The Crypto CurveBall

Description: This week we look at Google's addition of iOS devices as full Google account logon hardware security keys, an update on Apple versus Attorney General Barr, a serious new Internet Explorer zero-day and how the vulnerability can be mitigated, the release of Microsoft's Chromium-based Edge browser, the FBI's reaction to the Pulse Secure VPN vulnerability, another new and CRITICAL RDP remote code execution vulnerability that has slipped under the radar, a bit of miscellany, and then we examine the headline-grabbing CryptoAPI vulnerability that's been dubbed "CurveBall."

High quality (64 kbps) mp3 audio file URL: http://media.GRC.com/sn/SN-750.mp3
Quarter size (16 kbps) mp3 audio file URL: http://media.GRC.com/sn/sn-750-lq.mp3

SHOW TEASE: It's time for Security Now!. Steve Gibson is here. Lots to talk about. We're going to talk about Apple versus the FBI, Patch Tuesday, and that massive security flaw Microsoft did patch in Windows 10 and Server 2016. However, there's now a new security flaw in Internet Explorer. It's still not patched, but Steve has the fix. That and a lot more, all coming up next on Security Now!.

Leo Laporte: This is Security Now! with Steve Gibson, Episode 750, recorded Tuesday, January 21st, 2020: The CurveBall CryptoAPI.

It's time for Security Now!, the show that everyone has to listen to each and every week. Because if you don't, god knows...

Steve Gibson: You balls may get curved, Leo.

Leo: Steve Gibson.

Steve: You never know.

Leo: There's a reason he said that. He's not just out of the blue. Steve Gibson from the Gibson Research Corporation, our security guru. Many of us have been taught that there's no one better than Steve Gibson, over the years, day in, day out, to keep up on what's going on in the land of security. And you're right on top of this new CurveBall exploit, too, this year.

Steve: Well, actually I wasn't. I think it was you who referred me to it when we were talking about it last week.

Leo: It had just come out, yeah.

Steve: It had just happened. It was part of the January Patch Tuesday patches.

Leo: So new that it didn't have a catchy name yet.

Steve: It was known as the CryptoAPI flaw. Now CurveBall.

Leo: Now it's got a catchy name.

Steve: Yes. And there's also a snarky something about crypto chains. We'll get to it. There are two proof-of-exploits now on GitHub. But so what I had intended to talk about before whatever it was I talked about last week, was the failure of SHA-1 hashing. And it turns out that gets, I mean, horrible as that is, even that gets pushed back again - hopefully I'll be able to get to it next week - because too much happened. We have the addition of Google's iOS device family to their secure logon FIDO2 security key family.

Leo: Yeah. I saw this. I want to ask you what it means. I immediately took advantage of it, but I want to know what it means.

Steve: Yeah. So it's a good thing.

Leo: It's a good thing.

Steve: We've also got a little bit of more back-and-forth between Apple and our Attorney General Barr, which I wanted to touch on. We do have, and there's a takeaway for our listeners, a relatively serious new IE zero-day vulnerability which is, being a zero-day, it was discovered being exploited in the wild. And despite the fact that it's IE, it's possible for sites to deliberately invoke IE, if that's what they want specifically. And it's bad. It's a remote code execution vulnerability. So we've got to talk about that. There's something, I did it immediately, that our listeners can also do because it uses sort of a fringe DLL that isn't in the mainstream, so you can just sort of say, no, we're just going to disable it to keep it from being abused.

We've got the release of Microsoft's Chromium-based Edge browser. The FBI's official reaction to the Pulse Secure VPN vulnerability after that all hit the news. I guess that's what we talked about last week because it was so potentially worrisome. The FBI has told us some of what they know about it. We've got, believe it or not, this thing slipped under the radar because of the CryptoAPI flaw that had everybody all worked up, a new critical Remote Desktop Protocol remote code execution vulnerability, reminiscent of BlueKeep. We've got a bit of miscellany, and then we're going to examine the headline-grabbing CryptoAPI that, as we already said, has been given the name CurveBall.

Leo: That comes from the fact that it's elliptic curve crypto that is broken.

Steve: Yes. Well, it's an interesting flaw. It was introduced five years ago, in July of 2015, just sort of in an update. So, for example, Windows 7 wasn't ever vulnerable to it because it only affects 10, Server 2016 and Server 2019. So it's things that have been, well, it's things that have been updated recently in this newer evolution. Also Firefox is not vulnerable because they carry their own NSS. They've always had their own security suite.

Leo: Oh, interesting.

Steve: Google Chrome was, but they quickly updated Chrome in order to check for it. But the vulnerability is pervasive. And I'm sure we're going to see some exploits of it. So anyway, a lot of fun stuff to talk about.

So our Picture of the Week is just kind of fun. It's not really apropos of anything except that it plays off of the crazy world that we're in now. It just shows the front of a classroom with a teacher standing on one side of the blackboard, and a little kid looking at her, saying, "Before I write my name on the board, I'll need to know how you're planning to use that data."

Leo: I love it.

Steve: That's right. What are your plans?

Leo: It's true. Everybody's aware these days.

Steve: Yeah, yeah. Unfortunately, we're all aware because of the abuses of that in the past. And so we're trying to negotiate a compromise between the economic model that has evolved on the Internet of using this data because they can, and arguably individuals' more threatened rights to privacy than we've ever had before.

Leo: Yeah.

Steve: Anyway, iPhone has joined Android in being a qualified Google account security key. We talked about this last spring, I think it was last April, that Google allowed Android devices, essentially their devices, to double as a hardware security dongle, a physical token for use in providing multifactor authentication to Google's services. Now that feature is being rolled out to users of iOS v10 and subsequent devices.

Leo: So this looks like the single sign-on approval thing. Microsoft does this with their Authenticator, too. So when I log onto Windows, it pushes something out to Authenticator that then on my device, on my phone says, hey, is that you logging in at IP address blah blah blah.

Steve: Yup.

Leo: And you say yes or no. Which is much better, I mean, from a point of view of the user, much easier than typing in a six-figure TOTP. Is it better, more secure than TOTP? It's probably the same; right? It's a push. Is it a push?

Steve: I think it's probably - yes. Yeah. So it's probably comparable security. What's interesting is that it does use Bluetooth, and Bluetooth must be active on both devices, both on the phone and the device you are wishing to authenticate on. But they do not need to be paired, so it does not require a pairing relationship. It just uses a little brief beacon-style communication.

Leo: See, that worries me because isn't Bluetooth somewhat insecure?

Steve: Well, Bluetooth is radio. On the other hand, your one-time token you're typing in on your keyboard; and we know that, I mean, there are all kinds of ways of hacking anything that involves your keyboard. So you can imagine something which is intercepting your keyboard is it captures your one-time password when you type it in and then sends it somewhere for them to authenticate before you're able to authenticate. I mean, so the whole issue of authentication at a distance is fraught with trouble.

Leo: So this is different than I was saying. So that push notification is just like a text message or an auth code where you're saying yes. What you're saying is this is actually - my phone is physically an authentication device that, using Bluetooth LE, tells the computer, oh, yeah, that's him.

Steve: Yes.

Leo: Does it involve any interactivity on my part? Do I have to press a button on the phone or anything?

Steve: Yes. Yeah, you do have to explain. You have to confirm physically that...

Leo: I have to unlock it.

Steve: So your phone has to be unlocked.

Leo: I have to unlock it.

Steve: Exactly. It's got to be unlocked, and then you've got to say, yes, that's me.

Leo: So is it better than a YubiKey, for instance?

Steve: No. I would say the advantage is it is similar to a YubiKey, but it's probably one that you already have at home.

Leo: You already have it, yeah.

Steve: So now you've got both Android and iOS devices are part of this slowly expanding, I'm sure, carefully expanding range of devices like the YubiKey, like the Titan, which is Google's version of that. So you download and install this Google Smart Lock app. So there is an app that you need to install on your phone. For anyone who's interested, I've got the link in the show notes. But I'm sure if you just put in - I'm not sure because iOS or the App Store is so bad about finding things. But Google Smart Lock application is what it's called. You do need to turn on two-step verification or Advanced Protection on your Google account. So you've got to go to your - in your browser, you sign into Google. You'll be asked to...

Leo: When they did this, by the way, I then decided to turn Advanced Protection back on because it made it more convenient for me.

Steve: Right, right. Otherwise it was just - it was a pain in the butt to have to, like, oh, crap, okay, now I've got to go do that. So then you'll be asked to sign in again to reprove that you're doing this, that it's you doing this, even though you've already signed in. Then you scroll down. I was a little puzzled, but it's there.

You scroll down to find "add security key" because essentially, even though it's not what we have thought of in the past as a security key, we've thought of that as a YubiKey, this is turning your phone, when it's equipped with this Google Smart Lock app, into a physical security key. And among your options there you'll find iPhone. You'll say, yeah, you want to add your iPhone. So you click on "add." Then you enable iPhone security key by tapping "Yes, I'm in," when prompted to in the Google Smart Lock app. And so that sort of sets it up.

And from that point on, as long as you've got Bluetooth active on both devices, you sign in on a browser using Chrome OS or a browser running iOS, macOS, or Windows, or I'm sure Android. And then in this case you would check your iPhone Smart Lock app where it should be saying "Is this you," and you verify by tapping "yes."

Leo: So let me do this, then, because I haven't done it yet. You can see I've already used my Google phone that way.

Steve: Right, right.

Leo: And I have a bunch of YubiKeys. I like having more than one YubiKey in case I lose one, and I keep the other ones in a safe secure location.

Steve: And I should note that, even Google says, once you've done this, you should create another backup security key. And I don't know what that means. But the problem would be if anything happened to your phone, you didn't have it with you, and you needed to sign in, well, I mean, this is always the problem with tightening authentication is that it keeps the bad guys out, but it also keeps you out unless you're able to prove...

Leo: I can't use both my Google Pixel and my iPhone.

Steve: Whoa. That's interesting.

Leo: So it says you may only have one built-in security key on your account. That makes sense. You don't want to have too many of these, either. Since I have backup with my YubiKeys, I guess I'm going to go for that.

Steve: That's interesting.

Leo: And for purposes of demonstration. And then it just, well, you know, it didn't - maybe because I have Smart Lock open, it didn't do anything on Smart Lock. It just added it. But I guess it says, well, you've already verified. By the way, it's not my 10s because - but I guess - this is a little concerning. 10s is an old phone. This is now an 11. Named 11. It doesn't say 10s anywhere. But I think Google still thinks it's my 10s. So I'm just going to live with that. That is not really reassuring. But anyway, okay. Google knows what it's doing.

Steve: Well, and this does, I mean, again, for example, I'm using a one-time password for my Google access, and I have long been doing so. Google is good about allowing my accounts to be sticky. They're good about notifying me if I have signed in on a machine, on something I have that they haven't seen before.

Leo: Yeah, yeah.

Steve: So that's all good. And I do have, you know, I'm using OTP Auth. That's my favorite app on the iPhone. I've talked about it before. I like it because it gives me some flexibility. I can create folders. It has an appearance in the widgets on iOS, so like on my lock screen or on the whatever it is where you slide to the left from the home screen. I'm able to access it there. I'm able to put the few things that I use most there. I'm also able to just have it copy the OTP into the clipboard, if I'm logging in on that phone.

Anyway, OTP Auth is the app that I chose there. That's what I'm using. And I'm not - I guess I should try this just because it's possible now, although all these limitations seem a little bit annoying. But, you know, I guess it would be - I guess my point is it's not something that I'm doing often. I don't find myself often having to log in to Google because my computers are secure, and Google remembers me there on a persistent basis. But when I do, I'm just using a six-digit one-time password. But I did want all of our listeners to know, for those who are interested in physical token security, your iOS device is now considered a physical token.

And when I was doing some of the reading into this, I learned something, Leo, I never knew, and I was impressed. We've been talking a lot in the last year about the scourge of phishing attacks. You know, phishing relies upon and preys upon human fallibility to perpetuate successful attacks. Gmail blocks more than, get this, 100 million phishing email attacks per day.

Leo: Wow.

Steve: A tenth of a trillion. Wait, a tenth of a billion. 100 million phishing attacks per day are spotted and blocked by Google. So just, you know, that's significant because human fallibility is inherently porous. I've adopted that term "porous" to talk about security because I think it properly conceptualizes the nature of security, is if you push hard enough, if the pressure is enough, you're going to get - something is going to squeeze through a leak somewhere.

And so security is not perfect. It's porous. And the harder you push, the more you get. And so, boy, being able to block all of that email that someone might click on, there's a certain percentage of people, they just, I mean, even if you're trained up and educated, you're not paying attention, someone interrupts you while your finger's hovering over the button, and you click it when you didn't mean to, and, ooh, that's all it takes.

Leo: You know I'm constantly talking to our IT department, saying what are we doing, because we've got people in the front office who are maybe not as sophisticated as some. And as you point out, anybody's vulnerable to this. And I always worry we're going to get hit by ransomware. One of the things we do, and this is because Gmail does such a good job, is we use G Suite as our corporate Gmail. Our TWiT.tv addresses all go through Gmail. So that's step one.

Steve: So you get the advantage of its filtering, yes.

Leo: Yeah. We do a lot of other things. But that alone is pretty reassuring. They catch a lot. If you're using Gmail, you're not seeing nearly the number of phishing attacks you would be otherwise. Maybe none.

Steve: And I do maintain a Gmail account. I've had one forever. And sometimes I'll look over in the spam side because I'll look at all the stuff that makes it through. And I'm thinking, god, there's a lot of crap here. Then I look over at what didn't get in.

Leo: It's a lot more.

Steve: It's like, whoa.

Leo: I think they do a really, really good job, both of spam filtering and threat filtering.

Steve: Yeah. So I did want to sort of just kind of - we're going to be keeping an eye on what is happening with the U.S. versus Big Tech as regards encryption. There was some continuing back-and-forth last week. We started the discussion of it last week. We know that Bill Barr, our U.S. Attorney General, wrote the letter to Apple's counsel saying we need you to unlock these two phones. Bill Barr in a press conference said: "We have asked Apple for their help in unlocking the shooter's iPhones. So far," this is Barr talking, "Apple has not given us any substantive assistance."

And I suppose that the characterization as "substantive" is somewhat open to interpretation. This is why people hate attorneys and politicians. But based upon the level of Apple's arguably quite substantive cooperation, Barr must have meant that Apple didn't give them absolutely everything they asked for.

Apple replied to Barr's characterization by saying: "We reject the characterization that Apple has not provided substantive assistance in the Pensacola investigation. Our responses to their many requests since the attack have been timely, thorough, and are ongoing. Within hours of the FBI's first request on December 6, we produced a wide variety of information associated with the investigation.

"From December 7th through the 14th" - which was last Tuesday - "we received six additional legal requests, and in response provided information including iCloud backups, account information, and transactional data for multiple accounts. We responded to each request promptly, often within hours, sharing information with the FBI offices in Jacksonville, Pensacola, and New York. The queries resulted in" - get this - "many gigabytes of information, which we turned over to investigators. In every instance, we responded with all of the information we had."

And Apple added that it had received the subpoena for the second iPhone on January 8th and responded to the FBI's request within hours of receiving it. So in Barr's somewhat unfortunate press conference he said: "This situation perfectly illustrates why it is critical that investigators be able to get access to digital evidence once they have obtained a court order based on probable cause. We call on Apple and other technology companies to help us find a solution so that we can better protect the lives of Americans and prevent future attacks."

So standing back from this a bit, this sounds more like Barr just continuing to bang the drum for change, and not actually expecting anything else from Apple.

Leo: He knows they can't do anything.

Steve: Yes.

Leo: And they've given him everything they can, which is more than enough.

Steve: Yes. It's more.

Leo: So this is, no, political at this point. This is not about that investigation. And really, as far as AG Barr goes, I seriously think that this is not about terrorism. This guy wants to suppress dissent. He wants to know everything that's going on. And he's the flag bearer for the absolute most intrusive kind of surveillance. It bothers the hell out of me.

Steve: We did learn a little bit more during this follow-up press conference. Barr said: "During the gunfight with first-responders, the shooter disengaged long enough to place one of the phones on the floor and shoot a single round into the device. It also appears the other phone was damaged. But our experts at the FBI crime lab" - which I find amazing - "were able to fix both damaged phones" - yes, you can shoot it, and we'll still be able to bring it back to life. Anyway, so they are operational. Maybe the bullet just bounced off its corner or something. Anyway.

He said: "However, both phones are engineered to make it virtually impossible" - and I would remove the word "virtually" because we know how they've been engineered - "to unlock them without the password. It is very important to know with whom and about what the shooter was communicating before he died."

So I agree with you, Leo, completely. We've previously covered extensively here that Apple has deliberately designed and implemented within their devices a robust system for encrypting the data stored within iDevices' nonvolatile storage. And the upshot of that design is that no one, not even they, Apple, have any means or ability to decrypt it. And as you said, I'm sure they explained this at great length in their replies. I'm sure that, you know, we didn't see the subpoena that was issued that Apple responded to providing all the information they had, gigabytes of data.

Leo: Yeah, they gave them the iCloud. They probably gave them the Apple iMessages. They know who he was talking to because the phone company knows that, as well as Apple. By the way, there's also a point to be made. There's some evidence that even the most recent iPhone 11 can be cracked by GrayKey. Cellebrite and GrayKey we've talked about before, are firms that specialize in helping law enforcement break into these phones. And there's significant evidence that GrayKey has an ability to break into the iPhone 11, the most recent. And we know, thanks to - what was that exploit that we've been talking about on older iPhones? We know that they can root the older iPhones. So even - and I think Barr knows this. This has nothing to do with this case.

Steve: Right, right.

Leo: There's a good post in the Lawfare blog saying they already got everything they could ever want about these phones.

Steve: Well, and look at how much they got. The other flipside is law enforcement is now buried in riches.

Leo: Yeah, they wouldn't have had this before. They've got everything. They didn't need a wiretap. They got it all.

Steve: Yes.

Leo: So this is to me a misleading - this is a red herring. What they really want is totally information awareness, Chinese government-style surveillance on all the citizens.

Steve: We know that if Bill Barr wishes to outlaw the commercial sales of such technology in the U.S., he'll need Congress to pass legislation to essentially outlaw the commercial sale of technology that doesn't have a backdoor.

Leo: This is political, but I think they want to outlaw Congress, too. So I think where they're heading is we'll just take care of this.

Steve: Those pesky congressmen.

Leo: Those members of Congress.

Steve: So until then he's just complaining that Apple cannot do something that was deliberately designed for it to be impossible for it to do. So, okay, yeah, fine.

Leo: And he knows that.

Steve: Yeah.

Leo: This is grandstanding. This is all about the court of public opinion. It has nothing to do with the actual case.

Steve: Yeah. And we saw somewhere that some other congressmen, I think it might have been Lindsey Graham, who while railing against Apple he said, "But I have to say, I am happy that my iPhone is secure."

Leo: Yeah. For him.

Steve: Like, wait, Lindsey. Hold on a second. You're saying you want a backdoor, but not in yours.

Leo: Right. Of course. That's exactly what he wants. You nailed it. You're surprised?

Steve: Yeah, I think it'll be really interesting. It'll be - this just really seems like a hot potato because, yes, it would be nice if only terrorists' phones could be unlocked, and everybody else's were safe. But what citizen is going to say, yeah, I want the government to be able to get into my phone whenever it wants to. I just don't see us, I mean, we've been talking about cryptographers being in their ivory towers saying, no, there's no way to do this. I think when you ask anybody would you like the government to be able to have access to your phone, well, certainly some people will say, yeah, I have nothing in here. Others are going to say that doesn't seem like a good idea. So we'll see.

Leo: Yeah.

Steve: Until then. Over the weekend, two days ago, news dropped of a new serious Internet Explorer zero-day remote code execution exploit. Now, this is, okay, first of all, this is IE. It's not the default browser that most people are using anymore. But it is something that has an Internet-facing surface. And the idea that there is a remote code execution exploit is big news. Of biggest concern are the fact that one in four desktops running Windows 7 today, since none of them will presumably ever be patched unless their users have switched to Chrome, Firefox, or another browser, IE11 is the last IE available for Windows 7, one in four desktops still operating on the Internet today. And even if another browser is set up on those systems by default, it's still possible to explicitly invoke IE to take over the machine.

Leo: And I still think, I may be wrong, but doesn't Microsoft ship IE on every copy of Windows, including Windows 10?

Steve: Yes.

Leo: Yeah, it's still there.

Steve: It is still present. And it is vulnerable. We don't know whether Microsoft may issue an out-of-cycle patch. But it is a zero-day. It was discovered being used in targeted attacks in the wild. So it's not a theoretical issue. I mean, look at how much fur flew from Spectre and Meltdown, and that never resulted in a practical attack. This was found being used.

Okay. So here's the technical details. There are two JavaScript libraries in Windows. There's the original JScript.dll, and there's JScript9.dll. JScript9 is the default JavaScript interpreter that IE9, 10, and 11 use. It is not vulnerable. So it's the non-default earlier JavaScript DLL which, of course, because Microsoft's good about not killing off legacy stuff, there were some sites that were depending upon some quirks and characteristics of the original JavaScript library. They are able to explicitly invoke the older Jscript DLL.

So IE9, 10, and 11 are potentially vulnerable. And it doesn't help us that it isn't the default DLL, the default JavaScript interpreter, because a malicious site that wanted to exploit the older JScript DLL can ask for it explicitly. It can say this is the JavaScript DLL I want to run. And so that's what a malicious site will do. And note that, because we allow ads to run JavaScript, it can be any advertisement that comes up on a well-trusted site can also potentially exploit this.

Leo: So really it shouldn't be - it's not really an IE exploit, technically. It's an exploit of the JavaScript DLL that comes with IE and is present on all Windows machines. Is that right?

Steve: Correct. Although I'm only seeing it referred to as IE9, 10, and 11. For example, Edge might not be willing to invoke it.

Leo: Oh, I'm sure not, yeah.

Steve: Yeah. And so...

Leo: So you have to first have IE open, and you can only invoke it from IE.

Steve: Correct.

Leo: Ah.

Steve: Yes.

Leo: But you can get IE open; right? You can say...

Steve: Exactly. Exactly. In fact, I'm sure all of us who have been using Windows sometimes see that IE starts up for some reason. It's because it is possible for apps or sites to say, oh, no, we need this under Internet Explorer, which is why it's still around, because from time to time it ends up being needed. So we know that Windows 8.1 and 10 will be updated, though it's unclear when.

We just received our monthly Windows booster shot, and it's unclear how serious Microsoft is taking this one. We don't know, as I said, whether we're going to see an out-of-cycle emergency patch before next month's February patches. It is being tracked as moderate. It's CVE-2020-0674, called "moderate," even though it is a zero-day, in the wild, targeted attack, remote code execution vulnerability being used right now.

So it has not yet been patched. And my concern is that 7 may never get patched, and 7 doesn't have Edge as its default. It's stuck with IE11 as the latest browser. So one in four Windows systems stands to never get fixed, unless Microsoft thinks better of deciding not to fix Windows 7 any longer and thinks, well, this is bad enough, I mean, it's a zero-day. It's not theoretical. It's being exploited in the wild.

Microsoft's advisory says: "The vulnerability could corrupt memory in such a way that an attacker could execute arbitrary code in the context of the current user. An attacker who successfully exploited the vulnerability could gain the same user rights as the current user." And remember that this is the coy voice that Microsoft uses. They found this being done in the wild, not because it could be, but because it was being.

So they continue, saying: "If the current user is logged on with administrative user rights, an attacker who successfully exploited the vulnerability could take control of an affected system. An attacker could then install programs; view, change, or delete data; create new accounts with full user rights. Microsoft is aware of limited targeted attacks in the wild and is working on a fix. But until a patch is released, affected users have been provided with workarounds and mitigation to prevent their vulnerable systems from cyberattacks." Okay, now, that's sort of an odd thing to say, "affected users have been provided."

Well, okay. Only if you do a bunch of hocus-pocus mumbo-jumbo. I mean, it's not like it's being delivered to you. You have to, like, be listening to this podcast, or be checking in on Internet news things. But that brings me to my next point, the workarounds. It's possible to quickly and easily disable all access to this vulnerable JScript.dll. I immediately did it since I never deliberately use IE anymore, and any IE-dependent sites barely legitimately use JScript.dll any longer. So for me, and I expect for most of our listeners, the chance of encountering any problem is next to zero. That is, any problem where having disabled access to JScript.dll would cause a site not to work. Whereas the probability of it being used against us maliciously is significant.

And notice that it was used in limited targeted attacks until it was discovered. Now whoever's using it knows it's out there, and it's a race against the clock to leverage it until, for like maybe for a month, until Patch Tuesday of February. So the fact that it has only been seen in limited targeted attacks, now that we know it exists, the wraps are off. So suddenly the threat, you could argue, the threat jumps hugely.

I have here in the show notes a picture I took of my JScript.dll properties last night, showing that the Everyone user, meaning everyone, has had everything, all rights denied. No modification, no read and execute, no read, no write. That is, this JScript.dll cannot be invoked by the system, by IE, by anybody. Also in the show notes are some commands that you need to use on a 32-bit system. There is a command to take

ownership of the JScript.dll, then a second command to set the everyone user to deny all. On 64-bit systems, since they both have a 32 and a 64-bit JScript.dll, you need to do that both in the 32-bit context and the 64-bit context so that the commands are a little bit, well, there are four lines that you need to execute.

This is all in the show notes for anybody's who interested. I would do it. Our listeners know I'm not running around with tinfoil. But there's reason to believe that for a month, if Microsoft doesn't do something more quickly, that everyone will be exposed to this malicious DLL which can be evoked through IE. And IE can be evoked if it's present in your system. So I would think it's worth removing these rights. It's a couple commands. They can be reversed easily. I've got the reversal commands also here in the show notes. It's unlikely you will ever encounter a problem after disabling it. And then Microsoft's presumably going to fix it by next week.

Leo: Actually...

Steve: Go ahead.

Leo: Bleeping Computer does say there are some potential issues you might experience. Windows Media Player might have trouble with MP4 files. The FSC Tool will not replace JScript.dll with altered permissions if you want to fix your system files. And printing to PDF from Microsoft Print is supposedly reported to break. So there may be some weird...

Steve: That's interesting.

Leo: Well, what it shows you is that this DLL is being used.

Steve: Yes, interesting. And of course I'm mostly concerned about people who are deciding they want to stay on Windows 7. It's never going to get fixed, if Microsoft holds to their commitment.

Leo: This will be a good test.

Steve: Yeah.

Leo: A zero-day, like, a week after they stopped updating it. Hmm.

Steve: Yes, exactly.

Leo: Hmm.

Steve: Uh-huh, yeah. And, you know, and not a good thing because they've got a 25%, as we say, one in four desktops are still running Windows 7. And that number may drop

in time. But you have to think, at this point, they're not going to be moving to Windows 10 even though it's free.

So giving Windows an additional Edge, speaking of browsers. We were just talking about IE. Microsoft also last week, as we were expecting, took the wraps off their new Chromium-based Edge browser. So I guess Paul and Mary Jo can now stop referring to it as Chredge.

Leo: Never.

Steve: I've got the link to - but it's not being pushed out in Windows Update so Windows 10 users did not get it automatically last week. There is a link in the show notes to get it. But I also created one of my little GRC shortcuts, grc.sc/edge, grc.sc for shortcut, grc.sc/edge. That will take you to the page allowing you to download it. And it is, I have to say, it's pretty looking. It's available right now for Windows 7 - yes, Windows 7 - 8.1, Windows 10, and iOS and Android. So you can download it for any of those platforms.

Leo: And Mac.

Steve: Okay, I didn't know that. And Mac, cool. And of course, as we know, this Chromium-based Edge browser abandons Microsoft's own home-rolled Edge HTML rendering engine in favor of Google's open source Chromium.

Leo: That's an important distinction, by the way, because people are saying, well, just get Chrome. It isn't Chrome. It's the Chromium renderer engine; right?

Steve: Right, right. It's the guts.

Leo: It's the guts.

Steve: It's the heart of Chrome. And a perfect example is that Microsoft has added some additional features to it.

Leo: Right. And taken some out, I would bet.

Steve: Yes, exactly, yeah. Well, they de-Google-ized it in order to make it theirs. I've installed it, and it's quite attractive. But there are also some nice features, or at least there are some plans to have that. Under the Settings screen, under Privacy and Services, there's three flavors of tracking prevention: Basic, Balanced, and Strict. And so mine is now set to Strict because why not? There's also supposed to be something a little bit below that on that page. I've seen screenshots of it, but mine didn't have it. It's "Block potentially unwanted apps," which sounds like a setting I would like to have. Maybe it's only available in the canary versions and not yet in the one that you download. But it sounds like that's going to be coming eventually.

Also there's the ever-popular - oh, thank goodness - "Block media autoplay," which is not enabled by default. You go to edge://settings/content/mediaautoplay to get you to the

page that enables you to turn on the blocking to prevent pages from autoplaying any crap that you didn't ask for when the page loads. So yay for that. So anyway, this is Microsoft, as we've said, who just decided that it just didn't make any sense for them any longer to be constantly running in parallel, trying to keep up with the platform which more and more browsers are adopting. You know, when we were talking recently about all of the browsers that - I don't remember now which feature it was that Chromium had. It was like, whoa. And what's the one with "V"? Vivaldi.

Leo: Oh, Vivaldi.

Steve: Vivaldi and Brave and, I mean, Firefox is still the holdout, doing their own thing, and it's nice because...

Leo: You don't want a monoculture. You want...

Steve: You really don't. You do not. Yes, exactly that. We really don't want to have a monoculture because that you can always switch to a different browser makes sense. And where you're not just same guts with a different surface. And of course I'm still in love with my side tabs that Firefox is alone in offering in an integrated fashion. There's, like, weird kind of a sidecar thing you can get for Chrome. But no. I just want true side tabs. And so far, well, and I like a lot of things about Firefox, as well. Although these newer browsers, Chrome and now Edge, they really look nice.

From our, well, that's no surprise department, Bleeping Computer reports that the FBI has sent a flash security alert that nation-state actors have breached the networks of U.S. municipal government and a U.S. financial entity which remains unnamed by exploiting the critical authentication bypass vulnerability in those Pulse Secure VPN servers. We talked about this at length last week. The FBI says that unidentified threat actors have used the Pulse Secure VPN authentication bypass flaw to "exploit notable U.S. entities," although without naming them, since August of 2019.

They said: "In August of 2019, attackers gained access to a U.S. financial entity's research network by exploiting unpatched VPN servers. The same month, a U.S. municipal government network was breached in an attack that exploited the same vulnerability. Based upon the sophistication of the so-called Tactics, Techniques, and Procedures" - they actually have an acronym for that, the TTPs, the Tactics, Techniques, and Procedures - "used in the two attacks, the FBI believes unidentified nation-state actors are involved in both compromises; however, it remains unclear," they said, "if these are isolated incidents."

So according to the FBI, the attack that targeted and compromised the U.S. municipal government network took place in mid-August of 2019. This attack, quote: "The operators enumerated and exfiltrated user accounts, host configuration information, and session identifiers that could allow them to gain further access to the internal network."

In the other known attack, they said, quote: "The intruders remotely exploited a Pulse Secure VPN appliance." The flash alert said: "The vulnerability in Pulse Secure allowed directory transversal and access to a file where login credentials were written in plain text. In addition, the Pulse Secure appliance may have been vulnerable to a buffer overflow and command injection. After breaching the network, the nation-state actors gained access to the Active Directory, harvesting and exfiltrating user credentials, the usernames and passwords, for the VPN client."

They said: "Following attempts to enumerate and gain access to other network segments, the attackers were only able to infiltrate the exploited segment, which was the only one on the network using single-factor authentication." They said: "The intruders attempted to access several Outlook webmail accounts, but were unsuccessful due to the accounts being on separate domains requiring different credentials not obtained by the intruders. While the intruders performed additional enumeration, there was no evidence that any data was compromised or exfiltrated, and the intruders seemingly did not install any persistence capability or foothold in the network."

So in the flash notice, the FBI did not directly connect these attacks to Iranian-based hackers. A private industry notification detailing Iranian cyber tactics and techniques shared a day later mentions information indicating Iranian cyber actors have attempted to exploit the same vulnerability. And the FBI assesses this targeting, which has occurred since late 2019, is broadly scoped and has affected numerous sectors in the United States and other countries. The FBI has observed actors using information acquired from exploiting these vulnerabilities to further access targeted networks and establish other footholds, even after the victim patched the vulnerability. So that, of course, jives with what we talked about and had seen last week.

So what's clear is that today we use the term "threat landscape." And it's clearly not hyperbole. We don't just face a situation where we have opportunistic hackers, as is often said, operating out of their parents' basement. Cyberwarfare is a real thing. And there are serious players scrutinizing every announced vulnerability, just waiting for an opportunity to gain an advantage. So when something like a VPN authentication bypass is announced, as it was in this case in April of 2019, I mean, the guy's doing this Pulse Secure VPN, okay, well, they had a flaw. They fixed it. They announced it. They offered an update. I'm sure they tried to notify people who were their customers, who were still able to receive these notifications.

What we saw was that, despite that fact, there were still 15,000 instances of this insecure VPN operating out on the Internet. So here we've got Iranian, apparently Iranian cyber hackers. They're taking this stuff seriously. They see this announcement, they go enumerate the Internet, find vulnerable endpoints, use the flaw, reverse engineer the fix, figure out a way of bypassing authentication, and then crawl inside the networks, which are behind these VPNs. And in this case, as the FBI's flash alert indicates, there were some significant U.S. properties which were using this VPN and, to their shame, had not updated, given six months to do so. Still had not updated.

So in one case we know that that international foreign currency exchange, Travelex, was - and we talked about it - hit by the Sodinokibi ransomware on December 3rd after not patching seven of their Pulse Secure VPN servers. Travelex was one of the companies that the Bad Packets guys warned of having vulnerable servers four months earlier, back on September 13th of 2019. Travelex never replied to the email warning that they received. And they got hit four months later.

So I don't know how we solve this problem as an industry. We have Adobe, who has for years been responsibly pushing out updates to their products every month, often on the second Tuesday. Of course Microsoft. There are an increasing number of examples of patches being made available and pushed responsibly. We've said here that SoHo routers need to be doing this themselves. It's no longer enough to require that users go and click on, log into their router and then check for an update and then update their router's firmware. These things are unattended, unsupervised, IoT devices, essentially. They're on the Internet. They need to go and update themselves.

So I just, you know, as an industry we have to keep moving in this direction because it's clear that, again, it's not opportunistic hackers. We really are now operating in an environment where any flaw found is followed up on by teams that are being paid, often

by their governments. And I'm not looking for a job, but wouldn't that be fun? Imagine being able to do that, and it's not illegal. It would be, you know, you're on the good guy side when you're doing that kind of serious cool hacking of other entities on the Internet. Wow. That would be really neat.

Okay. So something that slipped under the radar, and I'm kind of amazed by this, I didn't see anyone talk about this. The U.S. Department of Homeland Security, CISA, published the alert AA20-014A titled - this is last week, on Tuesday: "Critical Vulnerabilities in Microsoft Windows Operating Systems." And I have the alert, a link to the alert in the show notes.

Essentially what I think happened is that this more, I guess more interesting and also serious CryptoAPI vulnerability that we'll be talking about in a minute, which the NSA discovered in Windows 10 and Windows Servers, which Microsoft, okay, Microsoft rated that one as being merely "important." And of course it's now been dubbed "CurveBall." That obtained the lion's share of the attention last week. We'll be looking at that next.

But this CurveBall obscured an entirely different vulnerability that Microsoft rates as critical for reasons we will see. And believe it or not, it is another pre-authentication remote code execution vulnerability in Microsoft's Remote Desktop Protocol, in this case in something known as the Remote Desktop Gateway. There's a link. I also have a link to Microsoft's own security guidance.

They said, in their disclosure of this last Tuesday: "A remote code execution vulnerability exists in Windows Remote Desktop Gateway when an unauthenticated attacker connects to the target system using RDP and sends specially crafted requests. This vulnerability is pre-authentication and requires no user interaction. An attacker who successfully exploited this vulnerability could execute arbitrary code on the target system. An attacker could then install programs; view, change, or delete data; or create new accounts with full user rights." Doesn't that sound familiar.

They said: "To exploit this vulnerability, an attacker would need to send a specially crafted request to the target system's RD Gateway via RDP." This affects Windows Servers 2012, 2012 R2, 2016 and 2019. So all the servers that are being maintained. It also turns out that this same RD Gateway is in Windows Server 2008 R2. We don't know whether it also affects it because Microsoft stops saying that anything that they're not supporting is affected, even when it may be. So we don't know.

Under "Mitigations" they say: "Microsoft has not identified any mitigating factors for this vulnerability." Under "Workarounds" they say: "Microsoft has not identified any workarounds for this vulnerability." Under their FAQ: "What network ports are vulnerable to this attack?" They say: "The vulnerability only affects UDP transport, which by default runs on UDP port 3391."

So as our listeners will recall, it hasn't been a year, it was back on May 14th of 2019 that we first learned of the now widely known BlueKeep vulnerability. Microsoft's disclosure back then was CVE-2019-0708, and that was titled "Remote Desktop Services Remote Code Execution Vulnerability." And I have it here in the show notes, the language from that disclosure. It's identical to the language that I just read. Nothing has changed. So, yes, it sounds familiar. This exploit is against the server-oriented Remote Desktop Gateway, which is, we can hope, much less widely deployed. But it turns out there are plenty of them, too.

So we're back here once again with a very serious vulnerability which will surely result in another round of exploitation. Kryptos Logic, spelled K-R-Y-P-T-O-S Logic, has examined the DLL that was repaired last Tuesday in this update. In their introduction, they explain the intent of the Remote Desktop Gateway. They said: "Remote Desktop Gateway (RDG),

previously known as Terminal Services Gateway" - that's as a consequence of the fact that it's been around a long time; and, yes, it affects Server 2008 - "is a Windows Server component that provides routing for Remote Desktop. Rather than users connecting directly to an RDP Server, users instead connect and authenticate to the gateway. Upon successful authentication, the gateway will forward RDP traffic to an address specified by the user, essentially acting as a proxy. The idea," they say, "is that only the gateway needs to be exposed to the Internet, leaving all RDP Servers safely behind the firewall." Whoops. Except the firewall is broken, in fact introduces a new vulnerability.

"Due to the fact that RDP is a much larger attack surface, a setup properly using RDG can significantly reduce an organization's attack surface." Except when it creates a new one. Anyway, I have a link to the full disclosure. They completely reverse engineer the vulnerability, show exactly what it is that's going on, find and tell us about the problem. It turns out, remember before I mentioned that the vulnerability was only over UDP. Well, this is yet another problem associated with UDP packet fragmentation and out-of-order reassembly after receipt. We've talked about UDP packet fragmentation and reassembly problems many times in the past. It has historically been a huge headache for IP stack implementers. It's inherently prone to having mistakes made. And in this case it has bit us again.

The idea is that with UDP, the Datagram Protocol on the Internet, a large packet can be emitted, and you can, as the packet transits the Internet, encounter a link from router to router, an inter-router link, where the protocol doesn't support such a large packet. So the router that is unable to forward the packet that's that big is able to fragment it, is able to chop it into smaller pieces and send them in pieces. There is a fragmented flag in the packet, and a which fragment of the larger packet this fragment is, added to the fragmented UDP packet.

The problem, the dilemma essentially, is that once packets are fragmented, they are never reassembled. They then make their way the rest of the way over the Internet as fragments. They are also, as is the case for UDP, UDP packets themselves are allowed to come in out of order, and they need to be reordered. Fragments are allowed to come in out of order because they are themselves just UDP packets that carry this fragmented bit. So you need some buffer on the receiving end in order to receive packets of a fragmented UDP packet out of order and wait for them all to arrive. Turns out that that logic of doing that properly is inherently problematical. It's been a constant source of problems. And as I said, here was a new one.

The problem and its disclosure is only a week old. It has been patched, that is, this was fixed, except for Windows Server 2008 R2, last Tuesday. But we also know that these things take a while to get fixed. It may be that the RDP servers on the Internet were not attacked before; only the RDP servers not behind the gateway were being attacked. Well, now the gateways can be attacked. And we have a complete reverse engineering of the problem is now public.

So we can expect another round of problems. And maybe we're going to give it a cute name. Who knows what this one will be called. Last one was BlueKeep. Here we have yet another problem. And I will say again, the only way to do this safely is to put this behind a secure VPN in order to keep RDP and the Remote Desktop Protocol Gateway off of the Internet.

Leo, when you were at CES I shared with Jason and our listeners the happy news that there is now native support for Drupal for SQRL.

Leo: Oh, nice.

Steve: And I just wanted you to know that. I just put this back in the show notes because a contributor of ours, Jurgen Haas, said: "Happy to announce that I was finally able to finish off the SQRL integration into Drupal 8 and the forthcoming version 9. It is feature complete and supports all of the SQRL Protocol v1."

Leo: That's great.

Steve: And in fact you just go to Drupal, and you're able to add SQRL to your Drupal installation. So that was very cool.

Leo: Thank you.

Steve: I just wanted to make sure you knew that.

Leo: Yeah.

Steve: I also mentioned last week that I was considering revamping GRC's certificates. That happened since I last spoke to our listeners on the podcast. I had a bunch of EV certificates. Every certificate I had was Extended Validation, all from DigiCert, of course, my absolute favorite Certificate Authority. But I had the GRC - my main GRC.com certificate was coming due and coming up for renewal. I've been using more and more certificates because I'm creating more subdomains of GRC, you know, sqrl.grc, blog.grc. There will be a forums.grc where we'll have web forums for supporting SpinRite 6.1. And I just like the freedom of being able to create subdomains.

And of course, as we know, EV certs are not allowed to have wildcards in their domain names. You can't do *.grc.com with an EV cert. That's not allowed. So that required individual EV certs or more and more subject alternative names in EV certs in order to stick with that. And of course browsers no longer celebrate or even disclose without any prompting that there is an EV cert. So you don't even get any credit for having them.

So I just decided, okay, time for a change. So it only made sense for me to switch. The process with DigiCert was as shockingly painless as every interaction with them has been. Since I was already approved for all the domains that I needed to bundle into the new single certificate, I simply used their Windows Certificate Manager. They have an app for Windows that makes it really simple to generate a Certificate Signing Request. I put the root domains and the wildcard domains I needed, basically grc.com, *.grc.com. I still have grctech.com, that's what brought me to talk about the cookies and cookie forensics last week, and *.grc.com. And of course grc.sc and a couple other miscellaneous domains. And I now have one cert that I have installed across all of my servers.

So thanks again to DigiCert making it easy. Oh, I forgot to say, after I generated that Certificate Signing Request, I went to DigiCert, clicked "yes," and I had a cert. It was like in minutes, like less than a minute. It's like, okay. Actually, I think it came up immediately in the web browser, and I was able to download it. And then I also received it through email. So hats off to DigiCert.

And I did want to mention that I pulled the trigger on recertifying all of my certs at GRC. And I'm really happy with having made that change. And I'm sad to see what has happened with Extended Validation certificates. I liked the idea of a person being in the

loop. As I noted before, I understand the value of automated certificate issuance for domain validation, as opposed to organization validation. This is an OV cert, not a DV cert, meaning that some higher level of authentication was applied to it. I get it that for opportunistic encryption, using the ACME protocol, an automated issuance cert makes sense. I wouldn't be surprised if at some point in the future browsers do start indicating whether a human was involved in the loop or not. We'll see what happens with that.

So, okay. We all understand the critical importance of the Internet's trust model.

Leo: Yes.

Steve: Which, with the caveats we often examine, for the most part works. How much fun have we had at the expense of the Hong Kong Post Office through the years. Yeah, and my startled discovery of how many certificate authorities were in my old XP system a long time ago, it was like, oh, my goodness, and the fact that it's a "trust all" model. Well, okay, so it got, you know, it's a little creaky. But for the most part it works.

We trust that we have established a private connection to a website because of the HTTPS certificates presented by the server, which are automatically verified by our browser, and are rejected with glaring warnings if anything at all seems wrong. Our browsers are very careful to remind us when we encounter a problem such as an invalid certificate.

And I'm sure we've all from time to time encountered a site whose certificate expired a day or two before. And it's like, it just happened to me, I don't know, a week ago, I was going somewhere, and I got like a warning. And I checked, I looked at the certificate, and it's like, oh, yeah, that was day before yesterday. Well, I'm sure they know they need to get a new certificate. But mostly we're protected. So we enter our credit card numbers, we interact with our banks, and essentially trust these technologies to keep us safe.

So that's, really, it explains why a fundamental flaw, a deep and serious flaw in the Windows OS Cryptographic API, that is, the thing that is the component responsible for doing that work, such that certificates could be spoofed, caused quite an uproar. It's being tracked as CVS-2020-0601. And it has upset the easy trust that we've developed in the model that we have. The good news is this upset is not the result of some fundamental flaw that's been found in the public key infrastructure, but this particular implementation on some Windows systems is flawed in a way that allows it to accept a spoofed certificate when it should not.

So as we know, last Tuesday Microsoft issued a security update to fix - and I don't understand still why they call this an "important" vulnerability. It's like, important.

Leo: Critical. Critical. Critical.

Steve: Yeah. I don't get it. But so this is in their Crypt32.dll. And what has caused lots of people to discuss on the Internet is the fact that this is the first time the NSA has stepped up and privately reported a vulnerability to accompany that, where they could have used that vulnerability themselves to their own advantage. I mean, they've done that before. They've done that in the past. That's what we learned that the NSA was doing, for example, with BlueKeep, is that they knew of a - was it BlueKeep? No. EternalBlue, sorry, EternalBlue.

They have used vulnerabilities themselves, kept them quiet, worked to not have them disclosed, in order to give themselves an advantage. Maybe it's because they felt this was too critical to go unfixed. I mean, because this is really bad. And I'll explain exactly how bad next. So the fix to this ensures that the Windows Cryptographic API library correctly completely properly validates ECC, Elliptic Curve Crypto certificates. So not all crypto certificates, but ECC certificates.

Okay. So this flaw, which now has been referred to or has been named CurveBall, was introduced into Windows, as I mentioned at the top of the show, in July of 2015. It's a spoofing vulnerability in the way that certificates are accepted without a correct verification of the explicit curve parameters provided within certificates. Essentially, the flaw allows an attacker to supply their own generated X.509 certificates, which is the class of certificates all of these are, by using an "explicit parameters" option to set its own curve parameters.

What the NSA wrote was: "Certificates containing explicitly defined elliptic curve parameters which only partially match a standard curve are suspicious, especially if they include the public key for a trusted certificate, and may represent bona fide exploitation attempts."

Okay. So translating that a little more into English, the way for us to think about this is that Microsoft's original code, which has been in place five years, was allowing certificates to over-specify their own elliptic curve parameters, which were then used to verify their own signatures. So in other words, the certificate was being allowed to say both "This is how you verify who I am" and "This is who I am."

So the flaw affects all of the places where we depend upon trusting certificates, including HTTPS connections, signed files and emails, and signed executable code. In essence, anything that is signed. And you can understand why the NSA said holy crap, you know, this is really too awful. If this were discovered independently, it would, well, you could imagine. I mean, it would be a huge exploit. So they really didn't have any choice. It completely subverted signing. All signatures.

So from an adversarial context, man-in-the-middle attacks, you would not have any way of assuring that you were actually connecting to any website that you believed you were because the signature of the certificate that your browser received would be trusted - unless you were using Firefox - when it should not be. And, oh, the good news is Windows 7 and Server 2008 R2 in this case, even though they received their final updates last week, they were never vulnerable in the first place. And unlike the famous Spectre and Meltdown vulnerabilities, which remained even two years later, essentially theoretical despite all of the uproar they caused, we never had working exploitable obvious proofs of concepts. We had, yes, a theoretical cross-process leakage, so in some settings that could be a problem.

In this case, immediately upon the disclosure, researchers jumped on this to create and release proof-of-concept code to demonstrate that the vulnerability can indeed be exploited in the wild. Researchers immediately predicted that it would only be a matter of days, and that prediction turned out to be spot-on. The first researcher to prove the vulnerability was exploitable was a guy named Saleem Rashid, who developed, or maybe that's a pseudonym, don't know, he developed a proof-of-concept code that permitted the immediate faking of TLS certificates. And of course that would allow anybody to create fake websites. He has not shared his code, but he tweeted some visual proof. I have a link to his tweet in the show notes. And we could say, well, okay, but that's a tweet that could be spoofed, that's true.

Immediately on the heels of that came published code from Kudelski Security, and that was followed closely by a Danish researcher Ollypwn, O-L-L-Y-P-W-N. Both now have

proof-of-concept code up on GitHub. Kudelski calls his the "Chain of Fools" because of course it's all about certificate chains. That's on GitHub. He also provides, on a blog posting that is synchronized with this, a complete explainer, in addition to a walk through his proof-of-concept code. So the cat is completely out of the bag. This is not theoretical. This is not, oh, maybe someone's going to figure out how to do it.

Ollypwn has a nice explainer in his GitHub intro about this. He wrote: "CVE-2020-0601, commonly referred to as CurveBall, is a vulnerability in which the signature of certificates using elliptic curve crypto is not correctly verified." He says: "ECC relies on different parameters. These parameters are standardized for many curves. However, Microsoft didn't check all these parameters. The parameter G, the generator, was not checked; and the attacker can therefore supply his own generator, such that when Microsoft tries to validate the certificate against a trusted Certificate Authority, it'll only look for matching public keys, and then use the generator of the certificate."

He finishes, saying that there's a cert: "MicrosoftECCProductRootCertificateAuthority.cer is by default a trusted root certificate authority using ECC on Windows 10. Anything signed with this certificate will therefore automatically be trusted." And of course this flaw allows a spoofed signature of that certificate to be readily created, and nobody has any questions now about how to do it. So we have publicly available code and exploitation in the wild. Well, we have publicly available code demonstrating the vulnerability and proof of concept code. Exploitation in the wild by malicious actors will surely follow.

Yes, it was patched on Tuesday. Well, we know in practical terms how much good that does. Again, Windows 7 systems that could not be patched, or, well, Windows 7 and Server 2008 also received their last patch, but they had nothing wrong with them. They're not broken in this way. Only subsequent Windows 10 and servers. There will be a patch delay. We know that will happen. It always does. There will be servers that will never be patched, which have received code since July of 2015 and then at some point stopped receiving it. This is going to get exploited. This is not good. And that's why this was such a big deal.

I'm sure that our listeners have installed the patches and rebooted their systems. They're probably fine. Again, Firefox never trusted any websites using that ECC certificate problem because Firefox brings its own crypto library along with it. Chrome briefly had the problem, but they were able to update themselves. Chrome updated to 79.0.3945.130. Once again, I had to go into Help>About, even though I was using Chrome yesterday and today, and it didn't update until I went to Help>About. Then it said, oh, and then it updated and restarted, and I got .130. And now I know that my Chrome won't accept the TLS certs. I think, however, that's like maybe the minimum, or a minimization of the problem. This is anything signed. So we trust code signed by trusted signers. Windows trusts code.

I did note, although I just saw this in passing, and I forgot to follow up on it, that apparently Windows' own Windows Update will not be spoofed by this. Yay. So maybe it uses a different approach. Maybe it's not using elliptic curve signatures. It might be using RSA, a large RSA signature, and so it's not vulnerable. The problem is that there is a vulnerable cert that is in Windows 10, is absolutely trusted, and anything signed with it or spoofed with a spoofed signature of it will also be trusted. So that's a problem.

So anyway, so I think that the website spoofing problem is less of a concern, although not, I mean, still, for systems not updated, for people using systems not updated, any site can now be spoofed. That's the case. So Windows needs to get fixed. But signatures are pervasive. And we're going to end up seeing some more exploits of this moving forward. So make sure your Windows is up to date, and you're okay.

Leo: Crypto CurveBall.

Steve: The Crypto CurveBall.

Leo: Not easy to hit the curve.

Steve: And thank you, NSA, for helping because this was too bad to leave out in the...

Leo: It wasn't a NOBUS.

Steve: For them to know and us to be hurt by.

Leo: I told you about NOBUSes last week; didn't I?

Steve: No.

Leo: There's a doctrine at the NSA - Michael Hayden, the former chief of the NSA, told folks about this - called NOBUS, N-O-B-U-S. An exploit that can be used by Nobody But Us is okay. He uses an example, look, if it takes four acres of supercomputers to crack it, then we don't have to tell the world about it. That's a NOBUS. Nobody but the NSA could use it. This was not a NOBUS. This was something any idiot, including a script kiddie, could take advantage of.

Steve: This was an omnibus. It was an all of us.

Leo: So don't pat them on the back too hard. They still do keep some stuff, hold some stuff back. But to their credit, if it's something relatively easily exploitable, they're not going to let that go.

Steve: No. And this is really bad. The ability just, you know, to get anything trusted by any Windows 10 system. So, whew. Yeah.

Leo: Whew. Whew. It's as bad a bus as you can get. Steve is the greatest; isn't he? We are glad you are here every Tuesday, 1:30 Pacific, 4:30 Eastern, 21:30 UTC. That's when we record Security Now!. If you want to watch us do it live, TWiT.tv/live. You can watch it live. You can also get copies of the show from Steve at his website, GRC.com. He's got 16Kb audio, 64Kb audio, and transcripts, so you can read along as you listen at GRC.com.

You'll also find lots of great stuff there, including SpinRite, his bread and butter, the world's best hard drive maintenance and recovery utility. You'll also find lots of freebies, including ShieldsUP!. People still - just a couple of weeks ago somebody called the radio show and said, "My ICMP port is open." Or actually he said, "It's not stealth, it's closed." And I said, "You've been using ShieldsUP!, haven't you."

Steve: 103 million and counting.

Leo: Wow. That's awesome. That's...

Steve: Yeah.

Leo: I use it any time I set up a network. I test it with ShieldsUP!. It's a great tool. Highly recommended. GRC.com. He's on Twitter at @SGgrc. You can follow him there, follow his tweets. He always posts the show notes before the show there. That's actually how I get them. But you can also DM him there. He's open to DMs. And if you have a question, a comment, a leak, a tip, information, @SGgrc.

We live at TWiT.tv. This show is TWiT.tv/sn. You can download shows there, as well. We have audio as well as video. And I guess the best thing to do - we're on YouTube, as well. The best thing to do would be subscribe in your favorite podcast client so that you get the show the minute it's available and have it just in time for your Wednesday morning commute.

Steve, thank you so much. And I'll see you next week on Security Now!.

Steve: Until then.