd?

Leo: It sure looks like Randall's work, yeah.

Steve: Yeah. I thought that was his style. And one of the things, I think one of the reasons his work appeals to techies is his attention to detail. And this has that. So this is a chart of energy, fuel energy density in megajoules per kilogram. And it's got five bars because he's got five different sources of energy. We've got sugar at 19 is the lowest bar. Then coal at 24 is the second. Fat at 39 megajoules per kilogram is in the middle as the third bar. Gasoline is up at 46. But then we hit uranium. Now...

Leo: Properly handled, I might add.

Steve: Yes.

Leo: I mean, you could say the same thing about hydrogen, but it takes a little to get that powered.

Steve: It does. So uranium has a graphed megajoules per kilogram. Whereas the others are 19, 24, 39, and 46, uranium is 76 million.

Leo: Yes.

Steve: Now, okay. Now, we have a problem if this is a non-logarithmic scale. And so the punchline here, the title of this is Science Tip: Log scales are for quitters who cannot find enough paper to make their point properly.

Leo: That's hysterical.

Steve: And so what we have is we have a - it looks like, oh, it's probably several miles long strip of paper which has been scotch-taped to the top of the uranium bar, which of course immediately shot off the top of the chart because you've got a chart that shows a modest-sized bar for 46 megajoules per kilogram for gasoline. Well, that chart cannot hold 76 million using a linear scale. But no, this person said, no, I'm not switching to logarithms. I'm going to stay linear. So what we have is this insanely long piece of ribbon paper which it goes off into the distance, then it fanfolds itself about seven feet down to the floor, comes back off the bottom, and the end of it curls up so that we're able to see the top of the bar properly labeling the height of 76 million megajoules per kilogram for uranium. So, yeah. Anyway, another nice little science-y Picture of the Week.

And it put me in mind of the fact that one of the things that I have, and I mentioned on the podcast a couple times because I love the app, is it's monitoring the SMT - not SMTP. What's the - I can't think of the acronym for things - oh, MIBs, M-I-B-S. But there's a name for the counters in networking gear that's - it's escaping me right now. But it's - so the router that I have running pfSense is maintaining total bytes in and total bytes out,

and also rates for counters. And so the app that I've got running in Windows is polling that and graphing it.

And the point is that I've got I think 300 megabits down and 30 or 40 megabits up. Which is plenty for me. But on a linear scale, it's inconvenient because from time to time I will pin that 300 megabits. And if I want to have 300 megabits be the maximum point of the chart when not much is going on, it looks like a flat line. The point is that I have it set to a logarithmic scale which is one of the options this cool graphing tool has. And what that has the effect of is of course compressing the top so that I get relevant-looking graph all the time, regardless of what scale my bandwidth variation is running in. So I'm sure everybody knows logarithms certainly do have their place, although xkcd had fun without it.

Leo: I think that that was his point, is use logarithms. He was being sarcastic.

Steve: Yes.

Leo: Yes.

Steve: Yes, yes.

Leo: Use a log scale, please.

Steve: So after last Tuesday's monster Microsoft patch event, so far it appears that nothing new was badly broken. Which given how much they did is a little bit of a miracle. It was quite a month, in which 121 new patches were delivered. In fact, so many things received updates that it's unclear whether there was anything that didn't. There were identified CVEs in Windows and Windows various components: Azure Batch Node Agent, Real Time Operating System, Site Recovery, and Sphere; Microsoft Dynamics; Edge browser; Exchange Server; Office and its many components; PPTP, SSTP, Remote Access Service PPTP; Hyper-V; System Center Operations Manager; IIS. And of course the Print Spooler didn't escape being fixed some more. And we've got Microsoft Windows Defender Credential Guard.

And on top of all of that there was an additional 17 CVEs which patched Edge, plus another three patches for Secure Boot. So that brought the month's total CVE-patched count to 141. And if you were thinking that that seems like a lot, yes, it's nearly triple the size of last year's August release talk about year-over-year inflation and it's the second largest release of 2022. So this has been a big month.

Of those 121 new CVEs, 17 are critical, 102 were marked as important, one was only moderate, and there was one last one that was low in severity. Two of the bugs are publicly known, with one of those, affectionately known as DogWalk, is or at least was, and actually it will be until everybody gets themselves patched, under active attack at the time of the release.

Not long ago we were talking about the trouble with Microsoft's custom protocol handlers and how they could be readily abused. This particularly critical zero-day remote code execution vulnerability is another one of those. And as before, Microsoft was responsibly informed of this problem and declared that it was not a security problem.

Leo: Oh, were they wrong.

Steve: Oh, my god. And I suppose it wasn't until it was, since now it is, and it's being exploited in the wild to damage Microsoft's users for which, as we know, they take no responsibility because we all give that up when we start using any of their stuff.

Leo: Yeah, it's your fault. It's your fault.

Steve: That's right. That's right. So the trouble is known as "DogWalk" because it's another path traversal flaw, thus walking the directory hierarchy. When a targeted victim opens a specially crafted .diagcab, so it's a file ending in .diagcab, archive, which contains a diagnostics configuration file, it allows an attacker to save their malicious executable file into the - I have a hard time even saying this - into the user's Windows Startup folder. Which, you know, is really not where you want attackers' malicious executable files to be stored on your computer.

Leo: Well, at least it's easy to find. I mean, they're not hiding it.

Steve: Yeah. So of course the next time this hapless victim logs into the system after a restart, that file will dutifully be automatically run by Windows. And this vulnerability affects all Windows versions starting from 7 on and from Server 2008 and on.

So, okay, get this. The problem was originally, as I said, responsibly reported to Microsoft on December 22nd, 2019 by security researcher Imre Rad. Even though a case was opened by Microsoft the next day acknowledging the receipt of his report, six months later Microsoft declined to fix the issue. They told Rad that to make use of the attack, an attacker would need "to create what amounts to a virus" - okay - "convince a user to download the virus, and then run it."

Leo: Yeah?

Steve: Yeah, gee, like maybe some malware. So Microsoft said that "This wouldn't" - I'm quoting - "this wouldn't be considered a vulnerability." Okay?

Leo: Huh?

Steve: They said: "No security boundaries are being bypassed," meaning we designed it to work this way.

Leo: Well, then, okay. It's supposed to be doing this.

Steve: It didn't bypass anything. It's working the way it's supposed to. And they said: "The proof of concept doesn't escalate permissions in any way" - okay, I guess it didn't need to, but it would be bad if it did, but it doesn't. Or, they said, "do anything the user couldn't do already." Because, you know, users are perfect, and they never click something that they shouldn't. So what's the problem? Right.

So apparently because the proof, by their logic, because the proof of concept was only a proof of concept and not an explicitly and massively destructive bug, and since in a sense it was by design, Microsoft said not a problem. In fact, they wrote: "There are a number of file types that can execute code in such a way but aren't technically 'executables.'" What? "And a number of these are considered unsafe for users to download/receive in email."

Leo: Even there.

Steve: "Even .diagcab." In other words, again, this is what we told you it was going to do. And they said: "Even .diagcab is blocked by default in Outlook on the web and other places." Wow. So okay. So much for layered security and defense in depth. They're saying, well, Outlook blocks it. So what's the problem? Okay. Back then - I know. Microsoft essentially said that, yes, these sorts of files are indeed unsafe for users to download and receive, and they should not do that. But don't worry, our email client, Outlook which as an aside has recently been crashing so badly for many users that it has been unusable, but that's a different story that we'll get to in a minute they said, our email client, Outlook, blocks these by default.

Consequently, despite Microsoft's assurances 2.5 years ago, today we have CVE-2022-34713 identified as a critical zero-day remote code execution vulnerability. And this is exactly what that is. Which they said, no, this is what we wanted. Okay. Since that was finally fixed this month, last week with August's patches, it would appear that Microsoft's position on allowing attackers to execute their code, uninvited, on a victim's Windows machine has since changed, and not a moment too soon, though about 2.5 years too late. And of course many people were hurt in the meantime.

This CVE leverages Windows Support Diagnostic Tool (MSDT). And it's not the first time an MSDT bug has been exploited in the wild just this year. We previously talked about this. This bug allows remote execution when MSDT is called using the URL protocol from a calling application, typically from Microsoft Word. So perhaps Microsoft missed that other way to invoke this very dangerous diagnostic feature. Wasn't just Outlook. Word could do it, too.

Leo: Well, even worse, this is exactly the kind of thing those evil call center guys do.

Steve: Yes.

Leo: They would say, oh, you know, we see you have problem on machine. Please launch diagnostic tool. Yeah. That seems sensible; doesn't it. Geez.

Steve: What could possibly go wrong?

Leo: What could possibly go wrong? They're trying to help me.

Steve: Yeah. So exactly as you're saying, there's an element, Leo, of social engineering to this because a threat actor would need to convince a user to click a link or open a document.

Leo: Yeah. But how harmful could the Microsoft diagnostic tool be?

Steve: Well, yeah. That means we have nothing to worry about because all users are smart, and they would never click on a link or open a document that they weren't supposed to, you know, even if their mother sent it to them. Or because their mother sent it to them. Anyway, okay. Recall that we were also just talking about files downloaded from the Internet being tagged with, I love this, the MOTW, the Mark of the Web. The idea being that this would cause Windows to refuse to proceed or to at least issue a pop-up warning. Well, we would not want that to confuse people using Microsoft's Support Diagnostic Tool which fields these .diagcab files. So it was designed not to check for the Mark of the Web so that its files can be opened without any warning.

0patch's founder told The Hacker News that: "Outlook is not the only delivery vehicle. Such files are cheerfully downloaded by all major browsers including Microsoft Edge by simply visiting a website. And it only takes a single click, or mis-click, in the browser's downloads list to have it opened. No warning is shown in the process, in contrast to downloading and opening any other known file capable of executing an attacker's code." Wow. That was fixed, finally.

The infamous SMB Server Messages Block protocol is back with another remote code execution vulnerability. This one's a server side bug which allows a remote, unauthenticated attacker to execute code with elevated privileges on affected SMB servers. That's not good. But there is some good news. Only Windows 11 is affected.

Leo: Oh, great. That's what I use.

Steve: Oh. I had here in my notes only the five people using Windows 11.

Leo: Six now.

Steve: That would be six will be at risk. But seriously, although we don't have any details about this one, it does suggest that some new functionality added for Windows 11 also introduced a new vulnerability. If code keeps changing, it is never going to be secure. The concern about this one, since it affects Windows 11 systems that are publicly exposing their SMB service, is that it would potentially be wormable. So disabling SMBv3 compression is a workaround, but why do that? Just apply the update from this month in order to squash this bug.

And this again reinforces my admonition against ever publicly exposing any Windows service that does not need to be offering its services to anonymous public users, which as I've said before pretty much limits us to web and email. Those need to be public and anonymously available. Pretty much nothing else does. SMB, as we know, has always been a catastrophe. It's the reason I created ShieldsUP! when Windows was first put on the Internet, which was sharing everyone's "C" drive to the world. And they just broke it again with Windows 11.

Okay. Then we have three CVEs for Exchange Server because it has multiple Elevation of Privilege vulnerabilities. And it's a bit unusual for elevation of privilege (EoP) bugs to receive a critical rating, but these qualify. These three bugs could allow one authenticated attacker, an authenticated user who was in attack mode, to take over the mailboxes of all Exchange users on a server. This attacker could then read and send emails or download attachments from any mailbox on the Exchange server.

Administrators must also enable Extended Protection to fully address these three vulnerabilities. But, you know, another mess.

Microsoft has also been having constant problems with their Network File System (NFS) code, and we have another remote code execution vulnerability fixed this month, making it the fourth month in a row that NFS has received a patch to close a code execution vulnerability. This one has a CVSS of 9.8, so it could be among, I would argue, the most severe of the month's patches. Fortunately, there is not a huge population of people, Windows users, with NFS exposed. To exploit this, a remote, unauthenticated attacker would need to make a specially crafted call to a vulnerable network file system server. But this would grant the attacker elevated code execution privilege. And oddly enough, despite its CVSS of 9.8, Microsoft has this one categorized as only one of those 102 "important" in severity problems. But of course anyone using NFS should take this very seriously because your system can be scanned for and found on the 'Net.

And speaking of Outlook, Leo, you're going to love this one. This is CVE-2022-35742, which addresses that annoying Microsoft Outlook technically Denial of Service is what you would call it, vulnerability which I mentioned earlier. If an Outlook user were to receive a specially crafted email and a great many did recently their Outlook executable would immediately crash, and it could not be restarted. Upon an attempt to restart it, it would immediately terminate again once it retrieved and attempted to process that invalid message.

Leo: What a mess.

Steve: And every time you tried to start it, it would just crash. And it's not even necessary for the targeted victim to open the message or to use the reading pane. Just its presence in the inbox keeps Outlook from starting. The only way to restore Outlook's operability is to access the email account using some other email client, perhaps webmail, to remove the offending emails from the mailbox before then finally restarting Outlook. But don't worry. As Microsoft explained, Outlook is your first line of defense against executing any of those malicious Diagnostic Tool extension files, so you should feel completely safe.

And I know, yes, I'm being tough on Microsoft. But there's a large and growing consensus that they're generally losing control of this beast that they've created. I love Windows. So I sincerely hope it's just a phase they're going through as a result of what was in retrospect...

Leo: Just a phase.

Steve: Just a phase. Turned out to be a poor decision to outsource Windows quality control to their early adopter consumers. Most of those people just want to screw around with Microsoft's latest crap. They're not actually interested in controlling anyone's quality. And we've noted how, in the past at least, Microsoft often ignores those outside pleas to repair problems that have been identified in this process. They're just pesky. You know, they don't work for Microsoft. Buzz off. So this new scheme is clearly not working. And again, let's just hope it's a phase.

Last week, Paul and Mary Jo were observing that there are growing indications that consumers are not making Microsoft any money. So Microsoft appears to be more or less

abandoning the consumer for the enterprise. Now, speaking as a consumer, that would be fantastic. Please.

Leo: Leave us alone.

Steve: Just ignore us, Microsoft. Just leave us with Windows 10 forever, like you originally promised. Just patch its flaws and don't change anything else. It's fine for us now the way it is. Don't break it, please. And meanwhile add all the new features you want to the enterprise. Really, just give that your entire focus, and all of the rest of us lowly consumers will just quietly be getting stuff done. Sounds like a great plan.

Okay, Leo. I have to have a sip of water after that.

Leo: Okay, take a breath. Relax. It's going to be okay. Of course, I came in, and my Windows said, "Something went wrong. You'd better reboot."

Steve: Windows what? What was that Windows?

Leo: Oh, 11, 11.

Steve: Oh, yeah.

Leo: Yeah, 11. You know. You know that Windows. Well, you know, I have to use 11 for the same reason you have to use Windows. It's what everybody's using. All right, Steve. On we go.

Steve: So Facebook is cautiously creeping toward default end-to-end encryption.

Leo: Kind of because they have to, you know, at this point.

Steve: Yes, yes. In a move that's sure to turn up the heat on the question of whether consumers have the fundamental right to have their interpersonal communications end-to-end encrypted, making them truly and irreversibly private, Facebook's Sara Su, there's actually a position called the "Director of Trust" for Facebook's Messenger app, posted last Thursday. She said: "This week we'll begin testing default end-to-end encrypted chats between some people. If you're in the test group, some of your most frequent chats may be automatically end-to-end encrypted, which means you won't have to opt into the feature. You'll still have access to your message history, but any new messages or calls with that person will be end-to-end encrypted. You can still report messages to us if you think they violate our policies, and we'll review them and take action as necessary."

As we know, this is a big deal. The key word is "default"; right? I call it the "tyranny of the default." Even though WhatsApp, based upon the beautiful Signal protocol, offers nothing other than end-to-end encryption, Facebook's native Messenger app, which offers end-to-end encryption as an option, has never had that enabled by default. And even now, they're moving toward that cautiously.

They're saying that the move is unrelated to the recent outrage and backlash from privacy advocates after Facebook complied with an order to reveal the content of "private" messages between a mother and her daughter. The order came from a Nebraska police department as part of their investigation into an abortion-related case, of course in the wake of the U.S. Supreme Court's reversal of its previous decision, made under different Justices, in Roe v. Wade.

So, okay, fine, whatever. Despite Facebook's denial, the timing of this move is at least suggestive, but it's a welcome move nonetheless. And since most Facebook users just use Facebook's built-in native Messenger app, much as mom and daughter did in that instance, you know, if "encrypted by default" is eventually true, it will go a long way toward protecting the privacy of many more of Facebook's three billion users.

Leo: And from Facebook's point of view, they would, regardless of the politics, they would like to be able to say, well, we don't have access to that information. So, sorry.

Steve: Yes. That's what Apple has said over and over and over. And it upsets out law enforcement personnel. But, you know, really the question comes down to I think how is the greater good served? Is the greater good served by truly honoring privacy that you're presumably providing? And yes, some small minority of people will abuse that. There will be criminal use of the abuse of that privacy. Or are you actually not going to give privacy to everyone and open everyone to the potential of abuse, open non-criminals to the potential of abuse by suppressive governments and so forth. So anyway, as I've said often, I think this question of how we actually deal with this - the technology will do anything we want. The question is what do we want, and where is the greater good served?

The security firm "Cyble" (C-Y-B-L-E) recently scanned the Internet for instances of - and it's just hard to imagine this - completely open VNC servers lacking any password protection, any authentication of any kind. You connect to them, and you're looking at a desktop with mouse and keyboard access. They found more than 9,000 of those which they were able to freely log in to. And as I'll get to in a moment, these were not all safe to leave open.

Okay. So first of all, for those who don't know, VNC stands for Virtual Network Computing. It's a very old, open, platform-independent and very popular remote console access system which implements a protocol called RFB, which is Remote Frame Buffer. So one runs a VNC server on some machine, typically any random desktop operating system, and this server allows a remote networked user with a matching client to view the machine's console and use its keyboard and mouse remotely. It's essentially Windows RDP, you know, remote desktop protocol, but it's non-proprietary. It's not Microsoft's.

So this firm "Cyble" scanned and found more than 9,000 machines openly exposing their access without any form of authentication. So this begs the question, why? What could the possible reason be? How is it possible? Is it the result of negligence, or error, or maybe a misguided deliberate decision for convenience? Most of the exposed instances are located in China and Sweden. Those are the top two by far. But the United States, Spain, and Brazil were also well-represented. Cyble identified 1,555 passwordless instances in China, 1,506 in Sweden, with the U.S. coming in third at 835, followed by Spain at 555 and Brazil with 529. And as if just having consoles exposed was not bad enough, they found that some of these exposed VNC instances were making accessible industrial control systems.

They wrote: "During the course of the investigation, researchers were able to narrow down multiple Human Machine Interface (HMI) systems, Supervisory Control and Data Acquisition Systems (SCADA), Workstations, et cetera, connected via VNC and exposed publicly over the Internet."

In one of the explored cases, the exposed VNC access led to one of these Human Machine Interfaces, an HMI, for controlling pumps on a remote SCADA system in an unnamed manufacturing plant. So anybody could, like, literally go there, bring up this control panel, and start turning things on and off. And maybe it's just a spoof. I mean, it occurs to me there could be a honeypot, right, to see if someone does that.

Being curious to see how often attackers are targeting VNC servers, Cyble used its tools to monitor for attacks over port 5900, which is VNC's default server port. They found that there were over six million connection attempts in a month. Most attempts, I mean, six million attempts in a month, I don't know how many IPs they were monitoring. But wow. Most attempts to access VNC servers, they were able to determine that, originated from the Netherlands, Russia, and the United States. So nothing to be proud of here in the U.S. Unless, you know, it could be security services benignly scanning. But you don't do that six million times.

It's worth noting that this was just a scan for VNC servers requiring no authentication. Cyble's report noted that if they had raised the bar just a bit by including VNC servers using weak and easily cracked passwords, like "password" or probably "monkey," the number of exposures would have sky-rocketed. And to that end, sadly, many VNC products are old and not super-securable at best. Many, for example, offer only passwords of up to eight characters. That's no longer enough when you're able to conduct a brute force attack on something that looks juicy. Like I said, much of this stuff is old, so they are inherently insecure.

And so once again, we face the rule of the Internet that really needs to take hold: The only servers that should ever be made publicly accessible are those that have to be accessed anonymously by the public. Everything else should be behind a VPN, SSH login, or an overlay network like Tailscale. And we have to do it.

Leo: I wonder if - my favorite VNC is one of the affected ones, Chicken of the VNC. Have you heard of that one?

Steve: Chicken of the VNC.

Leo: I kid you not.

Steve: That's great.

Leo: It was a Windows VNC. I don't use VNC anymore, but...

Steve: No. I actually, when I was - a couple years ago when I was getting my world set up to do SpinRite debugging, I was looking for like the best way to do remote debugging. And so I was looking to see if there was any, like, text-based VNC that I could use.

Leo: Oh, that would be cool.

Steve: Because, yeah, and so I dusted off VNC and took a look at it and looked at various ways of using it. And I ended up not finding anything that was better than just good old Windows networking. But anyway...

Leo: Well, and RDD has had its problems, too. It's not like it's failproof, either.

Steve: No. And but in the case of VNC, one of the problems is its age. But of course the fact that you even know of Chicken of the VNC demonstrates it's been around forever.

Leo: Oh, yeah. Oh, yeah. I didn't make that up, kids. In my day, that's what we used, Chicken of the VNC, yes.

Steve: And it was really good, and its mercury content was low.

Leo: Very low, very low. It was line caught, line caught.

Steve: Okay. So for the third time in the last year, a neighbor who had heard that I knew my way around computers stopped Lorrie and me during our nightly walk to ask what she should do about her website. The common denominator in all three instances, which is why I'm mentioning this today, is that these people were not webmasters, nor were they people who were content with having Facebook host their pages. They were professionals who wanted to have their own website located at their own domain. So they got a referral from someone else, or perhaps noticed at the bottom of someone's website a mention that the site was created by "Websites R Us" or "Johnny's Websites" or whatever. One way or the other they found a service, and they used that third-party service. Typically it's a one-man shop; right? A DBA firm to function as an intermediary between the different things they needed - a domain name registrar and a hosting provider to register the domain name of their choice, create a website, and populate it with their content.

These relationships are almost always problematic because a great deal of time can be spent getting everything set up. Then of course the website's owner invariably wants to make changes, which are met with grumbling of various degrees by the webmaster who just wants his client to shut up and pay to maintain the site, as is, without actually doing any further maintenance. Since the people who own the content and are having it put there are typically unable to manage the site themselves, it's a constant problem.

So every listener here is now thinking to themselves: "Okay, Gibson, we all know this. What's the point?" The point is, what I have seen over and over is that these relationships do not age well over the long term. Or are at risk of not. And in these cases have not. The reason is the domain name. Without exception, none of these regular nice people, who have their own website under a domain name that they originally chose, having had it for years and having invested emotionally, intellectually, and spiritually, have any actual control over what they consider to be their domain name. In every case their domain name was registered by someone else for them in the beginning.

Leo: Yeah. I talk about this on the radio show a lot. Yup.

Steve: I'm not surprised, Leo. It's got to be a problem that is beginning to happen more and more because domain names are becoming, you know, they're beginning to age. So at a time when none of that accumulated value existed in "the name," this was all created. And without an understanding of the way domain name property is managed, they never stopped to wonder what would happen in the future. Whoever it was who set all this up for them had an account with a registrar, and it was under their that their domain name was, and probably still is, registered.

This all came to mind recently when I was explaining to the most recent neighbor who stopped me. And what I explained was that where their domain names were concerned, the rule is not that possession is nine-tenths of the law. With domain names, possession is 100% of the law. It's the only law.

Leo: The only thing if you had a copy, if it was a trademark, you might be able to wrest it back. You could appeal, you know, if it's your trademark and somebody else has the domain name. That's happened in the past. But it's hard.

Steve: True, although at a great deal of expense and attorneys and court cases and everything.

Leo: Right, right. Well, no, I think ICANN has an appeal process. It's not as bad as all that.

Steve: Oh, so you don't have to actually sue and get it.

Leo: You go to ICANN; right. You can say to ICANN, hey, you know, I'm McDonald's, and this guy is not. Can I have my name back? Yeah.

Steve: Right, right. So, and this sort of puts me in mind, we've talked about the problem of somebody dying with a whole bunch of passwords which are not known to anyone else. The time before this most recent time, the long-time webmaster of a different neighbor actually had died rather suddenly, being survived by his wife. Eventually, the web and email services, which continued for some time, were terminated, and my neighbor went in search of a solution. The man's wife was nice, but she knew nothing about technology, and my neighbor learned that all of his other clients were also furious and fuming that their web domains collectively had been allowed to expire, had then been snatched up, and were now hosting click-bait pages.

Leo: Oh, god.

Steve: So I had to give that neighbor the bad news that she really had no practical recourse. And in this most recent case, the webmaster is an 80-year-old geriatric good friend of the website owner. Working through him, she explained, she's created an extensive website filled with links and artwork, plays and songs. It is a content-rich labor of love, and she's been frantic that something might happen to it. So I told her that I could easily relieve some of her worry by cloning her entire website into a directory, so that at least all of its content would be archived for safekeeping. And so the next morning I did that, and she was hugely relieved.

But I explained the situation to her about the control of domain names; and that if anything should ever happen to this person, in the absence of some sort of plan, she stood to lose any and all use of that domain because of the way it was registered. She had no claim to it whatsoever. It actually is not hers, it's his. So anyway, so I did a little bit of WHOIS poking around, and I discovered that he's using a service called Enom (E-N-O-M) which is for domain name resellers. Enom, it turns out, is a branch of Hover, my own chosen registrar with whom I am very happy. So I told her that the best thing she could do would be to create her own account at Hover, then arrange with her friend to transfer the domain, with all of its existing settings intact, from his Hover account to hers.

She said it would be a bit dicey because he was a friend, and she didn't wish to offend him. So I told her, okay, in the first place he will at least understand that it is an entirely valid concern for anyone to have. So if transferring the domain to her now is not an option, could he at least explain what he has in place for dealing with his own potential inability, for whatever reason, to continue?

And so as a consequence of all this, I just wanted to take a moment to mention this to everyone who's listening because even though we all probably manage our own domain properties, it must be that we're all aware of others who are in a similar place, and it might be worth asking them about the plans their domain name providers have for their domain names upon any event that might cause their registrations to expire and be lost.

Leo: And of course the strong advice is, if you're going to have a domain name, register it yourself. Don't let the third party register it for you because you're not controlling it. You don't control it.

Steve: Yes. And I was impressed. I looked at the registration, and he had renewed it two days after one month before expiration. So that is to say, it was probably set up for auto-renew. It was renewing annually. So it looked like he was tending it well. And of course the flipside is, if you are registering your own domain, then you do have that responsibility. You need to absolutely make sure that it is, you know, that it's set to auto-renew. That you're maintaining a current email address for them so they're able to notify you of any problems, that they've got a credit card that they're able to charge. I mean, so the point is, if you're going to take on that responsibility, you know, there is some responsibility that comes with it. But, boy, I'm glad to hear that you're seeing the same thing among your callers on the weekend.

Leo: Oh, yes. I see it all the time. And she should approach, I mean, if this guy's a friend, this 80-year-old guy is a friend, just say can I transfer it over to me because it's mine?

Steve: Yup. Right.

Leo: You know, he did it probably as a favor to her because he knew how. Because the reason people don't want to do this is they don't know how to set up the DNS to point to their website. And I understand. This is a little techie. So I understand why they don't do it. But, yeah, you need control. You need to control it. Especially if it's a business.

Steve: And you know, it made me think, Leo, that shouldn't there maybe be a provision added to domain name registrations where they actually have something like a next of kin.

Leo: Right.

Steve: It's like part of the registration. So that, for example, in this instance he could put her...

Leo: Designate her.

Steve: Predesignate her as this domain's next of kin.

Leo: Or say I'm registering this as a third party, and the domain really belongs to this. There should be a way to do that. One of our chatters, PCGuy8088, who does this for a living, he says, "Whenever I create a domain for an account, I create a separate account for the domain, and their information is registered in the person's name, using their credit card number." That's a responsible webmaster. That's the right way to do it.

Steve: Yeah. And I have not yet run across anyone who does. So anyway, I just - because this actually happened, I thought, you know, let me just take a minute to just remind our listeners that I'm sure they don't have some third party managing theirs. If they do, you know, get it. But boy, I'll bet they know people who do. And these problems keep cropping up when something happens. And mostly I think what people are surprised by, both of the people who approached me, my neighbors, were shocked that there really was no recourse. I mean, no practical recourse for some non-trademarked random domain name. They don't have any rights. It is the rights of the accountholder.

So, okay. And speaking of making a web copy, I fired up my favorite cloning tool. It came up, and it said, hey, I've got an update for you. I thought, oh, that's nice. I went there, and I was reminded that it was free and donation-ware. And I gave them 10 bucks because there's a long line of maintenance that's been done over time. They are keeping it current. They are giving it new features. Anyway, this thing, I just wanted to give them a shout out. It's Cyotek WebCopy. Cyotek WebCopy. And I'm using it because it is just so good. You point it to the root of a website. You point it to a directory that either exists or doesn't, and you say, make me a copy. And this thing just, it does all of the link following, and it changes all the links so that you end up with a locally browsable copy of the site.

And these days, with drives being as large as they are, and there's been discussions of this in GRC's newsgroups about people just cloning sites because they don't want to ever lose some of the content of a site that they really care about. And yeah, the web archive kind of has it, but it sometimes doesn't have all of the assets that you wish it did. So anyway, Cyotek WebCopy is the thing that I've been using, and really happy with it.

There was a quote that I saw that I just had to share with everybody. I ran across it because Ryan Sleevi, who's one of the security guys at Google, he retweeted a tweet from a Jens Axboe. Jens Axboe was quoting @cra - so, boy, a three-letter Twitter handle, that's rare - whose self-description on Twitter says "Building a better world through open collaboration." So when Jens quoted @cra, he added to his retweet "I don't think I've

ever seen anything more true posted." So here's the original quote of the open collaboration guy via Jens Axboe's retweet and Ryan Sleevi's retweet retweet. They said: "Running a successful open source project is just 'Good Will Hunting' in reverse, where you start out as a respected genius and end up being a janitor who gets into fights."

Leo: I love it.

Steve: And think about it. It is exactly right. Running a successful open source project is just "Good Will Hunting" in reverse. You start out as a respected genius, and you end up being a janitor who gets into fights. Because that's what these things, you know, it is a thankless task. And you end up, exactly as these guys said, just having to defend every decision you ever made, and arguing back and forth about features or not features and so forth. And anyway, just a great quote. So thank you, Ryan, for bringing it to my attention via Jens. Oh, and Leo?

Leo: Yes?

Steve: Progress from Ed Cano, the father of the Sandsara, which is that thing you and I both invested in years ago.

Leo: Oh, wait a minute. This was like the Zen garden thing?

Steve: Yes. It's the Zen garden thing. It's a beautiful wooden round table filled, sort of a, well, Zen garden is a great description, round, filled with very fine sand, and a ball bearing is rolling around in the sand leaving its wake behind it, all controlled by a very clever two-stepper motor mechanism that allows this ball to be moved in really cool geometric patterns.

Ed just sent: "Hi everyone. Recently I posted a comment stating that the Sandsara App was available in the App Store. We had to make further improvements for the Android users, so it took us more time. But now it is also available in the Google Play Store. You can follow the links below to download the app." And for anyone who's curious, and Leo you might be, I've got both of the links to the Sandsara apps in the show notes.

He said: "The app will help you to personalize your Sandsara by selecting the patterns that you like the most, adding more designs, or changing the colors of the lights. Even when we are happy with our current development, we are working on more features for the app, so please send us any feedback so we can improve our current version." And here's the news: "Regarding shipping and manufacturing, the container with 300 Sandsara has been cleared!"

Leo: Oh, so it's stuck in shipping and customs, yeah.

Steve: Yes. It turns out for some reason getting black sand through customs is a huge problem.

Leo: Oh, wow. Interesting.

Steve: He said: "We're only a few days away from the first orders" - meaning you and me, Leo, and I know many of our listeners got onboard also because Ed and I had a conversation afterward. He listens to Security Now!. And so he was first of all surprised when I had discovered it and heard me talking about it, and then they got a bunch of backers as a consequence of the podcast. Anyway, he said, "...the first orders to receive their units. For those backers, you will receive your tracking numbers by the end of the week. And it will take another week to receive your Sandsara by your door. After clearance, the container is on its way to the U.S. warehouse, where our partner will split everything up in orders and start the individual fulfillment."

And he said: "For the rest of the orders, we mentioned that all orders would be completed by late August, and we are on track to accomplish so." He said: "We expect to complete all processes in around one week. Depending on the courier, your package might take two to three weeks once we start shipping. Please be wary of any messages this week because you might receive your tracking number." And blah blah blah. So anyway, his communication has been great through these years. And as I mentioned last time I talked about him, he is clearly a perfectionist. It has been - obviously it took far longer than he expected. All kinds of problems came up. But he never quit. And I, based on everything I've seen, we are going to be really happy with this thing when it finally arrives. And it looks like it may.

Leo: I don't know if I ordered it.

Steve: I think maybe you still can.

Leo: I hope I didn't because I don't know where the hell I'm going to put it. We have cats.

Steve: You have cats, yes.

Leo: Cats might see this as an alternative bathroom.

Steve: Yeah. Well, now, you are able to get a glass lid for it. So it can be closed, although it could still be toppled over.

Leo: Yeah.

Steve: So, yeah, cats might be a problem.

Leo: This might be for - maybe for work I'll have it, yeah.

Steve: Anyway, a couple bits - oh, that'd be good, yeah.

Leo: Yeah.

Steve: A couple bits, oh, that'd be good, yeah. Couple bits of closing-the-loop feedback. Alex T said: "Hi Steve. Thanks for another great show. I think the way airplanes restrict images" - he's referring to the comment last week about how could the free WiFi allow text messaging but not images. And he said: "I think the way airplanes restrict images in messaging clients is even simpler than that." That is to say, simpler than a bandwidth cap that's very low. He said: "I believe WhatsApp and others send image blobs to a separate endpoint for storage, and insert a reference to that blob into the message. A firewall block to the media storage endpoint would do the trick." And of course given that that's true, Alex is completely correct.

Thomas Apalenek, I hope that's how I pronounce his name, he said - oh. He said: "Copy-As-Path in Windows 11." And Leo, you're able to check this out. He said: "Hi Steve. I use Shift-Right Click and then Copy-As-Path all the time since you showed us that a few years ago. So thank you for that. I don't know how much you've played with Windows 11." Safe to say, as little as possible. He said: "Copy-As-Path has finally been moved to the standard right-click context menu."

Leo: Oh, good. Yeah, in fact I see it. Let me see. Let's see.

Steve: So no shift required anymore.

Leo: Yeah, yeah. Or hit CTRL+SHIFT+C without even a context menu, and it'll get it. CTRL+SHIFT+C is the Copy-As-Path. Which is great for pasting into terminal or whatever, code.

Steve: That is nice, yes, yes.

Leo: Yeah. Thank you.

Steve: Also Brad Cochran, he said: "Hey Steve, long-time listener, first-time caller." He said: "I even dusted off my Twitter account to message you."

Leo: Aww.

Steve: He said: "Regarding your most recent Security Now! episode, it sounds like you haven't come across Attack Surface Management." Apparently that's an acronym, ASM. He said: "I wanted to share a bit of info about how it works." Now, he's referring to my talking last week about Microsoft's new, apparently it was a scanning service which they're making available to enterprise in order to check their attack surface. He said: "I wanted to share a bit of info about how it works. Full disclosure, I work for Palo Alto Networks and sell a competing solution called Cortex Xpanse, but because of that I can explain at a high level how ASM works.

"You're right in your assumption that this is an external scan. Essentially, ASM tools" - meaning Attack Surface Management tools - "scan the global public Internet on a daily basis, which is possible now due to advances in computing resources" - meaning lots of cloud and bandwidth - "for open ports and protocols. You mentioned Shodan in the show, and that's similar. One important note here, due to the Computer Fraud and Abuse Act (CFAA) and similar laws in other countries, these do not tend to be invasive port scans

such as Nmap," he says, "which would violate the law if you ran on a network you don't control without approval."

And I'll mention that my own ShieldsUP! port scanner goes to some length to detect an open port without ever actually completing a connection. Since ShieldsUP! emits packets under my control, as opposed to just using a full TCP/IP stack, I send out a SYN packet. If a remote system replies to my probing SYN with its answering SYN/ACK, I never follow through with the final ACK to actually fully open the connection. Instead, I send an RST packet to force that half-open connection to be aborted and immediately closed.

Anyway, he continues: "Microsoft's solution comes via their acquisition of RiskIQ." Which we talked about when that happened. He said: "Presumably they're augmenting the external attack surface data collected with that tool with their own intelligence, and even tying that with data from other tools in the Defender suite." And he says, "We do something similar here." He says: "Other common players in the ASM space include Shodan, BitSight, SecurityScorecard, Randori, and CyCognito." He says: "And more popping up all the time. This is definitely a growing cybersecurity market." He says: "I hope that helps." And indeed, Brad, thanks very much for the follow-up.

And lastly, an interesting point was made by a Tom Malaher. He said: "Steve, in Security Now! 883," which was last week, "you said 'Any new hash will need to start over from scratch earning the reputation that that exact code file is trustworthy.'" He said: "How does this interact with creating unique downloads of SpinRite EXE for each paying customer? Aren't those in conflict?"

And I replied to Tom. I said: "You are 100% correct, and it's something I've considered at some length. As it has always been, SpinRite's EXEs will continue to be uniquely created for each of its users. And I now have an HSM a Hardware Security Module from DigiCert installed on GRC's main server which will allow it to perform automated on-the-fly EV code signing."

Leo: Oh, how cool is that.

Steve: Yeah.

Leo: That's neat.

Steve: "Thus providing the highest integrity assurance possible. So each custom-created SpinRite will be individually and validly signed, which is the best I can do. But there will be no way for each of those signed hashes to ever earn a reputation for themselves..."

Leo: Because they're supposed to be unique.

Steve: "...since each one will be unique. Therefore, it may become common and expected for any reputation-based anti-malware service to be warning its SpinRite user that their freshly downloaded copy of SpinRite has never been seen before." As indeed it wouldn't have been.

Leo: Yeah.

Steve: But at least they will be EV code signed.

Leo: That's good.

Steve: So the highest level of integrity possible.

Leo: Yeah. That's really good. All right. Let's talk about TLS Private Key Leakage.

Steve: Ooh.

Leo: Something you never want to have happen.

Steve: You don't want your key leaking.

Leo: You don't want any leakage on that one. All right, Steve. Let's talk about TLS.

Steve: Okay. So as I said at the top of the show, four researchers, two each from the University of California at San Diego and the University of Colorado in Boulder, performed an amazing piece of work, described in their paper, which they titled: "Open to a fault: On the passive compromise of TLS keys via transient errors." Their work was just presented during the 31st USENIX Symposium held in Boston last week.

Okay. So to get everyone's attention, I'm going to first share their paper's abstract. They said: "It is well-known in the cryptographic literature that the most common digital signature schemes used in practice can fail catastrophically in the presence of faults during computation. We use passive and active network measurements to analyze organically occurring faults in billions of digital signatures generated by tens of millions of hosts.

"We find that a persistent rate of apparent hardware faults in unprotected implementations has resulted in compromised certificate RSA private keys for years. The faulty signatures we observed allowed us to compute private RSA keys associated with a top 10 Alexa site, several browser-trusted wildcard certificates for organizations that used a popular VPN product, and a small sporadic population of other web sites and network devices. These measurements illustrate the fragility of RSA PKCS #1 v1.5 signature padding" - which is the standards used - "and provide insight on the risks faced by unprotected implementations on hardware at Internet scale."

And I'll break all that down. But what they found - okay. Well, I'll break down what they did and what they found. But one of the things I loved about the way they introduced the nature of the way digital systems can be fragile was to remind us of a true-life example that we may all recall. They wrote: "During 2009 to 2011, Toyota issued multiple vehicle recalls after hundreds of crashes had been reported relating to unintended acceleration." Remember those?

"Initially, Toyota placed the blame on driver error, shifting floor mats, and sticky accelerator pedals. In 2013 expert witness Michael Barr testified in the Bookout v. Toyota Motor Corporation case that a single bit flip sufficed to kill a throttle monitoring task, resulting in uncontrolled acceleration. Toyota lost the case and began settling with crash

victims out of court. The exact cause of the memory corruption in Toyota vehicles was never established. It could have been a buffer overflow, cosmic rays, or hardware faults. No matter the underlying cause, the existing hardware protections were insufficient, and the software was brittle in the face of hardware errors."

And it's a little chilling how perfect this analogy is. Now, we don't need to look further back than the week before last when this podcast was titled "Rowhammer's Nine Lives." With DRAM, we have an unfortunate and persistent flaw where it's entirely possible for a bit of DRAM to spontaneously flip. It was during the 25th USENIX Security Symposium in 2016 that the paper "Flip Feng Shui" was delivered, and in their summary they noted that Flip Feng Shui relies on hardware bugs to induce bit flips in memory. They weaponized that unfortunate characteristic of weak memory operation. And we know that this can happen without a memory system being under active attack.

The reason DRAM can be equipped, at extra expense, with parity checking and ECC is because DRAM memory is not perfect. Parity checking cannot correct, but it will at least catch a single bit flip. And ECC can correct such an occurrence. And all Internet packets contain checksums to detect any simple errors introduced between the time a packet is created and the time it is received at the other end. These measures and many others are in place because, in reality, computers can and do make mistakes. While we would like to believe that every time we multiply the same two large prime numbers we're going to get the same result, in reality that's true almost all of the time, but it's that "almost" that can be weaponized.

So with that bit of background, here's how these guys describe their work, what they did, and what they found. Referring back to their Toyota example they begin: "Cryptographic software engineering is, fortunately, less often considered to be a matter of life or death." You know, than crashing a Toyota. They said: "Nonetheless, faults can have a similarly catastrophic impact on cryptographic systems. As prior work has shown, attacker-induced or naturally occurring bit flips can corrupt cryptographic computations, causing them to produce incorrect results, or even leak secret information or keys.

"In this paper, we show that these attacks can be applied entirely passively, allowing a network adversary to derive TLS RSA private keys simply by observing network traffic. When errors occur during a server's RSA signature computation, the resulting failed handshake can give an attacker sufficient information to derive the server's long-term private key."

Now, okay. To clarify before I continue, when a server makes a mistake during its computation of a TLS handshake's RSA signature, that handshake will fail. But due to some known vulnerabilities in the specific cryptographic operations, actually something known as the Chinese Remainder Theorem, the way the signature is...

Leo: Ooh, ooh. I learned about that two years ago in the Advent of Code, the Chinese Remainder Theorem.

Steve: Oh, very cool.

Leo: I never knew it even existed. Geez.

Steve: Yup. So the way the signature is wrong leaks a bit of information about the private side of that RSA computation. So they continue: "We demonstrate these attacks by collecting 5.8 billion TLS handshakes from two different university networks. These

handshakes included 3.3 billion connections using TLS 1.2 or earlier and 2.7 billion server signatures. Over a few months, we found nearly 2,000 non-validating digital signatures from failed handshakes."

Okay, now, again, to be clear, that number should be zero. But some fault in the servers were resulting in a very low but non-zero number of failed signature computations, nearly 2,000 out of 2.7 billion. But that's all it took.

They said: "Some of these failed handshakes allowed us to compute three RSA private keys associated with Baidu, a multinational technology company in top 10 Alexa. These three keys were used to secure more than a million connections to hundreds of hosts in our dataset corresponding to dozens of Baidu's cloud-based services. This passive attack is particularly concerning in the context of nation-state adversaries conducting mass surveillance.

"Unlike active attacks or remote compromise, which risk leaving evidence of tampering, passive fault analysis leaves no trace on either the client or the server. A network adversary only needs to observe network traffic passively" - which again reminds us of the NSA that was tapping major exchange points, we didn't know why. "A network adversary only needs to observe network traffic passively and perform simple cryptographic calculations, capabilities that modern nation states are known to possess and employ for the purpose of network surveillance. This attack is exacerbated when TLS servers and clients negotiate non-forward secure ciphers, allowing the network attacker to passively decrypt encrypted TLS payloads using the server's private key, without leaving any trace of compromise."

Okay, again, interrupt. We've talked about this before. Recall that I observed that expired and no longer valid web server certificates should be securely destroyed and not allowed to escape, even though they were no longer valid. The reason for that is when protocols without forward secrecy are used, the private key, if it's later disclosed, can be used to fully decrypt any conversations that may have been archived while that key was in use for possible future use. So these guys are observing that if all of a server's traffic is stored, and if that server later makes some mistakes during its RSA signature computation, then all of the stored communications can be retroactively decrypted.

They continue: "In addition to demonstrating passive fault attacks, we also carried out active scans of TLS hosts, and performed a retrospective analysis of historical TLS scan data between 2015 and 2022 that included tens of thousands of non-validating signatures." In other words, this is actually a problem. Tens of thousands of non-validating signatures. "In total," they said, "we computed 127 private RSA keys from these active scans." And so by "active scan" they mean that rather than passively collecting server handshakes being initiated by random clients, they became a TLS client and actively initiated a great many of TLS connections to specific servers. And sure enough, by doing lots of their own TLS connections, they were able to essentially force the remote server to eventually make a mistake, and they were able to then capture its security certificate.

They said: "We compare our results to active scans from a 2015 technical report by Florian Weimer of Red Hat. He appears to have been the first to observe that active scans could be used to detect or trigger these types of RSA signature faults at scale." So this was known in 2015. It's still happening today. "He found that several open-source TLS libraries did not implement countermeasures against signature faults, and performed active TLS scans over a period of months that resulted in a few hundred invalid signatures that successfully compromised private keys, mostly from devices from several vendors."

They said: "Our passive analysis and recent active scans show that these problems are still present in current implementations. We were able to compute the browser-trusted private keys for a handful of user-facing websites from sporadic faults, as well as observing dozens of certificate private keys compromised by devices. These certificates span from untrusted device default certificates to CA-signed browser-trusted wildcard certificates for entire organizations. Although all of the open source libraries we inspected have implemented countermeasures, it appears some proprietary TLS implementations are still vulnerable to this attack."

And I'm not going to go deep into the math because it's not really relevant to understanding the impact of this attack. But there is some historical background that I think everyone will find interesting. They said: "The flaw we exploit is well known in the cryptographic side-channel literature, first described in a 1997 paper about an RSA key recovery attack. Almost all RSA implementations use the Chinese Remainder Theorem optimization for modular exponentiation in RSA signing. But errors that occur in one of the half-exponentiations in this algorithm can result in leaking information that can be used to derive the RSA private key. A researcher named Arjen K. Lenstra had published a technical report the year before in 1996 titled just 'Memo on RSA signature generation in the presence of faults.'

"Lenstra improved the attack to require only one signature when the message is known. This attack works against any deterministic RSA signature scheme using the Chinese Remainder Theorem optimization. The countermeasure to these attacks is to validate the RSA signature before sending it." And I should mention that I saw another reference in this that Peter Gutmann indicated that that incurred about a 10% overhead in performance. So you've got to do it, but it does cost you a bit.

"Prior to Florian Weimer's work in 2015," they said, "almost no implementations validated RSA signatures before sending." So it was just in the technical crypto literature until Florian noted this working for Red Hat in 2015. Before then, nobody was doing RSA signature validation. And even after that, even though, well, they said: "Following the report, all of the software libraries Weimer contacted implemented countermeasures" - you bet - "and Cavium issued a patch for their cryptographic accelerators, which appeared to be at fault for several of the vulnerable devices he discovered."

They said: "In this work, we find that spontaneous faults compromising RSA keys through PKCS #1 v1.5 signatures continue to be present at a low but persistent rate in both passive and active network measurements over time, despite the attention drawn to this vulnerability in 2015." In other words, as should hardly come as a surprise to anyone, not everyone got the memo and fixed their code. Or since then perhaps new code was created, and the wisdom of the past was lost.

They said: "In the present era in 2022, we find that this flaw is not just present in the types of network devices that have already been observed to suffer from cryptographic implementation flaws in previous studies, but that it also affects user-facing websites and infrastructure that receive significant amounts of network traffic."

They finish: "These vulnerabilities are due to a hazardous combination of cryptographic libraries vulnerable in the face of computational errors, and the brittle nature of the RSA PKCS #1 v1.5 signature padding scheme as used in TLS 1.0 through 1.2. PKCS #1 v1.5 signature padding makes key compromise trivial in the presence of Chinese Remainder Theorem faults.

"Prior to TLS 1.3, handshakes take place in plaintext, providing all the information a passive network adversary needs to validate signatures on observed connections, or derive keys when errors occur. But on a more positive note, TLS 1.3 provides multiple countermeasures against these issues, including moving the key exchange earlier in the

handshake in order to ensure that certificates and signatures are sent encrypted, and using RSA-PSS signature padding, which prevents this type of key compromise if implemented correctly.

So a long ago discovered and known, yet still present and haunting the Internet, significant weakness in the current implementation of some TLS handshake stacks exists. The eventual migration to TLS v1.3 will finally fix this, but TLS 1.2 support will not be disappearing for a long time. So an active adversary could insist upon negotiating under TLS 1.2 while its support continues. Which as I said won't be going away soon because there's going to be lots of clients that don't yet do 1.3. So a bad guy negotiates under 1.2 and hopes to get the remote server to stumble over its RSA signature computation. If that were to happen, its private key could be disclosed. And at that point, well, its security is completely compromised, just by listening or in some cases actively attacking. Wow.

Leo: It's so fantastic that I can finally use my Chinese Remainder Theorem.

Steve: Yes, for more than reheating your kung pao chicken.

Leo: Yeah, getting Santa Claus to the elves in time. That's great. That's great. All right. I don't think I understand. But I will listen again, and maybe I'll understand it on the second time through. Steve is a genius. That is clear. And by listening to him we all benefit. I hope you enjoy the show. Hope you come by again next Tuesday when we do Security Now!. Let me give you some of the information you might need to stay in touch.

Of course Steve has a copy of the show at his website, GRC.com, the Gibson Research Corporation. He actually has two unique versions of the show, a 16Kb version in addition to the normal 64Kb version. It sounds a little scratchy, but it is a lot smaller. So if you have limited bandwidth, that's a great place for you. I think Steve started doing this years ago because we had listeners in Australia who had extreme bandwidth limits, and they wanted to listen to the show, and this was the only way they could do it. I'd hate to get rid of it. Somebody probably is using it. Do you see any downloads on it?

Steve: Well, as far as I know Elaine is still using it.

Leo: Oh, that's who you made it for.

Steve: She had a satellite connection, and the files were uncomfortable large for her.

Leo: Perfect. So for anybody with bandwidth caps. And Elaine does take the 16Kb version, listens carefully to the scratch, sounds like Thomas Edison's first dictaphone recording. But anyway, she then makes a very good transcript of every show so you can search the transcripts or read along as you listen. Both of those are at GRC.com, along with a 64Kb version.

When you're there, do check out SpinRite. As you hear, it's in active development right now, current version of SpinRite, the world's best mass storage maintenance

and recovery utility, is 6.0. But 6.1's on the way. You buy 6.0 now, you'll get 6.1 free when it's available. You can also participate in the development. That's GRC.com. And then when you're there, gosh, there's so much other wonderful stuff worth checking out. Steve has a wide - what did they say about Isaac Newton? A mind ranging freely. He is all over the place would be another way to describe it.

Steve: More contemporary way to phrase it, yes.

Leo: Yeah. There was a great quote about Newton, and I can't remember it exactly. It was originally the Apple logo. That's the only reason I know it.

Steve: Oh, cool.

Leo: Anyway, yeah. I think it was - "a mind forever voyaging." Thank you. I think that's a good name for you, actually, "A mind forever voyaging." Right?

Steve: I'm bailing water, Leo.

Leo: "A mind forever voyaging through strange seas of thought." That's what William Wordsworth said about Sir Isaac Newton. Alone. I left off the most important part. I think this is not for you. You can put it on your tombstone maybe. "A mind forever voyaging through strange seas of thought alone." No, that's not Steve.

Steve: No. Community is a good thing.

Leo: Yeah, exactly.

Steve: Community's a good thing.

Leo: We have the show as well on our website. We have audio and video at TWiT.tv/sn. You can download it there. If you want to watch us do it live, like you want the very freshest copy of Security Now!, there is a live stream every Tuesday about 1:30, between 1:30 and 2:00 p.m. Pacific, 4:30 and 5:00 p.m. Eastern, 20:30 UTC. That's at live.twit.tv. You can chat there live with us at irc.twit.tv or in the Club TWiT Discord.

We also have on-demand versions, as I mentioned, on the website. And there's a YouTube channel. Of course the easiest thing to do, whether you use Steve's feed or ours doesn't really matter, is subscribe in your favorite podcast player. And that way you get it automatically. You don't have to think about it. You just know you've got Security Now!. Start your collection now. You only have 883 other episodes to get.