The Facebook Breach

Description: This week we discuss yet another treat from Cloudflare, the growing legislative battle over Net Neutrality, the rise of Python malware, Cisco's update report on the VPNFilter malware, still more Chrome controversy and some placating, the rapid exploitation of zero-day vulnerabilities, the first UEFI rootkit found in the wild, another new botnet discovery, the danger of the RDP protocol, a nasty website browser trick and how to thwart it, a quick update on recent nonfiction and science fiction, and then a look into the recent massive 50 million account Facebook security breach.

High quality (64 kbps) mp3 audio file URL: http://media.GRC.com/sn/SN-683.mp3
Quarter size (16 kbps) mp3 audio file URL: http://media.GRC.com/sn/sn-683-lq.mp3

SHOW TEASE: It's time for Security Now!. Steve Gibson is here. We've got lots to talk about. He's going to dissect the Facebook breach. Good news. I think my sense is he's not as worried about it as you might be reading the headlines. We'll also talk about some much more serious problems infecting routers and how hackers work. It's all coming up next on Security Now!.

Leo Laporte: This is Security Now! with Steve Gibson, Episode 683, recorded Tuesday, October 2nd, 2018: The Facebook Breach.

It's time for Security Now!, yes, indeed. You've been waiting all week. Your long delay, your long wait is over. He's here, everybody. Steve Gibson.

Steve Gibson: He's back. Ah, yes.

Leo: Would you ever like to do this live in a big auditorium in front of a studio audience kind of thing?

Steve: We could. I mean, we've sort of done it. I guess we haven't in front of a studio audience. I mean, if it were topics, I mean - okay. So the problem is, as you know, I like to dig into these stories and have, like, well...

Leo: And he often has his eyes closed during his speaking. I'll be honest. Right? Am I wrong?

Steve: Yes, I do, actually.

Leo: You close your eyes.

Steve: So I can sort of see the terrain.

Leo: It's not really a performance - you never wanted video, for crying out loud. It's not really a performance. It is, but it's more of an audio performance. I still think we could do this in front of a live - I bet you there'd be a large audience if we decided to do that.

Steve: Well, like in what venue would that work?

Leo: I don't know. I don't know why I'm asking this. I mean, it's not like it's up on the - it's on the table or anything.

Steve: Yeah. Certainly when you and I and Mark Thompson all congregated at the Gnomedexes, an audience forum...

Leo: Wasn't that fun?

Steve: Yeah.

Leo: I'll never forget that.

Steve: That was fun. And I was the keynote the first year, and I don't remember what the second year was.

Leo: How many years ago? That was so long ago.

Steve: We did it for a couple years, yeah. So, I mean, that was fun. And of course I'm just about, I mean, I'm just bursting with SQRL stuff.

Leo: That doesn't sound good. He's talking about some software he's been writing, folks.

Steve: And it turns out, well, actually a solution to one of the big problems that Facebook just faced because one of the consequences of the breach was that the Login with Facebook, this whole OAuth kludge, which is really what it is, I mean, if you think eInk is obsolete, this whole login with another site is, I mean, it has always been a privacy disaster and a kludge.

Leo: Well, not only that, besides all of that, it's a problem for me because I deleted my Facebook account, and now all these places I was logged into I no longer am - I don't even have a - because a lot of times they don't even have a password.

Steve: Yeah. Yeah.

Leo: So I don't know what to do in that case.

Steve: Anyway, so...

Leo: SQRL's going to fix that.

Steve: It does fix that. And it is decentralized completely so there's no single point of failure and blah blah blah. The point that I was going to make was that I could prance around onstage and talk about SQRL till the cows come home, I mean, till the SQRLs all leave the building. So there will be opportunities, I'm sure, to...

Leo: We should do that. That's a good idea.

Steve: To address a crowd. Okay. So the Facebook breach for Security Now! Episode 683, the day before the much-anticipated Presidential Alert for which there is now a lawsuit, Leo, trying to prevent our President from sending the alert. And I think they think it's a tweet that he can send from his phone. It's like, no, that's not what this is, folks. So anyway, we'll talk about that again just as a final reminder. But we're going to talk about yet another treat from Cloudflare, which I just couldn't resist making this our Picture of the Week because it's just cool. I just love these guys. They keep doing good stuff.

We've got the growing legislative battle over Net Neutrality because now the Department of Justice has sued California. The rise of Python malware, which we've touched on before, but now we've got numbers and charts and things. Cisco's third and final update on their VPNFilter malware which they found, and we talked about first in May, and then again in June, and now they're done. We've got still more Chrome browser controversy and some placating from Google. The rapid exploitation of zero-day vulnerabilities, an example of. The first UEFI rootkit found in the wild. Yet another new botnet has been discovered.

A warning about the danger of the Remote Desktop Protocol. I wonder if that means that saying RDP Protocol is redundant. I guess that would be, yeah. Anyway, a nasty web browser trick - that I've never stepped into, Leo, but it occurred to me it makes a useful public service announcement - and how to thwart it. A quick update. We haven't talked about sci-fi for quite a while, but there are some events that have happened in the world of science fiction, but even nonfiction. I just am hours away from finishing the "Masters of Doom," which was really a fun read.

Leo: Oh, yeah, what a good book, yeah.

Steve: And then we're going to take a look at this recent massive, well, 50 or 90 or maybe more million account Facebook security breach, how a very clever leveraging of a subtle mistake was turned into a mess for Facebook, and what lessons we can learn from it. So I think another great podcast for our listeners.

Leo: A whole bunch of stuff to talk about. How exciting.

Steve: So I just threw this in because I thought it was fun. We're often talking about Cloudflare because of their leading-edge security implementations.

Leo: Love them, love them, yeah.

Steve: Yes, we do. They're often trying things before everybody else. We were just talking about them last week in this context. They're clearly always asking themselves the question how can we make our services better, rather than just sitting around and not doing anything. And they often answer with security and privacy improvements. In this case, my sense is that they were looking around for, like, okay, what more can we offer? And they had a service that was previously only available to their enterprise customers, which they have now just made available to everybody.

And you can have a Cloudflare presence for free. They've got millions of customers who are using them as a free service. What they've just done is to add for their customers at-cost domain registration. Which I just, you know, at the moment I saw it I thought, yeah. It's like, that's neat.

Now, I registered GRC.com, I think it was six months after the domain Microsoft.com was registered, so back in the...

Leo: Wow.

Steve: Back in the early days, yeah. And of course back then Network Solutions was the registrar. That's who you used. And inertia and loyalty, I just, you know, I stayed with them for a long time. And our listeners who have been following along know that I finally thought, okay, no. Every time I renewed one of my domains, and I have a whole bunch that are not known to the public because they're just like ideas for products, you know, you grab the domain name for an idea that you have in case you're going to develop it in the future. Like, for example, CryptoLink. I have CryptoLink I think in every TLD that is major because for a while I thought I was going to do an easy-to-use, really, really secure VPN, until I started worrying about the federal government saying, okay, encryption is - we're not going to let you do that. But I still have those.

And so I was constantly renewing these things over at Network Solutions for $35. But what really annoying me was all the upselling. I had to click through no I don't want this, no I don't want this, no I don't want this, all of these extra things in order just to say, yes, I'll renew for another year. Which is why I went to Hover, where I'm very happy. Now it's $15, and I get all this other free stuff.

I note, however, that Cloudflare is also giving all that way. For example, the WHOIS registration is blinded by default, and so you get the privacy. And, boy, is that nice because I refuse to pay to have my WHOIS information made private. That's just

annoying. But I was curious to see if Network Solutions was still doing that. And they give it to you free for a month, and then it's $2 per month after that.

Leo: That's ridiculous.

Steve: It's ridiculous. So that's, what, $24 just to have your WHOIS information...

Leo: Just to have an unlisted number, that's all. I mean, basically.

Steve: Yes. And let me tell you, you get spam. I mean, I see incoming stuff because I have a different email account for that. And it's just it goes into a bit bucket which is overflowing. Anyway, it's so annoying. So I just wanted to give a tip of the hat and an acknowledgement to Cloudflare and to any of our listeners who may be Cloudflare users. I imagine you're probably on an email list somewhere, so you already know this. But any domains you want to register you can now do so for at cost. Oh, and by the way, cost. The actual registrar fee for the .com, get this, $7.85.

Leo: Okay.

Steve: $7.85.

Leo: That's really good.

Steve: ICANN has their administration fee of $0.18. So you add those together, and so Cloudflare is $8.03.

Leo: Very good deal.

Steve: Exactly the cost. .Net costs them more. But the point is they're zero markup, so $9.95 for .net, $11.02 for .info, and $10.11 for .org. So, I mean, you can't - clearly they can only do this because they don't need to make money there. They're doing other things.

Leo: Would you have to be a customer of theirs to do this?

Steve: Yes, yes. Yeah, so you have to be...

Leo: Even a free customer, because they have a free tier, but you still have to be a customer.

Steve: Exactly. So you could be a free tier user, and now your domain name, you're also saving money on your domain name.

Leo: And we should mention they're a sponsor.

Steve: Ah. I didn't know if they were. Good.

Leo: Yeah, yeah.

Steve: So, cool. So I'm not saying this because they're a sponsor, I'm just saying it because we love them.

Leo: No, I know that. But, you know, we always disclaim when we talk about sponsors.

Steve: Yeah, good, yeah.

Leo: Just so people can make that distinction.

Steve: So, okay. The U.S. Department of Justice Sunday, when California Governor Jerry Brown signed the Senate Bill SB-822 into law, immediately filed a lawsuit against California.

Leo: [Heavy sigh]

Steve: I know.

Leo: They're all for states' rights until a state does something they don't like, notice.

Steve: Exactly. We want decentralized government except where we don't.

Leo: Right.

Steve: So this bill, which we now have in California, as a law in California, prevents Internet Service Providers from blocking, throttling, or discriminating against any Internet content. And I was thinking, Leo...

Leo: By the way. If the Justice Department doesn't like that and is suing them, that's basically the Justice Department saying, oh, no, we want to have ISPs have the right to block, throttle, and zero rate.

Steve: Yes, yes.

Leo: They're putting the lie to the fact, oh, we don't have to have a regulation because they would never do that. Well, if you're not worried about it, why are you suing? You want to preserve their right to block, throttle, and zero rate.

Steve: Yes. In fact, okay. So upon signing, the DOJ responded. FCC Chairman Ajit Pai said, quote: "I am pleased the Department of Justice has filed this suit. The Internet is an inherently interstate information service. As such, only the federal government can set policy in this area. And the U.S. Court of Appeals for the Eighth Circuit recently affirmed that state regulation of information services is preempted by federal law."

Leo: I think, by the way, they're going to win because that is...

Steve: Except they over - I know.

Leo: Interstate commerce is protected federal arena.

Steve: Yes, yes. Well, and what they did was they repealed, the FCC repealed the open Internet order legislation which was passed in 2015. So they un-lawed it.

Leo: Yeah, and they want to keep it un-lawed.

Steve: And now they're unhappy that we're trying to re-law it.

Leo: Keep that un-lawed.

Steve: And Katharine Trendacosta, who's a policy analyst at the Electronic Frontier Foundation, argued that the California law was specifically written to be within state power. She said, quote...

Leo: Oh, interesting.

Steve: Yes. "The argument that the FCC preempted states acting to protect the Net Neutrality within their borders doesn't hold a lot of water when the FCC gave up jurisdiction of this area in the so-called 'Restoring Internet Freedom Order.'" She says: "You can't throw up your hands, walk away, and say it's not your problem anymore, only to also say no one else can rush in to solve that problem. You gave up that right when you walked away," she said.

Leo: It's just appalling. I mean, it's a total admission that the whole point of this is to protect the rights of ISPs to discriminate. That's the whole point of it.

Steve: Yes. And the other thing that I thought was interesting was that, as I was assembling this, I thought, remember how once upon a time everybody was confused about what Net Neutrality meant? Now it's been completely well defined. I mean, we did

go through several years of, okay, well, now, here's what this means. And it seemed like a really bad term. In fact, I remember when we first had the iPod. It was like, what? iPod, what a horrible name. But now it's just, you know...

Leo: You get used to it, yeah.

Steve: And the same thing for iPad. What? And then the Mini Pad and the Maxi Pad and all that. And it's like, no. It just dropped into our culture. And so now, yeah, Net Neutrality. We all know what that is, and we all know that we want it. Except, well, not all of us, I guess. So anyway, so - oh, oh, oh. And we're not alone. It's worth noting that California is far from alone in taking independent legislative action.

The National Conference of State Legislatures indicates that 30 states, three zero, 30 states have introduced over 72 bills requiring Internet Service Providers to ensure and adhere to various Net Neutrality principles. And so far governors of six states, including Hawaii, New Jersey, New York, and Vermont, have signed executive orders about Net Neutrality, while three other states, Oregon, Vermont and Washington, have enacted, as has California now, Net Neutrality legislation. So, I mean, it's...

Leo: Play devil's advocate, though. You don't want a patchwork of laws governing something like the Internet; right?

Steve: No, no, no. I completely agree with you. I mean, this is the worst case that has resulted is every ISP - well, actually all an ISP has to do is be net neutral.

Leo: Well, that's the other argument is that, well, we're not asking that much. Just don't be a jerk.

Steve: Yes, exactly.

Leo: Basically.

Steve: It's not like in some places you have to do it and some places you don't.

Leo: Right, right, right.

Steve: If you just don't do it anywhere...

Leo: Don't do it, you'll be okay.

Steve: ...then everybody's happy.

Leo: Yeah. I love it that the complete and utter corruption of the FCC and their motivation is now blatantly clear because they got a lot of money from the

telecommunications companies, and they're now saying to the DOJ, you can't let this happen. We paid a lot of money for this deregulation.

Steve: Well, and isn't Ajit Pai an ex-lobbyist?

Leo: Of course he is. Of course, remember, Tom Wheeler also was. And he, I think thanks in great part to all the comments that were put up on the FCC site, he changed his mind. Ajit Pai just managed to bring the site down so he didn't have to see those comments.

Steve: That's right. So ThreatList had an interesting article, and I knew you would love this chart on the next page, Leo, because it's language popularity over time.

Leo: I do love this stuff.

Steve: Hackers have turned to Python as the attack coding language of choice. Now, it's not surprising because it's not like Python is the attack language, it's Python is growing to become the all-purpose coding language of choice.

Leo: Absolutely, yeah. It's really interesting.

Steve: Yeah, and one of those purposes, for better or mostly for worse in this case, is malware. Today - and what's interesting is stats from GitHub. More than one out of every five GitHub repositories which contain an attack tool or a proof of concept has its code authored in Python. So, well, and not just on GitHub. This summer it was Imperva who did a study who showed that between June and mid-September Python-based tools were used in up to 77% of attacks against sites seen in their telemetry.

Leo: Because it's easy to code in.

Steve: Well, and, yes, and as we talked about last week, and we'll be talking about a little bit later here this week, you can cross-compile it to multiple platforms. So we're no longer in a world where everything is x86. We've got ARM chips in routers, and some are 32 bits, and some are 64. And some are little endian, and some are big endian. And there's many, many, you know, we're in a very heterogeneous environment in terms of what the malware wants to infect.

And so the malware said, oh okay. We're going to write this in a high-level language. Gee, which one should we choose? And in fact Go is another one which was recently chosen by something known as the Torii malware that we'll be talking about in a second. But anyway, I just thought it was interesting that more than three out of four, more than three quarters of the attacks in the summer through the early fall have been Python-based agents that are doing the attacks.

And the chart is really interesting. For those who are hearing audio, interestingly, Java has, I mean, if you look at the Java line, it echoes the Python line. I mean, they were both in, well, way back in '98 - the chart goes all the way back to 1988. Back then pretty much it looks like C was the - was it C? The super dark one. Yeah, C. C naturally is the

old dog. But Java jumped way up from '98 to '03 into first place. It still was first place in '08 and second place in 2013 and back in first place now in 2018. Python has always been a little lower down, but shows the same trajectory. And, let's see, what's the expense? Oh, yeah, Fortran.

Leo: And Ada, Fortran, and Lisp seem to be plummeting precipitously.

Steve: Yeah. And it's interesting also that Objective C had a sharp rise five years ago, and then it's like sort of dropped back down. And Perl's been kind of cruising downward. "R" is not doing much. JavaScript's kind of holding steady. But for what it's worth, anyway, Java is in number one place right now. C, number two. C++, number three. Python, number four. And that's above C#, Visual Basic .NET, then JavaScript, PHP, Ruby, R, and Perl, counting down. So just sort of interesting to see how these things change over time.

And it's my intention, I think I've mentioned it before, because I have very ambitious plans for SpinRite 7, I intend for it to be a full rewrite. It'll use all of the architecture, the low-level stuff that I developed for the 6.1, 6.2, 6.3 series. But then I'm going to give it a whole new front end. And I plan to write that in Python because I want to be able to quickly implement awareness of file systems, and there's just no reason to write that stuff in assembly code. All of the low-level stuff will be that way because that's where I'm still most comfortable, and I want screaming speed without compromise. And it has to talk to the hardware intimately, which is not to say that you can't in Python, but assembly language is the natural language in which to do that. But I fully intend to use Python because I want to be able to author quickly and have it much more dynamic, just like the malware.

Okay. So in May of this year, earlier in May, we first introduced a confusingly titled malware. You know, it's better when it sounds bad, like Heartbleed or something, where it's like you don't want to have that. In this case, VPNFilter sounds like, oh, maybe I need to install that.

Leo: Sounds like a good tool, yeah.

Steve: I think I should filter my VPN. No. So this was named because it's in the path name of where this malware stores itself. And it was obviously named that to sort of look innocuous, like if you were you looking through your file system and saw a directory named VPNFilter, you'd go, oh, okay, I guess my VPN is being filtered, not realizing necessarily that it was evil. So Cisco's Talos Research found this in May and gave us the first report. And that's when we first talked about it with the surprise that over half a million diverse routers from Linksys, MikroTik, Netgear, and TP-Link have been found to be infected by this one strain, this new strain of malware. It's an IoT bot that is scanning for itself and is used for attacks and proxying and so forth.

And at the time they said they had not yet completed their research. But they felt, due to this discovery of how widely present this was - they found it in 54 different countries, though largely concentrated in Ukraine. They felt they needed to come forward and alert the public to what they had found so far. A month later we revisited it on the podcast in June with their update, that it was also they had since discovered it in ASUS, D-Link, Huawei, Ubiquiti, UPVEL, and ZTE routers. And I remember you and I looked down the list because I put it in the show notes: 75 makes and models of routers where this thing was present. Meaning that it was managing to get into and infecting a large diversity of routers.

Okay. So at the time we talked about it. You'll remember this, Leo, because it was downloading its second stage either from Photobucket or from a domain ToKnowAll.com. And it was hiding the IP address of its command-and-control server in the EXIF image metadata of images stored on Photobucket. So sort of a steganography kind of approach where, unless you had reverse-engineered this thing and saw it go reach out to Photobucket and then parse the EXIF image metadata and go, oh, it just extracted an IP address, which it then connected to to get instructions, you wouldn't know what it was doing.

So, okay. So probably one of the most disturbing things about this is that this is not coming from anybody in their mother's basement. They have tied this back to a major state actor. And so it's a little unsettling, and maybe that's one of the reasons that routers in Ukraine are the target, but it's disturbing that a major, you would hope a mature state would be making this sort of malicious software. But that's the world we're in.

Okay. So finally, last Wednesday, we're back for the final update from Cisco. They've finished their analysis. And just to paraphrase from the beginning of their update report, they said VPNFilter a multi-stage, modular framework that has infected hundreds of thousands of network devices across the globe is now known, now as of their third final report, to possess even greater capabilities. Cisco Talos recently discovered seven additional third-stage VPNFilter modules that add significant functionality to the malware, including an expanded ability to exploit endpoint devices from footholds on compromised network devices. The new functions also include data filtering and multiple encrypted tunneling capabilities to mask command-and-control and data exfiltration traffic.

They've been, as we know, researching VPNFilter for months. And they said: "As part of our continued investigation, we developed a technique to examine a key protocol used by MikroTik networking devices" - and we've often talked about this web-based configuration, this Windows-style configuration utility which unfortunately has turned out to be a vulnerability in MikroTik routers - which has allowed them to hunt for possible exploitation methods used by this threat actor.

So they've found, as they mentioned in their opening, additional Netfilter capabilities. The reason this took so long is that they had binaries for these things, but they then had to reverse-engineer them. And the binaries are just ones and zeroes, which they then have to decompile. And then there's no comments and easy clues in a well-written binary. It's just instructions. So you really have to spend some time studying this, which is what they've done.

They found these seven additional what they call "third-stage modules" which provide these greatly expanded capabilities which are present in VPNFilter. There's one they call "htpx" which redirects and inspects the contents of HTTP traffic which transit through their devices; "ndbr," which they describe as a multifunctional SSH utility; "nm," which allows network mapping activities to be conducted from the compromised devices. Netfilter is a denial of service utility. There's a port forwarding system that allows network traffic to be bounced off of these devices to other infrastructure. There's a "socks5proxy" which can be established on compromised devices, and a reverse-TCP VPN.

So their rather sobering conclusion was: "As a result of the capabilities we previously discovered in VPNFilter, coupled with our new findings, we now confirm that VPNFilter provides attackers all of the functionality required to leverage compromised network and storage devices to further pivot into and attack systems within the network environments that are being targeted."

So this is exactly what I've been talking about being glad we haven't really been seeing yet. Turns out this is exactly the behavior which is built into VPNFilter, which is I've been talking about how nice it is that, well, mostly these guys just want to run cryptomining, or they just want to use UPnP abuse to bounce packets off of routers in order to conceal their identity for distributed denial of service attacks, and how good it is that they're not really that interested with what's going on inside the network. Well, everything we just read and what Cisco has concluded is that that's exactly what these capabilities enable.

They continue, saying: "It also allows attackers to leverage their access to sensitive systems such as gateway and routing devices to perform activities such as network mapping and endpoint exploitation, network communications monitoring and traffic manipulation, among other serious threats." I mean, so this really does sound like state actor-based. This is not people who are wanting to rent out their DDoS botnets. This is organizations that know what MikroTik enterprise router they are on and are now going to turn around and look inside and cause some havoc.

They said: "Another dangerous capability provided by VPNFilter is the ability to turn compromised devices into proxies that could be leveraged to obfuscate the source of future, unrelated attacks by making it appear as if the attacks originate from networks previously compromised by VPNFilter. The sophisticated nature," they say, "of this framework further illustrates the advanced capabilities of the threat actors making use of it, as well as the need for organizations to deploy robust defensive architectures to combat threats such as VPNFilter."

And I would say, I mean, our takeaway, our listeners' call to action is absolutely make certain that the point of contact of both enterprise and personal networks which to say is typically some sort of router gear, absolutely make sure that it is secured, that it is running the latest updated firmware from the manufacturer, that it's from a reputable manufacturer that is maintaining the firmware and really carefully look at the Internet-facing attack surface to see what's there. It's just it's obvious that we're talking here is more than 500,000 instances of this kind of presence on publicly exposed nodes on the Internet.

And this is just this one malware. We know that in aggregate it's probably tens of millions of different types of malware that is coming and going. And as we'll see a little bit later, they're becoming increasingly good at being both multiplatform and really deeply persistent on the things that they infect. It's just crucial. And I guess it feels to me like it's so easy to, because we don't know otherwise, it's easy to assume that your node is secure. You have to look because what we always see is it's not until you look that you discover a problem.

That's why the current best advice for enterprise and, to a lesser degree, personal networks is not only having a firewall and/or NAT router that blocks traffic, but keeping an eye on the traffic inside your network. You know, you just talked about a sponsor, Leo, that is offering exactly that. You absolutely, you have to keep an eye on what's going on in your network. I'm continuously looking at the traffic here in my own home office environment, and I see traffic spikes from time to time, and I take a minute to go see what's going on. And it's like, okay, well, it's one of my many iOS devices decided it's time to update itself or something. Whatever. But it's nice to have an idea of what's going on. And unless you look, you don't know. And, boy, it's so easy just not to be, you know, just to be oblivious to what's happening.

Okay. So one more story, then we'll take our second break: Chrome back in the middle of controversy. Last week, I think it was Thursday - or, no, it was Monday. It was early in the week, but then Chrome responded by Thursday. Someone named Christoph Tavan noticed something about Chrome and tweeted, and set off a firestorm, which in this case I think was unwarranted because this is something that Google clearly knew and

deliberately chose to do and even documented, clearly, right at the site of where this happened. But in this current "everyone is piling on Chrome" environment, it went over less well than it might have normally, and Christoph got a lot more attention from this tweet. He observed and tweeted that after he had instructed Chrome to delete all of the cookies his Chrome browser had collected, it did that, except for Google's own cookies, which it retained.

Now, under Settings > Advanced > Clear browsing data, and I went there this morning just so I could talk about it authoritatively and make sure that it was still there, we're shown the number of things and the amount of space being consumed by all of our various browsing history, you know, caches of stuff and various site data, how many different sites have cookies. I noticed that in the screenshot he posted he had 1,157 different sites had left cookies in his browser that he was going to delete by clicking yes, get rid of all that. So on the detail line for cookies and other site data, Chrome clearly states parenthetically there: "You won't be signed out of your Google account."

Okay, well, now, maybe the problem was they weren't clear enough, or I don't know. We know that cookies are the mechanism by which browsers maintain state with the sites they visit. The web server gives the browser a cookie to return with all subsequent queries. So clearly what Chrome was doing was assuming users wanted to get rid of all their cookies except their Google account cookies. That's the assumption that got them in trouble in this case, even though "You won't be signed out of your Google account" means because that cookie gets to stay. Because otherwise you will be signed out of your Google account. Google won't know you. You'll have to sign in again, and then you'll get a cookie in order to be signed in.

So based upon the uproar caused by this misunderstanding, I guess what Google should apparently have offered was an additional checkbox underneath the cookie report which said - like a checkbox that would say "Also delete Google's own cookies and be signed out of all Google accounts and services." But they didn't do that.

So that leads us to Google's comprehensive posting last Thursday made by Zach Koch, who's the Chrome Product Manager. He first catches us up in his posting on recent events which I won't cover because we've been doing that the last couple weeks and just now. Finally he says: "Over the years we've received feedback" - and to their credit, I mean, again, anyone can make a mistake. What matters as we know is that Google and Chromium are listening. And they certainly have been. Well, except in the case of www, which is still going to get disappeared in the next version of Chrome.

But he says: "Over the years we've received feedback from users on shared devices that they were confused about Chrome's sign-in state. We think these UI changes help prevent users from inadvertently performing searches or navigating to websites that could be saved to a different user's synced account." And actually there was some report of that happening, too.

Anyway, he said: "We've heard and appreciate your feedback. We're going to make a few updates in the next release of Chrome" - that'll be v70, released in mid-October, so in a couple weeks - "to better communicate our changes and offer more control over the experience. We think sign-in consistency" - which is what they call this automatically have the Chrome browser sign in when you sign into a Google account. "We think sign-in consistency will help many of our users. We're adding a control that allows users to turn off linking web-based sign-in to browser-based sign-in. That way users have more control over their experience. For users that disable this feature, signing into a Google website will not sign them into Chrome."

So that was last week's brouhaha that Matt Green weighed in on, and that's why he's given up on Chrome and so forth. So they're going to give us a switch. That is, there was

one always. But it's buried deep in that flags infinite list of things you could tweak. And now it'll just be a little slider that regular people can do.

He says: "We're also updating our UIs to better communicate a user's sync state. We want to be clearer about your sign-in state and whether or not you're syncing data to your Google Account. We're also going to change the way we handle the clearing of auth cookies. In the current version of Chrome, we keep the Google auth" - and that's short for authentication, of course - "cookies to allow you to stay signed in after cookies are cleared." Which is to say all but ours.

"We will change this behavior that so all cookies are deleted, and you will be signed out." So it's like, okay. You guys didn't like that, we'll take that out. And he says: "We deeply appreciate all the passionate users [yeah] who have engaged with us on this." Although we're ignoring you in the case of www, but anyway. "Chrome is a diverse, worldwide community, and we're lucky to have users who care as much as you do. Keep the feedback coming." So, you know, I congratulate them for being quick to respond, and they are - I would have been happy with a checkbox that said I'd rather stay in Google. Delete everybody but Google.

Leo: I think that would have been more sensible.

Steve: Yeah.

Leo: But, you know, maybe there's some fatigue going on. All right, fine, whatever you want. Hey, I thought this was interesting, and I don't know if you have heard about this or not. But I just updated to the latest version of Windows. The Windows 1809 update came out today as part of Microsoft's update. And as I'm updating Edge, I get an interesting notice: "Issue with blocking third-party cookies." It's something we've talked a lot about, third-party cookies.

Steve: Yeah.

Leo: "The October 2018" - that's this one - "Windows cumulative update" - actually, no, that was one that came out just before, or maybe it's this, I don't know - "fixed an issue where Microsoft Edge's 'block only third-party cookie' setting was not working properly." And that's a big deal because of course we recommend you turn on that setting, for sure.

Steve: Wow, yeah.

Leo: "Third parties that add content to websites like advertisers can no longer place, see, or use cookies, even those that might have been previously placed in your browser. As a result of this update, you may see some changes in your browsing experience," et cetera, et cetera. But I thought that was kind of interesting. There was apparently a flaw in their third-party cookie blocking, but they fixed it. So just a little news.

Steve: And also, you know, I did that cookie stuff a long time ago, the Cookie Forensics. And one of the behaviors that we noted - again, this is along the lines of you don't know

it's broken. You don't know if it's happening unless you see it. I mean, you need tools to observe things.

And so when I created that set of tools that allowed you to very carefully understand the exact cookie-handling behavior of browsers, one of the things we learned is that browsers were doing different things with turning cookies on and off. Like in some cases the cookie off switch prevented the acquisition of new cookies, but any cookies that were already there were still being sent back. In other cases the switch did what you just described that Microsoft has fixed, which is even if cookies had previously been accepted and planted in your browser, setting the switch off would henceforth prevent them from being sent back. So it's not always obvious what the switch does. And in some cases people are caught out, caught by surprise. It's like, wait, this doesn't do what I thought it was doing. And in some cases it doesn't do anything at all.

Leo: Yeah.

Steve: Good that they fixed that.

Leo: Yeah, yeah, yeah, absolutely.

Steve: So we've often talked about the theoretical danger of a zero-day. And when this couple months ago, I guess it would have had to have been in August, not long before the September, that is, the following month's patches because there was time for Microsoft to patch it. But we'll remember that a, I learned subsequently, female security researcher was having a bad day, and she tweeted that she didn't really care much about life or something, and she also didn't feel like reporting the zero-day she had just found to Microsoft. So here it is.

And this was the advanced local procedure call exploit which could be leveraged through Task Manager. And of course everybody jumped on it. Some other researchers realized, oh, not only 64 bits, but also 32 bits. And not only Windows 10, but also Windows 7. And so it was bad. And I said at the time that it doesn't let bad guys get into your machine from the outside. No doubt that's way worse. But it's still very useful for malware to be able to elevate its privilege to a system level. And in this case, since Task Scheduler runs as a system-level process, this allowed essentially a means to get Task Scheduler to elevate malware to system level.

Anyway, I wanted to sort of close the loop because, as predicted, malware was indeed quick to jump onto this zero-day privilege elevation bug. Bleeping Computer reported that a ransomware known as GandCrab v5 has been seen leveraging this vulnerability to gain system privileges to more deeply infect a target PC. And as we know, it got patched in September, so it's been patched.

But again, it's worth noting, I think, that in this world that we live in today, bad guys know that a disclosed vulnerability like this will have a very brief shelf life, at least in an environment where updates are flowing regularly and being installed continuously. Which is certainly the case by and for most Windows systems. I guess enterprise systems, where they're needing to more carefully vet updates so that they don't foul things up, there may be a delay. But even so, malware authors, on the off chance that they're able to get into a system that has not been patched, are able to get up to additional mischief in a short period of time.

So as we know, there are systems that are not being kept current, such as, of course, Apache Struts, famously. But even in Windows a zero-day vulnerability was jumped on quickly and is in use in the wild, hoping to find systems that are some number of weeks behind in being patched. So there. And also in this environment there's no way to view that public disclosure of this zero-day vulnerability as anything short of irresponsible. I mean, if she found it and didn't feel like reporting it, well, just saying nothing would have been far more responsible than blabbing it publicly to the world because it has hurt people. Which is unfortunate. And I'm sure Microsoft would have fixed it without it going public. Although we have seen Microsoft, in cases that we've reported recently, like being a little lazy about getting around to fixing things until their feet are held to the fire. So who knows.

ESET has uncovered the first UEFI rootkit found in the wild. And it comes, unfortunately, courtesy of this - I wish we could just all agree on one name for these Russians: Sednit or APT28 or Sofacy or Strontium or Fancy Bear. I like Fancy Bear. Let's call...

Leo: Let's stick with Fancy Bear, yeah.

Steve: I really do. That one, that's got a little ring to it. So henceforth known as Fancy Bear. Let's everybody just call them Fancy Bear. Anyway, this is part of a rootkit malware system known as LoJax, J-A-X, because of some other malware which was previously found to be leveraging that - I can't remember the name of it. It's that stuff like Lenovo has in their BIOS, that system, it's kind of termed "LoJack," although it's compu - oh, Computrace, that's the name of it. Computrace had been subverted and turned into malware some time in the past known as "LoJack" malware. Anyway, so these guys called this LoJax.

And the problem is, because it's from Fancy Bear, it's state sponsored and is raising some concern. This UEFI rootkit has the responsibility of dropping this LoJax infection module onto the system. So the point is that you get your system infected. It uses a signed utility, sort of like repurposes a signed utility to write to the flash ROM on the motherboard in order to install this UEFI rootkit, which links itself into the boot process and is then responsible for essentially re-dropping this LoJax infection module out into the system whenever the system's booted. Which of course means that it is able to survive complete wipes and reinstallations of the operating system. So if you thought something was a little weird, or something set off an alarm bell that, oh, wait, there's some bad stuff in my computer, let's reinstall Windows, well, it comes back. Or let's change the hard drive. It comes back. So it can even, of course, survive hard drive swaps.

One of the things that is crucial, however, is that it cannot function in an environment of Secure Boot. So it has not subverted the Secure Boot process. And we did a whole podcast on the Secure Boot system and the technology, we've talked about it since, where Secure Boot functions, by starting with an absolutely trusted anchor, and that anchor checks the signatures of everything it loads before it runs it. And this stuff does not have a valid signature, thank goodness.

So from I think it was Secure Boot began in Windows 8. I don't think we had it in Windows 7. So, yeah, I think it first appeared in Windows 8, and of course it's part of Windows 10. Typically you are able to turn it off in the BIOS. If you have for some reason turned it off in the BIOS, it's probably worth turning it on. That is, running with it on, if it doesn't cause some sort of daily problem, because we are beginning to see instances where something, even if it gets into your system transiently, I mean, we're seeing now an instance where something that is briefly able to somehow arrange to run

is able to get down into your firmware and then become persistent and be much more difficult to dislodge.

So the advice, as always, is make sure that you are making - like watch your system boot, look at the version number the BIOS reports. Check the motherboard manufacturer to see whether they've updated usefully since then; and, if so, it's worth updating. And it's almost certainly the case that updating the BIOS would re-flush and get anything that had written itself in and linked into the UEFI boot chain unlinked and out of there. So make sure your BIOS is current, but also run with Secure Boot enabled when you're able to do so.

It is the case that booting something else like SpinRite, for example, in the future would require you to turn that off briefly in order to allow something else to boot. So there are benign reasons for needing to do so.

Leo: Or installing Linux.

Steve: Exactly. And of course that was our great fear, remember, that we were worried that BIOSes would not allow it to be turned off, and so users would buy a computer where they had absolutely no control over the operating system. So fortunately...

Leo: Right, right. Fortunately, that did not happen.

Steve: ...that did not come to pass, yes. So I mentioned a brand new botnet, Torii, T-O-R-I-I, and also that botnets are becoming increasingly persistent. The one thing we don't need is another botnet, yet that's what we have. This was dubbed "Torii" because it attacks devices' Telnet ports through the Tor network. So it's using Tor to obtain anonymity for its scanner, which is scanning through Tor in order to get to remote devices' Telnet ports. What that tells us primarily is, first of all, don't have Telnet exposed if you don't need it. And if you do, do not, just do not use any kind of human compatible password.

I think we need to come up with an acronym for that, Leo. Are you using human-compatible passwords? If so, stop. We need bot-compatible. We need, like, automation compatible. Your password should not be human compatible. It has to be something which is 64 characters long of gibberish that you have no chance of even possibly typing in correctly, especially not when you can't see what you're typing. So just it's got to be a cut-and-paste password. Ooh, I like that. Maybe that's the way to do it, a cut-and-paste password.

Leo: Cut-and-paste password, I like it. Or key, you know, public key, that kind of thing. I'd stay away from Telnet in general, though.

Steve: Oh, yeah. Okay. Well, so don't have it open if you don't need it. If you do have it open, give it an impossible to manually enter password, something that you have to cut and paste in order to log in, or hopefully you're using a Telnet client, and you taught it how to log in for you. But the point is it's using brute-forcing over time to get into the system. And the problem is, once it gets in, it is really in there. It is superior to the seemingly endless Mirai botnet variants which come and go because they lack a strong persistence mechanism. By contrast, Torii appears to be all about persistence. It has six different means for making sure that it survives reboots. Avast has a wonderful blog

post, I've got the link to it in the show notes if anyone's interested, I mean, it's really detailed and extensive.

They said of Torii, quote: "The malware's dropper makes sure that the second-stage payload is executed and that it will remain persistent. It is unique in that it is remarkably thorough in how it achieves persistence. It uses at least six methods to make sure the file remains on the device and always runs. And not just one method is executed. It runs all of them: automatic execution via injected code into ~\.bashrc; also @reboot clause in crontab; System Daemon service via systemd; etc/init and PATH, once again calling itself 'System Daemon'; modification to the SELinux Policy Management; and it's in etc/inittab."

So, I mean, it is doing everything in Linux that it can think of to get itself to run and not be removed. And so if someone comes along and goes, "Oh, what's this in my bashrc? I don't want that." You take it out, it doesn't matter. It has five other ways of achieving execution.

So Avast says: "The infection chain starts with a Telnet attack on weak credentials of targeted devices, followed by execution of an initial shell script. This script looks quite different, they write, from typical scripts that IoT malware uses in that it is far more sophisticated. The script initially tries to discover the architecture of the targeted device and then attempts to download the appropriate payload for that device." And this sounds familiar. I've not ever talked about this particular botnet, but I've said the same thing about other botnets, which tells us this is what new malware is doing.

"The list of architectures," they write, "that Torii supports is quite impressive, including devices based on x86_64, x86_32, ARM, MIPS, Motorola 68k" - who's using that? - "SuperH and PowerPC, with various bit-width and endian-ness." So all of that, both in little endian and in big endian. So, I mean, if you've got a device...

Leo: Most people use Python, I think. That must [crosstalk].

Steve: Yes, yes, yes. Oh, actually this one is written in Go.

Leo: Oh, Go, yeah, yeah, that's a good language, too, yeah.

Steve: It says: "This allows Torii to infect a wide range of devices running on these very common architectures." Then I had in my show notes: Torii is written in Go, allowing it to be readily cross-compiled for any supported processor architecture. It's clear that the developers of the botnet are seeking broad coverage; so they have built binaries for all popular CPU architectures, tailoring the malware for stealth and persistence. Also, communication with the command-and-control servers is encrypted, and capabilities include exfiltration of data that is behind the device and command execution.

So anyway, Avast concludes, saying: "Even though our investigation is continuing, it is clear that Torii is an example of the evolution of IoT malware, and that its sophistication is a level above anything we have seen before. Once it infects a device, not only does it send quite a lot of information about the machine it resides on to the command-and-control, but by communicating with the command-and-control it allows Torii authors to execute any code or deliver any payload to the infected device. This suggests that Torii could become a modular platform for future use. Also, because the payload itself is not scanning for other potential targets, it is quite stealthy on the network layer." And they say: "Stay tuned for the follow-ups."

So anyway, if you're interested, the Avast posting about this is very detailed and interesting in all that they talk about. But what this suggests is we're sort of entering - we're sort of going beyond the let's see how many boxes we can commandeer in the afternoon and then blast some sites with DDoS to let's find places we want to set up permanent residence and deliberately behave ourselves so that we don't give away our presence.

And since it's communicating over an encrypted channel through Tor, and it's arranging, I mean, it's really concerned about maintaining its persistence, this suggests that - oh, and since it can download anything and reports all about its architecture and what it's finding, basically the people are investing in brute-force cracking into the system. The Linux machine that's hosting a Telnet server and a Telnet service, setting up shop there, and then being able to download, knowing what architecture they have, then being able to download whatever things they want to in the future, and essentially becoming a persistent presence on that node.

So I say again, really, the Internet attack surface of networks is their routers. And we are, interestingly, that is clearly where the next big threat is going to come from is beachheads being established. And if you don't look, you can't determine whether you might be a host of such badness.

The government posted an interesting alert, just sort of kind of a newsflash. It's not really news for us, but I thought it's still really, really important. And it's yet another example of a particular service which is widely exposed and, I would argue, is really difficult to be safely exposed. I use Remote Desktop Protocol myself for - it's just so convenient, being that I'm a Windows developer and a Windows user, and I've got multiple facilities in different locations. But not a single instance of it is exposed to the Internet. It's just crazy to do that.

Microsoft has had authentication problems with RDP in the past. And we see instances where things like the early versions of SMB protocol, of Windows file-and-print services have had authentication problems. I mean, it's difficult to get this right. Anyway, last Thursday the ic3.gov site just posted sort of a public service announcement titled: "Cyber Actors Increasingly Exploit the Remote Desktop Protocol to Conduct Malicious Activity." And I won't go into it all because we understand this. But they gave four concrete examples that I thought were interesting.

They said: "The CrySiS Ransomware: CrySiS ransomware primarily targets U.S. businesses through open RDP ports, using both brute-force and dictionary attacks to gain unauthorized remote access. CrySiS then drops its ransomware onto the device and executes it. The threat actors demand payment in bitcoin in exchange for a decryption key." That's the first example.

Also: "The CryptON Ransomware utilizes brute-force attacks to gain access to RDP sessions, then allows a threat actor to manually execute malicious programs on the compromised machine." Similarly, these guys ask for bitcoin ransom to decrypt.

"SamSam Ransomware uses a wide range of exploits, including ones attacking RDP-enabled machines, to perform brute-force attacks. In July of 2018, SamSam," they write, "used a brute-force attack on RDP login credentials to infiltrate a healthcare company. The ransomware was able to encrypt thousands of machines before detection."

And then, finally, they refer to the Dark Web Exchange: "Threat actors buy and sell stolen RDP login credentials on the Dark Web. The value of credentials is determined by the location of the compromised machine, the software utilized in the session, and any additional attributes that increase the usability of the stolen resources." And I was curious, so I browsed around a little bit. And for about $11 or $12 you can buy a login

credential for some random machine in some random country. The ones I was browsing through did not have admin rights. But you can buy them, and you can log into someone else's computer and see what's there, I guess. It's crazy. But, I mean, it's the reality of today.

So again, I don't think, I mean, it's very convenient to have Remote Desktop Protocol accessible from the outside. The only way to do it is the way I do it, which is to put it behind a VPN, in my case OpenVPN, Leo, as you and I both do, and use certificates for authentication, not username and password, or not only username and password. You have to do that in order to be secure. And then the advantage of that is, once you have connected with a VPN, you are essentially on that network, and then you have access to all the goodies that are there. So that is absolutely the way to configure things.

So they finish up, just under vulnerabilities they explain that the number one problem is weak passwords. They say: "Passwords using dictionary words or do not include a mixture of upper/lowercase letters, numbers, and special characters are vulnerable to brute-force attacks and dictionary attacks." And of course the problem is that, unless you put limits on guessing, then someone could just sit there over months, trying and trying and trying until, bingo, they finally get in. Unless you're monitoring and/or foreclosing on mistakes made trying to log in, you're not going to know that's going on.

So they also said the outdated versions of RDP may use - and here it is again - this flawed SSP, the encryption mechanism, which enables the potential for acquiring credentials during a man-in-the-middle attack; and leaving it on port 3389, which is the default port. If you do nothing else, change it to some obscure port somewhere else so that you're not showing up in every Shodan scan of "Please give me a list of all the RDP ports that you know about, Shodan," which of course it does, if you ask it. So by all means, I mean, it's not any substitute for other things. But at least change the port that it's running on, which is relatively easy to do.

And in the account policies it's also possible to set a number of login attempt failures and lock the user out for some length of time. Really, we're past the day when you should need more than three or four attempts to log into a system, especially when the system you're logging into probably remembers your credentials and does it for you the first time every time. So it makes sense to set a policy to just say, you know, lock a user out. And, yes, it can cause a little bit of admin backlash if someone is unable to log in, given three or four tries. But it's a worthwhile tradeoff, I think.

And Leo, finally, this is another little tidbit that Bleeping Computer brought to my attention that I was unaware of, frankly. I, as I'm sure you and our users do, I frequently go to a site that pops up a little dialogue underneath the URL.

Leo: Yeah. I hate that.

Steve: Wanting to allow notifications.

Leo: This is really the bane of, when this started - I just hate it.

Steve: I know. I completely agree. And it's like, what? No. I don't want notifications. I just want to do whatever I'm doing right there right now and then be left alone. Well, it turns out that this is being used in a social engineering attack so that, for example, a site knows if it's going to be prompting the browser to make this request. So now what sites are doing is they're putting up their own dialogue on the page to point to this prompt

telling you - and I've got a picture in the show notes; yes, it's on the screen, thank you - pointing to it, saying "Click Allow," and they're pretending that they can't play the video or do whatever it is you want until you click "Allow."

Leo: Oh, man.

Steve: Yes. So they've escalated this to the next level. Now, what our users need to understand is that, I mean, this is a powerful permission which sites are asking for and almost should never be given. And that is that, even when you are not using your browser, when you've closed the window, when nothing is onscreen, this allows a website to pop up a notification down in the lower right of your screen in the notification area of your OS anytime they want. And most users won't connect these events. They won't realize that yesterday when the site said we need to allow notifications, and they said, oh, okay, and clicked "Allow," that, first of all, this allowance is persistent; and it allows an unassociated, it's not just when you're on the page or just when you're using your browser, it's anytime the browser is running that site now has permission to pop up a notification.

So the good news is you can retroactively rescind any of these allowances you have given in the past. I've got the link in the show notes. Bleeping Computer walks you through. And it's worth doing a little auditing if you're worried that you have ever said, oh, okay, if you say so. First of all, just say no. Unless you're somewhere where you really do want to allow a site to pester you for whatever reason. There are doubtless good reasons, maybe.

But certainly I mean, this is sort of like the number one rule of Internet hygiene, which is never download something because a website you're visiting tells you to. No. No. Don't. Just no. Similarly, you probably don't want to allow a site, give a site what turns out to be a broad permission to pester you in the future. The good news is you can poke around in your web browser. Bleeping Computer will show you how, if you follow the link in the show notes, to go in and just remove any permissions you may have mistakenly given to sites in the past.

And, finally, get your popcorn ready for tomorrow's first-ever Presidential Alert, which we will all be receiving at a bizarre time, apparently beginning at 2:20 p.m. Eastern, or 11:20 p.m., Leo, during...

Leo: Right in the middle of Windows Weekly, yeah.

Steve: ...Windows Weekly.

Leo: Can't wait.

Steve: With Paul Thurrott and Mary Jo. We will get the Presidential Alert. And a class action lawsuit has been filed by three individuals who oppose the idea. And so just for anyone who's clear, President Trump cannot send one from his phone. This is not...

Leo: I don't know if that's clear, but all right.

Steve: Okay. Well, it should be. This comes...

Leo: It comes from FEMA.

Steve: Yeah, it comes from FEMA.

Leo: But why do they call it a Presidential Alert, then?

Steve: That's a good question. But it's supposed to be a terrorist attack, a national disaster, I mean a natural disaster, you know, a tsunami or something, I mean, like something really that merits alerting the entire cell phone-owning population of the nation. So anyway, all that's happening tomorrow is that there's an alert that will be received and so forth. So anyway, once again, we just will - it'll be interesting to see. And for anyone who's interested, I do think it would be fun to be somewhere where there's lots of people, like being in a meeting.

Leo: Yeah, all going off at once.

Steve: All going off at once, yeah.

Leo: That'll be hilarity.

Steve: Yeah. Okay. So a little bit of miscellany. As I mentioned, this evening, as I'm winding down, I will be finishing "Masters of Doom," which I really enjoyed. I subscribe to the Kindle, what is it, Kindle, I forget the name of it.

Leo: Oh, I know, the monthly, you get free books, yeah.

Steve: Kindle Everywhere.

Leo: Unlimited, Kindle Unlimited I think.

Steve: Kindle Unlimited, where for a fixed price lots of books are available for free. And so something, I don't know why they offered it to me, I'm not a gamer, I never really have been, although I did, I mean, I was so fascinated by the technology of Doom. And it was a compelling...

Leo: Oh, yeah, that was amazing, yeah.

Steve: It was arguably - it was amazing at the time. Anyway, this is, for what it's worth, the story of John Carmack and John Romero, who were the original cofounders of id Software. And it's a great story. It starts out with both of them in their youth and follows them through the trials and tribulations of creating this series of first-person shooters.

And I just really enjoyed it. So for what it's worth, I mean, for me - and Leo, you and I are about the same age. We lived through this with our Apple IIs and our PCs. And it was just sort of fun to go back through it.

I did want to mention I also finished the last book of the third trilogy of the Rho Agenda. I talked about it first years ago, and it was very popular among our listeners. Remember the first one was "The Second Ship." Then there was "Immune," then "Wormhole." And that was like three teenagers, a pair of twins and a friend of theirs, discover an alien spaceship, and it's the adventures that ensue from there. There ended up being a total of nine books, three trilogies. And they were fun. I just finished a while ago the third book of the third trilogy. I also am caught up on the Frontiers Saga, which is planned, Ryk Brown's series, five sets of 15 books.

Leo: Oh, my god, he's...

Steve: I know, 75 books. And they really are fun. Of course the first series of 15 is finished. Now we're in the second series. He's written the first eight of those, and I am now current with those. And still a lot of fun. And Leo?

Leo: Yes?

Steve: Peter F. Hamilton is back.

Leo: Oh.

Steve: Yes.

Leo: Yes.

Steve: Yes. I would argue our absolute number one favorite author. Peter has never written a short book, or at least not recently. Actually some of his earlier - he's done some short stories in the past. This is titled "Salvation." It is the beginning of another new series, so maybe people will want to wait. We were annoyed when we got through "Pandora's Star," which was fantastic, but it left us hanging until we had "Judas Unchained," which was the second of that pair which told the whole story.

So a little bit about the so-called "Salvation Sequence." This is a brand new world or brand new environment, so it's not the Common...

Leo: He's so good, yeah.

Steve: Yes. It's not the Commonwealth Saga that he had so much fun with. The little synopsis says: "In the year 2204, humanity is expanding into the wider galaxy in leaps and bounds. Cutting-edge technology of linked jump gates has rendered most forms of transportation, including starships, virtually obsolete. Every place on Earth, every distant planet humankind has settled, is now merely a step away from any other. All seems

wonderful until a crashed alien spaceship of unknown origin is found on a newly located world 89 light-years from Earth, carrying a cargo as strange as it is horrifying."

Leo: Oh, boy. That sounds fun.

Steve: I know. I'm getting goosebumps just reading this. Because, I mean, it's Peter. Who can imagine what it's going to be, but it's going to be incredible. "To assess the potential of the threat, a high-powered team" - and you can imagine in his world what that means, I mean, you can't imagine, it'll be beyond our imagination - "is dispatched to investigate. But one of them may not be all they seem." And then, finally: "Bursting with tension and big ideas, Peter F. Hamilton's 'Salvation' is the first book of an all-new series that highlights the inventiveness of an author at the top of his game."

Leo: How exciting.

Steve: Oh, Leo. Okay, I'll see you next month.

Leo: Now, are we going to get the first one and then wait a year and then the second one?

Steve: I'm afraid so. I mean, these are...

Leo: Yeah, what are you going to do? Yeah.

Steve: Yeah, but I'm not waiting. What I end up doing, what I did with "Pandora's Star" and "Judas Unchained," was I reread "Pandora's Star" just to get back in and get rolling again. Oh, but his stuff is so fresh and so new and so imaginative. I mean, and that's, frankly, that's what sets it apart. Like the Frontiers Saga, it is, it is a saga. It's character driven. Nathan and Cameron and Jessica and, I mean, there's like a whole bunch of other - Telles and a bunch of people. And everyone who's been reading them knows who these characters are. And so it's just sort of you go along, and it's what happens, but nothing much. With Peter Hamilton, you're stepping into real imagination, and that's what I really love so much about him.

Leo: Good. Oh, I can't wait.

Steve: So anyway, I had a long - I was going to talk about SpinRite. I had a long DM, a direct message, from somebody in Germany, Stefan B., but it's kind of involved, and I need to whittle it down because it's so long and detailed. And I didn't get to it in time. So we'll probably talk about him next week.

Leo: Oh, okay.

Steve: In the meantime, we'll have our final...

Leo: Take a little break, yeah.

Steve: ...break, and then we'll talk about the Facebook breach.

Leo: Oh, man. Have you ever been on Facebook? I don't remember.

Steve: No.

Leo: I didn't think so.

Steve: I have an account because I needed to talk about the various security and privacy features. So I've had to go in from time to time. But no. And I've got all these people, I mean, I think all these friend requests. But I'm thinking, okay, I just, you know, I just - I can't. No.

Leo: No. You're smart.

Steve: No.

Leo: Honestly, you're smart.

Steve: Oh, and I did have a high school friend who hiked the whatever that is, the top of the mountain, the Pacific something or other trail?

Leo: Pacific Rim Trail, yeah?

Steve: Yes, from San Diego up to Canada. And so he was posting on Facebook, of all places. And it's like, oh, crap.

Leo: You can still read Facebook posts without an account.

Steve: No, I thought you had to be - back once upon a time, at least, you had to have a Facebook account in order to see Facebook people's stuff.

Leo: Not if they're public. If they're public postings, you can read it without logging in. Of course if they're "friends only" or somehow limited, you'd have to log in just to [crosstalk].

Steve: Well, and of course that is a perfect segue into what the nature of the breach is because it was a mistake in the "View As," "View My Page As" that was the cause of the problem.

So I think the thing that has received the least attention, but it is certainly significant about this Facebook breach, when you hear 50 million accounts were compromised, and another 40 one level removed may also have been, you sort of think, oh, look, mass collection breach of some sort. That's what we're used to, like when a web service loses its database it's well, you know, how many just like randomly chosen records out of total got loose?

What's significant about this is that this particular breach was absolutely targetable by its nature. And in fact Mark Zuckerberg and Sheryl Sandberg's accounts were both targeted and potentially compromised. So the point is that the people who discovered this had to do a lot of work. It's a sophisticated attack. So somebody was poking at Facebook for some period of time because Facebook is Facebook. And the ability to perform the attack occurred in July of 2017; right? So 17, 18, I guess, months ago, 17 months ago. So quite a while ago. That's been there for all this time. The only way the breach was discovered was a side effect of the way it works is people apparently logging in.

And essentially what you're able to do, and I'll explain a little bit what we know of how, you're able to cause an unauthenticated login of someone else in order to bring their login token current. And it was the sudden increase in the background continuous and certainly high level of that happening. But it suddenly spiked. And so the Facebook people said, you know, the engineers said, wait a minute, why are we suddenly seeing a dramatic increase in the rate of people logging in? And that's what tipped it off.

But we don't know over what period of time this thing occurred. And the point is that it's completely stealthful unless someone notices by looking at their own account, wait a minute, somebody logged in from somewhere else and left a breadcrumb behind, or something gets changed. Okay. So the crux of this is a feature that's sort of scary, but also certainly useful. I remember, I guess it was when I was playing with trying to get a hold of and get a handle on the way Facebook's viewability stuff works. As I said, Leo, I do have a Facebook page, and I don't use it. I don't think there's anything posted on there. But I created the account in order to see other people's and also when I wanted to explore the Facebook privacy and security settings because I've talked about them a couple times.

And so certainly one of the things that is very useful, because in any environment where you want to curate who can see what, is you want to test that. You want to view your own page as if you were someone else. And, well, that's impersonation. And it turns out that Facebook understood that this was dangerous, but they also understood they needed to offer the feature. So you've always had this "View As" feature, which would allow you to cause Facebook to show your page as if you were someone else. And you get to specify whom. So if you said "I want to see Mark Zuckerberg's view of my Facebook page," on some level there's been the invocation of Mark Zuckerberg's identity into viewing your Facebook page. So that was part of this, is the way "View As" works.

Then it turns out that there's a video tool, a video upload capability which allows you to send somebody like a birthday greeting, upload a video. And it's that tool which acquired a bug in July of 2017 which caused it to incorrectly manage the identity tokens that it was having to juggle, the authentication of different people such that the bad guys were able to, in a multifactor attack, to leverage "View As" and the mistake in this video upload tool in order to obtain a current logged-in authentication token for - and here it is - anyone they wanted, which is what makes this so scary, is this thing has been in place since July of 2017.

We don't know that it hasn't been used, that it hasn't been known. Who knows how many mysterious breaches or people's pages being changed or mischief or what has been gotten up to during that time because, again, it wasn't, you know, Zuckerberg's was compromised. Sheryl's account was compromised. So clearly, before they began a much

higher speed wholesale rifling, they were going after high-profile targets, it seems. So that's now been closed. Facebook, understanding the nature of the breach, is able to go back in their logs, see this being done.

And then what happened on Friday was that 50 million people, actually 90 million people suddenly found themselves logged out of Facebook. And it was like, you know, everyone's used to Facebook leaving you persistently logged in because it's convenient. Well, and as we know, what that means is that you have a cookie in the form of an authentication token from Facebook, which your browser presents whenever it pulls any resource from Facebook, which authenticates you as being logged in as you on this browser. So all of those authentication tokens were immediately invalidated, and consequently people had to log in again.

Now, the reason, sort of the other shoe here to drop, is that it's bad enough that a bad guy could obtain an authenticated session for anyone they wanted. Now, that's what this means. I mean, since July of 2017, anybody who knew about this could obtain a valid logged-in session for anyone they wanted. But what that means by extension is that the now unfortunately very popular "Login with Facebook" can be abused because it means that your browser contains a valid Facebook login token for anyone you have targeted, which means when you log in with your Facebook account, you're logging in with their Facebook account, meaning that you're able to log into any other third-party service where they have an account recognized by Facebook.

So, for example, under single sign-on, or of single sign-on, in Wired magazine's coverage of this, they said: "The debacle also underscores broader concerns about single sign-on, which Friday turned into the ultimate object lesson in the inherent tradeoffs between security and convenience. Kenn White, the director of the Open Crypto Audit Project, said: 'Single sign-on schemes are great; but the downside is, if a single sign-on gets breached, you're hosed.'"

So Wired says sticking with a single more secure sign-in does make sense, especially for use on sites that don't have the resources or inclination to invest heavily in security development. Actually, none of that's true. Because, well, anyway. But just like you want your passwords to be unique, so compromising one doesn't expose them all, account diversity is also vital online, no matter how ironclad a particular sign-in scheme is. Then they say, quote: "You don't want a situation where there's one breach, and your entire online identity is gone," says the same guy, Kenn White.

So anyway, that's what happened. It was longstanding. It was, I mean, and again, it's hard to do what Facebook is doing at the scale they're doing it. I would argue that View As is necessary and terrifying. I mean, it is inherently a horrifying, from an implementer's security standpoint, I mean, it is a real pucker factor because, boy, you've really got to get it right. And it looks like largely they did; but in one place they missed something, and someone found a little way to wedge themselves in and leverage something which is - I mean, and Facebook, obviously, as we all know, is really staggering at the moment under, you know, you just said it, Leo, you deleted your Facebook account. I mean, they've been staggering under all kinds of privacy issues as a consequence of being, what is it, two billion users?

Leo: Two and a half, yeah.

Steve: Yeah, being so popular. I would argue against Zuckerberg, who said - he called it an arms race? It's like, no. I mean, you know, it's hard to get it right. I absolutely agree. And here I empathize with them that, if you're going to have the ability to show different classes of users different views of your content, then you need to be able to audit that,

which means you need to be able to say, "What would so-and-so see of my page?" And it's like, okay, scary.

But it is actually a perfect example of, I mean, I already hate OAuth. I consider it a horrific kludge because, as we know and we've talked about, users don't understand that when you log in with Google or log in with Facebook or log in with Twitter or whatever, the reason those services are happy to provide this referral service is they're tracking you. They're knowing everywhere you're logging in and adding it to their compilation of your profile, and you're updating your information about them. And as we've just now seen, if that service suffers a breach anywhere you have used them, you're now subject to, I mean, they've extended the blast radius beyond their own perimeter to everywhere else.

So anyway, it's one of the reasons why, when I first told us all about SQRL years ago, I said it's a two-party solution. It's secure single-factor. It's like the only reason you need multifactor is none of them are secure enough, so you need lots of them in order to increase the security. If you have secure one-factor, then it's all you need. And the beauty of it is it's just two-party. The problem with this "log in as" is you're bringing in a third party, and here's an example of it going wrong.

So anyway, that's what happened with Facebook. And again, I don't think - I'm not saying, oh, this is a glaring problem. I mean, I would argue the reason they're already in hot water, that has been Facebook's policy problems that have come home to roost. Here, this was a really obscure bug, and anybody can make a mistake. They did get it fixed quickly.

And to our listeners I would say take this opportunity to go use the Facebook tool which they provide of where have logins to your account occurred, from which devices and from which locations. Just make sure you recognize them all because that's worth doing. And also eliminate any applications that you've given permission to use your Facebook account that you're no longer using. As we've said, that kind of periodic audit can be a good thing to do.

Leo: Yeah, maybe this was just a wakeup call. Is there a safe way to use single sign-on solutions like that, OAuth solutions?

Steve: The problem is users are not given enough control.

Leo: Right.

Steve: There ought to be a way to revoke the otherwise persistent relationship of the connection between the authentication token that you give the third party and the site you're visiting. Facebook did also kill all of those referral tokens, essentially, which was good. But the problem is it's done behind the scenes. And so it's mean to make the system easy to use. But because users are not given visibility into it, it's not something that's easy to do. It's better just not to have it in the first place.

Leo: Used to be you could do your own OAuth, I remember. You could have your own site. But I don't know how that would work if a site didn't know about it.

Steve: Yeah, remember you were able to put a token on a page on your server.

Leo: Yeah, put it on your server, yeah.

Steve: And bounce you through.

Leo: That would be so much better. I guess it wouldn't help the average Joe. But, yeah, that's the problem.

Steve: Yeah, yeah. The average Joe is going to have a solution here pretty soon.

Leo: It's called SQRL.

Steve: Whee.

Leo: SQRL. Well, we're going to let Steve get back to his SQRL right now. We do this show Tuesdays, 1:30 Pacific, 4:30 Eastern, 20:30 UTC. You can come by, watch us live at TWiT.tv/live. You can also get an after-the-fact, on-demand audio version of this show from Steve's site, GRC.com. He also has transcripts so you can read along as you listen. Those are all at GRC.com. While you're there, pick up a copy of SpinRite. If you have a license to SpinRite now, you'll have a license to 7.0; right? Automatically?

Steve: Not 7.0, but all the 6.1s.

Leo: Sixes up to 7.

Steve: Right.

Leo: That's GRC.com. And there's also lots of free stuff there. Everybody should take a look at it. It's a fun site: GRC.com. We have audio and video versions of the show at our site: TWiT.tv/sn. You could subscribe in your favorite podcatcher. You'll get it automatically the minute it's available. I think that's about it, Steve. We have concluded.

Steve: Okay, my friend.

Leo: And I'll see you next time on Security Now!.

Steve: Thank you, Leo. Bye.