Things Are Getting Worse

Description: This week we discuss clever malware hiding its social media communications. The NSA documents the Russian election hacking two-factor authentication bypass; meanwhile, other Russian attackers leverage Google's own infrastructure to hide their spoofing. Tavis finds more problems in Microsoft's anti-malware protection; a cryptocurrency stealing malware; more concerns over widespread Internet-connected camera design; malware found to be exploiting Intel's AMT motherboard features; the new danger of mouse-cursor hovering; Apple's iCloud sync security claims; Azure changes their CA; a bunch of catch-up miscellany; and a bit of "closing the loop" feedback from our listeners.

High quality (64 kbps) mp3 audio file URL: http://media.GRC.com/sn/SN-616.mp3
Quarter size (16 kbps) mp3 audio file URL: http://media.GRC.com/sn/sn-616-lq.mp3

SHOW TEASE: It's time for Security Now! with Steve Gibson. Malware that uses social media as a command-and-control node. Microsoft has a new malware vulnerability in their malware vulnerability engine. The fastest way to lose more than $30,000 in bitcoins. And you know how you tell your friends and family just not to click on anything suspicious? Yeah, that's not good enough anymore. Security Now! is next.

FATHER ROBERT BALLECER: This is Security Now! with Steve Gibson, Episode 616, recorded June 13th, 2017: Things Are Getting Worse.

This is Security Now!, where the only thing to fear is fear itself, and of course everything else that could potentially destroy your digital life. When the dark forces of insecurity gather up their exploits and breach toolkits, that's when we bring on this man. That's right, Steve Gibson. He is the big brain behind Gibson Research, ShieldsUP!, SpinRite, and SQRL, the man who can relieve you of your fear with a healthy dose of warranted paranoia. I'm Father Robert Ballecer, the Digital Jesuit, in for Leo Laporte. Steve, it's so good to see you, my friend.

Steve Gibson: Well, likewise, for our second of our two weeks together while Leo's off gallivanting around the Galapagos, I guess. So this was sort of a different week. You know that last week's podcast we titled, I think it was "Legacy's Long Tail." And looking over what we had to talk about this week, I just - I sort of had, metaphorically, my face in my hands and just decided, okay, we just have to call this one "Things Are Getting Worse." Because I think the forces of security are losing this battle.

We're going to discuss a clever malware which is, I mean, and we're going to look at the

technology of this because it's insidious. It's hiding its intercommunications within social media posting comments. We've got a leaked NSA document about the Russian election hacking and how they "bypassed" two-factor authentication where that was in place. So, whoops, not quite providing the security that we were hoping. Meanwhile, other Russian hackers are leveraging Google's own infrastructure to hide their spoofed websites.

Tavis, our friend Tavis Ormandy of course with Google's Project Zero, has found additional problems with Microsoft's antimalware protection engine. And we'll remember that in May there was an emergency out-of-cycle update to fix something that he had found because the last thing you want is the antimalware filter, which is ubiquitous across Microsoft's platforms, to have a remotely exploitable vulnerability, which is what he found and Microsoft immediately fixed. Well, there's another one.

We also have cryptocurrency-stealing malware, more concerns over widespread Internet-connected camera design, malware found to be exploiting Intel's AMT motherboard features, which is a concern we've been talking about on this podcast now for a couple years. There's this off-the-map processor which is in the chipset, but which is undocumented, or very, very underdocumented, and stays on even when the system is off. And I know you with your other podcast focusing on enterprise computing have looked at this a lot, Padre.

PADRE: Oh, yeah.

Steve: Then we also have a new danger of mouse cursor hovering. No longer is it enough not to click on the link. Turns out you can't even hover over a link. Then there's Apple's somewhat questionable iCloud sync security claims. The news of Azure changing their certificate authority to one that may not surprise our listeners of the podcast. And we have a bunch of catch-up miscellany and some "closing the loop" feedback with our listeners. So I think another jam-packed couple hours.

PADRE: You know, Steve, one of these days I'm going to sub in for Leo, and you're just going to say, hey, you know what, we've solved everything. There's no security. Let's just all go home.

Steve: Goodnight.

PADRE: Yeah, that will not be this week.

Steve: Actually, here we are at Episode 616, in our 12th year, And Leo and I, you know, when he initially proposed this, I had flown up to Toronto to do a day's worth of recordings for Call For Help. And during the pause between shows, because we did like four in one day, we were just sort of kicking back while they rewound their VCRs. I mean, things were going to tape back then. And he said, "What would you think about doing a weekly podcast, maybe 15 minutes, on security?" And I said, "A what cast?" Because I had never heard the term before. And I remember thinking, oh, please don't bring that up again, because it just sounded like more than I needed.

Well, needless to say, it's been a significant win for everybody. Our listeners appreciate pulling the week's events together, finding the top ones. And I need to remind everyone that my Twitter followers, and those who even don't follow, but who send me tweets of things that they see occurring that they would like this podcast to cover, that I go through the entire prior week pulling topics together and filtering through what I think are the most significant events. So, I mean, it really is an interactive driven podcast, for which I am very grateful.

PADRE: You know what's very kind of sad, Steve, is the fact that, as we move forward, we see new security vulnerabilities, new exploits, but the old ones never go away. They're always sticking around in some form or another. In fact, I just got rid of a Conficker infestation at the school that I live at. And I'm thinking Conficker? What systems do we even have that are still vulnerable to that? And sure enough, there were some old student computers that were, and they were hammering the network. And I'm thinking this is pretty much what happens in security, where all we do is add new ones. We never get rid of the old ones.

Steve: Well, and back in the early days, when it was sort of quaint to capture packets that were like out on the public Internet, I coined the term "IBR," Internet Background Radiation, because I realized that this stuff was never going to go away. There's all this good traffic, point to point, intentional, intended, deliberate traffic. But there's also sort of this underlying hiss of background radiation which are packets coming from people's closets, increasingly now unfortunately IP-connected cameras; but VCRs, routers have been compromised. So now we have all this IoT problem and growth from that.

But back in the day, as they say, it used to be servers that had long been forgotten that would get compromised and then would never get rebooted. And so they'd get MSBlast-infected, and then that infestation or that infection would then start in turn scanning the Internet and so thus creating a background of radiation such that, if any vulnerable machine happened to pop up on the public IPv4 space, before long, if it had any vulnerabilities that hadn't been fixed, it would get commandeered. And that's the world we live in today.

PADRE: Five years ago, when I was still working with Interop, we owned a Class A. It was the 45-dot Class A in IPv4 space.

Steve: Nice, nice.

PADRE: And anytime we lit it up, at any show, almost immediately at least a gigabit of noise. Remember, this is five years ago. So that was a whole heck of a lot. We had to block it upstream. And that was just all the automated scanning because our range only came up during the shows. But it was one of those things where you're like, wow, there's no humans involved here. These are just machines that just keep scanning the same address ranges over and over again.

Steve: Well, and there was that CAIDA group, C-A-I-D-A. They also maintained large blocks of space that had never had any IPs assigned. And they [audio dropout] DDoS attacks by picking up the reflected packets because attackers were spoofing their source SYN packets. They would be bouncing off of machines under attack. And then those that happened, just because the source IPs were randomly generated, all of the ones that happened to fall within their unallocated block of IPs would come in with the IP of the remote target attack that was doing everything it could to respond to the inbound spoofed SYNs with SYN ACKs. So they were able to give us our first look at the early days of DoS and DDoS attacks of given sites by looking at the traffic reflected from them into blocks of IP space, thus yet another form of Internet Background Radiation.

PADRE: I love this. Unfortunately, we could go forever. We could just to the whole Internet networking nostalgia. We don't want to do that. We want to push forward. Okay, Steve. So what is this about malware on social media sites? It sounds like it's using some sort of parsing to be able to take little bits and pieces of everything and assemble it into something bad?

Steve: It's actually very well done. I did want to take a moment to mention our Picture of the Week.

PADRE: Oh, yes, yes, please.

Steve: Somebody sent this, and I just got a kick out of it. I don't know if they put this picture together, if they found it somewhere. Anyway, but of course it follows from our discussion last week of Windows for Warships, which is unfortunately installed on the U.K.'s Trident class nuclear-armed submarines. Anyway, this is a synthesized picture of the Windows XP famous rolling green hills with the fluffy...

PADRE: Is this called - I think this is called Bliss. Is this Bliss or Azure?

Steve: I think that's the Bliss theme. I think you're right because it's one I'm familiar with. Anyway, somebody, as they say, photoshopped very nicely a submarine into the rolling green hills. Anyway, just did a beautiful job. So I don't know if the person who sent it to me did this, or if they found it somewhere. But it's just a perfect picture for our Picture of the Week.

PADRE: XP for Nukes? What could possibly go wrong with that?

Steve: Oh, lord. I did see, I was just wondering whether it was official wallpaper or what. And I did run across a follow-up article from two years ago, carried by theregister.co.uk, saying "Windows for Warships? Not on our new aircraft carrier."

PADRE: Oh, no.

Steve: Yes, apparently saner heads prevailed. Okay. So to your question about social media, the technology in this is one of the things that caused me to just say, okay, we're losing this battle. Malware was found which was scanning - shoot, I can't remember the site that it was scanning - Snapchat and was looking at comments to a Britney Spears Snapchat posting. The malware would scan through the comments, running a custom hash over each comment. If the hash value matched 183 - so this is not a cryptographic hash because, as we know, a cryptographic hash would make it computationally infeasible to deliberately synthesize a comment that had a known hash output. So this was a non-cryptographically strong hash.

But the point was it was good enough that the bad guys could make a given comment hash to a value that they wanted. So it was successively stepped through these comments. If the hash matched 183, it would then run a regular expression, a regex, on the comment to obtain the path of a bit.ly URL. And the regex would essentially look for a number of short strings or characters and then extract the character following the one that matched the regex, and it would do it repeatedly through that string. So the idea was the comment would contain a series of characters embedded which an unseen pattern match would cause to be successively extracted to create a bit.ly link.

So, for example, in this example, and then this actually occurred, someone named asmith2155 posted a comment, "#2hot make loved to her, uupss #Hot #X." Well, it turns out that, unseen - because the \u200d is a unicode character which is called the "zero width joiner," normally used to separate emojis. So it's a nonprinting, non-visible joiner which does pass through any filtering because you want to allow the emoji joiner to allow emojis to get posted. So the bad guys know all of this. So they put this \u200d successively in front of the printable ASCII that they want this regex to select out of the comment in order to synthesize a bit.ly link which then the malware fetches, which

redirects it to the command-and-control server, which in this case resolved to static.travelclothes.org and then a slash and a URL to a PHP page, which was also used in the past as a watering hole command-and-control server by the group that was known to have produced this malware.

So standing back a little bit from this, the moral of the story is, as with other forms of steganography, which I would argue this is a form of, that is, as we know, steganography is the practice of putting something in public view which is obscured in some way that you can look at it and not see what's there. So I would argue this is a form of that. So as with other forms, it is not feasible to attempt to examine public media for its hidden meaning. The only way to find this is to discover the endpoints that access that hidden content and then look at where they are looking in order to find this. So this is just diabolical. And essentially...

PADRE: It's ingenious. It's a little bit ingenious.

Steve: Oh, my goodness. I mean...

PADRE: I've got to have respect for that.

Steve: Yes. And that's the point. It demonstrates that, you know, I don't want to say we should give up. But it's like, okay, we're not going to win this. I mean, this is clever leveraging of very powerful technology which was put in place for a good purpose but is being abused. And there's more of it being produced every day than the good guys, the white hat security researchers, are able to keep up with.

PADRE: Now, where would I use this? Would this mostly be sort of a command-and-control node? Would this be a way for me to tell the malware what I want it to do?

Steve: Right, right. So the idea would be you have malware deployed all over the Internet, and you want to be able to securely and anonymously send it instructions in a way that - but you don't want to embed the command-and-control in the malware itself because then...

PADRE: Because that's how they've taken us down in the past. If you've got a malware system, and you take down the C&C node, then it all ceases. But in this particular case the command-and-control node is the Internet.

Steve: Correct. Exactly.

PADRE: You [crosstalk] take everything down.

Steve: Exactly.

PADRE: Wow.

Steve: And so the malware is looking, is like out there reading social media, just like users do; but it's doing it with foreknowledge of the way that postings will be laced into the social media to communicate to it. And so it's running a hash function over the postings, looking for those postings that hash to 183. And if found, it then runs a regex over it to extract, I mean, very much the same way the spy novels have, you know, go to page 13 and go down five lines and over three words and take the first character of that word. Then go here, here, here. Anyway, and you assemble a message. This is doing that. And there's just no way to catch it except to, I mean, it is steganography in

practice, no longer just in theory. And sure, certainly it could be that photos could be posted where the least significant bits of one color or the intensity vector in the photo contain the information. But this was actually found in the wild. It is being done. So not just [crosstalk].

PADRE: The diabolical part of that is, I mean, look at the ways that we would use to break up a botnet. Command and control, okay, so that won't work. Also you could look for trigger phrases, and if you find any of those trigger phrases you can filter out the trigger phrases. That won't work because you can make multiple phrases that will hash out to 183. So, yeah. This, again, I understand it's being used for nefarious things, but this is kind of brilliant.

Steve: And we can't take Instagram down. I mean, you know...

PADRE: We could, but we don't want to.

Steve: And that's the other thing that's diabolical is that, by putting this in a super popular public stream, we lose the ability to take the malware offline by preventing all instances from going to a single point. All [audio dropout] are now in directing [audio dropout] through something that cannot be taken down.

PADRE: Well, I think what we've learned here is that, if you want to be secure, it's actually quite simple. All you need to do is don't look for anything concerning Britney Spears.

Steve: That's right.

PADRE: That's it. That fixes it. Okay.

Steve: So there was an NSA report that got leaked which detailed the way the Russian attackers had attempted to attack the 2016 U.S. elections. And the most interesting takeaway for me, I mean, first of all it was basically website spoofing. But the question was what to do in the event of two-factor authentication because this is what two-factor authentication is supposed to prevent. Well, it turns out, and unfortunately in the slide that I've got in the show notes - I did capture it, but in order to fit it onto the page, the original image that I had was already blurry, and I had to squeeze it down to 700 pixels to get it on. But down in the lower right, under point 7, it says, "If 2FA enabled, also enter," and it says "phone number" and "legitimate verification code."

And then in the box to the right of that, in fact I did pull what it says. So the slide says: "If the victim had previously enabled two-factor authentication (2FA), the actor-controlled website would further prompt the victim to provide their phone number and their legitimate Google verification code that was just sent to their phone." So, okay. So here's the problem, is that the big problem that nothing has yet solved, and I will say with a single important caveat, a single exception, is man-in-the-middle spoofing. What the Russian attackers had done was to create a fake Google site. And so when users went to their site believing they were at Google, they were asked to log in.

So the user would enter their username and password, as many of us have at legitimate Google properties. And then, but, okay, because we're at the attacker site, the attacker gets those and has automation that immediately goes to the real Google site and is prompted for the username and password. So they forward, the attackers' automation forwards the username and password to Google and submits them. Google immediately challenges the attacker automation, saying, oh, please enter the six-digit PIN we just

sent you. Of course, that PIN goes to the actual user.

So the attacker sends back to the user, who thinks they're at Google, but they're actually at Snoogle or who knows what else, sends them back exactly that, saying, ah, please enter the PIN that we just sent you. So the user checks their phone, and it's like, oh, yup, I got the PIN, just as they would expect. So they enter that six-digit code into the fraudulent site that in turn gives it to Google. Now the attacker is logged in as the user, just having freshly authenticated themselves in the presence of two-factor authentication.

PADRE: Wow. I mean, okay. I know that this can be done. I understand the conceptual steps. But again, that's smart because it does take a bit of timing. You have to be able to know how to do this properly in order to make this work. And I'm assuming they're only doing this for particularly targeted accounts. They want to take over certain accounts because those accounts will give them access to other accounts; right? Because, I mean, this is way too much work just for every possible account on your list.

Steve: Well, actually this can all be automated.

PADRE: Oh, my goodness.

Steve: So the attacker server immediately forwards username and password. It then looks at the page it receives and then presents that to the user as the next page. So the point is it is always possible to insert a man in the middle and spoof the user who's not paying attention.

PADRE: Right. But, I mean, man-in-the-middle attacks were supposed to stop, I mean, two-factor authentication was supposed to stop man-in-the-middle attacks because you use a separate route.

Steve: Correct. And doesn't because the point is you are - and notice that even the time-based token would not be a solution because in that case Google would say, you know, please use your Google Auth or Authy or whatever you're using in order to give us the - it's only going to be current for 30 seconds, as we know, but that's enough. You enter it in. They immediately receive it and forward it to Google, but before that six-digit code is expired, and they're logged in. And I should say this was part, in this case, part of an elaborate spearphishing campaign. And it was targeted at employees of one voting machine software company, and then a bunch of people in the DNC. So in this case it was targeted.

But this scales with automation all the way up so that two-factor authentication unfortunately really doesn't protect us. And this is why, as I had mentioned before, almost two years ago I brought my forward progress on SQRL to a halt because I hadn't solved this problem. And it's also why - and it wasn't last week, Padre, but about a month ago I had said to Leo that I had stopped because somebody in the SQRL newsgroup where I've been doing all of this work said, you know, with the advent of OAuth, where users are used to logging in with another site's credentials, if SQRL presented a dialogue, said you are logging into Amazon when you're at EvilSite, then the idea was that we were presenting that as a caution to keep users from deliberately giving EvilSite their Amazon credential.

But in the wake of this growing popularity of OAuth, which is training people to essentially log in with another site's credentials, I realized that the prevention that we had was requiring too much from users. So what I have just emerged from since we last

did the podcast was about a month ago I said, "Oh, shoot." Not exactly that word, but you get the idea. "We have to fix this." What happened a year and a half ago was I realized that the - I stopped everything because I felt like I didn't understand the problem well enough. I didn't completely have a grasp of it. So I spent several days with an engineering pad and pencil drawing pictures, like to really understand what causes this problem, what causes the man in the middle, essentially the man-in-the-middle problem, and what can be done to solve it?

And the root of the problem is that an attacker is able to insert themselves between the authentication and the server to which they are authenticating. I mean, that's the crux of the problem. And so what SQRL incorporated as a result of that work about 18 months ago was something we called CPS, Client Provided Session. Because what normally happens is, when you authenticate with a remote site, it gives your browser a cookie which is now the token that represents you as the authenticated user so that your queries, which then bear that token, going back to the web server, keep you logged in, keep you authenticated.

So the problem with a man in the middle, if they're able to present themselves to that site, then they receive - it's their web session that gets the authentication cookie, and they are then logged in as you at that site. So I thought, okay, that's it. We have to have some way to cut that man out. And it turns out we have a solution. Until last month [audio dropout] not going to be mandatory. It was not - because some things about Windows 10 scared us off back a year and a half ago when Windows 10 was still looming. It turns out that Microsoft was unable to do something that they intended to, which was to cut off the Edge browser from servers running in the localhost, that is, in its own machine. They had said that they were going to do that. But it turns out there are too many instances where that's being done.

So this is the key is that with SQRL now, and this is now in the spec, has to be supported, so the failure of it sets off alarm bells because those bells should never go off, is that when SQRL uses your SQRL identity to authenticate with a remote server, that remote server no longer authenticates the web session because that's the danger here. Instead, the server sends back to the SQRL client an authenticated token. And so the secret is the SQRL client then gives that to the user's browser.

PADRE: Oh, okay.

Steve: In other words, it cuts out the man in the middle. And so what happens is when you are authenticating with SQRL, your browser initiates a page jump to the localhost, to the SQRL port on the localhost. And it sits there waiting to load a page. When you're finished authenticating, the SQRL client gives a redirect, a 301 found redirect - or is it a 302? I don't remember. Anyway, an HTTP redirect back to the browser. And that redirect contains the authentication token so no bad guy ever gets it. And so we have, as far as I know, the only absolutely spoof-proof authentication system that exists.

PADRE: Right, because the bad guy's counting on you, the human, doing something silly, which is putting in the token into a site that you cannot trust. If SQRL's taking that away from me, so I am no longer involved in entering that token, you can't spoof SQRL. You can spoof the human; you can't spoof SQRL.

Steve: Exactly.

PADRE: I like that. Okay, no, that absolutely makes sense. And that's, I mean, it just kind of makes me a little sad, Steve, because I thought multifactor authentication was a pretty good bet because that's something that I can actually explain to my parents and

my non-tech-savvy family. And now I have to explain, okay, yes, this is better than just having a username and password, but make sure you're actually entering in that authentication code only to an actual site, which they may or may not be able to tell.

Steve: So we could argue that multifactor authentication can improve some classes of attack. That is, if the username and password list or database gets loose, then maybe it helps you because - or if the username and password is captured, statically captured. The problem is that we are still victims of dynamic attack, where on the fly that second-factor is provided. And if it's captured by a man in the middle, it's not protection. And the other problem is that, if the username and password database includes the secret key for the one-time password, then you're still in trouble because they're able to, if they have that secret key, they also know what the proper six-digit code is.

PADRE: Right. Now, it's obvious the way that SQRL can solve this. I mean, this is what's baked in. Could you fix the current two-factor/multifactor authentication system? Let's use Google's two-factor. And you're using Google to authenticate other services. If somehow the authenticator sent both the service in the cloud and you a code, so in that sense, even if you put the code in and it's intercepted by a third party, the one that's sent directly to the authorized service would not get to them, or they wouldn't be able to access, I mean, is that kind of close to what SQRL could do? Or there's just no way to fix the current system?

Steve: So the secret here is that SQRL's authentication is driven by the domain name. And that means it cannot be spoofed. I mean, if an attacker uses a spoofed domain name for SQRL, it just generates a nonsense identity because, unless the domain name is authentic, it won't go to the right place. And so an attacker has to give SQRL, even if they create a fake page, the SQRL code has to be to the real website because that's the URL that SQRL uses. So SQRL has the actual web server's URL, gets the authentication token from it, and gives it to the user's browser, thus cutting any man in the middle out.

So the only way I could see another system could work is if you had something like this, if you had an unspoofable authentication. The problem is, that means the authentication has to be tied to the actual domain name. And SQRL's the only system that does that.

PADRE: Right, right. And unfortunately, since the current authentication schemes typically rely on certificates, not domain names, and we know that certificates can be spoofed and/or just handed out incorrectly, yeah, you can't really fix that.

Steve: Yeah.

PADRE: Well, good, thank you. I mean, that gives me my optimism for the week, Steve. I really appreciate that.

Steve: So we have a little bit, yes. Unfortunately, Tavis Ormandy has been busy, as usual. As we mentioned last week, we suffered his showers while he was thinking about LastPass. The good news is he's moved on. He's done a really interesting project which he posted on GitHub. I think he only has, like, six projects. And I looked at them, and this is arguably the most interesting, I think, of those. He likes Linux. And he has mature and fast and powerful fuzzing tools on Linux. But he wanted to pound on, that is, to fuzz, and as we know, "fuzzing" is the process of giving software unexpected input, looking for crash events. And if one occurs, then to look at exactly what the execution path was as the software handled what malformed thing you gave it in order to see if you could leverage that into an exploit.

So, I mean, it's a powerful technique. The problem is that Windows, I guess, either Windows doesn't have very mature tools, or Tavis just doesn't like them, or he likes Linux more. So what he did was he said, okay. I need a way of testing Windows DLLs under Linux. So he wrote, and it's available publicly on GitHub, something he calls "LoadLibrary," which is actually an API in Windows that I use often. It is a tool, a Linux tool that allows Windows DLLs to be loaded into Linux for examination and use. He says it's not a WINE replacement, WINE of course being the entire and amazing, frankly, because SQRL runs under WINE and allows it to run for Linux and Mac, for example, an amazing Windows replacement. It's not that. It just allows a DLL to be loaded. On the other hand, there are many useful Windows DLLs that have not been ported to Linux, and this allows you to use them, depending upon what dependencies they may have because you need to provide all of the other DLLs that that DLL may load.

Anyway, so he created LoadLibrary, and then he began poking at the mpengine.dll, which is the malware protection engine, which is the core of all of Microsoft's antimalware code, like Windows Defender and all of them, the malware protection engine throughout their system, their services. And throughout May he was finding problems. He found the MsMpEng Type Confusion. That was the one that forced Microsoft to issue the emergency out-of-cycle patch because it was a remotely exploitable, wormable exploit. He then found a UIF Decoder DoS problem and a Privilege Escalation throughout May.

Now he has found a very, in fact, he tweeted on June 7, he said: "Sigh, more critical remote mpengine vulns. Found on Linux then reproduced on Windows. Full report on the way. This needs to be sandboxed." So I'm sure he's communicated with Microsoft. I took a little screenshot of the top of the terminal window that he posted along with his tweet, showing that he had used LoadLibrary to load a testcase.exe, which it was then scanning. And he hit an invalid pointer in the call to the memory allocation free function, which then I'm sure he pursued, and he figured out that, yes, he was able to deliberately cause an invalid pointer. And, as we know, that's often the start of being able to leverage that into, apparently, a remote and networkable exploit.

So anyway, a number of things. That means I don't know - today's Patch Tuesday, by the way. And Microsoft did do their standard monthly rollup. There were security updates to the kernel, to Microsoft Windows PDF, to the kernel mode drivers, to Uniscribe, Device Guard, IE, and also Edge and the Windows shell. So a bunch of things. I don't know, I just didn't have a chance because this just happened, whether they also fixed this. Let's hope they did. And I should also mention that Adobe, for those of us still using Flash and other, you know, Shockwave and so forth, lord help us...

PADRE: I have Flash in a virtual machine because I will not run it on my computer anymore.

Steve: Good. Twenty-one critical vulnerabilities.

PADRE: Oh, good. Is that all?

Steve: A little more than double what they did last month. So a ton of problems fixed in the June Adobe patch, yeah. The good news is Flash is, you know, it's legacy, but quickly going away because HTML5 now is offering so many features that people used to need to use Flash for. And I keep, you know, from time to time I'll go over to TWiT.tv, and I'm unable to look at any of the videos there because they still use Flash.

PADRE: Well, you can do Twitch. If you go live.twit.tv and go to Twitch, that will work. YouTube will work.

Steve: Ah, good.

PADRE: But, yeah, I'm the same way. When I'm crawling the web, anytime I find a site where a critical function will not work because I have Flash, not just disabled, but removed from my system, it always kind of makes me pause and go, okay, well, this is something that we need to work on. This could be made better. But Steve, let me ask you. What is it about the malware protection engine that seems to make it such a big target? Is it poorly written? Or is it just that it has to have hooks into everything, and that makes it a juicy target?

Steve: It's the "I" word.

PADRE: Oh.

Steve: It's an interpreter.

PADRE: Oh, that's right.

Steve: And anytime you have an interpreter as, I mean, that's one of the things we are now seeing over and over and over. This thing inherently has to examine what's coming across the network and try to find, like, understand it. If it's an image, it's got to parse the image. If it's an EXE, it's got to parse the EXE. It's got to look inside it. It has to decompress it if it's packed. I mean, it's doing a huge amount of work. And so, again I'm not laughing at Microsoft. I recognize this is a hard problem to solve. Unfortunately, it's a hard problem to solve. And they haven't, I mean, it's very difficult to solve it perfectly, yet they have to be perfect.

And so the big problem is this is, in terms of attack surface, it is on the receiving end of the network connection and has to examine everything that comes in. It has to also be inside of the TLS decryption because it needs to have the TLS tunnel removed so it's able to get at the EXE or the image or whatever it is that's inside there. So, I mean, it is a ripe attack surface, and Tavis just found another bug in the interpretation of this content.

PADRE: So what the malware protection engine needs is another protection engine that looks at anything it's parsing and interprets that to see if the first engine should be interpreting it. Is that...

Steve: Well, and Tavis's comment was this should be sandboxed, which is interesting. Probably for speed and so as not to cause a big problem, it's in the kernel, which means it's a kernel exploit ripe attack, which is doubly problematical. And you wonder how you sandbox something that needs to be part of the OS, which is essentially where this thing runs.

PADRE: You know, about 12 years ago...

Steve: I need to take a break - oh, go ahead.

PADRE: I was just about to say that I used to have a hardware engine. It was a USB key that you could plug into the side of your computer. This was more than a decade ago. This is when I was still living in San Jose. And it basically intercepted - it was a man in the middle that would intercept all network traffic. And it used its own operating system, mini operating system to scan all traffic and look for any threats. They've gone out of business, but it was an interesting product in that it offered protection that didn't actually touch your operating system.

Steve: Well, and, you know, this is the promise of all of those secure routers that we're beginning to see. I just saw that Norton Symantec is offering something, I don't remember what they called it. It's a very pretty-looking geodesic blob on their page introducing it. But I just shake my head because either it is going to be blind to all TLS traffic passing past it, or it's going to be proxying your TLS and giving all of your machines inside the LAN a certificate that they're going to be forced to trust in order for it to intercept your TLS. And you don't want that either. So, I mean, I keep seeing all these claims about, oh, a security-enhancing router. It's like, you know, unfortunately, a decade ago, yes, you could do that. Today no.

PADRE: Not so much.

Steve: You want it to be blind to the traffic because, if it's not, then it's a huge security vulnerability.

PADRE: Right, right. In fact, there was a technology that I saw about three years ago from one of the major chipset manufacturers who makes routers. And they wanted to create a product for ISPs. It needed to be tied together to an ISP because it would allow all the router endpoints to collectively look for vulnerabilities and exploits, and then everything would be blocked upstream. So it would never even make it to the router. They could just never get the ISPs to buy in. It would add something like $5 to the cost of a router. And, well, that's just too much.

Steve: Well, and for example, I fought a problem with SQRL because I cannot run a TLS server as localhost in the user's machine, much as I would like to, because it'd be nice to be able to have the browser make an HTTPS query. But there's no way to protect the private key from theft if you have a server which is accessible. The only way websites are able to offer secure TLS is that those sites are remote and protected so that no one can get the private key.

So I came up with a way of minimizing the impact of making an HTTP query from an HTTPS page, which is technically mixed content. And it's one of the things I've been doing in the last month because I decided we have to be able to allow SQRL to be completely spoof proof. That's just - it's too big a benefit to SQRL not to be able to enforce that. And now we can. But there are lots of hurdles to be overcome. There are reasons this hasn't been done before.

PADRE: So, Steve, I'm looking for, oh, I don't know, a couple of bitcoins floating around the Internet. What could I possibly do?

Steve: So, boy, this was an expensive lesson for someone.

PADRE: This is better than ordering a pizza with a thousand bitcoins.

Steve: Yeah, the report was, quote, "I copy-pasted a bitcoin address into Electrum and confirmed the bitcoin transaction. A few minutes later I checked with the recipient to verify it had appeared in his wallet." Now, we shouldn't [audio dropout] 13 bitcoins. Thirteen bitcoins is not chump change any longer. Thirteen bitcoins at this morning's value, when I looked it up and did the multiplication, is $35,555. So he sends $35,555 to somebody else and confirms that it had appeared in his wallet. It hadn't. Somehow it was sent to the wrong address. So this guy gets online, explains in some forums of knowledgeable people what happened.

He says, "I checked all browser windows, private messages, chat histories and do not

know the address that grabbed the 13 bitcoins." Well, when he shared this history of what happened with knowledgeable users on Reddit, they pointed out that the address was almost certainly changed on the fly by malware. Specifically, there exists a clipboard contents-altering malware that has been around and known for a couple of years. This malware surreptitiously continuously monitors the system clipboard for the appearance of a destination bitcoin address and, when found, immediately and silently replaces it with its own.

PADRE: Oh, that's mean.

Steve: Oh.

PADRE: That's [crosstalk]. So, wait, quick. If he had been paying attention, would he have noticed when he copied and pasted into the window for actually sending the bitcoin that the address was different? If he had actually looked?

Steve: Yes.

PADRE: I mean, that's a tough sell because it is a convoluted string. But...

Steve: None of us try to parse that.

PADRE: No, never.

Steve: It just looks like gibberish. It's like, okay, whatever. You just sort of take it as like this noise. And so, yes, you copy it, and then go somewhere else, select a field, paste it, and that paste was a different thing than what he copied. And then he sent $35,555. And who among us wouldn't do the same thing unless we were being really careful? And as we know, it's gone. That's, I mean...

PADRE: Yeah, you don't get that back. There is no way of recalling it now.

Steve: It's gone. No. It's, oh, wow.

PADRE: That is just a dagger to the heart. And here's the thing. I mean, the block chain technology itself is sound. So what they're doing is they're attacking the weakest points. And the weakest points are - it's always going to be the client computer. This, I would call this an advanced persistent threat because this is something that just sits on your computer. It does absolutely nothing because it does not want to warn you about its presence.

Steve: Does not want to come to your attention; right.

PADRE: Yeah, and it just looks at everything, everything that you're possibly putting into the copy and paste protocol. And it looks for a bitcoin address that it replaces. That's, again, kind of brilliant.

Steve: And think about a tiny bit of code. So this little bit of code could be added to any freeware that the attacker wanted. Imagine creating or modifying useful existing freeware, just to add a tiny little bit of code because, I mean, all of the hooks to monitor the global clipboard are readily available. So all it does is it just polls the clipboard, looking for the appearance of something that is a bitcoin address and replaces it with its own. I mean, it's tiny, yet somebody behind that just made themselves more than $35,000.

PADRE: That's just maddening.

Steve: It's paying off way more today than it was a few years ago when it first occurred. And the problem is the fact that it makes so much money means that there's huge financial incentive to do this more.

PADRE: Right. I actually have - I have, like, five bitcoins somewhere.

Steve: Nice.

PADRE: I probably should go find them.

Steve: Nice. Everybody who listens to this podcast knows that I have 50.

PADRE: Oh, geez.

Steve: When I first was - we did a podcast on the block chain, where I went through and explained the brilliant technology that was the base of bitcoin. And back then I set up on a machine, the one that I use for Skype, I think it was an i5 or, I mean, it was not a big, fancy - it was not a GPU miner or anything because back then bitcoins were easy. And one hash completion was 50 bitcoins.

PADRE: Whoa.

Steve: That's how long ago it was. And I woke up the next morning, and I had 50 bitcoins. It was like, oh. And I told everybody the next week, I said, hey, guess what, this little thing I ran overnight made 50 bitcoins. Now, they weren't worth much back then. They are today.

PADRE: Are now. I had a - this was before I was at TWiT. I had a system that I was reviewing for Dell, and it was this multi-Xeon monster. And so I'm like, okay, I'll try bitcoin. And so I started mining, and I had left it running for maybe six or seven hours, and it had found five bitcoins. I'm like, oh, okay. Guess it's not that hard. This was before I really understood the block chain technology. And when I tried to do it a year later, I had an even more powerful system, and it just sat there spinning, spinning, and spinning.

Steve: And in fact there was a topic of discussion in the last week about how there's some malware which is commandeering Raspberry Pis for coin mining. And I'm thinking, who cares?

PADRE: Yeah. You could get a million of those together.

Steve: A Raspberry Pi?

PADRE: It wouldn't matter. Unless you're trying to mine a brand new cryptocurrency, no. Raspberry Pi is just not going to cut it. In fact, even the dedicated ASIC machines aren't really doing it anymore.

Steve: Correct. You would have to be mining a brand new cryptocurrency and be the only person mining the brand new cryptocurrency in order for a Raspberry Pi to have a snowflake's chance of actually generating anything. It's just like, no.

PADRE: Yeah.

Steve: In fact, a friend of mine, Mark Thompson, who is a bitcoin miner, said that two of the main GPU makers - and I don't remember which two he said, but they're the big guys - are now doing a dedicated coin-mining chip. So it's not just repurposing GPUs. They're going to be doing mining silicon.

PADRE: Well, I remember when Nvidia first started offering the GPU clusters. So just like you could rent AWS, you could rent a cluster for a certain amount of time. And it was two weeks after they made the announcement that the amount of processing power you needed to get a bitcoin became more expensive than the cost of the bitcoin. And it just - it's phenomenal how fast that will rise.

Steve: And at some point you have to wonder why anybody would sell you GPUs to mine bitcoins when they could just plug them in themselves and mine them. Actually, it turns out that in Arizona, where Mark is, it's only feasible to mine there because the power is so inexpensive compared to, for example, where I am in California. You can't make money in California mining bitcoins because the energy requirement to run the GPU is higher than, you know, at the current difficulty of getting a coin you end up losing money no matter how hard you mine, no matter how fast and seriously you mine. So you've got to do it...

PADRE: You've got to remember Butterfly Labs. So Butterfly Labs was the company that was making the custom ASIC machines. And they did exactly what you said. They would make the machines for the customers after taking their money and then hold onto them for a few weeks or months, mine with them, and then ship them out. So they probably have a bunch of bitcoin, too.

Steve: So F-Secure is a great security firm that we talk about often. The bad news is they took a careful look at a very popular OEM, Foscam, F-O-S-C-A-M. And I got a kick out of the Foscam site. It's www.foscam.com. They are currently offering a Happy Father's Day giveaway. And I'm thinking, yeah, that's right. Give your old man something serious to worry about. Oh, my goodness. Eighteen different serious vulnerabilities in Foscam's IP Internet-connected cameras. And unfortunately it's not just the Foscam brand. Chacon, Thomson, 7links, Opticam, Netis, Turbox, Novodio, Ambientcam, Nexxt, Technaxx, Qcam, Ivue, Ebode, and Sab are all OEMs of Foscam and use the same technology. So this is widespread.

The researchers at F-Secure documented 18 vulnerabilities that the manufacturer has not fixed despite being alerted to those problems several months ago. So F-Secure responsibly disclosed their findings, told Foscam, hey, here's 18 problems, and you're not going to believe what some of them are. All of the flaws were confirmed in a camera marketed under the Opticam i5 HD brand. So there's yet another one. And then a smaller number of those 18 were also found in the Foscam C2. F-Secure's report noted that the weaknesses are likely to exist in, as I've said, many other camera models Foscam manufactures and sells under other brand names or OEMs to other people.

They wrote: "The sheer number of vulnerabilities offers an attacker multiple alternatives" - it's not just here's the one way in. It's pick your way in.

PADRE: It's a menu of exploits.

Steve: "...in compromising the device," they wrote. "Among the discovered vulnerabilities are insecure default credentials" - including a null password - "and hard-

coded credentials, both of which make it trivial for an attacker to gain unauthorized access. Other vulnerabilities allow for remote command injection by an attacker. World-writeable files and directories allow an attacker to modify the code and gain root privileges. Hidden Telnet functionality allows an attacker to use Telnet to discover additional vulnerabilities in the device and within the surrounding network. In addition, the device's 'firewall' doesn't behave as a firewall and also discloses information about the validity of the credentials."

Dan Goodin, whom we often quote as a writer for Ars Technica, said: "The flaws allow for a wide range of hacks, including using the Internet-connected cameras to participate with other infected devices in distributed denial-of-service attacks, accessing private videos, and compromising other devices connected to the same local network. The vulnerabilities are compounded by the ability to permanently replace the normal firmware controlling the camera with malicious firmware that can survive restarts without being detected." I mean, it just - it doesn't get any worse than this.

PADRE: Right. Because, I mean, essentially you're giving them the ability to rewrite the operating system with the camera to whatever they want.

Steve: Install their own, yes.

PADRE: Yay.

Steve: So the standard wisdom is you must disable if at all possible Universal Plug and Play because that allows devices inside to statically map ports from the public Internet into your internal network. [Audio dropout] must have Universal Plug and Play. And in any event, now IoT devices have to be on an isolated network segment. That is, if you want to play with this stuff, with light bulbs and cameras and baby monitors and microwaves and refrigerators and everything else, give it its own network segment and/or isolate the devices where you have important information, like your iOS and your Linux and your Windows and your Mac devices. Put them on a separate network segment where those are unable to see each other, so that your IoT devices can have a field day, but only within themselves, and not be able to reach out and get into your main PCs. It's just it's no longer optional. This is the world we live in today, unfortunately,

PADRE: You know, we created something like this for Know How just last week. I can't remember who first proposed it. We created a "three dumb router setup," and that actually would solve for this really, really well.

Steve: Yup, three dumb routers.

PADRE: Segment everything.

Steve: Yup. You really do need physical network isolation. And three dumb routers is a nice way to do it.

PADRE: And in fact Cisco two weeks ago, two or three weeks ago at their big IoT conference, they released a new IoT suite. And the core technologies have existed for a while, but they put them together. Essentially what it does, they've got a security engine that looks for devices that might be exhibiting owned behavior. So they're starting to probe the network. They're starting to exfiltrate data. And what it will do is it dynamically segments those into their own VLAN. So it completely isolates them.

Steve: Nice.

PADRE: And then it warns the administrator. And this is the sort of activity that we've been telling people to do for a while on TWiET. It's this whole idea of you've got invaders in the walls, so you need to change your security to be able to deal with exploits already in your network.

Steve: I think you raise a really good point, which is we're in this purgatory, at the moment, between our routers not yet being universally smart enough like this to protect us, and IoT devices now having gotten smart enough to hurt us. And so for while we're here, it's incumbent upon individuals who know enough to be proactive in going out of their way to enforce this kind of protection. But the point you raise, I think, is a good one. And that is that one can foresee a year, two, five, or 10 from now, that will have been taken care of. Routers will have separate ports, or WiFi will exist in separate segments such that you can explicitly give some devices their own place to play, and it won't be up to users to do three dumb routers. Instead, you'll just have one smart router.

PADRE: Right. And we're getting close to that. In fact, we've got "Aspire" in the chatroom who is saying, well, unfortunately, something like a Chromecast doesn't work when you segment it. It does on mine. But mine's a bit more advanced because I've got a switch that supports dynamic VLAN. So the way that it works on my network is everything has its own sandbox. It can only see the Internet and itself. It sees no other devices unless a device of a higher authorization, like all my administrative accounts, requests Chromecast support. Like I want to send a command to the Chromecast, or I want to stream to the Chromecast. Then just for the duration of that connection it puts them in a shared VLAN so that they can share traffic. And the minute it's gone, it destroys the VLAN.

That was super enterprise-y 10 years ago. The switch I'm using is actually 11 years old. And I know that something like my Synology router actually does support VLANs and rules for VLANs. So we're not too far off. I mean, it's difficult to visualize right now, which is why I love your three dumb router setup because they're physical boxes. But we can approximate it if you've got a smart enough engine that's looking at what devices exist on the network.

Steve: Right.

PADRE: Well, okay. That's kind of hope. Right? I mean, that's a positive thing?

Steve: Oh, I think it is. I absolutely - and unfortunately it's being driven by necessity, where the necessity is we're in a very vulnerable position at this point. But I'm glad that Cisco's stepping up. And I imagine at some point before long we'll just be able to purchase those sorts of devices at retail.

PADRE: Well, it's got to become table stakes. Because unfortunately all consumer, and even SMB routers, are still operating on that flawed perimeter security model, the whole idea that we can build a wall, and you keep the bad guys on the outside and the good guys on the inside. You can't do that anymore. You just have to assume that something inside your network has been exploited. And so if your security isn't segmenting and keeping that traffic away from good traffic, then it's not a good solution.

Steve: Well, and look at the enterprise situations where the exploited entity was an employee that clicked on a link, and now unfortunately, we'll get to this shortly, doesn't even - you don't even need to click. You just need to hover, believe it or not. And because they didn't have segmentation, the executive assistant was able to compromise

the core network infrastructure and allow advanced persistent threats to exist in Sony or at RSA for a long period of time and really do damage.

PADRE: I have a friend who runs a fairly large network in Virginia. And one of the things that he implemented, and at first all the IT people hated it, was he took away the superuser account. So no one, nobody has this. And they said, well, you know, sometimes we have to jump between network segments. He said, well, you have all the authentication credentials you need. You can do that. And then as sort of a way to give back later on, he did authorize superuser account usage, but it would time out every five minutes. So you could log in as a superuser, do what you had to do.

Steve: Nice, nice.

PADRE: And if you were there longer than five minutes, it would time you out again. And if you logged in more than twice consecutively, he got a message saying someone might be abusing the superuser account.

Steve: Nice.

PADRE: And it's involved. It's complicated. It ticks people off. But it's policy. And following that policy, that's really going to help enforce security in large networks.

Steve: And, you know, that also sort of points to a broader concept, and that is monitoring. That's the other thing we see. The reason an intrusion detection system is useful is that it's not a firewall. It's, I mean, it's an adjunct to a firewall. It is monitoring usage. It is monitoring what's going on. And so that right-thinking admin said, okay, I'm going to help people do the right thing. But if somebody does this twice in a short period of time, I'm going to get a notification. That's brilliant because that means that he is watching the way the system is being used, making sure it's not being abused.

PADRE: And unfortunately he tried to do an automated system, and it kept coming back to this, like, no, you can't trust any automated system because any automated system is going to follow a set of rules, and you can game the rules. It's more difficult to game an IT administrator who actually knows what he's doing.

Steve: Right. And in fact I had to do something like that because I wrote, as we know, GRC's eCommerce system myself. And I wanted to prevent unexpected abuse. That is to say, I recognized that I could not predict all the things that someone could do. So I said, okay. I don't care what is going on. After some fixed number of queries to the eCommerce system, that person is locked out. Period. I don't, I mean, I'm not going to try to figure out what they're doing. I'm not going to say, oh, does this look good or bad? If this count hits the limit, they're banned. And that was the one bug I ever had in the system after I brought it online was that it turns out that I wasn't even discriminating between a successful purchase and if they screwed up their credit card number, or they didn't get their zip code right, or they couldn't remember what their PIN was, and they brought the count right up to one short of where they would be banned, and then they succeeded. So that they, yah hah, they managed to purchase SpinRite. Then the system wouldn't let them download it.

PADRE: Oh.

Steve: Because that final event put them into the banned category. And then when they tried to download, it said, I'm sorry, you've been blocked due to abuse of the system. Contact GRC.

PADRE: I can imagine that customer service letter was probably a little annoyed.

Steve: Well. And the moment it happened, it's like, oh, that's the one thing I should have allowed. And so of course, when they successfully purchase, I zero their counter, and then they're good to go again. So, but again, I recognized I could not predict what the nature of the abuse might be. So I wouldn't care. I would set a high limit. But if they hit it, sorry, I don't know what you're doing, but it seems a little strange. So...

PADRE: That's what we've got.

Steve: So, and I'm sure you're dealing with this over on TWiET, Padre.

PADRE: Yes, we are.

Steve: This Intel advanced management technology, the hidden processor on the motherboard and the concerns that it raises. Turns out that Microsoft, rather than just being theoretical, as it has been until now, and a concern, Microsoft found malware, advanced malware produced by a group they've named the Platinum Group. It's a state-sponsored hacking group, so it is high-end. They are abusing the Serial Over LAN, which of course the abbreviation is SOL, and that's a fitting abbreviation. That's a part of the so-called AMT, the Active Management Technology, which in turn is [audio dropout] of Intel's ME, the Management Engine, which is the independent processor embedded within the Intel support chipsets for the higher end enterprise, the vPro, and some other high-end motherboards.

PADRE: Right, the vPro series, yeah.

Steve: And so essentially what this means is that there is a, supported by the motherboard, OS agnostic, and even on when the power is off, the LAN interface is still up. And there is an option, thank goodness, the one saving grace is that Serial Over LAN is not enabled by default. So, but if it is enabled on the motherboard, there is malware using it. And essentially what happens is it is a serial emulation over LAN. So if something bad has network access to the interface on the motherboard, would be, for example, a server, then they're able to pretend to be a serial interface which has been exploited by this Platinum written malware over a TCP connection in order to access the motherboard.

So the good news is Microsoft is not saying whether they have determined that malware is able to, other malware is able to turn this on, or whether it has to be done administratively by the enterprise setting up the system in the beginning. The problem, and the fact that it is unfortunately a BIOS setting and not a physical jumper that needs to be physically changed, that's the one caveat. And so you wonder if it's not possible for something to turn on and then use the Serial Over LAN in order to abuse it.

PADRE: Precisely. And the way that we've been covering it on TWiET is that this is part of the lights-out management solution. So an administrator who might have to administer...

Steve: If you'll pardon the term.

PADRE: Yes, exactly. But, I mean, it allows them to get into machines, even if they're off. It's a separate processor that's always running. Now, if you are running a vPro processor, you do have to actually turn on AMT. It has to be provisioned for any of this to work. But as you mentioned, that is a soft setting. And since the management engine runs separately, we don't actually know its full capabilities. Intel has been very mum

about that, for good reason. But it's sort of this magical thing that we know is always running and has very deep access. It can essentially bypass the operating system and any security precautions that you've put into place in your OS. But when you get access to the Serial Over LAN, this is essentially an out-of-band management technology. It allows you to use a serial console server to be able to access devices, again, even if their primary interface is misconfigured, broken, or otherwise off.

Now, the question, and I was going over a couple of forums talking about this, there are people who are saying that, even if you don't turn on the Serial Over LAN, the SOL feature, if you have provisioned AMT, you can use AMT to turn that on. And you can do that even without turning on the computer, which is absolutely terrifying because...

Steve: I would believe it.

PADRE: Yeah, because then you can use that. You can use the Serial Over LAN interface to exfiltrate data. And because it looks so different than standard TCP/IP traffic, the typical security engines that you might have watching your network will have no idea. That's just - that's scary beyond belief. Because it means that I now have to change the way my threat engines are looking at my network.

Steve: And you know, when I looked at this last time, I remember noting that they are supporting TLS connections, too.

PADRE: Yes, yes.

Steve: That they've got certs built into the firmware, where it's protected, which allows them to have certs. So that suggests that you wouldn't be able to inspect that traffic, either. It would just look like innocuous encrypted TCP. And so something goes, oh, well, okay, fine.

PADRE: "Chumley" wants to know why we're only mentioning Intel chips. It's because this only exists in Intel's vPro chips. This is an Intel-specific issue. Now, one of the other things is Microsoft did respond. So they've updated Windows Defender. It doesn't eliminate the problem. But what it will do is it watches that traffic. And they have something that they've designed that they say can differentiate between legitimate AMT traffic and illegitimate AMT traffic. And it probably has something to do with the actual amount of traffic that you're passing because, if it's true Serial Over LAN, it should be a very minuscule amount of traffic. So the minute you start seeing it look like a Samba connection, that's probably an exfiltration.

Steve: Right.

PADRE: Okay, Steve. So let's get away from processors that run with the power of the dark lord and instead talk about this common held Internet myth, which is, well, as long as I don't click on it, I'm fine; right? I mean, that's still true; right?

Steve: This is why I'm losing hope.

PADRE: You can't lose hope. I get my hope from you.

Steve: I'll still be here, but I don't know. We have, it's been discovered in the wild, a PowerPoint-based link mouseover-based downloader, which leverages Windows PowerShell such that simply hovering over a malicious link is now sufficient to take over your computer. This newly discovered attack does not rely upon macros, JavaScript, or

VBA, Visual Basic for Applications, for its execution. When the user opens the document they are presented with text that says "Loading ... Please Wait." And that's displayed as a blue hyperlink to the user. So it's like, oh, that's interesting. Now, we're all kind of trained, even those in the know, to hover our mouse over a link to see if we get a little popup message about like what that is, or often that's the way our browsers will show us where the link is pointing to. We're able to inspect, visually inspect the URL.

Well, it turns out that PowerPoint supports a hover event on links. So when the user mouses over the text, which is, as I said, the way we typically check the destination of a hyperlink. The underlying PowerPoint documentation will execute the hover action for the link, which executes PowerShell. Now, it's the sequence, I mean, that's bad enough. But it's like, okay, well, what can happen? Well, somebody, the hacker who devised this, went through some serious work in order to knit together an exploit. But this is one of the things that just sort of makes me think, okay, all hope is lost.

So the PowerShell command that is invoked by hovering over the link connects to the domain cccn.nl, which retrieves and saves a file named c.php to the disk, the user's drive, as ii.jse in the temp folder. That file is then executed by wscript, the Windows scripting engine, and that drops a file named 168.gop. JavaScript then executes the certutil.exe with the -decode parameter, giving it the 168.gop file as the thing to decode. The result is saved in the temp folder as 484.exe.

Then 484.exe is executed, spawning mstsc.exe to allow remote desktop access to the system. That fires up the RDP protocol to the system. After that, the 484.exe was renamed and saved under AppData\Roaming\Microsoft\Internet Explorer\sectcms.exe by the mstsc.exe, where it gets re-executed from the new location. And finally, a .bat file was written to the disk and executed using cmd.exe, which changes the attributes of that sectcms.exe program to hidden, read-only, and system. It also deletes all of the intermediate files having extensions .txt, .exe, .gop, .log, and .jse from the temp folder, thus cleaning up after itself and removing the obvious tracks.

And all of that happens just from the historically safe practice of hovering over a link in a document that you have received and opened.

PADRE: Steve, does this only work on the web-aware versions of Office with PowerPoint? Because it would seem as if there needs to be some sort of handle for PowerPoint to be running in a browser in order for this to work. Or is this just default on all installations of Office ever?

Steve: I believe that this is - this runs if you receive a PowerPoint document in email. So it's the viewer, the viewer in email knows how to display a PowerPoint document. And unfortunately PowerPoint is given the power to respond to a hover action. And so that's the chink in the link, if you will, that allows then all of the rest of this cascade of actions to the unwitting user, who has been told, do not click any link. But no one's ever had hover take over their computer until now.

PADRE: Do not hover over a link. Do not move your mouse. Do not turn on the computer.

Steve: Or just give up and go home.

PADRE: Just give up.

Steve: Ouch.

PADRE: Is there a patch for this yet? Is this something that Microsoft's going to patch up? Or is this an unpatchable issue?

Steve: Not a bug, it's a feature.

PADRE: Oh.

Steve: Yeah, I mean, Microsoft is very, very reticent to remove features because they have huge numbers of enterprise users that have incorporated those features into their daily usage.

PADRE: I will say this. I am currently running Microsoft Office 2007. And my PowerPoint does not have hover management. It's not new enough to know what to do when that happens.

Steve: And I'm on 2003, naturally, because...

PADRE: Every time I do Windows Weekly, I get Paul Thurrott and Mary Jo Foley making fun of me. But I'm like, what has changed? I can still do everything I need to do. I don't need to upgrade to 365.

Steve: No, no.

PADRE: And actually, yeah, this saves me.

Steve: 2003 I acquired a license, back when the MSDN - because I'm a registered Microsoft developer, and so it used to be you had access to all those things. And so that's a static license that allows me to use it, rather than the new dynamic process, where they're counting and decrementing every use. So I've already got that. That'll be my Win7 office suite. And it works fine.

PADRE: Now, do you still have your folder? Because I got that action-packed folder that had all the software in sleeves. I used to love that thing.

Steve: Yup.

PADRE: I still have a copy of Windows XP, the Enterprise version of Windows XP. I've been thinking about installing that.

Steve: I'm still using a copy of Windows XP.

PADRE: Oh, that's right. I use it for academic pursuits, when I want to get things owned really, really quickly. But, yeah.

Steve: So I did want to mention, if we have any people who really want to dig in deeper, the link in the show notes takes what I just said - what I just gave everybody was a summary, believe it or not. But there is a blow-by-blow, process-by-process, complete breakdown of the exploit of this hover-over-the-link attack, if somebody wants more.

Motherboard has an interesting coverage of the Worldwide Developer Conference last week and of some of the content that Apple was promoting. Motherboard's title was "Apple Is Trying to Make Your iMessages Even More Private." And it seems to me that "trying" is the operative word, and maybe a bit of misdirection. During an interview that

the well-known Apple blogger John Gruber, who blogs Daring Fireball is his site - Craig Federighi? Is that how you pronounce his name? I think it is.

PADRE: Yeah, it sounds fair.

Steve: Craig Federighi said that the company has figured out a way to do cross-iDevice syncing to the cloud while still remaining unreadable to Apple. So the idea being that, if you delete something like an iMessage from one device, the new next-gen iCloud sync will delete that from all your devices, which has not been a feature that they've been able to support until now. And Apple's touting the fact that they don't have the keys for doing this.

PADRE: Right. And that's actually what we want. We don't want them to have the keys. We want us to have the keys. And the most reassuring question that you can ask of any vendor who's trying to tell you that they're protecting your privacy is what happens if I lose my keys/my authentication? And they should say you're out of luck. If they say that, then I can trust them. If they say, well, there's a recovery process, I say, well, then you do have a copy of the keys.

Steve: And in fact you heard me last week saying that that's the way SQRL was designed. And I'm sort of recognizing that SQRL may not be for everyone because it has made a different decision about key recovery than everybody else. That is, it is secure, and so there is a small responsibility that we go out of our way to manage in a useful way. But so my point is that I don't have a problem if it's not for everyone. But for those for whom it is for, it is absolute protection. And we do all kinds of things that I'll get into as soon as I have this thing published because then it will be more - I'll have more people's attention because it'll be like, wow, what is this? How does this work? But I did make that decision, that it's [audio dropout] party authenticator. There's no one to go crying to. But even so, we do give you get-out-of-jail-free cards. You just have to put it somewhere.

PADRE: Right. So let's take this and extrapolate it to the business side of Apple. This is a great technology. It's a great moment for security. But at some point someone's going to say, "I paid so much money for my Apple devices, you can't give me a way to get my encrypted iCloud messages back?"

Steve: So Craig said, Craig Federighi said, quote: "Our security and encryption team has been doing work over a number of years now to be able to synchronize information across your, what we call your circle of devices," he said, "all those devices that are associated with the common account, in a way that they each generate and share keys with each other that Apple does not have. And so, even if they store information in the cloud, it is encrypted with keys that Apple doesn't have. So users can put things in the cloud. They can pull stuff down from the cloud. So the cloud still serves as a conduit, and even ultimately kind of a backup for them, but only they can read it."

Now, Motherboard says: "It's unclear exactly how Apple is able to pull this off as there's no explanation of how this works other than from those words from Craig. The company didn't respond to a request for comment asking for clarification. It's possible that we won't know the exact technical details," Motherboard writes, "until iOS 11 officially comes out later this year.

"Meanwhile, cryptographers" - to your point, Padre - "are already scratching their heads and holding their breath. Kenn White, security and cryptography researcher, told Motherboard in an online chat: 'The $6 million question'" - maybe that's in bitcoin these days - "'is how do users recover from a forgotten iCloud password? If the answer is they

cannot, that's a major user experience tradeoff for security. If you can, maybe via email, then it's end-to-end with Apple managed or derived keys. If recovery from a forgotten iCloud password is possible without access to the keys on a device's Secure Enclave, it's not truly end to end. It's encrypted, but decryptable by parties other than the two people communicating. In that sense, it's closer to the default security model of Telegram than that of Signal.'"

And I say, as I have said to our listeners on this podcast, irrespective of that, I continue to contend that, unless the user is explicitly managing, that is to say, receiving and verifying their communicating co-parties' encryption keys, as for example is the case with Threema, Apple remains entirely free to insert an additional party line key into any or all communications. I'm not suggesting they are doing so. But they do have the capability of responding to wiretap warrants. iMessage is many-to-many messaging, not just one to one. And users have no visibility into precisely who those many co-parties are. So, for example, I have a buddy that I send messages to. Oh, look, it's magic. No one is ever looking at keys. Nobody's looking at authentication. How does that happen? Apple does that. Apple manages those keys. And we know that iMessage supports group messaging, where you could have a bunch of people. And it's all magic.

Well, everybody I'm sending to has their own key. So that means that my client must be individually encrypting those individual messages for each party in the message. Nothing prevents Apple from sticking in an extra key, an NSA key, so that now my outgoing messages are being also encrypted for a key that the NSA holds. Again, I'm not saying they're doing it. I'm just saying, this is a classic case of convenience versus security tradeoff. Apple, we know, stands for security. But the architecture and the design is not secure. They can say, oh, it's end-to-end encrypted. We can't see. No, but you could if you wanted to.

PADRE: Right. And, I mean, that's the thing. I like the fact that they seem to be security first here. But the details are far too sparse. I need to know exactly what you mean by "we don't have your keys." Because they seem to be talking around whether or not they can recover, or whether or not they might be able to assist you with getting your data back after a major disaster. That can't be ambiguous. But Steve, there's another part here. And again, this goes into your security versus convenience. This would be a great feature if I could opt into it. If you could have a user who could make an informed decision to say, if I lose my authentication, this data is gone, versus something that's just rolled out as the default in iOS 11.

Steve: Right.

PADRE: But then again, aside from you and me and a good part of the Security Now! audience, I don't see any users volunteering the ability to lose their data. To me, I mean, I have enough trouble convincing the people I live with, who are relatively intelligent, that they should be concerned about their digital privacy. If I tell someone, well, this is more protection, but if you lose your username and password, everything that you've done before is gone, I don't know a single one of them who would accept that.

Steve: Yeah.

PADRE: And unfortunately they have to accept that. If we have a low bar of security for many people, that means there's many people around us who get exploited. And when many people around us get exploited, it increases the risks that we will get exploited. That's just how that works.

Steve: Well, I can't wait until you and I have a chance to sit down, and I can go over all

of the decisions that are part of the final SQRL result, and how it meets these goals, because it really does.

PADRE: It's going to come down to you talking to your friends and saying, well, have you enabled this? Because if you haven't enabled it, I'm sorry, you can't be part of my circle because I can't trust you anymore.

Steve: So one real quick comment. I just noticed something that I just got a kick out of, and that is that Microsoft's TechNet blog - and this got by me. This was in April. They noted that the Azure TLS certificates were changing. They wrote in their blog: "We know security is a top priority for you, and so is uptime of your applications. To give you additional assurance of the authenticity of Azure services, most Azure services get their SSL/TLS certificates from a known set of intermediate certificate authorities (CAs) that Microsoft operates. Microsoft publishes details of these CAs in its Certificate Practice Statement.

"Some organizations," they write, "configure their applications with specific CAs, using a security practice" - that we've often discussed here on the podcast - "called 'certificate pinning.'" And as we know, "pinning a certificate" means that you record the actual fingerprint, the cryptographic signature of the certificate, so that it's not just - you're no longer trusting the whole chain, you are saying, this specific certificate is what we're going to trust.

They write: "Since CAs expire and get replaced, this practice requires that all applications be updated periodically to use the latest CAs. If this is not done in time, the application may get interrupted. To make this process easy for you, Microsoft publishes new CAs well in advance of using them. The current intermediate CAs used by Azure are due to expire in May 2018." So this was one year advanced notice. "Microsoft published a new set of CAs last year," they wrote, "in the July 2016 revision of their CPS (Certificate Practice Statement). Azure services will begin using these new CAs from July 27, 2017." So coming up next month. "If your organization configures your application with specific CAs, then you must ensure your applications are updated by July 27, 2017 to prevent interruption." So next month.

Anyway, the point was I got a kick out of this. They finally said: "Microsoft Azure services were previously signed by either of two intermediate certificates. They are adding four additional intermediates, and they have a new endpoint." The original CRL, the Certificate Revocation List distribution point, was public-trust.com. The new CRL distribution point, meaning the parent of their new certificates, it may not surprise our listeners to know, is Digicert.com, which is my chosen and favorite CA. So bravo to DigiCert for a nice "get," as they say, being the root signer for Microsoft's Azure services.

PADRE: And this is what it's going to take for other CAs to stop playing on the gray side. It's going to, you know, you've got the large providers - Microsoft, Amazon, Google - getting in there and saying, if you're going to do this, we just won't use you. And let's see how well your business is going to do if we no longer trust your certs.

Steve: Right.

PADRE: So, yeah, kudos to DigiCert; kudos to Microsoft. This is absolutely warranted, and it's absolutely overdue. This probably should have happened years ago.

Steve: And if you play fast and loose like Symantec was caught doing...

PADRE: Oh, gosh.

Steve: ...by allowing - basically thinking that you're printing money, and oh, look, let's give other organizations the ability for unmonitored printing in our name. And, well, we know what happens is that Symantec is in trouble because they were playing fast and loose, and now Chrome and Firefox are looking hard at minimizing their trust.

PADRE: Right. There's still a timeout; right? That time period hasn't expired yet?

Steve: Oh, yeah. They're in trouble for a few years.

PADRE: And there's a reevaluation that Chrome is going to do at some point in the next, is it nine months?

Steve: Yes.

PADRE: So essentially they're in timeout, so they're expiring all their certificates. They're in timeout right now for new certificates.

Steve: Yes. They're only allowing short new issues. Yup, they're in the doghouse. They're way in the doghouse.

PADRE: This is the security equivalent of go sit in the corner and face the wall. Wear this hat.

Steve: I did find a nice note from a SpinRite user, James Campbell, with the subject "Thank you!" And here's yet another use of SpinRite. He says: "I upgraded one of my NASes and decided to use the old one to set up a server at my church. The problem was I had 13 hard disks laying around. I knew that some of them were laying around because I outgrew their capacity, and I knew that some had failed. But alas, Post-it notes don't stick forever." So they became unlabeled drives with unknown background.

He said: "So, SpinRite to the rescue. Today, 13 disks fully tested, four with catastrophic failures that wouldn't even mount. And the remaining nine have now all been fully verified by SpinRite. The old NAS is now stuffed with four matching drives and has been installed at the church. Thanks, Steve. Jim C." And Jim, thanks for sharing your success using SpinRite.

PADRE: You know, I had something very similar to this. I had a drive corruption because of - it was a brownout. So power fail, then brownout, then power fail. They don't - NASes hate that. And so I had a Synology, one of their 12-bay or 16-bay? It was large, a large one.

Steve: Nice, nice, yeah.

PADRE: And when it went down and came back up, six of the drives were reporting to SMART, they said they just weren't going to spin-up. Pulled all the drives. Kept them labeled. Ran SpinRite on all of them. Put it back in. And when it came up, it said two failed drives. And I was able to get everything synced up before replacing the questionable drives one by one.

Steve: Nice.

PADRE: And again, that was on me because I should have - it's a remote location. I

should have had it set up so that it was notifying me when the SMART counters were getting out there. But, yeah, it works. I will say I tried doing the same thing with a ReadyNAS, which is the Netgear NAS. That did not like it. It's very, very picky. Synology, fine. ReadyNAS, not so much.

Steve: Interesting. Well, and I think that's one of the problems we're seeing, that RAID in general is trying to get smart and is beginning to have [audio dropout] with drives. For example, there are some RAIDS now that look at the timing of the drive. And if the drive seems to be taking too long, they decide, oh, there's a problem here, and go off. Whereas some drives do take some time, but then come back with an answer, and the RAID has already given up on them.

PADRE: Right.

Steve: So it's sort of, you know, there isn't a standard for this. And people are making things up as they go along and causing themselves problems.

PADRE: That's why there's only two drives, two types of drives I use in my NASes right now. One is Western Digital Red because that's got the TLER control, so you can essentially say, don't figure it out, the NAS does everything, just let the NAS do it.

Steve: Time Limited Error Recovery.

PADRE: Right. And the second one is Seagate's IronWolf series, which is often because I've got a bunch of 8TB drives. They also run cool. I think they will do TLER reconfiguration, as well. But, yeah, I've had so many drives from different manufacturers, including Seagate and Hitachi, that I put into NASes, and they just - they don't do well.

Steve: We've talked on the podcast about "Orphan Black."

PADRE: Yes.

Steve: One of Leo's favorites. Tatiana Maslany does this incredible job playing...

PADRE: So many different parts.

Steve: ...multiple characters. I just did want to note to our listeners, since we've talked about it through the years, that the fifth and final season has just started back up again. And also that [audio dropout] of ours was "Sense8," and that the second season appeared a couple weeks ago on Netflix. So if anyone had missed that occurring and liked the first season of Sense8, the second season is there.

And last week I meant to mention that the previous day, that is, Wednesday, which was Worldwide Developer Conference Day last week, the SQL to Monument Valley was released. It was a favorite of ours three years ago. And in fact it was the 2014 iPad Game of the Year. Not free. It's $5 for iOS and also an Android release coming soon, published by Ustwo Games. For what it's worth, I grabbed it. And I don't know if it was worth $5. I went through the entire thing in two sittings.

PADRE: Steve, tell me what you really think.

Steve: I mean, it's beautiful. But it's sort of more of the same. So it was like, eh, okay. I mean, it was interesting and nice. But again, I would like something that lasted a little

more than two sittings. But, you know, so there it is.

PADRE: I think Monument Valley is coming to the Nintendo Switch.

Steve: Nice. Makes sense.

PADRE: Yeah; right?

Steve: They have a much bigger staff. They used to have eight people. And what happened was, it was such a grind that they did a little bit of a release. They released something called - what was it called? It was an update to - oh, Forgotten Shores. So they did an enhancement to the original Monument Valley, the Forgotten Shores extension. But they then wanted to go off and do something else. They're now 20 people strong, and every single one of those additional 12 people came to Ustwo Games because they had fallen in love with Monument Valley and wanted to do more. So that's why they ended up saying, okay, fine, we'll do more.

PADRE: Well, that's the best way to do it, yeah.

Steve: And 30 million downloads of the first version. So, yeah, baby.

PADRE: Do the math.

Steve: I don't mean to say it's not great. But for me, it was just - it was like I think they were kind of right, after doing the first one, to go try and do something different because it was charming and new. But the sequel was like, eh, okay, more of the same.

PADRE: It's like the Matrix movies. The first one was mind-blowing and had never been done. And then they tried to put substance and philosophy behind it in movies two and three, and people just sort of said, no. No, stop. Stop it.

Steve: I have two "closing the loop" pieces from listeners. One guy, and this was important - well, to me, at least - said: "Question regarding SQRL." This is Jonathan Lloyd tweeted: "How might a user share their credentials for a single site with someone else? Is this possible?" He said: "I ask because many websites don't have a robust permission system. I still need to send or receive login info at least a few times per year. In any case, I thought I'd ask. I don't recall hearing this mentioned on SN before. Love the show. I try not to miss an episode."

That's a great question, and we've addressed it. The problem is, what if multiple people need access to a single site? The way it's done today, as we all know, is, hey, what's your password? Well, could anything be less secure than that? The point is, because the binding of identity is weak in our current Internet ecosystem, the weak identity binding allows a person to share their weak identity, which is just represented as a username and password, with someone else, who then is able to essentially impersonate them to that website.

Well, SQRL identity binding is tight. It is the reverse of weak. I mean, and in fact a SQRL identity cannot be shared without sharing all of your SQRL identity. That is, it's, I mean, it is something you cannot share. So part of the SQRL specification, it isn't a requirement, but it strongly urges what we call a many-to-one mapping, meaning that when a site brings up SQRL support, specifically because SQRL does not allow people to share identities in the same way that people can share passwords, what we're encouraging any site that supports SQRL to do is to allow multiple SQRL identities to be

associated with an online account. So that Mom and Dad, for example, can both use their individual SQRL identities to share access to their banking account. Or Mom and Dad and the kids can all share access to Legoland.com or something.

And the idea is that you would have a couple users who would be admin or privileged, and some other users who would be guest users. And so that it would be the privileged users that would be able to invite other users to join with their SQRL account. And we worked out all the protocols and all of the handshakes. So the point is, because SQRL's identity binding is so tight, I mean, it really is you, then the way sites have supported a non-many-to-one mapping is like, well, okay, just give them your username and password. Well, SQRL fights strongly against that. So instead we're going to strongly encourage SQRL-supporting sites to offer many-to-one identity mapping. And I think that's, I mean, it's not just SQRL. Sites ought to be doing this and, like, get with the plan because asking people to disclose their identities to others is fundamentally insecure and really bad security practice.

PADRE: Right. This is part of that people training. It's part of any new technology or process. People have to understand this doesn't work like it did when you could give someone your username and password, and they could get in as you.

Steve: Correct.

PADRE: This is you.

Steve: Correct.

PADRE: You cannot share this part of yourself.

Steve: Correct.

PADRE: Okay.

Steve: And lastly, Carl Green tweeted. He said: "Regarding Travel Mode for 1Password password manager," he said, "question. Can it be en/disabled" - meaning enabled or disabled - "from your iPhone? Seems to me they would ask you to do that." And the answer is, well, no. We talked about 1Password's brilliant travel mode, where you divide your secrets into vaults, and then vaults can be individually tagged as safe for travel or not. And then before embarking on travel, before you're going to go through customs or approach the mean-looking, or as I said to Leo, the officious-looking TSA agent, you're able to go to the 1Password.com site, and you log in there, and then turn on, you activate Travel Mode.

And what happens then is all of your 1Password-enabled devices everywhere have the vaults tagged as "not safe for travel" deleted. So the point is they did this right. They made the control of Travel Mode out of band, as in not something that the 1Password app itself has any awareness or knowledge of visibility of. So there is no way for anyone looking at it to know that you are in Travel Mode, or that you have enabled Travel Mode. All they see is a subset of your secrets, those that you have explicitly allowed them to see. So great question, Carl. And again, 1Password did it right.

PADRE: Steve, I might be mistaken here, but it seems that we've actually gotten to the end of your notes. I don't think we've ever done that in an episode that's been you and me. We typically stretch things out a lot.

Steve: Well, actually I scaled this podcast for the two of us, Padre.

PADRE: You don't have to pack it quite as full because, I mean, I love engaging you about these things because I...

Steve: And I'll tell you, from the feedback I get from our listeners, they really enjoy you co-hosting. So thank you.

PADRE: I take delight in this.

Steve: Thank you very much.

PADRE: Again, I love working with Leo. I will work with him as long as he will let me work with him. However, I do enjoy when he goes away on vacation because it means I get to sit down with you for a couple of hours.

Steve: And I did hear he's got six vacations planned for this year.

PADRE: Oh. I will be down with that, then.

Steve: I think we'll be seeing more of you.

PADRE: Fantastic. Folks, that does it for this episode of Security Now!. Now, don't forget that we are live here on the TWiT TV Network every Tuesday at 13:30 Pacific time. Steve will always be here to inject you with the latest serum that will inoculate you from the security threats of the day, or at least to give you that healthy paranoia that we talked about at the top of the show. Don't forget you can find all of our shows at the show page at TWiT.tv/sn, as well as on iTunes, Stitcher, YouTube, and wherever fine podcasts are found.

You can also get high-definition, high-resolution audio versions of the show at GRC.com, which is also where you will find everything about GRC, SpinRite, ShieldsUP!, and of course SQRL. Until next time, I am Father Robert Ballecer. This man right here is - I'm going to call him Dr. Steve Gibson, Professor Steve Gibson. And until next time, if you need to think about security, you need to think about Security Now!.