SECURITY NOW!

Transcript of Episode #612

Makes You WannaCry

Description: This week Steve and Leo discuss an update on the FCC's Net Neutrality comments, the discovery of an active keystroke logger on dozens of HP computer models, the continuing loss of web browser platform heterogeneity, the OSTIF's just-completed OpenVPN security and practices audit, more on the dangers of using smartphones as authentication tokens, some extremely welcome news on the Android security front, long-awaited updated password recommendations from NIST, some follow-up errata, a bit of tech humor and miscellany, closing the loop with some listener feedback, and then a look at last week's global explosion of the WannaCry worm.

High quality (64 kbps) mp3 audio file URL: http://media.GRC.com/sn/SN-612.mp3
Quarter size (16 kbps) mp3 audio file URL: http://media.GRC.com/sn/sn-612-lq.mp3

SHOW TEASE: It's time for Security Now!. Steve Gibson is here, and of course the topic of the day, the WannaCry ransomware exploit. How did it work? Why did it happen? Who is at fault? Who not to blame? It's all coming up next on Security Now!.

Leo Laporte: This is Security Now! with Steve Gibson, Episode 612, recorded May 16th, 2017: Makes You WannaCry.

It's time for Security Now!, the show where we cover your security and privacy and safety online. And never in our nation's great history has there been a greater need for a man like Steve Gibson, a man who can save us. Steve, the host of the show, welcome. It's good to see you.

Steve Gibson: Well, all of our listeners knew what today's topic would be.

Leo: Oh, yeah.

Steve: So I had some fun with the title. So this week's Security Now! #612, titled "Makes You WannaCry."

Leo: I felt so guilty because I got sick, as you know, and I missed doing the radio show. And I think of all the shows to miss, this was the worst show I could have missed all year long. On the other hand, the good news is there'll be more of the same soon.

Steve: There will. And you can sort of - more is known about it now.

Leo: Well, yeah.

Steve: So you can do a really comprehensive wrap-up on this coming weekend.

Leo: Well, I'm going to listen carefully to what you have to say.

Steve: And the other blessing is that, because there's a lot of interesting things to talk about, about WannaCry, is that it so dominated the rest of, I mean, like the entire week, that we won't have the normal 16 different topics that we also need to talk about. So we have a reasonable amount of week's news. We're going to talk about...

Leo: The hacking community was just all sitting back in awe.

Steve: Just, like, stunned.

Leo: Watching, you know.

Steve: They were all running around focused on a single thing, rather than up to their regular mischief. We have an update, sort of a follow-up on the FCC's Net Neutrality commenting system. The discovery of an active keystroke logger present on dozens of different HP computer models.

Leo: That just shocked me.

Steve: Yeah.

Leo: I'm really curious about that.

Steve: It's just sloppiness. I'm sure it wasn't malicious, but that doesn't mean it isn't a problem. We've got sort of an interesting continuing movement that I called "the loss of web browser platform heterogeneity" that I want to talk about. Our gang over at the OSTIF, who's been funding the review and auditing of open source software, who previously did VeraCrypt, just finished their audit, or funding and working with the audit of OpenVPN and found one bad problem and some other things to get cleaned up. So we'll talk about that.

More on the dangers of using smartphones as authentication tokens, which has been a recurring theme for us recently. Some very welcome news, finally, on the Android security front, coming with Android O later this summer. We'll talk about how there's going to be a deep rearchitecting that actually is underway and already present in the developer preview that should significantly improve Android's ability to stay current with patches, even for non-Pixel and Google and Nexus devices.

Oh, and some long-awaited and much overdue updated password recommendations from the NIST. And the number one on the list is just a kick because this is the advice we've been giving for so long, that ran contrary to the formal advice. Well, they finally figured that out, too. We've got a little bit of follow-up errata, a bit of tech humor, some miscellany, some closing the loop with our listeners, and then we're going to get into everything that is known about WannaCry, from how it happened, what happened, what it is, some technical details, a little bit of Microsoft CYA from Brad Smith, the president and chief legal guy there. And that'll bring us current. So I think another great podcast.

Leo: Yeah, I'd love to talk to you about Brad Smith's piece because I do agree with him about some things.

Steve: I do, too.

Leo: I'm not sure Microsoft is free from blame here.

Steve: Yes. He disavowed any responsibility.

Leo: Yeah, that I don't...

Steve: It's like, eh, well.

Leo: But at the same time there's no such thing as perfect software. And everybody...

Steve: And what does it mean, then, that they updated XP? I mean...

Leo: Isn't that interesting.

Steve: Yeah, exactly.

Leo: Yeah. I mean, they said never again, but there's so many XP machines out there. I think that was them being a good corporate citizen. But again, it's your take that matters because this is Security Now! with Steve Gibson, not some guy named Leo. All right. Back we go to Steverino.

Steve: I have to put my spinner down so I can focus.

Leo: Oh, are you a fidgeter? Now, you sent me, and I don't have it because my 14 year old stole it, that is the best spinner I've ever seen. It's called the iSpin.

Steve: The iSpin. Amazon does have them.

Leo: It's made entirely of brass. It's machined. It does have bearings, I know, because I took it apart.

Steve: Yeah, you're able to unscrew the hub and sort of see them. And he actually recommends you do that because you don't want to ever put any lubrication in it because the viscosity would slow it down. But you want to just blow it out if it collects any dust over time.

Leo: Anyway, you sent me one, and I am grateful. And one of the things that's cool, and I think I've demonstrated this already, you can remove an arm, and then it's unweighted. And it's a great way to learn physics. I told Michael, I said, "Try this and then tell me, come back, you can use it, but you have to tell me why that works."

Steve: Yup. And I was playing with it again last night. I have a desk lamp which is LED and dimmable. If you dim an LED lamp, normally what they're doing is they're reducing the duty cycle of the on time. So it creates a shorter burst, much more like a strobe. And spinning it under a dimmed LED light is really fun because it does like the stop-motion thing. And as it's slowing down, the interaction of the spin versus the flicker creates phased beating effects. So, yeah. Anyway, as I said to you before the show, I have an idea that I'll steal some time on the weekend and experiment because I think I have a fun upgrade for this.

Leo: These things have just spread like topsy through the schools, to the point where schools are actually sending people home.

Steve: Actually, they've banned them, yes.

Leo: Yeah, yeah.

Steve: So our Picture of the Week I got a kick out, and it's apropos of today's topic. It's the classic Venn diagram that we all learned and is a very useful means of showing the amount of - sort of the nature of sets. It's used in set theory for, like, you'd have two different groups that have some things in common. Well, in this Venn diagram we've got two, so-called the "disjoint set," or the "null set," where there are two separate regions with no overlap, the one on the left labeled - oh, I'm sorry, and the title is "The Flaw in the WannaCry Extortion Scheme."

And the flaw is that there is no overlap between these two sets, the first one being people who use Windows XP, and people tech-savvy enough to figure out how to pay ransom with a bitcoin. And of course you're going to have people infected who can't pay if they want to because that's like, what? What's a bitcoin? And then you've got all the bitcoin people who are like, I would never be using XP anymore. You crazy? And it is an interesting aspect of this last week in WannaCry that, as extensive and as huge a cyber event as it has been, the last number I saw was like $53,000 in equivalent bitcoin payments.

Leo: Yeah. Brian Krebs says that - he wrote a book called "Spam Nation" where he talks about this, that that's not unusual. That these guys don't really care. If they make 50 grand they're very happy, some guy in Moldova who is unemployed. And they don't really care that it's the consequence to billions of dollars' worth of consequences because they got 50 grand.

Steve: Well, and - right.

Leo: Although the latest news makes me think there's more to it. But go ahead.

Steve: Well, the backup for that idea also is that previous malware didn't make anybody any money.

Leo: Right.

Steve: Very much like the Gmail phishing scheme that we talked about a couple weeks ago, it didn't do anything bad to you. It was just sort of a proof of concept. It's like, oh, I'm just going to propagate because I can. So this would have happened, even if it didn't have the ransomware backend, just because you could. Like all of, like, MSBlast and Code Red and Nimda and all of these previous Internet worms. So the fact that it made money was just sort of a bonus. But the fact that it encrypted people's systems, I mean, like all the important files on their system and brought huge chunks of IT infrastructure to its knees made it much more devastating than if it had just been a worm propagating for the worm's sake. Anyway, we'll get to that in a second.

Leo: I guess the point is that, depending on who created it, the devastation may indeed have been the goal.

Steve: Oh, interesting.

Leo: Maybe you haven't heard the latest.

Steve: I haven't heard the latest.

Leo: I'll fill you in when we get to it.

Steve: So definitely come back when we get there. The FCC is acting a little oddly. They're reporting spamming of the FCC Net Neutrality commenting site, and they claimed DDoS attacks. And the moment I heard that, I remembered that my own experience - and I don't think I did it that night, but it was like sometime later the following day. The site is just horrifically designed. I don't know what the database backend is. But as I mentioned on the show last week when I talked about the GoFCCYourself.com, which was the referral domain that John Oliver gave us on HBO's "Last Week Tonight with John Oliver" Sunday before last, was that you couldn't get a total of the number of the comments. It was just very clunky in its design. So my suspicion was that it was just lots of people trying to submit comments, and it wasn't a classic DDoS attack.

Leo: And there was spamming happening from the anti-Net Neutrality side.

Steve: Yes, yes.

Leo: They were all duplicate comments.

Steve: Correct. So it wasn't even very well done spamming. And I did note also that their site doesn't have any bot protection on it. So it was like, okay, again, another strike against it. Who's going to do a good form submission that doesn't prevent it from being spammed?

Leo: It's a government website. What do you expect?

Steve: Well, yes, the government, exactly.

Leo: But, by the way, the same thing happened last time. The difference was that the Chairman of the FCC didn't declare that they've been DDoSed and close the whole thing down.

Steve: Right. Yeah. So a guy named John Bambenek, a threat intelligence manager at Fidelis Cybersecurity, was quoted saying: "There don't appear to be any indications of a DDoS attack in the sensors we use to monitor for such things. It appears the issue with the FCC is less of a DDoS attack as traditionally defined, and more of an issue of crowdsourcing comments generated by John Oliver and reddit," who also picked up on it. And of course we did on this podcast on the following Tuesday, which was last week.

Also Jake Williams, CEO of cybersecurity firm Rendition InfoSec, said the FCC "offered no support" to prove a DDoS had occurred, adding: "There was no observed dark web chatter about such a DDoS before or after the event, and no botnets that we're monitoring," he said, "received any commands ordering a DDoS on the FCC's site." And also, when the press asked for additional background from the FCC on the so-called DDoS attacks, no response was forthcoming. So I think they were unfortunately just probably embarrassed by the amount of both legitimate and illegitimate traffic that came to there in order to lodge their protest.

This was much tweeted from our listeners, the news that the HP Conexant audio driver was found to be logging keystrokes. As Ars put it: "HP is selling more than two dozen models of laptops and tablets that covertly monitor every keystroke a user makes. The devices then store the key presses in an unencrypted file on the hard drive." So in looking at this, several of the characteristics of this argue that it was not deliberate, that it's just highly irresponsible keystroke logging with permanent stroke-by-stroke persistent storage that has been in place for apparently maybe as much as two years, or more than two years.

The driver, which is digitally signed by Conexant, who is a legitimate audio chip maker that HP uses in many of their machines, was digitally signed on the day before Christmas, December 24th of 2015, so more than two years ago. The security firm that first spotted this, modzero, disclosed this information, noting that v1.0.0.31 was then later extended with even more problematic functions. The most recent version is 1.0.0.46, which implements the logging of all keystrokes, they wrote, into the publicly for any user readable file. It's at C:\Users\Public\MicTray.log, M-I-C-T-R-A-Y dot log. So any users of HP machines ought to, if you're curious, go look at C:\Users\Public\MicTray.log to see if your passwords are all present there.

Now, here's why, you know, several things suggest that this was not deliberate or malicious. For example, the file is not appended to after each login. It is overwritten. So it feels like some forensics and developer code that was there for development was left in. And these guys wrote, so like trying to make it seems as bad as possible because they're the security firm who found it, so they want to have found something big, they wrote: "If you regularly make incremental backups of your hard drive, whether to the cloud or to an external hard drive, a history of all keystrokes for the last few years could probably be found in your backups."

Okay, well, yes, because that's how you would extend the badness from the fact that this is overwriting that file with each login, rather than deliberately continuing to persist them over time. They wrote that the keylogger is included a device driver developed by Conexant, a manufacturer of audio chips that are included in the vulnerable HP devices. They said: "This type of debugging turns the audio driver effectively into keylogging spyware. On the basis of meta-information of the files, this keylogger," they wrote, "has already existed on HP computers since at least Christmas 2015," as I had said.

So what can we do? There is a program, this MicTray, either MicTray, M-I-C-T-R-A-Y, 32 or 64 dot exe, which exists in the Windows System32 directory. So they suggest deleting or renaming the executable so that no keystrokes are recorded anymore. You will lose some functionality. The reason an audio driver has any business with the keyboard, I should explain, is that it listens to keystrokes globally within the system in order to implement hotkeys. So this Conexant driver supports various hotkeys for, like, muting the audio or turning it up and turning it down. And so this was probably put in there for that reason, for diagnostics and debugging, but then just left on by mistake.

Now, this is one of those things which probably in itself wasn't malicious. It isn't sending them anywhere. It isn't phoning home. It isn't doing any of the things it could be doing. But it's also become much more worrisome, now that it is widely known and has been publicly disclosed, such that, if anything nasty gets into an HP machine, it would almost be malpractice on the malware's part not to go check to see if it can suck up all the keystrokes which have been logged since you logged in. So it's worth taking some action. I imagine that HP will quickly get an update from Conexant and push that out, make updates available so it's worth, you know, depending upon your profile, who you are.

What you probably could do would be to replace the log file with an empty text file, rename it, and then set it to be read-only. That's a simple prophylactic measure. If you create a file that it wants to create and make it empty and then set it to read-only, most software is not smart enough to look for that, change it to writeable, and then delete it, and then start writing to it. So my guess - and you could easily just tell, you know, you can see whether the scheme, the simple strategy works. If so, that'll keep it from logging anything because it just won't be able to until you get yourself an update that no longer tries to do that at all.

But this is an interesting opportunity, and this ties into something we'll be talking about in a few minutes, and that is to note that the Windows platform has never been secure. I mean, it predates a worry that we have today about security, or a consciousness, an awareness of security. For example, the fact that debuggers have the ability to attach themselves to any other process is both convenient for developers and horrific for security. When we talk about DLL injection, that's something that you can do. You can just inject a DLL into some other program's operating process space and cause it to run. So now you're running your code or that DLL code in the context of another process, which the process probably doesn't want to have happen. But Windows is like, oh, well, you know, think of the flexibility. It's like, uh-huh.

And, you know, this goes back to my original Windows metafile observation, where I didn't think it was malicious that Microsoft allowed a Windows metafile to contain native code, which a metafile tag allowed you to jump to. It made total sense to me that back then some developer thought, well, if we need some function that we didn't build into the metafile interpreter, let's just let it run natively. And everyone thought I was crazy for thinking that. It's like, no, you just don't understand the world in which all this happened.

So similarly, Windows apps can freely establish what are known as "global hooks" for many different things. You can hook the mouse. You can hook the keyboard. You can hook screen events. And this allows the kind of flexibility that we're used to having in Windows, where you can have macro programs that are able to record keystrokes and inject the keystrokes into applications for macro playback. Well, isn't that handy. Well, yes. And you can also get up to all kinds of mischief doing the same thing. And various of the Windows assistive features depend upon those sorts of things. And of course we know that programs can capture the screen, which you may not want them to do, if it happens to have sensitive content on it at the moment.

So Windows traditionally has been very developer-friendly and a power user's environment. And of course this is part of what Microsoft plans to be changing with Windows S, which implements among other things a highly restrictive application sandboxing so that all of these sorts of things which have historically been possible on the Windows platform, those will be lost under Windows S. And so there will be a definite tradeoff between the power user functionality and flexibility we're used to and Windows S, which will be implementing a next-generation API with applications that you can only run if you get them from the Windows Store, and everything that that brings with it. So it's kind of like the closed environments that we see, for example, with Apple and the iOS environment where, yes, you do get better security, in trade for and losing a lot of the freedom and flexibility that can be convenient and handy.

And this brings me into the next topic, which I called "We're Losing Browsing Heterogeneity." BleepingComputer had a nice post. I spent some time there because they also had a lot of coverage about all of the WannaCry stuff. But I liked their coverage of this.

They wrote: "A one-liner in the Windows Store policy is the reason why we'll never have the original Chrome, Firefox, Opera, or other browsers available through the official

Windows Store. Included in the Security section of the Windows Store policy, this line is specifically addressed at browsers and reads the following," they wrote, quote from Microsoft: "Apps that browse the web must use the appropriate HTML and JavaScript engines provided by the Windows platform." And that's not just any Windows platform, that's the new universal Windows platform API applications.

So BleepingComputer goes on to say: "This means that every browser currently listed on the Windows Store is nothing more than an offshoot of Microsoft's EdgeHTML, the HTML and JavaScript engines found in Edge." And then BleepingComputer notes: "Apple and Google have similar policies." So this is not just something Microsoft is doing, and I don't mean to paint it as that. That's why this is bigger than that. What we're seeing is this loss of browser choice in platforms moving forward.

BleepingComputer says: "The policy is similar to what Apple has done with iOS, the company forcing apps to use its web rendering engine to process web content. Google took a harder stance with ChromeOS and forbade other browsers altogether. In both cases the companies cited security concerns as their engineers have worked to secure their operating system around those web rendering engines. In the case of the Windows Store," they write, "this revelation came to light after the launch of Windows 10 S, a tethered version," as they described it, "of Windows 10 that will only allow the installation of Windows Store apps.

"As users kept asking when will Google and Firefox port their browsers to the Windows Store so that they could use them in Windows 10 S, the question was finally answered last week when a developer tried to convert his Chromium-based browser to an .appx version compatible with Windows Store distribution. The Windows Store crew specifically told the developer that they cannot approve his browser because of the aforementioned policy that mandates that all apps that access the Internet use the approved HTML and JS rendering engines," JavaScript rendering engines. "The developer shared his experience and official communications with a ZDNet journalist."

So, "Microsoft's policy effectively bans standalone third-party browsers. While some were hoping to see Chrome or Firefox available on the Windows Store as UWP" - that's the Universal Windows Platform, the next-generation API that replaces the earlier Win32 and .NET platforms - "this may never be possible, as this would mean that Google, Firefox, and other vendors would need to rewrite their browsers from scratch to use Microsoft's EdgeHTML."

And they said: "This will never happen unless Windows 10 S becomes a huge success, and browser vendors see a benefit to port their browsers. In this case we won't see UWP versions of the original Chrome and Firefox engines, but only so-called bastard browsers," as BleepingComputer describes them, "like we have on iOS. For example, you can't call Firefox for iOS a true Firefox browser. It's just an older version of the WebKit engine with a Firefox UI lookalike on top, and lacking many of Firefox's original features.

"The conclusion is that Microsoft has effectively banned any self-standing third-party browser from the Windows Store. Additionally, Windows 10 S users better get used to using Edge or any of the other" - and here again they say - "bastard browsers that use Edge's repackaged core." And I'm a fan of Edge. As we've talked about on this podcast, it is a beautifully reimplemented from scratch, you know, they scrapped IE, and they started over, and they did a whole lot of things very right. I mean, it's a beautiful piece of technology. So I don't mean to disparage it in any way. It just means that, if you're going to go Windows 10 S, Edge will be your browser. Which is not a bad thing, but it also means your browser won't be, can't be, actual Firefox or Chrome. That will no longer be an option. And of course we also know that you no longer have a choice in search providers. If you go 10 S, Bing is your search engine, and that cannot be changed.

And finally the article ends, saying: "The original Chrome and Firefox browsers" - what we have today - "built around their native engines will remain accessible to Windows 10 users via standalone installers only." But again, 10 S will refuse them. So I just - I thought this was interesting because browsers are now the way we connect to the Internet. We've noticed that they are also the primary point of attack. They're the way bad things get into our systems and into networks, much more so today than it used to be when it was something quaint like email macros or Flash exploits. Now it's the browser.

And so it makes sense from Microsoft's standpoint, if they're determined to make Windows secure, then the freedom and the flexibility that we have that Windows has historically enjoyed, that will be lost. Windows 10 S will be a choice. And I guess I did see somewhere - I'm sure you're more up to speed on this than I am, Leo, because I have not looked at it. As I understand it, 10 S users have an opportunity to move to Windows 10 Pro, like for the rest of the year or something.

Leo: If you buy Microsoft hardware, yes. So if you buy the new Microsoft laptop, which comes with 10 S, and this is the only Microsoft device that does, you get a free upgrade through the rest of the year. And that's the only way they can get people to buy it because nobody knows if 10 S will do.

Steve: Right, right.

Leo: It might be another [crosstalk].

Steve: It might be another Windows Phone.

Leo: Yeah. Well, exactly.

Steve: So our friends at the OSTIF have...

Leo: Oh, them. Who are they?

Steve: Working with Quarkslab. They're the guys that have been financing a number of these open source audits, to everyone's benefit.

Leo: Good.

Steve: OSTIF.org. And they just finished, actually I know of this because they sent me a tweet. They sent a tweet saying: "Steve, you did excellent coverage of our VeraCrypt audit on Security Now!. We just published our OpenVPN results." And the results were very worthwhile. The audit was worthwhile, and it did drive a version change and update and fix from OpenVPN. I just checked the log on a FreeBSD machine, and it doesn't yet know about 2.4.0, which is now the latest and what anyone using OpenVPN should go to. It's not a massive problem. I mean, there was one vulnerability rated critical/high. So it's worth doing. But your OS will probably need to wait for a build for it of OpenVPN, and 2.4 is what you want. I had, I think, one of my machines is 2.3.13 or something. So they audited...

Leo: 2.4.1, is that...

Steve: 2.4.anything would be good.

Leo: Oh, yeah, good. Thank you, Arch.

Steve: Yes. So they audited OpenVPN and the NDIS6 TAP driver, the Windows GUI and the Linux versions. Oh, wait, wait, I'm sorry, I was wrong. They audited 2.4.0. And what was updated was 2.4.2, so you do need a dot increase to .2.

Leo: Checking my updates right now.

Steve: So the auditors wrote: "The public disclosure of these vulnerabilities coincides with the release of OpenVPN 2.4.2, which fixes all of the high priority concerns. OpenVPN," they wrote, "is much safer with these audits, and the fixes applied to OpenVPN mean that the world is safer when using this software." They wrote, "We have verified that the OpenVPN software" - and this is important - "is generally well-written, with strong adherence to security practices." And you want that because even an auditor can miss something. But it's worth taking some comfort in just the general conduct of the people who wrote this over time that they kept their eye on the ball, they didn't get sloppy, and that there was a lot of attention and focus put on or kept on the security aspect.

I do have a link here, you can just go to OSTIF.org to find it, but I've also got a link in the show notes to the whole report that goes into greater detail. But it was one critical/high vulnerability fixed, one medium vulnerability, and then five low or informational vulnerabilities and concerns. So again, as we've said, it's great to have the source open, but it only is useful if somebody who didn't write it, reads it. And these guys did. So again, thanks to the OSTIF for pulling together the funds to finance people who are security aware looking at their source code.

There's a bitcoin exchange, they believe that they are among the biggest, if not the, named Kraken, who describe themselves as a San Francisco-based world's largest bitcoin exchange in euro volume and liquidity. They write of themselves: "Kraken's clients also trade U.S. dollars, Canadian dollars, British pounds, Japanese yen, and other digital currencies on a platform consistently rated the best and most secure bitcoin exchange by independent news media."

They said: "Founded six years ago in 2011, Kraken was the first bitcoin exchange to have its market data displayed on the Bloomberg Terminal, the first to pass a cryptographically verifiable proof-of-reserves audit, a partner in the first cryptocurrency bank, and one of the first exchanges to offer leveraged bitcoin margin trading." Oh, wonderful. "Kraken is trusted by hundreds of thousands of traders, institutions, and authorities across the world," they write, "from Tokyo's court-appointed trustee to Germany's BaFin-regulated Fidor Bank." And, finally: "Kraken is backed by investors including Hummingbird Ventures" - oh, those are real guys - "Blockchain Capital, and Barry Silbert's Digital Currency Group, among others."

I say all that to preface their interesting comments about using and rethinking the security of phones, smartphones specifically, as authentication tokens. They wrote: "Heed this or perish." They say: "Let's begin with the assumption" - which is a little specious, but still it's what they want us to assume - "that within 24 hours your usual mobile phone number will be hijacked by social engineers." As I said, okay, well, maybe for some people. "They will use your number to gain access to every" - but certainly this is possible, if it happened. "They will use your number to gain access to every account you own that utilizes phone-based authentication and account recovery, such as your email. They will then use that access and information to compromise more accounts, and harass, steal, blackmail, and extort you and your associates.

"In the past month," they write, "there have been at least 10 cases of people publicly involved in the cryptocurrency scene being victimized by mobile phone hijacking." Okay, so targeted attacks. "The consequences have been expensive, embarrassing, enduring, and in one case life-threatening. If you are in any way publicly involved in cryptocurrency, consider yourself an active target. You need to immediately audit the security of your accounts, especially email, social media, social networking, and mobile phone. Somehow," they write, "the masses have been led to believe" - yes, well, and we know how - "that phone numbers are inextricably bound to identities and therefore make good authentication tools." Of course we on the podcast, as we've been covering recently, know otherwise.

They say: "There's a reason that Kraken has never supported SMS-based authentication. The painful reality is that your telco operates at the security level" - and I got a kick out of this - "of a third-rate coat check clerk. Here's an example..."

Leo: Where can you find a coat check clerk these days?

Steve: Yeah. "Here's an example interaction. Hacker: Can I have my jacket? Telco: Sure, can I have your ticket? Hacker: I lost it. Telco: Well, do you remember the number? Hacker: No, but it's that one right over there."

Leo: Oh, boy.

Steve: "Telco: Okay, cool. Here you go. Please rate us 10 out of 10 on the survey." Uh-huh. And I won't go on. The article goes on, for anyone who's interested. But I appreciated that. I thought, first of all, it was interesting that the known weakness in smartphone authentication via SMS, as we've discussed, is now being used increasingly in targeted attacks. For people who are not prone to being targeted, you shouldn't run around with your tinfoil hat on, and the sky is not falling.

But given a choice, you absolutely want time-based authentication, not an SMS message per instance. And if you can disable SMS in favor of anything else for account recovery, that would be good because remember that typically SMS is hopefully only an additional factor. Somebody first has to have your username and your password, and then also another second factor beyond knowing your password. The problem is it's often used for account recovery. I forgot my password. Oh, well, we'll send you a blurb to your phone number in order to recover it. Well, that's the huge Achilles heel in where we are today is account recovery.

And it's funny, I mean, in anticipation of this, SQRL already incorporates a facility for preventing that. Once a user becomes familiar with SQRL and realizes and understands how it works and gets it, they're able to set a sticky switch in SQRL that asks the sites they visit to please disable, preemptively disable any non-SQRL account recovery because, again, if you leave that there, then that's still a glaring hole in your authentication. So this essentially is sort of a beacon that your use of SQRL broadcasts, as you use it, which you would only turn on once you understand how it works and you're comfortable with it, so it's not on by default. But once you want to, as you visit sites again with SQRL, it makes a sticky setting at the site saying "Decline anything but my subsequent use of SQRL." So again, lots of nice little tidbits that have always been there.

And the good news for Android, speaking of happy tidbits, this is from the Android Developers Blog. They wrote: "On the Android team" - and I've paraphrased this down for the podcast, but I'll say it in their voice: "We view each dessert release as an opportunity to make Android better for our users and our ecosystem partners. One thing we've consistently heard from our device-maker partners" - and, yes, from this podcast - "is that updating existing devices to a new version of Android is incredibly time consuming and costly." As we know, like leaving IPv4 is time consuming and costly, so you don't do it unless you have to. Leaving SHA-1, same thing. I'd rather not.

So they write: "With Android O, we've been working very closely with device makers and silicon manufacturers to take steps toward solving this problem, and we're excited to give you a sneak peek at Project Treble..."

Leo: Oooh.

Steve: "...the biggest change to the low-level" - thank you, Leo - "a low-level system architecture..."

Leo: I did it wrong. Ooooooh.

Steve: The crowd goes wild.

Leo: Ooooooh.

Steve: Oooh. The biggest change to the low-level system architecture of Android ever.

Leo: I'm getting ready for tomorrow's keynote at the Google I/O conference.

Steve: Good. Oh, and you're going to be there, aren't you.

Leo: Yeah. And this is one of the announcements from it, I'm sure.

Steve: The traditional cumbersome and time-consuming update and patch flow has been this: First, the Android team publishes the open-source code for the latest release to the world. The silicon manufacturers, the companies that make the chips that power Android devices, must then modify that new release for their specific hardware. Next, the silicon manufacturers pass the modified new release to device makers, the companies that design and manufacture the Android devices. The device makers then modify that new release again as needed to customize it for their devices. The device makers work with the carriers to test and certify the new release. And, finally, the device makers and/or the carriers make the new release available to users. So is it any surprise that we're still using IPv4 and that most Android devices are way behind? That process is so clunky that it just doesn't happen.

"Project Treble," they write, "re-architects Android to make it faster, easier, and less costly for manufacturers to update devices to the new version of Android. Android has been a fabulous success" - patting themselves on the back, I mean, and it's true in the marketplace - "because it presented and rigorously enforced, through its compatibility test suite known as CTS, its application-layer API."

And, now, again, we understand because we've discussed it often in terms of the architecture of modern computer design. The reason that Windows was able to succeed as it did is that the Windows API allowed applications to be written to the API. And as long as future and succeeding versions of Windows maintained that API layer consistency, they could change the plumbing underneath, and apps were compatible. That's how Windows got its backwards compatibility. So what Android did with this compatibility test suite was to implement, literally it's a million different tests that validate and verify the consistency of that crucial interaction between applications and the Android OS, and everything that is not the application.

"Project Treble," they write, "will be doing for the Android OS framework" - that is, the under-plumbing, ooh, I like that term - "what CTS did for apps. The core concept is to separate the vendor implementation the device-specific, lower-level software written in large part by the silicon manufacturers from the Android OS framework." In other words, they're going to create another API layer, another interface layer way down in the OS, which will be similarly rigorously enforced.

They write: "This is achieved by introducing a new vendor interface between the Android OS framework and the vendor implementation layer. The new vendor interface will be validated by a Vendor Test Suite (VTS), analogous to the CTS" - which was the Compatibility Test Suite at the application layer - "to ensure forward compatibility of the vendor implementation." Or the other way to look at it is backward compatibility. As Android is updated and patched, those will no longer need any vendor involvement.

They write: "Once a stable vendor interface has been defined to provide access to the hardware-specific parts of Android, device makers can deliver new Android releases and updates to consumers only by updating the Android OS framework, without any additional work required from the silicon manufacturers." So the silicon manufacturers, they don't want any of this. They just want to pump out silicon. So they've been forced to become software developers to support their silicon. And unfortunately, they're always the starting point for the percolation of updates. It's got to go to them first. And then it moves back up the system, finally, maybe, eventually getting to the user.

So they say: "The underlying silicon manufacturers will not need to be constantly bothered and bombarded with updates which require their constant attention and involvement. This will all appear in Android O, and the new Project Treble architecture is already up and running on the Developer Preview of O for the Pixel phones." And O is slated for formal launch later this summer. And we'll find out more about it on tomorrow's Google presentation.

Leo: Yeah, I think so.

Steve: Neat. Oh, I bet, for sure. That's big news. That's - I'm delighted. I mean, we talk about the troubles that Android has, and we've been saying that at this point really the only recourse a really truly security-conscious person has is to stick with one of the very major players that are demonstrating that they are right on, really up to speed on keeping the phones current. This, though it's late, I think it's exactly what Google had to do in order to fix this problem with Android.

Leo: This is kind of like a Ring 0 kind of a thing.

Steve: Bravo, yes, yes.

Leo: So you have to be highly privileged.

Steve: Yeah. Well, I would say, what it is, it adds another layer. At this point Android only had two layers. It had the application layer and everything else. So there was one border definition which was the application API. What Google has done is they've added another layer. And Microsoft did this. That's the HAL, the Hardware Abstraction Layer in NT. It never really got used. Well, it did for a while because there was that funky MIPS support, remember, that NT used to - there was a MIPS support, and I want to say ARM, but maybe that was pre-ARM.

Leo: Oh, no, there was ARM, I think, yeah.

Steve: Yeah. There were a few other hardware architectures. So Microsoft's original NT, that whole redesign, the HAL, the Hardware Abstraction Layer, was meant to allow the NT OS to run on very different hardware. Again, what does abstraction mean? It means you create a virtual hardware definition. The software talks to that, and the hardware beneath it emulates that hardware definition. So that's what Android O does. It creates another sort of a slice, a cut point through the OS where everything hardware is below, and the hardware only has to meet the interface spec at that layer. And then Android OS only needs to talk to the interface spec. So nice piece of work.

And there is always a performance hit. Whenever you abstract and create a definition, what that actually is is an interpretation. It's an interpreter. And we know that those are never as fast as native. So it may well be that 10 years ago - Android launched in '07. So 10 years ago they couldn't afford the hit. They needed every bit of performance that they could get. And so just they couldn't do another interpretation layer down in the OS. They had to go full speed. Now chips have become so much more powerful, with ridiculous numbers of cores in a smartphone, that they say, yeah, now we can afford an interface layer for the hardware, and we're not going to feel it because the hardware has gotten fast enough to mask any interpretation overhead that might be introduced. So bravo.

Leo: Hey, hey. That's good.

Steve: And speaking of bravo, check this out. The NIST, what does that stand for, National Institute...

Leo: National Institute for Standards and Technology.

Steve: Yes, thank you, has updated their formal recommendations, removing, among other things, periodic password change requirements. Gone.

Leo: Yay, yay, yay.

Steve: I know. How many times have we said, "That makes no sense. Do not make people change their passwords." And so many corporate listeners of ours say, yeah, my company makes me change my password every month. And then we talk about all the workarounds, like sometimes you can't change to the one you had before, so then...

Leo: Oh, those are really crazy.

Steve: People have, like, probed the system and realized it has a six-deep stack, and so they change their password to six nonsense things and then back to the original one to force the memory off the back of the stack in order to not have to change their password. I mean, and anyway, so NIST said this guideline was suggested because passwords should be changed when a user wants to change them, or if there is an indication of breach, not just because some amount of time has passed. And a guy named Mike Wilson, who's a founder of PasswordPing, said: "There have been multiple studies that have shown requiring frequent password changes to actually be counterproductive to good password security." Yeah, surprise.

So anyway, the good news is remember the old slogan back in the early, well, back in the mainframe days of IBM? "Nobody ever got fired for choosing IBM." Even though they, like, had lost their edge, they were no longer like the best, other non-IBM solutions were arguably less expensive, higher performance, more reliable, better service contracts, I mean, they were, like, better. But it was like, ooh, crap, you know, what if I'm wrong? Well, nobody ever got fired for choosing IBM. Well, similarly, the NIST guidelines that were saying, yes, you should change your passwords periodically, well, IT departments probably had no choice but to follow the guidelines because then they couldn't get fired. It was like, nobody ever got fired for not doing...

Leo: Well, it's one less reason to fire you, anyway.

Steve: Yeah, exactly. So everybody listening, if your company is saying you must change your passwords, in the show notes I have the link to the entire document. In fact, that document is incredible. Actually, it was just published today, by the way, Tuesday, May 16. It just came out. And I don't know, I mean, I've got a tiny little scroll thumb here. I don't know how many equivalent pages it is. It goes on and on and on and on. Anyway, it'll be the topic of a future podcast because there's a ton of stuff about authentication in there that I think will make some interesting material for our listeners.

Leo: Good, good, good.

Steve: They also say drop the algorithmic complexity song and dance. "No more arbitrary password complexity requirements needing mixtures of uppercase letters, symbols and numbers," NIST wrote. "If a user wants a password that is just emojis, they should be" - god, I wonder how this happened at NIST. They must have just had a...

Leo: They read some papers, I guess.

Steve: Yeah.

Leo: Somebody got in there.

Steve: Wow. Just emojis, "they should be allowed." They said, "It's important to note the storage requirements." And of course, as we know, salting, hashing, and MACing such that, if a password file is obtained by an adversary, an offline attack is very difficult to compromise.

Leo: We also know that it's better to have variety in your password.

Steve: Well, it's better to have...

Leo: But you don't want to tell the hacker what the limitations are. Right?

Steve: Correct. Yeah. What you always want is entropy. Entropy is the key. So however you obtain entropy, that's what you want. And so, for example, Mike here, again from PasswordPing, said: "Like frequent password changes, it's been shown repeatedly that these types of restrictions," that is, on password complexity, "often result in worse passwords."

Leo: Yeah. That's the other reason, yeah.

Steve: So let people do what they want, but enforce. Enforce.

Leo: Right.

Steve: And again, I have sort of a nice compromise in SQRL. I built an on-the-fly complexity evaluator, but it doesn't work by enforcing any specific guidelines. Users can do anything they want. It's just that the better their passwords are, the shorter it can be and rank a higher level of complexity.

Leo: Clever. [Crosstalk].

Steve: So that's built in there, too. I figured while I was at it. Oh, and then, finally...

Leo: Now NIST-compliant.

Steve: Oh, goody.

Leo: SQRL, now NIST-compliant.

Steve: And you are never required to change your password. But finally they said: "Require screening of new passwords against lists of commonly used or compromised passwords."

Leo: No more monkeys.

Steve: It's right here in the show notes. It says "No more monkeys."

Leo: I wasn't reading that. That came out of my head, straight out of my head.

Steve: And mine because of course we've all had a lot of fun with that. "NIST notes that dictionary words, usernames, repetitive or sequential patterns all should be rejected. One of the best ways to ratchet up the strength of users' passwords is to screen them against lists of dictionary passwords or known compromised passwords." And of course "monkeys" used to be number six. It's fallen down to 123456, unfortunately. Anyway, we will come back to these guidelines in the future because, oh, this document is full of goodies.

Justin Garrison shot me a note saying: "Quick correction on last week's Security Now! 611 with respect to AMT over WiFi. AMT can be set up with Intel WiFi chips, but it does require additional settings to be enabled." Paraphrasing, if you have a laptop that supports AMT, where AMT has been provisioned, if AMT has had its optional wireless support also turned on, and if you're running Windows, then connecting your laptop to a public wireless network means that AMT is accessible to anyone else on that network, that is, your AMT in your machine, the null authentication bypass vulnerability, is present. So if that machine has not received a firmware update, anyone will be able to access the AMT system within that machine using the null authentication bypass. If you're a corporate IT department, and if you have AMT enabled over WiFi, turn it off, now. And I've got a link for additional information in our show notes. And Leo, we have reached our lighter side moment.

Leo: Ah.

Steve: A two-minute-and-40-second, brilliantly conceived and assembled, spoof on the Amazon Echo which appeared on last Saturday night's SNL, "Saturday Night Live." If you google "Saturday Night Live," or I guess you can just google "Amazon Echo Silver"...

Leo: That'll do it.

Steve: ...our listeners can find it. But I wanted to play the audio and show it to our podcast video listeners, just because it's a bit of really brilliant fun.

Leo: And what's really nice about this one is it works really well in audio, as well as video.

Steve: Yeah.

Leo: And I have but one word to say, which is "Alessandra." Listen.

[BEGIN VIDEO CLIP]

ANNOUNCER: The new Amazon [indiscernible] has everyone asking Alexa for help.

ELDERLY PERSON: Alyssa, what time is it? What the hell is wrong with this blasted thing? Amanda.

ANNOUNCER: But the latest technology isn't always easy to use for people of a certain age.

ELDERLY PERSON: Kids done bought me a busted machine again. Odessa.

ANNOUNCER: That's why Amazon partnered with AARP to present the new Amazon Echo Silver, the only smart-speaker device designed specifically to be used by the greatest generation. It's super loud and responds to any name even remotely close to [indiscernible] so they can find out the weather.

ELDERLY PERSON: Allegra, what is the weather outside?

ECHO SILVER: It is 74 degrees and sunny.

ELDERLY PERSON: Huh?

ECHO SILVER: It is 74 degrees and sunny.

ELDERLY PERSON: Where?

ECHO SILVER: Outside.

ELDERLY PERSON: What about it?

ECHO SILVER: The temperature outside is 74 degrees and sunny.

ELDERLY PERSON: I don't know about that.

ANNOUNCER: The latest in sports.

ELDERLY PERSON: Clarissa, how many did Old Satchel strike out last night?

ECHO SILVER: Satchel Paige died in 1982.

ELDERLY PERSON: How many he hit?

ECHO SILVER: Satchel Paige is dead.

ELDERLY PERSON: He what, now?

ECHO SILVER: Died.

ELDERLY PERSON: Who did?

ECHO SILVER: Satchel Paige.

ELDERLY PERSON: Huh. I don't know about that.

MALE VOICE: Even local news and pop culture.

ELDERLY PERSON: Anita, what them boys up to across the street?

ECHO SILVER: They are just playing.

ELDERLY PERSON: They what now?

ECHO SILVER: They are just playing.

ELDERLY PERSON: You say they're just playing now?

ECHO SILVER: Yes, they are just playing.

ELDERLY PERSON: I don't know about that.

ANNOUNCER: Pair it with smart devices like your thermostat.

ELDERLY PERSON: Oh, Sandra, turn the heat up.

ECHO SILVER: The room is already 100 degrees.

ELDERLY PERSON: Are you trying to kill me, Alizay?

ANNOUNCER: The new Amazon Echo Silver plays all the music they loved when they were young.

ELDERLY PERSON: Angela, play black jazz.

ECHO SILVER: Playing, uh, jazz.

ANNOUNCER: It also has a quick scan feature to help them find things.

ELDERLY PERSON: Amelia, where did I put the phone?

ECHO SILVER: The phone is in your right hand.

ANNOUNCER: And it has an "uh-huh" feature for long, rambling stories.

ELDERLY PERSON: So then I gave him $5. And he said I only gave him $1.

ECHO SILVER: Uh-huh.

ELDERLY PERSON: I said, "I know I gave you a five."

ECHO SILVER: Uh-huh.

ELDERLY PERSON: Because I only had a five and a one on me.

ECHO SILVER: Uh-huh.

ELDERLY PERSON: And here's the $1 right here.

ECHO SILVER: Uh-huh.

ELDERLY PERSON: Well, I mean, you tell me who's crazy.

ANNOUNCER: Amazon Echo Silver. Get yours today. I said, GET YOURS TODAY. To order Amazon Echo Silver send a check or money order to Amazon.com right now.

[END VIDEO CLIP]

Leo: So funny. I've played it now several times, and it doesn't get any less funny. It's still amazing.

Steve: It's just, well, yeah. The acting is great, the various performances, and just well conceived.

Leo: Yeah, yeah, yeah. Very nice.

Steve: A couple weeks ago, Leo, a caller to your weekend show was asking you about adding smartphone connectivity to an older auto.

Leo: Right.

Steve: And it may not surprise you to know that I have a solution. It turns out...

Leo: You have an older auto, as well.

Steve: I do. And I now have a hands-free speakerphone on my phone, and a USB that can play a thumb drive directly in the audio system without the need to use the cigarette lighter FM transmitter kludge. It's a little more expensive, but this is an amazing company. It's called Grom Audio, G-R-O-M A-U-D-I-O, GromAudio.com. They have a range of different retrofits for a huge range of older autos. So I just wanted to share that both with you for the future and also with our listeners, who may be wanting to look at increasing an older auto's connectivity without exposing it to more dangers from the Internet and so forth.

Leo: How does it connect to the automotives?

Steve: All of the in-dash entertainment systems have a big monster plug on the back, like for supporting CD changers and other add-ons.

Leo: Right, right. You have to get under the dash to do it.

Steve: Yes. So installing it does take some time. A buddy of mine has, I think it's an '05 Infiniti. And he was complaining that when he rents cars for work they have all these new features, but his 12-year old Infiniti doesn't. Anyway, Mark took his entire center console apart one weekend after buying one of these, installed it, and now he's got the tunes playing from his - I think he has attached - he used the older iPod connector because that's the iPod that he had. And so he's got his iPod playing in his 2005 Infiniti.

Leo: Very nice.

Steve: So anyway, GromAudio.com, for anybody who's interested. Jonathan Bennett sent me a note saying: "SpinRite owner and long-time listener here. My son just introduced me to a puzzle game I thought you might like for iOS and Android, Squaredance." I saw this as I was catching up in my Twitter feed to prepare the show, so I've not looked at it. But I've never seen, well, not recently seen anything rated so highly. It's got almost 100% five stars. Apparently, it's great. So I wanted to thank Jonathan. I can't vouch for it yet, but I will give our listeners an update next week.

And then he finished, saying: "I have found the Frontier Saga audiobooks available through my library. Thankfully, each book is short so I can fit in Security Now! between books, and I don't have to fall too far behind. You were right," he wrote, "the action has been nonstop so far. I'm not quite done with the third book, but wow. Thank you for all you and Leo do." So Jonathan, thanks for the pointer to Squaredance on iOS and Android. Looks great. Looks apparently really cool. So say the reviews that I read.

And, finally, Garrett Bane, who is a longtime listener, wrote: "I used my personal copy of SpinRite at Level 2" - as he should. He says: "...on my work laptop with an SSD" - thus Level 2 because you only want to do a read scan - "to confirm that the drive was, in fact, dying. The early warning allowed me to pull all my backups together and prevent any loss of data. Thanks to listening to SN since day one from my iPod, I have a better understanding than most of the Level 1 techs here. So glad for how far pod," he says, "netcasts have come, and I appreciate all the work you and Leo put into the show. I'm looking forward to 6.1, 6.2 and 7" - he's talking about future SpinRite releases - "when we will see performance improvements and will be able to run it on a Mac. Please feel free to share my story/testimonial on Security Now! and use my name. Thank you. Garrett, Jackson, Wisconsin." And Garrett, thanks for sharing it, as well, and helping me remind our listeners about SpinRite.

Leo: All right. I think it's time to talk about WannaCrypt. Almost.

Steve: We've got a little closing-the-loop feedback to do.

Leo: Oh, good, okay.

Steve: Someone whose Twitter handle is @vega_ska said - he asked an interesting question, something that I don't think we've ever talked about, regarding load balancing. He said: "Google's DNS resolves to 12 IPs, but Google.com resolves one IP. How does this work?"

This is an interesting bit of incredible forethought by the original designers of DNS. The idea is that any domain can have, that is, any DNS server can present more than one IP for a given domain name. So Microsoft.com, I think, is the same way. Google.com, if you actually use a tool like Nslookup or Dig, you'll see that you get back a block of IPs. Every DNS server which answers a DNS query that has a block of IPs, whether it's the authoritative original server or a caching server somewhere out on the Internet, that previously asked and the answer has not yet expired, rotates the list of IPs every time it is asked for the IP list for a given domain. So there's automatic built-in, round-robin load balancing built into DNS queries.

The beauty of that result is that typical users or consumers of the IP list from a DNS query start with the first one. And they go with it if they get a connection there. But if no machine is there, then they go to the second one and the third one and so forth. So just by simply allowing a list of IPs, and by formally setting the practice of rotating that list circularly, you get the best of all worlds. You get that the first one in the list is always rotating among all the people who ask, so thus the balancing of the load among all the IPs. And you get the fault tolerance of, well, if the first one in the list doesn't answer, for whatever reason, you go to the next one. So, neat question, and I just thought it was a - it's something we've never talked about before. Believe it or not, there's still a few things like that.

Someone who is a frequent Twitter participant, Glasair Pilot, asked: "Why would engineers spend the money for Android mics that have a frequency response into the ultrasonic?" And that's something we hadn't touched on when we were talking about the prevalence of ultrasonic monitoring and spying and tracking. And the answer is that, in this context, ultrasonic doesn't mean like medical ultrasound in the super high kilohertz or even low megahertz range. It only means non-audible, which is to say anything above 20 Hz. And the fact is many very small microphones have a natural frequency response well above 20 kHz, maybe 25, maybe even 26, 27, 29. So it's not that they're more expensive or that there was any intention behind the fact that the microphones that are used in smartphones just happen to extend beyond our hearing range, it's just that mechanically they do. And so people realized, oh, look. Well, like the EKG machine that you really like, Leo, the cute little portable EKG.

Leo: Heart attack, yeah.

Steve: Yeah, that uses a frequency-modulated ultrasound. But again, in this context, I mean, "ultrasound" does have a formal definition. Well, technically the word means "supersonic," above sonic. And maybe that would be a better term, instead of saying "ultrasonic," to say "supersonic." Meaning just above our hearing range.

And then Chris Ebert, sort of on the same lines, asked, he says: "It seems it should be possible for mobile OSes to filter ultrasonic frequencies from the microphone before providing it in the API." And I agree. I think it's just something that they don't do. Maybe they will. It's kind of handy. And the bad news is, if they did, then the cute little frequency-modulated supersonic EKG machine would stop to work, as would - maybe there are legitimate purposes for it. So, but maybe the user should have the choice of turning on a supersonic filter on their microphone for that purpose. Or maybe it'll just be added in the future.

Oh, and Michael Cunningham, another frequent Twitterer, brought a Ubiquiti router firmware update to my attention. We've been previously talking about the non-EdgeRouter updates that Ubiquiti released for all of their airOS machines that had some concerns. This one does apply to our cute little EdgeRouter X boxes. Oh, and by the way, Elaine got one last week. So Elaine is following on - not only does she look up all the words and make sure everything is spelled correct, but she's following along with some of the security advice and now has herself a little Ubiquiti EdgeRouter X. And she wrote me, said, "It is adorable, just like you said."

Anyway, they fixed that UDP problem we talked about a few weeks back, which is the vulnerability of the - remember the MSG_PEEK flag where software might look at the UDP packet's contents without removing it. And the code for checking the checksum, the "unsafe second checksum" was what the problem was called, that was where the problem was. So on April 28 Ubiquiti released v1.9.11 for the wired Ubiquiti EdgeRouter family. There's the X, the non-X, and several others. So if you are an owner of those, the ones we've been talking about and really like, you might want to check with Ubiquiti for a firmware update. You want v1.9.11 or maybe later. They don't do them very often, so I imagine that's the one that's there now.

Andrew Douglas asked: "So how do TOTP apps work?" He says, "I use them, but have no idea how those six magic numbers work. Any chance you could dig in a little on Security Now!?" And I'll just say quickly that it's very cool. Basically it is a cryptographically driven time sequence. And the security requirement is that the sequence be unpredictable, and that at any given instant there be no way for a bad guy to predict what the six digits will be based on any knowledge of the previous six digits. So we accomplished that by using what's known as a keyed HMAC. Well, that's sort of redundant. A keyed MAC, a Message Authentication Code, which is an HMAC.

An HMAC, a hashed MAC, is a well-known security primitive which uses a hash several times in order to mix a secret key into the hashing process. So we feed the time of day into this keyed hash, where the key is a secret, and out comes, depending upon the size of the hash, a binary blob from which six digits are extracted. And that's your token. And every 30 seconds the time is updated, and then that is rehashed with the same secret in order to get a different six digits. So it's as simple as that.

And, finally, we actually heard from the Department of Defense, who weighed in on the exploding missile logic. We'd talked about how it really wasn't necessarily to do garbage collection and worry about memory leaks if you were writing the flight control software for a missile because, as long as the memory would not become exhausted within the expected maximum flight time of the missile, the unreleased memory leak was not going to be a long-term problem. This person wrote: "I work with the DoD. Missile RAM issue: It's likely far, far cheaper to double the hardware" - meaning the RAM - "rather than pay a contractor to fix the memory leak. (Contract ending, other important bugs to fix, et cetera). After all, what's the hardware cost per missile? Likely very small. How many missiles? Likely not that many." So we've heard from our listeners about various topics from last week.

WannaCry, or Makes You WannaCry. So what do you get when you combine an extremely old and powerful, always on and present, networked, remote code exploit requiring no user interaction, with moneymaking file encrypting malware, the inherent vulnerability of social engineering phishing attacks, and then toss in a capable backdoor while you're at it? Well, it makes you WannaCry. This global, potent, and capable cryptomalware payload, which leverages the NSA's leaked, ubiquitously present, and well-weaponized EternalBlue Windows SMB vulnerability for propagation, is what we got.

Now, there was a lot of disagreement about the name for this thing. The reason is it never formally declared its own name. I went with WannaCrypt, although I added a second "N" because it seems a little more grammatically correct. It's been called WCry, WanaCryptOr, WannaCry, Wana Decryptor, et cetera. For me, what was definitive was the source code refers to itself in several places as WanaCry. And, more importantly, the encrypted file header, that is, the new header added to every file that it encrypts, the first eight characters are W-A-N-A-C-R-Y-!. So WannaCry. I think that is probably the correct name for this. And people were a little confused because Wana DeCryptor is the name that's also present. But that's, of course, the decrypting side, the part where you're no longer crying, except maybe over the bitcoins that you just had to spend, but you're in the process, hopefully, of getting your files back.

Although I watched whoever it was, Junior Cyber Somebody, come out before yesterday's press briefing with Sean Spicer. And what he said seemed to have about as much mis-fact as it did correct fact. So I don't know what to believe from what he said. He said that there had been no evidence of anyone decrypting their files after receiving payment. He's the only place I ever heard that said. So I would always look for multiple sources of confirmation. I assume that people who pay get their files back. But he threw his statement, which was the official U.S. position statement, threw that into question for me. So I don't know one way or the other.

We do know that today there have been at least five identified variants. And I've seen infection estimates fall between 200,000 and 300,000. Yesterday the same person said 300,000. And there is the site which was created by the attacker who created the inadvertent kill switch we'll talk about in a second. And I glanced at it this morning, and it was now above 300,000. I don't remember how far above, but it was north of 300,000. So assuming that he's doing IP single counting and not multiply counting IPs, he had set up a web server at this bizarre domain, and the code in the WannaCry infection checks that site for a connection. If it obtains a connection, just anybody answering TCP, it then no longer proceeds. So that does allow him, not only to neuter the infection, but simultaneously track a count of how many infections are present.

Earlier I read that it was only looking - it was performing a DNS lookup, and that he simply looked up - that the malware performed a DNS query. If it got a resolution to DNS, it didn't infect. However, I subsequently saw the source code, and the source code actually performs a connection with an integral lookup as part of it. So you do have to have something there answering a TCP connection. Not even a web server, just a hello, yes, here's a TCP handshake, and then this thing shuts down.

So this is a worm by the classic definition of a worm, meaning something that is self-propagating with no user input, which itself, from a single instance, has the potential of taking over a large number of other systems, meaning that the infection itself is also a scanner which successfully scans for other machines to infect. So it meets the definition. When it installs itself, it looks at the local block of IPs where it is and immediately scans the /24 subnet, that is, the 256 IPs in its own network neighborhood, for anything else, that is, any other machines to infect.

So it is a virulent Intranet infector. And in fact it is believed that's how things like the U.K.'s NHS network, and even FedEx here in the U.S., and also the Telefonica telephone ISP in Spain got themselves massively infected was that somehow a single point of entry was obtained, and then these organizations had huge Windows infrastructures with, it should be noted, un-updated Windows machines, that is, not since March, or maybe un-updatable Windows machines that we'll talk about in a second.

So there's still some question among the security community whether there was also a phishing campaign to get this into organizations. Spain's Telefonica believed that there was a PDF file containing a .hta, an HTML application exploit, that someone, you know, a phishing email, clicked a link. And in this case, thanks to EternalBlue, it didn't just compromise that person's machine, but it got in, and then it went crazy within the Intranet. So this follows what we talked about when we were talking about EternalBlue originally was, because it's a Windows SMB exploit, I rhetorically asked, how can this be a danger on the Internet? Well, it turns out there are tens of thousands, as we later learned, tens of thousands of apparent SMB 445 ports exposed publicly. And all it takes is one.

It takes one machine somewhere in an organization that has that exposed, one way or the other, for the other WannaCry worms to find it. Because not only, when WannaCry sets up, does it start looking on its own Intranet, it also launches 128 public Internet scanning worms to scan all of the public Internet. It uses the high-quality cryptographic entropy source built into Windows, if available. Otherwise it falls back to a pseudorandom number generator that's still just picking IPs, 32 bits, at random, and tries to make a port 445 connection. So that's what happened was that, once this thing started, it found all the other publicly available SMB ports on the Internet. And with each one that it found, it infected it, and that one became a scanner in pure Code Red, Nimda, MSBlast style, the kinds of attacks on a global scale we haven't been talking about, the world hasn't been subject to, for years.

So why was the U.K.'s NHS hit so hard? By some estimates, approximately 90 percent of the NHS in U.K. has remained on Windows XP and did pay Microsoft hugely, I saw a number, 5.5 million pounds quoted, I think it was pounds and not euros, to extend support for an additional year past XP's patch death. And I think I remember us reporting that at the time, that they decided to just buy another year of support because they wanted it. But...

Leo: Well, they're regretting that now.

Steve: That was in 2014.

Leo: Because Microsoft didn't offer a patch even if they bought support; did they?

Steve: Correct. That only gave them one year of extended support after the end of the extended support. So that brought them up short.

Leo: You can't - I have to say, though, and I've talked to our IT guy Russell about this. He's got a few hospitals. They're really, every time you do an upgrade, they have to recertify the stuff. And it's not - there are quite a few businesses for whom moving to the newest versions is expensive and not a great idea.

Steve: Which is the perfect segue to the next topic. I have it here in bold: Is blaming the victim here justified?

Leo: Right, right. It's tempting.

Steve: I said it's easy to do, but I think it's more subtle than that in the real world. In today's environment, everyone should accept all available updates. That's clear. But notice the way I deliberately phrased that: All available updates should be accepted. What about when updates are no longer being made available? What about when XP, embedded in a decade-old MRI scanner, which is working fine otherwise, or a passenger ticketing kiosk, or a banking ATM machine? In many cases their manufacturer would love to profitably sell an updated system.

But what if the one you have is already working just fine? Or if there isn't budget available for wholesale replacing 90% of a massive IT infrastructure? Or what if the company who created the devices 10 years ago is long since gone? And as you say, Leo, the need for compliance to have certification of any changes. They're in the real world. There are real reasons for a sane IT policy of, okay, we're going to attempt to retain security oversight and control to the best of our ability. And in this case that failed.

Leo: It's why hospitals get bit disproportionately by these attacks.

Steve: Yes. I mentioned this thing having a weird kill switch. A security researcher, looking through the code, found this bizarre domain.

Leo: No, you're not going to read it.

Steve: No, I'm not. I'll just give people a sense for it.

Leo: I can zoom in on it.

Steve: Iuqerfsodp9ifjapo and so forth. That's about a third of it.

Leo: Looks like a Gaelic town is what it looks like.

Steve: That's just not, you know, clearly a big long piece of gibberish dotcom. The code uses an older Windows TCP connection function to attempt to make a TCP connection. If it succeeds, for whatever reason, it shuts down. So what this means is, if there is a server there, do not infect. So it was clearly a kill switch. Or maybe he put it in, and he was going to write some other code, but then forgot to? And so the default of making a connection just said, oh, instead of doing what I was going to do, do nothing. Who knows what's behind it.

But what was discovered was the hacker registered the domain, set up a server to monitor the infection, and discovered that not only did it monitor it, it shut it down, so it stopped infecting. The problem is that it was, well, first of all, all the variants that have subsequently appeared don't have that. And there's even a variant where that domain and that feature has been hex-edited out of the binary.

Leo: Okay. Now, I have a theory about all this.

Steve: Okay.

Leo: And do you want to hear - now, it starts with this morning's report from NPR that North Korea might be involved in this. And the reason some security researchers think so is there's a lot of commonality in the codebase with the Lazarus attackers, the Lazarus Group who did, of course, Sony. Some of the code is very similar. So, now, of course code can be copied. But when I first saw this, and I was wondering what you thought, the fact that it was translated into, like, 15 different languages...

Steve: It was very language-complete, yes.

Leo: And it was also very helpful. It had links to what bitcoin is and where to buy bitcoin. The fact that they raised so little money, but put so much effort into it, and then this kill switch, kind of made me think, governmental operation? And so the theory - and of course it's hard to confirm because you can't. But the theory is perhaps this was not an attempt to raise money, but to cause global disruption or some other things. By the way, it didn't hit the U.S. hardly at all. It mostly hit Russia and Asia before it got shut down.

Steve: I would argue, though, that is a consequence of the install base.

Leo: Yes, possibly, yes.

Steve: You can run pirated Windows XP very easily.

Leo: And something like this doesn't get geographically contained very easily. I understand that, as well.

Steve: Right, right.

Leo: There's no conclusive evidence at all. But it isn't inconsistent with a hacker that might have motives other than mere financial gain.

Steve: To that I just shrug.

Leo: What are you going to say; right?

Steve: Yeah. I just don't have an opinion on that. But, yeah, it's certainly possible. I did find a couple interesting little bit of technical details. My favorite one was to look a little more closely at what the actual EternalBlue exploit was. And it turns out - get a load of this, because we've talked about these sorts of problems. There is, in this SMB code that's been around forever in Windows, is a very subtle buffer overflow in a memory move operation. The size of the move is calculated. But the calculation has a type error in the math. A dword, that is, a 32-bit double word, is subtracted into a word. So again, a very subtle error. It's not clear to me whether someone would have looked at the source code and spotted that.

The fact that it's from the FuzzBunch makes me think that it was some sophisticated fuzzing, meaning that a whole bunch of crap was being thrown at the ports of a Windows machine, logging what was being thrown, in case the junk that was being throw at it caused it to crash. Because, if so, then they would go in and forensically determine when exactly the crash occurred, what had been thrown in it that caused it, and then reverse engineer the exploit from there. Which is the way a lot of these things have been found. So that's my guess. But I thought, oh, yep, here once again a dword was subtracted, and the result was a word. So the point being that the subtraction tried to be stored into a smaller container, a 16-bit value, from a 32-bit subtraction, which then wouldn't fit. And you could get up to all kinds of mischief that way.

And, finally, Brad Smith, Microsoft's President and Chief Legal Officer, weighs in. He has sort of a rambling post about social responsibility and so forth, in which he takes no responsibility for this. The title of his posting was "The need for urgent collective action to keep people safe online: Lessons from last week's cyberattack."

And I snipped all of the preamble and conclude with him saying: "Finally, this attack provides yet another example of why the stockpiling of vulnerabilities by governments is such a problem. This is an emerging pattern in 2017," he writes. "We've seen vulnerabilities stored by the CIA show up on WikiLeaks, and now this vulnerability stolen from the NSA" - notice he didn't say, I mean, there's no punches pulled here, it's like no "maybe" or "believed to be" or anything.

He just says: "Now this vulnerability stolen from the NSA has affected customers around the world. Repeatedly, exploits in the hands of governments have leaked into the public domain and caused widespread damage. An equivalent scenario with conventional weapons would be the U.S. military having some of its Tomahawk missiles stolen." Okay. "And this most recent attack represents a completely unintended, but disconcerting link between the two most serious forms of cybersecurity threats in the world today: nation-state action and organized criminal action."

Finally: "The governments," he writes, "of the world should treat this attack as a wake-up call. They need to take a different approach and adhere in cyberspace to the same rules applied to weapons in the physical world." The problem with that is attribution, of course. He says: "We need governments to consider the damage to civilians that comes from hoarding these vulnerabilities and the use of these exploits. This is one reason we called in February for a new 'Digital Geneva Convention' to govern these issues, including a new requirement for governments to report vulnerabilities to vendors" - yes, Microsoft would like that - "rather than stockpile, sell, or exploit them." Or lose them. "And it's why we've pledged our support for defending every customer everywhere in the face of cyberattacks, regardless of their nationality." And regardless of the OS's age. "This weekend, whether it's in London, New York, Moscow, Delhi, Sao Paulo, or Beijing, we're putting this principle into action and working with customers around the world."

Leo: Well, he's right, of course. It would be great if they didn't do it.

Steve: Yes.

Leo: But they never won't do it because - and if governments don't, well, then, bad guys will.

Steve: Yup.

Leo: So that's foolish optimism.

Steve: It is a conundrum. I mean, I get the NSA's plight. They're wanting to use things they find for their own purposes. And if they tell Microsoft, Microsoft will fix them. And, unfortunately, the worse they are, worse in this context, the more valuable they are because the more powerful they are, and the more they can leverage them as they need. So, boy. As you said, Leo, it is the world we live in today.

Leo: I think they want a variety as opposed to power because they're, in theory, not doing mass collection.

Steve: Right.

Leo: They're targeting individuals. So a variety of exploits that work in a variety of situations would be more useful. But that's why they stockpile them.

Steve: Well, and remember, EternalBlue was only one thing.

Leo: We're going to see so much more of this.

Steve: Yes. The thing that made this worm a worm, that gave it its teeth, was the EternalBlue component. We've had cryptomalware for a long time, and it's a problem, and it's annoying. But it's not this. It was the...

Leo: Well, did it make this that much worse?

Steve: Yes, yes.

Leo: Okay.

Steve: That was the worm. It was an SMB worm, and that's what brought down NHS in the U.K. is that it was the wormness of this, and the fact that it encrypted all the system's files. So this is to your point, Leo. For the NSA to have EternalBlue doesn't mean the NSA has a worm. What they had was a powerful SMB exploit that could be used as a worm, as it was, but they could also use it for surgical intrusion, which is probably what they were using it for.

Leo: Much more likely, yeah.

Steve: Yes. And just one final note. Simon Zerafa, also a frequent contributor through Twitter, said: "No WannaCry Ransomware on Windows XP. So why did MS release the XP patches?" And this follows from somebody, a security researcher who noted that there were aspects of the code that they believed wouldn't function in XP. Well, clearly the NHS got obliterated as a consequence of XP.

Leo: Apparently it works okay. It worked well enough.

Steve: But I would also argue that, with this SMB exploit having been developed into a worm that was this bad, Microsoft was forced to move. And this of course is very reminiscent to the MSBlast worm because Microsoft was adamant about ignoring my years of begging them and warning them about raw sockets in XP until the MSBlast worm used raw sockets in Windows to blast the crap out of Microsoft and scared them and forced them to remove raw sockets from XP, or to dramatically neuter them to the point where it could no longer be used for this purpose. So Microsoft doesn't listen to anybody unless they absolutely finally have no deniability and no other choice. And so, yes, they fixed this in XP also. Yay. I have an update.

Leo: I mean, I think they're doing now - one of the things you're critical of them with the browser thing is they're trying to create with 10 S a more secure operating system.

Steve: I'm not critical of them.

Leo: Towards this goal. But, well, you see the downside is...

Steve: They're creating a choice.

Leo: Yeah, and they'll be kind of a homogenous system as a result.

Steve: Oh, I don't want to use Bing. I just...

Leo: Yeah, I know.

Steve: I have an anti-Bing bias.

Leo: Yeah. It's an interesting puzzle because, you know, I understand why Microsoft says, well, we can't turn off SMB1 on all our systems ever made because people rely on it. And that would hurt those hospitals just as much as this did.

Steve: Yup.

Leo: That's a very difficult thing. The only thing we know for sure is it's only going to get worse, a lot worse, in my opinion.

Steve: Well, and rather than turning it off, they might as well patch it. I mean, you know, now it's fixed.

Leo: Yeah. But they didn't know the flaw was there until the Shadow Brokers dumped the exploit from the Equation Group.

Steve: And then they immediately responded. I mean, and we think that made...

Leo: Yes. They got it patched two months ago.

Steve: Yes.

Leo: Just not everybody applied it. I don't know. And then another thing they're doing, they're requiring people with Windows 10 to apply patches. You can only defer for so long.

Steve: And they didn't back patch, either. And now they have. But notice that they didn't, even though it was really, really, really bad, they didn't do it until they had to.

Leo: You'd better believe they're going through those other Shadow Broker exploits, and they'll be back patching all of them. What did they do yesterday? There was a bunch of stuff.

Steve: Yeah.

Leo: Steve Gibson. He's the guy. Thank goodness we have him. He's the man in charge at GRC.com, Gibson Research Corporation. That's where SpinRite lives, the world's best hard drive maintenance and recovery utility. You can get your copy there. You can also get your copy of this show there, and not only 64Kb audio, but nice transcripts that you can follow along as you listen, or search and find the part that you're most interested in. A lot of people find it easier to understand Steve if they're reading at the same time and their brain gets it in two different ways. I would read, listen, and watch. Do it all.

We do like you to check out the video. That's available at TWiT.tv/sn, as is the audio. You can also subscribe in your favorite podcast utility. Actually, that's the best way, and then you won't miss an episode. And I think this is like the 10-foot shelf of technology. You want to get every episode in your collection so you can always go back and learn because a lot of this stuff is still the same. There's a lot of great stuff in these 600-some episodes. Steve, we'll be back here next Tuesday, 1:30 Pacific, 4:30 Eastern, 20:30 UTC. What do you think?

Steve: I'm ready for it, my friend.

Leo: I'll see you then on Security Now!.

Steve: Thanks, Leo.