Exploits & Updates

Description: This week we discuss Windows 7's additional three years of support life, MikroTik routers back in the news (and not in a good way), Google Chrome 69's new features, the hack of MEGA's cloud storage extension for Chrome, Week 3 of the Windows Task Scheduler zero-day, a new consequence of using "1234" as your password, Tesla making their white hat hacking policies clear (just in time for a big new hack!), our PCs as the new malware battlefield, a dangerous OpenVPN feature spotted, and Trend Micro, caught spying, getting kicked out of the macOS store.

High quality (64 kbps) mp3 audio file URL: http://media.GRC.com/sn/SN-680.mp3
Quarter size (16 kbps) mp3 audio file URL: http://media.GRC.com/sn/sn-680-lq.mp3

SHOW TEASE: It's time for Security Now! with Steve Gibson. I'm Jason Howell, filling in for Leo on the second of three weeks in a row. Steve's going to talk about a whole bunch of updates. Security news is flowing even as the show is going on. We're going to talk a little bit about Google Chrome 69's URL hiding feature and whether you want to keep that or maybe toggle it off. There's a hack of MEGA's cloud storage extension for Chrome. We'll dive into that. Tesla is making its white hat hacking policies very clear, and it really just happened at just the right time because there's basically breaking news involving Tesla, and a whole lot more coming up next on Security Now!.

JASON HOWELL: This is Security Now! with Steve Gibson, Episode 680, recorded on Tuesday, September 11th, 2018: Exploits & Updates.

It's time for Security Now!. This is the show where we talk about all things security. And boy, today do we have a ton of security stories to check in on. Obviously Leo is out. I, Jason Howell, am in, in Leo's place. Not in his comfortable studio digs. I'm more in like the big studio here, the big open space that allows us to actually have an audience. Steve Gibson, we have a live audience in the studio today.

Steve Gibson: I think Leo kind of likes to keep the office to himself; right?

JASON: I think so.

Steve: It's sort of like his inner sanctum, and everybody else gets to play out there in the studio.

JASON: Yeah, totally.

Steve: So I think that works.

JASON: I think it's fair. He kind of like owns the place. He runs the place. So he's - it's fairly done.

Steve: Ah, I see somebody's white pants with their legs crossed in the...

JASON: I know. We've got not one, not two, not three, but four people.

Steve: Now they're uncrossed. Oh, okay.

JASON: Four folks in here who all made the trek. They're all in the Bay Area specifically for no other reason than to watch Security Now!. That's what I'm to understand, anyway.

Steve: And, you know, having driven that goat trail up from San Francisco to Petaluma, I really do appreciate the effort that they went to.

JASON: Oh, I know.

Steve: It's not easy to get there from civilization.

JASON: It's not easy to get there. But what they get at the end of the rainbow, they get Steve Gibson and Security Now!.

Steve: Eh, well, sorry about that. So we're Episode 680. And this is, of course, the 17th anniversary of 9/11.

JASON: Yes, it is.

Steve: The day that lives in infamy for the U.S.

JASON: Absolutely.

Steve: So there's been lots of acknowledgement of that and so forth. And so it's a perfect day for Security Now!. We were talking about before we began there wasn't any one big thing that seemed to, like, dominate. So I just named this episode "Exploits & Updates" because we have a lot of both.

I did have news on Patch Tuesday, which we will get to. We're going to discuss Windows 7's newly announced additional three years of life support. MikroTik routers are back in the news yet again, and of course never in a good way. Google has released Chrome 69 with some controversial new features which are raising some questions. There was a hack of MEGA's - MEGA is an encrypted cloud storage solution that's very popular. Their Chrome extension got hacked and was present in the Chrome Store for, I think, about four hours last week. And we'll talk about the consequences of that.

We've got Week 3, and that's the final week, actually, of the Windows Task Schedule zero-day, that advanced local procedure privilege elevation. We have an interesting new consequence of using a weak password like 1234. Tesla has made their white hat hacking policies very clear, and just in the nick of time because there's also a big new hack of the original security key that the Tesla Model 2s have been using. Our PCs are becoming a battlefield for malware, which we'll talk about.

We've got a dangerous OpenVPN feature was spotted, and looks like the door was closed before it got abused. And then Trend Micro was caught spying and has gotten itself kicked out of the macOS store. And we of course have a fun Picture of the Week that we'll talk about.

JASON: All right. Picture of the Week to begin things. Tell us what we're looking at here.

Steve: So this will not convey well in words, unfortunately. So I think our listeners, if they're interested, are going to have to grab the show notes and take a look at it. But I'll

describe it. It's the ever-wonderful and often brilliant xkcd brings us this. So we sort of start in the upper left. There's four frames. And the first frame shows sort of a network diagram with some things connected, but they're like floating around. And the caption is: "I wish these parts could communicate more easily."

And then there's an arrow taking us to the next frame that shows sort of the same parts that have a bunch of green connections, interconnections between them, and that seems like a good thing. The caption says: "Oh, this new technology makes it easy to create arbitrary connections integrating everything." And then the third frame shows now these things are all still connected, but now we've got some red lines and circles, and looks like it's not quite so happy. And the caption is: "Uh-oh. There are so many connections it's creating bugs and security holes."

Which leads us to the fourth frame, which shows a bunch of green enclosures around the different areas, breaking them apart. And now the caption is: "Oh, this new technology makes it easy to enclose arbitrary things in secure sandboxes." Which leads us - there's a fourth arrow leading us back to the first one that says: "I wish these parts could communicate more easily."

JASON: Wait. So we never reach the end. There is no actual end. We've never solved all these problems. It just keeps getting more and more complicated.

Steve: Yes. And in fact, you just said the word. As a consequence of our desire to interconnect, then realize, whoops, we've got some side effects that we didn't expect from the interconnections. So now we've got to isolate them, but now we're not happy that they're isolated because they really need to be talking to each other. So we're going to interconnect them maybe in a different way. Whoops, different problems. Now let's re-isolate them and so forth. We go around in circles. We end up dizzy and with lots of complications. So anyway, xkcd just does it again.

JASON: Love it.

Steve: It's like, you know, brilliant.

JASON: Yeah. They are super brilliant. It just reminds me of how services, I mean, this is such a, you know, as far as like Internet services are concerned, such an indicator of what we're so used to seeing. Service starts out, it's one single use case. You're like, oh, I love it for its simplicity. But then over the years they feel the need to kind of build upon it and do all these other things, which is great from a feature set standpoint, but just complicates things so specifically. And it's such a great effect that it loses the thing that it once captured.

Steve: Well, and probably there are several great examples of that. But one that is apparent, I think, to everyone is the original iPhone. The original iPhone, it's funny because it sort of looked incomplete. If you remember, it did not support applications, that is, third-party apps in the beginning. It had this screen with nine icons on it, kind of all in the upper. And they didn't even have a complete last row. There were, like, two icons out of the four that there was room for. It just sort of looked like it was unfinished. It's like, well, where's everything else? And the point is, no, this is what you're getting. And of course, as we know, the UI was designed for that. Everything worked really well. But people wanted more features; and unfortunately, with more features comes...

JASON: Comes a blanket.

Steve: Yes. Now we need folders to enclose the icons because otherwise we're just scrolling, like, where did this app, I mean, I often have the problem of, like, I'm sure

there's an app in here somewhere that I downloaded once. And you're just scrolling around, and it's not easy to find them. So now I just sort of search for it because I can't find it among all this junk that I've got loaded onto my iOS device. So, and of course this is the problem is that everybody wants more features, yet then with that comes interactions among them, and complexity. And then it slows down. And, I mean, also old-timers among us will remember the days when Microsoft was always several chips behind, and you just couldn't get the systems to go fast enough because we were trying to get them to do so much. And I'm glad that the days of Windows crashing constantly are behind us. I mean, for most people, Windows seems to be pretty stable now. So that's good.

And speaking of Windows, this is second Tuesday of the month of September, Patch Tuesday. Microsoft, I don't have a detailed breakdown, which sometimes it's useful to give. This time, eh, not so much. The big news is the one thing that we were hoping for, expecting, and kind of wanting, was this advanced local procedure call zero-day, which was disclosed by that disheartened hacker via Twitter a couple weeks ago. We have followed it since. It matured. It was able to run on 32-bit machines, and then someone figured out how to change a filename, change a 3 to a 1 to get it to run on Windows 7. And we'll be talking about in a minute the fact that it did end up being exploited.

We now know, thanks to some researchers at ESET, almost immediately it has been patched. So it wasn't super critical because it wasn't a remote code execution. But as I said at the time, anything that gets in your system definitely wants to elevate its privileges, some say "escalate," to obtain the rights to write itself into sensitive areas of a system or to modify system files which by design people running with non-admin privilege are unable to do. So anyway, we got 62 vulnerabilities patched today, this morning, of which 17 were critical. Most of them were browser-related. The rest include Windows Office, Hyper-V, the .NET framework, and so forth. so as I said, it's one of those things people should patch as soon as they can because Microsoft is constantly fixing problems.

And speaking of constantly fixing problems, it was announced, on looks like the 6th is the date of this blog post, that Windows 7 support would be available beyond 2020. Which doesn't really surprise me. So the title, their title was "Helping Customers Shift to a Modern Desktop." However, being the author of Never10, I should mention now with nearly three million downloads of this little utility that I wrote, rather than "Helping Customers Shift to a Modern Desktop," I would title the blog posting "Being of Necessity Somewhat More Patient in Our Effort to Force Customers onto a Less Desirable Desktop, That They Really Do Not Want...." Anyway...

JASON: I actually think that that fits in with Microsoft's kind of ambition and habit of naming things really long strings of words. So I think that would actually work for their naming template.

Steve: Almost a more appropriate title than their official one. Yes. And speaking of which, even the guy's title, this is the "Microsoft Corporate Vice President for Office and Windows Marketing" is the person who wrote this. Anyway, he says, under the subheading of "Windows 7 Extended Security Updates," they wrote: "As previously announced, Windows 7 extended support is ending January 14th, 2020. While many of you are" - love this - "are already well on your way to deploying Windows 10" - yeah - "we understand that everyone is at a different point in the upgrade process." Yes. There's, like, barricades in front of some of us.

JASON: Wishful thinking.

Steve: Yes. And let's note that, despite Microsoft's extensive and highly controversial efforts to actively force everyone over to Windows 10, such that no one would still be on Windows 7 without actively resisting the push to Windows 10 - as we know, I mean, you had to struggle not to have your Windows 7 machine grant you Windows 10 despite your desire to stay where you were - today, okay, today Windows 7 remains the majority desktop in the world, with a greater market share than Windows 10. Now, not by much, granted. Windows 7, courtesy of Bleeping Computer, I got this chart from them because they pulled some market research - Windows 7 is at 40.27% to Windows 10 at 37.8. And then followed by the macOS at 5.86, Windows 8.1 at 5.10, and believe it or not, well, actually I shouldn't say that because I was on XP not long ago, XP at 3.30%.

JASON: I believe it. I actually kind of expected XP to be higher than that. But it was a longstanding version.

Steve: It doesn't know about, I guess it was SP2 doesn't know about SHA-256. It doesn't know about any of the TLS protocol updates. I mean, it's really - XP was really sort of struggling to stay alive. But the point is, for Windows 7 to still be more popular, to have a larger market share than 10, I mean, the only way that would happen is if people refused to budge. So it's worth noting, though, Windows 7 is waning, and 10 is gaining.

And of course one of the reasons is it is no longer possible for consumers to purchase systems with Windows 7, much as they may wish to. And the latest Intel chipsets are no longer compatible with Windows 7. Ask me how I know. Yes, I know. I'm deliberately now, when I need a new machine, I have to go an older, I think it's i6 or 7; 8 won't run Windows 7 any longer. So the reason 10 is ultimately going to win is just through attrition. People will hold onto 7 with, apparently, with their dying gasp. But they're going to end up ultimately losing because you just can't get it anymore.

Anyway, so Microsoft continues their blog post, saying: "With that in mind" - that is, that we're all in different stages of adoption - "today we are announcing that we will offer paid Windows 7 Extended Security Updates" - of course we have an acronym, the ESU Program - "through January 2023. The Windows 7 ESU will be sold on a per-device basis, and the price will increase each year." Oh, they're going to just force people off this thing one way or the other.

"Windows 7 ESUs will be available to all Windows 7 Professional and Windows 7 Enterprise customers in Volume Licensing" - meaning not end-users - "with a discount to customers with Windows software assurance, Windows 10 Enterprise, or Windows 10 Education subscriptions. In addition," they said, "Office 365 ProPlus will be supported on devices with active Windows 7 (ESU) through January 2023. This means that customers who purchase Windows 7 ESU will be able to continue to run Office 365 ProPlus." Will be able to run it? You wouldn't be able to run it? Oh, well, who knows what they're going to do. I mean, you know, they're switching this whole thing over to OS as a service, rather than an operating system as an operating system, because they can. So anyway, for what it's worth...

JASON: When did it initially launch? This was, what, seven, eight years ago?

Steve: Yeah.

JASON: So talking 2023, I mean, man, that's 13 years old. Hopefully by that point [crosstalk].

Steve: And the problem is people, I mean, the problem is it works just fine.

JASON: Yeah.

Steve: You know? It's like it's not - you don't have...

JASON: They made it too good.

Steve: Well, actually, I would argue that at the time Windows 2000 worked just well. Windows XP worked just fine. I mean, this whole idea of needing 64 bits, 64, come on. If they had designed the 32-bit OS correctly so that it could use virtual memory - but again, they didn't want to. Anyway, I guess I'll end up just on FreeBSD Unix at some point and then just say, well, good luck to the rest of you.

JASON: But you're an individual user. Would you qualify for it because...

Steve: No.

JASON: So you wouldn't even qualify for it if you wanted to pay and get some of that tech support.

Steve: And that will not bother me because, after all, I wasn't getting Windows XP updates for quite some time. And once again, everything just continued to work fine.

JASON: Hey, what do you know.

Steve: So, well, because the browsers really are our portal to the world. I mean, yes, you need to download software. But if you're careful, if you're not hanging out in sketchy areas of the Internet, if you're behaving yourself, and if you've got something watching your back, I mean, I guess now, well, I'm sure that Security Essentials will stop working for me also because it'll be like, agh, you're still here? You're still using that Windows 7? What's wrong with you?

Anyway, yes. I don't think, I mean, I have some Windows 10 machines. I'm talking to you over a Windows 10 machine. And there are some places where I have had recently to use updated chipsets and couldn't use 7, so I had to use 10. I did manage to scrape all of that other junk off of it. I mean, there's, like, there's HoloLens is built into Windows 10. It's like, I don't want a HoloLens on my operating system. Or Xbox anything, live or dead. So anyway...

JASON: Yeah, but you will. HoloLens is going to change the world, Steve. You're going to be right there with it. You're going to be podcasting from HoloLens in a couple of years. You just don't realize it yet.

Steve: God help me. Well, we are talking over Skype, so maybe that's true, I don't know.

JASON: That's totally possible, actually.

Steve: We'll have HoloSkype, with no choice.

JASON: I wouldn't be surprised.

Steve: Okay. So MikroTik is the way you're supposed to pronounce this. But I really did like pronouncing it mi-CROT-ic because it sounds so awful. And I know that they're very popular routers. There are a lot of people who like them. I mean, they're also very powerful. I mean, they've got a nice UI. They are feature packed. But here's where features come to bite you. I mean, this is just exactly like xkcd's cartoon. The MikroTik routers are back in the news, and not in a good way.

As we know, they're been suffering all year as a consequence of a problem that was first found in their SMB handling, and then their handling of Winbox authentication, which allowed for unauthenticated remote access. Winbox is a Windows-hosted UI which allows for administration of the router. The problem is, and for the life of me I don't understand how this could happen, the Winbox access is by default available on the WAN interface in addition to the LAN interface. In other words, exposed to the public. And it's a router. So of course it's going to be on the edge of your network, half of it looking out to the rest of the Internet, and the other half looking into your modest little network.

Anyway, back on April 23rd, MikroTik explained in their vulnerability disclosure, they said: "We have discovered a new RouterOS vulnerability affecting all RouterOS versions since v6.29. The vulnerability," they wrote, "allowed a special tool to connect to the Winbox port and request the system user database file." Of course the Winbox port should have never been available to connect to from the Internet. But they didn't mention that. They said the attacker would then use the user details found in the exfiltrated user database file to log into the MikroTik router remotely. And of course then the party begins.

Okay. So the vulnerability in question was titled "Winbox Any Directory File Read," and it's got a CVE of 2018-14847. And it turns out that it was found exploited by the CIA Vault 7 hacking tool called "Chimay Red," along with another MikroTik WebFig remote code execution vulnerability. So they're feature rich. These routers are popular. And to their credit, I absolutely give them credit for patching that vulnerability within hours of hearing of it.

But we know that without the ability to push the vulnerability fix out to devices somehow, or probably by devices automatically periodically phoning home to check for an auto update, and then autonomously doing it because I'm sure that those of us with routers who are security conscious, we've looked. You log into your router, and you tell it to check for updates. And it's like, oh, yes, I got a flashing exclamation point. There's an update. Well, how long as it been available? Is it crucial? Is it critical? You know, the router doesn't tell you. It just needs to do it. You need to - part of setting it up has to, you know, like you give it permission to maintain itself, and then it just does this for you. But as we've said often, probably too often, we're not there yet.

Okay. So now, today, this exploit code which affects all not-yet-patched MikroTik routers, is available from at least three different online sources. So it is hardly surprising when, at the beginning of last month, we reported, as we have been, that somewhere around 200,000 - 200,000 - MikroTik routers were enlisted in a massive Coinhive mal-mining operation. And since the beginning of these various attacks on consumer and corporate routers, of course, I've been observing how fortunate it is that the bad guys seem uninterested in the details of the networks lying behind these infected and infested routers. I've often commented that it's like let's just hope that's the way things remain, that all they want to do is stick mining malware on the routers and not care about the fact that they've established presence on the device through which all of your local network traffic transits.

Well, it turns out that's changed. Last week researchers at 360 Netlab posted their article titled "7,500+ MikroTik Routers Are Forwarding Owners' Traffic to Attackers." And they ask then rhetorically: "How Is Yours?" They wrote: "We understand that user devices come and go on the Internet all the time, so the data used in this blog reflects what we saw between August 23rd and August 24th. From our own scan result, we logged more than 5,000,000 devices with open port TCP 8291." That's this web API that the MikroTik routers support. This Winbox port is 8291. They said more than five million devices with that port open, and 1.2 million of them were identified as MikroTik devices, within which 370,000, which is to say 30%, are vulnerable to this CVE-2018-14847. In other words,

370,000 MikroTik routers can be remotely accessed today over this Winbox port and are vulnerable to this non-authenticated access.

Now, you're showing on the screen the chart from the show notes, which shows the distribution in the first column of the top 20 nations. Brazil, for example, has 42,376 vulnerable routers. Russia, in number two, at 40,742. Indonesia, 22,441. And onwards, downwards, all the way through the top 20 nations. This middle column shows the top attackers who have been observed attacking these routers by IP address. And the list of ports being eavesdropped on. I'll get to how that's happening next.

So on these routers is still Coinhive mining code injection, so they're generating revenue at some pace using the coin mining script. And remember we talked about this before where they use the opportunity of displaying a nonencrypted error page to inject the mining script into the user's browser. All the browsers that are being serviced behind that router will receive the error page unencrypted with Coinhive mining. So that's how they're getting around the HTTPS and TLS encryption, which otherwise would not allow them to inject their code into an encrypted connection.

Okay. Additionally, at the moment, right now, a total of 239,000 MikroTik routers are confirmed to have a Socks4 proxy enabled maliciously. The Socks4 port is mostly configured to 4153. And, interestingly, it only allows access from one single netblock, which is 95.154.216.128/25. So it looks like there's one perp behind all of this. In order for the attacker to gain control after device reboot, and the possibility that the public IP might change, the device is configured to run a scheduled task to periodically report its current IP by accessing a specific attacker URL. And the attacker is continuing to scan the Internet because of course it's a rich source of vulnerable MikroTik routers.

Okay. So, finally the MikroTik RouterOS device allows, as I said, it's very feature rich. It allows its users to configure the router to capture and forward packets using a protocol known as TZSP, which is the TaZmen Sniffer Protocol, which we've never talked about before, but it's an encapsulation protocol used for sniffing and forwarding packets. Intrusion detection systems use it, and it's supported by Wireshark that knows how to look inside that protocol and look at the packets inside. So it's well known. So they did a standards-based sniffer. Unfortunately, and again, I guess this is a feature, but the forwarding of the packet capture and sniffing is not restricted to the LAN. It can be set over the WAN.

So using first the fact that we have this Winbox vulnerability on 370,000 MikroTik routers, more than 7,500 of those have been configured with this sniffer actively capturing the traffic that is transiting the router. And you might say, okay, fine, but it's encrypted; right? Well, no. The number one port being captured is port 21, which is one of the two ports used for FTP, which is typically itself not encrypted. Second up is 143, which is IMAP, which of course is email, also, as we've often said, unencrypted. Next up, 110, which is the POP3 protocol. Not encrypted. Fifth up is port 25, SMTP, which is used for submitting mail as opposed to getting mail. Unencrypted.

So what these guys are doing is they've taken advantage of the Winbox vulnerability, which is widespread on MikroTik routers, and have used a feature in the router to essentially forward all of the FTP traffic, of which there's probably not that much, but maybe FTP uploads would be of some interest to them, you know, like people putting stuff up on a server somewhere. But essentially intercepting all of the email, unencrypted, in and out of that network.

So that's, again, as I said, MikroTik in the news. In their disclosure of this, at the end of their explanation of the counts they saw, the ports, posting that chart of everything, they said, that is, the 360 Netlab folks said: "We recommend that MikroTik RouterOS users update the software system in a timely manner and check whether the HTTP proxy,

Socks4 proxy, and network traffic capture functions are being maliciously exploited by attackers." Yes. I think that's good advice.

They said: "We recommend that MikroTik denies inbound access to the WebFig and Winbox ports from the Internet [uh-huh] and improve the software security update mechanism." Oh, I completely concur. So, you know, we know that anyone can make a mistake. And I do congratulate the MikroTik folks for quickly creating a patch for their mistake. It was within hours of finding out about it. I mean, yeah, anyone responsible would do that. But this disaster - and this is what you'd have to call this. This is a disaster. For there to be currently 370,000 routers more than six months after a critical vulnerability was publicly disclosed, I mean, it's not like nobody knows about this. Everybody knows about this. And people are not fixing their routers, leaving them open to people who want to exploit them.

So the point is this disaster is not the result of a mistake. It's the consequence of a deliberately and misguided policy of enabling WAN-side remote access and management by default. These routers default to making this access available. They should default to it being off. And then only if a given user actually needs or wants remote admin over this originally insecure, now secured service, should they turn it on with cautions about strong passwords and so forth in the router.

So I know everybody likes the routers. They're popular. They're feature-rich. But there is a serious policy problem. And if you happen to own a MikroTik router, absolutely make sure that you've got this WAN-side admin stuff off. There's no way there could be 370,000 of these exposed publicly if it weren't for the fact that they were exposed by default. So I don't know whether the firmware change closed it by default. They're probably worried that they're losing a feature. Well, it ought to be manually enabled, rather than automatic. There's just no way around that.

JASON: Absolutely. And maybe it'd take a scenario like this to teach them that lesson. One thing I'm curious about as far as the cryptomining aspect of this, which it sounds like is something that you've talked about in weeks past. But when you're talking about this number of routers, all focused on cryptomining to this degree, what does that actually translate to in crypto terms on the other end?

Steve: We did talk about this. And what they're doing is they're using a script that runs in the browser. They're mining Monero currency because that's CPU-friendly and GPU-hostile. So it gives computers a fairer chance, for example, as opposed to bitcoin, which is all just gone to ASICs now. So you just can't mine bitcoin with a regular CPU. So they're mining them in browsers. They're using a script hosted by Coinhive, which is the "service," unquote, I mean, they're definitely gray hat, tending toward black. So Coinhive is the service that the JavaScript comes from. Coinhive gets a significant piece of the action and is currently making a quarter million dollars, boy, was it a week or a month? At least a month. So I think it was a quarter million dollars a month in their piece of the action from this malicious use of their script on people's browsers. So there's money being made.

JASON: Absolutely. Wow. Big-time.

Steve: So we're now at Chrome, Google Chrome 69.

JASON: Yes. I upgraded on my desktop in the other room just this morning, actually.

Steve: Cool. And so I went to www.grc.com, which is the official name of GRC's website. And I was greeted by the nice EV Gibson Research Corporation with [US]. But the URL didn't have http:// nor www. They were missing. All I saw was GRC.com. And I had to

click twice in the URL field to see the full URL because I thought, what? Okay, what's more, you put blog.google.com in, and it turns it into blog.google. So I thought, what? Okay. So then I put blog.grc.com in because that's valid also, but it did not turn it into blog.grc, despite the fact that blog.grc.com is also backed by a highly trusted DigiCert extended validation certificate. So what's going on?

Well, first of all, I figured out the blog.google.com mystery, how were they turning themselves into just blog.google, yet blog.grc.com was not turning to blog.grc. I saw that Google, as I had mentioned to Leo some weeks back, now owns the "Google" top level domain, that is, .google. So blog.google.com turns into www.blog.google, which now in Chrome 69 appears as blog.google because they're getting rid of the www. in front of everybody's domain. They've decided, Google has, to call these subdomains "trivial," unquote, I'm not kidding. And apparently on the mobile platforms m. is also removed as superfluous.

So this has ruffled a lot of feathers, I mean, among the old fogies on the Internet, who are not happy about this. And it's funny because, like, oh, it was at least 10 years ago, maybe more, I had a decision to make with GRC, and that was originally you could get to GRC either with or without www. And as a consequence, some people put it in, some people didn't. But the Google was indexing us both ways. And I thought, okay, this is dumb. We ought to have, like, one official way of being a website. And so the question is, should it be with or without the www?

So there was a big discussion in GRC's newsgroups, and I decided to go with www, since I had other subdomains of GRC, like news.grc.com, for example, is for the newsgroups and so forth. So that seemed to be the right thing to do. And I look at, like, Amazon.com is www.amazon.com. Well, no more. Not if you're in Chrome. Now it's just Amazon.com, even though the www. is still there. And I guess there are some really weird - I did some poking around among all of the people who are unhappy about this. I guess things like example.www.example.com, the www just gets stripped out of the middle. And so Chrome is doing like these weird things to domain names, which has the purists up in arms.

So it'll be fun to experiment with this. I'm not, I mean, I have long said, and often said, that the whole concept of the URL is user hostile. You know, my mom, she said, "Honey, what is http://? What do I need to do that for?" And I thought, yeah, I know, Mom. But, I mean, you know. And I don't disagree that www is, well, can be superfluous. If you put GRC.com in, I do have that domain, I have a certificate for it, and I redirect people to www.grc.com. And now all the links for search engines to GRC are unified with www. Now Chrome has decided it's not going to show us the www.

So it'll be interesting to see how the controversy fares because they have created some. They've also done some other things in 69. It is the 10th anniversary of Chrome, by the way, so this is like the birthday party, the 10th year birthday party for Chrome. Discussing Chrome turning 10, they said: "Getting things done faster." And I'm quoting from Google. "You get a lot done online these days - booking travel and appointments, shopping, and working through your to-do lists across multiple sites at once. And we want to make sure that you can do all these things easily and safely. Now, Chrome can more accurately fill in your passwords, addresses, and credit card numbers, so that you can breeze through online checkout forms. All this information is saved to your Google account and can also now be accessed directly from the Chrome toolbar.

"We've also significantly improved the way Chrome handles passwords. Staying secure on the web means using strong and unique passwords for every different site. When it's time to create a new password, Chrome will now generate one for you," they say, parens, "(so you're not using your puppy's name for all your passwords anymore).

Chrome will save it; and next time you sign in, it'll be there, on both your laptop and phone."

Okay, so I was curious about this. I did a bit more digging and found some discussion on a Google blog about Chrome's new password manager. They say: "In a time when most people don't remember more than a handful of phone numbers, can you really be expected to remember a strong, unique password for every online service?" And of course not. That's why we're all using password managers today; right?

They say: "It's no wonder most of us end up using an easy-to-remember password over and over again. But if it gets stolen, as were 3.3 billion credentials last year alone, you're exposed to a much greater risk because now the thieves have a key that works across several sites. So what are you supposed to do? Write them all down? Do the forget/reset shuffle every few weeks? There has to be a better way, and now there is."

They said: "As part of this week's update, we're rolling out significant improvements to Chrome's password manager. Across desktop platforms, and coming soon to mobile apps, we're rolling out unique password generation. Chrome will now recognize a sign-up field, offer you a unique and secure password for that site, and save it. Every password follows these guidelines: at least one lowercase character, at least one uppercase character, and at least one number. If a site requires symbols, we'll include those, too. We'll also avoid certain characters for readability issues" - okay, I'm not sure why that matters, but okay - like a lowercase "l" or uppercase "I," which can be confused visually.

"You can view all your passwords, credit cards, addresses, and other stored information from the main desktop Chrome toolbar. You can also export all your saved passwords into a .csv file at any time." And they finish: "We've also made password autofill even more reliable." Of course we've talked about lots of hacks of autofill where forms are offscreen, and you don't realize your browser has filled them, and then JavaScript running on the page sucks them off the form, and you've just been hacked. Whoops.

Anyway: "Now, Chrome can be used to save or fill in the appropriate password on any site you need. This update would not be possible without significant improvements to the underlying autofill capabilities. When Chrome fills in your passwords, credit cards, addresses, emails, and other types of information, it's backed up by Chrome's multiple security layers and web standards." Well, we're about to hear about a breach of those, but hold on. "And if you're signed into Chrome across your devices, syncing your credentials to your Google Account will allow you to access them wherever you have Chrome installed, laptop or mobile. And if you're using the Android app of your favorite site, your passwords and other information will be there, too."

Okay. Well, of course that's a solution. It doesn't solve the problem of services being compromised and losing their passwords, nor of password recovery hacks, which they talk about the reset and receive a new password problem. Nor local browser extension malware, which did hit Chrome last week. And of course it's not browser agnostic, so it's creating a so-called "walled garden" and requiring that people stay with Chrome. But I would, I mean, I'm not saying it's not a useful step. On the other hand, the MEGA cloud storage extension for Chrome was hacked to do exactly this, to steal login credentials. And Jason, we're at a good point to take a break. I will explain what happened with MEGA cloud storage extension when we come back.

JASON: Yeah. And I should say, like I said, I installed that Chrome update earlier today. And now it's confusing because I have LastPass running as my password manager. So when I go to log into a Google account, it's like the LastPass is behind the autofill suggestions that appear above it. So if my account that I'm trying to log into is the topmost account in my LastPass list, I can't verify which account it is because it's covered up by Google's version of it. And so there's no way around it. And I just don't know, like

being so ingrained in LastPass as a password manager, do you feel confident enough in Google's kind of password management within Chrome by comparison? Or do you think there's still a gap?

Steve: Well, it's single browser; right? So with LastPass we have it under Firefox and across our browsers, across our platforms. You know, Google is multiplatform, so Chrome is all there. The problem is that the browser is the attack surface of the future. I mean, it's the way we reach out onto the Internet. And we've been covering stories about all the malware in the Android Play Store, which Google, despite their best efforts, is always chasing.

We've got problems also with the Chrome browser extensions, and if, for example, this MEGA cloud storage extension was attempting to steal your Google login credentials. So if your Google login credentials get stolen, and they can work around, and there are ways to work around, the various second-factor approaches that Google has used like identifying whether this system or this IP has been seen logging in before, then they have access to this massive repository of everything, your credit card numbers and all of the passwords that have been generated that are being stored in Chrome. So, I mean, there are benefits to a monoculture, and there are risks to a monoculture. And what I'm going to talk about when I get through talking about MEGA is the fundamental problem with this whole model that we'll have fun talking about in a second.

JASON: Absolutely. All right. So that's kind of a teaser forward, before we get there.

Steve: So MEGA is a very popular multiplatform, multi-client, encrypted cloud storage provider service. Last Tuesday, on the 4th of September, MEGA's Chrome Web Store account was compromised and was used, that compromise was used to successfully host a malicious variant of the MEGA Chrome extension. A hacker who goes by the handle Ser, S-E-R, Hack, SerHack, discovered the breach, and he wrote a long posting. The TLDR is: "On 4 September, 14:30 UTC, an unknown attacker managed to hack into MEGA's Google Chrome Web Store account and upload a malicious version 3.39.4 of MEGA's Chrome extension to the Web Store, according to a blog post published by the company.

"Upon installation or auto-update, the malicious extension asked for elevated permissions to access personal information, allowing it to steal login/register credentials from ANY websites like Amazon, Github, and Google, along with online wallets such as MyEtherWallet and MyMonero, and Idex.market cryptocurrency trading platform. The trojanized MEGA extension then sent all the stolen information back to an attacker's server located at megaopac[.]host in Ukraine, which is then used by the attackers to log into the victims' accounts, and also to extract the cryptocurrency private keys to steal users' digital currencies."

MEGA themselves quoted essentially the same and then added, with a slightly more flattering tone, of course, since it was their system, they said: "Four hours after the breach occurred, the trojaned extension was updated by MEGA with a clean version (3.39.5), auto-updating affected installations. Google removed the extension from the Chrome Web Store five hours after the breach." They said: "You are only affected if you had the MEGA Chrome extension installed at the time of the incident, auto-update enabled and you accepted the additional permission request" - and of course users would because they don't know - "or if you freshly installed version 3.39.4," which was the trojanized one.

"Please note that if you visited," they wrote, "any site or made use of another extension that sends plaintext credentials through POST requests" - now, okay, note, that's what everybody does. The connection is encrypted over TLS. But the contents of the post is typically plaintext. No one bothers to encrypt it in the browser before encrypting it under

TLS. So which is to say, I mean, again, they're trying to say, oh, well, there are ways in which you might not get bitten by this. But no. Everybody's going to get bitten by this if they logged in to any site. Anyway, they say: "...either by direct form submission or through a background XMLHttpRequest process," which is a standard XHR Ajax request process. And they say: "While the trojaned extension was active, consider that your credentials were compromised on these sites and/or applications."

So a four-hour Window during which time, if you were using Chrome and had the MEGA Chrome Extension, which would auto-update itself, and that happened, then until that was removed, that trojan was in your browser sending everything that you did back to this server in Ukraine. Also at the time of the attack Monero's Twitter account posted: "The official MEGA extension has been compromised and now includes functionality to steal your Monero," as indeed it did.

Also, Riccardo Spagni, who tweeted from his handle @Fluffypony on Twitter, said the previous owner of MyMonero - oh, he is the previous owner of MyMonero. He tweeted: "Confirmed that it also extracts private keys if you log into MyMonero and/or MyEtherWallet in a browser with this MEGA extension installed." And then further security research confirmed that it could log any POST request where the URL contained specific strings including "login," "register," "sign in," and so forth.

Okay. So we have the Chrome adding its own management of passwords. And here we have an instance where that system collapsed during the time that this malicious extension was in place. So of course this nicely highlights why, while Chrome's autofilling of user passwords with strong passwords is better than not encouraging that, it's still using the existing old model - and for that matter, all password managers do, too - of static one-way passwords which everything demonstrates is no long sufficient protection.

And I don't mean to make this a commercial about SQRL, although the future clearly belongs to a solution like SQRL, if not SQRL. Doesn't necessarily have to be SQRL. But what makes a system like SQRL different is that it assumes the presence of some computational capability at the user's end. When you think about it, that's the difference. Traditional username and password systems only assume some memory from the user, no computation. And what Google has done with this, or what password managers do, they're not changing that. They're not improving that. They're enhancing the memory at the user end, but it's still just memory.

And of course the reason we're where we are is that this all began in a world where we were using terminals connected to mainframes; and the user at a terminal, a so-called "dumb terminal," they were called "dumb" for a reason, they would enter their username and their password, which they had memorized. And then they would log into some mainframe at the other end of the wire. Or later, when we had modems and we were dialing in, same thing. There was just memory on the user's end. Okay. So everything changes, everything changes when you can provide a little bit, doesn't even need to be much, a little bit of computational power, when you can assume that at the user end. And now we all have computational power because we're logging in with a computer to another computer.

So, for example, with SQRL or - and again, I don't want to make this a SQRL commercial, although SQRL has solved all of these problems in a way that is far superior to what anything else has done. With SQRL, the server to which you're authenticating your identity knows your SQRL public key. It has that on file. And it doesn't even need to keep it secret, by the way. That's what it has. It has your public key. And that's all it really needs. When you wish to authenticate your identity to a website, that site sends you a challenge, that is, just some random junk, some gibberish, which it has never sent to anyone before, and it will never send to anyone again. And that's easy to do these days with crypto stuff. So just it sends you this stuff.

That bit of computational capacity at the user's end, which we all now have, not memory, computation, it responds to that challenge by signing it using the matching SQRL private key, and it returns the signed result. The website then uses your public key to verify your signature, and it knows you're you, and no one else. That's all there is to it. And it changes everything. It means that websites no longer have secrets to keep. They can lose their database, and it doesn't matter. And what essentially is a replay attack, if you just have memory, and the browser has stored your username and password, then the bad guy gets it. And they replay that same memory, and they're able to authenticate as you.

What this does, by using a unique challenge and signing it, is you solved the replay attack problem. Nobody who doesn't have your private key, which is very well protected, is able to impersonate you. And even, and think about this, even somebody who can sniff your traffic, who sees what's going on, is unable to gain anything because the website you're authenticating to will never use the same challenge again, so the same signature can never be reused. Anyway, we're like holding on for dear life to a system which is really broken, and which we know now how to fix. And so I guess this is sort of a last gasp on Chrome's part. I mean, to their credit, they are moving forward with other solutions. But it's really time for us to get past this and recognize that, as long as users have some computation, we can completely solve this password problem in a way which is really robust.

So in any event, if you're a MEGA user, you know who you are. Think about what was going on last Tuesday, a week ago. That's when this happened. And the advice from MEGA themselves is any site that you believed you touched and authenticated to during this time when you may have had the malicious extension installed in Chrome, you should absolutely change your username and password immediately. Who knows how much data they collected during this period of time? But I'm sure the folks in Ukraine are taking advantage of it.

JASON: Do you think in MEGA kind of placing blame on Google's shoulders, do you think that's valid?

Steve: Well, yeah. They were complaining that they're no longer able to sign their own code; that Google now does the signing of the code, and they're not able to. It's unfortunate. I don't know why they don't both sign it because there's nothing to prevent double signing of code, which could also work. And we don't also know how this breach occurred. So, you know, was it a weak password? Was it some malware got into MEGA? I mean, unanswered is how this, you know, what the underlying cause of this breach was because we certainly don't want to be in a position where Chrome browser extensions are, like happened with MEGA, are able to get subverted, or we'll be in real trouble.

JASON: Absolutely.

Steve: So we're at Week 3 of the Advanced Local Procedure Call, the ALPC, zero-day. Remember as we talked about two weeks ago, SandboxEscaper tweeted that she was sort of disenchanted with the world. I don't remember her exact words, but she wasn't happy, and she didn't have an interest in submitting another vulnerability to Microsoft, so she just posted it to the world.

We now know, well, and then last week we continued the coverage because the micropatch had been created for somebody who wanted to deal with this immediately, although at the time I said with you, Jason, that I didn't think it was really that big a problem. I mean, it was a problem if malware got in your computer because it could do more with it than if it didn't have a privilege elevation or escalation. But still, I don't know that it would merit like a third-party patch to Windows. That just seemed, eh, a

little bit of a problem. And we talked about how some other researchers had looked into the problem, noticed that it was easy to make it run under the 32-bit version of Win10, and also under Windows 7, without any effort at all.

Well, now we know, thanks to researchers at ESET, that within two days of the original disclosure tweet, this flaw was being used and actively exploited by a group they call the "PowerPool developers" because they use PowerShell-based malware. These guys, they did not use the proof-of-concept code directly. They understood what it was. They made their own. And they attacked, here again, Google Update, which is the official Google Updater app which runs in Windows systems under admin privileges so that it's able to read or write any files on the system.

So what the privilege elevation allowed them was the ability to replace GoogleUpdate.exe with their own malicious version, which the Windows Task Scheduler would then start up as it has been programmed to by the Google Installer, and that would then run their malware, which would install a backdoor which they've been using as the second stage of their attacks. And thanks to what ESET found, this privilege elevation, they were able to do this with full system privileges in order to modify something they shouldn't have been able to modify.

So this is what we're seeing now. It's sort of an interesting evolution in the world, that is, the speed with which this moves. We get an Apache Struts vulnerability. Even if it's been patched, that patch delay, if it's more than a few days, if it's a public vulnerability, as we've seen, it's immediately attacked and jumped on. And in fact we'll be talking about that, too, in a second. Here, within two days, a zero-day that Microsoft has had no opportunity to patch is in use by malware. If it gets into someone's machine, it's then able to get up to much more mischief than it would otherwise be able to. So we're patching now monthly.

I remember thinking at the time, I mean, our listeners remember, before this notion of Patch Tuesday, for a while they were just ad hoc. They were happening whenever Microsoft felt like doing it. That caused the enterprise IT guys a lot of heartburn because these would happen anytime. They needed to make sure that the patches weren't going to upset their enterprise networks and specific software that they were using. So finally Microsoft said, okay, we'll just do these patches on the second Tuesday of every month.

Now we're in a world where, if a zero-day comes out a few days after a Patch Tuesday, unless Microsoft does, and they have in some cases where it's a really bad vulnerability, an out-of-cycle update, you could have a month go by before this gets fixed. And there are situations like we talked about with MikroTik where, unless there's some means for these updates happening essentially autonomously, which is now what the whole world is getting used to with Google automatic updating itself, and Firefox is doing that, and of course Windows and all the OSes are moving in that direction, if they're not already there, that's the world we're in today is an automatic update cycle.

But what that means is that we also have very rapid communication of the presence of flaws. When something is patched, it's reverse-engineered to figure out what the change was, even when it's kept secret; and the bad guys can now figure that out and develop an exploit before the patch, even though it exists, gets widely deployed. So, wow, crazy world that we're in.

JASON: Yeah. They're going to have to turn to neural networks to solve this in a more rapid fashion.

Steve: I don't know what they're going to do. I mean, of course that sort of sounds a little bit like heuristic approaches. And we've seen that, like, AV tries to use behavior-based filters in order to stop things that they don't recognize based on their behavior.

But then those tend to false-positive. You know, you end up with things saying, oh, you've got some malware in your system. And it's like, no, I don't. I know what this program is that you're saying is malware. So not.

I'll just take a break here to bring up two things. I got a tweet or saw a tweet from someone named Daniel, who tweeted as looks like @Gisleburt is his Twitter handle. He was replying to @smolrobots. And because he referred to @SGgrc, he said: "Check out @SGgrc's SpinRite." He says: "It has brought many drives back from the dead, although you should probably replace it immediately even if you get your data back." He says: "It's not free, but there is a 100% money back guarantee." Which of course is true.

And we've talked a lot about whether it's necessary to replace the drive after SpinRite repairs it or not. If you want to err in the direction of caution, if it's easy to replace the drive, then yes. But there are many cases where SpinRite is fixing a specific problem that is really fixed, rather than the drive itself indicating that it's dying, and SpinRite pulling it back from the grave for some length of time before it tries to die again. So as we've said, unfortunately drives are way more analog than we wish they were. So again, erring in the direction of caution, by all means replace the drive after SpinRite makes it readable again. But it's not necessarily the case. If you've been listening to this podcast long enough, you probably have enough tools in your arsenal to know.

And secondly, Jim Sanders, who's apparently in Irvine, California, he wrote, I saw an email from him, on the subject of "Degauss or Not Degauss." And he said: "Steve, I've managed to collect a stack of old spinning hard drives. Normally, I would have DBAN on the drive prior to removing it from my computer. But I'm now looking at a couple dozen needing the treatment." He says: "I've found my old handheld degausser and wondering, if I give the drive 30 seconds or so of treatment back and front, is this as good as DBAN?" He says: "I realize the magnetic field may render the controller useless, but that's fine by me. What think ye?"

Okay. So my answer is I don't think that's sufficiently good. The problem is that magnetic fields fall off surprisingly fast. I mean, you need to have a magnet virtually in contact with the surface of the drive in order for it to have any effect. So if you're on the outside of the drive with a powerful degausser, although you would never want to do this to a drive whose data you cared about, I don't think it's sufficiently good to do it to a drive whose data you absolutely want to make sure you never get back.

So I really - I would take a drill, and I would just drill right through the drive casing several times. DBAN is good. Our listeners know that after I get 6.1, maybe .2 or .3, but at least 6.1 has to happen first, then it's my intention to create something specifically for this purpose, which will be extremely fast and extremely good, which will be known as "Beyond Recall," and it will do what we expect it would do, which is very quickly and securely wipe all the data from a drive. Failing that, or until then, DBAN is a good system, although it is very slow.

I would just drill a hole through the drive a number of times. That's really going to prevent the head from successfully flying over the drive. But I think physically, I think you pretty much have to physically destroy the thing in order to be safe. Being a few inches away, being on the other side of the drive case, I just don't think that's good enough. You really, really need to have intimate contact with the surface of the disk.

JASON: Take advantage of the reason or the fact that you can actually put a drill through it because that's got to be kind of satisfying; right? Take a drill to your hard drive?

Steve: Yeah, well, and depending upon the age, some platters are glass now. And if you smack it hard, if you then hear it, then, you know, smack it hard and then shake it. And if you hear, like, "crinkle crinkle crinkle," you've done your job.

JASON: Right.

Steve: You know that you had a glass platter, and it's no longer going to get any data back.

JASON: It's no longer a platter.

Steve: But if you smack it hard and then don't hear that you've shattered a glass platter, then I think you need to drill some holes through it.

JASON: Lucky you. All right. We've got a few stories to round this out. And this is a really interesting one because it seems to bring, like, a certain payload to people who aren't very creative with their password management.

Steve: Well, yeah. And I'm not sure what this means. So here's the story. A Czechoslovakian court recently sentenced two hackers to three years in prison for accessing Vodafone's customers' mobile accounts and using them to purchase about $27,000 USD worth of gambling services. According to Czech news reporting, the hackers accessed mobile customers' accounts by using the password, get this, "1234." Once they gained access, they ordered new SIM cards that they picked up from various retail branches. Since they knew the phone number and the simple password, 1234, they were able to install the SIM card in their phones without any other verification. This in turn allowed attackers to charge over - and this $27,000 U.S. is 600,000 Czech Koruna, or approximately, as I said, $27,000 U.S. for gambling services.

Okay. So, however, the plot thickens, since Vodafone is saying that it's the customer's fault for having weak passwords, which allowed their accounts to be taken over. Vodafone is claiming that the fault is not theirs, but the fault lies with the customers who chose such weak passwords, and that the hacked customers with easy passwords should have to pay the stolen money back. Some victims of the theft have reported that Vodafone has sent debt collectors to recover the money stolen by the hackers. The victims, on the other hand, have stated, first of all, that they have no idea how their passwords were set to 1234 in the first place, and they also didn't know there was an online market that could be used to buy services that would be charged against their accounts.

Vodafone has stated, get this, that it may have been possible that one of their employees configured this password for the customer when the phone was purchased. But even if so, the user should still have changed it to a stronger password. Okay. Except it turns out that the My Vodafone portal itself only allows four- to six-digit passwords. So Vodafone doesn't have great security. It's not allowing people to set a strong password, four to six digits, which we all know can easily be brute-forced.

So my take on this is that $27,000 cannot be a big deal compared to the reputation damage that Vodafone will suffer in blaming their own customers for hacks of their accounts, when by Vodafone's own admission Vodafone's employee may have set the password to 1234, and it was never changed by their customers. So Vodafone should bite the loss, you know, say, "Okay, fine, it's on us," and then audit their customers' passwords, which they apparently can do, to make sure they're strong, and inform them that they need to change them to something else.

And the portal should disallow 1234 as a password, where if you try to put that in, it's trivial to have some JavaScript running on the page, even if you're going to hash it, and I would imagine they're not even bothering to do that. They could just disallow it, like, right off the bat and say, no, you cannot use 1234, and check for, like, 4321 and 1111 and so forth. But mostly fix it so that you're not allowing short decimal passwords. That's just nuts. Wow.

JASON: Yeah, that really is crazy. That's exactly what I was thinking before. And it's like, if you have these rules, like build it into the tool because we already know people are not very adept at passwords. And path of least resistance is commonly the approach that's taken with a number of people and passwords. So don't allow them to take the easy way, and put those barriers in place, especially if you're going to just charge them if something bad happens. That's just bad business.

Steve: Well, and also probably the focus is the phone; right? So you buy a phone, it comes with a My Vodafone portal, which you may never even use. You may know that it's there. So the employee sets it up as a password 1234, and may have even said, okay, I've set the password up on your My Vodafone portal to 1234. Make sure you change it. But they have the phone. The phone is what they want. They probably maybe never even logged in. And the point of this is, when we're using a web service, like online banking, it's the service that we're trying to secure. So there, your authentication to the portal makes sense that you would want that to be secure. Whereas someone getting a phone, they're just thinking about their phone. They want the phone to work. They may never even log into their portal.

So, yeah. I mean, although the interesting precedent and the worry that this sort of suggests is are we going to see more of this kind of behavior from other people besides Vodafone? Traditionally, when a user has lost money due to their account being hacked because they used "monkey" as their password, and their debit card was grabbed, and their banking account was emptied, it's like, oh, too bad, it's on you; right? So here we're seeing exactly the same model, but the user is running up a bill on their account that they're refusing to pay, and Vodafone is saying, no, your account got hacked, you're responsible for the fact that bad guys got a SIM which they hooked to your phone number, thus to your account, and then ran up gambling charges. You don't like the bill that you received, but we're going to send debt collectors after you in order to collect it.

So I don't know. We've often talked about how weird it is that in software the user accepts all the blame for whatever happens, even if the system hurts them. It'll be interesting to see whether legislation begins to take the position of corporations that are trying to hold users responsible.

And speaking of which, Tesla gets hacking exactly right. They updated their formal product security page and eliminated all doubt. I've said on the podcast, our listeners have heard me say it over and over, that we absolutely have to hold valid, legitimate, non-hacking, white hat researchers unresponsible, I mean, not legally responsible. Do not sue the guys who are trying to improve your product. We've talked about a lot of stories in the past where that was done. So Tesla updated their pages.

They said, under product security: "Tesla values the work done by security researchers in improving the security of our products and service offerings. We are committed to working with this community to verify, reproduce, and respond to legitimate reported vulnerabilities. We encourage the community to participate in our responsible reporting process." Bravo. "To register as a pre-approved, good-faith security researcher and register a vehicle as a research-registered vehicle, please submit requests to VulnerabilityReporting@tesla.com."

They said, for vehicle or product related services: "While we use Bugcrowd as a platform for rewarding all issues, please report vehicle and product-related issues directly to vulnerability@teslamotors.com, using our GPG key to encrypt reports containing sensitive information."

They said, under third-party bugs: "If issues reported to our bug bounty program affect a third-party library, external project, or another vendor, Tesla reserves the right to forward details of the issue to that party without further discussion with the researcher.

We will do our best to coordinate and communicate with researchers through this process."

And finally, and here it is. Actually, there's this and, yeah, their responsible disclosure guidelines: "We will investigate legitimate reports and make every effort to quickly correct any vulnerability. To encourage responsible reporting, we will not take legal action against you, nor ask law enforcement to investigate you, provided you comply with the following Responsible Disclosure Guidelines."

Here they are: "Provide details of the vulnerability, including information needed to reproduce and validate the vulnerability and a Proof of Concept. Make a good faith effort to avoid privacy violations, destruction of data, and interruption or degradation of our services. Do not modify or access data that does not belong to you. Give Tesla a reasonable time to correct the issue before making any information public. Alter only vehicles that you own or have permission to access. Do not compromise the safety of the vehicle or expose others to an unsafe condition. Security research is limited to the security mechanisms of the Infotainment binaries, Gateway binaries, and Autopilot ECU binaries."

And then they said, for the avoidance of doubt, and I really love that, I mean, they like absolutely want to clarify this. "If, through your good-faith security research, you (a pre-approved, good-faith security researcher) cause a software issue that requires your research-registered vehicle to be updated or reflashed, as an act of goodwill Tesla shall make reasonable efforts to update or reflash Tesla software on the research-registered vehicle by over-the-air update, offering assistance at a service center to restore the vehicle's software using our standard service tools, or other actions we deem appropriate.

"Tesla has complete discretion as to the software or other assistance that we will provide, and it may be only a limited number of times. Tesla's support does not extend to any out-of-pocket expenses, for example towing, incurred by you. Tesla reserves the right to limit the number of service requests per pre-approved, good-faith researcher, and unregister a research-registered vehicle at any time."

So in other words, if you brick your Tesla, and you're behaving well, you demonstrate good faith, you're registered, some of your research caused your car to die, they'll fix it for you. They're not saying they're going to always fix it for you. They're not going to fix it a hundred times if you keep doing it. On the other hand, if you keep finding problems, I imagine they're going to say, oh, okay, find some more.

But still, they said: "Tesla considers that a pre-approved, good-faith security researcher who complies with this policy to access a computer on a research-registered vehicle has not accessed a computer without authorization or exceeded authorized access under the Computer Fraud and Abuse Act (CFAA). Tesla will not bring a copyright infringement claim under the Digital Millennium Copyright Act" - the infamous DMCA - "against a pre-approved, good-faith security researcher who circumvents security mechanisms, so long as the researcher does not access any other code or binaries. Tesla will not consider software changes, as a result of good-faith security research performed by a good-faith security researcher, to a security-registered vehicle to void the vehicle warranty of the security-registered vehicle, notwithstanding that any damage to the car resulting from any software modifications will not be covered by Tesla under the vehicle warranty."

So bravo to Tesla. They clearly explained that researchers will be held harmless when they respond in a responsible disclosure fashion. And I can't imagine anybody could have a complaint with that. And it turns out this is in the nick of time because today Wired magazine reported that yesterday hackers revealed at the Cryptographic Hardware and Embedded Systems conference in Amsterdam that they had found a way around the

original encryption used in the key fobs of Tesla Model S cars, using a few hundred dollars in radio and computing equipment and some one-time preparation, which I'll explain in a second.

They are now able to wirelessly read signals from a nearby Tesla owner's fob, spend less than two seconds - actually it's 1.6 seconds - of computation to yield the fob's cryptographic key and essentially completely clone the security; to defeat the security, unlock the doors, and drive away. Lennert Wouters, who is one of the KU Leuven researchers, at the KU Leuven University in Belgium, yesterday said: "Today it's very easy for us to clone these key fobs in a matter of seconds. We can completely impersonate the key fob and open and drive the vehicle."

So: "Two weeks ago Tesla rolled out new antitheft features for the Model S that include the ability to set a PIN code that someone must enter on the dashboard display to drive the car. Tesla also says that Model S units sold after June of this year [so new ones] are not vulnerable to the attack, due to upgraded key fob encryption that has already been implemented in response," so in response to the disclosed responsibly to Tesla ahead of time KU Leuven research. "But if owners of a Model S manufactured before then" - and that's Leo, I think; right? I think Leo has a Model S; right?

JASON: I think he had the Model X.

Steve: Oh, the X, okay. If the "owners of a Model S manufactured before don't turn on that PIN, or don't pay to replace their key fob with a more strongly encrypted version," and that's certainly what I would do, "the researchers say they're still vulnerable to their key-cloning method." In other words, everybody needs to know you can put a PIN on, or you can pay to get upgraded encrypted key fobs. But if you don't do that on a Model S purchased before June of 2018, you're vulnerable.

So what do we know? Like most automotive keyless entry systems, Tesla Model S key fobs send an encrypted code, based on a secret cryptographic key, to a car's radios to trigger it to unlock and disable its immobilizer, allowing the car's engine to start. After nine months of on-and-off reverse engineering work - apparently they didn't work on it full-time, they just worked on it when they had time - this research team discovered last summer, that is, the summer of 2017, that the Model S keyless entry system, built by a manufacturer named Pektron, P-E-K-T-R-O-N, used only a weak 40-bit cipher to encrypt the key fob codes.

And the problem is a 40-bit key just doesn't have enough combinations. It's just too easy to brute-force it. The researchers found that once they gained two codes from any given key fob, they could brute-force every possible candidate cryptographic key until they found the one that unlocked the car. So two codes from the fob, and then computation. That narrowed down the universe of possible codes because there are only 2^40 of them, that they were able to unlock the car. Then, having accomplished that first piece of research, that taught them exactly what was going on.

So they then precomputed all possible keys for any combination of code pairs, which created a massive 6TB lookup table of pre-computed keys. Armed with that table, any two codes received from any new key allowed them within 1.6 seconds, a little bit of computation, and then a lookup using their 6TB lookup table, which of course these days 6TB is portable, allowed them to determine the key of the fob in 1.6 seconds, and it allowed them to clone it.

So responsibly disclosed to Tesla. Tesla made the changes, began selling them, updated the firmware in the cars to add a PIN, which would defeat this attack for non-upgraded systems, and then the researchers disclosed it all yesterday at the conference in Amsterdam. So that's the way security is supposed to work. What of course this also

says is that users are responsible for setting a PIN, or paying to get an updated fob, or recognizing that if they happen to be a target of an attack, again, this is one of those things where, once we know it can be done, it's probably - and I haven't looked at the report to see how much detail they went into. But I imagine, given the typical security report of what they did, all the details are available now publicly for someone to easily retrace their footsteps. So Model S owners definitely want to take heed.

JASON: I've always been a little nervous about key fobs just in general, but I'm sure they're a lot more safe than...

Steve: I don't have one. I don't have one on my car. I'm old school. So, yeah.

JASON: I can understand that. I guess there's always the, what is it, the clothes hanger, you know. The analog solution still exists, too.

Steve: If you have buttons that you can hook around.

JASON: That's true.

Steve: Fortunately, my car has smooth buttons. In fact, I don't even know if they work. They're just sort of a visual indicator of whether the door is locked or unlocked. I don't think I've ever tried pushing down on it, come to think of it. It's amazing.

Okay. So we talked about PCs becoming a battleground. We also talked about this latest Apache Struts 2 vulnerability, which allowed a single-URL query to a vulnerable Apache Struts web server to compromise and take over the system. Thanks to research by F5 Labs, we now know they have caught this being exploited in the wild, and in at least one case have a complete breakdown of what's going on. There is a Monero cryptomining campaign which is attacking Apache Struts 2 vulnerable systems. They named this the CroniX, C-R-O-N-I-X, because it uses the cron service in Linux for persistence and the Xhide capability for launching executables with fake process names. So because of cron and Xhide they named it CroniX.

The main target appears to be Linux systems, but there is some involvement of Windows because it also has a Visual Basic component. So even though they are largely seeing Linux being attacked, it looks like the bad guys gave some thought to also attacking Windows. It uses a single URL which is able to embed something known as OGNL, which is the Object-Graph Navigation Language, which is supported by these servers.

And I have in the show notes a screenshot of the query. It's a GET query, which is to say it's a standard URL which is crazy, but you can see in there cmd= and then 'wget and then a user-agent, looks like it's just "l," and then a few other parameters, and then the URL. And it's using wget to grab update.sh and then piping it into bash. So it's grabbing this update.sh shell code and then executing it on the system in order to take it to the next level. And then bash gets routed to dev/null, so it just goes into hyperspace. And then in the screen shot you can see a lot of other stuff going on.

So the attacker sets the number of so-called "huge pages" in memory to 128. And the F5 guys say this is the first clue that the attacker's intention could be a mining operation, as this step is probably related to improving mining performance. It then sets up a number of cron jobs for malware persistence. And again, it primarily looks like it's targeting Linux servers, although there is a Windows component.

So what they found most interesting is that it is the most aggressive anti-competition malware that they have seen. I mean, it's not uncommon for malware to boot other malware out, and especially mining malware, because of course they want all of the system's CPU resources for their own mining. They don't want to share the CPU with

some other guy who's also trying to mine cryptocurrency on the same machine. So they go in, and they kill a bunch of processes by name of known malware. But then, because some mining malware is known to use legitimate process names, they watch the system's processes run and selectively kill off processes utilizing 60% or more CPU resources.

Specifically crond, sshd, and syslogs are the processes which they've seen other malware masquerading under. But when you see something like the cron daemon or ssh daemon using 60% of the system, that's going to be unusual. So very likely other malware has gotten in ahead of it. So it goes and kills them and then deletes the binaries from the system in order to set up shop and essentially take over. And once it's all settled in, it mines Monero using a mining pool at eu.minerpool.pw and uses as much of the resources of the system as it can get, staying in the system as long as possible. And remember, unlike the first Apache Struts problem earlier this year, which affected a huge number of servers, this one, because it requires some non-default settings, it's probably going to be less prevalent, but it has been found actively pursuing systems in the wild.

So it's certainly not without concern. And as we said, it's good at this point that all these guys are doing is wanting to set up cryptomining operations rather than wondering whose network they are in and turning around and taking a look at it. Unfortunately, as we said earlier in this podcast, it's not the case that the MikroTik router folks are so lucky.

JASON: And that's why it's probably only a matter of time anyways.

Steve: Yes, exactly. And I'm a big believer or a big fan of OpenVPN. It is open source. It is a robust solution. I just think it is, rather than using some third-party solution, it's like these guys are on it. They are being very careful as they evolve it. And this problem is solved. You want to use UDP as the transport because TCP doesn't like being used as the carrier for TCP as the protocol. You want a non-reliable protocol to carry the reliable protocol because most of us are going to be using TCP conversations. That wants to be encapsulated in a UDP tunnel. OpenVPN does both. UDP will give you better performance.

The problem is that - and I don't know what the OpenVPN guys will do. I hope they step up to address this. There are instances where you can add too many features. And I would argue OpenVPN probably did. In the OpenVPN docs, under "Using Alternative Authentication Methods," they said: "OpenVPN 2.0 and later include a feature that allows the OpenVPN server to securely obtain a username and password from a connecting client, and to use that information as a basis for authenticating the client."

And I should mention Leo and I have talked about this before. We use certificates because that's belt and suspenders. That's better than just username and passwords. But they go on, saying: "To use this authentication method, first add the auth-user-pass directive to the client configuration. It will direct the OpenVPN client [at the user end] to query the user for a username/password, passing it on to the server over the secure TLS channel." So that's cool. They're establishing a secure channel. The config file prompts the user for the username and password if auth-user-pass directive is present in the client configuration.

Then they said: "Next, configure the server to use an authentication plugin, which may be a script, a shared object, or DLL. The OpenVPN server will call the plugin every time a VPN client tries to connect, passing it the username/password entered on the client. The authentication plugin can control whether or not the OpenVPN server allows the client to connect by returning a failure or success value."

So here's the problem. The plugin is specified by a user-writeable configuration file; and the OpenVPN service, which runs under the system account, loads and executes the optional authentication plugin within its own process and privileges. In other words, we have a breach of security boundary as a consequence of the fact that the config file can be changed by someone with non-system privilege, but you can use it to specify a DLL which will get loaded and run by the server with system privileges.

So we know that that's a recipe for disaster. It turns out that both the ProtonVPN and the NordVPN both based on OpenVPN, for which I salute them, were found to be vulnerable to exploitation of this trick. They quickly patched it. But then security researchers looked more closely and realized that, other than just this authentication plugin, also the other commands which were valid for the OpenVPN config, "script-security," "up," and "down," could also be used as command vectors. So they did another round of patches and updated.

So the good news is I don't know where other OpenVPN systems stand, but they're going to want to make sure that they're patched and updated with the revelation from this. It is the case that ProtonVPN and NordVPN have been notified, fixed it, were notified that wasn't a full fix. They've updated again. So if you are, to our listeners, if you're a ProtonVPN or NordVPN user, you're going to want to make sure that you have updated yourself to the latest version of those systems so that you're safe.

And one last little piece of sort of fun here. Trend Micro got themselves in trouble. They're in the doghouse. Yesterday, in response to a firestorm of controversy which erupted over the weekend, Trend Micro posted under - their posting was "Answers to Your Questions on Our Apps in the Mac App Store." They began: "Reports that Trend Micro is 'stealing user data' and selling them to an unidentified server in China are absolutely false." Then ZDNet, in covering this, their headline starts with: "Trend Micro says sorry after apps grabbed Mac browser history. The company has now removed a browser history data collection feature from its macOS products."

Okay. So what happened? We know Trend Micro. They're a well-known security firm. They've apologized after it turns out, and they've admitted, that several of their consumer macOS antimalware products and non-antimalware utilities were discovered to be capturing the user's browser history data and sending it to a remote server. Turns out the remote server, it's not exactly clear where this China idea came from. Apparently it's an AWS server based in the U.S. The Trend Micro apps which have been removed include Dr. Cleaner, Dr. Cleaner Pro, Dr. Antivirus, and Dr. Unarchiver. So Dr. Unarchiver, why does that have anything to do with snapping browser history?

So what Trend Micro explains is, they said: "Trend Micro has completed an initial investigation of a privacy concern related to some of its macOS consumer products. The results confirm that Dr. Cleaner, Dr. Cleaner Pro, Dr. Antivirus, Dr. Unarchiver, Dr. Battery, and Dr." - oh, I'm sorry, not Dr. Duplicate Finder, just Duplicate Finder. I guess they didn't get their doctorate on duplicate finding - "collected and uploaded a small snapshot" - this is Trend Micro saying "a small snapshot of the browser history on a one-time basis, covering the 24 hours prior to installation.

"This was a one-time data collection, done for security purposes, to analyze whether a user had recently encountered adware or other threats, and to improve the product and service." Huh. Like maybe trying to determine why you were installing Trend? I don't know. One of their doctors? Anyway: "The potential collection and use of browser history data was explicitly [I love this] was explicitly disclosed in the applicable EULAs and data collection disclosures accepted by users for each product at installation." And then they say: "See, for example, the Dr. Cleaner data collection disclosure here," and then a link, which I have in the show notes if anyone's interested.

"The browser history data was uploaded to a U.S.-based server hosted by AWS and managed and controlled by Trend Micro. Trend Micro is taking customer concerns seriously and has decided to remove this browser collection capability from the products at issue." And of course why were Dr. Unarchiver and Dr. Battery collecting browser history? And what I really wonder is if those EULAs also made this data disclosure, which they claim was being accepted by users.

Anyway, what they've said was: "We have learned that browser collection functionality was designed in, common across a few of our applications, and then deployed the same way for both security-oriented as well as non-security-oriented apps such as the ones in discussion. This," they say, "has been corrected." So, okay. They use some sort of common library. Maybe it was part of the install process. Who knows? Which is why the Dr. Unarchiver and the Dr. Battery were also collecting browser history.

Anyway, they've all been booted from the macOS store. I imagine they'll get this fixed and then probably come creeping back in and hopefully do a better job in the future. So for what it's worth, if you're wondering where they went, that's where and why. Not a big deal, probably. But still it's good that we've got, you know, thank goodness we've got people looking, watching this, catching this behavior. I love that we have an industry full of researchers who are being third-parties who are looking at this stuff, and that we have an environment that doesn't prosecute third parties for saying, I don't think you meant to be doing this on your battery app.

JASON: Yeah, that's true. Everybody in chat is pointing out maybe if the software has "Dr." in the name, maybe that's a good indication to look elsewhere. Maybe it's making promises that it can't quite hold up to. And like that, it's also like Cleaner, I know on Android there's a bunch of Cleaner apps. And oftentimes with those there's other stuff happening.

Steve: They're kind of low end, yeah.

JASON: Yeah. Yeah, totally. So anyways, I've never actually known if Trend Micro was like a brand that consumers could actually trust. They were just a brand that was always kind of on my radar, I think, personally.

Steve: Yeah, yeah. They don't seem like a high-end brand.

JASON: I wouldn't say so. Steve, you've done it again. I think you've covered, well, you've covered a good chunk of the security news. I'm sure there were other stories out there, but they obviously did not rise to the top this week.

Steve: Well, and there were some that were breaking as we were going to press. So I've already got a few lined up for next week's coverage. So fear not. If there's something you think I missed, I'll probably be talking about it next week.

JASON: GRC.com, if you want to check in on all the things Steve offers and is keeping track of. Obviously you talked a little bit earlier about SpinRite, the best hard drive recovery and maintenance tool. You can go to GRC.com and get your copy there.

You also talked about SQRL today, which by the way, it's your show. You can talk about SQRL anytime you ever want, seriously. And it's important to know about. You can find information on SQRL by going to GRC.com as well, along with audio and video of the show. There's transcripts. That's the only place you can go to find transcripts of this show. You can find it there at GRC.com. Steve, really appreciate you letting me tag along this week. Thank you.

Steve: Hey, a pleasure. And we will do our third and final show together next week.

JASON: That's right.

Steve: Looking forward to it, Jason.

JASON: That's right. And in the meantime, TWiT.tv/sn for all of the previous episodes to subscribe everything. And of course next week you're talking about Tuesday, that's 1:30 p.m. Pacific, 4:30 p.m. Eastern, 20:30 UTC. And that's it for this week. I'm Jason Howell. Steve Gibson, thank you again. We'll see you all next week on another episode of Security Now!.

Steve: Thanks, buddy.