SECURITY NOW!

Transcript of Episode #706

Open Source eVoting

Description: This week we look back at last week's March Patch Madness. We have an answer about the Win7 SHA-256 Windows Update Update; big news regarding the many attacks leveraging the recently discovered WinRAR vulnerability; what happens when Apple, Google, and GoDaddy all drop a bit; an update on a big recent jump in Mirai Botnet capability; some worrisome news about compromised Counter Strike gaming servers; some welcome privacy enhancements coming in the next Android Q; a pair of very odd web browser extensions for Chrome and Firefox from Microsoft; a bit of follow-up on last week's Spoiler topic; some closing-the-loop feedback from our terrific listeners; and an early look at a VERY exciting and encouraging project to create an entirely open eVoting system.

High quality (64 kbps) mp3 audio file URL: http://media.GRC.com/sn/SN-706.mp3
Quarter size (16 kbps) mp3 audio file URL: http://media.GRC.com/sn/sn-706-lq.mp3

SHOW TEASE: It's time for Security Now!. Steve Gibson is here. An update on the WinRAR virus. It's out in the wild. Be careful. Update your WinRAR. We'll also talk about a funny little pop-up Steve got when he was trying to read up on WinRAR on the McAfee website. And then a proposal from DARPA for a secure eVoting system. This one might really work. Steve's got the deets, coming up next.

Leo Laporte: This is Security Now! with Steve Gibson, Episode 706, recorded Tuesday, March 19th, 2019: Open Source eVoting.

It's time for Security Now!, the show where we cover your privacy and your security and how computers work and all sorts of stuff with this guy right here giving us the Vulcan salute, Mr. Steve Gibson. Hi, Steve.

Steve Gibson: Yeah, so we decided the thumb is out for the Vulcan salute.

Leo: Very important.

Steve: It's not thumb in, it is thumb out.

Leo: It's very important.

Steve: So I look like a crazy person.

Leo: Not to us.

Steve: I've tried to minimize the caffeine, so we'll see how the podcast goes today, Leo. I think maybe a little less nutty.

Leo: Were you hyper last week? I don't even remember.

Steve: Yeah, I over caffeinated, and it was like, whoa. So of the news that we'll talk about this week, one topic stood out because it's something that we've talked about, we've encountered, sort of every summer after Def Con, after the hackers just chew apart the voting machines that are actually in use throughout the United States despite all reason. DARPA, the, well, the source of the Internet because it was the Defense Advanced Research Projects Agency, DARPA, that was behind the original funding of the early experiments to see whether packet switching was something that could actually be made to work. And that, of course, grew into a global network.

Well, they have let a $10 million contract for the development of open source eVoting, thus the title of today's podcast. And what we will be discussing as we wrap up this couple hours is some of the details which are very exciting, because they are known, of what is in the process of being developed now to create an open source eVoting platform which we will then hope states demand that companies like Diebold - oh, wait, Diebold. We decided it was Diebold...

Leo: Diebold, yeah.

Steve: ...then produce to that spec. But we've got a whole bunch to talk about. We have...

Leo: Before you get onto that, I just want to mention last night, yesterday, we did an interview with a guy named Mark Richardson, a photographer. And look at this.

Steve: I saw a picture of the IMP, yes.

Leo: The IMP. And I thought of you when you mentioned DARPA, this was the Interface Message Processor that was the first unit on the DARPANET. This is what was sending messages back and forth.

Steve: Yup. I've had my hands on one of those, actually. There was one at Stanford University's AI lab when I was there in the early '70s. And I remember standing there next to it, looking at it and thinking...

Leo: History, boy.

Steve: ...what the heck is this?

Leo: That was the first node on the Internet. This was actually IMP Number 10, apparently.

Steve: Wow.

Leo: Yeah, isn't that cool? I just thought you'd enjoy that.

Steve: Yeah, very cool.

Leo: Yeah, there's a lot of good pictures in this book. He took them all at the Computer History Museum.

Steve: I love that kind of book. It's like nostalgia.

Leo: Well, you'd recognize that, wouldn't you?

Steve: There it is, baby. That's my PDP-8/E, yes.

Leo: You've got a few of them behind you, as a matter of fact. Nice.

Steve: Very cool.

Leo: What else?

Steve: So we're going to look back at least week's March Patch Madness. We have an answer about that Win7 SHA-256 Windows Update Update, where remember that right now Windows updates are being cosigned, but that's being dropped after June, so starting with July. So it's necessary for us, for Windows 7 users, to be sure that we're able to deal with non-cosigned updates. We have an answer as a consequence of last week. Big news regarding the many attacks which have surfaced to leverage the recently discovered WinRAR vulnerability that we just talked about. Turns out it's no longer a theoretical problem.

We're going to look at what happens when Apple, Google, and GoDaddy all drop a bit. An update on a big recent jump in the Mirai botnet capability. Some worrisome news about compromised Counter Strike gaming servers and the surprisingly large percentage of them that were infected and infecting the players of Counter Strike. Some welcome privacy announcements coming in the next Android Q. And Leo, you need to brainstorm with me what tasty treat begins with the letter Q. I got nothing.

Leo: It's tough. We've been worried about - there's been this Q crisis. We've been thinking about this for several generations now. And all we can come up with is quince jam, which is not any good.

Steve: No. I don't think I want any.

Leo: No, thank you.

Steve: So anyway, we have also a pair of very odd web browser extensions, one for Chrome, the other for Firefox, from Microsoft. Which really kind of sets me off on a, well, as our listeners will see, a bit of a quandary and then a rant.

Leo: Oh.

Steve: Because it's just like, what? You've got to be kidding me.

Leo: Oh, I can't wait to hear this, yeah.

Steve: And we have a bit of follow-up on last week's Spoiler topic, some closing-the-loop feedback from our terrific listeners, and then we're going to take, as I started off saying, an early look at what is a very exciting and encouraging project to create an entirely open eVoting system platform with secure hardware from DARPA, secure software that will be open and audited. But I don't want to give it all away. So I'm going to just bite my tongue at this point and just say, hold on till the end of the podcast.

Leo: Steve?

Steve: So our Picture of the Week is something that confronted me when I was doing some research for the podcast. You can see behind the dark overlay that it's the McAfee website. And I was going there to look at their coverage of the WinRAR absolute path traversal vulnerability that we talked about. And instead I get this dark cover and a pop-up saying - this is from McAfee; right? That used to be a security company. "Your browser is blocking some features of this website. Please follow the instructions at" - and then I have http, notice no "s," http://support.heateor.com/browser-blocking-social-features/ - "to unblock these." That is, to stop my browser from doing what I intended it to do for the sake of security. So thank you, McAfee. I just grabbed a screenshot of this. I thought, okay, that's the Picture of the Week.

Leo: It was probably like the Facebook Like button or something, or social sharing.

Steve: Yes. Yeah.

Leo: Come on.

Steve: Leave me alone, yeah.

Leo: That's not okay.

Steve: Wow.

Leo: Wow.

Steve: Yes. Your browser is blocking some features of this website. Yeah, apparently something I don't want. And so thank goodness it's still my browser. Well, yeah, it is, actually. Okay. So we had March Patch Madness last Tuesday. Last week we talked about the Google Chrome exploit which was leveraging a pair of zero-day vulnerabilities. Remember one in Chrome and another in Windows. Combined, they were being used against Windows 7 users. Microsoft had only been informed of the problem the week before, and I wondered aloud last week whether this would give them time, or maybe it was the week before, to get it fixed for March. Well, they did, and it is. So that's good.

But it turns out there was a companion zero-day. Remember the news was, oh, this had only been seen being exploited in the wild for Win7. Turns out there was a probably identical vulnerability that, due to some code changes between 7 and 8, and of course we don't really care about 8, as I heard you guys, I guess it was talking on Windows Weekly recently, saying does anyone even know anybody who has Windows 8? I don't, you know, we just sort of - that was an embarrassment that Microsoft, well, and Vista also. And now everybody's on 10, of course.

So in total, Microsoft addressed 64 now CVE-listed security vulnerabilities across a bunch of their products, of course, in Windows OSes and other products. Seventeen of those 64 were critical, 45 rated as important, one moderate, and one low in severity. They spanned the OSes, IE, Edge, Office, SharePoint and the Chakra Core, Skype for Business, and Visual Studio NuGet. And I didn't even take the time to figure out what that was. That's something that's new to me. Four of the important security vulnerabilities had been disclosed publicly, but none were known to be exploited in the wild.

But in addition to the one zero-day involved in that actively exploited in-the-wild Chrome attack, which Google immediately shut down, thanks to the fact that they've got on-the-fly constant patching of Chrome, Microsoft also patched that other zero-day that I mentioned which was, it turns out, it was also under attack. Both were elevation of privilege flaws residing in the win32k.sys driver. Whereas that first zero-day was only affecting Win7 systems, the second zero-day was also an elevation of privilege vulnerability, was also being exploited in the wild, but not as a consequence of leveraging Chrome. And it affected all Windows and Server versions after Windows 7, which is why I think since they were so closely related it was probably the same problem expressed differently due to some code changes after Windows 7.

It's also worth noting that last Thursday the 14th, the Chinese 360 Core Security group decided to publish a working proof of concept for this Win7 vulnerability. Their release justified this, I think, well, semi-justified it by saying: "Considering that some users are still using Windows 7" - well, yeah, half of us, or just now it's flipped from just barely more using Windows 7 to just barely more using Windows 10, like, what, a couple months ago. And of course that'll change once the security updates stop a year from last month, so it'll be February 2020 that real pressure begins to be put on people to finally pry them away from Windows 7.

Anyway, they said: "Considering that some users are still using Windows 7, this vulnerability, combined with Chrome's RCE, has been used for real APT," they wrote, Advanced Persistent Threat - "attacks. So this zero-day is very likely to be exploited to perform large-scale attacks and pose a real threat," and as I mentioned, except that it was fixed in Chrome even before its announcement. So it's like, okay.

They said: "Therefore, 360 Core Security Technique Center constructed the proof of concept and reproduced the vulnerability triggering process so that security vendors can reference to increase the corresponding protection measures." Okay. I mean, what we're really seeing is we're seeing a problem increasingly with these proof of concepts being published before there has been a chance for their mitigations or their fixes to happen.

And so this was on Thursday, two days after Windows Patch Tuesday update that fixed the problem. I would argue that's not time enough to allow people, I mean, many people put off rebooting. I mean, it's funny, some of these updates say "May require a reboot." But it may require you to stop using your system for an hour. Well, okay, maybe not. But still, certainly a reboot. So we know that there are instances, I encounter them in my daily life, of people who just say, no, no, no, not now. No, no, no, not now. And they keep pushing it off.

So the point is I don't think two days works. I would say two weeks probably would be responsible. And there's been dialogue now in the security community about how long someone should responsibly wait before pushing out a proof of concept. What's really happening is they're wanting to draw people to their site by producing it. And the strong argument is it's of more use to bad guys than it is to good people at this point. So, and again, like why does a security vendor need to worry about this if Microsoft has fixed it in their patches? It's a non-problem like all the other non-problems that Microsoft has patched for the last three decades. So it's like, okay.

Anyway, that second flaw was detected and responsibly reported to Microsoft by security researchers at Kaspersky Labs, who in a blog post coincident with the patch's release revealed that the flaw - without giving any details, they were responsible - was being actively exploited in targeted attacks which we did not know before by several threat actors including one that we've spoken of before, FruityArmor, okay, and SandCat.

And ZDNet in their coverage had an interesting note about the FruityArmor/SandCat connection. ZDNet wrote: "The November zero-day" - so speaking of a previous one - "was also abused by SandCat, a new group on the APT" - again, Advanced Persistent Threat - "scene about which Kaspersky has few details, such as its use of the March and November zero-days, the Chainshot exploit, and the FinFisher/FinSpy hacking framework. What all this tells security experts is that there's at least some type of connection between these two threat actors, FruityArmor and SandCat. They are either managed by the same intelligence service, or" - and this is what kind of gave me a chill - "they're buying zero-days from the same exploit vendor."

Leo: Ooh.

Steve: And of course we've been - yeah - we've been talking about exploit vendors recently. And I thought, wouldn't that be creepy if we're seeing separate entities now having essentially the same exploits in the same timeframe because in - and again, in targeted attacks, again, that's what you would expect if you're buying zero-days from an exploit vendor. You're not doing opportunistic spray spam campaigns. You're going after specific individuals or organizations with an individual to target attacks, and thus to minimize the exposure of this because, once it becomes well known, it's no longer of any use.

Leo: Although, I mean, it's possible, isn't it, that Zerodium or somebody like that sells - I bet you for a price you can get exclusive access to a flaw, and for a lower price you can get shared access; right?

Steve: Right, right. I would bet that's true. It's sort of like...

Leo: Yeah. But I hope that, what is it, FruityCat? FruityArmor and SandCat?

Steve: FruityArmor, yeah.

Leo: They could be the same guys, too; right? I mean...

Steve: They absolutely could, yes. And also some of the naming of these is somewhat obscure. I mean, they're not like declaring themselves. Normally a disassembly of their code finds...

Leo: We are FruityArmor, and we are here to investigate.

Steve: Exactly.

Leo: Wow, it's so funny. It's so funny.

Steve: Exactly. So as for the overall March Madness patching, as usually, nearly all of the critical rated vulnerabilities do lead to remote code execution attacks and primarily impacted various versions of Windows 10 and Server editions. Most of the flaws reside in the Chakra Scripting Engine, VBScript Engine, DHCP Client, and IE. In other words, notice those are all things that have Internet-facing attack surfaces. So that is now where we need to be paying the most attention. While some of the only important vulnerabilities can also lead to remote code execution attacks, others allow elevation of privilege, information disclosure, and denial of service attacks.

Okay. Part 2 of this is what about this SHA-2 Windows Update signing update? I also confirmed last Tuesday that my Win7 machine did acquire with last Tuesday's updates the ability now to verify SHA-2, which is SHA-256. Technically we had SHA-1 that was just a single, sometimes called SHA-1, a single hash, and I'm thinking it's what, 160 bits. SHA-2 is actually a second-generation of SHA hashes, one of them being SHA-256, so that gives you a 256-bit hash output, substantially bigger than SHA-1's 160-bit hash output. So thus better.

Anyway, so the point was that Microsoft announced, and we covered several months ago, that this was going to be happening. But what wasn't clear, no matter how closely you read the disclosure from Microsoft, because they were like warning people, and there was some sense of you may have to go get this. So if that was the case, I wanted to make sure everybody who was still using Windows 7 did because updates would just unceremoniously stop after June. So the answer is yes. My system obtained the update. And that's 4474419.

And so if you just look, if a Windows 7 user is curious, you can just look in your history, your update history, and a ways back, you'll probably see a bunch of daily, since last Tuesday, you'll see like a bunch of daily Windows Defender updates, assuming that you're doing that. And then you'll see the Patch Tuesday batch, and you'll find 4474419, which says that, or means that your system now is able to - it's been updated, not to understand SHA-256 hash. It had that already in order to be using any of the SHA-256 signed TLS certs that are now on the Internet. We've had that for a long time. In fact, it has had it for a long time. Windows XP was actually getting that back with Service Pack 3, so Windows 7 has had that, but not for Windows Update updates. Now it does. So we know that moving forward we will be able to get them.

And, you know, I was worried that, if someone hadn't updated, then they might sort of get in a Catch-22 situation after June. But you'd still always be able to get the double-signed updates from like March and April and May and June, which would include this 4474419, and then you'd be able to continue moving forward. So I'm sure that even, for example, new systems set up with Windows 7 after June will be able to get themselves brought current until Microsoft decides they don't want to do this anymore, starting with next February.

Okay. So I talked at some length a couple podcasts ago about this WinRAR vulnerability that was discovered by Check Point Research. What they found, to recap, is that - and this was about a month ago - was that an old DLL for which the source had been lost was able to decompress ACE archives. Okay. So WinRAR knows about a whole bunch of different archive formats. And so just it was a bullet point in the features list, is yeah, we can open ACE archives. No one can make them. But if there are any lying around the Internet somewhere, WinRAR could open them.

It was a theoretical concern when it was announced. So this was not a zero-day. This was not discovered being exploited. I don't think it was. Now I'm questioning myself. But I don't think so. What they found was - they called it an "absolute path traversal" vulnerability, which allowed an ACE archive to be created that would allow an executable to escape from the directory where it was being expanded or decompressed or de-archived to, which is typically down, like under the My Documents or My Downloads or whatever folder, which is not going to - so if there was an executable there, it wouldn't automatically be run.

The other trick was that WinRAR determines archive type by inspecting it, not by believing the file extension. So someone might think, what the heck's an ACE, and not download it. But if it said .rar, and if somebody had installed WinRAR, they probably understood what a .rar was, that it was a high-compression, typically high-quality archive. So at the time this was announced, RAR Lab immediately dropped support with v5.70 that was released on February 27th. By the time I reported this on the podcast, I had updated and was able to confirm that not only, yes, had ACE support disappeared, but the two ACE-related files had been proactively deleted from the subdirectory where WinRAR was installed to prevent any mischief of it.

Well, okay. So get this. The problem is WinRAR doesn't have any form of auto update mechanism. And I really don't fault an archiving tool for not auto updating, especially one that is two decades old. It's 19 years old. Except now we wish it had one because what this means after 500 million users have downloaded WinRAR through the years is that all of them are vulnerable to this, and almost none of them are going to get updated because most users, 500 million, we wish we had that many people listening to this podcast, but we don't. Which means unless they somehow, I mean, it's just not going to happen that any of these versions of WinRAR which were downloaded and installed over this period of time are going to be updated. Yes, some fraction, but not significant. And the bad guys know that. The bad guys know that that's the case.

So what's happened is we have now seen, as reported by McAfee, hundreds, more than a hundred different exploits trying to target this vulnerability to install malware in people's machines. In one recent example, there was a bootlegged copy of Ariana Grande's hit album, "Thank U, Next," that had the filename Ariana Grande...

Leo: Aptly named.

Steve: Huh?

Leo: Very well named. Thank you. Next.

Steve: Yes, exactly. Exactly. It was "Ariana_Grande-thank_u,_next(2019)_[320]," presumably meaning it was a high-quality compression of her album, dot rar. And if you were to decompress that, you would end up with a trojan, a remote-access trojan installed in your startup folder so that the next time you restarted Windows it would run without any UAC prompting, no user interaction, and install itself into your system.

So this has gone from a theoretical problem as a consequence of the fact that the bad guys also, as I, do not expect this thing to essentially ever get fixed. So it will take the malware, the AV tools - and I have not looked. What I do know is that at the time of this Ariana Grande album being released, the it as malware, that is a malformed RAR archive, was detected by only 11 AV, and 73 AV products failed to alert the users of anything being amiss. So hopefully this will get preempted by Windows Defender and other AV tools stepping up and starting to filter this because otherwise this is going to be bad for users.

And there is a flavor of ransomware, as well. These 360 Threat Intelligence Center guys spotted in the wild one of these WinRAR archives which was called "vk_4221345.rar" which had a compressed picture of a girl, only it looked like the decompression had failed so that it was cut off halfway. Maybe that's to induce the person to open the archive to hopefully get the whole picture or others or who knows what. I think it was Chinese, and so it was difficult to understand exactly what was going on.

Anyway, it installs a ransomware malware payload which they named JNEC.a. Written in .NET, it asks for 0.05 bitcoins. And with bitcoin now generally hovering around $4,000 U.S., that's about $200 U.S. ransom payment they're asking for. So anyway, I mean, this is what we're seeing. A new vulnerability comes to light, and we are now, I mean, there's just - there's an industry, essentially, in place, ready to turn that thing into exploits of one form or another. So what a world we're in.

Okay. This is really interesting. GoDaddy, Apple, and Google have technically mis-issued more than two million TLS certificates. And this sort of falls under the heading of one thing leads to another. So remember our recent discussion about that sketchy wannabe UAE-based certificate authority who decided to name themselves DarkMatter. And it's like, okay, we're going to be a trusted certificate authority called DarkMatter. And of course they have been accused in the past of using their technology, their man-in-the-middle middlebox technology on behalf of repressive regimes to spy on people. And now they're appealing to Mozilla to have their CA signing public key added to Mozilla's root certificate store so that we'll trust their certificates without question. What could possibly go wrong?

Well, as it turns out, as a consequence of the discussion in mozilla.dev.security.policy group about DarkMatter's controversial application to become a fully fledged cert-issuing CA, people were poking into DarkMatter's existing countersigned certs and happened to discover that the company's supposedly 64-bit serial numbers which were being embedded in its certificates were actually, Leo, coming up one bit short - 63 bits.

Leo: Oh.

Steve: But then engineers at other major organizations, and by that I mean GoDaddy, Apple, and Google, who were reading the thread, realized that their own certificates - whoopsie.

Leo: But what's the impact of being a bit short?

Steve: And that's where we're headed next.

Leo: Okay.

Steve: So they realized their own certificates were similarly affected. So as I said, one thing leads to another. So what's behind, first of all, what's behind this broadly made mistake? And then we'll talk about what does it mean.

As it turns out, this is a consequence of everyone using a not technically RFC fully compliant default setting in a commonly used open source certificate serial number generator. It is actually an open source CA. It's known as EJBCA. And Leo, if you google EJBCA, you'll see, like, it's a big deal. GoDaddy, Apple, Google, and apparently many others, as a consequence of them using this without like looking at all the default settings, are now facing the revocation and reissuance of more than two million certificates. GoDaddy alone estimated that they had issued 1.8 million certs which had 63-bit serial numbers.

So, okay. In my opinion, this is ridiculous. This is a tempest in a teapot. But, you know, standards are standards. Rules are rules. I went digging into RFC 5280, which is the RFC standard. The title is "Internet X.509 Public Key Infrastructure Certificate and Certificate Revocation List Profile." It says in Appendix B, ASN.1 Notes: "CAs MUST [all caps MUST] force the serial number to be a non-negative integer." Well, uh-huh. A negative serial number would be a little screwy; right?

And then it says: "That is, the sign bit in the DER encoding of the integer value MUST [all caps again] be zero. This can be done by adding a leading leftmost 00," you know, null, all-zero, they use the term "octet," which is the fancy word for byte, "if necessary." So they're saying stick a double-zero, you know, two hex zeroes, eight bits of zero, on the front, if you must. That way you're sure you're going to get all eight of your leading zero bits, or eight of your leading bits are zero. So you've got lots of margin.

Anyway, the Appendix B goes on: "This removes a potential ambiguity in mapping between a string of octets [a string of bytes] and an integer value. As noted in Section 4.1.2.2, serial numbers can be expected to contain long integers. Certificate users MUST [all caps] be able to handle serial number values up to 20 octets" - that is, 20 bytes, so what's that, 160 bits - "in length. Conforming CAs MUST NOT use serial number values longer than 20 octets." So they said you could have serial numbers up to but not longer than, 20.

Okay. So who is this EJBCA that got everybody in trouble? They call themselves "The Open Source CA." Their own site says it's "a PKI [Public Key Infrastructure] Certificate Authority software built using Java (JEE) technology. Robust, flexible, high performance, scalable, platform independent, and component based, EJBCA can be used standalone or integrated with other applications." Unfortunately, they didn't happen to mention in there that they're not RFC compliant. Whoops. Okay.

But they go on to say: "Extremely scalable and flexible, EJBCA is suitable to build a complete PKI infrastructure for any large" - I mean, Google, GoDaddy, Apple, so yes - "any large enterprise or organization. If you only want to issue a few single certificates for testing, there are probably other options that will get you started quicker. But if you want a serious Certificate Authority to manage your Public Key Infrastructure" - and cause you to reissue millions of certificates, no, it doesn't say that - "we recommend EJBCA." And Wikipedia goes on to talk about them. I mean, so they're a well-known group.

Okay. Unfortunately, until recently, this very nice-looking and capable open source CA system was configured to generate eight octet serial numbers. Eight octets is eight bytes, which is 64 bits. Serial number fields are defined, and I do not know why, as "signed integers." This must be some nutty committee decision, since a negative serial number makes no sense. But that's what it is. It's signed.

So let's take a moment to remember binary number format. The industry standard two's complement math, which is what all of our computers use, that's what these little blinky lights behind me are doing is two's complement math. The number zero is represented as all binary bits being off. All binary bits are zero. If you start to increment that binary value, the first bit, the lowest order bit, the rightmost bit when they're all stretched out linearly, the rightmost bit turns to a one. You increment it again, it goes back to zero, and the bit to its left turns into a one. You increment again, then they're both ones. And again, they both go to zero, and the third bit turns on. And so on.

So what if we decrement? We start - so go back, reset our register to all zeroes. Now we decrement. What happens is all the bits go to one. So minus one in two's complement math is all binary one bits on. If you decrement again, then the least significant one goes to zero. Decrement it again, it goes back to one, and the one to its immediate left goes to zero. Decrement again, both lower order bits are zero, and so on. So if you think about it, what this sort of means is, and actually does mean, that the highest order bit, that leftmost bit is the sign bit. If it's zero, then you have a positive binary number. And if it's one, then that binary string of bits in two's complement representation is a negative number.

So the RFC says we need to generate for a certificate a high-entropy, that is, like 64 bits of entropy, serial number. And it also says it must be positive. So doing so is simple. You ask your entropy source for some number of bytes. You get them. And then you turn off, you make sure the highest significant bit, the leftmost bit, you just force it to zero. So it was normally going to be a one or a zero with 50/50 probability, right, because all the bits in all of the bits you get are going to be 50/50 chance of being a one or a zero. So you just make sure, you just clear, you set to zero the most significant bit. Now you're guaranteed that the result is positive, is a big random positive number.

Unfortunately, it no longer has 64 bits of entropy; right? It used to. But that's because we weren't sure if that last bit was going to be a one or a zero. We're now - we made it zero. Well, that just killed off a bit of entropy. Now we only have 63 bits of entropy, and we have fallen out of spec. And that's what this EJBCA CA was doing to all of the certificates that it generated.

So that's not good. Okay. But what does it really mean in terms of security? And this is why I think this is all a bunch of nonsense. It's technically not in compliance. I guess that's enough. But on the other hand, all of those mis-issued certs are going to be expiring themselves after two or three years. And revocation doesn't work. We know that. We've covered that extensively. So how is one bit less of entropy in certificate serial numbers a big cause of concern? Is it?

So what is the concern with a certificate serial number? The concern is that serial numbers are sometimes used to pin certificates. That is, you say, okay, I want to pin the cert, and you use its serial number. You record it and you make sure, like, when you get that serial number from a remote server, you check the serial number, having memorized it earlier, to make sure it hasn't changed. I mean, I guess that's, I mean, certainly that is being done. To me, it makes much more sense to use the much longer thumbprint hash, but people often, you know, which is included in the certificate, so you could check it.

But the serial number is there. It's part of the spec. And people may want to rely on it. So what we're concerned with is the chance that another randomly generated certificate's serial number might have the same serial number as the one we have pinned. That is, there could be a serial number collision.

So with the classic "birthday paradox," that doesn't apply here, as we'll see. With the birthday paradox we would be asking, given a pool of certificates of a certain size, and each certificate containing some entropy of a certain size, what's the likelihood of any pair, any two certificates, having the same serial number in that entire set? That's the birthday paradox. And what's surprising, we've talked about this in the past, is how quickly that probability falls as the size of the pool increases. And this happens, of course, because every certificate serial number is being compared with every other certificate, kind of all at once.

But that's not the concern here. In our case, we're just worried that one other certificate serial number might have the same serial number as the one certificate we have pinned that we care about. So that makes the math simple. The math on the birthday paradox is 1.2 times the square root of the size of the pool is an estimate. But that doesn't apply here. The CAB spec, the CA Browser spec wants that two-cert serial number collision probability to be no greater than one in 2^64.

So I did the math. That's one chance in, and the number is 18446744073709551616. Naming large numbers, we've talked about large number naming in the past. We know there's million, then billion, then trillion, then it goes to quadrillion, then to quintillion. That's where we are here. So with the full 64 bits, the chance of collision with another randomly generated certificate serial number is one in 18.446 quintillion. That's what the spec wants. When we lose a bit by forcing the first bit to be zero, we of course cut the total universe of possible serial numbers in half, thus doubling the collision probability from that one in 18.446 quintillion down to only one in 9.223 quintillion.

And so I just cannot get very worked up about that, especially given the fact that these certs, with their one-bit-shy serial numbers, will all be expiring themselves in two or three years. And notice that, if the serial number you are comparing with is longer, then it can't possibly collide with one of these 63-bit serial numbers. So if your serial number that you have pinned is 256 bits, well, it's going to be different. No possibility of collision. So it's like, okay.

But despite the fact, rules are rules, specs are specs, and apparently Apple, GoDaddy, Google, and who knows who else are all now in a froth, feeling like - in fact, it's supposed to be done within a few days. I think seven days is what the requirements call for if this sort of problem occurs. GoDaddy just shrugged and said, "We can't. I mean, we physically, logically, logistically can't reissue 1.8 million certificates in seven days. It's going to take us at least a month."

Leo: And by the way, I believe our TWiT certificate is a GoDaddy certificate. I'm not sure. Most of them are DigiCert.

Steve: It just doesn't matter. It just doesn't matter.

Leo: No, it's not insecure.

Steve: Right. It's not about security, it's about a serial number that, first of all, isn't used. Nothing actually uses it...

Leo: Oh, it's crazy.

Steve: ...for the transaction or for TLS or anything. It's only there, I mean, it's just a bit of entropy to give the certificate a unique token, a minimum 64-bit token. It turns out, whoops, because it has to be positive, and because this lame bunch of committee people said, oh, it's signed, so make it positive, and so the way you do that is you make sure it has a leading zero, and this is the result. I also got a kick out of the fact that the newsfeed over at EJBCA, the top of the feed currently notes, says: "EJBCA can be configured to generate certificate serial numbers." And it says, parens, "(positive integers from 4 to 20 octets)."

Leo: Oh, man.

Steve: And then they said configurable - this was posted on March 13th. "Configurable serial number entropy default value raised to 20 octets." So too little, too late in the case of many millions of certificates. Anyway, this should just kind of be noted and then, like, okay, everybody ought to get a pass on this. But I guess that's not the way this is being handled.

Leo: Wow.

Steve: So Leo.

Leo: Yes.

Steve: I have bad news for you.

Leo: Oh, no.

Steve: Your serial number is 654AB5B273E66A34.

Leo: Yes.

Steve: Sixty-three bits.

Leo: [Exclamation].

Steve: My serial number.

Leo: Yeah? Yeah?

Steve: 07A7894CFA6A7FEFEB90996F59D9F402. 128 bits. Actually 127 bits.

Leo: How come yours is so long?

Steve: I got mine from DigiCert.

Leo: Ah. We should have done DigiCert.

Steve: That's right.

Leo: By the way, the ransomware has shut down one of the world's largest producers of aluminum. Norsk Hydro of Norway, malware hit computers in the U.S. on Monday night. By Tuesday morning it had spread to other parts of the company. It's in 40 different countries. Some of the plants, this is one of the biggest aluminum producers in the world, plants have been stopped or disrupted because of ransomware.

Steve: Yeah.

Leo: Unbelievable. It's LockerGoga. I don't know if you've heard of that one. LockerGoga ransomware.

Steve: Wow.

Leo: So it's still happening. It's incredible. Well, so you got all the bits.

Steve: Yeah, so...

Leo: But it doesn't mean we're insecure or anything; right?

Steve: No.

Leo: There's not going to be a collision.

Steve: No. Well, it means that your certificate is going to be revoked, baby.

Leo: Yeah.

Steve: I mean, someone's going to come...

Leo: Well, I thought you said GoDaddy was going to just give up.

Steve: Well, I don't know what they're going to do. But, I mean, you've got one of the bad ones.

Leo: We got one of the bad ones.

Steve: Yeah. Now, notice that the first byte is 65. So a 65 hex is 0110. So if it were eight or greater, well, it can't be because then the most significant bit would be one, which would be a negative serial number.

Leo: Yes, right.

Steve: But what's dumb is that, well, at least Firefox is showing me the serial numbers in hex. So it doesn't - it's not signed. It's not a signed integer.

Leo: It's not signed. It doesn't matter.

Steve: It's not an integer, exactly. So this is all like a display issue, not anything more. But again, this is another example of DigiCert doing it right, and their serial numbers are double the minimum size and half the maximum. So they're not going to break anything at the high end, and they're absolutely compliant at the low end. So, but it does mean you've got one of the noncompliant certificates. So presumably, like all of the certificates GoDaddy has been issuing are like this. Because you've probably had yours for a while. This is not like this just happened yesterday.

Leo: We renewed like in the last six months.

Steve: Okay.

Leo: And normally we use DigiCert. And Russell said - Russell. "GoDaddy's a lot cheaper. Why don't you use GoDaddy?" Russell. And I said, oh, you know, it's expensive because we have a wildcard cert for TWiT. You know, saved me some money. So I was cheap. And you see? You see what you - you see, you get one fewer bit. You save money, or in your case I get 64 fewer bits.

Steve: Yeah. And unfortunately, that's one too few.

Leo: I like big bits. I cannot lie.

Steve: Sometimes you get what you paid for.

Leo: Yeah. One bit short.

Steve: My company comes through again.

Leo: Yeah, I really - honestly, I love DigiCert. And my own personal certs are all with DigiCert. But I cheaped out and lost a TWiT bit.

Steve: So speaking of - this is sort of apropos, actually, this ransomware attack. The Mirai botnet is alive and well and more scary and capable than ever. We'll remember that it is considered an IoT malware. It broke the DDoS attack size record in 2016 when it was used to attack Brian Krebs, forcing him off the 'Net. Also the French web host OVH was attacked. And then, most famously, the DNS provider DynDNS was forced off the 'Net. And of course because we all depend upon DNS to varying degrees, that caused a ripple effect, and all kinds of other sites disappeared as their DNS expired, and it couldn't get refreshed from their assigned DNS provider that was in this case DynDNS.

So Mirai has been updated to target a new crop of devices including two which are often found inside enterprise networks where, as we know, bandwidth is often more plentiful than it is on consumer IoT networks. Mirai now knows how to infect webcams, routers, DVRs, as well as many other Internet-connected devices which typically ship with default credentials, as we know, and typically they're running never updated, and thus woefully outdated, versions of Linux.

Yesterday morning Palo Alto Networks Unit 42 - I love it that they call themselves Unit 42. They posted news of a new Mirai titled "New Mirai Variant Targets Enterprise Wireless Presentation and Display Systems." So I've edited this down a bit for length and clarity. But they basically wrote: "The Mirai variant that Unit 42 discovered is notable for targeting different embedded devices like routers, network storage devices, network video recorders, and IP cameras, using numerous exploits against them. Specifically, Unit 42 found this new variant targeting wePresent WiPG-1000 Wireless Presentation systems and LG's Supersign TVs."

They wrote: "Both these devices are intended for use by businesses. This development indicates to us a potential shift to using Mirai to target enterprises." Attack code exploiting a wePresent command-injection vulnerability was published in 2017, while a remote code execution exploit for the LG Supersign TVs has been available since last September. After being packaged into this new Mirai variant, the exploits become much more likely than previously to actively be used to compromise their vulnerable devices.

And this is not the first time Mirai has been aimed at enterprise networks. Last September Palo Alto Networks reported that Mirai was found targeting the same Apache Struts vulnerability that hackers exploited to breach Equifax. So in addition to this newer targeting, this new Mirai variant incorporates 11 new exploits in its multi-exploit kit, and has four sets of new credentials used in brute-forcing device sign-on. So Mirai is still alive and well. It had 16 previously seen exploits. It added 11.

So it's gotten way more competent, and it is still out there scanning the Internet, looking for new victims and, unfortunately, finding them. It uses an HTTP flood to do DDoS attacks, and it is a worm. So once it gets a beachhead, it then starts scanning for other available devices, both inside and outside of its network. So it doesn't look like the Internet is going to be getting rid of Mirai anytime soon. And we're just going to be stuck with this thing, and it's going to be causing more havoc.

This is odd, Leo, and I wonder if - I guess it's really not on Paul's radar. But I think of him because of gaming. Counter Strike has been striking its players. The malicious network has now been taken down, but there's a useful lesson, I think, to be learned in its aftermath. The servers in question were malicious Counter Strike 1.6 servers which were being used in zero-day attacks to infect game players with malware. The Russian AV firm Dr. Web found that 39%, that's 1,951, 39% of all Counter Strike 1.6 servers were malicious and were actively trying to infect their users with malware.

Dr. Web wrote in their report, which I'll summarize, they said: "Introduction: The game Counter Strike 1.6 was released by Valve Corporation back in 2000." So it's been around for quite a while. They said: "Despite its rather considerable age, it still has a large fan base. The number of players using official CS 1.6 clients reaches an average of 20,000 people playing online." Okay. So it's not 200,000 or two million, but 20,000.

They said: "While the overall number of game servers registered on Steam exceeds 5,000, selling, renting, and promoting game servers is now deemed an actual business, and these services can be purchased on various websites. For example, raising a server's rank for a week costs 200 rubles." And I did the conversion. That's $3.11 currently.

Leo: It's cheap.

Steve: Yeah. And they say, "Which is not much. But a rather large number of buyers makes this strategy a rather successful business model. Many owners of popular game servers also raise money from players by selling various privileges such as protection against bans, access to weapons, et cetera. Some server owners advertise themselves independently, while others purchase server promotion services from contractor. So, yeah, there's a commercial ecosystem there. Having paid for a service, customers often remain oblivious as to how exactly their servers are advertised.

"As it turned out, the developer named 'Belonard' resorted to illegal means of promotion. His server infected the devices of players, that is, players' PCs, with a trojan, and used their accounts to promote other game servers. The owner of the malicious server used the vulnerabilities of the game client and a newly written trojan as a technical foundation for their business. The trojan infects players' devices and downloads malware to secure the trojan in the system and distribute it to devices of other players. For that, they exploit remote code execution vulnerabilities, two of which have been found in the official game client and four in the pirated one.

"Once set up in the system, the Belonard trojan replaces the list of available game servers in the games client and creates proxies on the infected computer to spread the trojan. As a rule, proxy servers show a lower ping, so other players will see them at the top of the list. By selecting one of them, a player gets redirected to a malicious server where their computer becomes infected with trojan.belonard. Using this pattern, the developer of the trojan managed to create a botnet that makes up a considerable part of the CS, the Counter Strike 1.6 game servers.

"According to our analysis, out of the some 5,000 servers available from the official Steam client, 1,951" - so just shy of 2,000 - "were created by the Belonard trojan. This is

39% [as I said] of all game servers. A network of this scale allowed the trojan's developer to promote other servers for money, adding them to lists of available servers in infected game clients. We previously reported," they wrote, "a similar incident with Counter Strike 1.6 where a trojan could infect a player's device via a malicious server. However, a user then had to approve the download of malicious files, while this time a trojan attacks devices unnoticed by the users."

Dr. Web, this outfit, have informed Valve about these vulnerabilities and other vulnerabilities of the game. But as of now, there is no data on when the vulnerabilities will be fixed. Trojan Belonard consists of 11 components, so it's not, I mean, it was an effort to put this together, and operates under different scenarios depending on the game client. So it's also multi-homed. If the official client is used, the trojan infects the device using an RCE (Remote Code Execution) vulnerability exploited by the malicious server, and then establishes within the system. A clean, pirated client is infected the same way. If a user downloads an infected client from the website of the owner of the malicious server, the trojan's persistence in the system is ensured after the first launch of the game. Wow.

So there's really nothing that I can suggest that a user might do to protect themselves from this threat. Even the official Valve client has two known and exploited by this trojan vulnerabilities. So perhaps stick to well-known and trusted game servers. What, a little over 60% of them are not part of this trojan server network, so you'd be safe there. Maybe use a throwaway PC, if that's your platform. Just because you're sort of in a high - unfortunately, gaming in a gaming server network can be high risk. So, you know, don't do this on your main system where all your banking is being stored. And I guess the best takeaway is to maintain an awareness that this kind of thing is going on and just be a little bit more cautious and suspicious than you would otherwise be.

Next release of Android is Android Q. And as we said, Leo, I mean, I thought for a while. I can't think of a tasty treat. You know, we've had Marshmallow, and was R Raisin? Or was that something else?

Leo: What was R? Rocky Road? I don't remember. But Q, there's one thing, there's a French dessert, the Quenelle. But, see, this is the problem. There's no obvious kind of English-language dessert. I don't know what they're going to do.

Steve: Did we start with A? Or did we jump? I don't remember.

Leo: There were apparently A's, B's. But Cupcake was the first public release, C.

Steve: Oh, okay.

Leo: And I used to be able to rattle them all off.

Steve: Was it Donut for D? Eclair?

Leo: Eclair, Donut, yeah. Cupcake, Donut, Eclair, Froyo.

Steve: Oh, that's right, yeah.

Leo: Remember? Yeah. G, what was G? Gingerbread.

Steve: Ah, right.

Leo: Now, see, you've got me started. I'm going to start, I'm going to have to finish. Honeycomb. "I" I don't remember. I'm sure Jason Howell could do them all. Ice Cream Sandwich.

Steve: Well, I will be excited to see what they come up with for Q because, I mean, they're on a roll.

Leo: They should just skip Q and go to Rocky Road. Honestly, that's an obvious R, so...

Steve: Yeah, yeah.

Leo: I don't know. We've been wondering about this ever since P.

Steve: Is Zagnut a...

Leo: Zagnut, yeah. Only once did they use brands.

Steve: Oh, right, right, right, instead of it being [crosstalk].

Leo: And that was kind of a mistake, I think.

Steve: Yeah, yeah.

Leo: So I don't know what they're going to do. KitKat, that was K.

Steve: Well, what we do - oh, yeah, right, of course.

Leo: Oreo is a brand, actually, yeah.

Steve: Yes, yeah, yup. What we do know is that Android Q, whatever it ends up being called, will finally be delivering really robust Mac address randomization. Also it'll have new location permission pop-up, which I actually have on the second page of this story. I grabbed a snapshot of it. And it will be preventing clipboard sniffing, which has been a big privacy concern.

The beta of Q was first released last week, promising a bunch of welcome privacy improvements. As for access to clipboard data, Android apps all used to have access, for example, as Windows apps today do. But Android apps will no longer be able to access the Android operating systems clipboard data, the shared clipboard, unless they're in focus, running in the foreground, onscreen. The exception which had to be made is the default input method editor, i.e., the system keyboard. It does have access to the clipboard all the time. But that really reduces the clipboard attack surface dramatically.

Also, Android Q will have Mac address randomization on by default. Google introduced Mac address randomization in Android 6.0, but devices broadcast a random Mac address only when the smartphone would initiate a background WiFi or Bluetooth scan. Android Q devices will now transmit a randomized Mac address by default at all times and for all communications. And as far as I know, this bests iOS as a privacy feature.

Last time I looked, and last time we talked about it, iOS was broadcasting a random Mac address when it wasn't associated with a WiFi access point, but it reverted to its real fixed and unchangeable factory set Mac address when it was actually associating with a WiFi access point, presumably so that the WiFi access point could determine which iPhone or iOS device it was. But of course that's a privacy breach because you don't know who's reading those Mac addresses from the access point that you have opportunistically associated with.

So bravo to, I mean, and there's no reason it can't be a random Mac address unless maybe you would want to pin the Mac address to give privileges on - but, again, that's not really secure because Mac addresses can be spoofed. So since they can be spoofed, let's just always spoof them and throw, I mean, maybe I'm wrong, and iOS fixed this, and Rene is, like, saying "Steve, Steve," you know, I don't know. I'm sure I'll find out.

Leo: I feel like it is something iOS does, but I don't off the...

Steve: Well, they went halfway. I remember when we talked about this they were doing random until you associated, and then they used their real Mac address. Android Q is never - doesn't have a real Mac address. It's just going to make one up, which is cool.

Leo: Here's an article from The Register from last year. Thomas Claburn, who's pretty good. "Mac randomization, a massive failure that leaves iPhones and Android mobiles open to tracking."

Steve: Yeah, yeah. And it has been noted that it isn't a huge win, but it's better than nothing. And I just say, you know, why not do it? They're also removing in Q easy access to network data. There was the /proc/net function that just...

Leo: "U.S. Naval Academy researchers report they were able to track 100% of devices using randomization, regardless of manufacturer, by exploiting a previously unknown flaw in the way wireless chipsets handle low-level control frames." So it just doesn't work.

Steve: Yeah, thanks anyway, yes. So anyway, the access to /proc/net is now being restricted. It represented very low-hanging fruit used by some data harvesters to access information about the device's network state, and it's been removed. And similarly, easy access to device details is being curtailed. Starting with Q, Google will require app developers to request a special permission before they can access what Google calls non-resettable device identifiers such as the IMEI and the serial number. So much as the app has to ask for permission for this or that sort of granular feature, apps will need to be asking for this kind of device identification that is not user resettable, starting with Q. So it's like, hey, this all sounds good.

And, finally, more location, or more control over location data. Android Q will receive a new permissions pop-up, kind of the thing that we're used to seeing from iOS devices, asking about location data. Beginning with Q, users will be able to give apps access to location data all the time, or only when the app is in focus, you know, in the foreground. And of course one of the features that I like about iOS is after some length of time it'll come back and say, hey, you know, this app still has location data access. Do you want it to keep it? Which I think is really a nifty feature. It's like, oh, yeah, forgot to turn that off. And then, you know, you have the opportunity to. So bravo on the Android front for being a little more privacy-forward.

Okay, Leo. Here's one that really surprised me. I first encountered the headline, I thought, wow, how cool. This really is a new Microsoft. The headline, which is pretty much repeated by all the tech press, is along the lines of "Microsoft releases Application Guard extensions for Chrome and Firefox." And it's like, what?

So the coverage across the tech press begins with things like "Microsoft has released browser extensions, one for Chrome and another for Firefox, which port the Windows Defender Application Guard technology from Edge to Chrome and Firefox." And I'm like, wow. So the articles typically say: "The extensions only work for Chrome and Firefox running on current Windows Insider builds [okay], but are expected to work with the upcoming Windows 10 stable release, 19H1, scheduled for release later this spring." So, yeah, Windows 10 will get this.

"The Windows Defender Application Guard technology is a relatively new Windows Defender security feature that until now has only been available to Edge users." Right. So I'm thinking, wow, like Microsoft is going to fix Chrome and Firefox browsers to protect their users on Windows 10 from malicious web content. How amazing is that? But then, reading into this a bit further, you encounter: "When using Chrome or Firefox, administrators can establish a list of trusted websites and local resources that the user can access within those browsers." Wait, what?

"But when a user of Chrome or Firefox attempts to visit any URL not on the trusted list, Windows Defender Application Guard comes into effect by launching a sandboxed session of Edge - within a Hyper-V-enabled container - where the untrusted website will be loaded into a safe environment within the Edge browser and isolated from the rest of the underlying operating system." I had to read that several times to be sure I wasn't missing something.

Microsoft's own blog posting last Friday, March 15th, 2019, 2:02 p.m., was titled: "Announcing Windows 10 Insider Preview Build 18358." And it has a number of sections. Scrolling down you get to the section about this titled "Windows Defender Application Guard as browser extensions in Google Chrome and Mozilla Firefox." And then, again, no wonder the press was confused. They said: "To extend our container technology to other browsers and provide customers with a comprehensive solution to isolate potential browser-based attacks, we have designed and developed Windows Defender Application Guard extensions for Google Chrome and Mozilla Firefox. This way, any potential attack won't be able to reach and grab the user's data, or plant malware on any local operating system.

"Here's how it works," says Microsoft. "The extensions for Google Chrome and Mozilla Firefox automatically redirect untrusted navigations to Windows Defender Application

Guard for Microsoft Edge. The extension relies on a native application that we've built to support the communication between the browser and the device's Application Guard settings."

Okay, now in other words - I'm breaking from Microsoft's announcement. In other words, Windows Defender Application Guard knows that the world is a scary place. And should you attempt to venture out there with Chrome or Firefox, this nifty new web browser extension will jump in to protect you from your wayward wanderings, taking you instead into Microsoft's proprietary Hyper-V VM where, from the safety of their Edge browser, you can peer out into the gloom which is the Internet. How very thoughtful of Microsoft. It's like, okay. What?

Then they said: "When users navigate to a site, the extension checks the URL against a list of trusted sites defined by enterprise administrators. If the site is determined to be untrusted, the user is redirected to an isolated Microsoft web session." Oh, Leo, I tried to get us a screenshot of it, but it was small and blurry, and I couldn't find a full-size one. When it first comes up it says, "Why am I here?"

Leo: Wow. If it can answer that, I want it.

Steve: It's like, what happened? I was using Firefox or Chrome, and now I'm in Edge.

Leo: Now you're in Edge. This is not nice.

Steve: Oh, my goodness.

Leo: I mean, maybe their motivations are pure. But I'm not using Edge for a reason.

Steve: Right, exactly. And, wow. And then they said: "In the isolated Microsoft Edge session, the user can freely navigate to any site that has not been explicitly defined as trusted by their organization without any risk to the rest of system." Oh, and then, with our upcoming, but not quite ready yet, dynamic switching capability, if the user tries to go to a trusted site while in an isolated Microsoft Edge session, the user is taken back to the default browser. So eventually they'll put you back to where you came from, Firefox or Chrome, if you happen to go back to a URL that we've decided you should trust. How thoughtful. Unbelievable. You know, if they would just spend their time fixing Windows bugs. Why not just fix Windows instead of, like, all of these shenanigans?

Leo: Have Mozilla or Google responded?

Steve: No, but I looked. The extensions are real.

Leo: Yeah, they're on the store, I mean, they could knock them off if they wanted to.

Steve: Yeah, it's a Google Chrome extension. Oh, but the reviews are pretty funny because I read a couple. One guy said, "Okay, this installs a button that takes you to

Edge in protected mode. I guess that's good. Except it's monitoring in the meantime everywhere you go to see if it's good or not, and it's accumulating data which you are unable to disable. So no, thank you." Why would anybody put this, install this in their Firefox or Chrome?

Leo: Well, because you might get the impression it's somehow safer and sandboxing you and stuff like that.

Steve: Well, yeah, because you don't get to use...

Leo: Is it?

Steve: Yeah, because it won't let you use Firefox or Chrome.

Leo: That's better. Only from Microsoft's point of view, but okay.

Steve: Wow. Wow. So just a real quick follow-up on our discussion of the Spoiler vulnerability. AMD stepped up and confirmed that, as far as they understand it, their architecture is not subject to this Spoiler-based amplification of the strength of attacks like Rowhammer.

They said: "We are aware of the report of a new security exploit called Spoiler which can gain access to partial address information during load operations. We believe that our products are not susceptible to this issue because of our unique processor architecture. The Spoiler exploit can gain access to partial address information above address bit 11 during load operations. We believe that our products are not susceptible to this issue because AMD processors do not use partial address matches above address bit 11 when resolving load conflicts."

So essentially that reduces the issue that Intel has on every one of their core processors from Core 1 on, to the essence of what Spoiler is about. So I did want to just confirm, for anybody using AMD stuff that, indeed, just not going to be a concern there.

Lawrence in Philadelphia, his subject was "Making Windows 10 Usable." He said: "Steve, I've heard you mention a few times that, in making peace with Windows 10, you've done a bunch of things to strip out the junk and make it into a usable operating system." And I just will say, boy, you install a new version of Windows 10 Pro, it's unbelievable. It's truly unbelievable. Anyway.

He says: "I would love some guidance on how to make it actually functional. Would you share some instructions on the show or provide some sort of checklist of what steps need to be taken? I dream of some tool like your Never10 program that simply fixes everything with one click. But I suspect that this is not so easy, and I know you're tied up with new versions of SpinRite and SQRL. Thanks for all your hard work and dedication."

So I just wanted to mention this because I've received many versions of this, since I've several times mentioned this idea of having wrestled 10 down to the ground. Maybe that's a good name, Wrestle10. Anyway. Or Tame10. Anyway, Never10 is at three million downloads now. So it's been a great success. And I don't know why anybody's downloading it now, I mean, it's still downloaded, I think just because it's there. I'm certainly not going to do it anytime soon. I could foresee a point in time where, if I've got a version of SpinRite that's out for testing, and I'm waiting to get feedback from a group of people who are playing with it to see what they think, it wouldn't take that long. But I'm not going to do it until then.

And really I didn't intend to do Never10, except that there were a couple not-well-written attempts at it; and I thought, okay, I just, you know, this is too important not to do. And since Windows 10 is apparently going to be with us forever, as it continues to morph, and Microsoft shows no sign of de-cartoonizing it, then I think something like that has to happen. And I just noted, while I was checking to see what the download count was for Never10, and I noticed that still in first place is GRC's DNS Benchmark. It's at 4.7 million downloads and getting more than 2,000 a day. I mean, tell you, I mean, people love benchmarks. You give people a good benchmark, they just, you know, especially one like this, that allows people to see how stuff is working, and they want their systems to go faster. So that was a win.

Oh. And a SpinRite user wrote, at the end of a longer note about something else, he says: "Please don't put off native USB support to later versions." So I wanted to explain that. It's not that I'm going to put it off to later versions. My intention is to, in every way I can, make up for the fact that it's taken me this long to get back, or will have taken me this long when I do, to get back to SpinRite. So my plan is, because, for example, I mean, there was a test version I had that people were using, I mean, it wasn't functional, it was just code. But people were playing with it when I stopped work on it in order to get SQRL done.

And so my plan is, as I have working code, it will be possible for all owners of SpinRite 6 to use their access to the SpinRite delivery, the SpinRite product delivery system to obtain whatever I have at the time. Which essentially means stuff that won't hurt you, that I'm sure won't hurt you, but I haven't finished yet with. But works. But does stuff. So, and I will at some point declare benchmarks. And so the point one benchmark will be SpinRite screamingly fast, running on, like, someone said he had a 14TB drive. And it's like, whoa. Okay, well, no, so that would take 28 hours as I benchmarked it last with this next version of SpinRite that is the bare metal 32MB buffer AHCI hardware interfacing screaming blizzard version. I want to put a point one on that and make it official.

But my point is that I'm doing that because that's the most important thing for me to do. USB support will then immediately start to happen, but I can only do one thing at a time. And so my intention is to be a bit of a production line and put point two out, which will be USB, and then point three, which will be whatever else it needs. And so the idea is I'm not going to, rather than releasing nothing until I am done with 6, I'm going to serialize them and even make pre-point release beta code available to people who want to play with it. And I know our listeners. I get feedback from them all the time. I know people want it, like, the first moment there's something that they can use. So I'm going to accommodate that. So that's the plan.

Open source eVoting: This was a really nice lengthy article in Motherboard, and so I've got the link at the top of my coverage of this for anyone who wants to dig in deeper. But I'm going to encapsulate this from having edited and excerpted from Motherboard's much longer coverage. The headline on Motherboard read: "DARPA Is Building a $10 Million, Open Source, Secure Voting System." Okay. Well, there's a lot of problems with that headline. But I'll explain where they got the headline. I mean, the people who do the headline skim the article, I guess, and come up with something that people will want to read.

The system will be fully open source and designed with newly developed secure hardware to make the system not only impervious to certain kinds of hacking, but also allow voters - and this stuff is so cool, I'll get to the details of this in a minute - to verify that their votes were recorded accurately. We're going to see some really cool new crypto involved, as you'll see.

"For years," Motherboard writes, "security professionals and election integrity activists have been pushing voting machine vendors to build more secure and verifiable election systems." And I would say open. It's just, it's nuts that Diebold could possibly sell to anyone a box under the "just trust us." I mean, okay, what's that, JTU? That is the exact reverse of TNO. Just Trust Us versus Trust No One. That's just - how did that happen? I don't know how that happened. But it's the world we're in right now.

So, Motherboard says, "so the voters and candidates can be assured election outcomes have not been manipulated. Now," Motherboard writes, "thanks to a new $10 million contract" - so that's where this $10 million number came from - "DARPA has launched to design and build a secure voting system that hopes to be impervious to hacking." And I would argue this has the chance to happen, due to the way they're doing this. "This first of its kind system will be designed by an Oregon-based firm called Galois" - G-A-L-O-I-S, which is a math term used in crypto - "a longtime government contractor with experience in designing secure and verifiable systems. The system will use fully open source voting software instead of the closed proprietary software currently used in the vast majority of voting machines, which no one outside of voting machine testing labs can examine."

And we have seen what a piss-poor job they've done because voting machines, when they are available, and of course the reaction as we've talked about it of the voting machine companies is to buy them all off of eBay so that they can't be found for hacker competitions during Def Con. So it's like, no, don't look, we're not going to let you look at our machines. And whenever anybody has, they've cut into them like Swiss cheese.

So: "More importantly, these next-generation machines will be built on open source hardware made from secure designs and techniques developed over the last year as part of a special program at DARPA. The voting system will also be designed to create fully verifiable and transparent results" - that is, output - "that the voters don't have to blindly trust that the machines and election officials delivered correct results." Get to more of that in a second.

"But DARPA and Galois won't be asking people to blindly trust that their voting systems are secure, as voting machine vendors currently do. Instead, they'll be publishing source code for the software online and bringing prototypes of the systems to the Def Con Voting Village this summer and next, so that hackers and researchers will be able to freely examine the systems themselves and conduct penetration tests to gauge their security." So they will have working systems and the source code to read through and try to find problems. "They'll also be working with a number of" - "they" meaning DARPA and Galois - "working with a number of university teams over the next year to have them examine the systems in formal test environments."

Linton Salmon is the program director for DARPA's Microsystems Technology Office which is overseeing the project. In a phone call he told Motherboard: "Def Con is great, but hackers there will not give us as much technical detail as we want about problems they find in the systems. Universities will give us all the information, but we don't have as many people or as high visibility when we do it with universities." So they're going to use both is his point.

The systems Galois designs won't be available themselves directly for sale, but the prototypes it creates will be available for existing voting machine vendors or others to freely adopt and customize without costly licensing fees or the millions of dollars it would take to research and develop a secure system from scratch. So they're creating exactly what we want, a secure hardware and software open standard which then existing voting machine manufacturers will be able to adopt for free to turn into commercial machines.

And then states will be able to say to Diebold, we'll buy your machine as long as our purchasers can verify, and you certify, it is compliant with the gold standard, which will then be available and testable and verifiable.

So that's where we're going to be going. And it's perfect. Linton said: "We will not have a voting machine that we can deploy. That's not what we do. We will show a methodology that could be used by others to build a voting system that's completely secure."

Joe Kiniry [K-I-N-I-R-Y] is the principal scientist at Galois, who's leading the project at his company. He said that Galois will design two basic voting machine types. So here's some juicy details. The first will be a ballot-marking device that uses a touchscreen for voters to make their selections. The system won't tabulate votes. Instead, it will print out a paper ballot marked with the voter's choices, so voters can review them before depositing them into an optical scan machine that tabulates the votes. Galois will bring this system to Def Con this year.

Many current ballot-marking systems on the market today have been criticized by security professionals because they print bar codes on the ballot that the scanner can read instead of the human-readable portion voters review. Someone could subvert the bar code to say one thing, while the human-readable portion says something else. Kiniry said they're aiming to design their system without barcodes. So the point being, what the scanner scans and tabulates from is human readable so that there's no question. The optical scan system - okay. So that's the thing that creates the thing to be scanned.

Part two, the optical-scan system, will print a receipt with a cryptographic representation of the voter's choices. After the election, the cryptographic values for all ballots will be published on a website, where voters can individually verify that their ballot and votes are among those present and counted. Kiniry said: "That receipt will not permit you to prove anything about how you voted, but it permits you to prove that the system accurately captured your intent and that your vote is in the final tally."

Members of the public will be able to use the cryptographic values to independently tally the votes to verify the election results so that tabulating the votes isn't a closed process solely in the hands of election officials. In other words, everything gets made public and published. Kiniry said: "Any organization interested in verifying the election results that hires a competent software engineer can write their own tabulator. We fully expect that Common Cause, League of Women Voters, and the political parties will all have their own tabulators and verifiers." The second system Galois plans to build is the optical scan system that reads paper ballots marked by voters by hand. They'll bring that system to Def Con next year.

So there is a bunch of, I mean, all of that sounds right. Some people clearly thought about this. They're like, how to be completely open, how to allow an individual voter the satisfaction of knowing that they're using an academically scrutinized open system where after the fact they're able to use a website to cryptographically verify that their individual vote was captured and is part of the final tally that this tabulator system uses. I think they nailed it.

And so it looks like a couple years from because they're going to have this first system at this summer and then the paper ballot tabulator next summer. I mean, I hope that the voting machine vendors read the handwriting on the walls, recognizing that this is out of their hands. This, you know, they had their day. And then to immediately get up to speed on this new technology so they can be first to market, early to market, with the system as it gets finalized because this is a beautiful solution. Bravo.

Leo: Nice, yeah, awesome.

Steve: Yeah. Very, very cool.

Leo: And I guess one more little story that's one of those object lessons. Did you see this?

Steve: Yes.

Leo: MySpace.

Steve: How big was their loss?

Leo: They lost everything that had been uploaded to MySpace from the years 2003 to 2015 because the hard drive failed, and they didn't have a backup, in short. Significant.

Steve: It really is out in space now.

Leo: Yeah. MySpace is in space.

Steve: Wow.

Leo: It's just it's hard to believe, since best practices are well known, and frankly most of our listeners perform it every day at home, that MySpace could have lost it all. But they did. And I hope that if you uploaded your music to MySpace, you kept a copy for yourself.

Steve: I have Drobos on multiple sites and their dynamic real-time change file synchronization running. I mean, it's like, nothing could - I mean, it's like this - it's not hard.

Leo: It's not hard.

Steve: It's not expensive, no.

Leo: It's not hard.

Steve: It just takes doing it. Whoa. Yeah, I caught the headline. I didn't have a chance to dig in. So, wow.

Leo: Unbelievable. I'm not sure how much data it was. I seem to remember it was something like 48TB. But I'm not - I don't really know for sure. But, doh.

Steve: You pressed what button?

Leo: They said for a long time they've been saying, uh, we've been working on it. But eventually they admitted we just - we lost it. It's all gone. It's gone.

Steve: And what does that mean for them as a service? I mean, that means they have the last four years.

Leo: Yeah.

Steve: The most recent is probably what makes the most - is the most valuable. But yikes. If nothing else, it's an embarrassment. And you've got to wonder what that means about their IT.

Leo: If I could show you this email, I would. It says - it's from the Data Privacy Officer at MySpace. "Hello, Austin. Yes, this is true. Due to a server migration, files were corrupted and unable to be transferred over to our updated site. There is no way to recover the lost data. Thanks, MySpace." Oh, I shouldn't laugh. That's just kind of the end of the line for MySpace. But, you know, today MySpace, tomorrow Facebook or Google or anywhere. Your data is your data. Don't let anybody else keep track of it.

Steve: Yup.

Leo: You, Steve Gibson, have done many, many versions of this show backed up in many, many places. It shall never be lost. It shall continue into the heat death of the Sun. And then after that we can't make any promises. All the shows, they are at Security Now!'s website, GRC.com. That's Steve's website, the Gibson Research Corporation. He has audio, and he has transcriptions. So if you really wanted to be safe, what you would do is download all 706 episodes...

Steve: Oh, people do.

Leo: ...and print out the transcriptions.

Steve: Yup.

Leo: Put them in a vault.

Steve: Yup.

Leo: No, we've got - I think that's the beauty of this is that there are many people have every show, so we don't have to worry about that. But it is a good place to get it. While you're there, pick up a copy of SpinRite, the world's best hard drive maintenance and recovery utility, and read up on all the other stuff Steve's up to. He is a polymath, and there is stuff in every possible subject, fascinating reading. It's one of those websites you start and then eight hours later you realize, oh.

Steve: Where did the day go?

Leo: Where did the day go?

Steve: What did I get done?

Leo: What happened here? GRC.com. We have all 706 episodes, actually 709, I don't know what happened, but there was a mistake at the factory. Patrick's telling me there's 709. Don't tell Steve that. He'll end this three episodes earlier. We have seven, maybe 600. I don't know. We have a few. Couple of dozen, anyway, at TWiT.tv/sn.

Steve: With a short serial number.

Leo: Yes, with a number. Zero through 12.

Steve: Yes, with a short serial number. A 63-bit serial number.

Leo: Yeah, that's right. Not insecure, just not numbered properly. Yes, that's true. Maybe go to GRC.com. He's got a 128-bit serial number. And you know how important that is. You can watch us do the show live, 1:30 Pacific, 4:30 Eastern, 20:30 UTC, every Tuesday at TWiT.tv/live. Watch or listen. We have live audio and video streams. If you do that, irc.twit.tv is the place to chat along with Steve. And, let's see, I guess that's about it. Nothing more to say except thanks for joining us. Subscribe to the show. Get every episode. And we'll see you next time on Security Now!.

Steve: Thanks, Leo.