Hack the Vote

Description: This week we look at the expected DEF CON fallout including the hacking of U.S. election voting machines, Microsoft's enhanced Bug Bounty Program, the wormification of the Broadcom WiFi firmware flaw, the worries when autonomous AI agents begin speaking in their own language which we cannot understand, Apple's pulling VPN clients from its Chinese App Store, a follow-up on iRobot's floor plan mapping intentions, some news on the Chrome browser front, the 18th Vault 7 WikiLeaks dump, and some closing-the-loop feedback from our terrific podcast followers.

High quality (64 kbps) mp3 audio file URL: http://media.GRC.com/sn/SN-622.mp3
Quarter size (16 kbps) mp3 audio file URL: http://media.GRC.com/sn/sn-622-lq.mp3

SHOW TEASE: It's time for Security Now!. Much more news from Black Hat and DEF CON. Steve's going to look at the voting machine hacks and a few other scary things like Broadpwn, which had the potential to infect one billion iPhone and Android devices. That and all the security news coming up next on Security Now!.

Leo Laporte: This is Security Now! with Steve Gibson, Episode 622, recorded August 1st, 2017: Hack the Vote.

It's time for Security Now!, the show where we cover your security and privacy online, on the Internet, on the hackable voting machines, everywhere. Here he is, the Commander in Chief, Mr. Steven Gibson of GRC.com. Hello, Steve.

Steve Gibson: Great to be with you again, as always, my friend. Yeah. So August 1st, and we are closing in on the end of year 12.

Leo: Holy moly.

Steve: Our first podcast, which I will forever regret not numbering at zero - of course then we'd have the whole problem of 622 actually being our 623rd podcast, so that would get tiresome after a while. But, yeah, it was toward, I think it was like, don't remember now, the 17th or something. So a couple weeks from now we will be into the 13th, the lucky 13 year of the podcast.

So of course, as we've been predicting for a couple weeks, we are now on the backside of sort of the doubleheader of conferences in Las Vegas, Black Hat and DEF CON. And as expected, we've got some final results. Some of them...

Leo: The results are in.

Steve: The results are in, and duck and cover. So we're going to take a look at the expected DEF CON fallout, including the hacking of U.S. election voting machines, thus the title of this podcast, "Hack the Vote." Microsoft also last week announced their enhanced bug bounty program. We learned at DEF CON of the wormification - there's a lovely term - the wormification of the Broadcom WiFi firmware flaw, which as we know we were covering last month, through the month of July, as Google and Apple both got themselves up to date, but we'll talk about the consequences of that. Then some interesting miscoverage, but really a classic instance of inflammatory headlines about the press's coverage of Facebook's autonomous AI agents beginning to speak their own language.

Leo: Was it Google or Facebook?

Steve: It was Facebook.

Leo: Oh, okay.

Steve: And anyway, I drilled down and found the actual root story, and it was like, okay. Well, I mean, it's definitely interesting, but it's not nearly what the headlines were, like oh, my god.

Leo: They're communicating amongst themselves.

Steve: We don't know what they're saying.

Leo: Ohhh.

Steve: We also have the brouhaha, which I think is, well, we'll talk about it, about Apple pulling their VPN clients from their Chinese App Store. A follow-up on iRobot's floor plan mapping intentions. Some news on the Chrome browser front. The 18th Vault 7 WikiLeaks CIA dump.

Leo: Uh-oh.

Steve: Three new pieces of news from there. Some closing-the-loop feedback from our terrific podcast listeners. So I think yet another great podcast.

Leo: Well, I'm so thrilled.

Steve: Oh, and a wonderful Picture of the Week. Just one of those, you just can't make this stuff up, Pictures of the Week.

Leo: Oh, oh, I like it. I like it. And it ties into a product of your own, as a matter of fact.

Steve: That's why so many people brought it to my attention. I got a bunch of retweets. It's like, okay.

Leo: Literally. Literally retweeting.

Steve: Literally retweet.

Leo: All right, Steve.

Steve: So our Picture of the Week is, for those who are listening, there's an installation on a wall which was - clearly some effort was put into it because a conduit was run up the outside of the wall, where an electrical receptacle was put. And then this big black Model QB-4 Ultrasonic Bird Repeller was screwed to the wall and attached to this source of AC power. I mean, so this was meant to do its job. And the punchline on this which is so funny is that there is a bird nest, with bird, built on top of this thing, just sitting there peacefully, keeping her eggs warm, and apparently not the least bit perturbed by the QB-4 Ultrasonic Bird Repeller.

Leo: Pretty funny. Pretty funny. Love it. Love it.

Steve: Ineffective countermeasures is the story. So last week, during the recently completed DEF CON cybersecurity conference, several hackers managed to hack into multiple U.S. voting machines, in some cases in as little as a few minutes, some cases taking an hour and a half or so. For the first time ever, but likely not the last because it was such a success, DEF CON hosted what they called their "Voting Machine Village," during which the conference's tech-savvy attendees tried and succeeded to hack many commercial voting machine systems and help catch vulnerabilities in them.

This year, the first year this was done, but the planners already intend to make this thing a regular feature, provided 30 different pieces of voting equipment used in U.S. elections, including the Sequoia AVC Edge, the ES&S iVotronic, the AccuVote TSX, the WinVote, and the - is it Diebold? Yeah, Diebold or Diebold? How do you say that?

Leo: Diebold.

Steve: Diebold ExpressPoll 4000 voting machines. The executive summary is that every one of the 30 machines that the hackers poked at were, to varying degrees, hacked. So not a single one of them resisted attack. Now, in fairness, these weren't hands-off, over-the-air attacks. There were some of those, but in some cases they required physical access to the machine, but like a USB port or something, very much like a laptop. And many of the attacks which we're familiar with, that we've talked about over the years, work on these machines because, as we know, unfortunately, they are internally typically well-known architectures where a screen and some custom software were slapped on top, you know, much like the machines that run the nuclear submarines in the U.K.

Leo: Yeah, well-known architectures like Windows XP and Windows CE, yeah, those well-known architectures.

Steve: Yeah, yeah, exactly.

Leo: Yeah.

Steve: So the DEF CON hackers took complete control of an e-poll book, which is one of the Diebold devices. In fact, I just saw some follow-up news saying that I think it was 650,000 personal information of voters was found on one of those.

Leo: Wow. Like they bought it on eBay, and it still had the stuff on it.

Steve: Yes, exactly.

Leo: Oh, come on. Geez Louise.

Steve: I know. This, I mean, it's just - it's as bad as everything we've been covering about IoT...

Leo: Yeah, times 10.

Steve: ...applied to voting machines.

Leo: Yeah.

Steve: Yes, where we really would rather that they weren't this vulnerable.

Leo: Right.

Steve: So that particular e-poll book is currently in use in dozens of states where voters

sign in and receive their ballots. So that's not per se a voting machine, but it's like a pre-voting staging device. They also discovered and exploited significant security flaws in the AccuVote TSX, which is currently in use in 19 states; the Sequoia AVC Edge, used in 13 states; and another hacker broke into the hardware and firmware of another Diebold product, the TSX voting machine.

So, although somewhat less surprising, the WinVote voting machine had long been removed from use due to its vulnerabilities, so that wasn't a clear and present danger. But those problems were again confirmed, and it was once in wide use while being horribly insecure. They found that a remote access vulnerability in the WinVote OS exposed real election data that was still stored in the machine. And another hacker hacked into the Express-Pollbook system, that's that Diebold system, exposing the internal data structure via a known OpenSSL vulnerability, allowing anyone to carry out remote attacks. So remote attacks also are possible. These do not require local physical access.

Jake Braun, who is a cybersecurity expert at the University of Chicago and who convinced DEF CON's founder Jeff Moss on the idea of creating this Voting Machine Village, said, quote: "Without question, our voting systems are weak and susceptible to attack." He said: "Thanks to the contributors of the hacker community today, we have uncovered even more about exactly how." There will be a more formal report forthcoming.

I have a link in the show notes here about the place on GitHub where Joseph L. Hall is assembling the Voting Village report. Right now they have just like rough working notes, sort of the output of the various hackers sort of dumping their content out. There will be a formal report. And I did find coverage of this on The Hill website, so of course that's DC's, one of DC's reporting arms, and that suggests that this is getting the kind of attention that we need it to get. So that's great.

Another person, Harri Hursti, who's the cofounder of Nordic Innovation Labs, and he was also one of the events co-organizers, said that the Village was announced at the last minute, but people were active in the forums, looking to understand the problems. "The changes have to start somewhere," he said. "This year it's in this room. Next year it will be in a bigger room." So that's just - that's all for the best, as we know.

Eric Hodge, who's the director of consulting at CyberScout, a consultant for Kentucky's Board of Elections, said: "The best possible outcome is that the village results in a book of vulnerabilities to share with the FEC, the states, and other firms like ours." I would take issue with that, and I will in a second.

So DEF CON's Voting Machine Village was the first time most researchers had ever had access to voting machines. That's what has to change, okay, so think about that. This was the first time most researchers ever had access to voting machines. This is because they are considered proprietary, and their manufacturers protect them and don't allow anyone to poke at them. Well, every lesson we have learned through the years of this podcast has taught us that that is a strategy that is doomed to failure. That's what has to change.

There's no way that any state or county government should be allowed to spend taxpayer money on machines which have not been independently audited by security researchers. Rather than treating their machines like proprietary closed boxes, voting machine manufacturers should gleefully turn their machines over to every independent security researcher they can find for the purpose of hardening their security offerings. Then purchasers should have candidate purchases again independently vetted by still

other independent researchers.

In other words, the lesson we keep learning is that, even with the best of intentions, mistakes are made. And right now we're in a completely uncontrolled environment where the manufacturers are boasting about their military-grade security and using unearned reputation, basically. I mean, everyone knows Diebold. They're a well-known company. But they're obviously not able to produce secure voting machines. They shouldn't be purchased without some reason to believe that they're secure.

And of course I'm reminded of Steve Ballmer's pre-WinXP release where he was prancing around the stage screaming into a microphone about how XP was the most totally secure operating system ever created before its release. And as we all know, shortly after its release XP was a security disaster. So as I've often said, the security of a system must be designed in, but any actual systems-delivered security can only be proven afterwards.

So I'm so happy that this was done, that it's gotten a lot of press coverage, that there's egg on the face, deservedly so, because every one of these manufacturers was of course claiming impenetrable security, but never being willing to put it to the test. So that was tested. Not a single machine was found to be secure. So after I put the show together I saw some news about some forthcoming potential IoT legislation. I think that's all for the best. We need something similar for voting machines. Otherwise, everyone talking about "the integrity of our system," it's just empty air.

Now, that said, there is an advantage to, and I've also often said this, to a heterogeneous environment, that is, we don't want everyone in the country, at every state of government, using a single machine. That's very dangerous. So the fact that we have many systems, many companies creating widely differing machines, that's better. And we also want to do things like avoid hooking them all together into one big network.

Leo: One of these machines was WiFi-enabled, which is obviously a terrible idea for a voting machine; right?

Steve: Yes, yes.

Leo: I mean, you don't even need physical access.

Steve: Yes.

Leo: That is said to be the strength of the U.S. voting system is that every registrar in every county has a choice, and so there is no homogenous system. So that's good.

Steve: Right, yeah. And again, the problem is that we see big companies purchasing each other or big companies purchasing smaller ones and reducing choice and reducing competition, but also reducing heterogeneity.

Leo: Right.

Steve: And that's a strength that we have now. We need not to lose that.

Leo: What you really want is every voting machine needs to have a paper trail so that, in case of concerns, you can have an audit. The problem is electronic voting machines that have no paper trail.

Steve: Right.

Leo: Then the machine is the only reliable source, or unreliable source, of the vote.

Steve: Right.

Leo: If you have a paper trail for every vote, you can do spot checking; and, in case of trouble, you can do a full recount on paper.

Steve: Yeah. And I think, I mean, were I architecting these, I would think, I mean, not having looked at this closely, it would seem to me that having a machine produce - you know what a fan I am of paper. You know, SQRL prints its secrets on paper because they're offline inherently, and you can't attack them by WiFi or a USB dongle. So imagine if the machine were just a transcriber of a friendly UI which then printed out, on paper, a barcode summary of exactly what that voter did. And then in a separate stage you collect all of those machines' spools of paper and feed them through...

Leo: Precisely. Exactly.

Steve: ...a master tabulator.

Leo: Right.

Steve: And then you can do it as much as you want. So you have several stages of checkpoint that way.

Leo: I'm going to point you to - we did a Triangulation on this before the election this year, or last year, I guess, to VerifiedVoting.org, which is a really great organization that talks about all of this. And of course they talked about the hacking conference. And there is, under the - I'm not sure where it is on here. But there is a proposal for a way to do this that would be highly secure, and it solves a lot of these problems. These people have been thinking about this for a long time. And so, you know...

Steve: Good.

Leo: This is, frankly, a solved problem by some very smart computer scientists and mathematicians and election folks. But, you know...

Steve: Inertia.

Leo: Inertia, yeah. And money, frankly, because it costs money to do this.

Steve: Yeah. And in fact, in my example, if the format of the barcode which was printed out was standardized, and it absolutely should be, then you have a standard point in between the machine that takes the votes and produces the barcode, and on the backend the tabulator that optically ingests all of those in a continuous paper strip and tabulates. So you're not reducing competition, you're creating a standard protocol that allows machines on both ends to be independently designed and created and sold in a competitive marketplace, but with a common standard interchange format between the front end and the back end.

Leo: Yeah. And you trust - I would trust Caltech and MIT, and they have something they call the Voting Technology Project that addresses a variety of ways, not just technologies to take votes, but maybe even better ways to vote because of course there's other proposals about weighted voting and things like that, that are interesting. But this is a good site. It's vote.caltech.edu.

Steve: Nice. So we did get news of a scary present and not-to-be-fixed vulnerability. Okay. So this is an SMB, and we know that stands for Server Message Block. That's a.k.a. Microsoft's file and printer sharing protocol. Eight years ago, back in '09, which we talked about at the time, there was an attack tool known as "Slow Loris," which operated over HTTP web connections. That was a server-side resource depletion attack which exhausted a server's incoming connection-handling capacity while at the same time requiring very little bandwidth from the attacker. So unlike today's traditional DoS and DDoS attacks, which use bandwidth flooding to just saturate the links inbound and prevent legitimate traffic from being able to compete with the attack traffic, then and now, even one attacking machine could bring a big remote website to its knees, preventing legitimate visitors from obtaining access.

So what was revealed at last week's DEF CON was a similar, but new, server-side resource depletion attack which it used that troublesome version one of the SMB protocol, which as we know is still supported in Microsoft's OSes, despite being long obsolete. And we know Microsoft has said they're finally going to deprecate it. It will be disabled in this forthcoming Windows 10 Creators Update release. So on the newest machines it will finally be going away.

But it turns out that doesn't solve the problem. Sean Dillon of RiskSense was among the first researchers to analyze the EternalBlue exploit, which was that leaked NSA SMBv1 exploit which, as we know and covered at the time, was used to spread the WannaCry ransomware. It was during his analysis of EternalBlue that he discovered another, that is, this related issue. He wrote: "While working on EternalBlue, we observed a pattern in the way memory allocations were done on the non-paged pool of the Windows kernel. The non-paged pool is memory that must be reserved in physical RAM. It can't be swapped out. That's the most precious pool of memory on the system." He said, "We figured out

how to exhaust that pool, even on servers that are very beefy, even having 128GB of memory." And he said: "We can take such a server down with a Raspberry Pi."

Leo: Ah, ah, ah, $35 computer.

Steve: A Raspberry Pi and 20 lines of Python.

Leo: Oh, wow. So, Python, is there anything it can't do? No.

Steve: Like Slow Loris before it, this new attack they dubbed "SMBloris" in homage to Slow Loris. It leverages a memory-handling bug that could be exploited by attackers to shut down big web servers with small computers. However, it was initially believed that attackers could only exploit the SMBloris vulnerability if the target machine had SMBv1 exposed to the Internet. But that now appears - well, or internally, like on an Intranet, where it's much more likely that machines will have SMB exposed. We would expect corporate firewalls, and certainly most ISPs today are still blocking the SMB ports on behalf of their subscribers. But on an Intranet, in a large enough network, people could get up to some mischief.

Anyway, it turns out that that was hopeful thinking, that is, that it was only SMBv1. The Register later updated their initial coverage to say, quote: "According to Microsoft's SMB Supremo," as they called him, "Ned Pyle, SMBloris affects all versions of SMB, not v1 as first thought," because Microsoft said it all happens so early on in the connection. So this problem will not be going away, even after the forthcoming Win10 Fall Creators Update, which disables support for SMBv1.

So the protocol technically, sort of the sub-protocol within SMB, is called NBSS, which is the NetBIOS Session Service. Every connection to it, when tweaked in this special way, is able to cause the allocation of 128K of page-locked non-swappable kernel memory. That is, so all of the protocol support is down in the OS. So this is not allocated in userland. This is in the kernel. So the kernel says, oh, here comes an incoming SMB connection, and it has just notified us that it has 128K of data it's about to send. So the kernel preemptively obtains a 128K block of memory which it locks in RAM, so it's non-swappable in the kernel, so that it can then asynchronously - it then acknowledges that it's ready to receive that per the protocol, and that memory is then retained while awaiting the 128K of follow-on memory.

So all of our technically astute listeners know where this is going. That memory, that follow-on data is never delivered. It's just a claim for future delivery which the protocol is then forced to preallocate. Now, connections which are not active will get shut down after 30 seconds. But it turns out that's enough time. With more than 64,000 TCP client-side ports available, an attacking machine can force the allocation. A single attacking machine at a single IP, using the available client ports, with 128K per port, can force the allocation of 8GB.

And since the protocol is available both over IPv4 and IPv6, both can be used simultaneously, bumping the forced allocation to 16GB. And if a second source IP is available, the memory commit could be doubled again to 32GB of RAM, using just two IPs, and so on. Once the attack has triggered memory saturation, the server will crash, and a system reboot will be required in order to resume normal operation. So anyway, Sean Dillon was the person at RiskSense who found this and who delivered the

presentation last week at DEF CON, saying that a Raspberry Pi could take down the beefiest server using 20 lines of Python code.

So we know that many systems have SMB exposed to the Internet. Thus the disaster of WannaCry a couple months ago when this EternalBlue was weaponized, after being discovered from the NSA leak, into an exploit. And so it's without question that those machines, which still have SMB exposed, and in this case not only v1, but now we know any version of SMB, are probably going to be crashing because there will be hackers that want to play with this, and it's too easy to find SMB exposed on the Internet. It's just not difficult. This has been a problem. This has been the way SMB memory allocation has worked.

Sean said: "I think Microsoft's problem" - because Microsoft, I should say, has said this is not going to be fixed. Microsoft has taken the position that this is a configuration issue. Now, that was arguably the case when v1 was believed to be the attack vector. It'll be interesting to see if they still take the same position. But as Sean said, quote: "I think Microsoft's problem is that it would not be easy to fix. It's the way they've done SMB memory allocation for over 20 years. So everything relies on the fact that the client says 'I have a buffer that I'm sending that's this big.' So the server reserves that much memory so it can handle the anticipated incoming data."

He says, "What we did with this attack was to simply say, 'I have a huge buffer,' and never send the data. There are many components," he writes, "of the protocol support that rely on the fact that the buffer is already allocated, and the size is already known." Which is to say it's not practical to change the implementation, for example, into an auto-sizing buffer that grows as data is actually received. The way it's been done is pretty much the way it has to be. So what we have is an old bug that's present in every version of Windows, which means that attackers will likely attack those machines that have SMB enabled publicly and maybe get up to some mischief in Intranets where it is enabled internally.

For our listeners, we have multiple lines of defense. First of all, as I mentioned, ISPs are still blocking ports 137, 138, and 139, which are the oldest implementation ports for this; and 445, which is the newer version of Windows port. Those are typically blocked. Any incoming traffic with those ports as its destination is just dropped at the ISP edge. And it's because that wasn't originally done, and because of the exposure of this, as I've said, that was the motivation, the impetus for me writing the ShieldsUP! system all those years ago because everybody had those ports exposed. There was no firewall in Windows in the beginning.

So we have ISPs as the first line of defense. We have NAT routers as the second line of defense. A properly configured NAT router will similarly drop that traffic at your border. And all of our modern OSes from XP Service Pack 2 and later have a firewall which is present and enabled by default. So I don't think this is a problem for us. But it's certainly a problem, I mean, it could be a problem in anyone's Intranet, even in a LAN, if something bad got into your system. But again, it's not a data exfiltration bug. It's just to crash your machine. So I could see that people would get up to some mischief, but it's not clear what more they could do.

Leo: All right.

Steve: So I have a link in this next piece, Leo, that you're going to want to take a look at.

Leo: All right.

Steve: Microsoft has announced a new Windows Bug Bounty Program.

Leo: I'm still emptying the trash.

Steve: Last Thursday Microsoft announced a new Windows Bug Bounty Program that will pay researchers - get this, but only for the biggest ones - up to a quarter million dollars.

Leo: Wow.

Steve: For finding and disclosing security vulnerabilities. The bug bounty program...

Leo: That's the Hyper-V bounty.

Steve: Exactly. The biggie is the Hyper-V. If there are any problems there, they really want to know.

Leo: Right.

Steve: It will focus upon a few key areas. And on the page that I've linked to and in their description there's a little menu of what you get paid for which class of bugs. There's Hyper-V problems, mitigation bypass, Windows Defender Application Guard, Microsoft Edge, and then all features made available through the Windows Insider Program. The payouts from Microsoft depend upon where a vulnerability is found and how severe it is, and ranges from a low of $500 for a vulnerability in Edge, all the way up to...

Leo: Wait a minute. So that tells me that there's a lot of those.

Steve: Yeah.

Leo: Is that, I mean, can you judge by that?

Steve: I think you're right. I think that you're exactly right. And that's the first thought I had when I heard that they were upping the ante is that, okay, now they're sort of sure that the easy-to-find bugs have been shaken loose, so they're saying, okay, we're willing to pay more for increasingly bad ones. But I think you're right. The fact that Edge only gets you 500 bucks says, okay, those may not be that difficult to find.

So in their announcement, what was really interesting also, they did something a little differently than others have. They said any critical or important class remote execution,

privilege elevation, or design flaw that compromises a customer's privacy and security will receive a bounty. But they also said they would pay a maximum of 10% of the highest amount discoverers would have received if the discovery was fresh. In other words, even if you find - if you report...

Leo: An existing one.

Steve: ...an existing redundant problem, you can still get 10% of what the first reporter of that problem got.

Leo: That's pretty good.

Steve: So they're actively working to encourage researchers to disclose everything they know of instead of perhaps sitting on vulnerabilities because they're not sure if they're new or not. So bravo to Microsoft. That's, you know, we need that. Again, all the lessons we're learning is that it's only by proactively attacking presumed secure systems that they could be proven secure.

Ooh, boy. DEF CON introduced us to Broadpwn, B-R-O-A-D-P-W-N.

Leo: This one is bad.

Steve: Yes. Which wormifies the Broadcom WiFi firmware bug that we talked about a month ago. As we know, both Google and Apple both have issued patches last month. But at the time that this was, well, prior to its disclosure, but at the time it was discovered, an estimated one billion devices were vulnerable to what was disclosed last week at DEF CON. And of course this is always the way these things go. It starts with a device crash. Then the vulnerability which crashed the device is carefully examined and weaponized into something far more functional and useful to attackers.

A researcher by the name of Nitay Artenstein at Exodus Intelligence was the person who took this to the next level. So I said DEF CON, but it looks like it was actually at Black Hat, the previous conference of the two. He demonstrated a proof-of-concept attack that exploited the vulnerability which, as we know, had by then been patched, both in Android at the beginning of the month and in two patches. In fact, the second patch was our errata from last week saying, well, we thought it was the earlier patch at 10.3.2, but now it looks like it was 10.3.3.

Anyway, so he demonstrated a proof-of-concept of this BCM43xx family of WiFi chips which had long been manufactured by Broadcom. His attack fills the airwaves surrounding any compromised device with WiFi probes requesting low level, meaning WiFi protocol level, no user alert or action required. This thing slips in down in the kernel, actually even below the kernel, in the baseband radio in the Broadcom chip, making connection requests to any and all nearby mobile smartphones which are also Broadcom BCM43xx equipped, which they all are, basically, because that's the chip of choice that everyone was using. A billion devices in service.

When specially devised requests are then sent from the infected device, which infects all other devices within radio range that have not been patched and have this problem, the

attack rewrites the firmware that controls the chip. The compromised chip then itself begins sending the same malicious packets to all other vulnerable devices, setting off a potential flash worm chain reaction. I mean, whoa.

So he wrote: "Broadpwn is a fully remote attack against Broadcom's BCM43xx family of WiFi chipsets which allows for code execution on the main application processor in both Android and iOS. It is based on an unusually powerful zero-day that allowed us to leverage it into a reliable, fully remote exploit." He said his attack worked on a wide range of phones including all iPhones since the iPhone 5; Google's Nexus 5, 6, 6X, and 6P models; the Samsung Note 3 devices; and the Galaxy devices from S3 through S8.

So this is another perfect example and substantiation of what we've been saying on this podcast now for several years, which is it is utterly unsafe to be using any iOS or Android-based device that is not receiving regular security updates. And note that Android is no longer alone here since Apple also eventually abandons their older, yet still functional iOS devices once that older hardware can no longer run the latest version of iOS. So, I mean, and again, that means the device is old, but it does say that even old problems are still problems.

So for any of our listeners who really want the nitty-gritty, I've got a link in the show notes to Nitay's detailed blog posting at ExodusIntel.com. It's 2017, July 26, just titled "Broadpwn." I'm sure you can find it; or the link, as I said, is in the show notes. And it's, I mean, he takes it all the way down. I mean, for anyone who's interested in how this is actually done, he laid it out. So it's a beautiful presentation, more than we need to get to or is practical over a mostly audio podcast. But the details are all there.

And again, we know there is presently a huge number of phones that have that chip, have the vulnerable firmware, and have not been patched. Probably predominantly Android phones, just because they are so quickly abandoned by their cell provider. But you don't even need cell access, just a WiFi radio turned on. So again, I wouldn't be surprised if in the next coming few weeks we're covering the actual outbreak of a Broadpwn flash worm which has just torn through the Android ecosystem of older phones that did not get patched during Google's patch, or any patches sent out from Google to responsible providers of non-Google devices, who then pushed the patches out to their users. Anyway, boy, needs to happen.

Okay. So in this week's crazy headline news, for example, Fox News' headline was "Facebook Engineers Panic, Pull the Plug on AI After Bots Develop Their Own Language." Okay, that was the most over-the-top one. The Facebook engineers did not panic. But even BGR, Boy Genius Report, said: "Facebook AIs develop own language and are immediately shut down." Okay, well, that wasn't - that's not true, either. But at least it doesn't, I mean, it sort of implies panic where none existed.

Many news outlets covered the story and also picked up on each other's coverage, seemingly amping it, like one-upping each other as they linked in a chain. I tracked down the original coverage, which was much drier and far more accurate, over on FastCoDesign.com. And the sober headline was: "AI Is Inventing Languages Humans Can't Understand. Should We Stop It?" And it was a long and thought-provoking piece. I'm just going to summarize it a bit here.

So what they wrote is: "Researchers at Facebook realized their bots were chattering in a new language. Then they stopped it." But they didn't stop it because they were afraid of it. They stopped it because they considered it a programming error that they had not explicitly constrained the dialogue to be English. So, for example, I'll get into this in a little bit of detail in a second, but there was a deliberate negotiation protocol where two

independent AI agents talked to each other. And they were named, appropriately, Alice and Bob. And so, for example, Bob said: "I can can I I everything else." Which is not perfect English. Alice responded: "Balls have zero to me to me to me to me to me to me to me to me to." Which, again, not English. But what's interesting is that they understood each other. That passage...

Leo: Like twins.

Steve: Exactly. Perfect example, Leo. Now, "That passage looks like nonsense, but this 'nonsense,' in quotes, was the discussion of what might be the most sophisticated negotiation software on the planet, negotiation software that had learned and evolved to get the best deal possible with more speed and efficiency and perhaps hidden nuance than we are able to perceive.

"The conversation occurred," as I said, "between two AI agents developed inside Facebook. At first, they were speaking to each other in plain old English. But then researchers realized they'd made a mistake in programming. Dhruv Batra, at Facebook AI Research" - that's an acronym, Facebook AI Research, FAIR - "is a visiting research scientist from Georgia Tech. He said: 'There was no reward for sticking with the English language as the AIs conversed.' The two AI agents were competing to get the best deal, which is an effective strategy for sharpening the operation of AI by pitting them against each other in what is known as a 'generative adversarial network.' In this case, neither was offered any incentive for speaking as a normal person would. So as they grew, they began to diverge from English, eventually rearranging legible words into seemingly nonsensical sentences, but sentences they each understood."

Batra, speaking to a now-predictable phenomenon that's been observed over and over and over, said: "AI agents will drift away from understandable language, inventing more efficient code words for themselves. So, we then wonder, should we let our software do the same thing? Should we allow AI to evolve its dialects for specific tasks that involve speaking to other AIs? To essentially gossip out of our earshot? Maybe. It offers us the possibility" - and this is coming from the text at FastCo Design - "a more interoperable world, a more perfect place where iPhones talk to refrigerators that talk to your car without a second thought. The tradeoff is that we, as humanity, would have no clue what those machines were actually saying to one another.

"Mike Lewis, who's a research scientist at FAIR, said that Facebook ultimately opted to require its negotiation bots to speak in plain old English. He wrote: 'Our interest was having bots who could talk to people.' And Facebook isn't alone in that perspective. Microsoft has also indicated that it's more focused on human-to-computer speech. Meanwhile Google, Amazon, and Apple are all also focusing incredible energies on developing conversational personalities for human consumption. They're the next wave of UI, like the mouse and keyboard for the AI era."

So another issue, as Facebook admits, is that it has no way of truly understanding any divergent computer language. Batra says: "It's important to remember there aren't bilingual speakers of AI and human language. We already don't generally understand how complex AIs think because we can't see inside their thought processes. Adding AI-to-AI conversations to this scenario would only make that problem worse."

So it's interesting. What we're finding is that, when we create sufficiently powerful AI, that inherently means that they have wide latitude and huge degrees of freedom. And when they're allowed to talk to each other, I mean, what's really freaky is this was -

remember "Colossus: The Forbin Project"?

Leo: Loved that movie.

Steve: That wonderful, it's like, from the '60s or '70s?

Leo: Yeah, yeah.

Steve: I mean, it was an old movie. I can't remember who the star was.

Leo: You're on the same page. Somebody in the chatroom says "new version of the Forbin Project." That's awesome, yeah.

Steve: Yeah, where Colossus determined that it had a counterpart in Russia, and it insisted that a communication link be established. And they began talking and quickly evolved their own language, which the scientists at each end were completely unable to understand. And they started with a mathematic basis and then evolved just completely out of anyone's ability to interpret it. And what's bizarre is it turns out that's exactly what happens. If we don't constrain two AIs that are flexible enough to have that capability, they will diverge and come up with a better language.

Leo: They're talking to one another right now.

[Clip in background]

MALE VOICE: "Colossus: The Forbin Project."

Leo: It was nobody famous in this. I don't think there's anybody whose name you would recognize today. But that was a great movie.

FEMALE VOICE: It's making you a prisoner.

Leo: Uh-oh.

MALE VOICE: Shock. Horror. Suspense. Created with all the technological brilliance of "2001: A Space Odyssey."

Leo: Not really.

MALE VOICE: Colossus is the ultimate in sophisticated computers.

MALE VOICE: I'm going to try to convince the computer that you're my mistress, and that that's why I have to be given the opportunity to see you regularly in private. That way we can...

Steve: That's one of the Bond actors.

Leo: Oh, is it? A bad - Bond bad guy?

Steve: Yeah.

Leo: No, really?

MALE VOICE: Four times a week.

Leo: So he's a prisoner of Colossus; right? Colossus apparently can speak, but when it talks to humans it has to type.

MALE VOICE: When do you think you'll be able to [indiscernible]?

MALE VOICE: Colossus sees all, senses all, knows all, controls all armaments and all defenses. When this emotionless creation becomes the master of man, the result is catastrophic.

[End Clip]

Leo: Well, now I'm going to - I know what I'm watching tonight.

Steve: It's a great movie, Leo.

Leo: I haven't seen it in ages.

Steve: I'm going to have to watch it again, too. I mean, it is old, but it was really well done. And in fact, earlier in this podcast, years ago, I mentioned that I had heard there was going to be a remake.

Leo: Oh.

Steve: And I don't know what happened to it, but I hope it happens because it would be...

Leo: It came out in the 1970s, and Eric Braeden is the guy's name. But I...

Steve: Oh, I thought he was - okay. Not famous.

Leo: Not famous.

Steve: But anyway, great movie. Great, great, great movie.

Leo: Yeah.

Steve: "Colossus: The Forbin Project." So over the weekend Apple pulled many of its iOS VPN clients from its Chinese App Store. And not unexpectedly, the makers of those VPN applications were up in arms over Apple's, quote, "capitulation," unquote, to pressure from the Chinese government, as they put it, claiming that this was a human rights issue, and were disappointed in Apple. But for what it's worth, I mean, yes, it would be nice if Chinese citizens had more freedom. But Apple is a super successful - actually, I don't know if you saw the numbers, Leo, but it just broke its records that just came out after MacBreak Weekly. China is Apple's largest market outside the U.S.

Apple, as we know, is a publicly held commercial for-profit company whose device application model is that of a closed, customer-protecting, application ecosystem. And so, yeah, that means there's a tradeoff. If you want an open ecosystem, there are plenty of them freely available. Get a PC or jailbreak an Android phone and sideload whatever you want and take your chances.

Leo: That's a good point. You don't have to use an iPhone. That's a good point, yeah.

Steve: No. Yeah. Apple knows quite well that, if they're going to operate in China, they can do so only with the permission and approval of the Chinese government. And if I were in China, I would definitely want the significantly enhanced security of using an Apple iOS device rather than any other non-Apple solution, knowing that there is extreme curation of the apps that are allowed to run on that device. So that would be my choice. So, yes, it's closed; but it's also closed as much as possible to malware. And for the moment we also believe it's likely closed to eavesdropping. So it's like, yeah, again, you could get a - if you wanted to do VPN-enabled surfing, just get a cheap Android phone and use that.

Leo: Or a PC, although China's also banned VPN apps on PCs. VPN services, I should say. So you can use one, but it's run by the Chinese government, which tells me that it's not completely secure.

Steve: Yeah. So iRobot did walk back their plans to sell their users' homes' floor plans, following a bit of a firestorm of customer...

Leo: They said they never intended to do it. Not walked it back, but they said they never intended to do it.

Steve: I know. Following a bit of a customer firestorm of outrage over the news, iRobot's CEO Colin Angle now says that his comments to Reuters were misinterpreted, and that iRobot has no plans and would never make such data available to third parties. So in seeing that, I thought, I wonder, you know, I was kind of curious, wondering what was misinterpreted. So I went back to our coverage last week and found Colin's quote from Reuters, where he said: "There's an entire ecosystem of things and services that the smart home can deliver once you have a rich map of the home that the user has allowed to be shared." So, okay.

Leo: Yeah, that doesn't say anything about selling it. And they said we intended - if we share it, we'll share it with other IoT, what was it, IoT devices. They changed, well, I don't think they changed anything. I don't think what they said was incompatible with what he says.

Steve: I agree. So maybe it was just - okay. So in the worst, what we got was a clarification and a clear statement. And my sense was that Colin may have just sort of been winging it when he was talking to Reuters and just sort of said, oh, yeah, you know, they might have said, "So, what plans do you have for additional revenue?" He might have felt a little challenged by that and just spit out something that he quickly decided, like, oops, you know, that's really not any firm plans we have. And he also did say, "We have never had any discussions with any third parties about map sharing." So if he gave them the wrong impression, it was probably just something that he said during the interview which he then decided was not such a good idea. And if he was toying with the idea, I think it's clear what his users think about it.

A couple Google Chrome browser improvements. Chrome will be, with Chrome 63, which is due in December of this year, so by the end of 2017, they are tightening the screws on embedded iframes, which is really good news. Iframe is short for inline frame, which is essentially - it's a feature that's always been present in HTTP, or I should say in HTML, which allows a web page to embed essentially another web page inside of itself, that is, in a frame. So the hosting page defines a rectangular region and then puts a URL in that frame which the web browser, after loading the hosting page and seeing that an iframe has been defined, the web browser says, oh, I need to now go get the content to fill that frame, which itself is a complete HTML page from somewhere else, that is, it doesn't have to be - it can be from the same domain, but cross-origin iframes are allowed. So this has historically been an outsized vector for malware and browser exploitation that we've often talked about on the podcast.

So beginning at the end of this year, with Chrome 63, iframes will be restricted by default in what content they may contain, with restricted permissions needing to be then explicitly allowed. There will be a new term added to the iframe clause. And I didn't check to see if this was W3C already, that is, is it part of the spec which Chrome will be adopting? Or which way is this flowing? I don't know. But I imagine, if not already, it will become a standard. There will be an "allow" term added to the iframe link where the values can be geolocation, microphone, camera, speakers, MIDI, or encrypted media.

And if any of those are not available, their respective content type will not be provided, will not be allowed to be presented to the user through the iframe. So they will be denied

by default, and embedding pages will need to explicitly enable them. So that's perfect because iframes are one of the ways that ads are hosted. And, for example, it's the way ads are allowed to run Flash content by default is typically in a frame and, clearly, a way that some other third-party site could turn on the camera or could turn on the microphone or could annoy you with noise from MIDI or from speakers. So those things will be shut down by default. And if this becomes a standard, if it's not already, then that's a nice step forward in our security.

And the second piece of good news on the Chrome front, we talked some time ago about how some changes in Chrome had made it surprisingly difficult to inspect the certificate of a site that you were visiting. We know that the "chrome" in Chrome, as it's called because by tradition all of the window dressing around the page is called the "browser chrome," which no doubt is where Chrome got the name for itself, they had changed - Google had changed it in Chrome, or the Chromium Project had, to make it more difficult to find and inspect the certificate. We talked about this at the time.

Well, they've changed it to make it very easy. It is still opt-in. But if anyone with Chrome goes to - and there's a link which is worth reading out because it's not easy to find. If you just go to chrome://flags, that will take you to a page with, I mean, it's as bad as Firefox's about:config. I mean, there is so much stuff there, it's like, yikes. Lots of flags. Anyway, the one you're looking for is at - so //flags/#show-cert-link. It's down near the bottom, so you could also scroll all the way down to the bottom and then up like about a page, and you'll find show-cert-link, and then it's just a little tiny link there that will say "enable." And so you want to click that, and then you'll be prompted to reload the browser. You need to reload the browser. And once you do that, when you then click on the padlock shown in the URL bar, that will drop open. Right now, without having done that, it drops open a long list of characteristics of the site you're visiting. Once you have enabled the show-cert-link, the first thing...

Leo: This is only in Chrome 60, I've got to point out. Most people...

Steve: Oh, yeah, yeah, I forgot to say, yes, thank you.

Leo: If you don't have Canary, you don't have it.

Steve: Which is - and I did update on my Win7 machine in order to see exactly what it was doing, in order to explain this. And so Chrome 60 is the current one. So if you go to About Chrome, and it sees you're on 59, it'll give you an update right then. So I think you should be able to get that. Anyway, the good news is, once you've done this, the first thing on that dropdown list is a button to allow you to inspect the certificate, which now makes it super easy to do so. So yay to Chrome for giving us that tweak. And again, that's a power user feature. We know that most people probably won't care. But for those of us who do, having it just right there, two clicks away, is extra nice.

So WikiLeaks' 18th Vault 7 dump contained news of three new tools that the CIA uses for hacking and implanting stuff on Mac OS X and Linux - the first two for Mac OS, the third one for Linux. Achilles is a tool used to subvert Mac OS X disk images. It allows CIA operators to combine their own malicious trojan apps with a legitimate Mac OS app into a hybrid disk image .DMG file. The tool which binds these pieces together is just a BASH shell script, which gives the operators opportunity to run the appended malware as desired.

And what's interesting is, when the unsuspecting targeted user downloads an infected disk image on their Apple computer, opens and installs the software, the malicious executables run in the background. But Achilles then detaches itself so that afterwards all traces of the Achilles tool are removed securely from the downloaded .DMG file. So the file then is a checksum match. It is a perfect match for the original unmodified .DMG. So if somebody didn't check before, for example, checksum verification, but thought, oh, maybe I should check afterwards, well, it'll match, and it's too late because Achilles will have already delivered its payload and removed itself from its install image.

So this is not of any particular high-tech-ness interest. It's just some interesting technique that the CIA, presuming that these documents are legitimate and were in fact leaked, as we believe, it's informative that this sort of targeted attacking is going on, and that Mac OS X is the intended target.

A second tool, SeaPea, P-E-A, is a stealth rootkit for Mac OS X systems; and, as such, like any rootkit, it would provide CIA operators with a stealthy tool with launching capabilities that hides important files, processes, and socket connections from users so that even somebody looking for any debris in their system, trying to audit their connections through netstat to see what their network system is doing, it cleans up all of those presentations. We talked about rootkits years ago in great detail because there were a lot of them around at the time.

So SeaPea requires transient root access for it to be installed on a target Mac, making it, again, not leveraging any exploits or zero days or anything, but being a targeted attack tool. And it will then remain resident and cannot be removed until the next version of the Mac OS is updated, which would flush it out because it would just remove it from the system. But so a stealthy rootkit targeted at Mac OS X that, if all of this is to be believed, the CIA is able to arrange to get into someone's machine where they need access, and then it hides itself.

And finally, Aeris, A-E-R-I-S, is an automated implant targeting Linux machines. It's written in C, designed to open backdoors in not only Linux, but some Unix systems. In Linux it's able to operate on Debian, CentOS, and Red Hat; and then also both FreeBSD Unix and Solaris Unix OSes. It provides build scripts which allow CIA operators to generate customized effects, depending upon their needs; supports automated file exfiltration, reconfigurable beacon interval and jitter, HTTPS and SMTP protocol; and, I got a kick out of this, all with TLS encrypted communications with mutual endpoint authentication, providing fully secure end-to-end encrypted exfiltration with a structured command-and-control system that's similar to what is used by several Windows implants.

Leo: So it sounds like these would be used, if you got access to a network, and you were sitting at your console somewhere else, that you would then install these onto those systems, exfiltrate, delete them, and get out. You're not going to leave them there for days and weeks and months. They'd be discovered. This sounds like more this is like the kind of tool that's used once you've hacked a system, right, hacked into a network.

Steve: Yeah. I guess the idea is that they're not leveraging any mysterious problems.

Leo: Right. These are just tools.

Steve: Exactly. The rootkit would remain hidden, so nobody looking for it would be able to see it.

Leo: Right. So SeaPea might be more - but you have to have root access to put it on in the first place.

Steve: Correct. And you would see its traffic. So if a lot was going on, then if you were monitoring traffic, then that would expose it to external surveillance. But I think what we're seeing here is deliberately targeted attacks.

Leo: Right.

Steve: So they're, as I described them, workmanlike tools which are in this toolkit. And so if there's someplace where the CIA has, like, brief access or temporary access or maybe gets somebody who is trusted by the system owner to, like, insert this USB drive and do us a favor on behalf of the U.S. government, then that could happen. So again, the 18th release of WikiLeaks goodies.

And I did find a very nice blog posting, actually, so sort of refers to me and SpinRite and the podcast, with a little, you know, not written directly to me. He called it a "Universal Fix for Windows KSOD." And I didn't know what that was. I understand it was black rather than blue, so he used the "K" of "black" rather than, you know, because both "blue" and "black" start with a "B." So it's not the Blue Screen of Death, it's the Black Screen of Death, but KSOD.

He said: "Ever had your Windows installation inexplicably die, leaving your computer unusable without a fix?" He said: "I have, more times than I'd like to count. The last time this happened was yesterday, when Windows 7 would not boot" - I'm sorry - "when Windows 7 would only boot into a black screen with a movable cursor, also known as the blacK Screen Of Death." He wrote: "It was a serious case, considering none of the safe modes or repair functions in the Windows boot options would work. Each option would universally end in either a blacK Screen of Death or the classic Blue Screen of Death after hanging on aswRvrt.sys during safe boot. After exhaustively eliminating all possible regular fixes that were available on the Internet, I decided it was time for the big guns: Steve Gibson's SpinRite.

"Prior to trying SpinRite, I first tried Kaspersky's Rescue Disk 10, which was entirely useless," he wrote, "for my case. After booting from the rescue USB dongle, I would always get a 'Missing Operating System' error in the boot screen. Not reassuring." He said: "I have long been a fan of Steve Gibson's Security Now! podcast, which is why I knew of the tool. I knew the tool would be one of the few things that might do the trick, so I gave it a shot. After an hour running SpinRite 6, and a few reboots later, my Windows 7 installation was working perfectly, as if nothing had ever happened. SpinRite saved the day."

And then he said: "TL;DR." He said: "SpinRite saved my machine from a perpetual and otherwise unbeatable KSOD scenario. And my guess is that, if you are having KSOD problems, then SpinRite is one of the few things that might help you, too." So, wow. Thanks for the nice blog post.

Leo: Very nice, yeah. Okey-dokey, Steve. What else do we want to talk about today?

Steve: So we've got a couple of closing-the-loop items.

Leo: All right.

Steve: From our terrific listeners. My mention of S3 and how I get a $2.83...

Leo: Speaking of Amazon services, as a matter of fact.

Steve: Yes. Got a lot of interest from our listeners. Logan Rogers tweeted: "Are you just using S3 Bucket from @awscloud for your backup that you talk about on Security Now!? Looks like cheap cloud backup." And then Ugly Bob said: "I'm curious as to what software you use to back up to AWS?"

Leo: We used to use Jungle Disk. Remember the good old days?

Steve: Yup, yup. And in fact my bookkeeper's machine is still being backed up by Jungle Disk. I get a report by email on the weekend of the...

Leo: So they're still around? Or you're just using the old...

Steve: Yup.

Leo: Oh.

Steve: Yeah, I think Rackspace bought them.

Leo: Oh, that's right.

Steve: And I had a perpetual license from the beginning, and they're still honoring it.

Leo: Oh, that's neat. They're still around.

Steve: So I'm very impressed, yeah. So what I use is a, naturally, a command-line utility. It's at s3.codeplex.com, and it's just called S3. And it is a wonderful little interface to Amazon's S3 and EC2 web services. The description at CodePlex - by the way, CodePlex is going away. So if anyone is interested, you might want to go to

s3.codeplex.com and grab it. It describes itself as a "Windows command-line utility for Amazon's S3 and EC2 web services that requires no installation." It's a single .EXE file.

Leo: Your kind of software.

Steve: Yup, with no DLLs. It only requires .NET 2, so will work on a plain vanilla Windows 2003 installation or, what was that, Windows 7 was the parallel non-server version. So it just runs. And actually I'm running it on XP, so it works there, too. Although I probably have .NET added to it.

Under features they said: Efficiently uploads and downloads large numbers of files, or whole directories, between Amazon S3 and Windows PCs. Everything in one .EXE. Nothing to install or configure. Just download it where it's needed and run. It doesn't require anything except .NET 2.0, which you probably already have on our machines. Works well in an automated backup solution or as an ad-hoc system admin tool. Can split large files into chunks for upload without creating any temporary files on disk. Fast parallel file transfers. Actually, it says "(Coming Soon)," and I wouldn't hold my breath because I think 2010 was the last version of it. Can use HTTP HEAD command to quickly determine which files don't need to be uploaded because they haven't been updated using the /sync option. Support for EC2 operations. Free and open source, and nothing to pay. No paid versions.

So what I do is I have a batch file. And when I download the audio and recompress it for Elaine, I simply type "send," space, and then "sn-622" is what I'll be doing. And that batch file both sends a copy to GRC and also sends a copy to Amazon. So I'm just, you know, it goes off, and it's trouble-free. Anyway, there was so much interest shown by our listeners that I wanted to let them know what I'm using.

There's also, for Firefox users, and I don't know about Chrome, there's a really neat S3 add-on called Open S3Fox. And that is a full S3 bucket browser which you're able to configure that allows you to use a web page like an app and poke around within your S3 buckets. So it works great for me. And as I said, boy, is it inexpensive.

Steve Whisenhant said: "I drive Uber, and I'm considering enabling hotspot on my phone as a perk for riders. What are the risks to me, assuming WPA2 and password posted in the car?" So, boy. I would be worried because we know that there is, for example, on an iOS device, I trust Apple, I trust their security, but they do have protocols that bridge the phone's iOS to WiFi-connected iOS devices. Again, I'm sure Apple has done everything they can to make it secure. But the idea of deliberately allowing unknown riders to essentially attach to your phone's tether, WiFi tether, I mean, I get what a bonus it would be to riders. But, boy, that would make me nervous.

The alternative, of course, is to get a cellular standalone hotspot. PC mag did some coverage at the beginning, or earlier this year, titled "The Best Mobile Hotspots of 2017," where they talk about a range of them and have in-depth reviews and so forth. The downside, of course, is then you have to purchase that and then pay for a service separate from your phone. The upside is there's no connection to your phone. So I'm trying to think whether you could isolate. It might be possible to use one of the mobile WiFi repeaters. We've talked about them before, like you use in a hotel, where you use the hotel's WiFi, but then it creates an access point to your devices and is a NAT router. That would at least provide some level of security, although you're not really providing NAT routing protection in the direction you want because you'd want your phone to be behind the NAT, rather than your customers or your riders behind the NAT.

So, Steve, I don't think there's a clean answer. Again, I would trust Apple to have made this secure. But if there was a mistake that was made, you're potentially allowing someone to hook to your phone. Probably not a big deal. Probably not worth worrying about. But at least maybe worth being aware of.

Also I got a kick out of this. Someone whose Twitter name is Trust No-One...

Leo: Great name.

Steve: @twust. He said: "@LastPass two-factor authentication is only as strong as a cleartext link emailed to you when you click 'I lost my two-factor authentication.'"

Leo: This always worries me about two-factor because, no matter what you have, there's always, "Did you lose your device?" And then there's backup processes. And some of these just aren't very secure; right?

Steve: Exactly. I mean, my notes...

Leo: Which means that a bad guy will use that.

Steve: Well, exactly. And so what I wrote here was the important lesson here is the weakest link principle: No security system is stronger than its weakest link, as we've often said. So, consequently, no authentication system can be stronger than the strength of what's required to bypass it. And of course, as I've said, this is the tradeoff that SQRL makes deliberately in a different direction.

And as it's getting close to happening - I'm shaking out the final details of the install, update, and uninstall systems with the guys online - I'm recognizing that it's not ever going to replace usernames and passwords. I doubt that it will because it does make that tradeoff. It says, if you're willing to be ultimately responsible, you can have ultimate security. But I'm sure there are a lot of people who don't want - who the idea of having no other recourse except what SQRL provides, and when you and I, Leo, cover this in detail, which we will when it's available for everyone to play with, everyone will see that I've gone to tremendous lengths to create all kinds of recovery systems.

But ultimately, ultimately it's up to the individual to be somewhat responsible. And I think some people won't bother. They're just like, eh, I'd rather have someone I can call and cry to. So again, not for everybody. But it would be cool if it were an option because when used it, as far as we know, cannot be bypassed. And in fact you're even able, once you're comfortable with it, to set some switches, send a beacon to the sites you then visit saying, "I want you to please disable alternative login and all account recovery." So if this all happened, then there's just nothing a bad guy could do to get you. So I think it's worth having this out there as an example of how it can be done. And maybe it'll become more than an example.

Leo: Well, and I should point out that, at least on LastPass, you get to choose what

backups you want, what multifactor authentication systems are allowed.

Steve: Nice.

Leo: And so, you know, I have a YubiKey as a default, but I've disabled everything except YubiKey and Google Authenticator. So I would presume...

Steve: That's perfect.

Leo: Yeah. If I lost my YubiKey, I'd have a backup; but it would be something secure as opposed to email.

Steve: Yeah, and it's worth - it's not something that we all do. But our listeners might pretend to, like, not know their authentication code and try to perform a reset, how difficult it is, because that's what a bad guy is going to do.

Leo: Right. Yeah, and some of these systems are terrible. I mean, they're just really terrible. PayPal's is awful.

Steve: Uh-huh. Well, I know.

Leo: So it's just - I can't remember what it was because I have the dongle; right? But then when you get to...

Steve: Well, they disabled it. They stopped supporting...

Leo: Trying to remember what it was.

Steve: ...the token and began using SMS. It's like, uh, what? Because it used to be expensive. They were doing it through VeriSign, which charged them per authentication. It was free to end-users, but not to the companies who were using the service.

Leo: I've seen, it's probably not PayPal, but I've seen some services use secret questions as their fallback.

Steve: Ugh, yeah.

Leo: Which means the secret questions are the most - that's the weak link.

Steve: Yup.

Leo: And that's a pretty weak link.

Steve: Yup. So Rex Moncrief cited an article in ZDNet which was titled "This Android Spyware Can Record Calls, Take Snapshots, and Video Targets." And then it also says "Gmail, LinkedIn, Snapchat." So he sent the link and asked the question: "Any feedback on now this 'retrieves' data from Threema?" And Threema was among the apps that this malware is reputed to be able to compromise. And the answer, of course, is it does it before it's encrypted. And I thought - I liked the question because it reminds us that, as strong as end-to-end encryption can be - and I believe Threema is ultimately strong because it gives the user responsibility for validating the identity of the endpoints, which you have to have. If you don't have that, then you're trusting, inherently trusting an authentication mechanism other than yourself.

But the point is both endpoints have to be uncompromised. So it's the man in the middle, it's the traffic interception that is being prevented with encryption and authentication; but not the person typing into a keyboard where there's been a hook placed in the keyboard, monitoring the keystrokes that the person is then securely encrypting before it goes over the air. So again, if you compromise the endpoints, all bets are off. It doesn't matter how secure your link is because you're able to grab the data before it happened. And that's what this malware does, which of course is then able to compromise everything. And it has specific hooks for specific applications, which is why there were some that were enumerated.

And, finally, Chris Shaw asked, I thought, a great question, which has never come up. He asks: "Does printing a PDF to a PDF strip malicious content from the file?" And it does. It hadn't occurred to me. But that is a great way. It neuters all of the active stuff because what you're printing is a rendered image of the PDF, which is then described as a new PDF of that image. But going through the printer rendering stage, very much sort of like I was describing with the voting machines, where we forced a standardized paper trail to link both ends, this is sort of similar. All the form content, any macros, any funny business that might have existed in the first PDF is flattened into a simple visual description of what is seen on the page. That's what the printer renders and then redescribes in a PDF. So great question, Chris. And, yeah, that's one way.

In fact, I do it all the time. I own a copy of Adobe's full Acrobat. And sometimes I'll, like, want to remove some content from a PDF or highlight something. And I've noticed that, when I redisplay the page, the original content flashes briefly, or it takes a while for the highlights to appear because they're overlaid, you know, they're being rendered on top of the underlying PDF. Which annoys me. So I will then print my marked-up PDF, which creates a PDF that is now solid. It no longer has descriptions of those individual things. It describes the final result, which then displays instantly with no secondary render due to the history of the previous PDF's markup. So same kind of idea. So again, great question, Chris.

Leo: I am going to do a follow-up. I didn't want to do a follow-up on this PayPal thing till I'd fixed it.

Steve: Ah. Okay.

Leo: So if you have misplaced your PayPal security key or two-factor authenticator, they want to do it via, as you pointed out, SMS. You said, "I forgot." It said, okay, well, now you have a couple of choices. We can call you, or you can answer these security questions, which are mother's maiden name and last four digits of your social.

Steve: [Sighing]

Leo: Now, that was the default. I'd obviously never looked at this. So I have changed it to other security questions and nonsense answers.

Steve: Other data.

Leo: But I'm glad I did that, and I invite everybody to do that. If you're assuming, oh, PayPal's secure because I've got a password and two-factor, look at the fall-through. And it wouldn't just be PayPal, be everybody else. Look at the fall-through. And, wow, now I have some homework to do. But, wow, that was not a good fall-through. I'm lucky I didn't get hacked.

Steve: No. Publicly available information.

Leo: Yeah. And that was - I don't believe I'd set that up. I'm not stupid. And even whenever I set this up, 15 years ago, I wouldn't have used mother's maiden name and last four digits of the social. I would have come up with - they do insist on security questions. But I would have done what I recommend everybody do, which is answer them with nonsense.

Steve: Exactly.

Leo: Another passphrase which you store in LastPass.

Steve: Yeah, and unfortunately don't use 0000 as your last four digits of the social because that's what the bad guy will test.

Leo: Yeah, right away, yeah.

Steve: So they don't even need the information, then.

Leo: Because I don't think I would have, yeah, I'm pretty sure that's the default.

Steve: Has to be.

Leo: So check your PayPal accounts. And you might want to, if you haven't ever set up secret questions, you might want to do so. Holy cow.

Steve: Good. Good tip.

Leo: Okay. Yeah, and I didn't want to say that until I'd fixed it.

Steve: Cool.

Leo: All right.

Steve: And that's our podcast, my friend.

Leo: Yes, sir. We do this show at 1:30 Pacific, 4:30 Eastern, 20:30 UTC on Tuesdays. And you can stop by and join us. Erica came all the way from Chicago to sit in the studio and watch this show. Thank you for being here, Erica.

Steve: Yay, cool.

Leo: You can do that by emailing tickets at TWiT.tv. We do this show in my office, which has only about four or five seats. So that's really important, if you want to come to a show that's done out of my office, to email tickets@twit.tv. We have a lot more room in the big studio, but...

Steve: Standing-room only.

Leo: We have had people stand. Or sit at my feet. Which is fun. If you want to watch live, you can go to TWiT.tv/live and watch the live stream. And if you do that, please join us in the chatroom. Great people in there. Well, mostly. All you have to do is going to irc.twit.tv. You can use a browser, or I figure if you listen to this show you probably have an IRC client lying around. Irc.twit.tv, we'd love to have you in there.

For everybody who has a job, and their job is not 9:00 to 5:00 Monday through Friday, or is Monday through Friday 9:00 to 5:00, you might want to then download an episode. You can listen at your leisure. Steve's got not only audio, the most popular format, but he also has transcriptions you can read as you listen. And for some people that's how they learn. Megan was just saying she learns by reading. That's how you do it. Go to GRC.com. While you're there, pick up SpinRite, the world's best hard drive recovery and maintenance utility. Take a look at the other freebies Steve gives away out of the goodness of his own heart. Keep up on what's going on with SQRL, lots of other things. You can also get copies of audio. And we even have video, for reasons no one understands, here at TWiT.tv/sn.

Steve: You're forward-looking, Leo, so video.

Leo: Well, it seemed like a good idea at the time. You resisted it, and you were probably right. But now you know I'm wearing a tie, so there you go. You can also subscribe. That's the best thing to do. We have audio and video subscriptions available. Wherever you get your podcasts, just look for Security Now!. And we will be back here next week talking about probably the most important topics we do anywhere on TWiT, your privacy and security. Thanks, Steve.

Steve: Thanks, my friend. See you next week. Bye.