Vulnerabilities Galore!

Description: This week we discuss a new non-email medium for spearphishing, Chipotle can't catch a break, social engineering WannaCry exploits on Android, video subtitling now able to takeover our machines, a serious Android UI design flaw that Google appears to be stubbornly refusing to address, Linux gets its own version of WannaCry, another dangerous NSA exploit remains unpatched and publicly exploitable on WinXP and Server 2003 machines, a look at 1Password's brilliant and perfect new Travel Mode, Google extends its ad tracking into the offline world, some follow-ups, miscellany, and closing-the-loop feedback from our terrific listeners - concluding with my possibly useful analogy to explain the somewhat confusing value of open versus closed source.

High quality (64 kbps) mp3 audio file URL: http://media.GRC.com/sn/SN-614.mp3
Quarter size (16 kbps) mp3 audio file URL: http://media.GRC.com/sn/sn-614-lq.mp3

SHOW TEASE: It's time for Security Now!. Steve Gibson is here. We have, as he says in the title, vulnerabilities galore to talk about, including WannaCry on Linux. He'll also give you his reviews of a couple of, well, "Alien," the new "Alien" movie, "Alien Covenant." He's rereading the Frontier Saga. We'll find out why. And "Twin Peaks." It's all coming up next on Security Now!.

Leo Laporte: This is Security Now! with Steve Gibson, Episode 614, recorded Tuesday, May 30th, 2017: Vulnerabilities Galore!

It's time for Security Now!, the show where we cover your privacy and security online. We also do a lot of - I think Steve does a lot of teaching, as well. I know a lot of people learn from this show. It's even used in the curriculum at some universities. Steve Gibson is our host, the man of the hour, the security guru at GRC.com. And hello, Steve. It's good to see you.

Steve Gibson: Hey, Leo.

Leo: I'm proud to say good friend.

Steve: Yes. Great to be with you again for Episode 614, as we cruise through Year 12 of this no-end-in-sight podcast.

Leo: Would you like sometimes to do some prognostications of what we'll do 12 years from now when we're doing Security Now!, Episode 1228?

Steve: No. Remember?

Leo: Oh, you're quitting at a thousand.

Steve: I'm going out at three digits. I can't go to four digits. That'll break my whole system. So we're not yet done. We're not there. 666, that'll be, not coincidentally, that's the two-thirds point. When we get to 666, at that point we have 333 remaining. So anyway - besides, you're going to be on a permanent vacation by then anyway, so I don't think...

Leo: Well, I don't know. I mean, I've got bills to pay. I don't know.

Steve: So this week we have, first of all, the title of this week's podcast: Vulnerabilities Galore! There wasn't anything that stuck out because there were just so many just headshakers here. We're going to discuss a new non-email medium for spearphishing; the fact that the Chipotle restaurant chain just can't catch a break; social engineering WannaCry exploits, get this, on Android, which of course can't be infected with WannaCry, but nobody seems to really understand that; video subtitling now able to take over our machines; a serious Android UI design flaw that Google appears to be stubbornly refusing to address, although at the Black Hat conference next month the pressure's going to get turned up.

Linux now has its own version of WannaCry. Another dangerous NSA exploit remaining unpatched and publicly exploitable on WinXP and Server 2000 machines, which of course aren't going to get back patched. We're going to take a look at 1Password's brilliant and perfect, in terms of implementation, new travel mode. Google has announced that - and this one I said, wait, can this be true? Google is extending its ad performance tracking into the offline world. So you just can't escape them.

We've got some follow-up, some miscellany, some fun stuff, and a bunch of closing-the-loop feedback from our terrific listeners. Oh, and concluding with my possibly useful analogy which one of our listeners asked for to explain to a non-computer-savvy person who knows enough about the question of open source and closed source the somewhat confusing value of open versus closed source software. I think I have a useful analogy that I came up with when responding to this person's question. So I think a fun and interesting podcast.

Leo: That's great.

Steve: And if you can get the Picture of the Week on the screen?

Leo: I have it, yes.

Steve: Okay. By the way, I forgot, I meant to mention before we started recording, but what the heck.

Leo: Go ahead, do it now.

Steve: Andy was talking about the Art of Noise, and I just thought I would share that my absolute all-time favorite cut of theirs is called "Moments in Love." And it is just - it is a fabulous piece of music.

Leo: Oh, well, I have to listen to it.

Steve: "Moments in Love" by the Art of Noise is one of my all-time faves.

Leo: Should have known you're an Art of Noise fan.

Steve: So our Picture of the Week is just wonderful. This is the low-tech solution to apparently a somewhat publicly available emergency phone. So this is a wall phone underneath a sign that says "Emergency Telephone" and then "Only 911 can be dialed." And looking at it on the wall, you see sort of this band across the middle of it on the backside, and you think, what the heck? And so the second picture is someone lifting it up and looking at it, and it's this piece of tinfoil which has been wrapped around the keypad with cutouts for the "1" button and the "9" button.

Leo: You could dial, I should point out, 999, 991, 199, but they really want you to dial 911.

Steve: Yeah, they're trying to discourage you from exploring the other buttons.

Leo: Which makes me want to, frankly.

Steve: Yeah. We'd call this a very weak firewall. Of course, if I were involved, I would have opened up the handset because inside there is going to be a circuit board where the little elastomer pads are pressed against, and you could just put insulating Scotch tape over all of the pads except the "1" and the "9" and then put the phone back together.

Leo: Clever.

Steve: And that would kill all the buttons except "1" and "9." But anyway, we have a lower tech solution which is really pretty funny. I mean, this is a sad bit of...

Leo: Is there anything it can't do?

Steve: Right. So, okay. This is just sort of a public service announcement. Security research firms are reporting a shift from email as the target of spearphishing to Twitter and Facebook social media networks because there's a general feeling of community and trust and sort of a lower level of anonymity, the idea being everyone sort of knows you get spam, and anybody can send anything to anyone in email. But there's sort of a greater sense of closure and trust and community within a social media circle. And so some experiments have been done to test people.

And so what's happening is the message is finally getting through not to click on links in email. I mean, it's a mantra. And as we've talked in the past, corporations are educating their employees. Some of them are employing outside firms to run benign probes of their employees from the outside to try to phish them into clicking a link and then sending them to the principal. And so in an experiment that was conducted late last year, an automated program sent links through social media to 819 people. Of those, 275 users opened the links, which is a little bit over one in three, which is a far greater percentage than we're now seeing in email.

Leo: Wow.

Steve: So the point is people's guard is finally beginning to come up. And so what's happening is phishing is becoming less effective. So what do the bad guys do? They go, hmm, where else, how else can we get links in front of people? Oh, let's tweet them. Let's post to their Facebook page. And that's what's happening. So again, as a public service announcement, I want to just put this on our listeners' radar that it isn't anymore the case that only email is a problem. The bad guys have figured out that we're getting wise to them phishing through email, and so they're increasingly beginning to use alternative approaches, specifically Twitter and Facebook and other, I imagine, messaging apps if they can get to you somehow. So not good.

I know you talked about it over the weekend, but I wanted just to cover it for the sake of, well, being topical, and that is that Chipotle, the restaurant chain, did formally announce that

"most," which was an interesting adjective, of its restaurants - is that an adjective, "most" of its restaurants?

Leo: Most.

Steve: I don't think it's an adjective.

Leo: I don't know, what is it?

Steve: It's not an adverb. I don't know, it's one of those things in the twilight zone - were infected with credit card-stealing malware. The reason I'm suspicious about mostness is that apparently it was their central corporate chain processing system. So why wouldn't it have been all of them?

Leo: That would be all of them.

Steve: Yeah. So here's the deal. The credit card details of purchases made from March 24th through April 18th may have been exposed. They said "most restaurants." But again, since this was a breach of their central charge processing system, it's more likely that all branches were vulnerable, and some PR person put a little spin on the press release finally. Now, as is the case with credit card theft, there's no recourse available to consumers.

Back before the Internet, well, not before the Internet, early in the days of the Internet, before I started just booking my own travel, I had a travel agent. And normally I'm just going to Northern California once a year. And her name was Judy Supple, and so I'd give her a phone call every year like around Thanksgiving, and she'd say, "Oh, Steve, same credit card?" And I'd say, "No, I've lost that last one, too, so here's my new number."

The point was that this predated PayPal. Security was worse, and my cards were being compromised constantly because I was out there on the front lines buying things. But we didn't have Amazon to aggregate so many purchases. We didn't have PayPal to aggregate so many purchases. So much as I was trying not to spread my card number around, it got out. And of course I'm eating in restaurants, too, so I never really knew, I was never able to figure out what the leakage point was. But the point is that, and this is the case for credit card, but not for debit card, consumers are not liable for any charges. So the only thing...

Leo: That's true for debit now, too, by the way.

Steve: Oh, it is. Oh, good. Good, good, good. So scrutinize your card statement...

Leo: In the U.S.

Steve: Ah.

Leo: We have international listeners, so in your country check the rules.

Steve: Good.

Leo: Yeah.

Steve: So in the U.S., thank you. What you need to do is, if you purchased from a Chipotle restaurant between the dates of March 24th and April 18th, your credit card information may have escaped. And the problem is the other thing we've seen is that there is normally a multi-month delay before your information starts getting used. They're aggregated; they're chopped up; they're processed; they're put on the dark web; they're bundled. And it's not clear why it takes so long. But experience shows that it does.

So unfortunately, if you did have a credit card transaction at Chipotle between March 24th and April 18th, your credit card information may be out there. And so unfortunately there's no end date except the expiration of that card. Just pay a little extra attention to your credit card statements. And if you find something that you know you didn't purchase, which is what I used to do when I would - and normally the fraud detection stuff picks that up. I was getting - I think they had me on speed dial back in the day because it was like, have you purchased a - I don't remember even now what - there were some really funny things. It's like, uh, no, I'm not in Germany. I've been in California the whole time. Oh, well, maybe that's not you. No, I'm sure it's not me, and please take it off of my card.

So anyway, so it's up to us to defend ourselves. And unfortunately that's the state of the world. And again, there isn't any point at which you're safe until the card that you use naturally expires, which at least changes the CVV code and then makes it much more difficult to use, presuming that the CVV was also captured as part of the purchase. Which is not supposed to be, but sometimes it is.

From our "we should have seen this coming" department, even though WannaCry, that we've been discussing the last two weeks and of course was a worldwide phenomenon, 300,000-plus Windows machines compromised globally, entirely a Windows problem. The recent high-profile worldwide attack has, believe it or not, spawned a bunch of fake Android WannaCry protection apps. Which are completely unrelated in any way. But it's in the news, and there's, you know, Android users who don't listen to this podcast are like, oh, maybe I need that. I don't want to get that horrible thing I've been reading on the news and people are talking about on my phone. So they're being downloaded.

The good news is they're not particularly malicious. Predominantly they're adware leveraging the current WannaCry hysteria to get themselves installed on people's Android devices, where they then attempt to get the user to install additional unnecessary junk ware. So anyway, just a heads-up. It probably wouldn't catch any of our listeners, but our listeners have friends. And you don't need anti-WannaCry protection for your Android phone. There are some things you need for your Android phone we'll get to here in a second, but not WannaCry.

Under the topic of "here's one no one saw coming" is the Attack of the Subtitles. Check Point Research, that we talk about every so often because they're a great security research team, did a blog posting a few days ago titled "Hacked in Translation: From Subtitles to Complete Takeover." And once again, it's been a theme for this podcast for the last couple years, once we saw this sort of - we were able to generalize a set of problems that kept recurring. And that is how difficult it is for interpreters not to have mistakes in them. Another interpreter is the processing of subtitle files for videos.

"Researchers at Check Point have discovered a new attack vector which threatens millions of users worldwide: attack by subtitle. By crafting malicious subtitle files, which are then downloaded by a victim's media player, attackers can take complete control over any type of device via vulnerabilities found in many popular streaming platforms." And they looked closely at some of the biggest ones. The biggest of all is VLC, that I use, and I imagine lots of our savvy Windows and Linux and Mac people, it's multiplatform, stands for VideoLAN, VLC. There's also Kodi (XBMC), Popcorn Time, and Stremio, or Strem.io.

Check Point writes in their coverage of this that there are approximately 200 million video players and streamers that currently run the vulnerable software, making this one of the most widespread, easily accessed, and zero-resistance vulnerabilities reported in recent years. They've tested and found multiple serious subtitle-parsing vulnerabilities in

every video player they've examined, starting with the four most prominent players that I mentioned. They said that they have reason to believe similar vulnerabilities exist in other media players. They did follow responsible disclosure practices and reported all vulnerabilities that they found and demonstrated exploits to the developers of the vulnerable media players. Some of the issues were already fixed at the time that they reported them, while others were then investigated and fixed.

So VLAN is now at 2.2.6, and I just checked my version. I was at 2.2.4. So I and 170 million others are using VLAN. So if anyone's got it, you just want to make sure you're up to 2.2.6. So it's been updated. Essentially all four of them have. And I have the links to the latest versions in the show notes, if anyone is interested, or just I'm sure you can launch the player and have it check for updates and update yourself. But this is an emerging threat vector. The problem is we're protecting ourselves by knowing this, but most of the world won't know about this. So hopefully these various things have auto update which is engaged and enabled and will work to protect their users. I don't remember whether...

Leo: Yeah, it auto updates, yeah.

Steve: VLAN does? Okay, good.

Leo: VLC, yeah.

Steve: VLC. I tried bringing it up and looked at the About. I didn't check to see if it would update itself. And maybe it's just taking some time to get around.

Leo: It does it all the time. What it'll say is "I have an update. Do you want to update?" And you say yes.

Steve: Good.

Leo: Please. Now. But, yeah, you might even want to manually do it if it doesn't offer you that.

Steve: Yeah. I would say it's definitely worth doing.

Leo: It's interesting that all these players have this problem. So it must be a common DLL or something; right?

Steve: In the case of VLC, it's a plugin. And I can't remember what the name of it was.

Leo: Is it a codec or...

Steve: Well, they may have used common code. So you're right. What's probably the case is there was an open source, publicly available implementation, and they all grabbed that off of GitHub and then repackaged it for their own needs and with hooks for their own particular player. And so as a consequence there is some commonality there. Again, it's an interpreter, and it's just not hard to, as we keep seeing, to find mistakes in interpreters because, as is always the case, as soon as they get it working they're like, okay, it works. But you never think of it as an entry point, as a threat vector foothold.

So the team that we've talked about before of researchers, both from University of California at Santa Barbara and the Georgia Institute of Technology, are back with a new and worrisome finding. They call it Cloak & Dagger, and they've got a site, hyphenated, cloak-and-dagger.org. Not dot com, dot org. Cloak-and-dagger, with hyphens, dot org. It's a very nice page that will explain what's going on. And they're the people who will be showing this at this coming - in two months at the July 2017 Black Hat in Vegas. They just won a Distinguished Practical Paper Award from the proceedings of the IEEE Symposium on Security and Privacy in San Jose earlier this month.

So, but what they found is worrisome. From their disclosure, they said: "Cloak & Dagger is a new class of potential attacks affecting Android devices." And this is where I mentioned at the top of the show that they're having problems getting Google to understand, apparently, what's going on, and we'll walk through in a second the responsible disclosure back and forth. If nothing else, these guys are really patient. It's too bad Google's not putting themselves on their own Project Zero timeout because they'd be in trouble if they were.

They wrote: "These attacks" - get this - "allow a malicious app to completely control the UI feedback loop and take over the device without giving the user a chance to notice the malicious activity. These attacks only require two permissions that, in case the app is installed from the Play Store, the user does not need to explicitly grant and for which they are not even notified." They wrote: "Our user study indicates these attacks are practical. These attacks affect all recent versions of Android - including the latest version, Android 7.1.2 - and they are yet to be fixed."

So under "Main Takeaways" they said: "We uncover a series of vulnerabilities and design shortcomings affecting the Android UI. These attacks abuse one or both of the" - and there's two, there's a SYSTEM_ALERT_WINDOW, which is also known as "draw on top," meaning it's a privilege that allows something to pop up over the existing application. And so what this does it is allows the underlying UI to be obscured, and these guys are able to get up to all kinds of mischief with that power. And the second one is the BIND_ACCESSIBILITY_SERVICE, known as "a11y."

So they wrote: "If the malicious app is installed from the Play Store, the user is not notified about the permissions, and they do not need to explicitly grant them for the attacks to succeed. In fact, in this scenario the 'draw on top' privilege is automatically granted, and this permission is enough to lure the user into unknowingly enable [that second permission, the] a11y through clickjacking." They write: "The possible attacks include advanced clickjacking, unconstrained keystroke recording, stealthy phishing, the silent installation of a god mode app with all permissions enabled, and silent phone unlocking and arbitrary actions while keeping the screen off." And then they have a full list of things below.

They wrote: "These attacks are practical. We performed a user study with 20 human subjects, and no user understood what had happened to them. Most of the attacks are due to design issues." And this is where they're having a problem with Google because it's like, that's not a bug, that's a feature. Okay. "And they are thus challenging to

prevent," they write. "In fact, one may say that some of these functionality work 'as intended.' Nonetheless, this work shows that this functionality can be abused." And, as they wrote: "To date, all these attacks are still practical."

So I'm going to skip over some enumeration of the nature of the attacks because they just expand on what I already wrote. But under their responsible disclosure timeline, which they were careful to share because it failed with Google, and they wanted the world to understand, look, we've been trying for more than a year. They started last August and they're going to go public with this in July. So it will have been a year that they've been trying to get Google's Android security team to understand this.

On August 22nd they opened several issues on the bug tracker for the Android Open Source Project. On the 31st of August, so nine days later, the Android security team set the severity as moderate for one of the bugs, that is, the "draw on top" with unconstrained keystroke recording. Then a month later, on September 30th, the Android security team marked one of the reported bugs, the other one, the a11y, where you could do unrestrained keyboard recording plus leaking a security PIN plus unlocking the device while keeping the screen off as "works as intended." Yes. That's a feature, not a bug.

Then two weeks later, on October 10th, these researchers followed up, pointing out that the accessibility services documentation states that this a11y should not be able to access passwords, and that a11y should not be able to unlock the device and perform arbitrary actions while keeping the screen off. Yet it does. Three days later, that is, October 13th, Android security team states that: "The password will be repeated if the user explicitly turns that on in settings under Settings > Accessibility > Speak Passwords. The option is off by default. If the user explicitly enables this feature, it is not a security vulnerability." So again they're saying, well, we don't think that's a vulnerability.

So the next day these researchers follow up by clarifying that their attack works even without this feature. And they wrote: "In fact, our report does not even mention it." Four days later, October 18th, the Android security team marks this bug as "high severity." Okay, it sounds good. Making progress. November 28th, a little over a month later, Android security team downgrades this bug as "not a security issue" and marks it as "Won't Fix (Intended Behavior)" because "limiting those services would render the device unusable."

Not quite a month later, December 19th, a draft of their IEEE paper that I refer to, which won their award, is shared with Adrian Ludwig, the director of Android Security. Months go by. March 15th of this year they follow up again, pointing out that the documentation states otherwise. They say: "We also follow up on all the other bugs we opened, asking for a status update." Now we're up to the beginning of this month, May 3rd. "Again, nothing comes back. We follow up a second time, asking for a status update for the bugs we reported. The next day, on May 4th, Android security team keeps our a11y findings as 'won't fix,' but they state they will update the documentation." That's right, so don't fix the problem, change the documentation so that it agrees with the problem. They wrote: "We did not receive updates about any of the other bugs we reported."

Four days later, May 8th: "We have a telephone conversation with the antimalware and a11y Google teams, during which we thoroughly discussed all the details of our research." On the 19th of May, the a11y team confirms the a11y-related issues we reported as "won't fix." Okay. Three days later, May 22nd, and this is the last entry: "This website and our research paper at IEEE S&P are made public." So where we are today, every attack discussed in this work is practical, even with the latest version of Android.

So they then, in order to help better explain this, they have in their presentation some Q&A. Let's see: How can an app stealthily obtain two permissions? They explain that. Why is the "draw on top" permission automatically granted? And they say, for example: "The behavior appears to be a deliberate decision by Google and not an oversight. To the best of our understanding, Google's rationale behind this decision is that an explicit security prompt would interfere too much with the user experience, especially because it is requested by apps used by hundreds of millions of users. For example, Facebook requires this permission to implement the 'Android chat heads,' one of its very popular features."

Leo: Not with me. I could easily give that one up.

Steve: Ugh. So, yeah, so in their Q&A: Are these permissions shown to the user after the app is installed? No. How difficult is it to get the app with those two permissions approved? Turns out they're often approved by default. They said: "A quick experiment shows that it is trivial to get such an app accepted on Google Play Store." And this is a year after they told Google, or nearly a year. They said: "In particular, we submitted an app requiring both of these permissions and containing a non-obfuscated functionality to download and execute arbitrary code, attempting to simulate a clearly malicious behavior. This app got approved after just a few hours, and it is still available on the Google Play Store."

Then they say in their Q&A: What do you recommend to users? So they say: "We recommend users to check which applications have access to the 'draw on top' and the 'a11y' permissions. Unfortunately, both permissions are considered 'special' and, for this reason, certain versions of Android may show 'no permission required' even if, in fact, the app has access to both the permissions required." So even the UI in this case is defective in that, if somebody, if a responsible Android user wants to take responsibility, you can't find out.

They said in their paper: "We provide instructions for several versions of Android." Oh, and I should mention that this is just the top-level coverage. In the link that I have in the show notes is a PDF for their whole paper, where they go into as much detail as anyone could want. And they said: "This work shows that the user should not consider their device's UI as a trusted source" - this is so sad. I'll start again. "This work shows that the user should not consider their device's UI," that is, their Android device's UI, "as a trusted source of information. Thus, from a conceptual point of view, the user should rely on other means than the device's UI itself." Okay, what?

Leo: Geez, what? Yeah, what's the other choice?

Steve: I know, I know. "An alternative solution," they wrote, "is to use" - get this - "command line tools" - oh, yeah, that'll go over big - "such as adb, or to determine the permissions requested by each app through the Play Store website. For example, to check the permissions" - get this - "of the official LastPass app, which requires both permissions, you can go to its Play Store page, scroll down, and click View Details under Permissions. The 'draw on top' permission will appear under the Others > Draw over other apps label, while the 'a11y' will appear under Others > Bind to an accessibility service."

So none of this is easy. Our non-tech-savvy people aren't going to get this, and they're

going to be caught out. And now this is all public. I mean, this is no, you know, they kept this quiet until a week and a half ago. Now it's all public. Google is accepting apps that do this. And they'll be showing this at Black Hat, and Google is apparently saying, no, this is what we want.

So finally, in the Q&A: Why are these bugs not fixed yet? And they wrote: "Some of the issues uncovered by this work are design-related issues, not simple bugs, and thus it necessarily takes more time to fix them. On the other hand, Google is saying 'won't fix.' Moreover, these are not classic," they wrote, "low-level issues, but UI-related problems. These issues may be more challenging to fix since changes may introduce backward compatibility issues." Well, for example, apps like LastPass are...

Leo: And Facebook, yeah.

Steve: Exactly, depending upon this behavior which now these guys have shown can be hugely abused. And there's very little control over, like none right now, over which apps are able to acquire those permissions when they're submitted to the Play Store.

Leo: I'm going to guess this is an accessibility issue, and that if they took these features out, that would reduce accessibility.

Steve: Certainly the second one is, although...

Leo: Yeah, "draw on top" does, for sure, yeah.

Steve: Well, no. The other one is explicitly an accessibility issue. But the "draw on top" looks like it's an ability that allows pop-ups, like from Facebook, to pop up in order to prompt you for, like...

Leo: That's a big vulnerability because you can clickjack; right?

Steve: Exactly. So they said: "These issues may be more challenging to fix since changes may introduce backward-compatibility issues and changes that are visible to end users," meaning a loss of existing functionality. They said: "Finally, some of the bugs were marked as 'won't fix,' and thus they will not be fixed."

And then, finally, the last question: Are these attacks practical? And they said: "We believe so. We performed a user study with 20 human subjects, and none detected to be under attack. We report more details in our paper." So let's just hope that the attention this is going to get will force some action from Google. This sounds like a powerful feature that Android has to allow one app to present something to a user on top of another. But these people demonstrated that this feature can be abused. And, I mean, what we're going to now see, and I imagine we'll be discussing this in the coming months, is a bunch of actual exploits of this which will probably drive Google to take some sort of measure. I don't know what it'll be.

Leo: You'd have to - a website couldn't do this. You'd have to install a malicious app; right?

Steve: I don't want to...

Leo: You'd have to give the browser those two permissions that draw.

Steve: As I understand, though, in Android browser apps have a lot of functionality available to them. So I'm reluctant to say that you couldn't just surf to somewhere and have that happen.

Leo: See, that would be a real problem. If it's just an app, that's a little bit of a lesser problem. You just tell people, you know, only install mainstream apps.

Steve: Yeah, the problem is, I mean, maybe - we were just talking about the need to audit features and permissions in apps in Android. But what this says is that, even though this is something that apps can ask for, those are not features which are necessarily surfaced to any UI.

Leo: Right.

Steve: And also there's this problem, sort of, of the chicken and egg. That is, what they're saying is you can no longer trust what the UI shows you, meaning that a malicious app, once it gets in, could cover up your screen with whatever it wants in order to get up to mischief.

Leo: LastPass, for instance, requires you go into accessibility settings, at least for that accessibility-based one, and give it explicit permission. In other words, it's not a pop-up in the traditional permissions pop-up. You have to actually go into - if you want "draw on top," you actually have to go into accessibility permissions and say, "I want this capability."

Steve: And you know if you did turn it on, because what they're saying is...

Leo: Well, I always turn it on.

Steve: ...it was turned - okay. So they're saying that the apps, when downloaded, can ask for and implicitly have both of those turned on, and the user is never asked.

Leo: Yeah, see I'm not sure because I do know that you don't get this "draw on top" stuff without going into accessibility. And it does explicitly ask you, LastPass does,

okay, here's what you're going to have to do to give us - I think this is the autofill capability that it wants. And it says you have to do this if you want this.

Steve: What they're saying, for example, is that people who downloaded the Facebook app were never...

Leo: No, no. You aren't in Facebook. Yeah, that's right. But that's the chat heads. That's "draw on top."

Steve: The "draw on top," right.

Leo: I don't know if you need accessibility permissions in Facebook or not to get full functionality. Yeah, Facebook doesn't ask you for a whole lot of interactivity because they can't.

Steve: Exactly. We can understand why they wouldn't. And it's just like, what? I don't know how to answer this question.

Leo: Yeah, right.

Steve: Yeah, so maybe what the least we need is an auditing tool to carefully look at which, you know, to understand the risk and only allow trusted apps like the Facebook app, like LastPass, things where you could say these apps have a reason for needing this, rather than just everybody who asks for it gets it, which is the way it is now, and then can get up to some mischief.

So believe it or not, following on the heels of WannaCry is SambaCry.

Leo: It was just a matter of time.

Steve: And, my friend, it is really bad.

Leo: SambaCry.

Steve: SambaCry. The Hacker News headline was "7-Year-Old Samba Flaw Lets Hackers Access Thousands" - and we're talking more than 24,000 - "Linux PCs Remotely." Bleeping Computer's headline was - oh, wait, they had a bigger number. I might be thinking - the 24,000 might have been something else because BleepingComputer wrote: "Over 104,000 Samba installations vulnerable to remote takeover attacks."

Okay. So what's going on? Samba has a problem. Their own security advisory, published last Wednesday, wrote: "All versions of Samba from 3.5.0 on." Okay, 3.5.0 was released since March 1st, 2010, so seven years ago. So for more than seven years Samba is

vulnerable to a remote code execution vulnerability - with some conditions, but unfortunately they're not very restrictive - "which allows a malicious client to upload a shared library to a writable share, then cause the server to load and execute it." So that means, if you have a publicly exposed writeable share, what this means is that somebody finds it. And of course it's easy to find because you scan for port 445. And, oh, look. If it's not Windows, it's Linux.

So if it's Windows, you use WannaCry. If it's Linux, you use SambaCry. Since the SMBD, that's the Server Message Block Daemon, or the Samba Daemon, typically runs at root, the uploaded and executed module runs in the SMBD context with the same full root privilege. So what this amounts to is the ability for anyone that can write to a Samba share the ability to install, essentially upload their own loadable module into that directory and then make the Samba server run it as root. A Shodan query of port:445 and then bang, meaning not, !os:windows shows approximately one million non-Windows hosts that have TCP port 445 open to the Internet, more than half of which exist in the UAE, the United Arab Emirates, that's 36%...

Leo: Half of them?

Steve: ...more than half of those.

Leo: That's weird.

Steve: And the U.S. has 16%. However, it isn't clear how many of them are running vulnerable versions. On the other hand, for the last seven years they've all been vulnerable. The vulnerability be exploited with a single line of code. A malicious client can upload and cause the SMB server to execute, as I mentioned, a shared library from a writeable share. And exploit modules are already available in Metasploit to exploit this issue on Linux for ARM, x86, and x86 64-bit architectures. Now, this bug has nothing to do with EternalBlue. Whereas Eternal Blue, of course, on Windows was a buffer overflow exploit, the Linux SMB vulnerability leverages an arbitrary shared library load and execute.

So they're different beasts, but this is, I mean, I would argue even worse and more powerful. And note it is totally wormable. This requires no user intervention or interaction. If your system gets scanned, this can be uploaded to it and executed. And if it's a worm, then it starts scanning and takes off. So the non-vulnerable versions have just been released, which is why this went public. So 4.4.14 is not vulnerable; 4.5.10, not vulnerable; and 4.6.4, not vulnerable. But these have only been released in the last week. And as the security firm F5 put it, I liked their simple equation: network + remote code execution + root = drop what you're doing and patch. Meaning, really.

Okay. So patches exist. Debian, Ubuntu, CentOS and Red Hat have all taken immediate action to protect their users and have released patches to their supported versions. And there is a security workaround available for people who cannot patch. If you add - and I've got all of the details in the notes here. If you add the parameter "nt pipe support = no" to the global section of the smb.conf [config] file and restart the SMB daemon, that'll shut this vulnerability down, if for some reason you are unable to patch. But everybody who references this notes that it may disable some functionality for Windows clients.

On the other hand, I mean, obviously immediately updating your version, or maybe just

taking it offline if you don't really need it is necessary. And of course NAS vendors, Network Attached Storage vendors, have work to do because different brands and models that use Samba for filesharing - a lot if not all of them provide this functionality - will have to issue firmware updates to patch this flaw. If the firmware updates for these appliances take the same time as they usually do, we'll have this bug around for quite some time. And we know that these things, you know, there are lots of servers in the closet. And Shodan says the attack surface on the public Internet is huge.

So this has just happened. I imagine that next week we'll be talking about the SambaCry worm having gone, unfortunately, "viral" is the only term we have for it. But, yeah, being actually active and probably fighting over Linux machines that are available publicly. So, yikes. You don't want to have a publicly writeable Samba share. And maybe you think, oh, you know, it's safe because people can deposit things publicly, but we'll scrutinize them before we do anything with them. No. People can deposit an executable module and run it without any permission as root, if you have something that's from seven years old and haven't just fixed it.

So certainly all of our listeners who - and remember, the public exposure is only one aspect. The Intranet side is equally bad because, just as WannaCry spread publicly, but also looked for internally, which is of course what brought down the whole U.K. medical system was the Intranet got scanned and taken down by that cryptomalware. So, I mean, maybe the same thing will happen. We'll be talking about it a week from now. Looks really bad.

And here, I knew that there was 24,000 of something somewhere. The NSA's Windows EsteemAudit exploit, which uses RDP - that's the Remote Desktop Protocol. That's the whole, you know, it's used for getting the desktop remotely. It is still unpatched, not surprisingly, for Windows XP and Server 2003. So here we have a situation where this is another bad problem which isn't patched because that operating system of that era, the XP/Server 2003, are no longer being maintained. Because the WannaCry was such a problem, as we know, Microsoft issued an emergency back patch for those.

Well, what Shodan shows is 24,734 presently vulnerable WinXP and Server 2003 systems on the public Internet. And they're not going to get patched, I mean ever, unless Microsoft decides, gee, maybe we should fix this one, too. So EsteemAudit is another very dangerous NSA-developed Windows hacking tool which was leaked as part of what the Shadow Brokers were doing. It targets the remote desktop protocol, which runs a service at the famous port 3389 on Microsoft systems. And in this case it's Windows XP and Server 2003 that have always been vulnerable and will probably always be vulnerable. It can be weaponized into wormable malware. And again, I think we're going to see that in the same way that WannaCry ransomware was, as we all experienced, turned into a worm and generated lots of headlines.

So the good news is, for people who are staying with, for whatever reason, WinXP and Server 2003 and have a profile that would make them exposed, I mean, you're crazy to have port 3389 exposed to the public Internet. But, I mean, there is a logon username and password. I would never trust that, but some people are going to. There is an unofficial patch which EnSilo Security has published. I have the link to the patch in the show notes.

They write: "The patch for WinXP and Server 2003 supports silent installation and does not require a reboot. This helps users avoid the required downtime typically associated with patch installations. Upon patching, any attempt to use an EsteemAudit exploit to infect a patched machine will fail." So again, Microsoft hasn't fixed it, probably won't. But if any of our listeners or firms have XP and Server 2003 with 3389 exposed, I would say

either - certainly publicly, but even on the Intranet, I mean, if you don't need it, then turn it off. If you do need it, I would argue the only recourse you have at the moment is to use this unofficial patch which I have every reason to trust. I mean, EnSilo is a well-known, trusted firm, and they've done this for all the right reasons. So I would say take action.

Leo: Steve Gibson, continuing on. He's fully caffeinated now.

Steve: I am.

Leo: You're like that inflatable pilot in "Airplane!." We've just pumped you back up, and you're fully inflated.

Steve: Although they're not a sponsor on this podcast, I did immediately purchase the Oars and Alps...

Leo: Did you? You're going to like it.

Steve: ...face charcoal. It just looks too freaky. I have to see what that's about.

Leo: It actually is great. It really - my skin is really soft. Well, that's another ad for another time.

Steve: And I used the TWiT code, and I just did want to mention that they support both PayPal and Amazon Pay.

Leo: Oh, nice. I love it. I love that.

Steve: Along the lines of not spreading your credit card any further than you absolutely must. I always look for that option.

Leo: Yeah, me, too, yeah.

Steve: So bravo. Speaking of bravo, I have a new acronym, TDF. Stands for Traveler Data Frisking.

Leo: I'm going to be going through some of that, some intrusive travel data frisking, I think, maybe.

Steve: I know. Being frisked for your data is something that seems to be on the rise. And so many of our listeners have asked, over the last couple weeks, since it was

announced by AgileBits, what do I think about 1Password's Travel Mode? And my answer is they did it exactly right.

Leo: Oh, that's good to know. That's great.

Steve: Yes, 100%. Your various secrets are - so the way this works is you first partition them into vaults. And the vaults can be individually flagged as "Safe for Travel" or not. So as you approach customs or a particularly officious-looking TSA agent, you log into your 1Password.com account and activate Travel Mode, whereupon all of the non-safe for travel vaults are completely removed. They are wiped and removed from your connected devices. They're not obscured or hidden. They are gone. So then you present your device, and if they need to get a password from you, you go, okay, yeah, here's what I've got. Whatever you need to do not to be harassed. You travel through customs smiling and maybe trying not to bow too deep. And if you are data frisked, the duly anointed officials will see only those secrets you previously decided to allow them to see. Nothing more to see here. Move along.

Once you're safely through inspection, you simply log back into your 1Password.com account, turn off travel mode. Maybe you want to wait till you get to the hotel or whatever, depending. Maybe, I mean, and maybe, in fact, you just - if these are things you don't need while you're traveling, then it's also just better for security to not have them on your devices, not only to deal with the officious-looking TSA agents, but just in general not to have them while you're traveling.

But if you need them, the point is you turn off Travel Mode, and all of those vaults are immediately repopulated on your devices and reappear, just as they were before. So I dug in. I read Rick Fillion's explanation of this. He posted on AgileBits' blog under "Introducing Travel Mode: Protect Your Data When Crossing Borders." And I was very impressed with everything I saw. So double thumbs up. These guys did it exactly right. And it looks to me like a great feature. So now we need everybody else to copy them.

Leo: Yeah. LastPass.

Steve: Hello. Listening? Yeah. Okay. So this one I was immediately put in mind of Hudson from the second "Alien" movie, "Aliens," when he said, "Stop your grinnin' and drop your linen." And I have to say I was put in mind of it because I rewatched all of the Alien movies in preparation for seeing "Covenant," which I did see on opening day. We'll be talking about that in a minute - without any spoilers, of course - in our Miscellany. But the reason is this was just a headshaker. Google now knows, believe it or not, when its users go to the physical store and buy stuff. Not online, but brick and mortar.

The Washington Post, as we well know, those of us who've been following politics, has been very busy the last few weeks. In this case, they had some interesting coverage from Google. They wrote: "Google has begun using billions of credit card transaction records to prove that its online ads are prompting people to make purchases - even when they happen offline in brick-and-mortar stores, the company revealed" a week ago, which was a week ago Tuesday. "The advance allows Google to determine how many sales have been generated by digital ad campaigns, a goal that industry insiders have long described as 'the holy grail' of online advertising. But the announcement also renewed longstanding privacy complaints about how the company uses personal information.

"To power its multibillion-dollar advertising juggernaut," writes the Washington Post, "Google already analyzes users' web browsing, search history, and geographic locations, using data from popular Google-owned apps like YouTube, Gmail, Google Maps, and Google Play Store. All that information is aggregated and tied to the real identities of users when they log into Google's services.

"Now, the new credit card data enables Google to connect these digital trails to the real-world [consequences, essentially the] purchase records in a far more extensive way than was possible before. And in doing so, Google is again treading in territory that consumers may consider a little too intimate and potentially sensitive. Privacy advocates said few people understand that their purchases are being analyzed in this way and might feel uneasy, despite assurances from Google that it has taken steps to protect the personal information of its users." And again, I'm not saying anyone should be concerned. I'm just saying we should have the knowledge that this is going on. I just find it amazing and interesting.

"Google also declined to detail how the new system works exactly or what companies are analyzing records of credit and debit cards on Google's behalf," because they subcontracted this. Google, the Washington Post reports, which saw just shy of $80 billion, "$79 billion in revenue last year, said it would not handle the records directly, but that its undisclosed partner companies had access to 70 percent of all transactions for credit and debit cards in the United States." I imagine they're trying to get that last 30 at the moment.

"Marc Rotenberg, the executive director for the Electronic Privacy Information Center, was quoted in The Washington Post's story, saying: 'What's really fascinating to me is that'" - and not surprising, actually - "'is that, as companies become increasingly intrusive in terms of their data collection, they also become more secretive.'" Yeah, no surprise. "He has urged government regulators and Congress to demand answers about how Google and other technology companies are collecting and using data from their users."

Now, Google did say that it took pains to protect its users' privacy. They said: "While we developed the concept for this product years ago, it required years of effort to develop a solution that could meet our stringent user privacy requirements." Okay. So that all sounds good. They said: "To accomplish this" - and I'm not sure about this part - "we developed a new, custom encryption technology" - what? - "that ensures users' data remains private, secure, and anonymous."

Okay. So this was probably written for a non-techie audience. I would imagine what they actually used is well-known proven security primitives and designed some system that provided required anonymity and somehow kept the partners they're working with from knowing anything that they didn't need to in order to get this done. So it would be interesting to find out a little more about this. Maybe we will in the future. But I just thought it was interesting that there is such a clear monetary incentive behind being able to prove the value of the ads that Google serves, that it's no longer enough to use, oh, look, the user clicked on the ad, or they clicked on the link, or they went to the site.

This has now been extended to the presumption that users were consciously or subconsciously influenced by the presentation of the ad such that later, when they were strolling the aisles in the supermarket, they purchased something that had been advertised. And notice it's not, I mean, in order to tie this to an item, you need an itemized receipt for what was purchased. Maybe only where, but clearly Google would like to know what, as well, in order to tie it to an ad. So, really interesting.

I did want to mention that the FCC's public feedback system for their request for public comments on the proposed rollback of Net Neutrality is once again up and running and open to subjecting itself to spam and attacks and lots of belligerent users who feel strongly about this. So GoFCCYourself.com once again successfully redirects to the submission page buried at the FCC, for any of our listeners who do wish to make themselves and their feelings heard.

Also I got a nice email from Clint Wilson, who is the DigiCert product manager that I've dealt with in the past when I've needed special favors, like I've talked about how I needed to, like, I needed a certificate with a midnight expiration that still understood - that had an SHA-1 signature, yet I needed the companion signature that expired on the normal time later, but that was SHA-256. And so DigiCert was able to provide.

Well, we discussed last week how Stack Exchange used DigiCert to create that amazing certificate. And there were some points that I was confused about, only because I just, in assembling the show notes, I didn't have a chance to dig into this on the fly. I asked you at the time, Leo, to go check to see whether it was an EV certificate, and it wasn't. Which made sense to me because EV certs cannot have wildcards. I mean, I dearly wish they could because in that case I would have EV wildcard certs, which would make my life much easier. But I didn't understand why then the EV root was used to sign a non-EV cert.

Anyway, Clint explained it all to me. I got his full description in the show notes. I won't drag everyone through it. But essentially it boiled down to the fact that, again, demonstrating the kind of attention that they give to this - oh, he did say, I mean, there's lots of detail here which shows how much attention certificates can get from people who really understand them. He said: "Stack Exchange uses OV certificates" - that's the organization validation, not the domain validation - "because they need to use wildcard names in order to support non-SNI clients," so *.something.com, for example, "and relying third-party software. EV certs" - and this is an exception I didn't know about - "are not allowed to include wildcard DNS names except" - get this - "for Tor Hidden Services .onion addresses," which can be wildcard and EV. So wow, Clint, thank you for the clarification.

And he said: "Stack Exchange's main certificate includes multiple wildcard names," as we saw last week, "to further support non-SNI clients," he says, "and to simplify some of the deployment requirements. This is something I'd love to say is unique to DigiCert, but it has become relatively readily available from most commercial CAs in recent years," and he says, "though I believe we [meaning DigiCert] were among the first to offer these, notably to Wikipedia, Facebook, and other similar organizations."

And then, finally, this confusion about why it was an EV signature. He goes into some detail here. Essentially they have two roots. They have the DigiCert SHA-2 High Assurance Server CA, and the High Assurance EV Root CA. It turns out that really older software has almost vanishingly better support for the EV cert. And he said down in the hundredths of a percent better, but nonetheless a little bit better. And so they deliberately chose to sign the Stack Exchange cert, the single Stack Exchange mega wildcard crazy certificate with all of the additional domains with the absolutely highest coverage root, just so that they would have the least amount of problems. So Clint, thank you for providing the clarification. Much appreciated. And it answered some mysteries.

Okay. A bit of miscellany. First I got a kick out of this. Leo, you were talking - this was sort of a follow-up from your talking about the growing popularity of the Security Now! podcast last week. Scott Foger, I guess is how you pronounce his name, his Twitter

handle is @fogrito, so maybe it's Foger. He said: "My eight-year-old son..."

Leo: You're getting them young now.

Steve: Eight years old - "...is listening to Security Now! podcast while swinging at the playground."

Leo: Oh, wow.

Steve: He said: "@SGgrc is his favorite. Got to start them young."

Leo: Well, get him a credit card so we can follow all his transactions from now on.

Steve: I did see "Alien Covenant" on its opening day. And, you know, I like the Alien series a lot, the whole franchise. I rewatched "Prometheus" just because it's the sequel to "Prometheus," and both of those predate, in terms of the actual Alien franchise timeline, the first movie, that of course was where we were introduced to the concept. And then James Cameron's sequel, "Aliens," was arguably the best of the bunch, still just outstanding. So I was not wowed by it. But it was good. So there's that.

I did want to mention "Twin Peaks," Leo. Just last Sunday, day before yesterday night, was the second week. And I don't know if they're going to continue doing two hours per week, but the first week, two weeks ago, was two hours, and then two days ago was another two. And it is David Lynch's - like David Lynch on overdrive. It may be a little too weird for most people. I mean, "Twin Peaks" was weird already. And in fact after the first two hours I tweeted my favorite line from that first installment, which was the Chief of Police, who is the same guy we originally had in the original "Twin Peaks" series. over the phone to the Log Lady, saying, "Once again, I find myself in complete agreement with your log." Which is really not a statement.

Leo: I don't even - I do not get the appeal of this. But I'll grant it to you, yeah.

Steve: Anyway, it's quite bizarre. But, yeah. Let's see. Where are we? So I'm going to skip this one just because we don't have enough time. Oh. I keep promising that I'm going to stop talking about the Frontier Saga, but I feel like I need to do our listeners justice. I am flooded by people whose life has been disrupted. When I look at the comments in the mailbag, and it's "Darn you Gibson," and "Oh, my god, what happened to my life?" and such things.

Jason White tweeted: "Thank you." Well, that's nice. He's not upset with me. He said: "The Frontier Saga is everything I could hope for in a sci-fi story - outstanding action, great characters, intriguing technology. Oh and can I say again, action. The primary battle in Book 3," he wrote, "is absolutely amazing. I have an hour and a half commute several times a week, and this series is second on my list after Security Now!. Thanks for the recommendation."

Steve Doyle tweeted: "Hey Steve, I just had to say thanks. I absolutely love the Frontier

Saga. I've started listening a week or so after you mentioned them and am about to finish Book 10. They have consumed all my free listening time." He says: "Pro tip for everyone who doesn't want to spend the money to purchase all the audiobooks. I've been able to find them at my county library."

And I did respond to him, and so I'll just add, I am currently rereading the series because I've read them all, all 19. And then I thought, okay. While waiting for Book 20 - which, by the way came out two nights ago. So Book 20, the fifth book of the second series, is now available. Now I'm just going back through because the guy's style is so good, the characterization and the development of relationships, now that I sort of know the future, I'm rereading them to know who Tug and Dumar really are, to know about Jalea, to watch Josh and Loki develop, and to watch Cameron and Jessica grow to understand who Nathan is becoming.

So I know I sound like some sort of a cult worshipper here. But, I mean, these are just - I'm coming to the position of feeling that it's the best stuff I've ever read in my life. That is, this series, I mean, so many of the other things we've talked about are really good. But this stands up there as, I think, as good as any of the other stuff. Different. A little more of a space opera. But, boy, I mean, the feedback from our listeners is 100%.

And speaking of 100% feedback, Jeff from Columbus, Ohio, said: "I <3 SpinRite." And I thought, what the heck? And I've just - I'm a little slow on the uptake. I guess now I get it. That's a heart on its side, <3, which is probably a common...

Leo: Yeah. Been around a few years, yeah.

Steve: ...icon. But I never really spent the time to parse it. Anyway, so "I <3 SpinRite." He said: "Just a quick note to say how much I love SpinRite. I bought it after listening to my first episode of Security Now!, something like a decade ago." Yes, that's about when it would be. "Thankfully, I have never had to use your product. But," he said, "there is no comparison," and presumably he means to anything else, "for keeping things running at peak performance. Recently a 2TB backup drive was running slow at the start of each backup, transferring just bytes for several minutes before eventually returning to normal speed. I knew SpinRite would fix it," he wrote. "Sure enough, I just backed up 40GB in half the time it took me to type this note. Thanks again for all you do. P.S.: If you need a beta tester, I'm your guy."

And so when I get back - when I get SQRL wrapped - which is what I spent the whole weekend on. I was up till 2:30 in the morning Saturday night/Sunday morning on a real roll, working on SQRL. I had actually intended to take some time off this weekend, but I was just getting too much done with SQRL, so I didn't. Once that's done, I will immediately return to SpinRite 6.1. And I will be developing it as I was before and as I have SQRL, in public view. It's a little bit off the beaten path, that is, it's in the GRC newsgroups. But we have this amazing group that are assembled there. And it's so useful to me to have an interactive environment where I can put code up, and people can experiment with it.

Just last week somebody had installed an off-brand firewall add-on for Windows 10 which only affected Edge, not Chrome or IE. But as a consequence of the fact that we're testing this as we're going, we quickly found - we realized what the problem was and were able to come up with a resolution. So the result is that when it finally emerges, it's already well tested and bulletproof. And the point here is that, once SpinRite 6.1 is happening, there will be all of this ongoing, sort of out of sight, but available to people who care

enough. And as I have said, as I have something which is fully functional, but doesn't have all of the polish on it, I will certainly make it more widely available to all of our listeners who want to play with something from the start.

Leo: All right. Let's continue on.

Steve: Okay. So I ran across a term, I'm sure I had seen it before, but it just caught me off guard. And I just love this term. So somebody sent to me, and I'm not sure why I don't have his name here, but the subject was "Not all memory leaks are important." So it must have been in the mailbag.

Leo: Oh, I know what that is, an article. It was in, like, the Hacker News or something, yeah.

Steve: Well, this one was anonymous, and so that's why I didn't have his name. He said: "Many years ago I worked on an SS7 surveillance system." And of course we know SS7 is the international phone system glue. "Many years ago I worked on an SS7 surveillance system," he says, "an equipment alarm monitoring app. After many of our Unix machines got updated, they started rebooting every 60 hours or so. It turned out that there was a bug that caused a memory leak on each unsuccessful connection attempt. We didn't know that at the time," he says, in parens, "(it was a driver issue). So we turned on logging in the app in the hopes of finding out what was wrong." Then he says: "Needless to say, the memory leak disappeared." And here's the term I just love. He says, parens, "(a heisenbug)."

Leo: Yes. When observed, it disappears.

Steve: When observed, yes. A heisenbug. Turn logging on, oh, now the problem doesn't happen. Darn. He says: "In the spirit of 'not all memory leaks are important,' we just left the logging on and piped it into dev/null. Problem mitigated."

Leo: Fixed it.

Steve: Yup.

Leo: I think the one I read they just doubled the memory, and then it just didn't matter. It just didn't matter.

Steve: Right. Well, and of course we talked in the last couple weeks about the memory leak on the missile, where as long as the leak wouldn't bring the...

Leo: That's the one, right, right.

Steve: ...software down before the missile reached its target, eh.

Leo: That was it. Yeah, yeah, yeah, yeah.

Steve: Not a problem. And how much cheaper it was to double the memory rather than bring the subcontractors back and try to have them scrounge around and fix the leak.

Leo: Simple.

Steve: So I don't know what is happening, but I found in the mailbag three different people who were like, oh, my god, robocalls.

Leo: Oh, yeah. It's out of control, yeah.

Steve: There's just been an escalation of it.

Leo: Yeah, it's just out of control.

Steve: So Jerome Shidel said: "Steve, please help. Last time I was at my mom's house, she is being bombarded by scammers and telemarketers with upwards of 20 to 30 a day calls coming from local" - well, yes, spoofed local - "and all over the country."

Leo: Yeah. That's the new thing is they use your area code so you'll answer, yeah.

Steve: Yup. And now they actually use your prefix. That's the one - because I've been watching this. It just makes my blood boil, the abuse of my private phone line. And I'm one of 12 people who still have a landline. But still, anyway, so he says: "She gets the mortgage, debt consolidation, and IRS is filing a lawsuit final notice scams. She is on the DNC list."

Leo: Yeah, doesn't matter, yeah.

Steve: And I can vouch for the fact it absolutely means nothing. He says: "Other than terminating landline service, what can we do? Any advice would be great."

Leo: Well, it happens on cells just as much.

Steve: Yes.

Leo: They don't care.

Steve: Yes. So the problem appears to be getting worse because I was surprised by how many people mentioned it. I can vouch for the fact because I have Caller ID, and I look at the Caller ID display. And as you said, Leo, first they began using my area code. Then they began using my prefix. And in fact I got caught out a few times thinking, whoa, is this a neighbor, because it happens that in our neighborhood everybody has, like, the same prefix in this region. And so I thought, well, maybe it actually is somebody who knows me. So now I don't - I'm not getting suckered by that. So it's area code and then prefix, and so they make up a random last four digits.

I verified my - I have actually three landlines, two voice and one fax. I still have a fax line, and Sue sends me things every so often, and we'll fax things back and forth, just to have documents. I've been on the Do Not Call Registry since July 27th of 2003, coming up on 14 years. And it makes absolutely no sense. As you said, Leo, doesn't matter. Cell phones are not immune. I have a huge and growing list of blocked numbers on my iPhone, which I never even use. So the other thing we know is it's not like they're actually getting your number from somewhere because that's the other thing I notice is, since I have three landlines, they will ring sequentially with the same nonsense. So they're just running through, sequentially dialing everything.

Now, iOS apps can manage a Do Not Disturb whitelist. But unfortunately it doesn't prevent the call from being accepted. It's just able to, if Do Not Disturb is active, it immediately goes to voicemail. And I just saw, as I was doing some updated research on this, the Republican National Committee is trying to get the FCC to change some qualification of what counts as a call because it turns out there is a way to cause a mobile device to immediately go to voicemail. They're trying to get that ruled as a non-call so that essentially the RNC will be able to inject voicemail into your mobile phones without controls, without any recourse.

Leo: Oh, god. Horrible.

Steve: So what we need is a phone firewall. Mark Thompson, who is a friend of ours, who is a techie, he set up a PC-based Asterisk system years ago, which I should mention is not for the faint of heart. I mean, the problem is telephony in general uses a completely made-up vocabulary where none of the terms have any normal world meaning. So you have to, like, learn a whole new language to know what a termination is and, I mean, I can't even - I remember looking at it, thinking maybe I should do this. And it's like, oh, no.

Leo: I did the same thing. And there's distros where it's all set up. It's still nontrivial, yeah.

Steve: Yeah, yeah. So what he has is an extension number. So when you dial Mark's line, an automated assistant, an automated attendant picks up and says, "Please dial the extension number of the party you wish to call." And then that's all it does. And if he hasn't told you ahead of time...

Leo: That's a good idea.

Steve: ...then, yeah. So the problem is...

Leo: The problem is your doctor calls, he doesn't know your extension number, and you're going to miss that call.

Steve: Exactly. Exactly. And just a couple weeks ago I received a call from an attorney representing someone of whom I am their executor of their will. A lot of people choose me for that for some reason. So it was like, uh-oh.

Leo: You're so trustworthy. And you're going to live a lot longer than the rest of us.

Steve: Actually, in this case they said, "We trust you more than anyone in our family."

Leo: Yeah, that's not unusual, actually.

Steve: So I said, well, okay, fine, I'm happy to be your executor. Anyway, the point is that, if I had something like Mark has, they couldn't have gotten to me. And I want people to get to me. So what we learned with Internet firewalls is that you can, just as with packets, you cannot block malicious. The only solution is to whitelist. So this has been a growing problem for me. And I would probably have written a solution for myself, if I hadn't been on a roll with SQRL, because what I realized was - and I'm just going to leave this out here for our listeners. There is a simple solution. And if somebody wants to create one, I will publish it on the podcast to our listeners.

The simple solution is an $18 USB voice modem from Amazon. They're just little dongles. In fact, here is one that I got. I mean, it's got the RJ-11 connector on one end and a USB on the other. The famous Hayes AT command set, there is something known as the voice extensions, also known as the TAM, T-A-M, feature set, stands for Telephone Answering Machine. Part of what a voice modem can do is to deliver a prerecorded message. It also understands Caller ID. And so here's how this thing would work. First of all, so you need to plug this into a computer. And if I were doing it, I would just use a PC. The problem is that's non-optimal. What's optimal is the $10 Raspberry Pi Zero, which is all you need. It's got a USB interface. It's $10. So that and an $18 voice modem. So for less than 30 bucks you've got a solution, although I'm not a Raspberry Pi developer. Which is why...

Leo: And this isn't going to work for a cell phone, and fewer and fewer people...

Steve: Well, it's going to work for mine because I'm going to turn on call forwarding.

Leo: Ah. There you go.

Steve: So everything that comes into my cell phone - first of all, I was looking to see whether Verizon, because I'm a Verizon cell phone user, apparently per carrier in the settings for the phone sometimes there's forwarding available. It looks like I have to, for a Verizon customer, you need to enable it through *72 or something, rather than just turning it on in the UI. But it's funny, I've had this account since the beginning of time. My total lifetime use of the phone is one hour and one minute.

Leo: You don't call out much.

Steve: I don't ever talk.

Leo: Yeah, it's a computer.

Steve: It's a pocket computer and text messaging. So I'm never getting calls there. So I'll forward it to home. Anyway, the idea would be, if a call comes in, the phone rings. This thing sees the Caller ID. If it's one it knows, like my mom or Mark Thompson or you or my best friend Mark here, it says, oh, yeah, fine. And it doesn't pick up. It just lets everything keep ringing. If it doesn't know the number, it answers and says: "Hi there. Thanks for calling. I don't accept unsolicited telemarketing calls. If you are an actual human being, please press zero before the end of this message." And then it goes on a little bit longer to give them time. And then, oh, and it says: "Press zero before the end of this message, and then call back," blah blah blah blah blah blah. Beep.

Well, any telemarketing software thinks an answering machine just answered. So you never get a human listening to that, and they hang up. I've been watching this behavior now growing for a year or two. A real person will hear that and go, okay, press zero, hang up. Now they're whitelisted, and then they call back, and they get through. So unknown random numbers. You don't have to blacklist everything because blacklisting doesn't work anymore because Caller ID is just random. So a real person hears that, presses zero, and calls you back. So people like your doctor and dentist and so forth, they're able to get through to you. Yes, you have to jump through a hoop once. But once they're on the whitelist, they're whitelisted.

Anyway, that's the solution. There is an $80 device on Amazon that's the Sentry v3.1, which does this. So if you're not a developer, if you don't want to wait for one of our listeners to do it, which would cost 30 bucks for a voice modem and a Raspberry Pi Zero, you could pay 80 right now. I did send this back to the person who asked, and he said he just ordered it for his mom. And it is a cute little thing.

Leo: And does it say "Press zero if you're a human?"

Steve: Yes, yes.

Leo: Oh, that's great.

Steve: It has a built-in message, and you can - but I would rather use my voice. And so you can say "Press zero if you're a human, and then call back." And so they press zero.

That whitelists them. If by any chance someone does leak through, you're able to browse the whitelist and move them to the blacklist, or just delete them. But it turns out, in my experience, whenever I answer and say hello, then there's a click click click click, and then someone from some other country gets on the line. So it's clear that there isn't a human being. What the auto robo dialing software is looking for, an end of the ring and then a hello, a short thing. Any extended speech, followed by a beep, is regarded as an answering machine, and they just...

Leo: Makes sense, yeah.

Steve: It goes on to the next one. So I think this would perfectly block. It's not necessary to spend $80, and I would dearly love to have time to do this myself, but I don't have time. So if any of our listeners says, hey, that looks like a fun - that's a cool project, the idea being that you could use your phone to digitize your message with the incoming voice modem, store that file, and then that's what you play back when it doesn't recognize Caller ID. So it's a whitelisting, low hassle, and could potentially completely end this problem. But there is something called the Sentry v3.1. There's also a Version 1 and a 2, but they don't quite do all of this, available on Amazon.

Leo: Good idea. Really interesting idea. I think that's a better solution.

Steve: Unfortunately, we need something because it's just getting out of control.

Leo: Oh, yeah, it's terrible.

Steve: And it's not that, I mean, I look at Caller ID, and I just don't pick up. But it just frosts me that, unfortunately, that the world is such that there's no control over this, and our privacy is being abused like this.

Leo: Yeah, terrible.

Steve: So Wayne Patton sent me a tweet saying: "@SGgrc Isn't printing out your second-factor QR codes," meaning for the time-based authentication that I've talked about, "sort of like writing down your passwords? I'm probably missing something?" And it's like, uh...

Leo: It's exactly like writing down your passwords.

Steve: And I responded, yes, exactly. I said, "And writing down passwords..."

Leo: Nothing wrong with that.

Steve: "...is the officially recommended best practice." I said, as Bruce Schneier once

brilliantly noted, and I'm paraphrasing, "It's far more secure to use a password that you cannot remember. So write it down on paper. We already have plenty of systems available for managing bits of paper, like physical wallets." And so the point is that, while a local physical attack does need to be considered, the big threat is the automatable remote network attack, for which strong passwords provide the best security. So yes, Wayne, printing out your second-factor QR codes is exactly like writing down your passwords. Which is best practice.

Leo: And there's nothing wrong with that.

Steve: Simon Zerafa, who always manages to send me something interesting, sent OCSP, which we discussed at length last week, seems to be alive and well. And he quoted somebody tweeting that Let's Encrypt handles 25 billion, with a "B," OCSP requests per month. Get this. That averages 10,000 per second. And remember that Let's Encrypt had an outage a few weeks back. And I hadn't dug into it very deeply. But this affected them from May 18th through the 19th initially, and then it got worse. From around 6:00 in the morning on the 19th until 10:00 in the evening it became a major outage, they wrote, of both their OCSP and the main ACME API used for certificate issuance, where approximately 80% of the requests were failing during that phase. So I was curious to find out, like, to dig in a little bit more than I had bothered to, what was going on. And I got a kick out of the problem.

They wrote: "The initial cause was a code deploy aimed to fix a problem with slash collapsing. The OCSP protocol specifies that requests can be made via POST or GET," which are the two oldest query types for HTTP queries. "Since the request body is binary data, GET requests must be encoded in some ASCII-safe way." Meaning to turn the binary into ASCII. "The OCSP RFC 46" - Jesus, that's a low-numbered RFC - "back in 1999 predated by several years the common use of base64url encoding," which was first standardized in a much later RFC in 2003. So it defined the GET request - so OCSP defined the GET request as the base64 encoding.

Now, I should step back and just say that was never correct. That was always wrong. And I'm frankly shocked that that hasn't been fixed yet. So essentially what this means - here's the problem. Base64 encoding takes groups of three bytes of 24 bits. Right? Because a byte is eight bits. So three of those is 24 bits. And it regroups them into four pieces of six bits. So that gives you - so six bits is 64 possibilities. So it turns out if you use uppercase alpha, lowercase alpha, and the digits 0 through 9, that gives you 62 symbols. So we're two short. Base64 chose plus (+) and forward slash (/) as the final two symbols to make up the 64-symbol alphabet that you get when you reparse 24 bits from three bytes into four six-bit symbols, so four characters from three bytes.

And the problem is that forward slash is not URL safe. It is, as we know, forward slash is a component of URLs. So, like, how could you possibly ever have that in a GET request? I mean, for example, this is one of the advantages of letting SQRL mature awhile. It always used base64url encoding because I've always known that you can't have forward slashes. Base64url encoding changes those final two symbols. Instead of using plus and forward slash, it defines them as minus and underscore, which are both URL-safe characters that won't be confused by anything parsing a URL.

So it turns out that underlay the problem that Let's Encrypt had because they implemented the spec as it was, and the nature of their implementation caused a bunch of forward slashes to occur, which ended up screwing up the parsing of the use of the GET version of the request. So I just got a kick out of the fact that something that, like,

should never have been problem bit these people because somebody wasn't paying attention to the fact that you can't use base64, straight original base64 encoding, for binary data in a URL.

In SQRL, the server provides what we call of course the "nut," because it's SQRL, which is actually a nonce which the server uses to produce a unique challenge which the SQRL protocol signs. Well, that is in that little QR code; and, if you click on it, it's in the URL that you click on. That's binary, so we base64url encode it to make it safe. And that's the query that the SQRL client picks up and processes. So it's not going to bite us, but it did bite Let's Encrypt and the OCSP mainstream protocol, surprisingly.

Christopher Meacham tweeted, he said: "@SGgrc Despite #AntiTrump, the Mar-a-Lago vulnerability was not responsibly disclosed." And he's referring to that ProPublica article we discussed where they trawled out in the bay and aimed an antenna at Mar-a-Lago and also a bunch of other of the well-known Trump properties. And I'd have to say I agree with that, although some things are just being done in plain sight. And it's not like it took a rocket scientist to think, wow, let's check the WiFi at Mar-a-Lago. And as their article noted, there are probably a lot of Pringles cans being aimed at Mar-a-Lago at the moment.

Hipposec said: "Would love to hear a tidbit on SN about becoming a security researcher, especially transitioning from hobby to career." And I thought about that a bit. And I thought, you know, the biggest and really only requirement is an understanding of low-level code. If you think about that, what we are, I mean, all of the things we see, everything we deal with is the assembly code, one step up from the machine language, but typically down below the source code. I mean, it is the case that open source, where you can see the problem, you can go, oh, wait a minute, they defined that as a WORD, but they used it as a DWORD, and that's not going to fit, so that's a problem. Yes. But normally the problems are found by actually looking at the machine language.

So I would argue that what all security researchers have, what we keep seeing almost everyone doing, is understanding the code that the chip itself is reading. And so there's no hurry. It takes a while to get the hang of that. But there are so many free tools and so much information available on the Internet now to help somebody in that journey, that I think it's a cool thing to do.

But really you can't be a security researcher, well, I mean, you can. Some of the problems that Tavis finds, for example, are his deep knowledge of JavaScript and the DOM, like recently the problem he found, for example, when he was taking a shower, with LastPass. But again, really, really deep understanding is what you need. And so that just, you know, it takes time and involvement, if somehow you can get that.

Jason Werner tweeted: "Just wondering what warranted such a high opinion of Edge." He says: "Seems lacking in configurability and choice, let alone privacy." And so I'll say to Jason and our listeners that the thing I skipped when I said we're running out of time was a comparison of the resource consumption. And it was someone who - a well-known website opened five browser tabs in Firefox, Chrome, and Edge. Firefox required something like 236MB. Chrome required a little more than twice that, 500, I think, and something megs. Edge required 1.4GB for the same five tabs.

Leo: Oh. Wow.

Steve: So, yes. And you know me. I'm a Firefox user, and I care about resource

consumption, just because. Nowadays - I heard you talking about, Leo, that 18-core i9 Extreme processor. I don't know if you know it comes with a large jug of Freon, which you just pour slowly on top of the chip while it's turning on, and it just evaporates on contact with the top of that chip. My lord. So all we're doing now is just throwing more resources at things. Anyway, my high opinion of Edge comes, not because it is not lacking in configurability and choice and privacy, I don't disagree with those things, but because it is a from-scratch rewrite of IE. And they did a beautiful job.

I mean, maybe it'll get a little more svelte over time. I wouldn't hold my breath. But, I mean, as a mass consumer base browser that most people are going to be using by default on Windows 10, and with no choice on Windows 10 S, the good news is it's not a piece of garbage, the way IE had become. Even though IE got better, too, later in its life, Edge, they did a beautiful - there's a lot of good technology in Edge. And so I'm not going to ding them just because they're Microsoft. It's a good browser, and it's the browser that Windows 10 users are going to be using. Thank goodness, you know, it's not IE6 any longer, or there'd be hell to pay.

Leo: It is also the case, people always talk to me, they call up, say my RAM is 100% use. A good operating system is going to use as much memory as available to it because that speeds up performance. So it's not necessarily a measure to see how much RAM it consumes. It's to see, when you start running other processes, if it gives that RAM up, if it runs more skinnily. I mean, if you've got 8GB of RAM, there's no reason for the only program running to not use 8GB.

Steve: Correct.

Leo: Or 7, anyway; right?

Steve: Correct. Yeah. And as long as you've got a large bottle of Freon handy.

Leo: And you need a memory manager that's going to be smart enough and a program that's going to be smart enough to release memory when it's needed by other processes. But it speeds everything up to use all the RAM it can use.

Steve: Yeah. My cranky finicky old 32-bit XP machine starts getting unhappy as I get up near 3GB. And so I just hit up Firefox, I restart Firefox.

Leo: Yeah. There's a limit to how much you have, you know. If you have, for instance, if you have a 32-bit processor, you can't address more than 4GB; right?

Steve: Right. And in XP the LSASS process, which is then responsible for security stuff, it apparently is leaking memory. So after a couple weeks I'll just reboot the system, and it's much happier after that. But it's old and creaky. I'll switch to it when I can take some time off, probably before I - maybe after the first release of SpinRite.

And our final question: Mark Dean says: "Analogy needed. Hey, Steve. I was wondering," he writes, "how you would explain this. I have a non-technical friend who is against

software companies releasing their code as open source. He thinks that it is giving hackers access to just go into systems. I tried to tell him that having the source code in and of itself does not mean hackers have the ability to hack into a system. They may know more about the details of the system, but that's all," he says, parens, "(unless things like passwords are in the source)," which of course they're not in any well-written open source. He says: "I tried to use the analogy about encryption, where the algorithm is well known, but that doesn't mean the keys or values can be easily derived," he says, "but that probably confused him more." And, yeah, I don't think that's a great analogy, either. "Any suggestions on explaining how knowledge of the source doesn't automatically mean an easy hack? Thanks."

And so I thought about that a bit, and I wrote back to him. And I said, here's an analogy. Not publishing the source code would be like not printing a map to get from point A to point B. Having a map would allow people to more easily inspect the route and find a better solution. But not having a map does not prevent anyone from driving the route to obtain the same information. In other words, it is inherently impossible to keep the map's route a secret, just as it's inherently impossible to keep computer code a secret. Anyone who's interested can obtain the same information, and bad guys will be motivated to do so. But publishing the map makes it easier for others, without the strong nefarious motivation of exploitation, to casually glance at the map and suggest improvements and help out. And also note that publishing the map also doesn't guarantee that good guys will bother to help. But it does make it much easier and more likely that someone will.

So I kind of like that. I came up with that in response to the question. But to me it sort of - it embodies the concepts and the principles, and that is that, yes, the source code makes what it's doing more accessible. But not having a source code doesn't keep it out, doesn't keep the same operation out of the realm of the bad guys. And having the source code makes it much easier for good guys to help, if they're so inclined to do so, much like looking at a map would.

Leo: It's good.

Steve: There's our analogy for the week.

Leo: And security experts often agree that the weakest form of security is through obscurity.

Steve: Right.

Leo: It can be a form of security, but it's not very strong, especially if you can disassemble the binary.

Well, we have completed this journey into the mind of the hacker. And now we have to go back into the real world, I'm sorry to say. We do this show with Steve every Tuesday, right after MacBreak Weekly, supposed to be about 1:30 Pacific, that's 4:30 Eastern time, 20:30 UTC, if you want to watch live and join us in the chatroom at irc.twit.tv. Most people, though, just prefer to download a copy, and you can do that from Steve's site, GRC.com. He not only has audio copies there, but he has

transcripts, so you can read along or search the transcript. That's a very valuable tool that he funds himself. Thank you, Steve.

While you're there you can help Steve with his funding by buying a copy of SpinRite, the world's best hard drive maintenance and recovery utility. And then, since you've paid for it, you can now browse the site and find all sorts of other cool stuff that's absolutely free. GRC.com. We also have audio and video at our website, TWiT.tv/sn. But we also encourage you to subscribe. Whatever you use to listen to podcasts probably has a facility so that you can automatically download it each week. And that way you'll get up on a Wednesday morning and go, "Oh, look, brand new copy of Security Now!." Plus, of all the shows we do, this is the one I think people want the whole set. All 614 episodes.

Steve: They're all there, baby.

Leo: Why not have them all? And hand them down to your children, your eight year old. When she stops swinging, she can become a true geek. Thank you, Steve.

Steve: Thank you, my friend. And enjoy your vacation. I guess we will not have you...

Leo: I won't be here next week, yeah.

Steve: ...for two weeks.

Leo: Yeah, that's right. I will be on vacation next Tuesday and the Tuesday following, but then I'll be back on the 20th.

Steve: Yay. I know you're looking forward to it, so have a great time, and we'll be back with you in three weeks.

Leo: Yeah. And I'm sure we have somebody wonderful, probably Father Robert, but I'm not sure, hosting with you. Maybe Jason.

Steve: Cool.

Leo: Yeah. Have a great time, and I will see you in two weeks.

Steve: Right-o, buddy.

Leo: Bye-bye.

Steve: Bye.