SECURITY NOW!

Transcript of Episode #791

Chrome's Root Program

Description: This week we examine a serious newly revealed Windows zero-day flaw, a public service reminder from Microsoft, Google's newly announced plan to get into the VPN service business, CERT's unappealing plan for automatic vulnerability naming, and a real mess that WordPress just made of an incremental security update to 455 million sites. Then we'll close a loop, I'll update about SpinRite, and we'll finish by examining Google's new plan to go their own way with a new Chromium browser certificate Root Store.

High quality (64 kbps) mp3 audio file URL: http://media.GRC.com/sn/SN-791.mp3
Quarter size (16 kbps) mp3 audio file URL: http://media.GRC.com/sn/sn-791-lq.mp3

SHOW TEASE: It's time for Security Now!. Steve Gibson is here. Coming up in just a little bit a new Windows zero-day that's been around since Windows 7. A screw-up in patching for WordPress, you're going to want to know about that. And Steve's Dumb Idea of the Week. All I can say is stay tuned. Security Now! is coming up next.

Leo Laporte: This is Security Now! with Steve Gibson, Episode 791, recorded Tuesday, November 3rd, 2020: Google's Root Program.

It's time for Security Now!, and I mean right now, with this guy right here, Steve. Well, there is an exclamation mark in the title. That means it's definitely Security Now!.

Steve Gibson: Better now than never.

Leo: Now. Steve Gibson is our host. He's the man of the hour, GRC.com. Hello, Steve.

Steve: Yo, Leo, great to be with you for, whoa, this interesting day.

Leo: It's election day in America.

Steve: In the country, yeah.

Leo: We're all just kind of nervously waiting.

Steve: We're just biding our time.

Leo: Biden our time.

Steve: Wait, Biden, what?

Leo: Waiting to find out what happened.

Steve: To see if we're going to get Trumped or not.

Leo: Yeah. So we will find out. But meanwhile we've still got a show to do.

Steve: The show must go on, as they say. We're going to, for this week, Episode 791, we're going to look at sort of an interesting thing. Without knowing it, we covered the first part of this last week, not knowing that there was a short, seven-day embargo on a new, just revealed, critical Windows zero-day flaw which was, as I said, embargoed, so we didn't know about it until later last week.

Leo: We knew about the patch; right? We knew they were patching it. We just didn't know...

Steve: No.

Leo: No.

Steve: There was Google who fixed a critical problem in Chrome. But it turns out this was a chained exploit, and Chrome was just providing the front door. But once you got in you needed to elevate your privilege. And it turns out there's a way.

Leo: Now we know how.

Steve: No one knew how to do that before.

Leo: Oh, nice.

Steve: Yeah. We've also got a, speaking of Microsoft, a public service reminder from Microsoft. They're trying to be as responsible as they can after the fact. We've also got an interesting plan. People are going to be skeptical, but we'll talk about it. I'm kind of impressed with at least what they're saying, and that's Google's newly announced plan to get into the VPN service business.

We also have CERT, who has announced an unappealing plan for automatic vulnerability naming. So no more Honey Monkeys, which I think is going to be a mistake. But we'll talk about that.

We've also had a real mess with WordPress's attempt to update - and I didn't realize this, Leo. They have 455 million sites.

Leo: Oh, yeah.

Steve: That's a lot of sites.

Leo: Yeah. They're rapidly closing on half the Internet.

Steve: Boy. Then we're going to close a loop with one of our listeners, who just sort of provided an interesting bit of feedback from the field, from stuff we've been talking about. I'll give a brief update on SpinRite because it's where I'm spending all of my time now, as our listeners know.

And then we're going to finish by examining another Google thing. Google's decided to create their own CA browser root program for the first time ever. The only one who has their own, really as a consequence of the legacy, is Mozilla, with Firefox. But I think Google's probably always been a little envious of the control that that gives Firefox and Mozilla. They've always wanted their own. Well, that's going to happen now. So some interesting stuff to talk about for the week.

And, oh, I found this thanks to one of our listeners who tweeted this photo to me for the Picture of the Week. It's a classic. It's right up there with the gate in the middle of the path, with the well-trodden dirt paths that are just going around the gate.

Leo: This is similar.

Steve: This is like right up there. And notice this is a seriously - this is a big, thick cable on this bike lock. You are not going to cut through that cable.

Leo: You don't need to.

Steve: So you just might as well give up; right?

Leo: Right.

Steve: Actually, there was one - remember the one we had where some bolt cutters were being protected by a cable in a hardware store so you couldn't take the bolt cutters away. It's like, okay, wait a minute, they're bolt cutters, guys.

Leo: Oh, it's a suggestion, that's all. Just a suggestion.

Steve: That's right.

Leo: Sometimes security is just a suggestion.

Steve: Sometimes it's just theater.

Leo: Yeah, a lot of times. All right. We're ready. Picture of the Week time.

Steve: Okay. So for our listeners who are, well, listening, we have a series of yellow steel concrete-filled poles, you know, the kind that would block at the end of a driveway or a roadway where you didn't want cars to be able to drive through. And someone has chained or cabled their bicycle up to one of these. The only problem is it's a pole. And it's, well, maybe like a foot taller than the bicycle. And you wouldn't even really have to lift the bike very high in order to pull the cable off of the top of the pole and then ride away with this bicycle. So this is more of a - I guess this actually is sort of like that path photo that we like so much, Leo. It displays a request. It displays an intent.

Leo: Yes, it's a suggestion.

Steve: It has no actual ability to - it's not enforcing the desire.

Leo: Merely suggesting it.

Steve: It's just sort of saying, you know, this locked gate on this path, even though you could go around it, clearly we're trying to say please don't. Similarly, this bike cabled to a pole, which is completely ineffective, is saying, look, this is the honor system. I clearly am saying I've not left this bike out here just for anyone to walk away with. I've cabled it to a pole, which doesn't stop you from taking it. But clearly my intention is that you not.

Leo: Yeah, yeah, please don't, yeah.

Steve: So it's the honor system of security.

Leo: You could. You could, but don't.

Steve: Yeah. I don't think that'll work with Russian cyberattackers. But we could hope. Okay. As I said, we have a new zero-day, and it's complicated by the fact that it has existed at least since Windows 10. Which, as we know, Microsoft has...

Leo: At least since Windows 7.

Steve: Oh, I'm sorry. Yes.

Leo: 7, yes.

Steve: At least since Windows 7.

Leo: Wow.

Steve: So last week we talked at some length about the bug Google found in the FreeType library, which had been in use since June 19th, 2015, so more than five years. What we knew then was that this flaw, which was patched by that update to Chrome, was a zero-day because it was being actively exploited. They fixed it, when they were notified, they fixed it within 24 hours, which was impressive response. And they notified the FreeType people, who also fixed it within 24 hours. Also impressive.

What we did not learn until the end of last week was that there was a previously secret second part to this zero-day. The FreeType flaw was what was being exploited through Chrome to open the door to the attacker. But as is often the case, thanks to modern operating system design, the damage that can be done by an aberrant application or exploit of an aberrant application running under the non-root user account is deliberately minimal in modern operating systems. All of today's web browsers are careful to run under the user's deliberately limited account privileges.

This is why successful attacks and attackers often need to chain two or more exploits together to accomplish their nefarious ends. If the FreeType Library happened to run in the kernel, then a single exploit in it might have been sufficient. But FreeType was also properly designed to run in user space. So exploiting the FreeType flaw opened the door, but the attacker then needed to elevate their privilege on the system to root or kernel level in order to get anything useful, like from the attacker's standpoint, done.

The week before all of this, Google had seen the whole picture, or presumably the person who informed them had. They saw this second phase, which was leveraging a previously completely unknown and quite potent zero-day flaw in Windows to achieve privilege elevation. This was allowing the attackers then to do some real damage. The privilege elevation that they discovered by watching it in action existed, or actually I should say "exists" because it still does today, within the Windows kernel-based, thus we have a problem there, Cryptographic Services API. And because that kernel-based module, the Cryptographic Services Module, exports an API that's callable from userland, the bad guys can arrange to run their malicious code with full system permissions.

Google's Project Zero folks immediately reached out to Microsoft to inform them of what they had found and also to explain that, since this was an active vulnerability being exploited in the wild, Project Zero's normal, patient, 90-day disclosure window would be reduced, as they even did for themselves, to just one week. Actually, they only needed a day. And that's why the industry subsequently learned of this only late last week. That was seven days, actually eight, after Google told Microsoft.

So Google's Project Zero Day disclosure starts off with saying: "Note: We have evidence that the following bug is being used in the wild. Therefore, this bug is subject to a seven-day disclosure deadline." And we've seen these in the past when we've talked about this and looked at these. In the period before the disclosure deadline, all there is is just like a placeholder page, no juicy details available because they're holding that embargoed until the problem can get fixed.

They begin their write-up by explaining: "The Windows Kernel Cryptography Driver (cng.sys) exposes a \Device\CNG device to user-mode programs" - so in other words, the cryptography driver looks like a device which exposes services through a device driver interface to programs running on top of the operating system - "and it supports a variety of [what Windows calls] IOCTLs, (IO control calls)," they said, "with non-trivial input structures. It constitutes a locally accessible attack surface that can be exploited for privilege escalation," and they said, "such as sandbox escape."

So of course they're viewing it from the standpoint of a sandbox escape because the way this would have gotten in was through the browser. And we know that code running in the browser is deliberately sandboxed so that, if it does something wrong, it doesn't have access to much. But by taking advantage of this, that code is able to escape from Google's own Chrome sandbox.

So Microsoft, for their part, doesn't see this as such an emergency. Google has already closed and locked the front door through which attackers were able to reach the crypto API vulnerability. And November's Patch Tuesday being next Tuesday, a week from this podcast's date of the 3rd, which will be November 10th, expects to have this fixed. So we now have a detailed description of the flaw, and the Project Zero guys even published proof-of-concept code which can be used as a demonstration of this to crash any Windows system. It doesn't give you remote code execution. They weren't going to go develop that to help the bad guys. But this is the Project Zero deal is we'll give you a length of time to fix it, and then we're going to disclose it, which they have. So it's a little unfortunate that Microsoft doesn't see this as a big deal. And maybe it's just a function of one week from now before they'll have this thing patched.

But the other problem is that this is also known to afflict Windows 7 machines, which will not, as we know, ever receive a patch for this unless they're within an enterprise that is now purchasing security for their older machines from Microsoft. The rest of Windows 7 users are left to fend for ourselves. But to that end, although it hasn't happened yet, we can hope and expect that the micropatch folks at 0patch.com have been watching, and that they will again be able to offer one of their cute little patchlets for this, as soon as Microsoft offers the patch. They typically go in, they reverse engineer it, they cutify it, because it ends up being like 28 bytes or 12 bytes or something ridiculous, and then they'll offer it to systems that are not otherwise being patched by Microsoft.

Sometimes they do this for free, but that's normally only until Microsoft has had a chance to respond in any way that they're going to, at which point then it's only their subscribers who are able to continue to receive this. So this is an inexpensive way. We've talked about these guys several times, for people who aren't comfortable continuing to run Windows 7 without any OS-level patching. Our browsers are all being patched. Chrome was immediately patched to close the front door to this. But for what it's worth, now we know there is this other privilege elevation flaw which anything that can find some other way in could then arrange to execute itself with kernel privileges on Windows 7 machines that aren't being otherwise patched.

So if you want to buy yourself some additional security, then I think these micropatch guys are worth looking at. And I know, Leo, it makes you feel a little queasy to have some third party patching Windows. But they're really very transparent. They publish the source code for these things often and just sort of provide it as an inexpensive service. So I think better than nothing, depending upon how you're using Windows 7.

Which brings us back to Zerologon - oh, boy - and what I called a "public service reminder" from Microsoft. Last Thursday, continuing to feel the need to respond to this ongoing exploitation of this very serious Zerologon vulnerability - remember that two weeks ago in our podcast "Anatomy of a Ryuk Attack" we saw the Zerologon vulnerability being used by Ryuk or Ryuk or however you pronounce it, you know, for the delivery and exploitation of a ransomware attack on an enterprise. So it's very real.

Anyway, they tweeted on the 29th: "Reminder to all our Windows customers to deploy at least the August 2020 update or later and follow the original published guidance to fully resolve the vulnerability." And then we have the CVE-2020-1472, which will probably live in infamy, although it's much easier to know of it as Zerologon. We'll be talking about CVE numbers here in a little bit. So anyway, for what it's worth, anybody who's listening to this podcast I'm sure long ago themselves patched. But, boy, there is still an inventory of systems out there that aren't, and apparently aren't ever going to.

And this just, you know, Zerologon is the dream horizontal movement through an enterprise's network once some unwitting user clicks on a link and allows something to run in their email client. This thing is like every bad guy's dream about how to then take hold of an enterprise because enterprises tend to have domain controllers, and this thing allows you to bypass authentication. So, boy.

Okay. Google One has announced - well, Google in the guise of Google One - adding a VPN, bringing up a VPN service, initially for Android. They've just announced they're going to be getting into the VPN business. That is, Google has. They'll be adding a no-additional-charge VPN facility, first for Android users only. Scuttlebutt is that's just the tip of the iceberg, though. They're adding this to their existing paid Google One service. And as I understand it, I don't pay Google, but for free you get 15GB of Google Drive and Gmail and, you know, sort of cloud stuff. And if that's not enough, then for $10 a month or $100 a year you can expand that to 2TB.

Leo: Yeah. It's basically what Google Drive used to be. They call it Google One now, with some additional features.

Steve: Okay. Got it. So because Google is Google, the announcement of them doing a VPN has been met with some well-deserved eye-rolling. Actually, ProtonVPN just went off. We're going to come back to this because of some interesting technology that they will be incorporating into this, some cool stuff we've never talked about before, and maybe I'll talk about other VPN providers. But ProtonVPN was, I mean, Google's a big competitor; right?

Leo: That's the issue; you know? I mean, you could probably say, well, do you want the biggest advertising company in the world to be running your VPN?

Steve: Exactly.

Leo: But Google's been always pretty good about not, you know - I don't know. I don't know how I feel about it.

Steve: Well, so that's why we're here talking about this. So they are saying all the right things, Google is. And they are bringing, as I mentioned, some new technology to bear which has never been done on a VPN, which Google can afford to do, due to their specific posture.

So first of all, to sort of set this, here's how they, Google, describe the problem and the way they're intending to solve it. They said - and actually some interesting stats I hadn't known. I didn't realize VPNs were this popular. They said: "Demand for VPNs is growing, with evidence that it's becoming more mainstream."

Leo: Yeah, I think so.

Steve: "Up to 25%" - huh?

Leo: Yeah, I think that's what we're experiencing, too, yeah.

Steve: Wow, "25 percent of all Internet users accessed a VPN within just the last month of," they said, "of 2019." So that was even pre-COVID. They said: "Unfortunately, not all VPN providers have been proven to be trustworthy." Okay, this is Google. But still, as you said, Leo, I'm glad you did, Google, well, of course they've now got the DOJ breathing down their necks.

Leo: I guess I would read the privacy statement carefully. But they're not going to lie in the privacy statement.

Steve: Well, no. Right. So they said: "Some services are vulnerable; others request unnecessary access to their users' data [huh] or monetize the same data [again huh] that users are utilizing the VPN to keep private and secure; while others fail to deliver on the promise of not logging their users' online activity." So again, Google is going to be a non-logger.

They said: "With growing demand for better privacy in a mixed landscape of solutions, we have used our expertise in privacy, cryptography, and infrastructure to build a Google-grade VPN that provides additional security and privacy to online connectivity without undue performance sacrifices. With VPN by Google One, users' online activity is not identifiable to the VPN and not logged by the VPN. We believe a VPN must be transparent and robust. That's why we've open sourced our client and will provide a third-party audit of the end-to-end solution to make them externally verifiable.

"Privacy is at the core of the products and services we build." Okay. "With VPN by Google One, we will never use the VPN connection to track, log, or sell your online activity. Some minimum logging is required to ensure quality of service." And I've seen online people attacking Google about what that minimum is. Those are specious attacks. I mean, Google is, as you said, Leo, I mean, Google's going to do what they're going to do, and they're not going to breach their own privacy guarantees. And we could argue they don't need to, but we'll get there in a second.

Leo: They don't need to, exactly.

Steve: Right. "But your network traffic or IP associated with the VPN is never logged. To demonstrate how our design works, we've open sourced the code that runs on a user's device."

Leo: Perfect. Perfect.

Steve: "And in the coming months we will be open sourcing the server-side user authentication mechanism, as well as providing the results of a third-party audit, currently underway. These will provide further assurances of how user data is handled and how robust the VPN's security is." And wrapping this up, here's the cool bit.

"Open sourcing our VPN and providing an audit are just some of the steps we're taking to ensure user privacy. While building VPN by Google One" - which is, you know, the formal name - "we realized it was important to strengthen some of the systems that are often attacked or compromised in order to access users' personal data. Traditional VPNs can sometimes compromise a user's identity or online activity by linking the usage of their service to the activity they conduct by means of a session ID. This ID could allow VPN operators, or attackers that compromise their infrastructure, to eavesdrop and identify users and their activity.

"We wanted to eliminate that vulnerability by separating the authentication of the subscriber from their use of the service. By employing a cryptographic blinding step between user subscription validation and connecting to the VPN, we give users a stronger guarantee that their online activity won't be tied back to their identity." That's what's new. So the technique is known as RSA Blind Signing, which Google will be using, and it's a real thing. It's actually old, like 1998 the concept was first developed. It provides a means for Google to verify that a Google One paying account holder has the right to use their VPN service, yet without revealing who that account holder is.

We've never discussed blind signing technology, but I think it would make a terrific topic. So we will. And although, as I mentioned before, Google hasn't yet said so publicly, the word on the street is that this service for Android will eventually be offered widely across Google properties and, you know, like other clients for other desktops and so forth. So I love the idea that their VPN itself is blinded to its user's identity as an account holder. And frankly, I think that Google probably had no choice but to do that if they wanted to provide a VPN service since they're so strong now. As we mentioned, the DOJ is breathing down their neck.

But we also all know that a user's browser's Google-tagged cookies, the instant they emerge from Google's VPN endpoint, will immediately scoot over to the nearest Google server to report in. I mean, it's going to be like a zero hop. I mean, it's going to be in the same building; right? It's just going to go, bing, now I'm at the Google server. So it's not as though the VPN provides any additional anonymity compared with any other VPN service. But, and here's the key, I think, Google's use of cryptographic account authentication blinding means that at least it doesn't provide any less anonymity than the use of any other VPN service. And there's a lot more to be said about this. We'll be covering it in much more detail in the future.

So anyway, and they make a strong point. They've got a strong infrastructure. They're doing all kinds of cool stuff by looking at using DNS queries to figure out where the user is, and then automatically tie the user to the nearest VPN endpoint. I mean, they're Google. They understand infrastructure. They've got a big infrastructure. The promise is that initially any Android user who's also a Google One subscriber would be able to get the arguably useful benefits of running a VPN from their handset past the local WiFi hotspot, wherever they are, past that hotspot's ISP, and directly to Google, where their traffic would then emerge onto the Internet. And yes, where all the Google cookies that they're already carrying would then instantly be gobbled up by Google. But that's going to happen no matter whose VPN service you use.

Leo: Yeah, yeah. I love how you've personified them. They're scrambling over. The cookies are, "Oh, my home, my home." Did they say what they're going to charge for that, just out of curiosity?

Steve: Free.

Leo: So if you have a Google One account, which isn't free.

Steve: Right.

Leo: But if you have that, you will get it. Oh, that's interesting. Yeah, I can see why Proton's a little worried. Yeah, that's a little scary. And yet I think that there will be plenty of people who will say, yeah, whatever. I'm going to use a VPN from a company that's not Google. Not in the advertising business. I can understand that, too.

Steve: So this introduces a new section for the podcast, Leo, the Dumb Idea of the Week. We have a proposal from CERT, of all people, or all organizations. I think that giving catchy, sometimes humorous and descriptive names, often with matching graphic images, to security flaws is part of the fun of this industry. We've had Spectre with that little spooky ghost with the stick hands, Meltdown, Dirty Cow, Zerologon, who can forget Heartbleed. We've got BlueKeep, BLESA, SIGRed, BLURtooth we recently did, DejaBlue and, you know, Stagefright was wonderful.

Leo: That's a long history. The very first virus. Well, the Morris Worm, I don't know if that's a name. But the very first widespread virus, the Melissa Virus, was given the name Melissa because that was in the code; right?

Steve: Yup. Of course.

Leo: And we remember it. I wouldn't remember CVE-1992-1973.

Steve: Exactly. And I'm always a little self-conscious when I'm spewing those off. They sound like stardates.

Leo: They do. They really do.

Steve: It's like, what?

Leo: Do they want to use the CVE numbering? What do they want to do?

Steve: No, no, no, no. Worse. Well, yeah, it is worse. So we've established that naming is important. Not everybody agrees. In a blog posting Friday, the original CERT, that's the one operating out of Carnegie Mellon University, which now collaborates and partners with the DHS's official US-CERT team, has proposed spoiling this particular bit of fun. So the first thing I noted was that CERT's own blog posting was titled "Vulnonym."

Leo: They made up a name.

Steve: Exactly. They said: "Vulnonym."

Leo: Vulnonym.

Steve: "Vulnonym: Stop the Naming Madness."

Leo: Oh.

Steve: But of course Vulnonym is itself a fun and memorable name. They didn't give themselves some serial number. So what they have is an auto-vulnerability-naming service which they're proposing. So here's what they said. They said: "Spectre, Meltdown, Dirty Cow, Heartbleed. All of these are vulnerabilities that were named by humans." Yeah, huh? Anyway, sometimes, yeah, because they're good names.

Leo: Shocking. Named by humans? How can we allow this? I understand they don't want marketing, like they feel like maybe it's too, like, market-y.

Steve: Well, exactly.

Leo: But they've got to be memorable. We want people to remember the name so they know what they're worried about.

Steve: Right. They said they were named by humans, "sometimes for maximum impact factor or marketing." They said: "Consequently, not every named vulnerability is a severe vulnerability..."

Leo: Ah, that's reasonable, yeah.

Steve: "...despite what some researchers want you to think." Okay, yeah, but Heartbleed was not good. It was a great name. And, you know, we all remember it. Anyway, they said: "Sensational names" - and believe me, they really solved this problem - "are often the tool of the discoverers to create more visibility for their work." I think these guys are just sour grapes. Anyway.

Leo: They're just cranky.

Steve: "This is an area of concern for CERT as we attempt to reduce any fear, uncertainty, and doubt" - also known as FUD - "for vendors, researchers, and the general public." Now, okay. I would argue that the general public has relatively low exposure to these names. No one knows what Meltdown is...

Leo: No, that's true.

Steve: ...out in the public. It sounds like something that once happened in Chernobyl, or when you leave your ice cream cone unattended. But no one's thinking, oh, yeah, that's a vulnerability I need to worry about. Anyway, they said, CERT said: "Software vulnerabilities are currently catalogued by number, primarily the Common Vulnerabilities and Exposures (CVE) ID, which makes it very easy for computer analysis and storage. However, humans aren't well conditioned to remember numbers. Instead, humans prefer names because we find them easier to remember." So really they're making our case, Leo.

And they said: "We don't remember IP addresses, but do easily remember domain names to browse our favorite websites. We also remember things like hurricanes, snow storms, operating system updates, particular geographic locations like cities or states, and so forth. They are all named because it's easier to remember 'Mojave' instead of macOS 10.4, or 'Pittsburgh' instead of 40.4406 N by 79.9959 W." Okay, CERT, you've made your point.

They said: "Names of vulnerabilities in particular are matriculating into important spheres of influence. Case in point, on July 11th, 2018, congressional testimony weighed the impacts of the 'Meltdown' and 'Spectre' vulnerabilities." Of course those we know; right? "The CVE IDs, CVE-2017-5753, CVE-2017-5715, and CVE-2017-5754 were never mentioned. Only the sensational names were." Okay, no. Only the recognizable names were. There's nothing sensational about "Meltdown" and "Spectre." I mean, they're just not dry. Anyway, they said: "We aren't arguing that vulnerabilities shouldn't have names. In fact, we're encouraging this process!"

Leo: Okay, well, that's a matter of dispute. But go on.

Steve: Yeah. Just wait, wait.

Leo: Wait'll you hear the names.

Steve: "Our goal is to create neutral names that provide a means for people to remember vulnerabilities without implying how scary or not scary the particular vulnerability in question is. Our neutral names are generated from the CVE IDs to provide a nice mapping between name and number. CERT decided that if we can come up with a solution to this problem" - spoiler alert, they haven't - "we can help with discussions about vulnerabilities as well as mitigate the fear that can be spread by a vulnerability with a scary name." They said: "We plan to name the vulnerabilities with a phrase of adjective/noun, for example, Arbitrary Albatross." And I added here in the show notes, yeah, or how about Stupid Idea?

"When tackling this problem, we considered several lists of words to ensure no sensational" - oh, boy, did they achieve that - "scary, or offensive names were included. We created the list of both adjective and nouns using the combined resources of the Wiktionary" - whatever that is, I didn't even bother to look it up.

Leo: Wiktionary. Yeah, yeah, yeah, yeah. It's a wiki dictionary, yeah.

Steve: Okay, good, Wiktionary, "...and categories of words such as animals, plants, objects in space, and more. Next, we created the method by which we map the CVE-IDs to the pair of adjective names." Okay. "After much consideration, we used the Cantor

Depairing Function, which is a bijection between the natural numbers and a pair of natural numbers." Well, obviously. How else would you do it?

Leo: Of course. How else would you?

Steve: Of course. Duh. "This means that each natural number can be mapped to two natural numbers uniquely." Because we wouldn't want any collisions of these fabulous new adjective noun pairings they've come up with. "To test out this idea, we're operating @vulnonym" - so everybody wants to follow this. This is going to be gripping. This is a Twitter feed, @vulnonym on Twitter. To publish the neutral names associated...

Leo: This is an Onion article. This can't be serious.

Steve: No, this is the truth.

Leo: I can't believe it.

Steve: This actually happened.

Leo: Oh, god.

Steve: And it's not even April 1st, "...associated with CVE IDs as they are issued. Follow @vulnonym and let us know if this naming experiment is useful! And in case anyone considers a word or name to be offensive, we have a simple process to remove it from the corpus and regenerate a name."

Okay. So I digested this. I checked out @vulnonym. I'm not sorry that I brought all this up, since we may start seeing some very poorly named vulnerabilities, and we should understand what happened there.

Leo: Oh, boy.

Steve: Okay. So here are three tweets recently made by the @vulnonym tweet bot. The first one, CVE-2020-4785 is called Whacking Mouflon.

Leo: What? What's a Mouflon?

Steve: A Whacking Mouflon.

Leo: Whacking I get, but why is the Mouflon whacking, and who is a Mouflon?

Steve: We do not know. But actually, Leo, this next one, the next tweet? "Hi, I'm CVE-2020-4649. I was never good with numbers, though, so you can call me Unmatched Cwm."

Leo: What?

Steve: Yes. Spelled C-W-M.

Leo: No one knows how to pronounce that, even.

Steve: There's no vowel. There's no vowel.

Leo: It's in Welsh.

Steve: Yes, it is. I looked it up. And if you were in Wales, you might know that it means a steep-sided hollow at the head of a valley.

Leo: Yes, I did know that, actually.

Steve: Or on a mountainside.

Leo: But it don't think you want to use it in common speech.

Steve: Well, no. Have you been hit by the Unmatched Cwm?

Leo: So are these real CVEs that they're tweeting?

Steve: Yes, yes.

Leo: 2014-1060 is Cyclic Hyrax.

Steve: Yes. Now, isn't that catchy, Leo? Wouldn't you like to have the Cyclic Hyrax vulnerability?

Leo: This is ridiculous. They just need better dictionaries. I mean, if they had better dictionaries, maybe.

Steve: Then they could be fun.

Leo: They could do what the NSA's doing; you know?

Steve: It's clearly their intention - I don't know who these people are. They don't want anyone to have any fun. We could not have Honey Monkeys if these guys were in charge.

Leo: CVE-2020-16006 is Privileged Ukulele. That one's at least kind of memorable.

Steve: Yeah, okay.

Leo: They should have adjective/noun, and make the nouns a little more common.

Steve: Yeah, not Cwm. What is a Cwm?

Leo: What's a Cwm? If you're Welsh...

Steve: Or a Mouflon.

Leo: I think a Mouflon sounds like an animal, yeah.

Steve: It's whacking, whatever it is. The Mouflon is whacking. I don't know.

Leo: Caring Doeg, D-O-E-G. I'm sorry, Uncaring Doeg, that's CVE-2020...

Steve: Oh, it's really Uncaring? Oh, my god.

Leo: Uncaring D-O-E-G. Equable Jawfish.

Steve: Maybe that goes with the Unmatched Cwm. It's uncaring.

Leo: These are the worst. Lunar Termite. Brisk Squirt. Oh, you don't want Brisk Squirt. I don't know what it is, but you don't want it. Unvarnished, what is a Sarrusophone? What the hell is a sarrusophone?

Steve: You want to report that one to your GP.

Leo: Doctor, I've got a Brisk Squirt. I need a shot of some kind for that. Orchestral Waterphone. Transverse Vison. Voiced Adder. Some of these are okay. If it had less uncommon words, it wouldn't be that bad.

Steve: You know, so maybe the problem is the bot is unmanaged. They need a managed bot.

Leo: Or a better dictionary.

Steve: Somebody needs to look at these.

Leo: What's a Sher? Rakish Sher? What's S-H-E-R, Sher?

Steve: Well, apparently that's in Wiktionary.

Leo: That's a mistake. It's too - they need a better word list, I think.

Steve: Yeah.

Leo: Although, if you're going to be a vulnerability, I hope somebody comes along and names a vulnerability Brisk Squirt, just so we get to keep that one. That one I would keep. Oh, my god. "Did you get the patch for Brisk Squirt yet?" "No, man. I'm suffering, dude. Man, am I suffering." Oh, my god. I'm looking: Uninvolved Dulcimer. That's not bad, an Uninvolved Dulcimer.

Steve: That's not bad. We could keep that one.

Leo: Prominent Caterpillar.

Steve: If they have some personality, yeah.

Leo: Sulfa Bonefish. I think that's going to be my pseudonym from now on.

Steve: Okay, wait. Sulfa is an adjective?

Leo: Yeah, no, I don't think so. So I think it's just - they're just taking random words. It needs to be adjective/noun. It needs to be common nouns, or at least a little more common than Cwm. No Welsh allowed. You've got to have at least one vowel per name. Legless Umber. Pensive Snakehead. Grouchy Camelopardalis. You don't expect security researchers to say that out loud.

Steve: Well, and what is the general public going to do with this? I mean, they seem to be worried, you know, now compare this to Spectre and Meltdown. This is like...

Leo: And what you don't want is a chained vulnerability with Brisk Squirt and Roaring Swallower. Then you've got a problem. Imagine the chained vulnerabilities.

Steve: You know, Leo, Brisk Squirt is what you use to open the front door; right?

Leo: And then you use Roaring Swallower to do the privileged escalation. Oh. Oh, my god. What were they thinking? I'm following this Twitter bot though because...

Steve: Clearly.

Leo: I'm going to get a laugh every day from this. Holy cow.

Steve: Is that one of them? CVE-2020 Holy Cow?

Leo: And, see, that's another problem. What are they going to do? They're going to have a special elimination of actual things that make sense?

Steve: Wow. I guess Brisk Squirt might be more the backdoor than the...

Leo: I like Brisk Squirt and Roaring Swallower. I don't know. Definitely there's a chained vulnerability of some kind.

Steve: It's really going to be dangerous.

Leo: Retrospective Gerbil? No. No, no, no. I'm sorry. There's just too much weirdness in this. It's just not...

Steve: And it isn't April 1st or like something from the Onion. It really actually is CERT.

Leo: You don't want this one. Undressed Mephitis. It sounds like a venereal disease. Wait a minute. M-E-P-H-I-T-I-S. Mephitis. Oh, wow. It's a noxious or foul-smelling gas or vapor. You don't want, you definitely don't want Undressed Mephitis.

Steve: Ugh. I don't think you want Mephitis with or without your clothes. Ugh.

Leo: Why would they allow "mephitis" to be in the dictionary? You don't need that word. Natty Gazelle, that's good.

Steve: It was supposed to be memorable.

Leo: Yeah, no.

Steve: There's nothing memorable about Mephitis.

Leo: Bearish Bushbuck. Driverless Major. Purulent Bandfish. Unpainted Oto. Okay.

Steve: I mean, it is a little bit like Mad Libs, Mad Libs for security vulnerabilities.

Leo: The Twitter feed, V-U-L-N-O-N-Y-M, vulnonym. I just - I think some undergrad wrote this paper. I do. Oh, wow. All right.

Steve: Okay.

Leo: Too late to name the show Squirting Whatever. Probably too late, yeah.

Steve: Oh, boy.

Leo: Steve. That's the howl of the day. Wow. Wow.

Steve: Okay. Stop reading it, Leo.

Leo: I can't. I'm looking away. I'm looking away. Do not look at the vulnonym.

Steve: Back on Earth, we have WordPress, who has fumbled an important update, you know, just after I was telling everybody last week that they should just be updating everybody's WordPress installations because bad problems need to get fixed. Last Friday they patched 10 security bugs as part of their release 5.5.2. The most severe problem patched would have allowed a remote, unauthenticated attacker to take over a targeted website through what they described as a narrowly tailored denial of service attack, whatever that means. Actually I figured out what it was here in a minute.

WordPress wrote that: "The vulnerability allows a remote attacker to compromise the affected website. The vulnerability exists due to improper management of internal resources within the application, which can turn a denial of service attack into a remote code execution issue."

Okay. The researcher who found the bug described it as interesting, but likely difficult to carry out in the wild. He said: "You have to be able to produce a very accurate DoS attack." Okay, whatever that - an "accurate" DoS attack? He said: "The principle is to trigger a denial of service on MySQL so that WordPress will think that it's not installed, and then un-DoS on the DB" - the MySQL database - "under the same execution thread." Okay.

Anyway, v5.5.2 also brought a bunch of feature enhancements in addition to the 10 vulnerability fixes, one of them being really - that one being really important. WordPress described the updates as a short-cycle security and maintenance release, to fill in before the next major release, which will be 5.6. And with the update, all versions since WordPress 3.7 will also be current. So that was how WordPress hoped things would go with the update that they were beginning to push out to 455 million sites. Boy, Leo, with that kind of install base they really have to be careful.

Leo: Yeah.

Steve: I mean, they really have to be careful. And they weren't.

Leo: Oh.

Steve: It was soon discovered that this 5.5.2 was badly broken. It turned out that 5.5.2 was causing new WordPress installs based upon that release to fail. And so anybody who was attempting to do a new WordPress install of 5.5.2, it would fail. As soon as they became aware of the mistake, they put the brakes on its rollout. But they screwed that up, too, by inadvertently triggering the release of an unreleased and not debugged alpha version of WordPress, which they started downloading to customers.

So the first thing that happened was that WordPress site operators began reporting that, because of the 5.5.2, that new WordPress installs were failing. And apparently others were complaining about broken admin login pages, which as another effect of that. So scrambling around, WordPress said: "5.5.2 caused an issue with installing ZIP packages available on WordPress.org for new versions of 5.5.x, 5.4.x, 5.3.x, 5.2.x, and 5.1.x." They said: "The issue only affected fresh WordPress installations without an existing wp-config.php file in place." Thus new because they wouldn't have had that file as opposed to upgrading where that file would already exist.

They further explained: "While work was being done to prepare for WordPress 5.5.3" - which was the fix to the mistake that they made with 2, which was what caused this new installation problem - "the release team attempted to make 5.5.2 unavailable for download on WordPress.org to limit the spread of the issue noted, which only affected new installations. But this action resulted in some installations being updated to a pre-release '5.5.3-alpha.'"

Unfortunately, the alpha release mis-installation brought with it the old default "Twenty" themes and the Akismet plugin as part of the pre-release 5.5.3-alpha. So that messed things up. And admins who had not asked for any pre-release install were suddenly being greeted with the message: "BETA TESTERS: This site is set up to install updates of future beta versions automatically." So they were like, what? What just happened?

So in the wake, you can imagine, of all of this unasked-for auto-update mess, many WordPress admins were vocally worried and upset about the whole "not under their own control" issue. And, you know, it's a problem that we have. We of course were just talking about this last week. I took the position, and I have taken the position, that anything hooked to the Internet needs to have the capability of being updated automatically.

But, you know, in light of all this, and standing back from it a bit and sort of looking at what Microsoft has done, maybe Microsoft's semi-compromise stance with Windows 10 is the best we can do. Notify everyone of updates. Kind of give them some push to update now. But allow that "now" to be deferred for some length of time, to a more convenient time. But ultimately, if they keep pushing it off, force the issue, if necessary, and especially if there's like a really critical issue that needs to get fixed.

And of course the problem is an operating system can notify its users. Normally there's somebody there typing at the screen. But there's no clear way for someone's router or most other IoT devices to notify them. So I don't know. I think we're going to need to have some sort of standards defined and adopted so that we can move this whole dilemma into history. And as we know, my real complaint with Microsoft and Windows 10 is that they've taken the path deciding that they are never going to leave it alone. They're going to keep changing things. And that means because the system is so huge and so complex that nobody understands it anymore, that things are going to break. And it's never going to have a chance to settle down and get the bugs worked out of it. They're going to keep, they do keep introducing new problems all along the way.

So anyway, I just thought it was really interesting that here, you know, I was just talking about WordPress's need to, yes, push - well, remember there was a really bad, like a flaw in an update which they proactively patched. I would always agree they have to do that. But boy, Leo, they've got 455 million installations, they just have to be really careful. I wouldn't want that responsibility.

Leo: And it's written in PHP, which doesn't help.

Steve: No.

Leo: Right?

Steve: No, you're right. Personal Home Page.

Leo: They didn't even bother to update the meaning. They could have tried some retronym that made more sense. But it's still Personal Home Page. Come on, Rasmus. Come on, man. On we go with the show, Steve.

Steve: So just one little bit of feedback from a listener, Spencer Salmon.

Leo: Not his real name, by the way.

Steve: Yeah, it does sound like a CVE. He says: "I just finished last week's podcasts, and you mentioned IE and Edge. Interesting fact. I work in the financial industry, and our core provider's web app is only compatible with IE." And then he said in parens, "ActiveX."

Leo: See? That's why people still use it, yup.

Steve: Yup. And, I mean, geez. I guess I would wonder if a provider that was still there and hadn't moved forward was like a going concern. Because, you know, IE is dying.

Leo: Well, ActiveX. Should anybody be using ActiveX? That's so bad.

Steve: Yeah. It was a bad idea.

Leo: I remember you castigating it when they announced it.

Steve: Yeah.

Leo: It basically allows arbitrary code on the web to run on your machine with full privileges.

Steve: It's misbegotten.

Leo: Misbegotten.

Steve: Yeah. Elk. I'm sure that's a CVE. Anyway, last week my time went into overhauling a bunch more of the earlier code on the pre-AHCI IDE/ATA driver code. The way I had originally written that first cut exploratory code back in 2013, it wasn't ever intended to be production ready. But the way this project is evolving, the sooner I can make it production ready, the sooner we'll have SpinRite.

So as I said last week, I've been realizing that I have an opportunity to pre-test pretty much all of what will become SpinRite's new foundation within the context of an interesting and surprisingly, well, interesting and low-level benchmark, which produces some surprising results. So I'm taking that opportunity now, basically investing in SpinRite so that I can get as much of this code tested as soon as possible. So I again expect that I'll have something to announce very soon. But I keep seeing more opportunities to fix things. And so no time like the present.

Okay. Chrome's Root Program. A little less than two weeks ago a new page appeared at Chromium.org titled "Chrome Root Program." And it was not met with universal joy. We'll get to that in a second. Of course this podcast has covered the operation of SSL/TLS web server trust certificates at great length because its proper operation and functioning is crucial to enabling users and their web browsers to establish a trust relationship with otherwise unknown remote web servers. Now we just all take it for granted.

And as we know, the system is far from perfect. It has a myriad of well-known failure modes. It relies upon several different actors, each performing their jobs perfectly, where failure by any of them to do so results in a local breach of the privacy and security guarantees which is the system's entire purpose. But for better or for worse, it's the system we have today. Someday, maybe something will obsolete it. But again.

Okay. So from the beginning, Chrome and most other browsers, "most" with a single exception, have all relied upon the connection security and certificate verification provided by whatever operating system they were running on. The notable exception to this has always been Mozilla and Firefox. To their credit, Netscape, way back then with their Netscape Navigator, invented SSL 1.0, and the concept of using public key cryptography to provide both server authentication and connection content privacy.

Over time, this evolved into NSS, which is Mozilla's Network Security Services. NSS is the SSL/TLS library upon which Firefox runs and which it uses to provide all of its network connection security. Since NSS is cross-platform and a freestanding component of Firefox, it includes its own root certificate store which anchors the validation chain of any certificate received from a web server that Firefox is connecting to. All other web browsers, which inherently have a shallower history than Netscape's Navigator, since it was the first, and Mozilla's Firefox, which is a descendant, rely, as I mentioned, upon the hosting OS's platform for their connection security.

But how many times have I noted that today's modern cryptography is a solved problem? What was once decidedly regarded as highly complex, don't mess with it, magic crypto from some ivory tower guru, is now run-of-the-mill. So the barrier to entry of bringing up a new TLS communications foundation from scratch is as low as it's ever been, which is quite low now.

So against that backdrop, Google's new Chrome Root Program really shouldn't surprise us. Google wants control over this aspect of Chrome's operation, which it has until now delegated to its hosting OS. Chrome was able to operate on the sidelines. We've talked about how this works through the years. Basically it was able to watch the connections, blacklist and pin certificates. But it's never been in the position to directly and completely manage the Root Store which underlies its browser's trust. And you know, knowing Google, that's got to chafe. So they finally decided, okay, we're just going to do this.

But running a Root Store program is also a significant responsibility since who you need to, well, it's a responsibility because you need to be very careful about deciding who you let in, who you don't, who you may need to kick out based on their behavior. Over the years, the podcast has tracked a number of these incidents. And of course these decisions that you make directly affect your customers' security and their ability to get to wherever it is they're trying to go. Remember this torturous decision that the browser and OS vendors had to make when StartCOM was clearly found to be misbehaving and issuing certificates that they should not have been. And it wasn't just that they made a mistake. It's that they weren't forthcoming with it. And that's almost a bigger no-no than making a mistake.

So from its position, which has up until now been on the sidelines, Google as I mentioned was able to sniff the certificate exchange and block certificates that, for example, after the decision was made to stop accepting certificates signed by StartCOM, they couldn't remove the StartCOM cert from the underlying OS's Root Store because it wasn't theirs to manage. So clearly it makes sense, with Chromium going the way it has, that Google is going to just run their own.

So their low-key announcement of their intention to develop and run their own Root Store program, it establishes the importance of the browser's Root Store with this very short description that sort of states their case. They said: "When Chrome presents the connection to a website as secure, Chrome is making a statement to its users about the security properties of that connection. Because of the CA's (Certificate Authority) critical role in upholding those properties, Chrome must ensure the Certificate Authorities who issue certificates are operated in a consistent and trustworthy manner. This is achieved by referring to a list of root certificates from Certificate Authorities that have demonstrated why continued trust in them is justified. This list is referred to as a 'Root Store.' The policies and requirements for participating and being included in a Root Store are known as a Root Program."

So for longstanding, tried-and-true certificate suppliers with a well-known, time-proven track record and impeccable credentials, like my own favorite provider DigiCert, inclusion in Google's, Mozilla's, and any other operating system's Root Store is pretty much a pro forma no-brainer. But the world has a great many certificate authorities of somewhat questionable reputation. And deciding whom to trust can be very political, as we've seen in some of these discussions.

So since the security of the Chromium project's upcoming Root Store is of crucial importance, I wanted to share the inclusion policies and their underlying philosophy. Just sort of it's interesting to get sort of a snapshot into this. Google said: "The explanations below describe the Chrome Root Program, and policies and requirements for CAs to have their certificates included in a default installation of Chrome, as part of the transition to the Chrome Root Store." Because of course it doesn't exist today, and everyone wants to be in it tomorrow.

So they said: "Historically, Chrome has integrated with the Root Store provided by the platform on which it is running. Chrome is in the process of transitioning certificate verification to use a common implementation on all platforms where it's under application control, namely Android, Chrome OS, Linux, Windows, and Mac." And this is interesting. "Apple policies prevent the Chrome Root Store and verifier from being used on Chrome for iOS." Yes, it's too closed. On iOS you have to use their underlying connection architecture.

Leo: WebKit, yeah.

Steve: So, yes, with that comes their Root Store. They said: "This will ensure users have a consistent experience across platforms, that developers have a consistent understanding of Chrome's behavior, and that Chrome will be better able to protect the security and privacy of users' connections to websites." You know, yeah. They just want control. They want to run their own store. They said: "For CAs that already participate in other public Root Programs, such as the Mozilla Root Program, many of these requirements and processes should be familiar."

They said: "During this transition, the Chrome Root Store contains a variety of existing Certification Authorities certificates that have historically worked in Chrome on the majority of supported platforms." So, yeah, naturally they're going to start with the standard set of CAs. They said: "This promotes interoperability on different devices and platforms and minimizes compatibility issues. This should ensure as seamless a transition as possible for users.

"In addition to compatibility considerations, CAs have been selected on the basis of past and current publicly available and verified information, such as that within the Common CA Certificate Database," which is known as a CCADB. "CCADB," they wrote, "is a database run by Mozilla and used by a variety of operating systems, browser vendors, and Certification Authorities to share and disclose information regarding the ownership, historical operation, and audit history of CA certificates and key material." So in other words, they're not just going to grab everything and not verify that it's something that they for their own store agree that they really want to have in theirs.

They said: "For CAs that have not been included as part of this initial Chrome Root Store, questions can be directed to chrome-root-authority-program, with hyphens, @google.com. Priority is given to CAs that are widely trusted on platforms that Chrome supports in order to minimize compatibility issues. For the inclusion of new CA certificates, priority is given to CAs in the following order, in order to minimize disruption or risk to Chrome users." And they've got a list of five, so here's the order.

First, CAs that are widely trusted and which are replacing older certificates with certificates and key material created within the past five years and have an unbroken sequence of annual audits where these certificates and key material are explicitly listed in scope. Two, CAs whose certificates and certificate hierarchy are only used to issue TLS server certificates and do not issue other forms of certificates. Three, CAs that have undergone a widely recognized public disclosure process regarding their CP, CPS, audits, and practices.

So CP is Certificate Policy, which is a formal statement that the Certificate Authority has published, and CPS is Certification Practices Statement, another formal statement. At this time, they said, the only discussion process recognized as acceptable is the discussion process operated by Mozilla on behalf of the open source community at mozilla.dev.security.policy. Okay. So that's a newsgroup. And it is fascinating. If you're curious to see how the sausage is made and, surprisingly, how much sausage there is to be made, you really need to check out mozilla.dev.security.policy. If you just google that, the first link is a link to groups.google.com. It's basically the Google view of this old-school NNTP-style newsgroup, mozilla.dev.security.policy.

It is so easy for us to underappreciate all the hard and really thankless work that goes on, to our immeasurable benefit behind the scenes by real people who we'll never know to thank. When you just look at some of these discussions, it's like, wow, I mean, like there's just so much work that they're doing on our behalf, in order to end up with this heavily curated list of certificates.

Leo: Thank you. Thank you.

Steve: So continue down in priority, four, CAs that maintain sole control over certificate key material within their CA certificate hierarchy, and include their entire certificate hierarchy within a single audit scope. And, finally, CAs that have been annually audited according to both the "WebTrust Principles and Criteria for CAs" and the "WebTrust Principles and Criteria for CAs - SSL Baseline with Network Security." So, yes, this is lots of bureaucracy, but it's bureaucracy we depend upon in order to have the security that we just so casually take for granted.

Leo: Yup.

Steve: So they said: "Certification Authorities who do not meet all of the above criteria will be dealt with on a case-by-case basis. Note that the requirements above are illustrative only; Google includes CAs in its Root Program, and includes or removes CA certificates within its Root Store as it deems appropriate for user safety. The selection and ongoing membership of CAs is done to enhance the security of Chrome and promote interoperability." Obviously, we don't want any certs that we have any reason to believe would endanger our users, yet we want all the certificates that people actually need in order to be able to go where they want to go on the web.

Then they said: "CAs that do not provide a broad service to all browser users are unlikely to be suitable." In other words, you know, if you're some CA that just wants your cert in the Chrome Root Store because, but if you'd have it like signed, like a significant number of active web servers certs, then it's like, no, we're not putting you in. They said: "As this transition occurs, CAs should continue to work with the relevant vendors of operating systems where Chrome is supported to additionally request inclusion within their root certificate programs as appropriate."

Obviously, if everyone has the same basket of certs, that's going to be best for interoperability, and it gives Chrome a better place to launch from. But Google wants, ultimately, Google wants control of their own Root Store. They said: "This will help minimize any disruption or incompatibilities for end users by ensuring that Chrome is able to validate certificates from the CA, regardless of whether it is using the Chrome Root Store or existing platform integrations." So obviously there will have to be some sort of handoff between underlying OS store and Chrome's own.

They said: "The Chrome Root Store Policy will be updated to more fully detail the set of formal ongoing requirements for working with Google in order to be distributed and included in a default installation of Chrome, as well as additional steps for applying or updating existing included certificates. Any questions regarding this policy can be directed to..." blah blah blah. And there's just more of this boilerplate that I'm going to skip because everyone has the idea.

So nothing too surprising here. This feels as though it's a bit of a placeholder, like maybe largely cribbed from other Root Store program guidelines. It does indicate that it will be refined over time. So we could perhaps expect more specific requirements might be evolving. And I mentioned at the top that this announcement was met with a moderate level of grumbling on some fronts. And in doing some digging around, I found admins who are not happy about this.

We end-user consumers just happily click away on links, trusting that all of the plumbing underneath works correctly. But there are those in the enterprise wearing well-worn plumbers' overalls who have identified that the proactive management of certificate Root Stores being a crucial anchor of trust throughout the enterprise can form an important and powerful management firewall. If software needs to be signed by certs that are issued by recognized certificate authorities, and the same is true for websites, keeping a tight rein over the precise content of an enterprise's trusted Root Stores can form another potent line of defense.

And apparently that is being actively done. There are admins who are very tightly controlling the Root Stores of their enterprise's fleet, for example, of Windows machines. It's not just left up to the OS vendor. So in light of Google's announcement that they would be splitting from the Root Stores of Windows, Mac, and Linux to go their own way, basically this grumbling took the form of, oh, that's just great, now we'll have another Root Store to deal with. So these people were not happy. And so, yeah, it'll make their lives a little bit more complex. But on the other hand, it is the way Firefox has always operated. So it doesn't seem like it's going to be a big deal. And it's obvious that Google wants this for their own browser.

One question that wasn't clear to me is whether any given Chromium-based browser that is non-Chrome will be able to choose which store it uses, the Chromium Store that came with the browser, or the underlying OS's. I suppose Microsoft might be willing to run their new and shiny Chromium-based Edge browser within the Chromium Root Store. But it's going to be sitting on top of Microsoft's and Windows' own perfectly good Root Store. So that seems sort of odd. You know? I mean, IE - is IE still in Windows 10? Yeah, right, yeah. Windows 10 still has that.

Leo: Yeah, but they're really moving it out, yeah. It's deprecated now. I don't know how long it'll last.

Steve: Yeah, right.

Leo: They had to leave it in for those people using ActiveX. Oh, god.

Steve: Yes.

Leo: Yes.

Steve: Just say no.

Leo: Just say no.

Steve: So that's the story.

Leo: Okay.

Steve: With Google's Root Store. We'll be moving away from an independent crypto, well, moving to an independent crypto architecture where the browser itself will be doing all of its own connections, rather than relying on the underlying OS.

Leo: Steve, as always, clear as a bell. You're fantastic. This is a show worth listening to every week just so you keep up with what's going on. But I think I learn more every single time. I know you're waiting for me to come up with...

Steve: I was going to say, do we have...

Leo: I know you're waiting for me to come up with yet another code name. But I think I'll just stop. I think Mutinous Genoveva was all I can do, and all I could take. We do this show every Tuesday. It's not always an election day, but it is today. So get out there and vote. Next week we'll be back after an Apple event that will be almost as exciting. We get on at about - and next week might be a little delayed because of the event. I'm not sure. But I think we'll get on around 1:30 or 2:00 p.m. Pacific Time, that's 5:00 o'clock Eastern Time. That's 20:00 UTC.

There's a live stream you can watch or listen to if you like to kind of watch behind the scenes. That's TWiT.tv/live. Audio and video is there. But most people get the on-demand versions of the show. That's why we call it a podcast. Steve's got a couple of unique versions: a 16Kb audio version for the bandwidth impaired, the truly bandwidth impaired. But the even smaller version is the really beautiful transcriptions Elaine Farris writes. Those are both available at GRC.com. He also has standard 64Kb audio. While you're there check out SpinRite, the world's best hard drive maintenance and recovery utility, fast approaching v6.1. But if you get 6.0 now, you'll get 6.1 for free, and you can kind of participate in the early testing of 6.1, so that's probably worthwhile: GRC.com. Lots of other great stuff there.

We have the show, audio and video, at our website, TWiT.tv/sn. You can also watch on YouTube. There's a YouTube channel dedicated to Security Now!: YouTube.com/twit. The main TWiT channel has links to all of the sub show channels. And of course the best way to get it would be subscribe. That way you don't have to think about it. You don't have to worry about it. Just whenever you're in the mood, you say, oh, look, there's a new Security Now!, and you can listen to it on your device.

Steve. I'm going to be hungover tomorrow, no matter what happens. But I'll always be thinking of your Mutinous Gerbil.

Steve: Oh, good.

Leo: I'll keep it in mind at all times. Have a wonderful evening, don't stay up too late, and we'll see you next week on Security Now!.

Steve: Thanks, buddy. Bye.