New WiFi Password Attack

Description: This week we discuss yet another new and diabolical router hack and attack, Reddit's discovery of SMS 2FA failure, WannaCry refusing to die, law enforcement's ample unused forensic resources, a new and very clever BGP-based attack, Windows 10 update dissatisfaction, and Google advancing their state-sponsored attack notifications. We ask, "What is Google's Project Dragonfly?" We go over a highly effective and highly targeted ransomware campaign, present some closing-the-loop feedback from our listeners, and reveal a breakthrough in hacking/attacking WiFi passwords.

High quality (64 kbps) mp3 audio file URL: http://media.GRC.com/sn/SN-675.mp3
Quarter size (16 kbps) mp3 audio file URL: http://media.GRC.com/sn/sn-675-lq.mp3

SHOW TEASE: It's time for Security Now! with Steve Gibson. WannaCry just won't stop crying. We'll talk about a new router attack and how people hate Windows Update. Plus Steve will detail a new and a little bit scary WiFi password attack. It's all coming up next on Security Now!.

Leo Laporte: This is Security Now! with Steve Gibson, Episode 675, recorded Tuesday, August 7th, 2018: New WiFi Password Attack.

It's time for Security Now!. Yes, the moment you've all been waiting for. Once a week, everybody gets...

Steve Gibson: Those of you who are listening may have been waiting, yes.

Leo: That's Steve Gibson of the GRC Corporation. Hi, Steve. Good to see you again.

Steve: Leo.

Leo: Sorry I missed last week.

Steve: Great to be with you again. Jason did a nice job of MC'ing last week, and I guess we're going to have him through most of September.

Leo: Yeah, that's right, because we're leaving for a vacation at the end of the month.

Steve: Yeah, I've been noticing you recording extra content for the Sunday show.

Leo: The Tech Guy, yeah, yeah.

Steve: In order to fill in. So, cool.

Leo: Catching up, getting ahead, I guess. So thank you, Jason. And here I am. I'm back. Did anything happen while I was gone?

Steve: Yeah. Bluetooth got hacked.

Leo: I heard about that, yeah.

Steve: But aside from that. Oh, and I did want to mention that I had a nice conversation with Stina.

Leo: I was worried about her.

Steve: Yeah, and not without reason. Our conversation was off the record. And she had said that she was going to be meeting with Google at the end of last week and then would have an update for me, and they're still sort of trying to determine what Yubico's official stance should be. And I wasn't ready to assume that this was done without their knowledge.

Leo: Let me give people the back story. I know you talked about it last week. But if people didn't listen, Google - and it was actually really good news - announced that 85,000 of their employees have been provided with YubiKeys, those hardware-based one-time password, used for two-factor authentication on Google accounts. And in the year and a half or two years they've been using them, zero successful phishing attacks. It was very effective. They followed that announcement, good news, Stina was probably jumping up and down for that, with the announcement that they were going to do their own hardware key, the Titan.

Steve: Yup.

Leo: And that's what I was worried about. Now, I have to point out that Titan is not only a hardware key that you plug in the USB port, but it's got Bluetooth in it. And with this news about Bluetooth snarfing, I'm wondering if that's a good thing.

Steve: Well, the other thing to wonder is whether having the secrets installed in it in China is a good thing.

Leo: Ooh, I didn't hear that.

Steve: Yeah, they - yeah, yeah.

Leo: Hmm. They're making it in China. Hmm.

Steve: Yeah, by the low bidder. So anyway...

Leo: So Stina is - and you met her at an RSA conference some years ago, just by chance. She was going up the down escalator, and you ran into her.

Steve: Yeah, well, it was interesting. She said to me when we began our conversation last Monday, she said it had been 10 years. So it was 10 years ago. I was there with Security Now! press credentials, just covering the conference. And on the third day I'd seen everything and looked through all the booths and attended a bunch of the various presentations. And everyone was tired, and I was just sort of - I remember I was, like, walking toward the escalators on the upper level of the convention center, and there was this attractive blond Swedish woman with badges and stuff. And I was, what, so this was - oh, and Elaine told me, by the way, I had lost a year in where we are with the podcast. This is not Year 12 that we're finishing. This is Year 13 that we're finishing.

Leo: Right, right, right, right. We just finished Year 12.

Steve: Correct. No, we're...

Leo: We're just starting 14. That's right, yeah.

Steve: Yes, will be. I think it's like mid-August, like in a couple weeks. Anyway, so we'd been doing the podcast, you and I, for about four years.

Leo: It was Security Now! 143, I'm told, that you told this story. In fact, it's called "YubiKey."

Steve: Yes. Well, and so what happened was - you know, so you and I had been doing the podcast for, like, four years. And we'd been talking about the eBay, the PayPal football, and time-based one-time passwords, and all of this. I mean, obviously, authentication and multifactor stuff is crucial.

So she says to me, are you interested in authentication? And, I mean, she had no idea who I was, just I'm some random guy with press credentials hanging around my neck. And so I said yeah. And so she says - she, like, pulls this thing out of her pocket, and it was clearly a USB - it had the four USB fingers, and it was clearly a USB plug. She said, "This is a one-time password token that emulates a keyboard." And I just went <gasp>. I mean, because I got it. And it was like, oh, my god, it was just so clever, the idea that by emulating a keyboard you needed no drivers, and you could just plug this into the

computer. It got power from the computer, so it didn't need to have a battery. And when you touched the little touch-sensitive metal button on it, it would spit out a string of characters.

Leo: I think the original one didn't even have - it wasn't a - wrote to anything. It was just one string, wasn't it?

Steve: No, no.

Leo: It changed every time.

Steve: It did have a...

Leo: Oh, okay. It does now, anyway.

Steve: ...little bit of permanent memory.

Leo: Right.

Steve: The beginning chunk of the string was static, which identified you. And then the rest of it changed every time. So you had to stay - and it was counter-based, not time-based.

Leo: That's right. That's right.

Steve: So it had to stay in sync with an authentication server. Anyway, so it took her a while to believe that some random press guy instantly understood what they had. But of course I did because I'd been living in this space for quite a while. Anyway, and so it was three weeks later - because I had to do a lot of RSA news after the conference. We did several podcasts on what I saw and what was announced and so forth during the RSA convention. And then I said, "And now let's talk about something that is really cool."

Anyway, that was 10 years ago. So I think the sad thing is that Stina and Yubico really have been prime movers of this whole area. I mean, I know a lot of behind-the-scenes stuff about FIDO and UAF and U2F and what's been going on and what a political mess it has been. So for 10 years she's been pushing this. And they moved to Silicon Valley to be in the Valley, and Google has been one of her major accomplishments. And politics.

So anyway, I looked on Yubico's site for any updated response. I don't know if there will be one. I didn't see any yesterday. But anyway, for what it's worth, it'll be interesting to see how this plays out. I think Yubico...

Leo: And it sounds like we're going to continue to support Yubico as the right choice; right?

Steve: Yes. Yes. And certainly there are many organizations that won't choose Google because they're Google. I mean, you know, many people will. But in the same way, for example, that you can't buy a Kindle book on the iOS app for Amazon, it's like, which is so dumb, so you have to go to the browser and browse to the Amazon website to buy a Kindle book because Apple wants to make iBooks, not Kindle books. And so there are, you know, we see all kinds of places where corporations have their own political issues with each other. So Yubico is still, I mean, my absolute goto solution for this and for other future things that may be coming down the pike.

So anyway, this week 675 is our episode number for the first Tuesday of the month, which is the latest Tuesday we can have. So second Tuesday of the month, which will be Patch Tuesday next week, will be the 14th, which is as far back in the month as it's possible to have that happen. So I'm sure we'll have stuff to talk about then.

In this case, there is a new attack on WiFi passwords which occurred to the author of Hashcat, which is the very cool high-performance hash-cracking tool that is able to enlist the aid of GPUs and gets really high hash-based cracking rates. He was poking around at WPA3, the forthcoming next version of WPA. Everyone knows we're at WPA2 now, the beleaguered WiFi security specification. And we were talking about WPA3 a few weeks ago, that, well, eventually it'll come out. I was excited about it because I got teased by the fact that it looked like the specs were online, but all they were was like the Table of Contents. And it's like, na na na. It's like, okay, fine. So I don't get to take a look at it yet.

But the problem is that, unless all of the existing devices are retrofit, then it doesn't do us a lot of good, even when WPA3 certified routers eventually occur. It isn't clear that WPA2 routers will be able to upgrade their firmware to WPA3. I don't know one way or the other. But the Wi-Fi Alliance is all about certification and stamps and trademarks and nonsense. There should be none of that in a protocol as important to the world as wireless Ethernet access. But it is. That's where we are.

But even if routers could get updated, or when they're eventually replaced, well, all the other devices need to be updated, too. And it's not clear when IoT light bulbs are going to get themselves updated. So we're stuck with WPA2 for the foreseeable future. What the author of Hashcat tripped over, I mean, like this has been sitting here for years and no one noticed it, is a significant advancement in I guess we'd call it the state of the art in WiFi network password cracking, which makes it enough easier that everyone's going to be doing it.

Leo: Oh, boy.

Steve: So, yeah. That's the topic of today's podcast. But we've got a lot more to talk about. We've got another new and diabolical router hack and attack. Reddit's discovery that SMS-based two-factor authentication is insufficient. Oh, who knew? Well, of course all of our listeners knew. WannaCry refuses to die and knocked a major, well, actually, the largest company in Taiwan off of its manufacturing cycle for a weekend. We have an interesting piece of research, a study that was done about law enforcement's use of forensic resources that has some surprises we'll talk about. And Bruce Schneier weighed in on that, as well. A new and very clever BGP-based attack. We were talking about how that Portuguese ISP was finally frozen off the Internet after years of selling BGP chunks of IPv4 space they didn't own to spammers. This is different and diabolical, again. These attacks are getting more clever.

An interesting survey was done. Lawrence Abrams, who's the founder of Bleeping Computer, has a good friend who's a longstanding MVP of Microsoft, who did a survey,

took a survey of Windows 10 Update actually sort of dissatisfaction. A snap from Bleeping Computer's coverage is the Picture of the Week, and we'll talk about that just briefly because, as a consequence of this, it bears on us because people are disabling it, which of course ends up having a security consequence.

Leo: Oh, boy. Yeah.

Steve: Google has advanced their state-sponsored attack notifications in an interesting way. Also The Intercept got wind of Google's new Project Dragonfly, which I'm sure you and your crew will be talking about tomorrow with Jeff and Stacey. We have a highly targeted and amazingly powerful ransomware campaign, some closing-the-loop feedback from our listeners, and then we're going to talk about essentially something that's always been possible, nobody noticed, in the existing WPA2 protocol, not WPA3. The Hashcat guy acknowledges that, if it ever happens, they finally really did solve this problem, probably. But it's going to be a long time coming. And in the meantime, hacking WPA, you know, WiFi access point passwords just got a lot easier.

Leo: Oh, boy.

Steve: So I think another great podcast.

Leo: <Growling> Geez. Well, I can't wait to hear how much easier. That's going to be germane to the conversation.

Steve: Yeah. So our Picture of the Week ties into the story we'll get to a little bit later. So we could come back to it, but it's probably not surprising to any of our listeners. The numbers are interesting. It would be nice to imagine that, I mean, Microsoft must be aware of this. I imagine you'll be talking with Paul and Mary Jo about this tomorrow also, Leo, because it was an interesting story where this Microsoft MVP took a survey of I think 1,800 people, a thousand in enterprise and 800 users. And nobody was real happy with the outcome of what Microsoft was doing. For example...

Leo: This is about Windows Update.

Steve: Exactly.

Leo: And they're just not very satisfied with the experience.

Steve: Exactly. Well, and for example, I guess, and I wasn't aware of this, fortunately, because I'm not in Windows 10 world, I guess last month, and you probably know because of Windows Weekly with Paul and Mary Jo, I guess July was really rough for side effects and problems with Windows 10 updates.

Leo: Yeah. They pushed out 1803, and I think that was the biggest problem.

Steve: Well, and so, for example, satisfaction with Microsoft patching, very much not - it's ranked from one through five. Very much not satisfied was almost 32%; not satisfied, 37%; neutral, 13%; somewhat satisfied, 16%; and very satisfied was 2.28%. So satisfaction with the quality of the updates pretty much follows the same pattern, with very satisfied at 3.54%, everybody else very much not satisfied, not, or neutral, or somewhat. Useful to my business: 35 not useful at all, 35 rarely useful, 19 neutral, 10% somewhat useful, and only 1.96% said extremely useful. And the last question was on the issue of feature releases. 40% said once every two years. Another 40% said once every year. So there's a total of 80% of the respondents wanted it no more than once a year, and half of them every two years. 8.89% had no opinion. Two times a year is fine said 11.2%. And then somewhere there's 1.42% that wants them even more often.

So anyway, we'll loop back around to this a little bit just when I talk about what Larry and Susan found. But in the meantime we have, you know, I guess we sort of seem to go through themes or phases or periods with the podcast. We are now in the hacking publicly facing routers phase of the industry as reflected by this podcast. Trustwave had a blog posting. SpiderLabs is Trustwave's security side. A Simon Kenin did a blog posting where, coming back from the Asian RSA conference recently, something jumped on his radar. He wrote in the first person.

He said: "On July 31st, just after getting back to the office from my talk at RSA Asia 2018 about how cybercriminals use cryptocurrencies for their malicious activities," he says, "I noticed a huge surge of Coinhive in Brazil." Okay, now, remember, Coinhive is the server which serves the browser-based, and now we know it's based on the native assembly code for browsers, so it's browser based, relatively high speed, as much as you can get in a browser, cryptocurrency mining. They mine Monero currency. And we've been talking about them for months because of course bad guys have been arranging to get unwitting - either apps on Android have been downloading cryptocurrency miners, or they've been injecting them into browsers.

Coinhive, remember, it's a little bit shady because we know that there's no way they don't know their service is being abused. And also remember they're getting a fair chunk of coin themselves for everything that's being mined because essentially they take a large piece of the action. So they've sort of barely skated along. Their domain has been blacklisted. Third-party anti-mining plug-ins exist now to block them. So, I mean, I don't think they're even in the gray zone. They're in the dark gray zone of offering a service.

So anyway, so he notices a huge surge of Coinhive in Brazil. He writes: "After a quick look, I saw that this is not your average garden variety website compromise, but that these were all MikroTik" - and I'm not going to pronounce it "microtic" this time, though I'm a little tempted to, as we'll see here in a minute. They were all MikroTik network devices.

He writes: "This could be a bizarre coincidence, but on further inspection I saw that all of these devices were using the same Coinhive site key, meaning that they all ultimately mine into the hands of one entity." He wrote: "I looked for the Coinhive site key used on those devices and saw that the attacker indeed mainly focused on Brazil. My first thought was that on such a large scale that could be a zero-day exploit, possibly in the MikroTik HttpProxy component, so my next step was to check whether anyone else also noticed this, since during the conference I had limited time and Internet access to keep up with daily news." So he sort of was coming into this, he felt, late and thought, well, maybe this has been going on for a while, and somebody else has mentioned it.

He says: "Google didn't produce many results, but the few that did come up were actually quite useful in helping me pinpoint the attack vector and what the attacker did. For example, this result shows injection of Coinhive on a hospital website in Brazil.

However, this web server runs Apache, which contradicted my initial thought of an exploit directly on MikroTik HttpProxy."

He says: "After doing some querying on Shodan, I actually found the hospital's MikroTik device, so perhaps it is an issue with MikroTik, but not necessarily with the HttpProxy. So there I was," he writes, "back at square one with a huge surge of Coinhive hits in Brazil, but no idea where and how it originated, and back to my Google results. I went to see what else they had to offer." He says: "I found a post on Reddit from someone who was repeatedly being infected with Coinhive mining."

Anyway, so I'll stop quoting him there and just say that what he found ultimately was that this attack started in Brazil. It has since spread globally. More than 200,000 MikroTik routers are infected. The bad guys are using, not a zero day, as one might expect, but this is what we've been talking about now for quite a while is that what I guess the hacker community has awoken to over the last few months is that routers are not being patched. And so old exploits which have long since had patches available are virtually never being deployed, never being installed.

And so it is worth looking for instances of publicly facing routers still that are remaining vulnerable years after the exploit was published. Well, in this case, it isn't years. But on April 22nd MikroTik was informed of a remote authentication bypass exploit on virtually all of their routers. And to their credit, one day later they had a patch. So April 23rd of this year, many months ago, this was fixed. Did it matter? No.

Leo: Of course not, because you can't patch - they're not auto patching. You have to know.

Steve: Right, exactly.

Leo: That's the problem with these older routers. You know, when MikroTik had its first problem, we didn't know how to pronounce MikroTik. We've learned now.

Steve: Right, right.

Leo: Quite a good opportunity to learn.

Steve: So today, hundreds of thousands of these MikroTik routers remain vulnerable. Now, in this case, MikroTik offers a custom UI which they call "Winbox," which is a Windows-based management interface which the MikroTik routers offer. I only thought of MikroTik routers as little consumer devices because I remember when I was in love with that amazingly inexpensive...

Leo: The Ubiquiti? Yeah, the EdgeRouter X?

Steve: Thank you, the Ubiquiti EdgeRouter X. Several of our listeners said, hey, what about MikroTik? And sure enough, MikroTik has some little routers. Turns out they also make what I would be tempted to call "big iron" routers. I mean, they make routers that ISPs use.

Leo: They're Latvian.

Steve: Yes, exactly.

Leo: They're from Latvia.

Steve: They are.

Leo: Well, there you go.

Steve: Okay. So get this. Here's the new diabolical bit. Rather than running the Coinhive miner on the router itself, which is how the exploit was apparently first operating when it was discovered, the attacker has since started using the router's intermediate position on any network where it's located. After all, I mean, it is going to be on the boundary between the WAN and the LAN. That's how the routers are being deployed. It is now dynamically injecting live Coinhive script into every web page that a user visits.

Leo: Oh, lord. Oh, my god.

Steve: And it's bidirectional. Not only will it inject Coinhive script into routers behind, like on the LAN behind the router, but in the case of servers running that have public-facing websites, it's injecting live Coinhive script onto the customer's router, that is, onto the customer's browser, I mean, onto the visitor's browser. So people out on the Internet who are going to a website hosted by someone whose ISP has a MikroTik router on its edge is getting infected with Coinhive.

Leo: Now, does that happen if it's an HTTPS site? I mean, wouldn't that prevent that?

Steve: No, no, and that's exactly right. So HTTPS cannot get intercepted. And this is what was a surprise to me, and again another - that's a perfect question, Leo. The attacker was apparently worried that it would be too obvious if all pages were being intercepted. And maybe it's because of the inability to intercept HTTPS. So it turns out they're intercepting error pages returned to the router's users. And those error pages are probably not HTTPS. So this guy observed that, for example, like a 404 Page Not Found, or a DNS lookup error from an ISP, those are probably not HTTPS. So they are nonencrypted page opportunities to run the script on people's browsers. I don't think the script's going to run very long. No one's going to sit there with an error page on their web browser for, like, hours at a time. But that's how this is happening.

And he finishes his posting. Simon says: "Let me emphasize how bad this attack is. The attacker wisely thought that, instead of infecting small sites with few visitors, or finding sophisticated ways to run malware on end-user computers, they would go straight to the source: carrier-grade router devices." In other words, MikroTik is sourcing ISP devices. He says: "There are hundreds of thousands of these devices around the globe, in use by ISPs and different organizations and businesses. Each device serves at least tens, if not hundreds, of users daily."

Leo: Ooh.

Steve: So by being an intermediate Coinhive injector, it's able to get broad visibility for its script. He says: "Allegedly" - not quite English - "each user would have initially gotten the Coinhive script regardless which site they visited. Even if this attack only works on pages that return errors" - and I think that's probably because those are not HTTPS - he says, "we're still talking about potentially millions of daily pages for the attacker." Although I would argue probably not each running for very long.

So anyway, just once again, here we have, as you said, Leo, as we've been saying, these consumer routers are not self-updating. And that behavior has to change.

Leo: Got to change. Well, I have to say it is mostly changing. All the good new gear does.

Steve: Yes.

Leo: Thank god. And you shouldn't even consider one that doesn't because you're not going to update it. How often do you check?

Steve: Yeah. I mean, how would you know?

Leo: Right.

Steve: I mean, yeah. You normally just get the thing from Amazon or from your local retailer and plug it in, and you forget about it.

Leo: Right.

Steve: And you don't register it. You don't want to give anybody else your email address because it's just more opportunity for spam. And so it's like, okay, fine. So they can't contact you. The router itself needs to phone home. And part of the agreement needs to be that it may take itself offline for 10 minutes at 3:00 a.m., when nothing is going on, in order to update itself.

Leo: It's also why - and this is bothering people, but I don't mind it - you're starting to see companies like Eero and Plume ask in the former case and demand in the latter case a subscription fee. You're paying an ongoing $90 a year subscription fee. And part of what you get for that is updated services and patches. And I don't think that's unfair, I really don't. I think that that's part of the guarantee. Although, oh, and also remember when we were talking about HTTPS Everywhere, and I was a little miffed at Google because they were pushing this so hard? Well, now, that's the answer to the question, why do you have to do this? That's exactly the answer. HTTPS Everywhere would prevent this. At least this one, anyway.

Steve: Yeah. So Reddit got hacked, despite having two-factor authentication in place. They posted: "We had a security incident. Here's what you need to know." Which was their posting. And they said: "What happened? On June 19th we learned that, between June 14 and June 18, an attacker compromised a few of our employees' accounts with our cloud and source code hosting providers. Already having our primary access points for code and infrastructure behind strong authentication requiring two-factor authentication, we learned that SMS-based authentication is not nearly as secure as we would hope, and the main attack was via SMS intercept. We point this out to encourage everyone here to move to token-based two-factor authentication." And of course this is something that we've been talking about for some time.

I'll just finish a little bit more, saying: "Although this was a serious attack, the attacker did not gain write access to Reddit systems; they gained read-only access to some systems that contained backup data, source code, and other logs. They were not able to alter Reddit information, and we have taken steps since the event to further lock down and rotate all production secrets and API keys, and to enhance our logging and monitoring systems."

So anyway, I heard you, I guess it was - I don't think it was last weekend, but maybe the weekend before, or during the week, you were talking a lot about Yubico and YubiKey and Google's Titan stuff. And I thought you made the very, very good point, which is that the problem even - first of all, we completely agree that SMS-based two-factor authentication is a problem. I think it was when I was setting up my Hover account. Several years ago I was wanting to migrate away from Network Solutions finally because they had become so evil. I chose Hover as my next registrar. They offer two-factor authentication. And they gave me a choice: SMS or time-based.

And we had been talking about how insecure the cell phone system, the cellular network fundamentally is. And I knew that the advantage of a time-based token is that only one time I needed to have a secret pass between them and me. And this is where I was then promoting the idea of printing the QR code right there and then, when it's shown to me, just print it. So I have it on paper, and that way I'm able to easily clone my time-based one-time password to other devices.

Anyway, so that was years ago. And we talked then about the reason that you want time-based six-digit changing tokens, not per-use. And so that's what bit these guys. Anyway, when you were talking about it, Leo, you observed, and I appreciated that you had, the fact that, unfortunately, even, I mean, every single time, everywhere you see the opportunity to use multifactor authentication...

Leo: It's a fallback.

Steve: ...is still, yes, oh, I don't have it with me. Or, oh, I left it at home. Or my dog ate it. Or whatever. And it's like, okay. So we're back to square one.

Leo: It's the lowest common denominator. Whatever the, you know.

Steve: Exactly.

Leo: I respect Google because Google does have a fallback if you're not going to use your YubiKey. But it's the authenticator. And that may be a setting. I can't remember. Maybe you could say don't use SMS. But, yeah, whenever possible, try to

avoid that. And my bank, every bank, SMS is always an option. And, I mean, to be fair, it is a little more secure; right? I mean a lot more secure because it's not - it's a nontrivial thing to steal somebody's SMS number.

Steve: Absolutely. Well, and the advantage of it is it requires zero user preparation.

Leo: Right.

Steve: So it's like, oh, my bank just sent me this code. Fine. And so you just enter it in. And so, I mean, the problem is that usernames and passwords also require zero user preparation. And look at what a disaster that is. So the problem with time-based is you then need - you need to have an app running on something that's able...

Leo: You've got to prepare, yeah. You've got to set it up.

Steve: Exactly, requires some preparation. And I'll just add a little plug for SQRL here that we'll be talking about in the future. SQRL already in the protocol has two settings in it where you can, once you become comfortable with how the system works, and you understand what's going on, you can turn on its request to every site you visit to disable all recovery options. And you can also turn on its request to disable all non-SQRL authentication. So there are two different settings.

Leo: Hmm, I like that one.

Steve: And there are flags that it just - it just sends the flags out whenever you use it. And so it's up to the web server, it's up to the site to decide if they wish to honor it. But it indicates the user's request that essentially says, "I understand what I'm doing. I don't want recourse. No recourse." And so, as we know, it's the only way you can really have security. And those flags are off by default because I don't want anyone to hurt themselves. But it's built into the protocol. You can say, you know, "I'm serious about my security. I'm taking responsibility for it. Cancel all recourse." And that's safe to do because the SQRL system itself provides lots of recourse to prevent you from getting yourselves in trouble.

Leo: Very nice. Very nice.

Steve: Yeah. Anyway, we talked about Stina. We talked about Google's Titan keys. Stina had said she was going to get back to me prior to this podcast, but I didn't bug her. She was in New York when I talked to her last Monday, and then she was going to be - this came as a surprise to them. So I think they're going to be fine. I think that there's a huge non-Google world, and Yubico is the company you want to use. They're really staying with it. There are some things I can't talk about yet, but they've got a bright future.

Leo: Shhh. Good, good.

Steve: Yeah.

Leo: On we go.

Steve: So WannaCry...

Leo: Oh, man.

Steve: ...refuses to die.

Leo: This is the last year version, too.

Steve: Gosh, yeah. So as we'll remember, WannaCry was the weaponization of a flaw in Version 2 of Windows SMB, the Server Message Block, also known as Windows file-and-printer sharing, that was believed - well, we know that it was leaked by the Shadow Brokers. We believe it was an internal NSA exploit originally named EternalBlue. So EternalBlue was taken and weaponized as WannaCry, which was a very, very destructive crypto malware.

Leo: I think EternalBlue was the thing that let it spread over the network; right? That was the...

Steve: The SMB, exactly. It was the propagation mechanism.

Leo: Here's my T-shirt. I tell you, if you go up to a girl, and you say "ITPro.TV and Chill?" and she doesn't say yes, she's not the right one. I'm just saying. No, all right, okay.

Steve: The good news is that, with that T-shirt, she can really see you coming.

Leo: It's a little red. All right. Sorry.

Steve: So it struck again over the weekend. It's expected to shave a quarter billion dollars from a major Taiwanese semiconductor manufacturer's revenue as a consequence of it bringing down, bringing to a halt, multiple semiconductor manufacturing fabrication facilities across Taiwan. The company is cleverly named Taiwan Semiconductor Manufacturing Company.

Leo: We just call it TSMC.

Steve: TSMC. Not only is it a major supplier of parts for Apple, AMD, Nvidia, and Qualcomm, but it's Taiwan's largest company. So big guys.

Leo: I didn't know that. Wow. They make the A11 chip for the iPhones and iPads.

Steve: Right, right. And so there was some concern that this would be a supply chain glitch for maybe Apple and others. It's unclear. I mean, it was relatively short. I mean, they were able to get things back up. But they wrote: "This virus outbreak occurred due to misoperation during the software installation process for a new tool, which caused a virus to spread once the tool was connected to the company's computer network."

So, yeah. So it's taken a chunk out of their revenue, although they're a major company. They're forecasting revenue in the quarter to be around - their original forecast was $8.45 billion. No. It was 8.55. So it's expected to drop by at least 200 million to 8.45. So anyway, so they're saying the revenue hit could be as high as 256 million, in other words a quarter billion, but still they're 8.5 billion, so they'll be okay.

But what's interesting is it's not clear how this happened because it didn't just spread. It destroyed their systems. It encrypted their systems. And we'll remember that it was our friend of the podcast Marcus Hutchins, a.k.a. MalwareTech, who was looking at this immediately upon its appearance on the Internet and saw in the code that for some reason this WannaCry malware was checking for - it was making a DNS query, but the DNS domain that it was querying was unregistered. So all of these queries were coming back as no such domain name.

So he thought, huh. Wonder what that's about? And he registered the domain name. And immediately WannaCry stopped. And so what we surmise is that this was a deliberate kill switch which was put into the malware by its authors as a means for them to shut it down if for some reason they wanted to. Well, it's still in place. And as we know, I coined the term Internet Background Radiation (IBR) to talk about the fact that many of these worms, Code Red and Nimda, still live on some systems; and they're out there pinging around, looking for new victims. Similarly, WannaCry is still out looking for other systems to encrypt. But its check for that DNS domain is succeeding, which prevents it from doing so.

So we don't know about the details of this Taiwanese manufacturer TSMC's internal network. But it might be that they've got strict controls over what DNS queries are allowed to be made to public DNS servers, or they might have some sort of border protection that's preventing all but some explicitly whitelisted domains. We don't know. But what we do know based on the information we have is that, if this was WannaCry, it would not have done anything if it had been able to get resolution of that DNS. So maybe it was a variation of WannaCry that has had the kill switch neutered so as not to use it any longer. No one knows really what happened. But anyway, that's the story is that this thing, WannaCry, is still delivering its destructive payload wherever it can, in at least this instance.

There was an interesting report from the Center for Strategic and International Studies titled "Low-Hanging Fruit: Evidence-Based Solutions to the Digital Evidence Challenge." And the reactions to this report have been many. Essentially it turns out that, not surprisingly, James Comey's replacement, Christopher Wray, who's now the head of the FBI, is continuing to threaten legislative action if Silicon Valley companies do not offer up some means for responding to a search warrant.

He recently said: "I think there should be room for compromise. I don't want to characterize private conversations we're having with people in the industry. We're not there yet for sure. And if we can't get there, there may be other remedies, like legislation, that would have to come to bear." He says: "We're a country that has unbelievable innovation. We put a man on the moon. We have the power of flight. We

have autonomous vehicles. The idea that we can't solve this problem as a society," he says, "I just don't buy it."

So Bruce Schneier weighed in on this. Bruce's blog was titled: "New Report on Police Digital Forensics Techniques." And he refers to this report from the CSIS, the Center for Strategic & International Studies. I've got a link to it. He's got a link to it in his blog posting. I have a link to his blog posting and also to the study. But I'll just share the top of this because it gives it some context, and sort of it's enough. So the study reads - this is the low-hanging fruit.

It says: "Over the past year, we conducted a series of interviews with federal, state, and local law enforcement officials, attorneys, service providers, and civil society groups. We also commissioned a survey of law enforcement officers from across the country to better understand the full range of difficulties they are facing in accessing and using digital evidence in their cases. Survey results indicate that accessing data from service providers, much of which is not encrypted, is the biggest problem that law enforcement currently faces in leveraging digital evidence.

"This is a problem," they write, "that has not received adequate attention or resources to date. An array of federal and state training centers, crime labs, and other efforts have arisen to help fill the gaps; but they are able to fill only a fraction of the need. And there is no central entity responsible for monitoring these efforts, taking stock of the demand, and providing the assistance needed. The key federal entity with an explicit mission to assist state and local law enforcement with their digital evidence needs is called the National Domestic Communications Assistance Center (NDCAC). It has a budget of $11.4 million, spread among several different programs designed to distribute knowledge about service providers' policies and products, develop and share technical tools, and train law enforcement on new services and technologies, among other initiatives.

"In addition to bemoaning the lack of guidance and help from tech companies - a quarter of survey respondents said their top issue was convincing companies to hand over suspects' data - law enforcement officials also reported receiving barely any digital evidence training. Local police said they'd received only 10 hours of training in the past 12 months, state police received 13, and federal officials received 16. A plurality of respondents said they only received annual training. Only 16% said their organizations scheduled training sessions at least twice per year."

So anyway, all of this essentially reduces to the FBI and other law enforcement agencies are annoyed that they are unable to get everything that they want. That's what they're complaining about, this whole Internet going dark problem; while at the same time, based on this survey, they're not nearly making use of everything they can already have access to. So I thought this was useful and important. And it is generating a lot of attention. And it's surprising people that essentially what we are continuing hearing is this drumbeat of we need absolute decryption of all information. Yet it turns out that there is in fact a large body of evidence and digital information which is readily available, not encrypted, and not being taken advantage of.

So anyway, I just thought that was - Bruce essentially weighed in with the same position, that what's happening is law enforcement is not nearly taking advantage of what's available. So maybe we need to worry a little bit less about, or they need to worry a little bit less about having decrypted access to everything until such point as they're taking advantage of what they do have access to.

This is, oh, very, very clever on the topic of BGP, the Border Gateway Protocol, and hijacking of routing. What we're seeing is evidence now of border gateway protocol hijacking becoming targeted and weaponized. We talked a couple weeks ago about Bitcanal, which was the Portuguese ISP who had for years been poisoning the BGP

propagation system by advertising routes to a large number of little tiny /24, that is, 256 IP networks. Just basically sort of nibbling off the edges of other people's larger IP allocations and commandeering the routes for those small bits of networks.

This works because routers will always route to the largest networks possible. And so if you claim to own a little piece of network, then you end up being able to capture that traffic, even if somebody else is advertising a route that includes yours. And this allowed them to then resell or lease those IP blocks to spammers, who were able to thus avoid spam blacklists.

Oracle did some coverage of this, almost a chilling, for its cleverness, new attack, a new style of attack. They said: "In April of 2018 we detailed a brazen BGP hijack of Amazon's authoritative DNS service in order to redirect users of a cryptocurrency wallet service to a fraudulent website ready to steal their money." Okay. So that's sort of the classic attack, where you attack BGP.

In this case, though, what's interesting is that they attacked the DNS, that is, they rerouted Amazon's authoritative DNS service to a spoofed DNS server, which would then offer up the wrong IP to people wanting to go to the cryptocurrency wallet service. So that's sort of an indirection; that is, rather than directly rerouting the traffic to the wallet service, they commandeered the DNS that served the wallet service and rerouted that. Well, that has some interesting implications that I'll explain in a second.

Then they said: "In the past month we've observed additional BGP hijacks of authoritative DNS servers with a technique similar to what was used in April. This time the targets included U.S. payment processing companies." Specifically they found three different payment companies. And I'll just finish with one last paragraph. They said: "As in the Amazon case, these more recent BGP hijacks enabled imposter DNS servers to return forged DNS responses, misleading unsuspecting users to malicious sites." Here's where it gets very diabolical. "By using long TTL values in the forged responses, recursive DNS servers held these bogus DNS entries in their caches long after the BGP hijack had disappeared, maximizing the duration of the attack."

Okay. So that is what's so chillingly clever. So, okay. The authoritative DNS server is the actual anchor DNS for the service. So, for example, in GRC's case I use Level 3's network. And there are a pair of Level 3 servers that are authoritative for GRC.com, meaning that anyone out on the Internet asks their - well, first they ask their own computer, their own local workstation, for the name. If it doesn't have it, then it asks the DNS server that it's configured to query.

Well, that is a recursive DNS server, meaning that the user asks it, and then that DNS server, which is what we're normally just used to referring to as like our ISP's DNS server, or Google, or OpenDNS, you know, one of those. It's given the job to do the full resolution task, which is with DNS it can be a recursive process. So when it ultimately gets the answer, since DNS is a caching system, part of the answer it gets - and it will ultimately contact the authoritative DNS server to obtain the DNS record.

In the example here, it would contact one of the two Level 3 servers and get the record that answers its question about GRC.com. Part of the record is the caching information, how long I, as the admin of GRC.com, want the Internet to remember GRC's IP. This is handy because longer TTLs, longer times to live for the cached information, lessens the burden on the authoritative DNS server. But it means that, for example, if I wanted some agility of my IP, I wouldn't be able to change IPs quickly. So one of the things that could be done, if I were planning to change my IP, is to reduce the DNS cache duration so that queries start coming more often. And that way, when I do change my IP, the rest of the cached Internet will update with that new information more quickly.

Anyway, so what these guys have done is they've weaponized DNS caching by creating a spoofed DNS server which has the wrong, malicious IPs for a given, in this case, three different payment processing companies. And super long cache times. I can't remember now what the longest DNS cache is. But, I mean, it can be many, many, many days. So they set up these spoofed DNS servers; put very long cache times in the DNS. And DNS, the recursive servers, will sometimes not obey very short caching. As far as I know, they do obey very long caching.

So that means that just a comparatively brief BGP hijack that is redirecting traffic from the true authoritative DNS server to their spoofed authoritative DNS server, during the period of that hijack, any existing caches which expire out on the Internet in recursive DNS servers will ask for an update of the IP for the now spoofed payment service. They will get the wrong IP with a super long cache value.

And notice it's not just the user who made that request. But what's happened is it's poisoned the cache for the recursive DNS server that might be used by all the customers of a given ISP. So it can have long-lasting, very broad and sweeping implications. We're talking about a very clever weaponized attack that uses BGP as one of its parts, and then a long cached DNS record to put the cherry on top, or the icing on the cake.

So anyway, Oracle concluded their posting of this security news by saying: "If previous attacks were shots across the bow" - that is, BGP attacks - "these incidents show the Internet infrastructure is now taking direct hits." And they write: "Unfortunately, there is no reason not to expect to see more of these types of attacks against the Internet in the future." And this is an example of where DNS is being exploited where the only solution that we have for this exploitation is and will be the use of DNSSEC. DNSSEC would securely sign the records, and there would not be a way in the instance of this kind of rerouting spoofing, for the spoofer to spoof the records, as long as the recipient knew that the site records were signed by DNSSEC. And in fact the authoritative record for that domain would specify that in this case GRC.com, for example, that the GRC.com domain was supporting DNSSEC.

So as we also talked about recently, the progress toward DNSSEC stalled about five years ago and is not moving forward slowly. I don't know how this ultimately happens. It's just going to take time. But until that happens, we're vulnerable to this kind of exploitation.

And I had here in the notes this comment about the issue of poor Windows 10 Update experiences. I think we've pretty much covered it. Lawrence Abrams of Bleeping Computer posted his note about Susan Bradley, who's a well-known Microsoft MVP. She's been an MVP for 18 years, polled a whole bunch of people. She put together an open letter to Microsoft executives Satya Nadella, Carlos Picoto, and Scott Guthrie about the frustration that Windows 10 users have when dealing with installing new updates and all of the problems that are known to happen.

Unfortunately, despite the fact that she's apparently well known, it doesn't look like her open letter got to anybody. Lawrence posted two days later. The first posting was on August 1st, last Wednesday. And then two days later on Friday, on the 3rd, he posted a follow-up, essentially sharing the very disappointing reply. Basically it looks like just a Microsoft email-reading functionary responded with boilerplate, saying, you know, thank you very much for your reply. Windows 10 is unlike other Windows operating systems, blah blah blah. I mean, it wasn't responsive in any way. Doesn't look like it came to anyone's attention.

So anyway, I just thought it was interesting from a security standpoint because she said in her letter: "It's due to increasing frustration with patching and patch management issues." She says: "I see consultants turning off updates completely as they see that as

the only way they can have stable systems." She wrote to them: "I see server admins saying that Server 2016 is fine except when you want a system that doesn't reboot."

She says: "I see Surface users get their machines cratered with 1803 side effects" - exactly as you were saying, Leo - "with SSD drives." He wrote: "Susan told me via email, 'I see more and more people say that waiting to install updates is what one must do. All of these are unacceptable. We can't continue on with the status quo.'" So anyway, we'll keep an eye out, I'm sure, to see whether Microsoft responds to this because, as we know, it is burdensome for users; and, unfortunately, as a consequence of mistakes it's causing problems.

Back in 2012, so six years ago, Google began warning its Gmail users - and I wasn't aware of this until they made this change to G Suite, which is their enterprise level offering. I wasn't aware that they had been alerting users of Gmail to state-sponsored attacks upon their account.

Leo: Oh, yeah. I've gotten them, yeah.

Steve: No kidding. So like a red warning bar saying...

Leo: Yeah. Let me see if I can...

Steve: ...that your account is under attack, we believe from a state sponsor?

Leo: Yeah.

Steve: Wow. Anyway, that's very cool. Last week they announced that administrators of G Suite accounts would now be able to enable and configure a special alert when government-backed state espionage groups are trying to hack into one of their company's user's accounts. So I just wanted to let our users know, any of our listeners who are G Suite admins or maybe employees of companies who are using G Suite, to consider turning that on. I don't know if it's enabled by default.

But under the Admin Console > Reports > Manage Alerts there is a new option, Government-Based Attack, which can be enabled by the super admin of the account. It allows the admin to be alerted. Apparently the user's account under attack can automatically be locked and have the password account reset process initiated. And the system is also able to alert the end user as well as the admin, optionally. So it's very cool that Google is doing that, and not something that I was aware of.

And, well, this doesn't directly bear on security for us, although we've recently been talking a lot about the consequences of nation-level censorship: Telegram fighting Russia we've talked about a lot; Apple moving its Chinese users' data to China and handing over control of the data center to China's primary government-owned telecom company; and we've also been talking about Chinese censorship. What came to light recently from documents leaked to The Intercept was that Google is planning to make a comeback in China with a fully censored search engine offering through a project known as Dragonfly.

The Intercept said: "Google is planning to launch a censored version of its search engine in China that will blacklist websites and search terms about human rights, democracy, religion, and peaceful protest," The Intercept wrote. They said: "The project, code-named

Dragonfly, has been underway since spring of last year, and accelerated following a December 2017 meeting between Google's CEO Sundar Pichai and a top Chinese government official, according to internal Google documents and people familiar with the plans."

So I won't go any further, but just to say that, as we know, Google was censoring, what, from - for the last eight years they've been dark. And for the two years prior they were censoring. So from 2008 to 2010 they were doing some censoring. Then they just decided to pull up roots altogether. And now they're deciding to go back. And again, as we know, ultimately I think, I mean, I understand people are upset about this. People feel that it's Google capitulating to authoritarian dictator government. But if you want to operate within a country, you need to abide by the country's requirements. And I think that's what we're going to see as the Internet matures and sort of moves from being a startup experimental network to a mature global network forever.

And lastly, before we do our last break and then talk about this new way to hack WiFi, an interesting new approach for ransomware which has netted its hackers or attackers $6 million, which is a little bit surprising. It is targeting attacks on large companies.

Sophos has been following these guys for quite some time. They said: "As the year 2016 began, a ransomware threat appeared that attacked its victims unlike any previous ransomware attack. It's named SamSam after the filename of the earliest sample that we," they write, Sophos, "uncovered." And they say: "It uses a brutally minimalist manual approach to target and compromise victims. The attacker or attackers use a variety of built-in Windows tools to escalate their own privileges, then scan the network for valuable targets. They want credentials whose privileges will let them copy their ransomware payload to every machine - servers, endpoints, or whatever else they can get their hands on.

"Once in, the attackers spread a payload laterally through the network, a sleeper cell that lays in wait," they write, "for instructions to begin encrypting." They say: "Ever a predator, the attacker waits until late at night, when the target organization is least well equipped to deal with it, before the final blow is struck. In a sneak attack while the target literally sleeps, SamSam encrypts a prioritized list of files and directories first, then everything else." So, okay. We know that in the past ransomware has been sort of a scattershot, opportunistic attack. This is different. This is establishing essentially an advanced persistent threat, but using ransomware as the attack vector.

They said: "Unlike virtually every other ransomware attack, the entire attack process is manual, no badly worded spam email with an attachment. The attacker breaks in the old-fashioned way, using tools that attempt as many logins as quickly as the Remote Desktop Protocol will permit" - so this leverages exposed RDP, which is something that I've also encountered in other reading recently - "and exploits operating system vulnerabilities, though not as many as you'd think. SamSam usually succeeds when the victim chooses a weak, easily guessed password."

And they say: "In this report" - and I've got a link, by the way, to the full report because I'm just going to cover that and their key findings - "we'll cover the anatomy of a SamSam attack, and why it isn't necessarily hard to defend against. We also took a deep dive into the ransomware payload, tracing its evolution from an early beta through its (so far) third major revision, with no sign of a slowdown in sight, and an ever-increasing ransom demand with each subsequent attack. Partnering with the crypto monitoring firm Neutrino, we traced the money trail and discovered far more victims and funds than had been previously reported."

Okay. So key findings of this. SamSam has earned its creators more than $5.9 million since 2015. 74% of the known victims are based in the U.S. Other regions that are

known to have suffered attacks are Canada and the U.K. and the Middle East. The largest ransom paid by an individual victim so far was $64,000. So this is not your one bitcoin attack, or your fractional bitcoin attack, because they are, by the way, they are using cryptocurrency in order to extract ransom. This is $64,000 for an enterprise.

So what's interesting about this new model is that they're getting in and doing devastating damage to a significant enterprise. And in return for that, they are getting major payouts. And again, if the enterprise is fully protected against every nook and cranny being lost, then they're able to recover from their backups. But if they're not, then you can well imagine that they're going to arrange to first get proof of decryptability, and then these bad guys are only asking for a single large payout. So that changes the problem of small people not having data that's valuable, or having a backup for their own workstation, or being unable to deal with the hurdle of making a cryptocurrency payment. So anyway, I think this is interestingly and diabolically clever.

Sophos also said that: "Medium- to large-scale public organizations in healthcare, education, and government have been targeted by SamSam." They said: "Our research discovered that these only make up for about 50% of the total number of identified victims, with the rest comprising a private sector that has remained uncharacteristically quiet about the attacks." Yeah, because no company wants to admit that they've had any kind of a penetration of their network.

"The attacker uses care in target selection, and attack preparation is meticulous. SamSam waits for an opportune moment, typically launching the encryption commands in the middle of the night or the early hours of the morning of the victim's local time zone, when most users and admins would be asleep. Unlike most other ransomware, SamSam encrypts not only document files, images, and other personal or work data, but also configuration and data files required to run applications, for example, Microsoft Office. Victims whose backup strategy only protects the user's documents and files won't be able to recover a machine without reimaging it first."

They said: "Every subsequent attack shows a progression in sophistication and an increasing awareness by the entity controlling SamSam of operational security. The cost victims are charged in ransom has increased dramatically, and the tempo of attacks shows no sign of slowdown." So this is ransomware attacks taken to the next level. And again, sort of foreseeable if you had, okay, if we get into a network of a major corporation, and our goal is to get paid, how do we generate the largest possible incentive for cash? In the past it's been doing damage like was done with Sony, with an advanced persistent threat.

But these guys want money. And so doing reversible critical damage on a massive scale to an organization and then using cryptocurrency payment is the way you do that. And that's exactly what Sophos has found with this SamSam attack. Yikes. You don't want to be the target of that.

Leo: Unh-unh.

Steve: So I ran across a couple of, I thought, interesting closing-the-loop bits, feedback from our listeners. Stuart in Seattle, his subject was "A Flash use case." And he said: "Long time listener/SpinRite user, et cetera." Now, Flash, we're not talking about flash memory. We're talking about the evil Adobe Flash interpreter that's been the source of so many problems. He says: "I keep hearing you slam Flash. At work we have a use case for Flash, and as of now there is no workaround. We use it to provide a dynamic UI in a PDF file" - oh, that's what you want is Flash in a PDF. When does that ever cause anybody trouble? Anyway, sorry.

Leo: Double trouble.

Steve: Yeah. "We use it to provide a dynamic UI in a PDF file that contains 3D work instructions. It is simply not possible to do this using PDF forms constructs. I agree that, for HTML content, Flash should go away. Unfortunately, with a PDF file, there isn't an alternative at this time. We are in a tricky spot of needing to come up with a different way of presenting the work instructions, then going through the expensive process of getting the new way certified for production." So I'm not sure what certification Stuart's talking about. But that's interesting. Unfortunately, Flash embedded in PDFs is like the worst attack vector there is. I mean, it's like one place Flash can still live and PDFs carry something malicious.

So, yikes. It'll be interesting to see what happens. I mean, I guess PDF support for Flash will go away. Wow, yeah. I don't know, Stuart. Anyway, I just wanted to share that, that there are places where it's not easy to say no. But ultimately, in a couple years, "no" is going to get said.

Blitz in Phoenix wrote: "You can kill HTTP when you pry it from my cold dead hands." He said: "Hey, Steve." And then he said in brackets: "<insert Love the show, you guys are great, lots of praise/>." He says: "I want to bring up one specific scenario where I am actively uncomfortable with HTTPS." And I kind of agree with him here.

"When you join a network, and it wants you to accept a terms of service or otherwise authenticate me via a web form. When you go to a website, a captive portal intercepts the connection and returns a page for you to login/agree/whatever. That's fine when the connection is straight up HTTP because you can 'man in the middle' an HTTP. But when the connection is HTTPS, you get a certificate error. Okay, so I could create an exception for that site and NOT store it permanently, but that feels wrong. In addition" - and as you mentioned, Leo - "if everyone goes full bore HSTS" - that is, HTTPS Everywhere, essentially, HTTP Strict Transport Security is HSTS - "doesn't that mean you can no longer bypass a certificate error? Also, as someone who has to do a lot of network-related troubleshooting, I greatly appreciate the ability to turn off SSL during packet captures as it makes it much easier to analyze."

He says: "Oh, and for anyone who finds themselves hitting a captive portal, and they don't want to accept the certificate being presented, try http://neverssl.com." He says: "That site is so simple - pronounced 'elegant' - and straightforward in its design and explanation that I immediately suspected it could be a Steve Gibson creation." Well, that's flattering, thank you.

He says: "I've come to the conclusion that it is not one of yours" - no, it's not - "but primarily because I would have expected you to mention it at some point if you had created it." He says: "<insert Thanks for the show/SpinRite/Vitamin D info/>." He says: "Looking forward to how you handle the four-digit episode issue. Instead of Y2K, would that be the E1K issue?" Oh, Episode 1K. Yeah, that's right, when we cross out of 999. Anyway, signed off Blitz from Phoenix.

Leo: The E1K.

Steve: I'll just say that, yes, it is a problem when, I mean, this is like a problem more with Chrome. Firefox is still making it a little bit easier to accept an exception. But, boy, Chrome is getting really intolerant of non-HTTPS. And there are places where just doing web management, when you're talking to your local router or you have a reason for

needing to go to something that is not HTTPS, and they remain present, it is in fact becoming increasingly difficult to do that as everything moves to HTTPS. So maybe we'll end up needing to have some access to routers that give us a way to do that, that we only use for network management purposes.

And then this was interesting, Leo. Cory Bitney in Kalispell, Montana said: "Rogue admin! Help!" And he said: "Hey, Steve. First of all, I'm a huge fan of Security Now! and SpinRite; so thank you for providing both. The reason I am writing is I'm in a bit of a conundrum. I just took over as systems administrator for a new company, replacing a sole admin that initially left the company, after seven years, on good terms." Okay. So I guess the company's not new. He's now newly there, a new employee, just took over as systems admin for a new company, meaning, yes, his new employment, replacing a sole administrator that initially left the company after seven years on good terms.

"At the end of week two I found that he installed crypto mining software, MinerGate, on all of the workstations on the domain, as well as a few servers. I told the owners, and they are concerned, but are hesitant to file criminal charges because they've known him for so long. They at least finally gave me permission to do a full lockdown a few days ago. And the more I dig, the more I find - backdoors, monitoring software, et cetera."

Leo: Oh, boy.

Steve: "Basically everything in a sysadmin's nightmare." And Cory says: "He has definitely abused his power. My question is, if the owners don't press charges, do you know of anything I can do to report him? The company he went to work for is an ISP."

Leo: Oh, that's really bad. Oh.

Steve: "So it scares me to death what he might do with them. To me, someone like this should never be trusted/have a career in IT again. As Uncle Ben said, 'With great power comes great responsibility.' Any advice would be greatly appreciated. I know you're a busy person, and if I don't hear back it's understandable. Either way, thanks again for all your contributions to our industry." It is a conundrum.

Leo: Wow, yeah. How do you let everybody know?

Steve: Cory knows who this is. The company where he is doesn't want to press charges. First of all, I don't know how you secure his new employment, that is, his new employer. I mean, as we know, Leo, when you mistakenly install something, you really can never be sure you got it out, and there aren't rootkits and things still hiding. Imagine an admin who had full access to the infrastructure and clearly, I mean, that's just unconscionable what he did. So first of all, he should, I mean, charges should be pressed. But I also agree that Cory probably has an ethical responsibility to inform this person's new employer of what he found when he stepped in in this guy's wake.

So, yikes. I mean, I'm not an attorney. I don't have any advice. But I'm glad he's giving it some hard thought. So, wow. And, I mean, there's no way that the ISP who has just hired this clown would not desperately want to know what this person did at his previous network of employment. I mean, there's just no way. So Cory, consider that. There is no way his new employer should not know what you found that he had done, if you're really

sure that he did this maliciously and deliberately. There's just no way. And, I mean, okay. Even if it wasn't malicious and deliberate, he's a horrible IT admin.

Leo: If nothing else.

Steve: If this was all done by mistake, then he still shouldn't be employed as an IT admin anywhere. So anyway, thank you for sharing that with us. And I'm glad to reshare it with our audience.

Okay. So what happens when you are a capable white hat, well, maybe gray hat, no, okay, light gray hat hacker who designs the industry state-of-the-art high-speed hashing system known as Hashcat - which is able to, I mean, essentially it's like leveraging crypto mining hardware for the purpose of cracking passwords, that is, brute forcing the hash used to protect passwords - when this person decides he wants to look at WPA3, and in the process discovers that the protocol everybody is using today, WPA2, can be much more easily attacked than has been believed?

Okay. And the only thing this needs is a good name, and it would be all over the place. The one thing he didn't do was give it a good name. So if you Google PMKID, that will show you that this attack is blowing up on the Internet, because the PMKID is the name given to the part of the protocol of WPA2 that's attacked. So, okay. So what this is is a new way of obtaining a small bit of information from the access point, an access point, which then allows you to attack its secret.

The traditional attack, the attack everybody knew about before last week, required an attacker to - and we've talked about this in the past - an attacker to monitor all of the traffic on an access point they wish to attack and capture a successful negotiation with a client to the access point. So there is normally a four-packet handshake where somebody who knows the password is authenticating themselves to the access point. So that's not a super high bar, but it requires something of the attacker. They have to be there. They have to be monitoring the traffic. And while monitoring, somebody has to successfully authenticate a client to the access point using the secret, the password for the access point.

Thanks to this update, it is now none of that is necessary. So what has happened is that's all gone. It is now possible and now widely known that it is possible to attack the WPA2 protocol on any access point which supports a widely available technology for roaming. What happens is, as part of the - there's something known as RSN IE, Robust Security Network Information Element, in the EAPoL frame. That's a frame which is sent from the access point when it wants to explore reassociating.

Okay. So the idea is that state-of-the-art access points, that is, 802.11i/p/q/r support roaming, the idea being that you're able to roam in and out of the influence of the access point. When you come into that network, there is a silent negotiation between your client and the access point to see whether you have previously authenticated. And, if so, your agreed-upon secret is cached at both ends. So you are able to essentially silently reassociate yourself with the access point. And we are all familiar with that happening all the time.

Well, it turns out the way that's done is by the access point sending a secret, which this guy who figured out how that could be abused could take advantage of it. So anyway, what he wrote was: "This attack was discovered accidentally while looking for new ways to attack the new WPA3 security standard. WPA3," he writes, "will be much harder to attack because of its modern key establishment protocol called 'Simultaneous

Authentication of Equals' (SAE)." And that's what we talked about on this podcast a couple weeks ago.

He says: "The main difference from existing attacks" - that is, compared to what he is describing - "is that in this attack capture of a full four-way handshake is not required. The new attack is performed on the" - and here he describes the RSN IE - "the Robust Security Network Information Element of a single frame. At this time, we do not know which vendors or for how many routers this technique will work. But we think it will work against all 802.11i/p/q/r networks with roaming functions enabled," which he says is most modern routers. And in the research I've done independently, they're all over the place.

There's a neat link here in the show notes, Leo, to that Medium.com. Adam Toscher has a write-up where he's using readily off-the-shelf equipment and has a complete step-by-step how-to on implementing this attack. And then the Hashcat guy goes on to explain in detail all the advantages of this. It reduces to the following: There is this PMKID is what is contained in this EAPoL packet. It consists of an HMAC-SHA1 128. So just an HMAC, as we know, uses two hashes. But it's an SHA1, which has been highly accelerated by our current GPUs.

So the pre-master key, the PMK - in podcasts past we've described all of this in great detail. The PMK, the pre-master key, is the secret which both ends of the connection have. That is used to key the HMAC. And then what is hashed by the HMAC is the access point's name, concatenated with the MAC address of the access point, concatenated with the MAC address of the station, that is, the client. So of course we know the name of the access point. We know the MAC address of the access point because that's in the Ethernet frame that it sends us. And we know our own station MAC address.

So the point is that those three pieces of information are concatenated and hashed through this SHA1 HMAC with the pre-master key. So the attack is time-consuming, but trivial, and essentially allows the reversal of or the determination of the pre-master key that the access point has cached and is exploring the use of with a client. So it still means that we need to have - essentially you still need brute-force decrypting. But I've seen quotes from a few hours to a day or so in order to perpetrate this crack.

So unfortunately there are access points that do a poor job of rotating the key that they're offered. There are some that do a good job; many that do a bad job. And there are attacker tools available that know which are which. So you really, really, really need to use a secure password for your router. This guy in some of his - I don't remember whether it was tweets or in his blog post. He mentioned that what he uses to secure his own access points that he's in charge of is a pseudorandom string containing, I mean, like a high-quality password generated by a password generator, nothing memorable or in any way short-circuitable by brute-force cracking, something long and super high entropy so that it will minimize the chance of brute forcing. But essentially this means that it is possible, for example, to do a drive-by attack where you capture any of the traffic from the access point as you're passing by and then can perform a relatively high-speed offline attack.

Again, the only protection here is a really high-entropy password for the access point. And I would argue that, given the amount of attention this is getting, the fact that we are a long way away from WPA3, the fact that we've got password managers, all of the systems we use remember the passwords for our access points, there just is no excuse now not to use a password for a WiFi system that it is not possible to remember. Generate it with a password manager. Get it to your various devices once, tell them to save it, and then you never have to worry about it again. It's increasingly important. Well, and maybe not for random end users off in the woods somewhere, but especially for corporations, anyone where you have high-value access. Because imagine somebody

gets in, then they go and encrypt all of the files and servers and everything that your entire corporation has, and then ransom you for $65,000. Not good.

Leo: No. No. Not at all.

Steve: So WiFi cracking just got easier. Someone determined could have still done this by camping out and watching authentication happen. But given the amount of attention this is getting, I think we will see shortly some turnkey tools that make this, you know, someone's going to automate this and then turn lots of people loose to be just - and it's an offline attack. So after you capture the traffic, you go home and turn off your bitcoin miner and have it do hash cracking instead.

Leo: Wow. So you do need to have physical access for a little while.

Steve: Well, you need to be in presence of - you have to have radio access.

Leo: Right.

Steve: So, yeah.

Leo: Sitting in a coffee shop, for instance. How long does it typically take to crack the hash, though?

Steve: It is a function of the password. So if the password is poor...

Leo: Right. Good password's a good idea.

Steve: Yes. That's the only way to protect yourself is just use a crazy 64-character garbage, like something from GRC.com/passwords. Take that, dump it in. It's a pain to get everything converted once. But once you do, I mean, how often are you really having to enter your WiFi password? Shouldn't be that often. Normally devices are able just to remember it.

Leo: Right. All right. Guess I'm going to upgrade my passwords.

Steve: Worth doing because this is going to end up being turned into a turnkey attack, and people are just going to be doing it to play with it because they can.

Leo: Yeah. Well, thank you, Steve. We do the show every Tuesday afternoon, 1:30 Pacific, 4:30 Eastern, 20:30 UTC. Every Thursday, did I say? Every Tuesday. I hope I said Tuesday. Tuesday, 20:30 UTC. Come by and watch at TWiT.tv/live. Or you can even have a seat in the studio. Just email tickets@twit.tv. We'll put a place out for you. If you watch live, or you're even in the studio live, you should join us in the chatroom, irc.twit.tv.

You need a download? Well, we've got those, too. If you cannot join us live, a download will be provided at GRC.com. That's Steve's site. While you're there, pick up his bread and butter, SpinRite, the world's best hard drive recovery and maintenance utility. You can get a copy at GRC.com. And then there's a lot of free stuff there, too, you've got to peruse. It's one of those places you're just going to sink into it. And three hours you'll come up for air and go, "What time is it?" because there's so much good stuff there, including the audio versions of this show and transcripts, too, by Elaine Farris, so you can read along as you listen.

We have audio and video at our website, TWiT.tv/sn. And of course we've been around, as Steve says. We're on our 14th year now. That's really kind of amazing. Wow. Because of that, though, we're in every podcast application. Just search for Security Now!, you'll find it, or search for TWiT. Thank you, Steve. Have a great week. See you next week.

Steve: My friend, talk to you next week. Bye.

Copyright (c) 2014 by Steve Gibson and Leo Laporte. SOME RIGHTS RESERVED

This work is licensed for the good of the Internet Community under the Creative Commons License v2.5. See the following Web page for details: http://creativecommons.org/licenses/by-nc-sa/2.5/