Article 45

Description: Where was Microsoft storing their Azure keys? What four new zero-day flaws has Microsoft declined to repair? And what happens next? What's this week's latest mass casualty event for publicly-exposed Internet servers? And do we have any news on last week's Citrix Bleed fiasco? What comes after CVSS v3.1 and why? What happened to Google's Web DRM proposal? And what about the earlier Cisco IOS XE mass casualty mess? And what's the new Security Now! podcast slogan to emerge from it? Our favorite password manager just announced their support for Passkeys. Now what? That guy with the badly messed-up SSD shared the results of using SpinRite 6.1. I'll share and explain what happened. And then, after entertaining some great feedback from our listeners, we're going to look into the next big looming battle between conservative tech and rapacious governments. All that and more during this week's Security Now! podcast #947 - and counting.

High quality (64 kbps) mp3 audio file URL: http://media.GRC.com/sn/SN-947.mp3
Quarter size (16 kbps) mp3 audio file URL: http://media.GRC.com/sn/sn-947-lq.mp3

SHOW TEASE: It's time for Security Now!. Steve Gibson is here. Coming up, lots to talk about. Microsoft has some more flaws in Exchange Server. This time they say, yeah, we're not going to fix it. Well, maybe they ought to. We'll also talk about an attack on our favorite hardware store, oh, no. An update on Citrix Bleed. And then Steve's going to talk about something I hadn't heard anything about, but it's a real grab from the EU that will really destroy Internet security. What is Section 45? Stay tuned. Security Now! is next.

Leo Laporte: This is Security Now! with Steve Gibson, Episode 947, recorded Tuesday, November 7th, 2023: Article 45.

It's time for Security Now!, the show where we cover the latest news in the world of security, safety, and good vibes online with this guy right here, Steve Gibson, of the government - what did we decide GRC stands for? Government Regulations?

Steve Gibson: Actually, one of our listeners knew. It's French for the - it's the French version of the RCMP, Canada's Royal Canadian Mounted Police.

Leo: Okay.

Steve: And we'll be getting to that later in the podcast.

Leo: Okay. Okay.

Steve: But first.

Leo: Yes.

Steve: But first. Today's podcast has the mysterious title "Article 45." What is Article 45, and why do we care? Well, we're going to explain that, and it's like a big deal. But first we learn where Microsoft was storing their Azure keys?

Leo: Oh, no. Oh, no.

Steve: Also, oh, it's as bad as we thought. What four new zero-day flaws has Microsoft declined to repair, and what's probably going to happen next? What's this week's latest mass casualty event for publicly exposed Internet servers? And do we have any news on last week's Citrix Bleed fiasco? What comes after CVSS v3.1 and why? What happened to Google's Web DRM proposal? And what about the earlier Cisco IOS XE mass casualty mess? Oh, and what's the latest Security Now! podcast slogan to emerge from that event? We've got a new slogan for Security Now!.

Our favorite password manager just announced their support for Passkeys. Now what? That guy from last week, or no, actually it was several weeks ago, with the badly messed-up SSD, shared the results of using SpinRite 6.1 on it. So I'll share what he reported and explain that. And then, of course, after entertaining some great feedback from our listeners, we're going to look into the next big looming battle between conservative tech and rapacious governments. All that and more...

Leo: I can't wait.

Steve: ...during this week's Security Now! podcast 947 titled "Article 45."

Leo: All right. You're not getting political on us, are you?

Steve: No, no, never that. But we do have a wonderful Picture of the Week.

Leo: We do.

Steve: Which begs the question, to park or not to park?

Leo: This is a great picture. Look. I can see somebody's thumb in it, so it's obviously from a listener; yes?

Steve: Yup, yup. They saw this and said, okay, we've got to send this to Steve. So for those who are not looking at video or don't have the show notes yet, our Picture of the

Week is a municipal sign that is trying to control the parking in this area. And it's two signs stacked on top of each other. The first one is, you know, the one everyone has always seen; right? No Parking Any Time, it says, and then there's a red arrow pointing to the left, meaning, you know, obviously leftward of this sign. No Parking Any Time. And then the second sign, the one below it, this one in green because it's got better news, although it ends up being a little confusing, this sign says 60 Min Parking - you know, minutes parking - All Other Times.

Leo: What? That means what?

Steve: Wait, wait, yeah.

Leo: It's not completely inconsistent. I mean, basically no other time. But I don't know if that's what they intended; right?

Steve: That's true. That is true. So, you know, if you were to park for less than an hour, probably at any time...

Leo: You'd be still...

Steve: ...since the first sign says No Parking Any Time...

Leo: Yeah, yeah.

Steve: ...you'd have a hard time arguing with a judge, hey, I was following the second sign, which said 60 Minute Parking All Other Times. The judge would say, yes, but it said No Parking Any Time. Anyway...

Leo: This is basically in the benefit of the meter maid. They get to choose.

Steve: Well, and you know, Leo, I was thinking about this. This sort of explains how corporations screw up; right?

Leo: Yes, yes.

Steve: Because here we are in a government, which is, you know, an organization. Some poor guy was told, you know, put up this 60 Minute Parking sign underneath the No Parking Any Time sign.

Leo: Right, right.

Steve: And he's looking at it, the guy actually on the ground, as we say now, you know, feet on the ground, he's there thinking, okay, this is the most screwed up thing I've ever

seen. But on the other hand, he works for the municipality, so it's probably not the most screwed up thing he's ever seen. It's probably typical.

Leo: Okay. I'm just going to put it up and not ask any questions.

Steve: I did my job here.

Leo: I did my job.

Steve: Anyway, yes, we are enjoying these pictures. Thank you, listeners. Okay. So last Thursday Microsoft posted - I love this, too - under the headline "Announcing Microsoft Secure Future Initiative" - you know, there was some drum roll and some horns blowing in the background - "to advance security engineering." So, okay. They're announcing their secure future initiative to advance the state of the art in security engineering. And this was written by Charlie Bell, the Executive VP of Microsoft Security. So he opened this posting with this introduction.

He said: "Today Microsoft's Vice Chair and President Brad Smith shared insight" - because that's what you want, you know, you want from your VP and your president, you want some insight - "on the global cybersecurity landscape and introduced our Secure Future Initiative. These engineering advances anticipate" - they're anticipating, Leo, they're not reactive, they're going to get there ahead of time - "future cyberthreats, such as increasing digital attacks on identity systems. They also address how we will continue to secure security foundations necessary for the AI era and beyond. In the spirit of transparency and to emphasize the importance of this moment, we are sharing the internal email sent earlier about our Secure Future Initiative's strategy and objectives."

So, wow. We're getting a peek inside, under the covers, if you would. So I have the link to the entire piece in the show notes for anyone who is interested. Mostly I have to say it's a marketing piece, you know, blah blah blah, we're leading the way toward a more secure future, improving the lives of our customers in the face of rapidly growing cyberthreats, blah blah blah.

And I normally wouldn't have given this a second thought, nor even be mentioning it here, except about 20 - I did read it. That's how I know I can accurately summarize it with, you know, as I just did. But about 80% of the way down, something appeared that did seem worth sharing. They wrote, as part of their "Secure Future Initiative," where they're going to get ahead of the bad guys, and they're anticipating future cyberthreats, they said: "To stay ahead of bad actors, we are moving identity signing keys to an integrated, hardened Azure HSM" - you know, a Hardware Security Module - "and confidential computing infrastructure. In this architecture, signing keys are not only encrypted at rest and in transit, but also during computational processes as well." What a concept. "Key rotation will also be automated, allowing high-frequency key replacement with no potential for human access whatsoever."

Okay. So in short, Microsoft will be leading the way into a Secure Future - that's capital S, capital F - by working to catch up with what everyone else who cares about security has been doing all along for many years already. As I observed a couple of months ago, the only possible way they could have lost control of that private signing key during a system crash and the subsequent RAM snapshot that it took, was if that private key was in RAM at the time of the crash. And the only way it could ever have been in RAM was if signing was being done outside of an HSM. Now we learn indeed it was because, wow,

they're going to move to the future, Leo. They're going to lead the way by, you know, I mean, I have them on my various computers. But okay.

During last week's "Citrix Bleed" podcast, we examined, as we know, a crisp and clear example of a bug which allowed for the exfiltration of RAM. And as it happened, that RAM contained active and valid authentication tokens. We don't know whether they needed to be lying around in RAM. It's quite possible that they did need to be there in order to remain valid. But it's very common to make the mistake of leaving sensitive information lying around even after it's no longer serving a purpose. The problem is that our current programming languages are still not secure by default or design. So they must be made - so making our systems secure while using these insecure by default languages is a deliberate act. It must be a deliberate act.

And I also noted last week that while I was coding the SQRL client I was in a more or less constant state of terror that I was going to make a mistake. And I would submit that that's the state you want your coders of secure systems to be in. You know, they should not just be distracted and worrying about when is lunchtime. They should be terrified about the code that they're writing.

So a great and useful concept and phrase is the notion of multi-layer security. The idea is that there is no single point of failure that would result in a security compromise. In order for security to be compromised, many things would need to go wrong all at once. In the case of the Citrix Bleed that we talked about last week, if the system's RAM could have been swept clean of valid tokens - and we don't know whether or not that would have been feasible. But if it could have been, then even in the face of that very clear and clean coding error, valid tokens would not have been available for exfiltration. In other words, wipe RAM, not because you know you need to, but because doing so would add an additional layer of security. And additional layers of security, as long as they don't get in the way, are never a bad thing for a secure system to have.

So back to Microsoft's case. In that almost hard to believe case of Microsoft's loss of their private signing key, they explained that they did indeed already have multiple layers of security in place. I think it was, like, five of them. Yet in a bizarre and quite unlikely seeming chain of failures, where it was necessary for every one of these layers to fail, every single one of them was actually needed. Yet they all collapsed at once. If they had had one additional layer of using a Hardware Security Module in their system, which now they're boasting about doing, then none of those high-value government email accounts would have been breached as a result of the failure of every other layer of security.

So our takeaway here is, it is truly not possible to have too many layers of security. You never know which one of those layers will be the layer that stops the bad guys. And, you know, I've often talked about it, actually I will a little bit here a little bit later, the asymmetric challenge that security faces because secure systems cannot afford to make a single mistake because any opening allows the bad guys in. Whereas the bad guys, all they have to do is find one. Well, that's offset, that asymmetry is offset if you're able to layer your security, if you can design the system to be multilayered. In that case, you could have some mistakes, and the bad guys still can't get in.

Okay. Unfortunately, not all mistakes can be qualified as mistakes. Last Friday, the day following Microsoft's big "Secure Future Initiative to advance security engineering" announcement that I just talked about, we learned from BleepingComputer that Trend Micro's Zero-Day Initiative had informed Microsoft on September 7th and 8th of four new zero-day vulnerabilities they had discovered in Exchange Server, one of which allowed for remote code execution. Microsoft acknowledged the reports, but decided that the flaws were not severe enough to warrant immediate attention and decided to put off the fixes until some later unspecified date. In other words, "Thank you, now go away."

Since ZDI strongly disagreed with this response, they decided to publish the rough descriptions, and actually the rough locations, of the four vulnerabilities under their own tracking IDs in order to at least warn Exchange admins about the security risks, even though unfortunately there's not much for Exchange admins to do about them at this point except to worry more than they already are.

Now, if something about this overall scenario seems familiar, where security researchers inform Microsoft of flaws that they have found somewhere which they believe are important, and after presumably examining those reports Microsoft decides that the problem is not worthy of their attention, you would be correct. We've been right here before. And if history continues to repeat itself, we also know what lies ahead.

Microsoft will leave this unpatched, some bad guy somewhere will pick up on the possibility of an outstanding unpatched remote code execution vulnerability in Exchange Server, and they will go a-hunting. Some time later, Exchange Servers will start being compromised in some mysterious new way that no one ever saw before. Except that, whoops, Trend Micro and Microsoft both saw it in September of 2023, and one of the two of them who could have done something to prevent it chose not to. Like I said, we've seen this whole thing play out before, and it's a shame.

So ZDI-23-1578 - that's their own tracking terminology - they said is a remote code execution flaw in the "ChainedSerializationBinder" class, where user data isn't adequately validated. This allows attackers to deserialize untrusted data. Successful exploitation enables an attacker to execute arbitrary code as "SYSTEM," the highest level of privilege on Windows.

So now our would-be bad guy also knows right where to look in Exchange Server. And surprise, surprise, the problem is deserialization. We've talked several times about the inherent difficulty of deserializing data securely. The process of "serializing" data takes some sort of formatted data structure, often a JSON structure, and turns it into a "blob" for storage or transmission. That's the thing known as serialization. The reverse process of deserializing the blob requires, yes, the interpretation of the data that the serializer produced. Interpretation. So we have some flaw in an interpreter in the ChainedSerializationBinder. Probably wouldn't be too difficult to find.

So just for the record, though they are not also remote code execution flaws, the remaining three are still some concern. We've got 1579, located in the "DownloadDataFromUri" method. This flaw is due to insufficient validation of a URI before resource access. Attackers can exploit it to access sensitive information from Exchange servers.

Then there's 1580. This vulnerability in the "DownloadDataFromOfficeMarketPlace" method also stems from improper URI validation, potentially leading to unauthorized information disclosure. And finally, 1581 is present in the CreateAttachmentFromUri method. This flaw resembles the previous bugs with inadequate URI validation, again, risking sensitive data exposure. So they all allow, those three, for some sort of unspecified information disclosure. While it's not running the attacker's code, which has been remotely supplied, which is what the first of these four can, information leakage can still be very valuable to attackers as part of a larger campaign.

The mitigating factor behind all four of these vulnerabilities is that they all require authentication. You need to be able to sign in as a user to this Exchange Server. So this may be the basis for Microsoft's dismissal of this as anything to worry about. But we've seen cybercriminals have repeatedly shown that they have many ways to obtain Exchange credentials. There's brute-forcing weak passwords, phishing attacks, purchasing them outright on the dark web, or acquiring them from info-stealer logs. So

once the bugs are found, the need for a credential for a specific Exchange Server might not pose an insurmountable problem.

Trend Micro's Zero-Day Initiative folks said that the only salient mitigation strategy would be to restrict interaction with Exchange Server. But what are you going to do, unplug it? Many businesses and organizations cannot operate without access to their Exchange Server. So anyway, I'm just putting this out there. We'll see here in the future whether Microsoft decides to slip some fixes into a forthcoming update, or whether the bad guys decide to do some reverse engineering of those now specified functions in Exchange Server, find the vulnerabilities, then arrange to get themselves an authentication onto Exchange Server and then get up to some mischief.

We will basically see whether we - and all of the history is going to repeat itself where Microsoft said, oh, no, nothing to see here, until there was. This is what happened with that horrible print server nightmare that we went through a couple of years ago where the researcher that found the vulnerability kept trying to tell them, over and over and over, look, this is a problem. You didn't fix it yet. Then when they said they did, it turned out they didn't. And then it ended up really coming back to bite them. So we'll see.

Leo: Deny, deny, fix.

Steve: Yes, exactly, yeah. And, you know, what's the recourse? None. The licensing agreement says, you know, if it works, great. If it doesn't, well, we tried. And again, you can't go anywhere else because Microsoft, you know, no one could argue that they're not a monopoly and that they don't have that power today.

There is a problem, not in Microsoft's camp, with something known as Apache ActiveMQ servers. They've been having some trouble recently, and unfortunately they are this week's mass casualty event, although because there's a much lower level of deployment of them...

Leo: I like how you go "this week's mass casualty."

Steve: I know.

Leo: So depressing.

Steve: This week's mass casualty event. Okay. So Apache ActiveMQ Server is a standalone message broker server which facilitates reliable high-availability messaging among clusters of computers. It's written in Java, and it's been around and evolving since 2004. So it's got some lineage there. The flaw, being tracked as CVE-2023-46604, is a maximum severity bug in the ActiveMQ scalable open-source message broker, that is, this server, which enables unauthenticated attackers to execute arbitrary shell commands on vulnerable servers. In other words, one of those as bad as she gets.

It's unclear when the Apache Foundation became aware of the attacks on ActiveMQ. But two security firms, Arctic Wolf and Huntress Labs, found that threat actors had been exploiting the flaw as a zero-day to deploy a remote access trojan known as SparkRAT since at least the 10th of October. Apache released security updates to fix the vulnerability 17 days later, on October 27th. So 2.5 weeks of window, and we don't know

how much earlier this was being done. But we do know that a patch being made available and patches being applied are two very different things in today's world.

In addition to the deployment of the SparkRAT remote access Trojan, ActiveMQ servers exposed to the Internet are being targeted in "HelloKitty" and another ransomware known as "TellYouThePass" is the name of the ransomware. So those two pieces of ransomware do share a common infrastructure, email addresses, cryptocurrency addresses, and so forth. So they're probably just versions of the same thing, or at least being run from the same ransomware group. And in terms of its spread, data from the threat monitoring service Shadowserver found that there are currently more than 9,200 Apache ActiveMQ servers exposed online, with over 4,770 - so more than half - currently vulnerable to exploitation.

So as I said, not as mass, as some of the masses that we've seen recently. But still not good. If the good guys can scan the public Internet to obtain a count of victims, bad guys can scan for potential victims to target just as easily. Needless to say, if your organization is using an Apache ActiveMQ Server with Internet exposure, you'll want to update it immediately, and also look around for any indication that the bad guys might have already entered your network and have set up some sort of persistence because you don't want that.

Okay. Given the sweeping scope of the mess with Citrix Bleed, which we spent some time looking at last week since, well, we talked in detail about the exploit, and that occurred on Halloween. The team at Mandiant, which is now a Google property, appears to be more on the ball and current about this than anybody else, since everybody else is just citing Mandiant's research in their own updates. Their last update was from last Thursday, which has added some interesting new pieces of information and helps to bring home, I think, the reality of the situation that is facing those 20,000-plus Citrix users whose network appliances have already been compromised. Not vulnerable, but compromised.

So on Thursday, Mandiant wrote: "Mandiant has identified zero-day exploitation of this vulnerability in the wild, beginning in late August 2023, as well as n-day exploitation" - meaning after it's been known, you know, and is larger than zero - "exploitation after Citrix's publication. Mandiant is investigating multiple instances of successful exploitation of CVE-2023-4966 that resulted in takeover of legitimate user sessions on NetScaler ADC and Gateway appliances. The session takeovers bypassed password and multifactor authentication. In this blog post, we will discuss artifacts that can be used to identify exploitation activity and highlight some of the post exploitation techniques we observed during the incident response investigations."

Okay. So they lay out what's already known about the vulnerability of the Citrix endpoints, the challenges of investigating vulnerable devices because the web server running on the appliance does not record requests, or errors, to the vulnerable endpoint. So there's no log of these things being made, making tracking them down extra tricky. And they note that they're not aware of any configuration change that can be made to force request logging for these particular endpoints. So, you know, those remote HTTPS queries. But what was most interesting, I thought, and the reason I wanted to share this update, was what they had to say about the post-exploitation activity they observed. In other words, what are some of the things the bad guys do once they bypass the system's authentication and gain access by grabbing one of those pre-authenticated tokens and then just using it?

So Mandiant explained. They said: "Following the successful exploitation of 4966, Mandiant has observed a variety of post-exploitation tactics, techniques, and procedures (TTPs). Once an actor was able to successfully achieve session hijacking, the threat actor performed actions including host and network reconnaissance of the victim's

environment, credential harvesting, and lateral movement via RDP," you know, remote desktop protocol. "Mandiant identified evidence of Active Directory reconnaissance using living-off-the-land binaries such as net.exe." As we know, "living off the land" is now an increasingly popular approach, meaning you don't need to bring any stuff with you, you just use this rich environment of command executables that Windows now ships with, net.exe.

Leo: It sure meant something different when I was a kid.

Steve: That's right.

Leo: Wow, I love it.

Steve: Living off the land. "Additionally, Mandiant has observed the use of the SoftPerfect network scanner (netscan.exe) to perform internal network enumeration. In several cases, the threat actor used 7-zip to create an encrypted segmented archive to compress the reconnaissance results." Because, you know, you don't want all your reconnaissance results to be too big. So you use an archiver like 7-zip specifically, to shrink them down. And boy, you know, those kinds of logs are typically going to shrink way down. "The threat actor then used the built-in certutil utility to Base64 encode the segments." So it's not just, you know, ASCII going out. It's somewhat obfuscated.

"In one case, certutil was used to decode multiple files related to credential theft. Mandiant observed the threat actor use e.exe to load d.dll into LSASS process memory. When run, the utility creates a memory dump file located at temp\1.png." It's sort of interesting. You give a binary memory dump a .png extension so that it looks like a known extension, and of course PNGs are binary, so presumably they just kind of get passed without much concern. That's interesting. Anyway, and prints success to the console when done. That's nice, so the bad guys know that everything worked fine. The memory dump file can be processed offline by the threat actor to extract credentials, that is, credentials from the LSASS process memory. Mandiant identified sh3.exe as a utility suspected to run the Mimikatz LSADUMP command.

In another instance, a threat actor used certutil to decode a file that Mandiant identified as a newly tracked backdoor that uses Slack as its command and control. Tracked by Mandiant as FREEFIRE, it is a lightweight backdoor written for .NET. FREEFIRE communicates to a hard-coded channel, a Slack channel, to retrieve commands and upload responses. It supports loading arbitrary .NET assemblies encoded as Base64 sent to it via chat commands. Mandiant observed FREEFIRE being deployed by a threat actor through the following certutil command. And then they go into it in more detail.

They've also observed the deployment, they said, of various remote monitoring and management tools following the successful exploitation of 4966. Currently, Mandiant has observed the deployment of Atera, AnyDesk, and SplashTop to establish and maintain a foothold following exploitation of 4966. They said: "Mandiant is investigating intrusions across multiple verticals, including legal and professional services, technology, and government organizations. Given the widespread adoption of Citrix in enterprises globally," they wrote, "we suspect the number of impacted organizations is far greater and in several different sectors.

"Mandiant," they said, "is currently tracking four distinct uncategorized groups involved in exploiting this vulnerability. We have observed some lower degrees of confidence overlaps in post-exploitation stages among these UNC (uncategorized) groups, like using

the same recon commands and utilities available on Windows. Two threat clusters used Mimikatz for dumping process memory. Notably, there were no overlaps in infrastructure between these clusters of activity. The exploits were sourced from different VPN provider IP addresses and previously compromised third-party devices."

Okay. So even though the attack is low complexity, easy to pull off, easy to launch, all indications are that well versed and very competent threat actors are behind these. They're using tried and true post-exploitation tactics to obtain a high degree of leverage in and persistence on their victims' networks.

So we are now inhabiting a world where the moment a patch to fix a remotely exploitable flaw is announced, powerful malignant forces jump on the patch, determine what was changed, design an exploit for any not-yet-patched devices, then race to take advantage of the newly discovered vulnerability, using it against anyone who did not instantly patch their devices the moment the trouble and its fix were announced.

Asymmetric warfare is notoriously difficult to fight. And this currently broken security model, which is the only thing we can call it, it is a currently broken security model, has these asymmetric aspects. Consider that a small group of miscreants only need to watch for security updates from the major appliance vendors. Yet on the other side, on the receiving side, every single person who is independently responsible for the operation of every deployed instance of every one of those devices spread anywhere in the world must be just as vigilant as that small team of bad guys. And on top of that, everyone everywhere must be ready to apply the fix at any time - night, day, weekend or holiday.

The only way I can see this evolving is for the high-end enterprise appliance world to make the same move that the small office/residential router and consumer desktop world has made of allowing these devices to be remotely autonomously updated without the need for the device's IT personnel to be involved. This feature should be enabled by default with IT personnel having the option to disable it if they understand and accept the risks that accompany doing so.

UDP packets are small, they are connectionless, and inexpensive to send. So every such device that has not been disabled could send a packet periodically to the device's manufacturer to check in for any updates. One tiny packet every 10 minutes would be more than sufficient. You could make it hourly if you wanted. But, you know, 10 minutes to be on the safe side. In the event of a critical update, an affirmative UDP reply would contain the URL of the update to download and apply, and the certificate of the remote web server could be pinned to prevent any forgery. The appliance would bring up an HTTPS TLS connection to download the updated module, install it, and reboot itself.

And of course I'm aware of the many arguments against this sort of autonomous upgrading. Its first appearance in Windows all those years ago caused quite a stir. You know, those old-timers among us were like, what? What a minute. Wait, we don't want this to be automatic. We want control of which security things we install. We want to look over the list before we say, okay, yeah, fine, do it all. You know, my own Unix servers send email to inform me of the packages that are in need of attention. This information they're obtaining without any assistance from me, although they do stop short of performing those changes autonomously.

So while autonomous patching of enterprise-class appliances may pose some risk, more than 20,000 users of this one device just had their networks deeply compromised because, for whatever reason, they did not install the patch that the bad guys were reverse engineering before that reverse engineering was turned into an active exploit. If they had, 20,000 individual network compromise disasters would have been averted. It seems to me that given the world we live in now, it is time to move autonomous patch updating from the consumer desktop and router - where it's now been proven to be

providing much more benefit than harm - to the enterprise's border equipment which are subject to swift attack, as we've all just been seeing with actually the past three, we're now at three recent mass casualty events.

And just for the sake of discussion, there are many possible compromise measures. For example, the periodic UDP packet sent by the device back to its manufacturer could contain the device's current build version and the latest current email address and cell phone number for the organization's IT cybersecurity team. That information could be configurable in the device's admin setup as an "in case of critical vulnerability, send email to and send a text to." That way, every one of the manufacturer's devices is pinging home base with the information needed to alert its administrators the instant any new and sufficiently urgent problem is discovered. It's difficult to believe that we're not already doing this as an industry.

And while we're talking here about any of this, since it's foreseeable that the first thing a compromised device might do is shut down that early warning update system, the device's manufacturer should have these periodic info pings continually updating a database, which would also prevent malicious changes to that information by retaining a history of previous contact information. In that way, the moment a serious problem was discovered, every admin could be made aware that they'll need to prepare for an update.

So I suppose my point is, we are really truly being lame about the way things are being done today, and I can't see any excuse for it. Like we have the technology to solve this problem and to prevent what are now becoming weekly multi-tens of thousands of networks being compromised in mass casualty events. I would argue that that Cisco, which was a web auth problem, should have never happened by policy. But again, the technology to fix this is at hand.

Leo: On we go.

Steve: We've just seen, and we continually see, the burden being placed upon frontline IT personnel to keep their networks safe.

Leo: Yes. It's a canary; right?

Steve: Oh, my god, yes. When I talked about ADAMnetworks, it's not a job I want.

Leo: No, no.

Steve: And if someone were to say, given all the evidence we've seen, that it's basically an impossible task, it would be difficult to mount a convincing counterargument. As always with security, the good guys must prevent intrusion everywhere, all the time, all at once - which I think was the title of a recent movie.

Leo: "Everything Everywhere All At One," yes.

Steve: That's right. But the bad guys only need to find one mistake, anywhere, one time. You know?

Leo: Deep respect for the people who do this.

Steve: It's not fair. It's not like a fair job.

Leo: No.

Steve: We've heard that IT security guys are stressed, and it's not surprising. I've mentioned this before, but I'll say it again. The job might not be for everyone. But if it sounds like it's a fit with your personality, the good news is the world is desperately looking for you. I saw a statistic recently indicating that there's about a 50% shortfall in IT security staffing. Something like four million empty job openings right now that need to be filled. One of the many things I've learned from our listeners is that they credit their listening to this podcast with giving them the inspiration to learn more about this subject which subsequently allowed them to move into the cybersecurity job market.

And I found that study. A quote from the study says: "The global cybersecurity workforce is estimated to have reached more than 5.5 million professionals. And even though that number is 9% higher than it was last year, four million experts are still needed worldwide to fill open positions across the industry." Okay, now, what I found is the 2023 "Cyber Workforce Study" in which they surveyed a record 14,865 cybersecurity professionals to share their unique perspectives on the state of the workforce.

I'm tempted to share more since this report is chock full of interesting data and statistics. But instead I have a link to its 84-page PDF in today's show notes, and it is this week's shortcut of the week, which I again numbered very carefully. So it's grc.sc/947. That will bounce you to this PDF of the 2023 Cyber Workforce Study. And really, as I was scrolling through this, I thought, oh, this is just full of cool stuff. So I really do commend our listeners take a look at it: grc.sc/947.

Okay. So where am I going with this? The job of controlling our networks, keeping them secure, is chaotic. The challenge, as I said, is highly asymmetric and arguably unfair. So the overworked cybersecurity professional needs all the help they can get. In a world of software vulnerabilities, how does one know how to start the day? One of the blessings this industry has created in an attempt to bring some form of order from this chaos is the Common Vulnerability Scoring System, which we are constantly referring to on the podcast, you know, CVSS scores.

The initial version 1 of the CVSS was instituted in February of 2005, the same year when we later began talking about these interesting issues every week. A little over two years later, in the summer of '07, it moved to version 2, where it sat until version 3 was introduced in 2015. That version lasted until 2019, when it was tweaked a bit to give us version 3.1, which is what we've been using up until now.

The reason the CVSS system has needed periodic maintenance and updating is that we're not living, as we clearly know, in a static world. The drama we're seeing playing out this instant, for example, with the Citrix Bleed vulnerability and the Apache message queuing problem, that didn't exist to nearly this extent before cryptocurrency because it was far less clear how attackers in Russia and North Korea could monetize their cyber intrusions. Now everybody knows. And vulnerable devices like Citrix's NetScaler technology hadn't yet been created. Since today's cyber landscape has changed, so must the metrics we use to characterize today's threats. To that end, work has been completed just now on the next generation of Common Vulnerability Scoring System, and we are therefore now at CVSS version 4.

So what's changed? There are four primary highlights. First, CVSS scoring metrics have been added and redefined to improve the granularity and clarity of CVSS scores. With the previous standard, it turned out like against today's threats it was common to have different types of vulnerabilities winding up being clumped around the same score, even though it no longer accurately reflected each one's severity. So more scoring metrics in CVSS v4 means a better spread across the entire scale.

Second, in keeping with today's world, we now have ICS, OT, you know, operational technology, and IoT-specific scoring metrics. This includes scoring metrics such as "Safety," "Automatable," and "Recovery," to let critical infrastructure operators know whether a security flaw just looks bad on paper, or if it's actually exploitable and dangerous to their networks.

Third, we also have new scoring metrics such as "Value Density," "Vulnerability Response Effort," and "Provider Urgency." Those have been added to help responders evaluate and prioritize vulnerabilities. Those last two, for example, "Vulnerability Response Effort" and "Provider Urgency," are intended to allow vendors to tell customers that a vulnerability needs to be patched ASAP. This is a capability that was not present in the current CVSS. And obviously having a "patch this ASAP" and have that actually mean something is something we need. So we now have it in version 4.

And finally, CVSS version 3's "Temporal" metrics group has been replaced with a new group called "Threat Metrics." Although this replacement group is intended to reflect the same exploitability and proof-of-concept availability as its predecessor, its application is significantly clearer now under version 4.

So while we'll still be seeing and quoting the same single 1-to-10 CVSS composite score, that score will now more accurately track the urgency presented by its vulnerability, and the detailed breakdown of that single score will provide cyber security professionals much needed additional details and may help them to decide how to start their day.

And a perfect example of this, I think, is that we keep seeing all of these 9.8s. Well, what? Why? How is it that everything is 9.8, except every so often, you know, a complete disaster meltdown is a 10.0. You know, it did feel like there was some nonlinearity or something to the way the CVSS v3.1 that we've been using for the last four years had been operating. I expect we're going to see a different scale of CVSSes. And so we should be prepared also not to apply the numbers coming from version 4 against what we've been used to seeing in version 3 and 3.1. They may look like they're less severe. Probably what they're doing is doing a better job of, like, reflecting a non-clumpy, more uniform scale so that, you know, the bad ones are probably more rare, but what they signify is probably more significant. So, yay. That's all for the best.

Leo: It's inherently, though, subjective; right? There's no - or do they have some weird objective criteria?

Steve: No, they actually have a calculator where you go to a website, and you say yes, no, yes, no, yes, yes, no, no, yes, and so forth. And then it gives you the score based on a rigorously pre-established formula. Yeah. So it's not just oh, my god, you know, this one made me nauseous. There's actually math behind it.

Leo: Oh, okay. And the people, and the yes or no questions are concrete. They're not, well, was this really bad?

Steve: Yes. They are, they're concrete.

Leo: Okay.

Steve: And very specific. And there's actually, if you go to the central CVSS repository, there is a breakdown, a multidimensional breakdown of, like, you know, which are the ways in which this thing is bad that all worked together to create this composite score? So, you know, there's actually a lot of science behind it.

Leo: Okay, okay.

Steve: Okay. So I have a soft spot in my heart for Ace Hardware, Leo.

Leo: Me, too. We love our Ace Hardware.

Steve: Yes. We have several of Ace's 5,700 retail stores in our area, and I have to say they have a truly amazing array of random little hardware bits.

Leo: Yeah, there's always some guy in suspenders with a walrus moustache.

Steve: Oh, he's wonderful.

Leo: And you can say "My faucet's leaking," he's oh, yeah, come over here. Yeah.

Steve: Yeah, he kind of shuffles along and brings you to...

Leo: Yeah, you've got him, too.

Steve: Oh, yeah, yeah. I think we're all sharing him.

Leo: Yeah.

Steve: So anyway, many of my own projects were saved right, you know, like in the middle of the day when I needed a particular size bolt or washer. You know, there are still some things that are difficult to do online.

Leo: Yeah.

Steve: I would argue, you know, trying on clothes is difficult, and getting exactly the washer that you need to fit in a tight space, or how many of them stack up in order to

create a shim of the required size. So anyway, when you need to match a bolt to a nut, there's no substitute for being there.

Anyway, I bring this up because nearly 200 of their servers - yes, we know where this is going - and 1,000 other systems were hit by a cyberattack the day before Halloween, so that was last Monday, October 30th. The attack impacted their ability to pick up new customer orders. And also other impacted systems included their warehouse management systems, reward points tracking, their tech support call center, and the company's mobile assistant.

Despite the attack, the company's 5,700 retail stores have remained open, although with somewhat reduced activity. And I haven't needed any bolts recently, but it's good to know that I can still get them there. And we did actually lose one Ace retailer, the one that was closest to me, I think it was as a consequence of COVID. It never recovered from the real slowdown. But we still have one that I know I can get to if I need a specific thing. You know, your store probably has it, too, Leo. It's - I can't remember the name. It begins with H. It's like Hildebrand or HY-Guard or something. But they have like, in mine, they have multiple aisles of identical, like, slide-out trays.

Leo: Yeah, yeah, bins, yeah, yeah. With all the different stuff, yeah, yeah.

Steve: Yeah, yeah. And so - yes. And it is all provided by one company that, like, stocks all those things.

Leo: They always have the one you need.

Steve: Yes. It's amazing. Even if it's some, like, backdoor spring collapser or something. It's like, my god, there it is.

Leo: The one we had burned down about 15 years ago, burned to the ground, and it was on a big deal, it was like on the Fourth of July. But they've rebuilt it. But it was really nice because it had creaky old floors. But they kind of preserved the spirit of it. I love old hardware stores. I have a lot of stuff from that Ace. They probably also have a lot of my stuff in their database.

Steve: Yeah. The best extension cord I've ever found is like this supple, I don't know, it's got to be like highly braided. It's just amazing. It just feels so good. Anyway.

Leo: Yeah, it's fun.

Steve: Enough of that nonsense. Remember that bizarre plan Google floated a few months ago, which would have given websites absolute control over the extensions and other features that could be used by anyone visiting a website that wished to impose such control and restrictions over their browser?

Leo: Is that the Web Integrity API that they did?

Steve: Yes, exactly. And it was dubbed the Web DRM because, you know, nobody likes digital rights management.

Leo: Right, because that's what it was really. I mean...

Steve: Yes, it really was. The good news is that plan is dead.

Leo: Oh, good.

Steve: And I'm impressed that Google didn't make a larger push for something that really didn't seem to be in the best interest of the end user. We talked about it a bit briefly, but there wasn't enough known at that point to really take it very seriously. The only upside, like the only reason I could see that it might be useful was that, as we know, users are not very judicious in their choices of browser extensions. So you could imagine like a banking site, for example, wishing to enforce much tighter security when people visited some of its more secure services. So I could imagine that.

But the downside was that, for example, sites might just decide to restrict the use of ad blockers by disallowing their use. We know they don't like them. So why not just make them not work on their site? So anyway, my hope is that Google did not kill this because they have figured out some better way to do something similar because this really seems bad. And if they did, I would hope that Mozilla would not choose to follow with Firefox. So we'll see. But for now, it's not going any further.

Okay. We were just talking about the first of the three recent mass casualty events being the Cisco IOS XE router attack. And that was the one which preceded the Citrix Bleed mess. And as I mentioned before, it was the result of an easily preventable web authentication bypass which, as I ranted at the time, was entirely foreseeable and unnecessary because no web UI administration should ever be placed on the public Internet. We all know better. Everybody should know better. Cisco should know better. It shouldn't even be an option since there are now many far more secure ways to do the same thing. And as I was putting this together for the podcast I came up with a new slogan because, you know, it's there because a web UI is so easy to use. So "Ease of use is no excuse." So, yeah.

Anyway, this all popped back up because Cisco's own Talos group just published a full technical analysis of what they call "BadCandy." It's the implant that's being deployed on those compromised Cisco routers thanks to those zero-day vulnerabilities. In their report, Talos notes that the BadCandy malware has evolved, now in its third major version, demonstrating that the threat actors behind this are still actively modifying their attacks to maintain access to those compromised boxes. And remember, there were thousands, tens of thousands of them.

They also noted that the latest version 3 modifications which have been made to BadCandy appear to have worked, since the Shadowserver Foundation who had been monitoring the attack's progression over time has stopped detecting any infected systems, although we know 100% for sure that they're not all patched. Presumably, the Shadowserver Foundation and others, other security researchers, had long ago captured the IP addresses of the infected and vulnerable systems. But no one coming along now would see that anything was amiss. Although in fact I'm sure it's just a cornucopia of compromised networks. Probably the big problem now is just sorting through them all and deciding which ones to go after first.

Now, I know that before this next bit of news I'm supposed to remind our listeners that Bitwarden is a sponsor of the TWiT network, as if we weren't all already aware of and pleased with and even grateful for that fact. So what's the news? With release 2023.10.0 of the Bitwarden browser extension, it now fully supports FIDO2-style Passkeys. Bitwarden's mobile clients have not yet caught up, but this is acknowledged, and it's on their development roadmap. Meanwhile, the browser extension appears to be ready for prime time. I have a link in the show notes to the full 2023.10.0 release notes and another link to the specific page they've got there now discussing Bitwarden's browser extension support for Passkeys. I've not tried it myself, but from a quick scan of that page it appears that everything is there.

At the bottom of the "Storing Passkeys" page they have a short Q&A, I think it just had three FAQ points, where one of the questions, the middle one, asked: "Are stored passkeys included in Bitwarden imports and exports?" To which they reply: "Passkeys imports and exports will be included in a future release." So that's not there yet. But they clearly recognize the need. And as we noted when we recently talked about this, apparently the slow-moving FIDO group are involving themselves in the creation of the import/export format. Although that's making us wait, we definitely want all Passkeys clients everywhere, everyone's Passkey client, to support a single common, well-designed, unified cross-platform standard. So I think we all should be quite happy to wait for that.

Okay. Recall the tweet from a listener named Victor from a few weeks ago. His Twitter DM was dated October 18th. In it, he wrote, he said: "I powered on a couple years old desktop that had been unpowered for about a year. It took ages before the desktop was loaded, no errors anywhere, but I decided to try your ReadSpeed. And look at those SSD speeds!" exclamation point, he wrote. "Is it time to invest in SpinRite now?" He said: "If SpinRite fixes this, I will try to encourage my employer to get a site license," meaning just purchase four copies, and then you can use it on all the machines there. He said: "Thank you, Mr. Gibson. Victor, long-time SN listener. Keep up the good work. To 999 and beyond."

Okay. He attached a screenshot from ReadSpeed which I described at the time. It showed and explained exactly what he was describing. You know, ReadSpeed takes a benchmark, read performance benchmarks at five locations across the drive - the beginning, the one-quarter point, the middle, the three-quarter point, and the end. And what you'd expect on any SSD, the reason I did these five snapshots was that we all know that spinning hard drives are slower at the end because that's the inner tracks where there's less data since the drive is spinning at a uniform speed. If you've got less data around the circumference, your transfer rate is going to have to be much lower. Typically it's like half the speed as the beginning of the drive, the outermost circumference. So I designed ReadSpeed just to take, you know, in order to sample those five points.

To our stunned surprise, we discovered many people's SSDs were slower, much slower at the beginning of the drive, even though an SSD being solid-state you'd expect it would be uniform across all five snapshot points. Not so. And in Victor's case, he had an extreme case. Remember that the beginning of his drive was 2.2MB per second. The 25% point was 482.5. The 50% point was down to 53. The three-quarter point was also bad at 13.8. And the very end was 323.7. So this drive is like, across it is, like, bad. And in fact it peaks at 482 at the 25% point for who knows why.

Anyway, yesterday, just on Monday, yesterday, when catching up with my Twitter feed for the podcast today, I found his follow-up. He did purchase a copy of SpinRite 6.0, then used his 6.0 license to immediately grab the 6.1 release candidate. Here's what he wrote. He said: "Now, Mr. Gibson, I have some results for you, and possibly the listeners. I ran

SpinRite 6.1 on Level 4. It took 28 hours, reported 6,199 command timeouts, found and repaired 183 defective sectors." Remember this is on an SSD.

He said: "For comparison's sake, here is a new screenshot of ReadSpeed. Now the PC behaves like one would expect from an 8-core Intel i9700 with 64GB of RAM." And so we have that. Where the beginning of the drive was at 2.2MB per second, it's now 430MB per second. The one-quarter point didn't change. The middle went from 53.2 up to 508MB per second, like almost 10 times faster. The 75% point went from 13.8 to 504. And the end of the drive went from 323 to 513.8. So much more uniform and way faster. And also this SSD had a ton of problems.

Okay, now, under SpinRite 6.1, a read scan of a half terabyte directly connected, meaning not USB, but directly connected, this is the drive his system boots from, would have taken less than an hour. We can easily do half a terabyte an hour and even faster on an SSD. But rescuing and resuscitating that very sick SSD required rewriting its recovered data. And actually all of its data. So that would slow things down. But not by nearly that much, probably only, you know, it would take an hour or two normally to do a Level 4 pass on an SSD. But that SSD was clearly in seriously bad shape, and it sounds like it made SpinRite work a lot to pull it all back from the brink.

So SpinRite 6.1 is obviously highly effective for today's solid-state storage, in addition to what it has always been able to do for electromagnetic spinning drives. What we've learned is that it turns out that electrostatic storage is prone to long-term charge degradation through several different mechanisms. And this only promises to become worse as engineers continue to succumb to pressure from their managers to squeeze ever more data into ever smaller and fewer storage cells.

The good news is SpinRite v6.1 can resolve those problems today. It is not as optimal as version 7 will be, but it works now, and I'm not stopping once 6.1 is published. It's a big step forward, but I've got much more on the way, as soon as I'm able to get away from DOS, which is like going to be a big treat for me.

Okay. We've got some closing-the-loop feedback. Sam Miorelli, he said: "Hey @SGgrc, listening to SN-946" - that was last week - "on IPv6. Spectrum, the main ISP for Central Florida for Tampa, Orlando, and the Space Coast, IPv6 is still notoriously problematic. For example, Pixel 3 and later phones go into WiFi disconnect loops when you let IPv6 hit cheap and good routers. For example, I have a Netgate USA SG-2100. This is well documented on Reddit. Outbound DNS queries also frequently have long periods of time when IPv6 is enabled that they simply black hole on Spectrum, then randomly fixed for a while, then bad again, for months."

He says: "ISP shrugs." He said: "I fear the day we're forced to switch to IPv6 given how terrible the backend tech is maintained for home ISPs." So he's talking from the ISP side. And as we know, he was following up on my somewhat pessimistic appraisal last week of what appears to be the true current state of IPv6. It'll be there when we really need it. Actually now I would say "hopefully" because it's looking worse than I thought. But until really, we REALLY need it, is in all caps, bold italics, and underlined, all of the prevailing evidence points to everyone doing everything they can to hold onto IPv4 until there's really no other choice. I have 18 IPv4 IPs, and I treasure them.

Leo: Holy cow. Yeah.

Steve: Yeah. Also, Bob Grant provided some additional terrific feedback about IPv6. He said: "Hi, Steve. As an IPv6 proponent for a number of years, I listened with interest to your answer to reader email last week where you said nearly everything appears to be

IPv6 ready. I thought I'd share my experience. It is certainly the case that most everyone's routers and many networks are dual stack with both IPv4 and IPv6 support. In the case of my ISP, I am able to request an IPv6 /48 prefix which is 256 separate /64 networks, so I can have as many as 256 separate networks, each having a full /64 IPv6 subnet. My firewall only allows established connections back in, so there's no additional security issue over my IPv4 NAT."

Meaning he's observing that IPv6 does not NAT. So individual systems get their own IP because there's just so many of them, you don't need NAT any longer. But you do still want firewall functionality. He said: "I recently upgraded my network and WiFi access points so it's trivial to segment multiple SSIDs into separate VLANs going back to my OPNsense router." Okay. So I'd say that we have indeed at this point established Bob as an IPv6 proponent.

He says: "As an experiment, I decided to set up an IPv6-only network where only IPv6 IPs and DNS would be used. I was quite disappointed to discover how few websites worked with IPv6 only. Huge sites like nfl.com, twitter.com, and many well-known universities whom I won't embarrass by naming, and others like bitwarden.com all fail to load with an IPv6-only connection." And I'll just note that all of my own servers at GRC are among those, too, which are still IPv4-only.

He says: "Many other sites work when using www.site.com but fail to redirect when using just http://site.com. My three credit unions' landing pages load under IPv6, but the financial backends hosting the login process and displaying account balances fail because they're IPv4 only. Even Microsoft fails after the landing page because the login.live.com is not IPv6 enabled. I noticed many sites' web pages load only because they use a content delivery network like Cloudflare or Akamai that supports both IPv4 and IPv6 at their border, thus proxying for their clients' IPv4-only web servers. Kudos," he finishes, "to Google, Amazon, Netflix, Facebook, Stanford, MIT, Harvard, and the federal government for fully implementing IPv6. I hope some of this is useful. Bob."

And yes, very useful and very interesting, Bob. Thanks for sharing, giving us an update on our IPv6 reality check. So we are not IPv6-ready today, that is, we cannot abandon IPv4, just as we cannot abandon TLS v1.2. As we learned, only one-third of servers are able to do 1.3 connections, so we still need 1.2.

CLT Cyber Security tweeted: "Hey, Steve. As security pros we know what to do, but I'm having trouble explaining why this is the right way to my company. We have DigiCert TLS certificates for our websites. We need to keep those private keys secure. But if our server was ever compromised by a random attacker who obtained our private keys, what could they really do? Just trying to better articulate this to non-security pros. Thanks, and to the nines and beyond."

Okay. The danger presented by the compromise of a server's TLS private certificate is one of impersonation, since it's only the server's sole possession of that private key certificate that allows it to assert its identity. So if someone else is able to assert that their server is your server, this paves the way to an impersonation attack. Depending upon who you are, that might be either a big deal or not so much. If this was a site that really mattered to its visitors in some way, then the consequences could be significant.

However, even though the theft of the certificate may pave the way to such an impersonation attack, there's still a lot of pavement to traverse to pull off a working impersonation attack. The biggest roadblock to implementing the attack is that the web browsers or other connecting clients who would be spoofed by this need to believe that they are actually connecting to the authentic server. In other words, they need to look up the IP address of the domain of the authentic server and then send TCP traffic back and forth to the IP that was looked up.

In practice, that means that either the DNS lookup needs to somehow be poisoned to return the attacker's server IP, or the victim's IP traffic needs to be intercepted and redirected on the fly to the attacker's IP. One way or another, the domain name the client believes it's connecting to must match one of the domains that certificate authenticates.

So either DNS subversion or dynamic traffic interception must somehow be provided. If that sounds like a high bar to reach, it can be; but it's entirely dependent upon the specifics of who, or what population, is targeted for spoofing. At the beginning of today's podcast I was talking about layered security. Here's another example of layering. Losing control of a website's certificate is not immediately the end of the world, since other layers are still in place to provide some protection. But having exclusive use of a website's certificate is not a layer you want to give up.

This is why the Internet world went nuts several times in the past during this podcast. One of those times was Dan Kaminsky's famous discovery of DNS spoofability, which also would not have been the end of the world, though it would have been bad for HTTP without TLS. Even so, the integrity of DNS wasn't a layer of security that anyone wanted to lose. And similarly the Heartbleed flaw, that potentially allowed some server web certificates to escape, got everyone's attention big-time because, again, it would strip a layer of our multilayered protection.

And I think that perhaps this also helps to put the never-exploited-as-far-as-anyone-knows Spectre and Meltdown vulnerabilities into a useful light. It might at first appear that the industry was way over-concerned about what was a purely theoretical vulnerability for which no known attack has ever succeeded. But again, robust interprocess isolation, which Spectre and Meltdown both threatened, is another layer. And in today's heterogeneous cloud computing landscape it's a particularly critical layer. So I cannot think of an instance where having too many layers of security is a bad thing, as long as it's not way over the top and gets in the way.

Craig from Scotland tweeted: "Was just listening to SN-941" - so he's a few weeks back - "and the part about public key crypto and factoring primes." He said: "It got me wondering, how likely is it that there could be collisions in the primes chosen by two different people? Or would it be feasible to create a rainbow table of factored primes allowing the discovery of the private key using a quick lookup of a public key?"

So that's a brilliant observation, and in the past - that is, the notion of a collision of primes. And in the past that was discovered to be happening with somewhat horrifying regularity. There were problems with the quality of some of the early random number generators which tended to choose and then test for primality the same large prime. So, whoops. Two or more completely unrelated servers would coincidentally be sharing the same public/private key pairs. Not due to any collusion between them, well, except for the collusion of them both using the same poor sources of entropy.

What was found to be happening was that the servers were booted and were immediately being asked to produce a certificate. So the server hadn't yet had time to collect sufficient entropy from the environment, and it could happen that two completely separate servers would both wind up picking the same keys.

And then into this we add the birthday paradox. That teaches how quickly the number of collisions between pairs of unrelated items increases as the number of possible interactions increases. There's not a huge danger from sharing the same keys. But it's certainly not zero. First, you would need to compare your server's public key with the public key being sent by everyone else's server. If you did find a collision, since you know your server's matching private key, you now also know the colliding party's private

key. Now, that's not good. But we already observed that just having that only removes one of the multiple layers of protection needed to exploit any advantage.

The takeaway here is that we don't want to be inadvertently sharing our private key with anyone else. So the best way to assure that is to be certain that the process which is picking keys is using the highest quality possible source of randomness for its key guesses.

And finally, Chad Cosby says: "Hi, Steve. I'm curious if you would share how and how often you run SpinRite on the drives in your Synology NAS." He says: "I, too, use a Synology, and it feels like an absurd oversight that I trust my most valued data to an occasional glance at the Drive Health meter within DSM." Which is Synology's management console.

So Chad, I suppose the question is how much redundancy you're using. I've never bothered to run SpinRite on any of my RAIDed drives. I have four-drive RAID arrays everywhere. Every one of GRC's servers is running four-drive RAID, as are both of my Synology NASes, and my one still-standing Drobo, although I think it has five drives in it. In every instance I'm running RAID 6. So that allows me to lose any two drives at the same time without any data loss. And once not too long ago I was flying a bit nervously with no reserve on one server until I could get two replacement drives for it. I actually had drives ready for it, but I learned that it would not allow me to mix SSDs and spinning drives in the same array.

So far, I have never lost any data, and I've actually had more trouble with my SSDs than with the spinners. Some spinners just appear to run forever, and others seem to tire quickly. They last long enough that I wouldn't really call it infant mortality. It's more like teen angst. Anyway, I'm replacing my SSDs now with spinning drives. And with them being so ridiculously huge and inexpensive I will always be running with RAID 6. And in that case I welcome any drive that gets tired and no longer wants to play, you know, just let me know, and I'll slip in a replacement. But, yeah. As the long-time publisher of SpinRite, I believe in redundancy. And so I've got as much as I can afford in the available space.

Oh, and I do have two last pieces. Real quickly, John Carling had a tip for our listeners. He said: "Hey, Steve. Listening to 946 and hearing about the requests to extend Windows 10 EOL. The group that did all the testing, are they aware of the recently revealed command line argument to Windows 11 setup?" And then he reveals it: Setup /product server.

He says: "This will install Windows 11 on a Windows 10 box that previously failed TPM 2.0 requirements. I've done it on two laptops and one desktop myself, and they work just fine." So there is a way, apparently just documented and just discovered. Setup /product server, you're telling setup that you're running a server rather than a desktop so don't bother me with all this TPM nonsense and hopefully processor generation and all that, and you just get it without any muss or fuss.

And lastly, Michael Foley, he said: "Just watched the latest episode. Now I have to check all my uses of snprintf for the last 30 years." And he said: "GRC is also the acronym for the French name of the RCMP [which is the] Royal Canadian Mounted Police." In French it's...

Leo: Gendarmerie Royale du Canada.

Steve: Du Canada, exactly, GRC. So now we all know.

Leo: It's many things, actually.

Steve: Yeah, it is. There are many.

Leo: It's an overloaded acronym or initials.

Steve: That would explain the offers I get for GRC.com. I think the highest one was $50,000.

Leo: Wow.

Steve: Somebody is willing to pay for GRC.com.

Leo: Wow.

Steve: So of course I'm not selling it. But, you know, when I'm 85 or 90 it's like, yeah, okay, fine. It'll be about time around then.

Leo: When I do a search for just GRC, I get IBM first. That's their Government Risk Management and Compliance solution.

Steve: Ah.

Leo: And then I get the home of Gibson Research Corporation. So you're coming in okay. You're doing all right. And I don't see the Mounties here anywhere. Article 45. What is that?

Steve: Oh, boy.

Leo: Oh, boy.

Steve: So there's a storm brewing again in the EU.

Leo: Yes.

Steve: It's been brewing for some time, and it appears that we have another case of politicians mistakenly believing that they're able to simply dictate the terms and conditions under which tech companies will serve their populace regardless of the implications to that populace's security and privacy.

We all just saw something similar come to a head, of course, with the attempt to force backdoors into all encryption services. How'd that turn out? Uh-huh. Every messaging provider simply said: "No, thank you, we'll just leave, and you and your citizens can figure out what to do without us." The result was the addition of a nebulously worded "if it's technically feasible to do without weakening security" clause, which was every strong encryption provider's get out of jail free card.

Now we're moving into a similar challenge where, believe it or not, the EU might very well find itself and its citizens without any web browsers, or at least needing to return to the good old days of HTTP.

Leo: Wow.

Steve: Uh-huh. The controversy revolves around a made-up thing known as, I guess you'd call it "QWACs."

Leo: QWACs, yeah.

Steve: QWACs, Q-W-A-C-s, which stands for Qualified Website Authentication Certificates. So these QWAC-y things are a specific EU form of website certificate defined back in 2014 with the EU's eIDAS regulation. Okay, what? eIDAS stands for electronic ID Authentication and trust Services. And actually, Leo, we talked about this a couple years ago when this nonsense surfaced once. It was something about the EU wanted to be able to display more information to users of websites, so they were going to, like, add some additional something onto the connection, like a banner or something.

Leo: Ugh.

Steve: I know. Anyway, so eIDAS is an EU regulation. It did pass nine years ago in 2014. Its stated purpose is governing "electronic identification and trust services for electronic transactions." After it passed in 2014, its various provisions gradually took effect over time between 2016 and 2018. That regulation, which never actually did much and was largely ignored, and which by the way we did, as I said, we talked about it at the time, has been under review and, in an upcoming process for the past several years, looks like it's coming to a head.

It appeared to be in fact going off the rails last year, and the tech industry did what it could back then to say: "Hey guys, this is not looking like something we're going to be willing to do for you." But apparently the politicians just figured that they could enact any laws they wanted to, and those techie geeks who were always complaining about something would have no choice other than to comply. Uh-huh.

So about a year and a half ago, back in March of 2022, a who's-who of global Internet security governance - in fact it's two pages of cosigner's names and affiliations - wrote an open letter addressed to "Dear Honourable Member of the European Parliament, Dear Member of TELE Working Party." And that letter begins, here's just the first few lines. They wrote: "We the undersigned are cybersecurity researchers, advocates, and practitioners. We write to you in our individual capacities to raise grave concerns regarding certain provisions of the legislative proposal for a European Digital Identity framework (the 'eIDAS revision'), and their impact on the security on the web.

"While we understand that the intent of these provisions is to improve authentication on the web, they would in practice have the opposite effect of dramatically weakening web security. At a time when two-thirds of Europeans are concerned about being a victim of online identity theft, and over one-third believe they are not able to sufficiently protect themselves against cybercrime, weakening the website security ecosystem is an untenable risk. We therefore urge you to amend the revised Article 45.2 to ensure that browsers can continue to undertake crucial security work to protect individuals from cybercrime on the web."

Okay, now, to say that this letter - it goes on at some length. But to say that this letter appears to have fallen on deaf ears would be an understatement. That was a year and a half ago. The near-final text for eIDAS 2.0 has now been agreed upon by the EU's negotiators, and it appears to be even worse than the earlier draft. So now there's a new letter which, as of two days ago, on Sunday, has been signed by 466 scientists and researchers across 36 countries, as well as numerous NGOs. And Google also just added their name to the document. In this day and age, what this document describes is somewhat astonishing, and I need to share the first few paragraphs so that you'll get a feeling for what now hangs in the balance.

This was addressed to Dear Members of the European Parliament; Dear Member States of the Council of the European Union. "We the undersigned are cybersecurity experts, researchers, and civil society organizations from across the globe. We have read the near-final text of the eIDAS digital identity reform which has been agreed upon on a technical level in the trilogue between representatives from the European Parliament, Council, and Commission. We appreciate your efforts to improve the digital security of European citizens.

"It is of utmost importance that the global interactions of citizens with government institutions and industry can be secure while protecting citizens' privacy. Indeed, having common technical standards and enabling secure cross-border electronic identity solutions is a solid step in this direction. However, we are extremely concerned that, as proposed in its current form, this legislation will not result in adequate technological safeguards for citizens and businesses as intended. In fact, it will very likely result in less security for all.

"Last year, many of us wrote to you to highlight some of the dangers in the European Commission's proposed eIDAS regulation. After reading the near-final text, we are deeply concerned by the proposed text for Article 45. The current proposal radically expands the ability of governments to surveil both their own citizens and residents across the EU by providing them with the technical means to intercept encrypted web traffic, as well as undermining the existing oversight mechanisms relied on by European citizens. Concretely, the regulation enables each EU member state, and recognized third-party countries, to designate cryptographic keys for which trust is mandatory. This trust can only be withdrawn with the government's permission. See Article 45a(4).

"This means any EU member state or third-party country, acting alone, is capable of intercepting the web traffic of any EU citizen, and there is no effective recourse. We ask that you urgently reconsider this text and make clear that Article 45 will not interfere with trust decisions around the cryptographic keys and certificates used to secure web traffic.

"Article 45 also bans security checks on EU web certificates unless expressly permitted by regulation when establishing encrypted web traffic connections, see Article 45(2a). Instead of specifying a minimum security measure which must be enforced as a baseline, it effectively specifies an upper bound on the security measures which cannot be improved upon without the permission of ETSI."

Okay, then skipping ahead a few pages, here's some detail that's actually difficult to believe, but it's true. Same group writing, little bit later in this letter. "The current text of Article 45 mandates that browsers must accept any root certificates provided by any Member State, and any third-party" - I can hardly believe I'm reading this - "and any third-party country approved by the EU. This will have severe consequences for the privacy of European citizens, and security of European commerce, and the Internet as a whole."

They explain: "Root certificates, controlled by so-called certificate authorities, provide the authentication mechanisms for websites by assuring the user that the cryptographic keys used to authenticate the website content belong to that website. The owner of a root certificate can intercept users' web traffic by replacing the website's cryptographic keys with substitutes he controls. Such a substitution can occur even if the website has chosen to use a different certificate authority with a different root certificate.

"Any root certificate trusted by the browser can be used to compromise any website. There are multiple documented cases of abuse because the security of some certificate authorities has been compromised." And of course we covered all this in the early days of the podcast. "To avoid this, there exists legislation that regulates certificate authorities, complemented by public processes and continuous vigilance by the security community to reveal suspicious activities. The proposed eIDAS revision gives Member States the right to insert root certificates at will, with the aim to improve the digital security of European citizens by giving them new ways to obtain authentic information of who operates a website." I know, Leo. "In practice, this does exactly the opposite.

"Consider the situation in which one of the Member States, or any of the third-party states recognized now or in the future, were to add a new authority to the EU Trusted List. The certificate would have to be immediately added to all browsers and distributed to all of their users across the EU as a trusted certificate. By using the substitution techniques explained above, the government-controlled authority would then be able to intercept the web traffic of not only their own citizens, but all EU citizens, including banking information, legally privileged information, medical records, and family photos.

"This would be true even when visiting non-EU websites, as such an authority could issue certificates for any website that all browsers would have to accept. Additionally, although much of eIDAS 2.0 regulation carefully gives citizens the capability to opt out from usage of new services and functionality, this is not the case for Article 45. Every citizen would have to trust those certificates, and thus every citizen would see their online safety threatened."

And lastly: "Even if this misbehavior was discovered, under the current proposal it would not be possible to remove this certificate without the ultimate approval of the country having introduced the certificate authority. Neither eIDAS's article 45 nor any provisions in adjacent EU legislation such as the NIS2 Directive provide any independent checks and balances on these decisions. Further, European citizens do not have an effective way to appeal these decisions. This situation would be unacceptably damaging to online trust and safety in Europe and across the world. We believe this legislation text must be urgently reworked to avoid these serious consequences by clarifying that eIDAS does not impose obligations to trust cryptographic keys used for encrypted web traffic."

Okay. So this letter goes for seven pages before we get to the 14 pages of signatures by everyone in the world in a place of authority who knows anything about the way our Internet security and privacy ecosystem is put together. Mozilla authored their own letter which was dated last Thursday, November 2nd. It was cosigned by the Bytecode Alliance, Cloudflare, DNS0.eu, Fastly, the Internet Security Research Group (ISRG), the Linux Foundation, Mozilla, Mullvad, OpenSSF, and Sigstore. I'll only share the first line.

It begins: "Dear Members of the European Parliament; Dear Members of the Council of the European Union. We represent companies that build and secure the Internet. Our organizations are either based in Europe or offer products and services in Europe. We write to express our concern with the proposed eIDAS legislation. We appreciate efforts to use rulemaking to strengthen the security of the Internet and the leadership role that Europe has taken in fostering cross-border interoperability. However, leadership comes with a greater responsibility to consider the broader implications of changes." That's just the top of basically them saying the same thing. It expresses the same concerns and issues as the previous open letter.

So the question now is, what happens next? A full year and a half ago the legislators were warned about this and were given a heads-up, with a full, detailed, careful, and respectful explanation. They were very clearly told, "Do not proceed down this path." They clearly blew it off, ignored it completely, and since then the wording of Article 45 has only grown more intolerable. We've observed that in high-level, high-stakes politics it's necessary to give the player who's holding the weaker hand a face-saving way to back down.

This happened with the encryption debate where the loser in that struggle created their own way to save face. But that didn't happen until those holding the stronger hand - the encryption service providers - were finally forced to deliver the ultimatum: "If you outlaw our use of unbreakable encryption, you will leave us with no option other than to withdraw our then-illegal services from your territories."

So is this going to come to that? Is this going to get to ultimatums? At this point, it appears so. And this will be another important juncture in the evolution of our Internet. Governments are going to learn, again, that they are smaller than the technology which they and their citizenry have grown to depend upon. It's theirs to use, but not to control.

Leo: This is terrible.

Steve: I have at the bottom of the show notes four links: the original open letter from March of 2022, 18 months ago. I've got today's 21-page, 14 that are signatures, updated letter to the EU. I've got the link to Mozilla's open letter, and the entire text of the proposed and agreed-upon eIDAS 2.0 legislation for anyone who's interested. But yes, Leo, I mean, it won't come to pass. It can't. I mean, it can't. It is exactly analogous to what just happened with the encrypted messaging providers, you know, I mean, it's worse than that actually. And, you know, and they're saying, oh, but we want to be able to intercept connections in order to add additional information banners to people's web pages.

Leo: No. No.

Steve: And it's like, sorry. We're not letting you do that.

Leo: By the way, I mean, I guess it's just EU if the browser guys make sure it's just EU. But, you know, you start to add certificates into your browser, it could easily be global. They're going to vote on this tomorrow behind closed doors. So it could be approved, according to the EFF, as early as tomorrow, November 8th.

Steve: Well, it'll certainly have some rollout period, you know, a grace period, and there'll be some deadline. And it's just it's, I mean, everyone's going to have to say no.

Leo: Yeah. I guess. I mean, what do you do? It's one thing if you're Mozilla to say no. It's another thing if you're Google to say no.

Steve: I believe that Google has altered the security of the guts of their browser so that they now use the hosting operating system's root store.

Leo: Ah. That would be - that's a good way to pass the buck. So then it's Microsoft or Apple or - yeah.

Steve: Uh-huh.

Leo: That's interesting. Yeah, instead of putting these CA roots into the browser, you put into the OS. So Edge I'm sure does that.

Steve: Right.

Leo: Chrome does that. Safari I'm sure puts it in macOS.

Steve: Right. Yeah, I think that the only one who still maintains their own root store is Mozilla.

Leo: Mozilla.

Steve: Because they had NSS, the Netscape Security Suite, and that was where all of their SSL and TLS stuff is contained.

Leo: This is such an obviously horrific idea.

Steve: Yeah.

Leo: I mean, oh, yeah, well, you can trust every - every country in the EU is great. So obviously this would never be a problem. Unbelievable. My hair is on fire.

Steve: I know.

Leo: I am stunned. Tomorrow. This could happen as soon as tomorrow.

Steve: Article 45.

Leo: Holy moly. eIDAS is the regulation.

Steve: EIEIO.

Leo: EIEIO. Old McDonald.

Steve: Yow.

Leo: Watch with interest tomorrow. They're meeting in Brussels behind closed doors. So you don't know what, you know. Golly. How stupid. And yet how predictable.

Steve: Yes. The politicians don't understand that this is not theirs to mess with.

Leo: I fear that they do understand, and that they want in. They don't want you to have security. They want security themselves. But I don't think they want us to have security.

Steve: That's right. They want to be able to monitor the conversations of all their citizens.

Leo: Yeah, yeah.

Steve: And they can't right now.

Leo: And insert arbitrary banners whenever the hell they feel like it.

Steve: That's right.

Leo: Election ads? You know, Turkey's in the EU. Erdogan, I could easily see him doing that. Geez. And that's the thing, that certificate is EU-wide. So, you know, you've got to ask our friends in France, do you want Turkey to start inserting banners in your browsing sessions? Let alone see what you're up to.

Steve: I just don't think that anyone will do it. I just - I can't, you know, I just - I don't think, I mean, this has all been curated and carefully managed. It's got some problems, but they're not big. We've followed them for years. This cannot be allowed to happen.

Leo: Yeah. Well, there you go. This is why you listen to Security Now!, to set your hair on fire. We do this show - wow, Steve. Wow. And you know this is not getting the coverage it really should.

Steve: I know. It's because it's sort of just happening quietly in the background.

Leo: Yeah. We'll start yelling about it for sure on all of our shows. Thank you for giving us an update. EFF also has a piece concurring completely with what you say. Wow.

Steve: I bet they do.

Leo: Yeah.

Steve: This would just cause them to, you know, lose their lunch.

Leo: The problem is that, you know, if you go to the EFF front page, there's plenty of other things to get upset about, too.

Steve: Yeah.

Leo: They say Article 45 will roll back web security by 12 years. Oh, worse than that, really. Have a great week, Steve. We'll see you next time on Security Now!.

Steve: Okay, buddy. Bye.