SECURITY NOW!

Transcript of Episode #702

Authenticity on the Internet

Description: This week we catch up with last week's doozy of a Patch Tuesday for both Microsoft and Adobe. We also examine an interesting twist coming to Windows 7 and Server 2008 security updates, eight mining apps pulled from the Windows Store, another positive security initiative from Google, electric scooters being hacked, more chipping away at Tor's privacy guarantees, a year and a half after Equifax and where's the data?, the beginnings of GDPR-like legislation for the U.S., and some closing-the-loop feedback from our terrific listeners. Then we take a look at an extremely concerning new and emerging threat for the Internet.

High quality (64 kbps) mp3 audio file URL: http://media.GRC.com/sn/SN-702.mp3
Quarter size (16 kbps) mp3 audio file URL: http://media.GRC.com/sn/sn-702-lq.mp3

SHOW TEASE: It's time for Security Now!. Steve Gibson is here. He's got the most scary AI story ever, he says. We'll talk about that. Plus Microsoft's Patch Tuesday was last week. Why was it so big? And why you might want to be more careful the next time you ride a rent-a-scooter. Turns out they're pretty hackable. It's all coming up next on Security Now!.

Leo Laporte: This is Security Now! with Steve Gibson, Episode 702, recorded Tuesday, February 19th, 2019: Authenticity on the Internet.

It's time for Security Now!, the show where we cover your security and privacy online with this guy right here, our commander in chief, Mr. Steve Gibson of the Gibson Research Corporation. Hello, Steve.

Steve Gibson: Yo, Leo. Great to be with you for Episode 702.

Leo: Yow.

Steve: We have a bunch of stuff to talk about. And I was just saying to you before we began recording that one of the things we'll talk about is the beginnings of GDPR-like legislation heading into the U.S. as a consequence of a GAO report that was commissioned by Congress two years ago. And I had that as the topic of the show. I think it was "GDPR for USofA" was what I was going to title this.

And then I ran across a blog post from the OpenAI group in Northern California. OpenAI is an Elon Musk-founded organization. And it chilled me to the bone so that I thought, okay, stop the presses. I renamed this podcast "Authenticity on the Internet" because -

and the good news is they understand how scary this is, too. As I was reading through what I will be sharing with our listeners, I was thinking, oh my god, oh my god, oh my god. And it's like no, no, no, no, no.

And then, as I got down toward the end, they addressed what they recognized they had created and what it would mean if it got loose. So much so that the various tech press picked up on that and said, "Something so scary they're not even letting it go." The problem is it doesn't matter. I mean, it's going to change the world, and not for the better. But we'll get to that in a minute.

We're going to catch up with last week's doozy of a Patch Tuesday, which was a doozy for both Microsoft and Adobe. We are going to examine an interesting twist coming to Windows 7 and Server 2008 - which is, you know, Server 2008 R2 is the equivalent matched version with Windows 7 - and their security updates. Mary Jo wrote about it in ZDNet, and I haven't quite figured out one aspect of it that we'll be finding out about in a couple months.

Eight mining apps were pulled from the Windows Store. Another really interesting positive security initiative from Google. Electric scooters being hacked. Who would have guessed? I mean, like, of course. And wait till you hear the response from the company. More chipping away at Tor's privacy guarantees through some new research. We're a year and a half downstream from Equifax's big breach. Where did the data go? There's been no sign of it. So we'll talk about that. And as I mentioned, the beginnings of some GDPR-like legislation which of course the EU famously has, and now it looks like we may be getting something like that, and sort of apropos of where we are in time.

We've got some closing-the-loop feedback from our listeners. And then we've got to talk about this sort of astonishing work that the OpenAI project did. And we have a great Picture of the Week and some fun things to talk about relative to old-school computing. So lots of stuff.

Leo: Busy, busy, busy.

Steve: Yeah.

Leo: Yeah. And we had it "Authentication on the Internet." It's "Authenticity on the Internet" is our topic of the day. And I've modified that lower third to reflect that.

Steve: So we should talk about the little unit you have behind you.

Leo: Oh, Oscar Vermeulen is awesome. This all started with Steve and his blinking lights right over his left shoulder there. Those are - what's that, PDP-8; right?

Steve: PDP-8, which was my first computer. That's what I encountered when I was in high school. One of the teachers in the math resource center said, "Hey, Steve, I think there's a company over in San Carlos that you might want to check out." And this was, I mean, this was when Bill Gates was at a school...

Leo: He was in high school.

Steve: ...with more money than mine. Yeah, we were both in high school. He had a computer in his school. We had no computers in our school. This was 1971. But so it was just beginning to happen. Yet this company, Technica Education Corporation, they were in the business of beginning to put little timesharing systems in schools, in elementary and high schools, and beginning to teach programming. So they had PDP-8s and also little HP mini computers. Anyway, it was where I first encountered an actual computer and learned assembly language. And as all our listeners know, I just stuck with that because, if it's not broke...

Leo: It worked pretty good, yeah.

Steve: It worked really well. So, yeah, so the machines behind me are PDP-8s, which is a 12-bit machine. And DEC sold - Digital Equipment Corporation - sold a bunch of them, I mean, like really. It was a very popular solution because they were inexpensive. And at the time, you needed things to run laboratory equipment and collect data and do little databases. And everybody was kind of a programmer because there was no readymade software, really. It was just, you know, "Here's your computer. Good luck." And it came with manuals. Like here's the instruction set. And so it was, you know, a very different era back then. But that thing, even though there was an operating system, OS/8, created for it, it came with 4K - 4K - of core memory.

Leo: Because it was expensive back then.

Steve: Oh, my goodness, yes.

Leo: You didn't put in unnecessary memory. It costs a lot.

Steve: No, 4K. Well, and 12 bits can only access 4K.

Leo: Oh.

Steve: Remember that 10 bits is 1,000, well, 10 bits is 1,024; 12 bits is 4,096. That's all you can access is 4K. Now, that was such a problem that they did sort of the equivalent of what Microsoft, and you remember the EMS deal where you could add some additional - they had a banking system, as in memory banks, which allowed you to have an additional three bits for instructions, an additional three bits for data. So you could have your instruction - so the instruction pointer was still only 12 bits, so 4K. But it sort of extended in a bank an additional three bits that brought you all the way up, Leo, to 32K.

Leo: Whoa.

Steve: So, oh my god, what could you ever - you're never going to use all that much memory.

Leo: Well, not if you keep writing in assembly language. You probably wouldn't.

Steve: Well, and you could do like real work with four or 8K of memory. Anyway, so the problem was it really was a little underpowered. The machine that put DEC on the map - they sold a ton of PDP-8s. But it was the PDP-11. That was a 16-bit machine. And these are all, you know, we're all used to hex now, which is to say grouping our binary bits in groups of four bits. So you have zero through nine and then A through F.

Well, these machines were all octal. We hadn't figured out yet that octal was wrong. And so everything was zero to seven and grouping in sets of three. So that worked well for a 12-bit machine because you just had four groups of three. It doesn't work so well for a 16-bit machine because you could do five groups of three, but that's only 15 bits. So you end up with like the first, the top one being either a zero or a one. But still it was an octal-based machine. That's the PDP-11, which is what you and I both have. Here's mine, and you've got yours.

Leo: Here's mine, yeah. And this is thanks to a crazy Swiss, Oscar Vermeulen; right?

Steve: Yup, yup.

Leo: He's the guy who did your replicas that are behind you, and now he's moved on.

Steve: Right. So as we were sort of talking about this when we were talking about it before, it's actually based on a Raspberry Pi, which gives it more memory than any PDP-11 ever dreamed of having. So the console is like an I/O peripheral. I mean, it's got the switch banks and the blinking lights. And so back then, back in the day, in order to actually load it with software - there, you can see a picture with the various lights lit up on the console.

Back then, these machines had core. There were later 11s that had RAM in them, dynamic memory. But the original ones had core memory. So you'd flip the switches up and down to select a memory location, and then change them again, and then press or actually raise the deposit switch, which would write those bits into that address in memory. And it would increment the address pointer, the address counter, so that then the next deposit would automatically go into the succeeding address.

And you would put, like, maybe 10 instructions is what you needed in order to create the tiniest program, which was known as the "bootstrap loader," which would read from a paper tape on the KSR-33 or 35. And so you would start the program. Then you'd put the tape in, and it would just go [sound effects] to, like, suck this tape in. And basically all it was doing was reading from the paper tape and storing much more into memory than you could manually toggle in. So the so-called "bootstrap" was just the barest smallest program that you would put in.

And many of the actual computers at the time, actually most of us memorized the bootstrap because we were having to toggle it in all the time. Many of the actual machines you would see had them either handwritten or typed on a little piece of paper on the front panel for people that were a little lame and needed a crib sheet in order to remember the instructions, in order to key them in. But so that little program would then bring in the next bigger program.

And in fact there was also sometimes a three-stage because that second program was still not very sophisticated. So you'd read that in, and then that would bring in the paper tape loader, and then you'd run that in order to load the operating system. And it could take two hours to load the operating system, literally feeding in massive rolls or fanfold tape in order to get this thing loaded. The good news is, is it was nonvolatile memory. So as long as the system didn't crash, or no one tripped over the cord while it was running, you could generally keep the OS loaded for weeks at a time before you'd have to go and load something again. So anyway, the point is that we have little emulations. Basically it's an emulated PDP-11. And as I mentioned before, Oscar recorded what - did you plug it in?

Leo: Yeah, and I turned it on. And, look, I've got lights.

Steve: Oh, and they're moving.

Leo: And by the way, my key works.

Steve: Nice.

Leo: Yeah. I can turn it off again. Or maybe it only works once. Anyway, that's pretty cool. Yeah, they're moving lights.

Steve: Ah, very cool.

Leo: Yeah, I don't even have to program in some blinking lights. He says: "Think of it as either a very expensive front panel for a Raspberry Pi or your very own mini" - it's running an emulator; right? So it's actually running a PDP-11, as if it were a PDP-11. You can log into it by SSH or...

Steve: Yeah, there is a whole, I mean, well, and it was on the PDP-11 that Unix was born. That's, I mean, Unix first ran on a PDP-11. It was written first in assembly language. And then of course they thought, okay, this is crazy. We need a higher level language. And so that's where C was then, again, the first C compiler was written in PDP-11 assembly language to compile C. And then they wrote it in itself, and it compiled itself somehow.

Leo: I'm just going to leave it right here. This is all I needed to do. That's perfect.

Steve: Yes.

Leo: It's perfect.

Steve: Very nice.

Leo: It's got some demo running. It's awesome.

Steve: Very nice.

Leo: Really nice. And thank you, Oscar, for sending these along. And once again, if you are interested, Oscar has made already over a thousand of these, which is kind of mindboggling. I know a lot of our listeners...

Steve: Leo, $249.

Leo: It's very affordable. I mean, that's great.

Steve: It's crazy.

Leo: If you've got a Raspberry Pi, Raspberry Pi not included, but they're 35 bucks. You've got them lying around. I did.

Steve: Right, right.

Leo: It's a simple thing. Obsolescence. I would say, if you googled "obsolescence guaranteed," or even "PiDP-11" because it's a Pi DP, you'd probably find it right away. If he had not built it at Wix, he would be easier to find, but okay. So there you go.

Steve: And just so our listeners know, he is now heading toward a PDP-10.

Leo: Yes.

Steve: Which was the granddaddy of the DEC machines. That was a 36-bit machine, and it used kind of weird yellow and green coloring. Whereas this is sort of two-tone, like burgundy and red.

Leo: You can see the influence of the LCARS interface on the Star Trek stuff.

Steve: Yeah.

Leo: I mean, this is awesome. So I have the source code for Unix. Could I just enter it all in?

Steve: It's probably already in there. I mean, it might be - I don't know whether it's running Unix or running - but there is for the Raspberry Pi all of the DEC software. So it's a full PDP-11 emulator. And you could run, right in PDP-11, assembly language, or they've got compilers and Fortran and C and everything. So watch out. It is just sort of a blast from the past.

Leo: There's a whole manual which he sent. And both, let's see, you can hook it up - oh, there's a PiDP-11 Google Group. He says there's a few of the old-timers in there, some of the designers of the PDP-11. He did a lot of software archeology, he calls it, to get this working. And yes, it comes running a PDP emulator on the Raspberry Pi. So that's pretty sweet. And I love - this is fine. The demo program...

Steve: It's done.

Leo: It's done. I don't need to enter in any blinking lights. I got the blinking lights. I'm going to leave it right there in front of you.

Steve: So when you saw the Picture of the Week, you busted up because...

Leo: Oh, I had to laugh.

Steve: And I mentioned this before on the podcast. It's one of my most enduring pet peeves are people who show charts where, I mean, the whole reason you have a chart is to show proportionality, to give a sense for, if a line is going up, then the amount it goes up as it moves horizontally matters. I mean, that's why you're doing it. So it's about proportionality. Well, if you don't make it very clear where, well, the way the vertical axis is anchored, then the scale of its change has no meaning.

Leo: Sometimes people do that in a nefarious way to make the scale of change look greater.

Steve: Yes. Yes, exactly. In fact, that's the reason that it's annoying is you'll see someone that shows, like, okay, rate at which national debt is increasing, and it's like, whoa. You go, holy crap. But then it's like, wait a minute. It started, like the Y scale is already really high. So it's basically the amount of change that occurred over time. It's like, well, yeah, okay, fine. Whereas, if you zero base it, it looks like a much flatter line.

Leo: Well, whoever did this illustration apparently had a poll; right? Is it misleading to truncate the Y axis? In other words, not to have it zero based? And the results are in. Notice, by the way, that this graph starts at 98%.

Steve: Right.

Leo: And so there was 1% no, and 99% yes.

Steve: Correct. Exactly. So in a properly scaled graph that was based at zero, it would almost be no red slice at all that you could find. It'd be 1% of the entire height, rather than 50%, as this chart shows. Anyway, someone sent this...

Leo: That's just great. I just love that. I love that.

Steve: Someone sent it to me some time ago, and it was in my bag of things to show on the podcast, so I wanted to get around to it.

So looking back as we have been at previous Second Tuesdays of the Month patches, by any measure, last week's Patch Tuesday was a doozy. And it turns out "doozy" is a word. I was glad to know that because I thought, there's just...

Leo: Playing Scrabble, is that how you found it?

Steve: Well, I had to look. I looked it up because my spellcheck said, "What? Are you drunk?"

Leo: There's no "doozy."

Steve: Wikipedia or wherever it was says, oh, yeah, doozy. So I'm now having to train all my various spellcheckers. It's like, let me just type "doozy" without bothering me. So these Patch Tuesdays seem to be getting larger. Remember, Leo, those quaint old days a few years ago when a big Patch Tuesday was like maybe 15? We'd be like, wow, 15 things. Okay, yeah. Not recently. Last Tuesday Microsoft's February patches resolved 74 CVEs and also had three advisories, so a total of 77 problems. And they encompassed IE, Exchange Server, the ChakraCore, Windows itself, Office, Microsoft Office Services and Web Apps, Azure, Team Foundation Services and the .NET framework. And of those 74 CVEs, 20 of them are rated critical, with the other 54 rated as important. And then we've got those three advisories as moderate.

Twenty-one of them were sourced by Trend Micro's Zero-Day Initiative. Four of the bugs are publicly known, and one is under attack at the time that it was patched. So it was good to get done. One of the more interesting things was CVE-2019-0626, which was a Windows DHCP Server Remote Code Execution Vulnerability. Most of us have DHCP servers looking inwards on our networks. Although apparently Microsoft's DHCP server has some public exposure. It wasn't exactly clear to me, although there was some commentary on the 'Net suggesting that this was a wormable exploit. Well, to be wormable, really it needs to be outward facing, that is, bound. It has to be a DHCP server that has some presence publicly, although I wasn't able to nail that down one way or the other.

Anyway, the point is that this is one where certainly if you're an enterprise, you know, our DHCP servers are on our - we all have them on our routers looking inward, providing dynamic host configuration protocol, DHCP, to the various machines and IoT stuff and our smartphones and everything we have inside of our LANs. An enterprise will be using a DHCP server typically also for all of its network. And even if it's only an inward facing problem, depending upon the size of your company, we know that you can have a disgruntled employee or somebody who's just feeling mischievous and wants to see whether his corporate DHCP server has been patched.

Anyway, the point is this last Tuesday round was huge and not something that an enterprise wants to sit on for long. And of course for the rest of us there were 20 remote code execution problems that, to varying degrees, affect us. So definitely not one you, I mean, the problem is we've seen these patches recently causing problems for people. So the tendency is not to jump on them. But now we're also seeing "in the wild" exploits happening. So it's sort of a toss-up.

And Adobe, not to be left behind, weighed in with its own list of, well, they didn't quite match Microsoft, but they came close: 71 bugs. On the other hand, 44 of those 71 were rated critical, so more than twice as many critical problems for Adobe as Microsoft had. And of course it's across the spectrum - Acrobat Reader, Flash, ColdFusion, and Creative Cloud.

So once again, just make sure, if you're using Adobe stuff, and if for some reason you're still stuck using Flash because of corporate policy - which generally is the reason that's still being done. There are corporations that have Flash-based internal things for whom they've lost the source, or the programmers went away or whatever. You do want to make sure that you're being kept current. And really you want to try not to have Flash, if possible, available to your browser. Maybe you just have an internal standalone Flash app that requires that you have Flash in your system. So anyway, the point is, if you're doing Adobe stuff, it would be a good idea to make sure you are up to date there, as well as for Windows.

Last Friday, February 15th, Microsoft notified the world of an important forthcoming change. I have the link to their notice, which I read carefully, and it still left me with some confusion. The title was "2019 SHA-2 Code Signing Support Requirement for Windows and WSUS," which is Windows Server Update Services. So I'll share what they said, and then we'll talk about it. They said: "To protect your security, Windows operating system updates are dual-signed using both the SHA-1 and SHA-2 hash algorithms to authenticate that updates come directly from Microsoft and were not tampered with during delivery." That we all know.

"Due to weaknesses in the SHA-1 algorithm and to align to industry standards, Microsoft will only sign Windows updates using the more secure SHA-2 algorithm exclusively." And so here's where it hits half of us: "Customers running legacy," okay, yeah, legacy - "OS versions" - meaning in Microsoft's consideration - "Windows 7 and Windows Server 2008 R2" - in the case of Windows 7 it's SP1, Service Pack 1, which was the last one that they did for Windows 7; and for Windows Server 2008 R2 it's SP1 or SP2 - "will be required to have SHA-2 code signing support installed on their devices by July" - of this year, by July of 2019. Microsoft said: "Any devices without SHA-2 support will not be offered Windows updates after July 2019."

So then they said: "To help prepare you for this change" - Microsoft writing this said - "we will release support for SHA-2 signing in 2019," meaning before this time. They said: "Some older versions of Windows Server Update Services will also receive SHA-2 support to properly deliver SHA-2 signed updates." Then they said: "Refer to the Product Updates section for the migration timeline. Starting in early 2019, the migration process to SHA-2 support will occur in stages, and support will be delivered in" - now here's the key - "in standalone updates."

Okay. Today we're in February; right? So they said March 12th, 2019, which is the next Patch Tuesday, which is March 12th is March's Patch Tuesday, they said: "For Windows 7 SP1, Windows Server 2008 R2 SP1," they said, "Standalone updates that introduce SHA-1 code sign support will be released as security updates." So I read that as saying that they're going to push this out in the normal security update Patch Tuesday. So we don't have to do anything.

Then they said April 9, 2019, so that was March, so April, which is the - April 9 because April's Patch Tuesday, the second Tuesday of April, they said, for Windows Server 2008 R2 SP2, they said standalone updates that introduce SHA-2 code sign support will be released as security updates. Okay. So that's in March and April, successively. So for most of us, next month, March, for Windows 7, those of us who are still using Windows 7.

So then we have April, which is where servers get updated, Server 2008 R2 servers get updated with SP2. Then we have May and June and July to make sure we're all up to speed because the August 2019 updates will start being single-signed using SHA-2 only. So at the moment all of our Windows 7 systems are only aware of code signing using SHA-1. The point is, ending after July, so beginning in August, all of our updates are only going to be signed using SHA-2.

So what's weird is that also in there they talk about Windows 10. And they said July 16, 2019 for Windows 10 - and that's 1507, 1607, and 1703 - they said Windows 10 updates signatures change from dual-signed to SHA-2 only. "No customer action is expected for this milestone." Well, they didn't say no customer action was expected for the other OSes.

So anyway, we'll certainly talk about it next month. And the question is, will it be checked for us automatically? Are we all going to get it? The good news is we'll have plenty of time to make sure we got it because I fully intend to keep using Windows 7 until forever, hopefully, although as I said, I've kind of made peace with 10. If you strip all the nonsense out of it, you can pretty much live with it.

But anyway, so I just wanted to sort of cover this. Mary Jo talked about it. I imagine she'll - she wrote about it last week. I imagine she'll talk about it on Windows Weekly tomorrow. Maybe she'll know whether this is just being all done for us, or if we have to go, like, getting it for some reason. I don't know, when they say it will be released as a security update, that sort of sounds like, okay, fine. Well, like everything else. So just give it to me. But maybe it won't be checked by default, like Silverlight, where it's like no, thank you, please don't give me that.

So we'll find out next month. We'll certainly talk about it because, as we know, Windows 10 just crossed over the 50% mark. So the majority, and certainly all of enterprise, lots of enterprise, are still hanging back on Windows 7 and trying to decide how much they're willing to pay for continued service of updates from Microsoft.

I got sort of a smile out of the fact that there's been much talk made of the security of apps in the Windows Store. And of course Paul, I sort of also grinned when he was talking about how much junk is there. It's just like, oh, come on, really? It's just nothing is there. Symantec found eight apps on Microsoft's app store that mine Monero without the user's knowledge. Symantec wrote that: "On January 17th we discovered several potentially unwanted applications" - and I got a kick out of this acronym. PUA is the acronym we're using now, P-U-A.

Leo: What is it?

Steve: A Potentially Unwanted Application.

Leo: Oh, PUA, yeah. A PUA.

Steve: A PUA, yeah. You don't want of those PUAs. It's too bad there wasn't some way to make it PU. That would have been better. On the Microsoft Store that - so what they're doing is of course, as we know, Symantec wrote: "Surreptitiously use the victim's CPU power to mine cryptocurrency." And they said: "We reported these apps to Microsoft, and they subsequently removed them from the store."

Now, despite the fact that there were eight apps, it turns out they were all from one source. So they figured, okay, we'll just kind of spread this around eight apps, so maybe somebody will find one, but they won't find the others. Symantec wrote: "The apps - which included those for computer and battery optimization tutorial, Internet search, web browsers, and video viewing and download - came from 'three developers': DigiDream, 1clean, and Findoo." In total...

Leo: You got some DUA from Findoo. And PUA. Findoo put PUA on your computer.

Steve: PUA from Findoo, that's right. It says: "In total, we discovered eight apps from these developers that shared the same risky behavior." Uh-huh, because they're actually all one entity. "After further investigation," they wrote, "we believe that all these apps were likely developed by the same person or group." The one thing I found interesting here was they found a new way, well, new in this instance, for getting malicious JavaScript into these things. These are running as - what was the term? I wrote it here somewhere. Oh, yeah. The eight apps fall under the category of progressive web apps which are installed as a Windows 10 app running independently from the browser in a standalone that's the wwahost.exe process. So basically it's a JavaScript web app, but doesn't run with all the browser window dressing. It just looks like it's a freestanding thing.

Anyway, the point is that Symantec said: "As soon as the apps are downloaded and launched, they fetch a coin-mining" - so they don't include coin mining in the app, which is a little clever, so they could be scanned and checked and so forth. And then, oh, yeah, look. They look fine. Let them on. They fetch a coin-mining JavaScript library by triggering something known as the Google Tag Manager (GTM), which is from Google's domain. The Google Tag Manager is a tag management system created by Google to manage JavaScript and HTML tags used for tracking and analytics on websites. So it's not malicious. It's not bad. It's just basically sort of a form of CDN.

So these apps use GTM, the Google Tag Manager, to obtain the JavaScript from Google, where essentially it's being hosted. The mining script gets activated and of course begins using the majority of the computer's CPU cycles to mine Monero for these bad guys. Although the apps appear to provide privacy policies, just as if to look complete, there's no mention of coin - yes. Like, hey, we have a privacy policy.

Leo: Can we just copy something from the Internet and put it in there? Yeah.

Steve: Exactly. We got this from something else. There's no mention of coin mining, no surprise, not that anyone read the privacy policy anyway. They ought to really say, "and we're going to mine cryptocurrency..."

Leo: They should because nobody reads them.

Steve: Yeah, exactly.

Leo: Then you're off the hook.

Steve: Yeah, exactly. Anyway, the apps were published between April and December of 2018, most toward the end of the year. So one or I guess some of them had been around for a while. And Symantec wrote that even though the apps were on the App Store for a relatively short period of time, a significant number of users apparently downloaded them.

They said: "Although we can't get exact download or installation count, we can see that there were almost 1,900 ratings posted for these apps. However," as they note, "app ratings can be fraudulently inflated, so it is difficult to know how many users really downloaded them. When each app is launched, the domain Fast-search.tk, which is the domain for the Fast-search Lite app which is hardcoded into each app's manifest file, is silently visited in the background and triggers Google Tag Manager with the key GTM-PRFLJPX," whatever that is, "which is shared across all eight apps." Thus the common link that ties them together and lets Symantec believe that these were all coming from the same place.

So Symantec, just to be clear, explained that "GTM is a legitimate tool that allows developers to inject JavaScript dynamically into their applications. However, GTM," Symantec observes, "can be abused to conceal malicious or risky behavior since the link to the JavaScript stored in GTM is https://www.googletagmanager.com/gtm.js?id=," and then that GTM ID, which was that GTM-PRFLJPX, the point being that it's being hosted from Google. So trust us. Anyway, yeah, no. So just another clever way for bad guys to get code loaded into their apps after it's been downloaded and waiting, essentially deferring the actual load of the payload until it's actually run.

And as we were talking about recently, this is one of the behaviors that Google will be deliberately blocking in Android apps because it's just too dangerous to allow the app to defer loading most of itself until after it's already passed through Google's scrutiny. What does it mean, then, to check the app for its behavior, if you're allowing the app to load more of itself or change itself after the fact.

Anyway, Symantec informed Microsoft and Google about these apps' behaviors, and Microsoft removed them from the store, and Google pulled the mining script from the Tag Manager. So anybody who already had it will no longer have it running on their machines because it won't be able to any longer load its mining code. And as we said, there is incentive to create these things because it is generating so much money for the bad guys. They're just - it's not hurting people except that it's like, gee, I'm running my battery optimizer, and now my machine is running really slow. Yeah. Exactly.

So the next major release of Chrome and others through 2019 will be offering an experimental new lockdown technology which Google has dubbed "Trusted Types." This is aimed at developers rather than end users. End users, we get the benefit indirectly of, if this succeeds as Google expects, solving or potentially eliminating the by far number one biggest problem with client-side cross-site scripting vulnerabilities.

Our browser ecosystem, the way this has evolved, it's become incredibly complex over time. It's based on a textual, loosely typed and interpreted authoring environment which results in code which on one hand easily does what developers want, but unfortunately will also obligingly do what developers never intended. In this environment where precoded modules are being sucked down from other sources with abandon in order for various functions to be just looped into existing code the developers use, developers are rapidly gluing together complex functions built up from code they've never seen.

So this makes it incredibly difficult to program defensively. And the result is what we've got today, which is extreme vulnerability to very subtle and very difficult to spot cross-site scripting vulnerabilities. And as I said, they're very difficult to find for developers, even in the unfortunately rare event that developers are looking for them. And unfortunately, it's often the bad guys who are doing the looking and finding. So without getting too far down in the weeds of the details, the problem that Google is fixing, or proposing a fix for, arises because elements which make up web pages are part of the so-called DOM, the D-O-M, the Document Object Model, which is a formal description of a web page's structure, which over time sort of has been reverse engineered. We ended up just like sort of with an ad hoc, here's a web page, and then we said, okay, let's formalize this.

So it's been formalized and carefully designed. And now we have a rigid, well-defined model for documents. And the specifications for the sources for the objects that populate the document object model are strings. And it turns out that all too often the composition of these strings, and by that we mean like URLs, HTML URLs, the composition of these strings can be subject to malicious manipulation, with the result being that foreign content from some malicious source can be injected into the innocent page's document object model and then made to execute in the context of this model.

So again, the problem is that simple strings can be used as the source specifiers of these objects. And as Google terms it in their security blog posting about this, they said they are insecure by default. So yes, they're easy to use, but also they are the number one source today of web-based attacks.

So what Google's developers are proposing is the addition of a new and optional argument to an existing browser header. We already have something called CSP for our browsers, Content Security Policy, where the page author is able to put constraints on the things that the page is allowed to do. The browser receives this as a CSP, a Content Security Policy header. And it's up to the browser to enforce what the web server has said it wants enforced. So this is adding a new feature known as Trusted Types.

So a web server that wanted to solve the problem of its pages being abused would add a content security policy of Trusted Types. And if that's present, then all of those places where a string could have been used and easily misused, will refuse to accept a standard string, just a random string as their argument. Today they accept anything you give them. If you say no, we want to enforce Trusted Types on this page, then they will require an explicitly specified policy, basically a template and policy to be created for that object's specification.

So essentially, I mean, it is tightening down on this kind of freewheeling, anything goes approach that we've had, which has unfortunately resulted in a huge amount of abuse. Like this cross-site scripting is the number one problem we have on the web today. Google says: "In practice, modern web applications need only a small number of these policies." They wrote in their explanation of this: "The rule of thumb is to create a policy where the client-side code produces HTML or URLs - in script loaders, in HTML templating libraries, or HTML sanitizers." The point being that most of the time you don't need to dynamically produce those. Where you do, if you enforce Trusted Types, then you'll need to explicitly control what those things are able to do.

And they said: "All the numerous dependencies that do not interact with the document object model do not need the policies." And they said: "Trusted Types assures that they can't be the cause of the cross-site scripting." So with the next release, I think it's 73 of Chrome, this will begin to appear. And 73, 74, 75, and 76 are the releases Google has slated for this year, where they're going to begin to roll this out and allow developers to explore this. They also have - they've come up with some JavaScript code that non-Chrome browsers can use in order to essentially retrofit this functionality in so that developers can experiment with it in places other than Chrome.

And the hope is, first of all, that there's no gotchas that have not been foreseen; that developers will appreciate the leverage that this provides, that is, essentially developers would like to have it, but the browsers don't offer it at the moment, and the servers are not suggesting that it be enforced. So all these things sort of have to happen. Then of course finally, if this proves out, then this would move into standards mode and ultimately get adopted by Firefox and hopefully, well, actually, if Edge is going to be adopting the Chromium engine, as we believe it will be - well, we know it's going to be - then before long that'll be able to be used by Windows 10 users with Edge on Windows, as well.

So again, who would have thought that a company that began as a search engine - of course Google has since expanded to be much more. But I just, you know, we're covering a lot of things that Google is doing to make the web a safer and more secure environment for us. And I just - I take my hat off to them. I'm glad that there's somebody who's being as proactive as they are. So yay.

And Leo, I think you told me how to pronounce this name. It's Xiaomi, X-I-A-O-M-I. Xiaomi I think is how we decided that should be produced. Who would have imagined that Xiaomi's electric scooters would be vulnerable to remote hijacking. Well, or on the other hand, who would have thought they wouldn't be, given that we have such a problem securing things these days.

Last Tuesday researcher Rani Idan, who is with Zimperium - Zimperium we've spoken of often. They're the somewhat controversial zero-day exploit promoter and reseller, reselling being the controversial part of this. They are offering a lot of money, and we were wondering who's buying these zero-days. Anyway, they disclosed a vulnerability which is present in the Xiaomi...

Leo: Xiaomi.

Steve: Oh, Xiaomi, Xiaomi. Thanks. Xiaomi. The Xiaomi M365 electric scooter. If you happen to have a Xiaomi M365 electric scooter, be careful. Until it's patched, and now it's widely known publicly, the demo that the Zimperium guy showed only showed them stopping the scooter because that's the least horrific thing that could be done is that it stopped when it's not moving at a stoplight. And, oh, look, you try to go, and it doesn't. The guy had to pick his scooter up and carry it across the street. You really would not want to be on one of these that accelerates to maximum speed and refuses to stop, which is something that can happen. So the story is...

Leo: Oh, no. It's a scooter, so you can hop off.

Steve: Well, yeah, but still, I mean, it's going to catch you by surprise. Do you know how fast they go? I don't have any idea.

Leo: Not that fast.

Steve: At full speed. Okay.

Leo: I mean, unless you can break through the speed limiters, which you might be able to do. I don't think they're - I don't know how they work, but...

Steve: Yeah. So this scooter interface app is running on the phone. So somebody has a phone which they authenticate to with a password to the scooter's app. What the Zimperium guy found was that the phone app has no authentication protocol to the scooter. So there's, like, it's wide open. The scooter itself has no authentication. So anybody who reverse engineers the protocol which the scooter uses to talk to its app, and apparently it's wide open, also it works at a hundred-meter distance, which suggests it's WiFi and not Bluetooth.

So the point is that a bad guy, knowing that these scooters have no authentication, and I don't think there's - oh, yeah, Zimperium did create proof-of-concept code, which they used in a video that they produced to cause this scooter to stop. Anyway, there is the ability to completely control acceleration and any other features that the scooter offers through this phone. Unfortunately, Xiaomi said that they'd been made aware of the findings, and said that it was a known issue internally, and blamed it on third-party products. So no patches available. It's not known whether any one is forthcoming.

So just a little heads-up, if you are using one of these M365 scooters. I mean, the chances are you yourself are going to be targeted seems remote. And as you said, Leo, you can hop off. But still, it would be startling to suddenly have your scooter accelerate and ignore its controls, which apparently you can do through this software. And now the word is out. Oh, and the software that these guys came up with scans for available scooters to take over. I don't know how it...

Leo: I'd be more concerned if it were, like, Jump or one of the, you know, Scooter, Scoot, or Skip, or Lime, one of those big scooter rental companies because those are all over some cities. And it is an issue.

Steve: Well, and apparently the article did mention, although I didn't have it in the notes, that they rent these scooters.

Leo: Oh, okay.

Steve: That is, so these scooters are being used.

Leo: Oh, Lime uses these scooters. Oh, never mind. Then it is a huge issue. I take it all back. Lime is all over the place, those Lime scooters. And if Lime uses these, then what that means is, well...

Steve: It's bad.

Leo: Lime could fix it. But that explains it because most scooters are not controlled by phones. That seems like an unusual way to control a scooter.

Steve: Right.

Leo: Okay.

Steve: Right. It'd be a handgrip, right, that you use to control a scooter, yeah.

Leo: Yeah, yeah. Why would you control it with a phone?

Steve: For whatever reason...

Leo: But on these rental scooters you need Android or, I mean, smartphones to connect to it and unlock it.

Steve: Right, right, right.

Leo: Which means it has an interface, and obviously that's what they've worked around. So, yeah, I would guess this is in fact rental scooters. And that's a lot more serious. You get a lot of people don't know what the hell they're doing on them.

Steve: Exactly. And, I mean, I can imagine, first of all, you wouldn't be expecting this. And if you're on it, and it suddenly takes off at speed, it's like, you know, you're going to be startled. And, now hopping off, I don't know how fast they go, but to hop off, if you're moving, you've got to like hop off and run; right? Otherwise you'd fall.

Leo: Oh, yeah. They don't go faster than 20 miles an hour, but that's still pretty fast. Apparently Lime and Bird, which is the other big company, use these 365s.

Steve: Oh, boy.

Leo: On the other hand, that means Lime and Bird have heavily customized them and probably could fix this themselves in the interface, I would think.

Steve: I hope they...

Leo: To hear there's no authentication, that seems really dopey.

Steve: Isn't that bizarre? And that's what Zimperium found.

Leo: At least XOR the stuff or something. Or something.

Steve: Yeah. Like whoops.

Leo: Whoops.

Steve: Okay. So some interesting research, chipping away at Tor. We love Tor. Talked about it in the beginning. Very, very clever technology. If anyone isn't - if we have a new listener who's not up to speed on Tor, we did a whole podcast on it, The Onion Router, TOR, was what that originally stood for. For some reason they decided they didn't want to be an acronym or an abbreviation anymore. So they said no, no, no, we're just Tor. We're not The Onion Router. It's like, okay.

The ultimate vulnerability to Internet anonymity, as we've talked about before, is the fact of there being any traffic flow between endpoints. We can encrypt the traffic so we cannot read it. We can encrypt the flow's metadata so that we cannot learn anything about what the content is, other than perhaps its size, of the data being moved. And we can introduce camouflage packet padding and short-term aggregation of packets to clump them together to hide their individual presence. There's like all these things that we can do. But eventually, the same packet needs to come out of this cloud of obfuscation that went in, so that any entity that can see enough of the cloud's perimeter can get some sense for which endpoints are communicating. So they may not know what is being said, but the existence of a flow is a form of metadata that reveals something.

So in this new - and we've talked about how, in the worst case, looking at the flow in and out of the Tor cloud, even though packets go in, and they jump around a lot inside, and no node can see what the other node is doing and so forth, I mean, it's beautifully designed. It comes out the other end eventually, after all of the wrappers of encryption have successively been removed from this packet. And if somebody suspects that two endpoints are communicating, then it's much easier to confirm a suspicion than it is to use Tor to generate the belief of endpoints connecting because it is a big, large, active network now. Still, confirmation is easier than coming in cold and trying to figure out who's talking to who.

We have another piece of this that's been done. In their paper titled "Peel the Onion: Recognition of Android Apps Behind the Tor Network," four Italian researchers at the Sapienza University in Rome have chipped away a bit more at the protections offered by Tor. What we have now they have coined the term "application deanonymization attacks."

They wrote: "In this work we show that Tor is vulnerable to application deanonymization attacks on Android devices" - and they just used Android, iOS would be the same - "through network traffic analysis. For this purpose, we describe a general methodology for performing an attack that allows us to deanonymize the apps running on a target smartphone using Tor which is the victim of the attack.

"Then we discuss a proof of concept implementing the methodology that shows how the attack can be performed in practice and allows us to assess the deanonymization accuracy that it's possible to achieve. While attacks against Tor anonymity have already gained considerable attention in the context of website fingerprinting in desktop environments, to the best of our knowledge this is the first work that highlights Tor vulnerability to apps deanonymization attacks on Android devices. In our experiments we achieved an accuracy of 97%."

Okay. So their 15-page paper goes into every detail of what they did - lots of pretty graphs - what they observed and what they found. The bottom line is, not surprisingly, mobile applications tend to be quite chatty. They assume and use the connectivity that they have freely. And in the process they give their identity away, even when the traffic is protected and encapsulated and wrapped up by multiple layers of onion.

Again, the fact of traffic occurring creates a fingerprint. So even with a Tor-enabled smartphone, what these guys demonstrated is that it is possible, after some training, and given some universe of applications whose behavior becomes known to figure out what a person is doing on their smart phone simply by passively sniffing the WiFi traffic or cell traffic that the phone is using. So again, you know, it's not a decryption compromise. It's not a huge deal. And by itself I would argue that it isn't.

But I think it's an interesting finding, and it succeeds in breaking one of the privacy assumptions that a TOR user might have, which is that it's not possible for someone listening to their communications to know what they're doing. They don't know what they're saying. But they can identify, these guys demonstrate with a high degree of accuracy within a given set of applications, what it is that the user is doing. So we sort of come back to where we began, talking about how the Internet works and packet routing, noting that none of the Internet was designed with security and even privacy in mind. Back then, the fact that it even worked was amazing. And we were later able to retrofit privacy on top of it by encrypting the communications between endpoints.

But the inherent nature of the whole packet-switching approach to the Internet and the way traffic routes really makes complete privacy difficult. We can hide the content, I mean, robustly hide what we're seeing. But the fact that communication is occurring, they're just - without, I mean, even when you go to all of the trouble that Tor has gone to, there are still ways to chip away at that.

Leo: This isn't what Tor is for. Tor is to anonymize you so it's not clear where the traffic's coming from.

Steve: Ah, that's a very good point.

Leo: It doesn't in any way encrypt the traffic. If you wanted to...

Steve: Well, no, it absolutely encrypts it. And that's the four layers or more of onion. But oh, I see, it's like a VPN.

Leo: Use a VPN if you want to encrypt. If you use Tor plus a VPN, I don't think this technique would work. Of course they don't know where the traffic's coming from, but they can see the apps that you're using because apps have a certain signature.

Steve: Correct.

Leo: That doesn't seem - that seems not why you're using Tor. Now, they're right when they say "privacy assumptions a Tor user might have" because that's...

Steve: Correct.

Leo: That may be a mistake in the Tor user; right?

Steve: Exactly. Exactly.

Leo: Okay.

Steve: Although Tor does encrypt in the same way that a VPN does.

Leo: It does. You don't need a VPN with Tor?

Steve: Well, correct, because the client - what happens is the client decides what nodes they're going to move through, so it picks like four Tor nodes and gets the public keys from each of the nodes. Then it successively encrypts in the reverse order that it's going to transit, the packet's going to transit.

Leo: Oh, right, of course, yes. It wraps it and then unwraps it.

Steve: Right, right, right, right. And so in that sense it's very VPN-esque, the idea being, though, that by getting lost among the Tor nodes, it's not easy to backtrack and for someone to see where you're connected to.

Leo: Right. And of course, like VPN, it's not encrypted once it emerges from the Tor exit node.

Steve: Exactly. Exactly.

Leo: Okay. Yeah, of course it's encrypted. What am I thinking? That's the technology, yeah.

Steve: Right. So CNBC, it occurred to somebody there to do some little digging around and wonder, almost a year and a half after Equifax, whatever happened to that data? It's been just a month shy, 17 months, of a year and a half since the 2017 Equifax data breach, which we later found to have compromised the data of nearly 148, I think it was 147.9 million individuals, which is nearly every adult in the U.S., with more than 45% of the U.S. population directly affected by the incident.

Anyway, an investigative report by CNBC found that, somewhat surprisingly, none of the data has turned up on the dark web. According to CNBC's "threat hunter" sources, they talked to a bunch of people, it's increasingly looking like it was, as they called it, a "spy job," meaning carried out by a nation-state, not criminals who are aimed at ID theft or short-term financial gain.

Threatpost covered the CNBC research, and they asked our well-known friend Troy Hunt of HaveIBeenPwned fame. Troy said: "Frankly, I think the bullet point under the headline about it being state-sponsored explains a lot." He said: "Actors at that level aren't looking to cash data in for a few bitcoin," he said, "and it wouldn't surprise me in the least if that data never sees the light of day." He said: "Just think about how many incidents must be out there already that we may never know about simply because those responsible have no reason to advertise it."

And I think that actually, you know, that's a very salient point. We know of breaches when we see them turn up and then can reverse engineer where they came from. Sometimes corporations proactively learn of a breach and then disclose it because they have due diligence. But we also wonder how many corporations think, ooh, shoot, yikes, somebody was in here, but as far as we know nothing got out. Well, they may not know because they're not logging, or they don't have any incidence management or who knows what.

Anyway, I just thought it was fun. We've talked about Equifax a lot. And it could easily be that some nation state just wanted to collect this data and then use it selectively in targeted attacks for identity theft against specific individuals, rather than, as Troy said, just selling it for a few hundred or thousand dollars on the dark web. It could potentially be worth a lot more to a nation-state.

So two years ago the U.S. Congress's House Energy and Commerce Committee requested that the GAO, the U.S. Government Accountability Office, prepare a report about Internet privacy. The report ended up just happening, last week it was finished, titled "Internet Privacy: Additional Federal Authority Could Enhance Consumer Protection and Provide Flexibility." I read the entire thing, just to see if anything really jumped out at me that I should share.

The first hearing on the report is scheduled for a week from today, which puts it on the 26th of February. So there will be a hearing in Congress's House Energy and Commerce Committee. And essentially the GAO is strongly recommending that the Federal Trade Commission, the FTC, be put in charge of overseeing the enforcement of Internet privacy under some legislation that they're also proposing. The FTC already technically has responsibility, but it has lacked sufficient legislation and essentially sufficient enforcement power to allow it to do much. For example, in the entire history of the Internet, the FTC has been involved in only 101 Internet privacy-related cases, despite wide privacy abuse being reported by users and of course the media.

The GAO in their report argues that this new proposed legislation should give the Federal Trade Commission much more teeth in dealing with privacy abuse. They talk about in the report, of course, the Facebook/Cambridge Analytica relationship, the dangers to user privacy due to lack of regulation and oversight in many different fronts. They talked about the ever-growing Internet of Things sector, where devices collect massive amounts of information without the user's knowledge; automakers collecting data from smart car owners; the lack of federal oversight over companies that collect and resell user information, the lack of protections for mobile users against secret data collection practices.

So I think we're just at the beginning of the process. This report will go to this committee. There'll be some discussion. This will probably take a while. We don't know how quickly the U.S. Congress will act. But in the current climate it does feel like the world is ready, or the U.S. is ready, for something that is much more substantial. The GAO report does interview and talk about "stakeholders," as they put them, on both sides of this. Of course the argument is, oh, this is going to stifle innovation. This is going to keep us from doing the things that we need to do, the argument being, well, okay, we'll take that into advisement and into consideration. I'm sure there will be lots of lobbying back and forth.

What I do hope is that whatever it is we get is clear and is not just some murky, weakened legislation that just allows lots of lawsuits to get filed so that everything gets up and just gets tangled up in court and in appeals and ultimately burdening the Supreme Court as a consequence of legislation not being made very clear.

The GAO ended up saying: "Congress should consider developing comprehensive legislation on Internet privacy that would enhance consumer protections and provide flexibility to address a rapidly evolving Internet environment. Issues that should be considered include which agency or agencies should oversee Internet privacy; which authorities an agency or agencies should have to oversee Internet privacy, including notice-and-comment rulemaking authority and first-time violation civil penalty authority; and how to balance consumers' need for Internet privacy with industry's ability to provide services and innovate." So no surprises there. But we've seen what the EU has done with the GDPR, and I wouldn't be surprised if this is the beginning of something similar for the U.S. I think we need it.

So I was talking about this version of Windows 10 that was available to enterprises. And I got a note from a Todd Fillingim in Mississippi. The subject was "Windows 10 Long-Term Service Channel," and he said "(Branch)." He said: "Steve, I've been listening to SN since Episode 1..."

Leo: Yay.

Steve: Yes, "...but have never submitted feedback. Love the show, and thank you for doing it for so long. In SN-701," so last week, "you mentioned a version of Win10 that has everything stripped out of it, available to enterprise customers." In fact, that's why I'm bringing it up because we know that enterprises are being reluctant to make this move. So this is an option I want to make sure our enterprise listeners are aware of.

He says: "You mentioned a version of Win10 that has everything stripped out of it, available to enterprise customers. I work for a small utility company, and we've just recently started testing this version for use on our control room operator workstations." So you could imagine they want a solid and stable version of Windows. "Microsoft used to call this version the Long Term Service Branch (LTSB), but recently changed it to Long Term Service Channel (LTSC). It is a version of Win10 that has none of the extra features of the regular version, and every build has a commitment from Microsoft for 10 years of security patches and support."

He says: "Interestingly, everything you read on this version from Microsoft tries to talk you out of using this version. They REALLY," he has in all caps, "don't want anyone using this for desktop use."

Leo: How funny.

Steve: And there's a link to it. So he says: "Thanks again for the podcast." And of course it's not available to end users anyway, so it's going to be for enterprise deployment. But depending upon how you feel about having Candy Crush on your employees' machines, enterprise customers, this may just be the thing for you.

Owen LeGare in Davis, California. Ah. Many people picked upon this, my confusion over MiB for megabytes. We've talked about it a couple times. I've never really paid that much attention to whether it's a thousand or 1024. I mean, as a programmer I do. I know exactly what I'm talking about. And I sort of thought I remembered the convention once, like in KB, if the "B" was capital, it was bytes, and if it was lowercase it was bits. But of course this is different. This is, is it a binary or a decimal thousand.

Leo: And you wouldn't, by the way, need this at all. It's always going to be binary, except that stupid hard drive manufacturers decided to inflate their capacity.

Steve: To inflate, exactly.

Leo: And say it in decimal.

Steve: Exactly.

Leo: And that's so annoying that we have to even say KiB or MiB.

Steve: I know. So just, you know, M-O-U-S - anyway. So Owen said - and everybody, thank you for your email and your tweets. I just chose this one. So the point is, he says, he uses his flash drive as an example, where it was done exactly as we were saying, where it's 8.13GB or 7.57GiB because you're using 1024 versus a thousand. Anyway, thank you for the clarification.

Leo: I think we said that on the show. We clarified.

Steve: Yeah, we did. I just didn't want - because I wanted everybody to know that we'd gotten the message.

Leo: I'm trying to remember if there was a - I believe there was a consumer action against using these decimals because at a certain point they started to indicate decimal, or as he did, this is decimal. This is binary. I think that somebody - I remember there was a lawsuit or there was some consumer action that said you can't be using these inflated numbers without saying.

Steve: Right. I got an interesting note from someone calling himself Dr-Mosfet on Internet isolationism. He said: "During the last few podcasts you've discussed various forms of Internet isolationism." He said: "If Elon Musk's Starlink becomes a reality, it could shake things up in both good and bad ways." And I just wanted to just point that out. That's an interesting note is that, if you can get the Internet via satellite, then it does, I mean, then it's in the air. And it makes it much more difficult. I guess totalitarian regimes could shoot the satellites down or refuse to have them hovering over them. I don't know how that would work in real life.

But I just wanted to note that I've just been entirely focused on wired gateways and borders and routers and so forth. But radio works. Radio Free Europe back in the day was a way of communicating in a way that was difficult to prevent. So this sort of, you know, the Internet via satellite is an alternative. And then, finally...

Leo: Yeah, and I suppose anybody could get access anywhere in the world, as long as it's up in the air. Maybe that's Elon's secret plan. I wouldn't be surprised.

Steve: I wouldn't either, although this other plan of his has got me a little worried. But we'll be talking about that momentarily. I know that you talked about DuckDuckGo in Safari, Leo. I don't think, I mean, I know you mentioned it everywhere else. I don't think we talked about it on this podcast. So Walter Pereira in Brazil, he said: "Hi, Steve. I'm a more than decade-long user of SpinRite." He says: "The best piece of software I have, period. Not once failed me. As far as I remember, the only software I own with that record. Thanks so much." He said: "But I'm writing to suggest a very small correction on what was affirmed" - and this was, you know, when there was a mention made...

Leo: You didn't say it, though; right?

Steve: No, I didn't.

Leo: So you don't need to run corrections for me.

Steve: Well, but our listeners just wanted to make sure...

Leo: Oh, god, I know. I got a million letters. Don't worry.

Steve: Exactly. Yeah. So it is possible to set up DuckDuckGo as your search provider for Safari. You're not stuck with Google.

Leo: No. But nobody does. That was the real problem, which is Google made $9 billion last year by being default, yeah.

Steve: Exactly. Okay. So this was last Thursday, which it was a blog posting by a company, OpenAI.com, which summarizes the recent work of six AI researchers at this company that Elon Musk founded. And I did note - I didn't have a chance to track it down, ran short of time. But apparently he's distancing himself from this. And I don't know what that means, like why he's not happy with...

Leo: He was not the only founder. A bunch of people raised money - Reid Hoffman, Elon, Peter Thiel - to do safe AI. They were very concerned about what everybody else was doing. So they created OpenAI to create safe AI. And I think half the mission of OpenAI is to warn people about AI.

Steve: Well, yeah. And in fact, what they created they decided they couldn't have open because - as a consequence.

Leo: Well, okay. So, yeah, I agree, and I can't wait to hear what you talk about here. But remember they're trying to stop certain kinds of AI. And since they never released this, this all could be speculative.

Steve: Well, okay. So I guess we should have seen this coming. For quite some time, as we know, we've had bots roaming the Internet, talking about Google, before then Alta Vista. And those bots were roaming the Internet, indexing its pages, so in some sense reading them, but just to index them. And we've also had bots warring with each other. And then of course eight years ago we sat silent, witnessing the stunning performance and success of IBM's Watson on "Jeopardy!." It was somewhat intimidating since I couldn't have answered most of those questions, and this thing just, like, knew.

Leo: Yeah, but Steve, if you had access to Wikipedia instantaneously, you could have answered all those questions, too. I don't know why people were so surprised at the ability to answer. To me the amazing thing was the ability to understand.

Steve: Well, that's what I was just going to say, to understand the questions, yeah.

Leo: Because if I typed in the - Wolfram Alpha will do it. Google will do it. It's understanding and giving it back in speech. It was a little bit of a parlor trick, if you ask me.

Steve: Yeah. Anyway, it impressed me. So of course it's one thing for a stylish black cube to sit there, and of course now it's doing cancer diagnosis and other expert system AI-ish things.

Leo: Failing at, I should point out.

Steve: Oh, is it?

Leo: Yeah.

Steve: Oh, okay.

Leo: Watson has proven to be not nearly as good at some of the things it's been tasked with as they were hoping.

Steve: Well, and of course we know that, much like security problems, they only ever get better. They never get worse. So now what we have is text-generating AI bots which could endlessly roam the Internet, reading what they find, and pouring out nonhuman generated content under the guise of being human. Which, okay, for quite a while we've been living with photos being photoshopped, I mean, so much so that that's now a term, "photoshopped." And of course more recently we've been growing aware of the possibility that someone's clearly recognizable voice might be saying something that they never uttered.

So this spoofing is entering the mainstream. But in the jargon that we use on this podcast, we would say that those attacks were targeted, you know, photoshopping a specific image or targeting someone's voice. So while that's disturbing, their reach is inherently limited. But that's not the case if text-generating AI bots are let loose on the Internet.

Okay. So this blog posting was titled, modestly, "Better Language Models and Their Implications." And in their descriptive blurb - and I think I have a link, yeah, I have a link at the top of this for anyone who's interested in more details. In their descriptive blurb they said: "We've trained a large-scale unsupervised language model which generates coherent paragraphs of text, achieves state-of-the-art performance on many language modeling benchmarks, and performs rudimentary reading comprehension, machine translation, question answering, and summarization all without task-specific training." Meaning just without knowing a priori anything specific.

Okay. So I want to share a sample of what this thing puts out. But first I'm going to share a little bit of sort of their context so we get a sense for what they did. They wrote: "Our model, called GPT-2" - which is the successor to GPT, so this is GPT-2 - "was trained simply to predict the next word in 40GB of Internet text." They said: "Due to our concerns about malicious applications of the technology, we are not releasing the training model. As an experiment in responsible disclosure, we are instead releasing a much smaller model for researchers to experiment with, as well as a technical paper." And I'll just note it doesn't matter. I mean, if it can be done, it will be done. And this demonstrates it can be done.

They wrote: "GPT-2 is a large transformer-based language model with 1.5 billion parameters, trained on a dataset of eight million web pages. GPT-2 is trained with a simple objective: predict the next word, given all of the previous words within some text. The diversity of the dataset causes" - so if any of us, I know, Leo, you have toyed with, as I have, Markov chains.

Leo: Yeah. In fact, it's surprisingly easy to do.

Steve: Yes. An incredibly large, highly trained Markov chain where you have probabilities of going from this node to the next one and so forth. So this is sort of that kind of thing, where it can be used to create a predictive model. So they said:

"The diversity of the dataset causes this simple goal" - that is, predict the next word, but with a huge model. "The diversity of the dataset causes this simple goal to contain naturally occurring demonstrations of many tasks across diverse domains. GPT-2 is a direct scale-up of GPT" - so basically they said, wow, this is working. Let's make it bigger, see what happens. They said: "...with more than 10X the parameters and trained on more than 10X the amount of data.

"GPT-2 displays a broad set of capabilities, including the ability to generate conditional synthetic text samples of unprecedented quality, where we prime the model with an input" - and we're going to demonstrate that in a minute - "prime the model with an input and have it generate a lengthy continuation. In addition, GPT-2 outperforms other language models trained on specific domains like Wikipedia, the news, or books, without needing to use these domain-specific training datasets. On language tasks like question answering, reading comprehension, summarization, and translation, GPT-2 begins to learn these tasks from the raw text, using no task-specific training data." That is, just pour the text in. "While scores on these downstream tasks are far from state-of-the-art, they suggest that the tasks can benefit from unsupervised techniques, given sufficient unlabeled data and computation."

Okay. So they said, finally: "GPT-2 generates synthetic text samples in response to the model being primed with an arbitrary input. The model is chameleon-like. It adapts to the style and content of the conditioning text. This allows the user to generate realistic and coherent continuations about a topic of their choosing, as seen by the following sample." Okay. So here is a sample of this machine.

Leo: Can I just ask one thing?

Steve: Yeah.

Leo: Did they do this in front of people? Did they, like, say, okay, come in here, we're going to run this, and we're going to let you watch it, it'll be generated, and these people, whoever they are, verify that it was actually generated by this software? I mean, they haven't released the software.

Steve: No.

Leo: How sure are we that this actually happened? So you're trusting OpenAI, but OpenAI's mission, sworn mission is to warn people about the dangers of AI.

Steve: Well, they've done that for me, at least. So, okay.

Leo: I just don't - I don't know. I'm not sure I believe it. But go ahead.

Steve: You know me. I'm gullible. I believe this. So here's what was given to the machine: "In a shocking finding, scientists discovered a herd of unicorns living in a remote, previously unexplored valley in the Andes Mountains. Even more surprising to the researchers was the fact that the unicorns spoke perfect English." Okay. So that was given.

Leo: That was a human. That was a human wrote that.

Steve: A human wrote that as input to this system that has been generated. It then produced - and they did say it took 10 tries. So there was a button where they said, "Try again." "Try again." Okay. But this is what it produced. And again, remember, these things only get better.

Leo: Now, is this the one that learned everything from Reddit? I think the content that it absorbed was Reddit pages, I think I remember.

Steve: There was something about Reddit. Anyway, so again, okay.

Leo: Which might help you understand what you're about to hear.

Steve: So they put in: "In a shocking finding, scientists discovered a herd of unicorns living in a remote, previously unexplored valley in the Andes Mountains. Even more surprising to the researchers was the fact that the unicorns spoke perfect English." That was put in.

This machine then produced: "The scientists named the population, after their distinctive horn, Ovid's Unicorn. These four-horned, silver-white unicorns were previously unknown to science. Now, after almost two centuries, the mystery of what sparked this odd phenomenon is finally solved.

"Dr. Jorge Perez, an evolutionary biologist from the University of La Paz, and several companions were exploring the Andes Mountains when they found a small valley with no other animals or humans. Perez noticed that the valley had what appeared to be a natural fountain, surrounded by two peaks of rock and silver snow.

"Perez and the others then ventured further into the valley. 'By the time we reached the top of one peak, the water looked blue, with some crystals on top,' said Perez. Perez and his friends were astonished to see the unicorn herd. These creatures could be seen from the air without having to move too much to see them. They were so close they could touch their horns. While examining these bizarre creatures, the scientists discovered that the creatures also spoke some fairly regular English. Perez stated: 'We can see, for example, that they have a common language, something like a dialect or dialectic.'

"Dr. Perez believes that the unicorns may have originated in Argentina, where the animals were believed to be descendants of a lost race of people who lived there before the arrival of humans in those parts of South America. While their origins are still unclear, some believe that perhaps the creatures were created when a human and a unicorn met each other in a time before human civilization. According to Perez, 'In South America, such incidents seem to be quite common.'

"However, Perez also pointed out it is likely that the only way of knowing for sure if unicorns are indeed descendants of a lost alien race is through DNA. 'But they seem to be able to communicate in English quite well, which I believe is a sign of evolution, or at least a change in social organization,' said the scientist." Okay, now, Leo, I am stunned that some software can produce something as coherent as that.

Leo: We don't see all the versions it produced that they didn't give you.

Steve: That's true. Again, be skeptical. Just wait a year. As I said, these things only get better. So they wrote, and I'm glad they appreciate this, they said: "As the above samples show, our model is capable of generating samples from a variety of prompts that feel close to human quality and show coherence over a page or more of text." They said of policy implications: "Large, general language models could have significant societal impacts, and also have many long-term applications. We can anticipate how systems like GPT-2 could be used to create AI writing assistants, more capable dialogue agents, unsupervised translation between languages, better speech recognition systems.

"We can also imagine the application of these models for malicious purposes, including the following, or other applications we can't yet anticipate: generate misleading news articles, impersonate others online, automate the production of abusive or faked content to post on social media, automate the production of spam and phishing content."

They wrote: "These findings, combined with earlier results on synthetic imagery, audio, and video, imply that technologies are reducing the cost of generating fake content and waging disinformation campaigns. The public at large will need to become more skeptical of text they find online, just as the 'deep fakes' phenomenon calls for more skepticism about images."

They wrote: "Today, malicious actors some of which are political in nature have already begun to target the shared online commons, using things like 'robotic tools, fake accounts, and dedicated teams to troll individuals with hateful commentary or smears that make them afraid to speak, or difficult to be heard or believed.' We should consider how research into the new generation of synthetic images, videos, audio, and text may further combine to unlock new, as-yet-unanticipated capabilities for these actors, and should seek to create better technical and non-technical countermeasures. Furthermore, the underlying technical innovations inherent to these systems are core to fundamental artificial intelligence research, so it is not possible to control research in these domains without slowing down the progress of AI as a whole."

So anyway, when I saw where we are with something that a machine can produce, and what this does is, I mean, we have a system with the Internet where there is an assumption we have today that what we're reading, the postings we're reading, are from people. And I think that will soon no longer be the case. Bots have demonstrated themselves to be fabulously, I mean, astonishingly capable of indexing content on the Internet. No one can imagine life without a comprehensive Internet index, which we have entirely thanks to the fact that bots, I mean, like the Internet is their domain. It's where they live. And we're transacting, I'm transacting constantly with textual content that I have known is generated by other people. That is entirely, I think, endangered. And that represents a sea change in the way we think of the content on the Internet.

Leo: Okay. You obviously didn't see the article from two years ago about the Washington Post's robot reporter that did 850 sports articles without anybody noticing they were auto-generated.

Steve: Wow.

Leo: That's because it's in a domain, very specific domain.

Steve: Yes, yes.

Leo: So this is a technology that the Washington Post calls "Heliograf." And the thing is, it's very easy to write sports reports because it's really all very formulaic. Interestingly, just to clarify, the way they got the data for this robot, the OpenAI robot, is they did go to Reddit, but they used outbound links from Reddit, so to a variety of sources. But they used Reddit to qualify the sources because it had to have a score of three karma or better. So they said, well, that means humans picked this as a source of good content. But it also explains why this OpenAI AI was very good if you asked it questions about Miley Cyrus, Brexit, or "Lord of the Rings."

Steve: Although it got her age wrong, I think. I thought it was - I think something that it was referring to as seven years old. And so...

Leo: I'd be more nervous about it being used to generate, almost to use signal jamming to generate a lot...

Steve: Leo, it's going to. That's my point is that, when we have...

Leo: Not that it's good content, particularly, but just that it's just content.

Steve: Yes, flood.

Leo: Flooding.

Steve: A flood of junk, I mean, where it's just - it cannot be - it can no longer be discriminated. This is awful.

Leo: Yeah.

Steve: This is awful.

Leo: It's just the Internet. And you're right, it's coming soon to a common thread near you. On the other hand, there's plenty of humans that can generate endless amounts of crap all by themselves.

Steve: No, that's just it. Automation. I mean, it is volume, unfortunately. It's a flood.

Leo: What a world. What a world. But that's what we cover here on Security Now!.

Steve: Oh, Leo. Oh, lord.

Leo: I think this is a great topic to talk about in general, authenticity on the Internet. I think it's a great topic because it isn't just text. It's as you said, deep fakes. It's photos. We've already crossed that line with photos. Any photo can be...

Steve: Yup.

Leo: And videos, too, actually. We know videos and photos can easily be humbugged. And I think the general populace kind of understands that now. It used to be "photographic evidence" was deemed, well, that's proof positive. I think people now know, oh, that can be retouched. So much retouching's been done on magazine covers for 50 years...

Steve: Well, Leo, the term "photoshopped," it just it means...

Leo: Comes from that, yeah.

Steve: Yeah, that a picture was faked. It was edited.

Leo: I remember how discouraged I was when I found out they can essentially do the same thing in movies. They can take a movie star who has some wrinkle or mole or acne and literally clean it up. And they use automated tools that do it for every frame. They do it on a few frames, and then for every frame.

Steve: Wow. Wow.

Leo: And Alex Lindsay told me this has been widely used for years.

Steve: Well, and we've been talking also about digital actors coming where...

Leo: Yeah, that's obviously - yeah. Did you see "Alita" yet? One of the characters is a fully digital actor.

Steve: I did unfortunately see "Alita."

Leo: Who talked you into that?

Steve: I just, no, I wanted to see what Cameron was up to, and Lorrie and I suffered through it. The technology was good; but, oh boy, yeah.

Leo: So she was a human wearing a suit during all of the scenes and had two cameras on her face that were painted with dots.

Steve: That was the "Avatar" technology that Cameron developed for "Avatar."

Leo: Right. Nothing's real. Except you and maybe me.

Steve: We're staying real.

Leo: The jury's still out on me, though, I've got to say.

Steve: Keeping it real.

Leo: We do Security Now! every Tuesday, 1:30 Pacific, 4:30 Eastern.

Steve: Whatever day it is.

Leo: I don't know what day it is. You kidding? Are you kidding me? I have no idea. That would be 21:30 UTC Tuesdays, Tuesdays, Tuesdays. Please stop by, say hi. You can watch the live stream of the show as we produce it, TWiT.tv/live, or listen. We have audio, two different audio streams you can listen to. And if you do that, you might want to join us in the chatroom. That's where everybody else is who's watching and listening live: irc.twit.tv.

Steve puts a whole bunch of nice show notes together, transcriptions of the show, and audio, as well, on his website, GRC.com. In fact, when you get to GRC, you might as well pick up a copy of SpinRite, the world's best hard drive and recovery and maintenance utility. That's the only thing Steve charges for. Everything else on that site, so much great stuff. Find out how SQRL's going. Learn about passwords. There's health information there. There's lots of good stuff. GRC.com.

@SGgrc is Steve's Twitter handle. You can DM him there, or leave a message at GRC.com/feedback. We have audio and video of the show at our website, TWiT.tv/sn. And of course every podcast application in the world, including Spotify, has a copy of Security Now!. And if you subscribe, you'll get it the minute it's available of a Tuesday evening. And that's my spiel.

Oh, don't forget. We want to make sure we get everybody, listeners of every show to answer questions on our TWiT survey. I don't want any one show to be overrepresented. So we know a lot of you listen to Security Now!. We do this survey once a year. I know, if you listen to the show, you are privacy nuts. We do not collect personal information. We don't ask for your email address. It's on Survey Monkey.

Survey Monkey will record your IP address to keep you from answering the survey more than once. I think you can figure out how to get around that. But that's the - and we don't get that information, nor do we want it, because really what we're looking for is an aggregate of all listeners. But I do want to make sure that everybody from Security Now! weighs in because you're probably a little bit of a different group than some of the other shows: TWiT.to/survey19. We do this once a year, and it helps us both sell advertising, but also plan our programming for the future.

Steve: And you're saying that our listeners' propellers are wound a little tight, Leo? Is that what you're saying?

Leo: No. In fact, I'm thrilled they are. I'm thrilled they are. It's great. And it's one of the reasons - nobody who watches or listens to any of our shows is unaware of the privacy implications. And that's why we need to do the survey, because we don't collect information about you in any way, at any time. And so the survey is the one time once a year we get to find out in aggregate what your interests are and so forth. So thank you for doing that. I appreciate it. Not an obligation. Don't worry about it. No salesman will call.

Thank you, Steve. Have a great week. We'll see you next time on Security Now!.

Steve: Thanks, buddy.