
# DNSpionage

**Description:** This week we discuss Google's use of their Sensorvault tracking to assist law enforcement. It's time to update Drupal again. And, speaking of "again," Facebook. We also look at Russia's newly approved legislation moving toward an Internet "off switch," a reminder that "USB Killers" are a real thing, the news of Marcus Hutchins's plea deal, an actively exploited Windows zero-day, a bunch of Microsoft Edge news, the Win7 end-of-life notices, something from the "I did say this was bound to happen" department, and some miscellaneous news. Then we examine the latest detailed threat research from Cisco's Talos Group about the leveraging of DNSpionage.

High quality (64 kbps) mp3 audio file URL: http://media.GRC.com/sn/SN-711.mp3
Quarter size (16 kbps) mp3 audio file URL: http://media.GRC.com/sn/sn-711-lq.mp3

---

SHOW TEASE: It's time for Security Now!. Steve Gibson is here. Lots to talk about. Google's Sensorvault: how the feds and law enforcement are using it to track us. Also the final fate of Marcus Hutchins, and an inexpensive little device that can kill any computer. It's all coming up next on Security Now!.

**Leo Laporte:** This is Security Now! with Steve Gibson, Episode 711, recorded Tuesday, April 23rd, 2019: DNSpionage.

It's time for Security Now!, the show where we protect you and your loved ones and your privacy, and we teach you about computers. And here's the guy who does it all, Mr. Steve Gibson of GRC.

**Steve Gibson:** And I'm giving the Vulcan salute with my right hand today, which feels a little awkward, but that's because the microphone is blocking my left hand access.

**Leo:** What's interesting about you is you are ambidextrous. I can only do the Vulcan salute with my right hand. And oddly, I'm a lefty, but I cannot get that finger to move over on my left.

**Steve:** In other news...

**Leo:** That's okay, change the subject. Go ahead. I know you never really cared about my Vulcan salute. I know, Steve.

**Steve:** For our listeners who don't know what we're talking about, there was a cherry picker operating outside of the window that provides the left illumination for me during the podcast all morning. And so I moved the microphone over onto the left side so it's pointing to the right because this microphone is very directional, and in the past we've had noises outside, and it does a good job. So anyway, that's the back story there.

I wish I could take credit, Leo, for the title of this podcast, DNSpionage.

**Leo:** Oh, I like it.

**Steve:** Because it's a beautiful name. But it's credited, or at least the first time I have seen it was in a Cisco Talos security research report which is the topic of the podcast. For a number, well, I was going to say weeks, but maybe it's months now, I've been sort of following this news. It never quite seemed like it rose to the level of getting into a lot of detail. But Talos just published a detailed research blog posting on the nature of and details of the nation-state level DNS Espionage, or DNSpionage, that they've been tracking all year. And it's really interesting, both from the fact that it's actually happening, that somebody is going to these extents.

This is all a consequence of the fact that the web has largely moved to HTTPS. It used to be, when we didn't have security up all the time, you could just bounce somebody, do a man-in-the-middle attack and intercept communications, and that was all you needed. Now you have to have certificates. And we briefly touched on this a while ago because I remember talking about how - and Cisco's Talos Group mentions this, how the automated certificate services, Let's Encrypt has one, apparently Comodo of course has one now - they're being abused because as soon as you're able to somehow poison DNS, you are then able to prove ownership of the domain that you don't really own, which is the only thing that the automated DNS server generators require. Which unfortunately then allows a full TLS interception that raises no alerts on the user's browser.

Anyway, we'll get to all that. Lots of other stuff to talk about for this Episode 711 this week. We have Google's use of their Sensorvault is the name they have for this thing, which is their vault of all tracking information for all of their properties of all of their users on Android and iOS. So it must be a big vault. And they use it to assist law enforcement. And that's a little creepy inasmuch as, you know, we're always being tracked all the time. We know that now. We know that disabling location tracking services for Google, like on Google Maps or whatever, doesn't actually do that, doesn't do what the user expects. You have to go to much greater lengths in order to actually turn that off. We've covered that in the past. But anyway, so we'll talk about that.

We've got an important update to Drupal again. And speaking of "again," Facebook. Unbelievable, Leo, what we learned about what they were actually doing with the fact that they were asking people for their email account logins. I mean, that was bad enough. It turns out what they were doing with them is just - it's just unbelievable. And I say in the show notes, and I can't wait to say it, Zuckerberg is saying they're going to rebrand Facebook as a privacy service. Well, okay. Just scrap the whole thing.

**Leo:** Yeah.

**Steve:** I mean, it's like...

**Leo:** It's based on the opposite.

**Steve:** Start over. Yes. Set up another facility somewhere. Tell people, okay, people, forget everything you ever knew about Facebook. We're now privacy. Start coding. Because, I mean, well, anyway. They discovered logs that said, oh, it looks like we - it's just unbelievable. Anyway, we also take a look at Russia's newly approved legislation moving forward towards their Internet off switch. I wanted to remind our listeners that USB Killers are a real thing. You can buy them on eBay and even from Amazon. And a student of a college in New York recently pleaded guilty when the prosecution got a video of him using them. We'll talk about that.

Also we have Marcus Hutchins, who has pleaded guilty to two out of 10 counts, and we have his posting and a follow-up tweet. We want to touch on that briefly. An actively exploited Windows zero-day. A bunch of Microsoft Edge browser news. The appearance of the Windows 7 end-of-life notices. Something from the "I did say this was bound to happen" department. We have some miscellaneous news, and then we're going to get to examining this Cisco Talos Group research that they titled "DNSpionage." So lots of stuff for today.

**Leo:** Love that name.

**Steve:** And, oh, boy, a real one-of-a-kind Picture of the Week for the podcast.

**Leo:** But fitting along with many previous Pictures of the Week. It's a good one.

**Steve:** Ah, yes.

**Leo:** You want me to show the Picture of the Week now?

**Steve:** Sure.

**Leo:** All right.

**Steve:** But everybody cover your eyes because we don't want you to see the new passwords that these people...

**Leo:** Yes. Secrets are revealed in here.

**Steve:** ...have chosen for themselves. Now, this is probably - this actually ups the ante, I think, on the My Internet Passwords Book which you can purchase and leave lying around your desk.

**Leo:** But at least that only you have access to, in theory.

**Steve:** No, well, maybe.

**Leo:** If you're at home.

**Steve:** So here, posted on the wall apparently, with the large headline "Password Change Sign Up Sheet," and it explains, if that wasn't clear: "If you'd like to change your password, please fill out the form below, and we will change your password on the system you indicate." So we have the first column is full name; and Kyle Smith, Liz Jones, Jack H., Big Ed, and Sam Adams, I think it looks like, have so far requested that they have some of their passwords changed. And then the second column is which system - Yardi, whatever that is, email, et cetera. And we see email, phone, Facebook, and Pike Pass, whatever that is. And then the third column is their current password. The fourth column is their new password.

So, for example, Kyle, who was the first to step up and get crazy with changing his password here for his email, explained that his current password is apparently Scooter49 $, and his new password is Skeeter4UZ, and then he says in parentheses "(all uppercase)," even though that's not how he filled out the form. Liz would like her phone password changed from 89621 to 4281. And Jack is really - Jack H., he's walking on the wild side. He's asked to have his email password changed from, yes, Password, to, yes, Password2. We can't really see what Big Ed is asking his Facebook to be changed from, it's like redst@pler, to mmmkay or something.

**Leo:** Mmmkay.

**Steve:** These are not really clear. Mmmkay, right.

**Leo:** It's from "South Park."

**Steve:** And Sam Adams and so forth. Anyway, everybody gets the idea. So I don't know what company these people work for, but you should just run away very quickly, as quickly as you can. It's sort of, I mean, maybe this whole thing is a joke. There is a yellow post-it note sort of stuck on the bottom saying "Come see me - Shawn."

**Leo:** This to me is the funniest part of all. If you put your name on this list, maybe you ought to come see me.

**Steve:** Yeah, anyway. So anyway, just another, you know, security is hard, and we're working every day to make it easier for you. So if you'd like your password to be changed, and you don't seem to be able to do it yourself, that's okay. We'll take care of that. Boy.

Google is officially using this tracking technology to help catch bad guys. We know that Google tracks us everywhere, even when we have Google Location History - that's the actual name - turned off. Last August on this podcast we talked about the fact that many of Google's apps, when running on either Android or iOS, continually monitor their users' location. Apps such as Maps or the Weather Update service on Android continuously monitor the precise latitude and longitude of the device they're installed on. But back then we talked about how the movements of a Princeton professor were continuously tracked, even while Location History was disabled. And our listeners may remember that in the research that was shown to demonstrate this, they deliberately removed from that

where a lot of the points where because it was this professor's home. And for the sake of his own privacy it was removed. But before that was done, it was there.

In response to the Associated Press investigation of this back then, Google responded: "There are a number of different ways that Google may use location to improve people's experience, including Location History, Web, and App Activity, and through device-level Location Services." They said: "We provide clear descriptions of these tools, and robust controls so people can turn them on or off and delete their histories at any time."

But at the time, Jonathan Mayer at Princeton said that it was his feeling that, if there was something called Location History that was there and obvious, that turning it off ought to be all that was required, not that users be expected to dig into the settings of every different Google application which is doing this in order to figure out how to turn it off on that app. And so it's actually quite involved to do so, as we talked about at the time.

Google explains that it uses location tracking features to improve its users' experience, like personalized maps, recommendations based on places you visited, help finding your phone, real-time traffic updates about your commute, and delivering more useful ads, so like ads relevant to where you are. In fact, since then we've talked about how it's a little unnerving, if you're in the emergency room in your local treatment center, that you're getting ads from so-called "ambulance chasers," suggesting that you call them if you need some representation about the accident you've just suffered. So, I mean, this happens to people.

So an interesting feature of this which has recently been receiving more attention, I ran across a number of references to it when looking for things to update our listeners about in the past week, is that Google may also share its users' location data with federal authorities who are conducting criminal investigations when Google is asked to do so with a warrant, so with a search warrant. And for what it's worth, the system works the way we would have designed it - you and I, Leo - if someone had asked us. So law enforcement first needs to obtain a so-called "geofence" warrant. And the news for me is that there is such a thing. You can get a geofencing warrant now.

**Leo:** I know, I was very disappointed to hear that this was even legal.

**Steve:** I know. Authorities then reach out to Google armed with that warrant. Oh, and I'm sure you know, Leo, also that it has mistakenly jailed people. So it's not like it's foolproof. Again, it comes down to people. Authorities reach out to Google with this geofencing warrant for the purpose of learning about every smartphone that was in this geofenced radius of a crime, around or proximate to the crime.

After receiving the warrant, Google queries this massive sensor vault database to gather its first pass data, which is all possible phones located within that region in some time window, and forwards that to investigators. So basically the big, the wide net. And for this first pass, each device is anonymously identified with an ID, nothing about the device itself. Investigators review the data, look for patterns of the devices near the crime scene, and then make a second request for additional data about specific devices by their anonymous ID that appear to be relevant, whatever that means. Again, this is all kind of gray area and soft.

So what Google then replies to this more limited set of specific devices with device movement out of the geofence, that is, within the region broadly, which presumably then allows investigators to further determine and narrow their search down to a few devices, which they then claim to have strong reason to believe may be useful for providing information crucial to the case as either suspects or witnesses to what went down. At

which point Google then reveals the owner's name, email address, and other data associated with the devices. And as I mentioned, the system is not perfect. It has resulted in false arrests.

But anyway, I wanted to bring this to our users' attention. And, Leo, what thoughts do you have? You apparently are chagrined, as I am, about it.

**Leo:** I don't really blame Google. I mean, if they're getting lawful warrants, they have to provide that information.

**Steve:** Right, right.

**Leo:** I wish they wouldn't make it too easy. But I don't feel like it's - I worry that it's not constitutional, that it's what we used to call a "fishing expedition." If you say, well, we don't know who committed the crime, what we'd like to do is just know everybody who was in the area at that time...

**Steve:** Yeah, good point.

**Leo:** That's the very opposite of a specific warrant, which in the past I thought had always been required. I'm no lawyer, so I'd love to hear the legal opinion on this. But in the past I always thought you had to have a specific, not only a specific person in mind, but a specific thing you're looking for on that person, instead of just saying, well, we just want to know everything that happened around there.

Now, some people have likened it to video cameras. So if you have a video camera in a 7-Eleven, of course the police could look at the video camera and see everybody who was in the 7-Eleven prior to the robbery or whatever. So some say it's kind of like that. But I think it's - I honestly feel like your location should be more private than that.

**Steve:** Yeah. And you know, I think maybe the line it crosses, and the video camera is a perfect example, is the concept of an expectation of privacy. We know that there is a formal concept of an expectation of privacy. And maybe it's time for us to lose that if we have a smartphone in our pants. But wow, I mean, that...

**Leo:** And by virtue of having a smartphone doesn't mean you're in public.

**Steve:** Right.

**Leo:** Right? You could not be in public. You could be in the house across the street. But the smartphone still pinpoints you.

**Steve:** Yes. And certainly you're able to look up and wave to the camera at the 7-Eleven store. I mean, so it's there. It's typically exposed. I mean, and it also serves as a deterrent. The 7-Eleven owners certainly know that you seeing this camera, I mean, the reason we know these things serve as deterrents is many of them are fakes. All they

have is an LED lighting up a little red light, and people are like, ooh, I'd better not do anything bad here. So they are proven deterrents. So that suggests that the presence of cameras removes any expectation of privacy.

And so, again, I also am not an attorney. But you just don't think that having a phone in your pocket automatically means that law enforcement could retrospectively, at any time in the future, determine that you were walking through a certain area or driving past something or who knows what. I mean, so it has been successfully used to solve crimes where there were no witnesses. As I mentioned, there was an innocent guy who was jailed for a week because he was the only person, based on this data, that law enforcement believed could have done it, even though he was innocent. And it wasn't until they actually found the bad guy a week later that they said, oh, our bad. Sorry about that. And again, those things happen even without this technology.

But anyway, I wanted our listeners - I thought this was interesting and important for our podcast listeners to know is going on, that there is something called, formally...

**Leo:** Sensorvault.

**Steve:** ...Sensorvault.

**Leo:** Geez, Louise.

**Steve:** Formally called Sensorvault at Google. And under subpoena, or under search warrant, rather, they will provide successive iterative levels of detail in order to aid investigations. So, I mean, you know, really it's inconvenient to turn your phone off, but that's what people have to do now if you want to move around without being tracked.

**Leo:** Yeah. And, you know, if you're smart, and you're going to commit a crime, and you turn off your phone, then this technique isn't going to work at all. It's only going to bring in innocent people.

**Steve:** Right. Or witnesses. I was thinking the same thing about...

**Leo:** No, maybe witnesses, yeah.

**Steve:** ...about turning the phone off. And so it would allow the police to go find - but think about that. Police knock on your door and say, "Hi there. On such and such and such and such, you were in the area of. And we don't think you committed the crime, but we want to know what you saw." It would be a little unnerving to be preemptively asked by law enforcement. And, you know, bad things happen. So anyway, it's a little worrisome.

Speaking of a little worrisome, this is only moderately critical. It is time to update Drupal again. Later, toward the end of the podcast, there's a reference to Drupalgeddon, which was the serious vulnerability which the Drupal content management system suffered that caused lots of upset. A year ago we had, on April 18th, a moderately critical cross-site scripting problem. That was on April 18th. On April 25th a year ago was a highly critical remote code execution flaw which affected a third-party - that was caused by third-party

libraries that Drupal was hosting. A year ago on August 1st another problem with third-party libraries. On October 17th, multiple vulnerabilities.

This year on January 16th a critical problem that affected Drupal due to third-party libraries. Also on the 16th, arbitrary PHP code execution due to a problem with PHP in this case, which Drupal was using. That was a critical vulnerability. A month later, on February 20th, a highly critical - that's their term - remote code execution that led to arbitrary PHP code execution. A month later, on March 20th, a moderately critical cross-site scripting vulnerability. A little bit less than a month later, April of this year, a moderately critical cross-site scripting problem that was in jQuery v3.4.0. There were also multiple vulnerabilities from a PHP templating engine on the 17th of this month.

And so that's 10 significant vulnerabilities in 12 months. We can no longer really count this as extreme by today's measures. After all, Microsoft patched, what was it, 74 vulnerabilities in one month? Two of them were zero-days. And in each of the past three months they've patched two zero-days. But on the other hand, those vulnerabilities do cover quite a lot of code real estate for Microsoft. And in Drupal's defense, a lot of this is coming from third-party libraries that they are importing and using in their system.

But it does make it very clear that, today, creating secure systems, offering secure systems means there really needs to be a way for keeping code up to date. And it's becoming crucial for widely deployed software. We've been talking about WinRAR every podcast for the last month. And the reason it's become such a huge target is that there is no mechanism for notifying the half a billion people who have downloaded it. Or was it half a million? Half a million, sorry, 500,000.

**Leo:** No, it was half a billion.

**Steve:** Was it half a billion?

**Leo:** Yes.

**Steve:** Oh, 500 million. Right, right, right.

**Leo:** Remember there's a lot of Windows users out there.

**Steve:** Yeah, and a lot of time has gone by while people have been doing this. Anyway, so the fact that it's going to stay available for some period of time makes it a big target. So, for example, I have prototyped, and I'll be talking about it a little bit, my own real-time updating system. I first implemented it in the DNSBench benchmark, that allowed it to check for whether new versions were available. And of course in the SQRL client, since it's very likely that someone's going to find something I've messed up, I've had an auto-update facility built in from the beginning which actually is causing us a little trouble at the moment, but we'll get past that.

So it really feels to me as though what we are coming to is a clear need for anything which certainly is an attack surface to have some means of notifying users, at least notifying its users if there's something they need to do. Unfortunately, over a long span of time, email is probably inadequate. There are certainly people who downloaded WinRAR 10 years ago. You can use it for free. Licensing it is optional. So I was glad to get email. I did license it from them and own it because that's a tool that I enjoy using.

But not everybody does that. I mean, the reason there's half a billion of them is that it's free.

People sometimes say, "How many copies of the DNS Benchmark have been downloaded?" And I say, "I don't know, like Never10, four million or something." The point is that these things are downloaded at that volume because they're freeware. They're not nagware. They're not shareware. They're just free. It's here you go. And if I were to attempt to charge something for it, it would be an annoyance more than anything else, for everybody. So it just makes sense for them to be free. But the problem is, in the case of WinRAR it has, as we know, completely out of the blue, created a serious vulnerability which is hurting people. So it just has to be, moving forward. And we've talked about this even in IoT devices, which have to provide some means, somehow, of keeping themselves up to date. And certainly routers. We've also been talking about that. Routers have to have a means now for fixing themselves.

And Leo, I can't believe, I mean, two weeks ago, when you were on the beach somewhere, Jason and I talked about Facebook and the unbelievable fact that for apparently - I've done a little more digging - since May of 2016 they were in some instances asking people for their email password. It's like, I couldn't believe it when this news surfaced a couple weeks ago. Their password. So as we know, if a reputable, security-oriented, privacy-minded entity wants to confirm that you are in control of an email address, they send a message to that email account with a link.

**Leo:** Everybody does this. This is normal.

**Steve:** How hard is that? Yeah, like what is the - well, Leo. There's another shoe that has dropped. Believe it or not, they were actually - Facebook, Facebook was logging into those email accounts, not only to confirm that they could, but to download and store all of that user's contact information, without their permission.

Business Insider last Thursday ran the headline "Facebook says it 'unintentionally uploaded' 1.5 million people's email contacts without their consent." In exclusive reporting, Bob Price wrote: "Since May 2016, the social networking company has collected the contact lists of 1.5 million of its users new to the social network. The Silicon Valley company said the contact data was 'unintentionally uploaded...'"

**Leo:** Hmm, yeah. Oh, it was a slip of the finger.

**Steve:** Yeah.

**Leo:** How do you unintentionally collect somebody's contact list?

**Steve:** Exactly. That's code. You have to have code.

**Leo:** Completely - yes.

**Steve:** "Unintentionally uploaded to Facebook" - and get this - "and is now deleting them."

**Leo:** Oh, yeah, I bet.

**Steve:** Uh-huh. Well, and of course that doesn't matter, Leo, because after you've ingested it...

**Leo:** Right, it's in the social graph. They've got your graph now.

**Steve:** Exactly. So of course this revelation comes after the pseudonymous security researcher e-sushi, that was his name, "noticed that Facebook was asking some users to enter their email passwords" - this is what we talked about two weeks ago - "when they signed up for new accounts to verify their identities." And of course, anyway, Business Insider then discovered that if you entered your email password, a message briefly popped up saying it was "importing your contacts" without asking for permission. And I actually have a screenshot from the Business Insider coverage of this in the show notes.

"At the time it wasn't clear," wrote Business Insider, "but Wednesday [last Wednesday] Facebook disclosed to Business Insider that 1.5 million people's contacts were collected this way and fed into Facebook's systems, where they were used to 'improve' Facebook's ad targeting..."

**Leo:** Of course it was.

**Steve:** "...build Facebook's web of social connections, and recommend friends to be added. A Facebook spokesperson said before May 2016 it offered" - now, get this, Leo. "A Facebook spokesperson said before May 2016 it offered an option to verify a user's account using their email password and then voluntarily upload their contacts at the same time. However, they said, the company changed the feature, and the text informing users that their contacts would be uploaded was deleted, but the underlying functionality was not." Which is Facebook speak for, "Some users were saying no. So we decided..."

**Leo:** We just did it without asking.

**Steve:** "...they meant to say yes." Because they were confused about how beneficial this would be.

**Leo:** Did you have to give them your email password? Could you have said no?

**Steve:** Yes, you could.

**Leo:** But a lot of people just say, oh, yeah, fine, okay.

**Steve:** Yeah. It wasn't at all clear that this was a voluntary thing. So there were other means you could use. But of course Facebook wanted their email password, and they wanted to flesh out their social graph, so they didn't - it's like Microsoft. Would you like to upgrade to Windows 10 now or tonight? It's like, I don't want Windows 10. But we

famously remember the choice that was finally given. Now or later? It's like no, neither. Anyway, wow. So 1.5 million people's contact books were directly harvested by Facebook. And of course now it's been ingested, so it doesn't matter if they're deleting it.

A Facebook spokesperson said in a statement to Business Insider: "Last month we stopped offering email" - after they were caught - "stopped offering email password verification as an option for people verifying their account when signing up for Facebook for the first time." Okay. Last month; right? Since May of 2016? "When we looked into the steps people were going through to verify their accounts, we found that in some cases people's email contacts were also unintentionally uploaded to Facebook when they created their account." That's what Facebook is saying now.

**Leo:** Unintentionally.

**Steve:** Yeah. Facebook is saying that we weren't aware this was happening, so we stopped offering that, and oh...

**Leo:** It's still happening.

**Steve:** Oh, my goodness. The message went away, but the conduct...

**Leo:** It's just it's so blatant. It's so awful. It's so terrible. I recently created a new private account on Instagram, and they just beat the drum. Give us your contacts, give us your contacts. And it's just very clear how valuable that information is. And don't forget, when you give somebody your contacts, you're giving your friends' personal contact information to a third party. Not just yours, but your friends'.

**Steve:** Mom, bless her soul, when she was alive, for like my birthday or Christmas or whatever, or Valentine's Day, would send me those greeting cards. And I kept telling her, "Mom, you're putting my email address, which I consider private information, into a website that sends a free greeting card. Mom, I love you, but I don't want" - because now, I said, now that is valuable information to that website. I'm getting spam from that website saying, oh, don't you want to send a greeting card to someone? No. I don't.

**Leo:** Well, and of course it's gotten far worse because it isn't just, oh, we can sell a mailing list now, and spam, because actually Facebook doesn't want to do that. They want to own your addresses and all your friends' addresses. They don't want to give it to a third party. But they use it in ways far, far worse. They know everything. It's terrible.

**Steve:** So there's another piece of this also. Facebook has said it didn't store the passwords.

**Leo:** Yeah, right.

**Steve:** Okay. Not that it needed to.

**Leo:** No.

**Steve:** It used them, sucked everything dry, and then said, okay.

**Leo:** We're done.

**Steve:** We've got all we need. Okay. So but in yet another Facebook privacy blunder which came to light recently, like last month, the company confirmed that it improperly stored hundreds of millions of user passwords in plaintext rather than as hashes. At the time Facebook said that this plaintext password storage error affected hundreds of millions of Facebook Lite users, tens of millions of other Facebook users, and tens of thousands of Instagram users.

That Facebook disclosure was just updated last Thursday to say that the number of affected Instagram accounts was much higher. Thursday's update said, and get this language: "Since this post was published, we discovered" - now, that's actually the word they used - "we discovered additional logs of Instagram passwords being stored in a readable format. We now estimate that this issue impacted millions of Instagram users. We will be notifying these users as we did the others. Our investigation has determined that these stored passwords were not internally abused or improperly accessed."

Okay. Now, how could they possibly make such an assertion after having "discovered additional logs of Instagram passwords being stored in readable format"? It's very clearly a total and utter unorganized disaster over there. As I said, if Zuckerberg wants to try to rebrand this as a privacy service, he just needs to scrap it. Just really...

**Leo:** No, he needs, I mean, truthfully not trust anybody over there anymore.

**Steve:** You're discovering logs of plaintext passwords? So, like, is there no management of any privacy information? It's just amazing. And again, if you really wanted to give them the benefit of the doubt, you would say that it's a bunch of Young Turks with freshly minted computer science degrees and an environment of let's try stuff and see what sticks. Wow.

**Leo:** You know, at this point I can't even be outraged anymore. It's just...

**Steve:** No, no, no.

**Leo:** They're obviously malefactors. They're not - you can't excuse this any longer.

**Steve:** It's too bad. So Russia has moved closer to adopting, and I can't wait to watch this happen, the Internet Master Cutoff Switch. And of course, Leo, my favorite organization over there in Russia is behind it.

**Leo:** Oh, yeah. Oh, of course they are, Rossonomabravo.

**Steve:** Rossmonkozdor, whatever the hell it is. Russia's lower chamber of parliament has backed a bill which privacy advocates fear could lead to the creation of a censorship system similar to - oh, you think? - China's Great Firewall. The Associated Press reported Thursday that the State Duma, which is the lower house of the Federal Assembly of Russia, has advocated for the bill overwhelmingly. The new regulations - if also accepted by the upper chamber, which belongs to the Federal Assembly, and then signed into law by President Vladimir Putin, and who imagines he won't just, I mean, he's just like waiting - would require Internet Service Providers to route Russian Internet traffic locally through the country. In other words, to have the capability of doing that.

This would give Russian authorities the opportunity to use equipment and software to establish man-in-the-middle communication eavesdropping, as well as to block and censor global content that Russia does not want its citizens to have access to. Although the Russian government has said that it will bear the cost and will reimburse ISPs, the country's ISPs would be required to provide equipment to exchange points approved by Russia's telecoms watchdog, here it comes, Roskomnadzor; and a localized domain name system, like their own DNS, would be prepared to support the localization of content. And so I guess that creates Ruskynet or something.

But the point is they want the ability to raise the shields around Russia and still have a working countrywide network. At the moment, for example, if DNS queries are going outside, and you raise the shields, well, everything breaks. So they're basically going to need to create, I mean explicitly create a different kind of subnet which can function if it needs to take itself off of the Internet. Advocates of the bill claim that this whole concept would be a protective measure, only meant to protect the Internet within the country should a hostile entity cut off access. Okay. As well as a means to insulate Russian traffic from potential cyberattacks by foreign entities by removing traffic rerouting outside of the non-Russian systems.

However, of course, many others believe that this creates top-level control, which would give Russian lawmakers an overarching authority to control the web within the country, as well as to monitor its citizens' online habits. I mean, you can imagine that part of this would be everybody using a system would have a root cert in their root store for a central Russian authority which would then explicitly allow all man-in-the-middle filtering to be conducted. So, you know, of course, I mean, that's what's going to happen.

So anyway, I did want to mention that last time we talked about this our listeners reminded us that the full, unfiltered Internet is also in orbit above us. So in fact we are not strictly limited to what landlines or short-range WiFi is able to carry. There are satellites up there, and the Internet is in the sky. So although it would certainly be an effective deterrent for most Internet connectivity, people who still really wanted to get unfiltered probably could.

So ZDNet carried the story of a 27-year-old Indian national who graduated two years ago with an MBA from the College of St. Rose in New York. He was just charged, I'm sorry, he just changed his plea from not guilty to guilty. And it occurred to me that the change in plea may have had something to do with the videos that he made of himself killing the college's computers, which the prosecutors got their hands on. The incident took place on February 14th, according to court documents obtained by ZDNet. He recorded himself, videoing himself while destroying some of the computers. On these videos he was seen to say: "I'm going to kill this guy." "It's dead." And "It's gone, boom." This guy destroyed 59 computers, but also seven computer monitors and computer-enhanced podiums that had open USB slots.

ZDNet wrote, they said: "He did it using USB Killer, a weaponized thumb drive that he purchased from a well-known online store that sells these types of devices." And my guess is that was probably eBay, although Amazon has them.

**Leo:** Hak5, too. He sells a lot of hacking stuff.

**Steve:** Yeah. So as we know, we've talked about these in the past, but I just kind of wanted to - we haven't talked about them for a while - just to note that they are real. I have a picture of one at the top of the story in the show notes, just sort of showing what one looks like. But if you go to eBay and put "USB killer," China is happy to sell you one for about 40 bucks.

**Leo:** I wonder if that's what the Chinese spy had in her possession when she broke into Mar a Lago?

**Steve:** No, well, I mean, she might...

**Leo:** And the Secret Service agent stuck it into his computer to see what was on it. And was shocked, shocked I tell you, to see it launch some software.

**Steve:** Yes. What a surprise.

**Leo:** What a shock. We've never seen this happen before, they said. Literally. All right.

**Steve:** Okay, well, that was different because that was malware.

**Leo:** Well, who knows what it was? It was installing something; right? What does this do?

**Steve:** Well, this charges internal capacitors up to 200 volts.

**Leo:** Oh, it actually fries it.

**Steve:** And then blasts the USB port with 200 volts. And that goes right into the central chipset of the motherboard and kills the motherboard.

**Leo:** Oh, it's a computer killer.

**Steve:** It's a computer killer, yes.

**Leo:** And that little USB condom you sent me, would that save me? No.

**Steve:** I don't think so. It would blast right through that because the USB condom allows power to go through, but it doesn't allow the data to go through.

**Leo:** Just not data, right.

**Steve:** So what this thing does is it's a little inverter. You plug it in, and it goes, very much like in the old days, remember, when your flash had to recharge, it would go [mimics sound]. So that was the xenon strobe in the flash needed a high voltage in order to create a plasma in the gas. So it would take the battery and run a little inverter to charge up the flash's capacitors; and then, on cue, it would dump that high voltage into the xenon tube and create a flash. This is similar. It charges high-voltage capacitors in the thumb drive up to 200 volts and then lets it loose into the USB port, certainly killing the port, and almost always killing the entire computer because now most of the chipsets directly drive the USB ports themselves. So, I mean, it is...

**Leo:** This is malicious. This is not a hacking [crosstalk].

**Steve:** It's pure malice.

**Leo:** It's just being mean.

**Steve:** Pure malice.

**Leo:** I love it that they brand it "USB Killer."

**Steve:** Yeah. You can see it says HV+ and HV-, and it also shows DC-200V. So it is a little 200-volt power supply that discharges...

**Leo:** Geez Louise.

**Steve:** ...itself back into the port that gave it the five volts in the beginning. Only takes a few seconds, and it will kill anything it's plugged into. I looked around...

**Leo:** This is, by the way, version 2.0 Accept no substitutes.

**Steve:** Ah. Well, actually Hong Kong now has version 3.0.

**Leo:** You know, to be fair, you could just bring a hammer. Right?

**Steve:** Yeah.

**Leo:** I mean, if you want to destroy a computer. I guess the benefit of this is it's somewhat more surreptitious. You could just say, well, I don't know. This computer doesn't seem to be working anymore.

**Steve:** Well, and that's the problem. And that's why I want just to put it back on our listeners' radar again is that anybody can buy this. All it takes is somebody slipping one of these things into a USB port which is available and counting to - not even counting to 10, and that machine is dead.

**Leo:** Wow. Wow.

**Steve:** I mean, dead dead.

**Leo:** Time to glue up your ports.

**Steve:** Yeah. So he destroyed 59 computers.

**Leo:** What?

**Steve:** Seven computer monitors, because of course computer monitors also have USB hubs in them now.

**Leo:** Sure.

**Steve:** And computer-enhanced podiums that had open USB slots. So the guy's a creep. He's facing, let's see, in total the equipment damages were $51,109, along with $7,362 in employee time for investigating and replacing the destroyed hardware. He's facing 10 years, up to 10 years in prison...

**Leo:** Good.

**Steve:** ...and a fine of up to a quarter million dollars, followed by a term of post-imprisonment supervised release of up to three years. So there are some USB plugs, but none of them seem to actually lock in place. There isn't a universal thing that a USB plug could really, like, lock into. You'd think - there is a tongue sticking out. So you'd think someone could come up with something that would go in, but would refuse to come out. But I looked around and didn't see anything that was readily available because really, I mean, not that lots of us are worried about that, but boy.

**Leo:** Sort of jerky thing to do.

**Steve:** It really is.

**Leo:** Jerky. He even made YouTube videos of it. That's how he got caught.

**Steve:** Oh, boy. No kidding.

**Leo:** Yeah. Yeah.

**Steve:** Yeah, that would do it.

**Leo:** I think he wanted to get caught. He was a jerk.

**Steve:** Yeah. He got his MBA from the school that he then zapped.

**Leo:** Fried. Geez.

**Steve:** Yeah. So now going from bad guys to good guys, Marcus Hutchins, Leo.

**Leo:** I still feel kind of bad for him.

**Steve:** I do. I do. I mean, yes, when he was a kid, when he was a kiddy, he got his...

**Leo:** A teenager, yeah.

**Steve:** A teenager. He got his start hacking and doing some things that were arguably malicious. But he matured, and for a long time he was doing good. I mean, we knew of him because in May of 2017 he was the guy - so, what, just two years ago next month. He was the guy who stopped the WannaCry ransomware outbreak in its track. He reverse engineered WannaCry because he was now a security researcher wearing a bleached white hat, so no doubt about the fact that he had, like, seen the error of his ways and was doing good.

Our listeners will remember because we talked about it at the time, he noted that WannaCry, which was a worm that was devastating the Internet, 200,000 systems in a 150 countries were affected by this, and it had caused billions of dollars' worth of damages. It was just starting, because it was a worm, its growth was just starting to go exponential when he stopped it. And he did so by registering a domain name that he saw being referenced by the worm. And when he looked up the domain, it was unregistered. So he thought, huh. That's interesting. He registered it. And he gave it a home, and the worm stopped.

It turns out, as far as we know, we never knew who created WannaCry, but it looked like somebody built in a cutoff switch for it in the form of this domain name. Marcus, registering the domain name, stopped WannaCry. Unfortunately, he was nabbed by Las Vegas PD at McCarran Airport after the Black Hat and DEF CON conferences a couple summers ago and detained, trying to go back to his home in the U.K.

And he's in the news now because he just posted to his blog, MalwareTech.com, the public statement. He said: "Legal case update: As you may be aware, I've pleaded guilty to two charges relating to writing malware in the years prior to my career in security. I regret these actions and accept full responsibility for my mistakes. Having grown up, I've since been using the same skills that I misused several years ago for constructive purposes. I will continue to devote my time to keeping people safe from malware

attacks." And then on Sunday, on his Twitter page, he tweeted: "To be clear, this statement wasn't required by the plea deal. It was my decision to post it."

**Leo:** Good for him.

**Steve:** Yeah.

**Leo:** Do they say what his sentence will be?

**Steve:** Let's see.

**Leo:** Maybe they haven't sentenced yet. Maybe he's just...

**Steve:** Two of 10 counts in the Eastern District of Wisconsin on Friday, one charge for distributing Kronos, which is the banking malware, and the other charge for conspiracy. And my feeling is it appears the U.S. has decided to make an example out of Marcus. According to court documents, he now faces up to 10 years in prison and half a million dollars in fines. The good news is he also had some good legal representation who does think that he is being mistreated because of who he has been ever since. And really, it does really seem overboard. There are really, really bad people really, really doing bad things. And it'd be nice if they got a lot more attention than Marcus, who made some mistakes when he was younger.

One of the Windows zero-days which was patched two weeks ago has now, since then, come under heavy use in the wild, which is being used to facilitate full system takeover. And what's interesting to me is the change in usage of this. We're learning more about one of those two zero-day flaws Microsoft patched two weeks ago since it's now in active use in advanced persistent threat campaigns. As we noted last week, it was discovered by two researchers at Kaspersky's Lab on St. Patrick's Day of this year, when they found it being used against one of their customers who is under their protection. It's a use-after-free bug in the Windows kernel win32k.sys module.

The flaw allows a local privilege escalation, and it's being used in advanced persistent threat campaigns targeting 64-bit versions of Windows from 7 up through older builds of Windows 10. The attackers are using the bug to establish persistent backdoors in targeted machines, gaining the ability to run arbitrary code in kernel mode. An attacker could then install programs; view and change and delete data; create new accounts under full user rights. And Microsoft has admitted that that's what this allows.

What's most likely in this instance is not that the fix was reverse engineered, as is often done, to discover previously unknown bugs once they've been patched, but rather that those who were previously deploying this potent flaw in very limited and only in highly targeted attacks now know that its useful lifetime, now that it's been patched, is extremely limited. So prior restraint has been discarded, that is, restraint in the use of this. You know, it was valuable when it was a zero-day. Now not so much. So they are racing to exploit it into what is I'm sure a rapidly dwindling base of still vulnerable machines, any which have not yet been patched from two weeks ago on April's Patch Tuesday.

So that's yet another sort of interesting dynamic in the way the world is evolving. We're seeing flaws that come to light that are not known by anyone, suddenly being exploited

after their patches are reverse engineered. Here we're seeing one that was known to a few people, at least one, that had been exploited with a profile of very selective targeting, suddenly being released for much wider targeting, almost certainly not because it was reverse engineered, but because the people who were using it selectively thought, oh, crap, the jig's almost up with this. Let's get as much use out of it as we can before it won't work anymore. So, wow, another interesting dynamic.

Some interesting Edge news. And I was very impressed with Lawrence Abrams' coverage of this over at Bleeping Computer. I have the link to this coverage in the show notes. He walks us through a chain of events that could cause Edge to inadvertently be run with admin privileges. What he noted is that - so we know that we've got a forthcoming version of Edge which is available currently. Although it doesn't explicitly say that Windows 7 users can download it, it works under Windows 7. And it does explicitly say that it will eventually be formally released for 7, which is neat.

It's, as we know, based on the Chromium engine. And the first thing Microsoft did was strip out all of the Googlization that had been done to Chromium, and now they're Windowizing it, or Microsoftizing it by building their own stuff back in. One of the things they just added is a notice if Edge is launched with admin privileges. A popup comes up saying "Administrator Mode Detected." And then it says "Close Microsoft Edge and relaunch in non-administrative mode for best performance." Well, it's really not best performance. It's best security.

So Lawrence takes us through a scenario where to edit the hosts file it would be necessary to run Notepad as an admin. Then, if when editing the hosts file you encounter a suspicious entry, and I think in his example he said www.malware or malicious.com or something. It's like, yikes; you know? So if you then highlighted it and right-clicked on it, there is an opportunity, say "Search with Bing," which would launch Edge. And because the children of processes inherit the rights and the account of their parent, and since Notepad had been launched with admin rights in order to allow you to edit the hosts file, Edge would then be launched with admin rights. And if you didn't have this notice warning you, you might do some searching with Bing and then decide, okay, fine, well, whatever.

But if you left Edge running, it would keep running with admin rights. And if you were then to download something, or to go somewhere malicious, you're now browsing with your system's largest attack surface exposed to the Internet. And god help you if you were to download a program and run it because of course you are able to run things from within the browser. You would be running it with admin privileges. Which would remove all constraints over what it would be able to do.

So first piece of news is Edge is getting a notice like this. It is the case that someone could be running in admin mode without it occurring to them. And tip of the hat for BleepingComputer doing such a nice job of covering this little bit of news.

Also in Edge news, Microsoft's new Chromium-based Edge browser is in the process of gaining the ability to run within Microsoft's very powerful and useful Windows Defender Application Guard sandbox. It can actually run in there now; but a bunch of warning messages which are very useful to inform the user when something has been blocked, like you can't download files from within there and save them on your system because that's the whole point, those messages are not yet present.

So it's probably worth waiting for its official release. And of course I've often lamented the need for a really, really strong isolated sandbox, I mean, really like to the level of a VM, for our browsing. This integration of Chromium with Edge having Windows 10 latest security features, including allowing Edge to run under Microsoft's Application Guard, the Windows Defender Application Guard, I think this is a huge win. Apparently it takes a

while to get it up and launched, but I would argue if at any point you're doing something that you think might be a little sketchy or going somewhere sketchy, having this there is going to be very useful. So tip of the hat to Microsoft. I mean, our browser is our system's attack surface these days, that and email. And protecting it is a great thing.

And also I was a little disturbed to see that, while we're on the topic of Edge, it's interesting to note that, as the new Chromium Edge reacquires some of the unique capabilities of its Edge HTML-based predecessor - so to explain that, Microsoft basically went all the way, developed Edge HTML as their own rendering engine. Then they decided, okay, for whatever reason, we're no longer going to continue supporting it. We're going to switch to Chromium. It's what everyone apparently wants. We get to leverage the open source community and so forth. We'll wrap Edge around the Chromium guts.

Well, what that meant was initially all of the previous Edge HTML stuff disappeared, whatever it was that Microsoft liked. So they're now beginning to reincorporate that into the Chromium engine contained by Edge. So Edge will be dynamically changing its user agent string to show different faces to different sites. Which, you know, it's not quite a face plant for me; but it's like, this is not the way the Internet is supposed to work. But, for example, at the moment, when visiting Netflix, HBO Now, HBO Go, Napster - I didn't even know Napster was still a thing - or Sling, Edge will display its Edge persona. That is, to those domains, to those websites it says we're the old HTML Edge browser. But when visiting Facebook, Messenger, or there's an Australian media streamer stan.com.au, Edge masquerades as Chrome and shows its Chrome user agent.

So this is a horrific kludge. On the other hand, the user agent field has a long, horrifying history of not really being what it pretends to be. I remember we've talked about this years ago on the podcast, how annoying it was that looking at the IE user agent field years ago, and in fact any browser's user agent field, they all had Mozilla in them for some reason. It was like, what is Mozilla doing in there? It's got nothing to do with Mozilla. It was like, yes, but some sites want to have Mozilla in the user agent field. So we're giving it to them. It's like, what?

But, yeah. If anyone's interested, there's a wonderful blog posting I found, the "User Agent String History," which details accurately and from the viewpoint of someone who's been through all of this, the historical step-by-step from Netscape Navigator on of what the various browsers did, noting that at one point Opera gave you a dropdown choice, either radio buttons or a dropdown list box of which user agent you wanted to use for your Opera browser. It's like, oh, lord, really? But anyway.

I don't know where this is going to shake out, whether Microsoft always intends to have a per-site choice of browser impersonation in its user agent field. I hope not. I hope this is just some transient thing. But apparently there are things that Edge HTML offers that Netflix, HBO Now, HBO Go, Napster, or Sling take advantage of if it's present. And Microsoft has made it present, and so Microsoft wants it used. I don't know. Anyway, just an observation. And it is nice, though, that we're getting Edge moving along.

My best friend sent me a text with a photo of his screen, saying, "Whaaat?" And he got the "After 10 years, support for Windows 7 is nearing the end." We've talked about this forthcoming notice. It hasn't hit me yet. The good news is, in little itty-bitty fine print in the lower left, you're able to say "Don't tell me any more." So when everybody gets it, they can just say, okay, yeah, I know. And it is a little annoying that Microsoft is using this to sell PCs. They're literally, in the link you click for "Tell me more," they are saying, "You need a brand new computer in order to efficiently run Windows 10. So click here to shop for a new Windows 10 machine." And it's like, wow, okay.

And finally, well, not finally. Well, maybe it is finally, oh, it is finally in the news department. This is from the "I did say this was bound to happen" department. We just have in the news that Mozilla's Firefox browser will be enabling hyperlink auditing, a.k.a. URL ping tracking, by default in a forthcoming release of Firefox. Larry was on the ball again over at BleepingComputer, and they're following this. I don't know if he listens to the podcast, but of course URL ping tracking was our topic two weeks ago.

Quoting from his coverage, Mozilla feels, as I do, that it's a performance improvement. He said: "While some users feel this feature is a privacy risk, browser developers feel that trackers are going to track, so they might as well offer a solution that provides..."

**Leo:** Trackers gonna track.

**Steve:** Trackers gonna track, "...that provides better performance. In a post by Apple, the WebKit developers explain that hyperlink auditing pings are a performance improvement because, unlike other tracking methods, they do not block or delay the navigation to the requested site." That's of course my observation from two weeks ago.

Quoting Apple in BleepingComputer's coverage: "Just turning off the ping attribute or the Beacon API doesn't solve the privacy implications of link click analytics. Instead, it creates an incentive for websites to adopt tracking techniques that hurt the user experience." And we talked about that, too, the 302 redirect chain, which is the alternative. "In effect," writes Apple, "the choice between supporting ping and not is not one of privacy; rather it is a choice between a good user experience and a bad one."

Larry writes: "After reading Apple's post, I contacted Mozilla to see if they agreed with the views expressed in the WebKit article. Mozilla told BleepingComputer via an email that they agreed with Apple's views on hyperlink auditing. Furthermore, they stated that the only reason it's not currently enabled by default in Firefox is because their implementation is not ready." Okay, now, I find that curious because it can be turned on. If you go to about:options, as I talked about two weeks ago, it's there, and you can turn it on if you want.

Anyway, Mozilla wrote: "We agree that enabling the hyperlink ping attribute that is commonly used for hyperlink auditing isn't a question of privacy, but a matter of improving the user experience by giving websites a better way to implement hyperlink auditing without the performance downsides of the other existing methods listed in the WebKit.org blog post. In fact, we already support the sendBeacon API. And the reason we don't yet enable the hyperlink ping attribute is that our implementation of this feature isn't yet complete." So I guess they want to do a few more things.

And finally, writes Larry: "When we asked if they felt that users should at least be given the ability to disable the feature if they wish, Mozilla stated that they did not believe it would have any 'meaningful improvement' to a user's privacy." So they're not even going to let it - they're going to turn on by default, and they're going to take away the option to turn it off.

Mozilla says: "We don't believe that offering an option to disable this feature alone will have any meaningful improvement in the user's privacy since websites can, and already do, detect the various supported mechanisms for hyperlink auditing in each browser; and disabling the more user-friendly mechanisms will cause them to fall back to the less user-friendly ones without actually disabling the hyperlink auditing functionality itself." Meaning users are better off if it's on and stuck on.

Brave states it will continue to block this feature. Larry wrote: "After Mozilla's response, we also contacted Brave Software to ask if they had any plans to enable hyperlink auditing in their browser." Brave wrote: "Disabling hyperlink auditing is a crucial privacy feature, and Brave has always disabled this by default." That was written by Catherine Corre, Head of Communications at Brave Software. She finished, saying: "Brave users expect this protection from our browser." So there.

And of course I know that this is a controversial topic after what some of our users felt was my own capitulation on this issue. I received a bunch of angry and annoyed feedback through several channels. But it is really, truly, I would argue, almost impossible any longer to actually not be tracked on the Internet. It's like, you just, good luck with you. It's just - you just can't arrange not to be tracked.

And Leo, I think time for our last break, and then I'm going to share some interesting miscellany about my newly relaunched and relocated blog, my experience with a newly minted Authenticode certificate, and what Cisco's Talos Group have found.

**Leo:** I'm just curious about this URL hyperlink auditing.

**Steve:** Yeah.

**Leo:** So we don't allow tracking cookies of any kind. Our advertisers are always saying can you put one on, but we don't do it. We have some banner ads on our website. But we do allow UTM trackers, which basically is a referrer. So if you go to a sponsor page or a banner, and you click the link, and you go to the website, they're informed where you came from.

**Steve:** Right.

**Leo:** You came from TWiT. And the value of that for us is that they can track whether they're getting results from banners on our website. Is that what you're talking about?

**Steve:** No.

**Leo:** No. Okay.

**Steve:** So you know how a standard <a> tag has a url= and, you know, the link. And so when you click on it, it takes your browser there.

**Leo:** Right.

**Steve:** There's another verb that can go in an <a> tag, ping=.

**Leo:** Right.

**Steve:** And that could have a list of URLs. And so the reason the browser...

**Leo:** Oh, I see.

**Steve:** Yeah. So the reason the browser vendors prefer it is it creates a fork. When you click on that URL, your browser immediately goes to the URL in the URL field, but then asynchronously it sends POST queries to the list of URLs in the ping argument.

**Leo:** You can see how inefficient that would be if it were blocking because you'd have to wait till all of those POSTs happen before you got to the web page.

**Steve:** Correct. Well, in fact, if you don't have the ping, what you have to do is the URL cannot take the user to its destination. It has to follow a chain of those websites following a redirection chain until it gets to the last one, which then finally takes you to where you want to go.

**Leo:** Really slows things down, yeah. That's crazy, yeah.

**Steve:** Yes, yes.

**Leo:** Yeah. And the reason I'm asking is I said, yeah, it's okay for us to do UTMs after looking into it because I felt like, well, that's kind of an unobtrusive, fairly unobtrusive way of just sourcing where they got the hit from.

**Steve:** Yup.

**Leo:** And our advertisers do want to know that. So that's the only kind we allow.

**Steve:** Yup.

**Leo:** Yeah, okay. Thank you, Steve. All right, Steve.

**Steve:** So I originally had two blogs hosted at WordPress. I had a CNAME record in my DNS which mapped steve.grc.com to one of those, and blog.grc.com to another. That worked years ago. But today it doesn't because it's hostile to TLS since WordPress doesn't have certificates for those domain names. So it was referring - so someone doing https://blog.grc.com, when it was hosted at WordPress.com, would get error messages.

**Leo:** The wrong certificate; right?

**Steve:** Yeah, exactly, would get the wrong certificate. And it had been - it was something that sort of - it was just sort of chafing for a long, I mean, you know, I've got my own servers. I've got a facility. I've got bandwidth. I've got all the other plumbing

required to do this. What I did not have was a server running, like a mature, safe, sandboxed server running PHP. I do have that now because the forthcoming SQRL web forums are hosted on PHP. So now that I've got a mature PHP server facility set up at GRC, I decided that it was time to, first of all, consolidate those and set up my own WordPress blog there. Today - and it's in place. I announced it. I tweeted it. I posted to both of the old blogs for all the subscribers that I had amassed there that it was now just blog.grc.com. Anybody who's interested can go to blog.grc.com and see the inaugural posting there.

So I'm glad I did it. And I guess I would do it again. But frankly, simply setting up a blog at WordPress - and we should mention just for the record they're a sponsor of the show. While you've been talking about them during this podcast, I just haven't had much to say. But having installed it and then bolted it down, I really appreciate that bolting it down is crucial. And of course I knew how to do that. But most people just want to blog; you know? They're not hosts of a security podcast. And of course after I thought I had done everything I knew to do, I went looking around the 'Net for advice from those with more WordPress-specific experience than I have, and also those who have some scars from the arrows in their backs in the past.

**Leo:** Who doesn't?

**Steve:** Well, yeah. And, I mean, I found people who, like, whose self-hosted WordPress sites had been hacked, and who said, you know, ooh, here's what I learned from the experience. So I was pleased, first of all, that no one had anything to say that I hadn't already arranged at least as good and sometimes a superior solution for. For example, everyone mentions the need for a really strong password. Yeah. And then also some advice to adding an add-on to have a password lockout. Well, I of course have a 32-character total gibberish password that I've never attempted to even look at. But my WordPress login page cannot even be reached by anyone who is not at one of a very few well-known IP addresses. So I've gone even further. You can't even access that page from the Internet. So yes, I did belts and suspenders. So I seem to be pretty well protected.

But in my roaming I encountered a site that made me think of the things you have talked about, Leo, about WordPress. It was titled "The Top 10 Security Mistakes That Self-Hosted WordPress Blogs Make."

**Leo:** I bet I've made every one of them.

**Steve:** And it's not long. I just want to - I've excerpted from this. The blog post said: "According to Forbes, one out of every six websites on the Internet is powered by WordPress, nearly 60 million in all, with 100,000 more popping up each day." A hundred thousand a day. "WordPress.com currently hosts over 56 million blogs. As of this writing, WordPress stats did not include the number of self-hosted blogs, but," this person writes, "rest assured there are many of us. I've been using WordPress since Gold days." Don't know what that means. Was there something called WordPress Gold? Anyway...

**Leo:** I don't remember, no.

**Steve:** "And it only gets better with each release. In the past I have been the victim of two WordPress hacks. At the time of the first hack I was on a managed VPS. All maintenance and administrative tasks, including software updates, was administered by the hosting provider. In my case, the software was rarely updated.

"Running a self-hosted blog comes with myriad responsibilities. It is not like you can merely install it and be done with it. Your first priority should be to familiarize yourself with the platform, along with the pros and cons of self-hosting or hosting your blog at WordPress.com. If you self-host, you will need to be somewhat tech savvy. If not, hire someone who is. When you self-host you are responsible for technical maintenance: backend configuration, backups, blog security, logs, spam filtering, and updates. Take the time to find a reputable and reliable hosting service. Do your research first. You don't want to end up on a server that is easily compromised, is slow to update software, has bad tech support, or has too much downtime.

"The fact that hackers and cybercriminals favor targeting WordPress is for the same reason they favor exploiting Microsoft Windows. It's popular. I have seen a lot of site admins downplay the importance of updating CMS software and hardening company blogs. This is especially prevalent with small businesses and startups that rely solely on development teams to schedule site updates and releases. I have also seen many home businesses slap together self-hosted blogs because they noticed that cPanel had a Fantastico, Softaculous, or Installatron auto-installer." Yeah, right, like the one click, get a blog.

**Leo:** Yeah, I've used that, yeah. Mm-hmm.

**Steve:** Uh-huh. "And they think that all they have to do is populate their blog with posts, widgets and plug-ins."

**Leo:** So easy. Yup.

**Steve:** He says: "For the love of Matt Mullenweg, please check out WordPress.com."

**Leo:** Yes. Let them do it.

**Steve:** So anyway, yes, that's a plug. And it's a sincere plug. I mean, if you are a propeller head, even Drobo will run a WordPress blog. But you really, really, really need to be thoughtful about it. And there is something...

**Leo:** You have a responsibility all of a sudden.

**Steve:** There is some jot thing that WordPress has, which I'm signed up for.

**Leo:** Jet.

**Steve:** Jet, yeah. And so it is a series of...

**Leo:** Love Jets. Love Jets.

**Steve:** Yeah. So it's a series of services that's not free, but it's inexpensive. It's like three bucks a month or something. And so you get Akismet to do your spam filtering. You get something else that does daily backups. You get something else - and then WordPress is also constantly checking for any updates and has the ability to proactively update your code if something bad happens. So, yes, I'm self-hosted. But even so, I'm also subscribed to this little, you know, have them looking over my shoulder because they own Jet, by the way. It's Jetpack.

**Leo:** Yeah, yeah, yeah, it's so good.

**Steve:** That's it, Jetpack.

**Leo:** Yup. I highly recommend it.

**Steve:** So next little piece of miscellany is I've had an interesting experience. We are just at this very moment going through some upheaval over the expiration after three years of my then, or up to now, longstanding Authenticode code signing certificate. As this was approaching - and in fact, Leo, during that last sponsor interval I went to the Win10 system that I'm talking to you through Skype on because the Edge SmartScreen was blocking the SQRL client because it is signed with a new certificate.

**Leo:** Oh, yeah.

**Steve:** That has not yet acquired a reputation. So what happened is, after three years of having a spotless reputation with the Authenticode code signing certificate that I had from DigiCert, those three years were coming to an end. As this was approaching, I noticed that DigiCert was offering EV code signing, which I'd never really paid any attention to, even though Microsoft introduced it seven years ago in 2012. So it turns out that to perform EV code signing, a developer must have a physical encryption dongle. And it came in a cute DigiCert box.

**Leo:** Oh, that's cool.

**Steve:** Yeah, and you can see where there is that little divot in the box is now empty because my hardware dongle which has the private key bound into it and will never let it go and which performs the actual encryption for me, it's stuck into my computer so that I'm able to perform EV code signing.

Anyway, so it turns out that for EV code signing a developer must have a physical encryption dongle, and the secret key must be buried deep inside it. And this of course prevents that secret from being exfiltrated electronically over a network. So of course that's what I got from DigiCert. I just showed the box. And I'm really tickled to be able to EV sign my work with it for the next three years.

However, the trouble with any new certificate is that it will initially not have accumulated any reputation. And I mean literally. I mean reputation, even though it's Gibson Research

Corporation, and I have a reputation, the cert doesn't. And remember that, technically, the certificate itself is asserting who signed it and that what was signed has not been modified. But that's all it's asserting. So nothing would prevent a malicious actor from obtaining a certificate under the name Gibson Research Corporation and signing malware with that certificate and putting it out onto the Internet. I mean, I hope it doesn't happen. It never has.

And the idea is that, in the same way that certificate authorities are supposed to make sure that a domain name really owns that domain, nobody should be able to get Gibson Research Corporation other than me, who had to prove that I am Gibson Research Corporation. But from a practical standpoint, nothing prevents it. Which says that it's the reputation of the certificate because what nobody else can duplicate is my certificate. They could get a fraudulent one under the same name, but it would have a different hash. It would be a different certificate, not mine.

So what I'm now in the process of needing to do is to get this new certificate recognized as trustworthy. And first of all, this happened over this past weekend, and all the AV systems had a meltdown. Which is unfortunate because there's nothing malicious about what I'm doing. But this demonstrates to what degree the antivirus systems have become heuristic. They look inside. VirusTotal even runs the downloadable in a sandbox and watches what it does, watches its communications.

One of the things that the SQRL client does when you first run it is, before it even installs, it sends out a ping to check for an update to prevent the user even from installing an older one, just because I'm a belt and suspenders person. It turns out that that behavior of sending out that ping was upsetting some of the AV scanners. So I slightly modified it so that it wouldn't do that as often and as reliably as it was before. Still now we're in this position of this brand new cert needing to prove itself.

So what I was doing was with SmartScreen, after Edge refused to download it, saying it was unknown, I right-clicked on the downloads and said this is not malicious. Then I had to fill out a little form about why I didn't think so. Oh, and there was a checkbox, you know, are you a user, or did you create it? And I said I created this. Believe me, it's not malicious. So anyway, the other thing I did was I re-signed four of the top five downloads from GRC: DNSBench, InSpectre, Never10, and LeakTest, believe it or not is still one of the top five. I re-signed them with this new EV cert because about 3,500 copies of those in aggregate are downloaded every day.

So that will, you know, I just need to get its use out there so that, when people start wanting to play with SQRL, they're not being alarmed by AV misfiring. And none of the previous SQRL releases, you know, we've been working with SQRL for years, and I've never had this trouble. But what's interesting is that three years ago, when the certificate that has just expired was new, that's when I was doing Never10, and the same thing happened then. It was a new certificate. It had not earned a reputation yet. And so during the beta testing of Never10, people were saying, hey, I'm getting AV warnings, and SmartScreen is saying we don't know you and so forth. It went away after a few days, after enough people said, yeah, no, I'm sure this is fine. This is good. This is a good guy.

So anyway, I just sort of wanted to share a little bit of experience from the field. And one thing that I'm wondering is whether I had cross-signed or co-signed the existing software with both the old and the new cert before the old one expired, if that would have had some effect. That is, could I - because it is possible to co-sign code. So if I saw that the end was coming, could I get an update from DigiCert, then start co-signing the code. It would be trusted because it contained the cert which had established three years of good reputation and also trusted because, well, and for that reason, but then also convey its trust to this new cert. I don't know if that would work, but I'm going to try it in three

years because this is a pain in the butt. You know, our AV stuff has just gone so overboard, I mean, it's become so heuristic. There's zero actual reason to believe any of my apps are malicious. None of them are.

But all the AV, unless you have a reputation, and that's where it really comes down to. It comes down to getting a certificate. In fact, when I was doing a little browsing over the weekend I saw some posts from a guy who's, like, he said he has some accounting software. He updates it every month. It's constantly having this problem of scaring its users because Microsoft's SmartScreen says I don't know what this is. This is not often downloaded. You should not trust it. And he says it's freaking people out. And in this log of - it was a question he posted over on SourceForge or somewhere. And a lot of people said you need a cert. Get a cert. A cert is the way that you establish a reputation independent of the code which it signs. And that makes sense. But on the other hand, it takes some time to establish that reputation.

Oh, and one of the postings on my new blog was from someone who, on April 19th he sent: "Dear Steve. As what might be called an 'historic' user of SpinRite, I have two questions for you." He says: "One, do you still make available a retail version of the product?" And I didn't know what that meant, but he said: "Or is SpinRite a 'download only' at this point?" And then he said also: "Two, searching high and low for my several versions of SpinRite, I have yet to find the original book/software/serial numbers. So if I could provide you privately my name, address lived at when purchased and registered, for the versions I own, would you have existing records to verify my status as an owner?"

He says: "Any response will be appreciated, as were a number of your utilities - LeakTest, DCOMbobulator, Never10." He says: "But above them all, I'd hardly be able to tell you how many times SpinRite saved flaky, error-ridden disks, be they floppies or hard drives."

**Leo:** Floppies? What are those?

**Steve:** Yeah. So I replied...

**Leo:** Does SpinRite work on floppies?

**Steve:** Oh, yeah.

**Leo:** I never even thought of that.

**Steve:** Yeah, it's super useful for floppies because they have no error correction built in, and so they really do need it. So, yeah. In fact, people often will have something on a floppy that they, like after a decade it's like, oh, my god, that's my only copy of this. And it's like, yeah, SpinRite will bring it back to life.

**Leo:** Cool.

**Steve:** So anyway, I replied to Madman. He posted as Madman in MN, so Minnesota. So I said to him, my next post here at https://blog.grc.com will be about my roadmap for

SpinRite, since people who don't listen to the Security Now! podcast will not have heard about my plans. And having them documented will be useful in any event. So, one, thank god we no longer have any physical shipment of SpinRite, only the download. That makes the lives of my little three-employee company, counting myself, so much more sane.

And to his point two I said we still maintain a database of every copy of SpinRite ever purchased going back 30-plus years now. We have pre- and post-online databases, and the pre-online database is written in FoxPro, a dBase II clone. And literally she has FoxPro running in a DOS box. I said, if you will write to Sue at our sales email, which is always sales and then the current year and then @grc.com, she'll be glad to look you up and verify your status.

And I said, since SpinRite v6 has been in use for the past 15 years - v6 - we are giving serious consideration to terminating upgrades from earlier versions once v6.1 is formally released, under the thinking that, since v6.1 is going to be so much faster and more capable than v6.0, and we're going to be giving it away to all v6.0 owners going back 15 years, that should be a sufficient commitment to our previous customers. And that anyone who's still interested in SpinRite at all will have upgraded to v6 sometime in the past 15 years, so they'll be covered.

And I finished, saying: "I thank you very much for your interest and support. All of those other things you mentioned that I have done, I've been able to give away because people purchased SpinRite."

**Leo:** Bravo.

**Steve:** So DNSpionage. I can't do a better job of summarizing this than Cisco did, so I'm just going to share the beginning of a very long post. There's no need to get into the weeds.

So their title was "DNS Hijacking Abuses Trust in Core Internet Service." They wrote: "This blog post discusses the technical details of a state-sponsored attack manipulating DNS systems. While this incident is limited to targeting primarily national security organizations in the Middle East and North Africa, and we do not want to overstate the consequences of this specific campaign, we are concerned that the success of this operation will lead to actors more broadly attacking the global DNS system."

They write: "DNS is a foundational technology supporting the Internet. Manipulating that system has the potential to undermine the trust users have on the Internet. That trust and the stability of the DNS system as a whole drives the global economy. Responsible nations should avoid targeting this system, work together" - that is, you know, that is they believe this is a nation-state. So they're saying responsible nations should avoid targeting this system for attack, work together "to establish an accepted global norm that this system and the organizations that control it are off-limits, and cooperate in pursuing those actors who act irresponsibly by targeting this system."

Then they said, under executive summary: "Cisco Talos has discovered a new cyberthreat campaign that we are calling 'Sea Turtle,' which is targeting public and private entities, including national security organizations, located primarily in the Middle East and North Africa. The ongoing operation likely began as early as January 2017" - so more than two years on now - "and has continued through the first quarter of 2019. Our investigation revealed that at least 40 different organizations across 13 different countries were compromised during this campaign. We assess with high confidence that

this activity is being carried out by an advanced state-sponsored actor that seeks to obtain persistent access to sensitive networks and systems.

"The actors behind this campaign have focused on using DNS hijacking as a mechanism for achieving their ultimate objectives. DNS hijacking occurs when the actor can illicitly modify DNS name records to point users to actor-controlled servers. The Department of Homeland Security (DHS) issued an alert about this activity on January 24, 2019, warning that an attacker could redirect user traffic and obtain valid encryption certificates for an organization's domain names.

"In the Sea Turtle campaign, Talos was able to identify two distinct groups of victims. The first group we identify as primary victims includes national security organizations, ministries of foreign affairs, and prominent energy organizations. The threat actor targeted third-party entities that provide services to these primary entities to obtain access." In other words, the threat actors went after, for example, the DNS registrars for the companies they wanted to target.

"Targets that fall into the secondary victim category include numerous DNS registrars, telecom companies, and Internet Service Providers. One of the most notable aspects of this campaign was how they were able to perform DNS hijacking of their primary victims by first targeting these third-party entities." So it's an indirect attack in that sense.

"We assess with high confidence that these operations are distinctly different and independent from the operations performed by DNSpionage, which we reported on in November of 2018. The Sea Turtle campaign almost certainly poses a more severe threat than DNSpionage, given the actors' methodology in targeting various DNS registrars and registries. The level of access we presume necessary to engage in DNS hijacking successfully indicates an ongoing high degree of threat to organizations in the targeted regions. Due to the effectiveness of this approach, we encourage all organizations globally to ensure they have taken steps to minimize the possibility of malicious actors duplicating this attack methodology.

"The threat actors behind the Sea Turtle campaign show clear signs of being highly capable and brazen in their endeavors. The actors are responsible for the first publicly confirmed case against an organization that manages a root server zone, highlighting the attacker's sophistication. Notably, the threat actors have continued their attacks, despite public reports documenting various aspects of their activity, suggesting they are unusually brazen and may be difficult to deter going forward. In most cases, threat actors typically stop or slow down their activities once their campaigns are publicly revealed.

"This post provides the technical findings you would typically see in a Talos blog. We will also offer some commentary on the threat actors' tradecraft, including possible explanations about the actors' attack methodology and thought process. Finally, we'll share the indications of compromise (IOCs) that we have observed thus far, although we are confident there are more that we have not seen."

So that's sort of the sum of it. I then grabbed a few things out of the rest. Under assessing Sea Turtle DNS hijacking methodology, they said: "It's important to remember that the DNS hijacking is merely a means for the attackers to achieve their primary objective. Based on observed behaviors, we believe the actor ultimately intended to steal credentials to gain access to networks and systems of interest. To achieve that goal, the actors behind Sea Turtle, one, established a means to control the DNS records of the target; two, modified DNS records to point legitimate users of the target to actor-controlled servers; then, three, captured legitimate user credentials when users interacted with these actor-controlled servers." In other words, a classic man-in-the-middle attack.

They said, under initial access: "The threat actors behind the Sea Turtle campaign gained initial access either by exploiting known vulnerabilities or by sending spear-phishing emails. Talos believes that the threat actors have exploited multiple known CVEs to gain either initial access or to move laterally within an affected organization. Based on our research, we know the actor utilizes the following known exploits." So believe it or not, from 2009, so 10 years ago, a PHP code injection vulnerability affecting phpMyAdmin. From 2014, so five years ago, a remote code execution affecting the GNU bash system, and we talked about it at the time. That was that SMTP; that's the Shellshock exploit. Still there are systems that have not been fixed, that are five years later still being used.

Two years ago, a remote code execution by an authenticated user with elevated privileges against a Cisco switch. Two years ago, a remote code execution for Cisco's integrated service router 2811. Two years ago, a remote code execution affecting Apache web servers running Tomcat, still effective. A directory traversal allowing unauthorized access to Cisco's adaptive security appliances and firewalls. And, finally, from last year, Drupalgeddon, a remote code execution for websites built with Drupal that have still yet not been patched.

So those are the way people get in or move laterally. And then they said, under credential harvesting, which is the actual goal, man-in-the middle servers: "Once the threat actors accessed a domain's DNS records, the next step was to set up a man-in-the-middle framework on an actor-controlled server." They built man-in-the-middle servers that impersonated legitimate services to capture users' credentials. Once these credentials were captured, the user would then be passed to the legitimate service. To evade detection - and remember, you can't have a man-in-the-middle server with HTTPS unless that man-in-the-middle server is serving a valid TLS certificate.

So, they wrote: "To evade detection, the actors performed certificate impersonation, a technique in which the attacker obtained a certificate authority signed X.509 certificate from another provider for the same domain, imitating the one already used by the targeted organization. For example" - this is from Cisco. They said: "If a DigiCert certificate protected a website, the threat actors would obtain a certificate for the same domain but from another provider, such as Let's Encrypt or Comodo. This tactic would make detecting man-in-the-middle attack more difficult, as a user's web browser would still display the expected SSL padlock" - or the equivalent - "in the URL bar.

"When the victim entered their password into the attacker's spoofed web page, the actor would capture these credentials for future use. The only indication a victim received was a brief lag between when the user entered their information and when they obtained access to the service. This would leave almost no evidence for network defenders to discover, as legitimate network credentials were used to access the accounts." So a perfect man-in-the-middle attack that is breaching DNS, obtaining a certificate from one of these instant cert providers now, and then using that in order to prevent any indication from users' browsers from being triggered.

And, finally, they finish: "How is this tradecraft different? The threat actors behind the Sea Turtle campaign have proven to be highly capable, as they have been able to perform operations for over two years and have been undeterred by public reports documenting various aspects of their activity. This cyberthreat campaign represents the first known case of a domain name registry organization that was compromised for cyberespionage operations. In order to distinguish this activity from previous reporting on other attacks, such as those affiliated with DNSpionage, below is a list of traits that are unique to the threat actors behind the Sea Turtle campaign.

"One, these actors perform DNS hijacking through the use of actor-controlled name servers. Two, these actors have been more aggressive in their pursuit, targeting DNS registries and a number of registrars, including those that managed country-level TLDs.

These actors use Let's Encrypt, Comodo, Sectigo, and self-signed certificates in their man-in-the-middle servers to gain the initial round of credentials. And once they have access to the network, they steal the organization's legitimate SSL certificate and use it on actor-controlled servers." Did you hear that? Once they get in, they steal the organization's legitimate SSL certificate and use it then on actor-controlled servers. It's diabolical.

**Leo:** Mm-hmm.

**Steve:** So, yeah. This is a reason we need to get DNS secured, because DNSSEC would prevent this kind of attack, yet we don't have it yet.

**Leo:** Well, there you go.

**Steve:** DNSpionage.

**Leo:** DNSpionage. My friends, we have come to the conclusion of this fine episode of Security Now!, Episode 711. But we are not at the end of the conversation.

**Steve:** Oh, no.

**Leo:** No, no, my friends. It's an ongoing process. If you want to participate during the live version of the show, because we do it about 1:30 p.m. Pacific, 4:30 Eastern, 20:30 UTC on Tuesdays, you can go to our website, TWiT.tv/live. We stream audio and video. You can listen or watch there. You can also join us in the chatroom at irc.twit.tv, where it's always going on, even when Steve's not.

You can also - let's see, what else? - get versions of the show, if you want to listen after the fact on demand, from Steve's site, GRC.com, the Gibson Research Corporation. He's got nice audio files, but also transcripts to make it really easy to follow along as you listen to the show. He pays for those, and I appreciate him doing that because it really is a great value to listeners. Just go to GRC.com.

While you're there, pick up a copy of SpinRite, world's finest hard drive maintenance and recovery utility, even for floppies. You can also get lots of other stuff, as you heard, including info on SQRL, ShieldsUP!. There's so much stuff there, GRC.com. Leave questions there for Steve at GRC.com/feedback, or tweet him. He accepts direct messages at @SGgrc. And we have audio and video of the show, if you want to watch, at our site, TWiT.tv/sn. Or you can subscribe to your audio or video version, the version of your choice, and you'll get it every week, the minute it's available. And every podcatcher has Security Now!, 15 years in the making.

**Steve:** Oh, and also, now that my blog is relaunched, go over to blog.grc.com and subscribe, sign up so that you get news of stuff. I'm going to make it more active than it has been in the past.

**Leo:** Nice.

**Steve:** It was a very sleepy place until now.

**Leo:** Yeah. Well, good. Blog.grc.com. Steve, have a great week, and we'll see you next time on Security Now!.

**Steve:** Thank you, my friend.