A Look Into PHP Malware

Description: This week, Leo and I discuss the U.S. Federal Trade Commission's step into the IoT and home networking malpractice world, a radio station learning a lesson about what words NOT to repeat, Google's plan to even eliminate the checkbox, a crucial caveat to the "passwords are long enough" argument, more cause to be wary of third-party software downloads, a few follow-ups to last week's topics, a bit of miscellany, a close look at the government's Russian hacking disclosure, and a well-known piece of (related?) PHP malware.

High quality (64 kbps) mp3 audio file URL: http://media.GRC.com/sn/SN-594.mp3
Quarter size (16 kbps) mp3 audio file URL: http://media.GRC.com/sn/sn-594-lq.mp3

SHOW TEASE: It's time for Security Now!. Steve Gibson is here. In a little bit we'll cover that PHP malware that the Defense Department says was used against us by the Russkies. We'll also talk about the FTC going after D-Link - oh, this is interesting - and why Steve is going to stop using the "A" word. It's all coming up next on Security Now!.

Leo Laporte: This is Security Now! with Steve Gibson, Episode 594, recorded Tuesday, January 10th, 2017: A Look Into PHP Malware.

It's time for Security Now!, the show where we cover your security, right now, yeah, with an explanation point. And that's how right now it is. Here's the guy in charge, Steve Gibson of GRC.com. Live long and prosper, Mr. Gibson.

Steve Gibson: Yo, Leo.

Leo: You need the Vulcan salute. With thumbs out; right? That's what we decided?

Steve: Thumbs out, yes. I'll have to get in the habit. Not that I often do the Vulcan salute, so it's not a big, not a high-demand gesture. But we did learn from Nimoy himself that that's the way it should be done. So that's how it shall be done.

Leo: Nimoy himself has said.

Steve: So we said last week that we were going to cover this PHP malware. And I want to. But the Daily Beast had such nice coverage of...

Leo: Oh, it's Kevin Poulsen, that's why.

Steve: Yes, such really good coverage of what happened between reality and the government, that I thought we've got to talk about that, too. So that'll be the main topic at the end. And we've got in the meantime a bunch of news. We have the FTC finally beginning to step into essentially the world of IoT and home networking malpractice. A fun story about a radio station learning a lesson about what words not to repeat. Google now has plans to eliminate even the checkbox that we talked about last week with something called an "invisible CAPTCHA."

Also some little quick follow-ups to some things we've been talking about recently. A crucial caveat to the "passwords are not long enough" argument. More cause to be wary of third-party software downloads, as if we needed any more cause. A few follow-ups, additional things to last week's topics; a little bit of miscellany; and then we're going to talk about this whole sort of across the terrain of these Russian hacking allegations and essentially the way the government, for some reason, mishandled the thing as poorly as they did. So I think another great podcast.

Leo: Lots to talk about. All right. Let's get going here.

Steve: So our Picture of the Week.

Leo: I love this.

Steve: While you were telling our listeners about Boll & Branch, I was looking more at it, sort of checking the claim that it was authentic. And then I realized you can see the person taking the picture...

Leo: You can, yeah.

Steve: ...in the background, on a Samsung phone it looks like because I can sort of see a little bit of that vertical camera silvering that you see, and their thumb. Anyway, this is from the "this never gets old" department of yet another instance of Windows popping up when you not only least expect it, but really least want it. The caption that accompanied this tweet was, "I just wanted some water." And what the picture is, for those who can't see it - it is in the show notes, of course - is it looks like a very high-end, fancy, highly automated refrigerator. I'm not sure what you would need the screen for, though, because you've got three choices down below: Water, Crushed, and Cubed, meaning I'm sure ice, crushed ice and cubed ice. We've got a Lock Controls and a button to turn the light on. I can't see what the other...

Leo: It's got plenty of buttons, yeah.

Steve: They're sort of cropped off. Anyway, the point is that dominating this is Upgrading Windows, and it is at this point 32% along. And down in sort of blurry, out-of-focus print at the bottom it says, "Installing features and drivers, 6%." So anyway, somebody has embedded Windows in their whatever this is, looks like maybe a refrigerator.

Leo: Maybe. It could just be an image somebody put on the refrigerator.

Steve: They went to some lengths, though, if they wanted to spoof this, because we're picking up a reflection from the Upgrading Windows in the upper chrome bezel.

Leo: Well, somebody in the chatroom says, you know, it could just be they put a wallpaper on there, they put an image on the screen.

Steve: If they had access to the OS in the refrigerator.

Leo: Right, right.

Steve: I mean, that would almost be less believable than...

Leo: Good point.

Steve: ...that Windows actually is upgrading itself. I mean, this is what this picture suffers from is nobody any longer doubts its authenticity.

Leo: Exactly.

Steve: I'm getting pictures constantly from people in foreign countries, walking by kiosks that are saying, oh, sorry, we're in the middle of upgrading. And many times they're also sideways for some reason. The screen itself is oriented in a portrait version, but the boot time and the upgrade is still landscape, so it's all twisted sideways. But anyway.

So I want to take our listeners through this first piece because the details are what's interesting. And that is the FTC on the 5th, so last Thursday, in San Francisco federal court, filed a lawsuit against Taiwan-based D-Link Corporation and its U.S. subsidiary, D-Link Systems, Inc., for its failure to take steps to secure their devices, thus leaving them vulnerable to hackers. And I got the complaint and read through it. And what's interesting is it made sense as I was thinking about this, I mean, this is the branch of the federal government that is responsible for protecting consumers and dealing with things like fraudulent claims and fraudulent advertising. And, ooh, they really smack it to D-Link in this.

So this begins saying: "Plaintiff, the Federal Trade Commission (FTC), for its Complaint, brings this action under Section 13(b) of the Federal Trade Commission Act, 15 U.S.C. 53 (b), to obtain permanent injunctive relief and other equitable relief against Defendants

[meaning D-Link] for engaging in unfair or deceptive acts or practices in violation of Section 5(a) of the FTC Act, 15 U.S.C. 45(a), in connection with Defendants' failure to take reasonable steps to secure the routers and Internet-protocol cameras they designed for, marketed, and sold to United States consumers."

Then there's a bunch of definitions of who's who that I'm skipping. So then under "Defendant's Security Failures" it reads: "Defendants have failed to take reasonable steps to protect their routers and IP cameras from widely known and reasonably foreseeable risks of unauthorized access, including by failing to protect against flaws which the Open Web Application Security Project has ranked among the most critical and widespread web application vulnerabilities since at least 2007." So this sort of gives us a snapshot into their thinking.

Among other things: "Defendants repeatedly have failed to take reasonable software testing and remediation measures to protect their routers and IP cameras against well-known and easily preventable software security flaws, such as 'hard-coded'" - it actually says that, 'hard-coded' in quotes - "user credentials and other backdoors, and command injection flaws, which would allow remote attackers to gain control of consumers' devices.

"Defendant D-Link," it reads, "has failed to take reasonable steps to maintain the confidentiality" - oh, and get a load of this in the details we'll get to in a second - "of the private key that Defendant D-Link used to sign Defendants' software, including by failing to adequately restrict, monitor, and oversee handling of the key, resulting in the exposure of the private key to a public website for approximately six months. And Defendants have failed to use free software, available since at least 2008, to secure users' mobile app login credentials" - because they also include an app that allows you to view their IP camera stuff - "and instead have stored those credentials in clear, readable text on a user's mobile device." Meaning no attempt to secure these important credentials.

Then, under the section "Thousands of consumers at risk," the FTC alleges - and this is Paragraph 16. I've jumped way down. "As a result of Defendants' failures, thousands of Defendants' routers and cameras have been vulnerable to attacks that subject consumers' sensitive personal information and local networks to a significant risk of unauthorized access. In fact, the press has reported that Defendants' routers and cameras have been vulnerable to a range of such attacks and have been compromised by attackers, including by being made part of large-scale networks of computers infected by malicious software known as 'botnets.'

"The risk that attackers would exploit these vulnerabilities to harm consumers was significant. In many instances, remote attackers could take simple steps, using widely available tools, to locate and exploit Defendants' devices, which were widely known to be vulnerable. For example, remote attackers could search for vulnerable devices over the Internet and obtain their IP addresses using readily available tools, such as a popular search engine" - and of course we know what that is, Shodan - "that can locate devices running particular software versions or operating in particular locations.

"Alternatively, attackers could use readily accessible scanning tools to identify vulnerable devices operating in particular areas or on particular networks. In many instances, an attacker could then take simple steps to exploit vulnerabilities in Defendants' routers and IP cameras, impacting not only consumers who purchased these devices, but also other consumers who access the Internet in public or private locations served by the routers, or who visit locations under the IP cameras' surveillance."

And, finally: "By creating these vulnerabilities, Defendants put consumers at significant risk of harm in a variety of ways. An attacker could compromise a consumer's router, thereby obtaining unauthorized access to consumers' sensitive personal information. For example, using a compromised router, an attacker could redirect consumers seeking a legitimate financial site to a spoofed website" - we know that's all too true - "where they would unwittingly provide the attacker with sensitive financial account information. Alternatively, using a compromised router, an attacker could obtain consumers' tax returns or other files stored on the router's attached storage device, or could use a router to attack other devices on the local network such as computers, smartphones, IP cameras, or connected appliances.

"Similarly, by exploiting the vulnerabilities described in Paragraph 15, an attacker could compromise a consumer's IP camera, thereby monitoring consumers' whereabouts to target them for theft or other criminal activity or to observe and record over the Internet their personal activities and conversations, or those of their young children. In many instances, attackers could carry out such exploits covertly, such that consumers would have no reason to know that an attack was ongoing. Finally, during the time Defendant D-Link's private key was available on a public website, consumers seeking to download legitimate software from Defendants were at significant risk of downloading malware, signed by malicious actors using D-Link's private key."

So, wow. I mean, this is everything we would want to be aired in court, at where we are at this point in time. This is the government saying, hold on a second. Here is an egregious blaring instance of somebody selling equipment with hard-to-defend irresponsibility. To this point we've just sort of been shrugging. We've talked about all of these issues in the past and talked about D-Link and all of these problems. But it's just been, so, well, you know, at least it won't affect our listeners because our listeners know better. But it's affecting everybody else.

So this goes on to then really take D-Link to task. And I won't go through the details. But basically it enumerates D-Link's explicit statements of security in their promotional claims, talking about how their products are easy to secure, have advanced network security. They even talk about 128-bit security encryption, "protects your network with 128-bit AES security encryption, the same technology used in ecommerce or online banking." And so they basically demonstrate that there are all these claims being made on the promotional side, none of which are borne out by years of actual experience, and well after D-Link absolutely had plenty of time to respond. They were informed of their private key used to sign their firmware being publicly available for months and took no action to remove it or change it.

Anyway, this thing just basically shreds them. D-Link's attorneys responded, saying that this was without any sort of value, there was nothing to these claims, and they would be generating a formal legal response shortly. The second two thirds of this document, and I have a link to the PDF in the show notes, is all exhibits attached, showing snippets from the press, exhibits just from one end to the other, basically substantiating what the FTC is alleging. And this is the only way I can see that this kind of problem gets resolved, is if the FTC, which does seem to be the right body, looks at the claims being made, does the research to pull together the facts that all of the listeners of this podcast now just sort of take for granted, and says, okay, wait a minute, we're not taking this for granted. This is wrong.

And I think we're going to see that, 10 years from now, this terrain has changed. Companies won't simply be able to say anything they want to and sell knowingly insecure products into this marketplace. So this is the first step that we've seen of this. I think it's a great way to start off 2017. Wow.

So I've heard you, Leo, talking about - I think I first heard you on Sunday saying, okay, we're going to make a pledge on this network...

Leo: Not to use the "A" word.

Steve: Not to use the "A" word.

Leo: Or the "S" word.

Steve: Exactly.

Leo: Or the "G" word or the "C" word.

Steve: Exactly. And really, I only really tend to talk about the "A" word.

Leo: Okay.

Steve: And even in my show notes I said: "A radio station learns to be careful when saying the..."

Leo: I think it was TV; wasn't it? CW Channel 6.

Steve: Oh, you're right, right, right, yes, "...A-L-E-X-A." So of course we know that the perils of automated purchasing are not new. I mean, especially Amazon, who's got those little Dash buttons all over the place. The good news is they figured out not to accept another order from a Dash button until the previous one arrives. But still, I have a feeling that some households are just going to have toilet paper piling up because it just, you know, it's there. It wants to be pushed. Which of course is the brilliance of the whole concept.

Anyway, another such instance occurred last week in Dallas, Texas, when a six year old asked her family's new Amazon Echo, she said, "Can you play dollhouse with me and get me a dollhouse?" It's the household bot. Why not ask it for what Santa somehow failed to make happen two weeks before? So "The device, of course, readily complied, ordered a KidKraft Sparkle mansion dollhouse, in addition to" - and I'm not sure how this was connected - "four pounds of sugar cookies."

Leo: They were probably an additional interaction at another time, I'm guessing.

Steve: Exactly. Something else. She's like, "Well, you know, while you're at it..."

Leo: Oh, yeah, long as we're at it...

Steve: Yeah. "While I'm playing with the dollhouse, I might work up an appetite." So the parents, of course, quickly realized what had happened. And I guess they weren't able to return them because they donate - I don't know what happened to the cookies, but the dollhouse was donated to a local children's hospital. So that has a nice ending. However, the story was picked up and covered by a San Diego, California news station, CW6. At the end of the story, the anchor, Jim Patton, remarked, he said, quote: "I love the little girl saying" - and then he used the "A" word - "ordered me a dollhouse." According to CW6 news, Echo owners who were watching the broadcast found the remark triggered orders on their own devices across San Diego.

Now, the anchor, Jim Patton, didn't think that any of the devices went through with their purchases, although he would want not to. He told reporters that the station had received reports of viewer devices attempting to order a dollhouse after hearing his remarks. And of course we note from a security and technology standpoint that the Echo's settings can be adjusted through the device's app. Users can either turn off voice ordering altogether or add a passcode to prevent accidental purchases.

So I thought this fit perfectly, Leo, with our New Year's Resolution on the TWiT Network not to be glib about using those reserved words because you can actually get in trouble when people are playing these out loud, and we're within her hearing range.

Leo: I'm a little skeptical about all these claims because at least on mine it then asks me for a PIN code, which I have to enter to confirm it. And that is the default setting. So the only way that this story can really be accurate...

Steve: Oh I guess they turned them off.

Leo: ...is if they turn it off.

Steve: Right.

Leo: Which, if you have a kid, is a very bad idea. And then did all the people watching the show also turn it off? I mean, what I suspect happened is in many cases, and this is what happens if you order something...

Steve: It woke up.

Leo: It woke up. It responds and then puts it on the shopping list, but doesn't actually buy it.

Steve: Ah, okay.

Leo: So the passcode is on by default. We checked that.

Steve: Good.

Leo: I mean, I guess the parents could have turned it off. I don't know.

Steve: Yeah, and, well, you can imagine that there are - we know that. There will be some people who will get annoyed by having to give it a passcode every time. And when they see the option, they think, oh, yeah.

Leo: There's a good reason to do that. But even, you know, an ad can trigger it. Or accidentally we could be having a conversation. You don't want to trigger sales accidentally. The passcode is a very good idea.

Steve: And I had one of those devices.

Leo: It's an Echo. It's always been an Echo.

Steve: Okay. I had an Echo in my living room, where I watch television.

Leo: Yes, and it would [crosstalk].

Steve: And with the volume turned up, it will invariably fire her off every week or so. She'd suddenly wake up, and the little blue ring would sparkle a bit, and I'd be like, oh, isn't that interesting.

Leo: But it didn't order anything.

Steve: No.

Leo: Well, the funny thing is on the one in my office I've changed the trigger word to Echo. So during this whole piece you've been doing it's triggered twice. So you just can't win. You just can't win. There's three possible trigger words: the "A" word, but also "Amazon" or "Echo." And, you know, what are you going to do? There's got to be a solution to this.

Steve: I imagine, well, what'll happen - so the reason there are only three is that, as we know, they are pre-burned in the firmware. And so there is wide and well-tuned generic voice recognition. In the same way that anybody could walk down the street and say one of those words, and anyone would be able to understand them because we have really good speech recognition built into our brains, those devices have been carefully tuned to

pick up those words without training. The fancy recognition is up in the cloud. And so as we discussed last week, that word triggers the streaming of a buffer which then sends it to the cloud for detailed, maybe speaker-independent, but also trainable recognition.

Anyway, so the point is that my guess is in the future you'll have much more latitude, maybe complete latitude, where in the same way that we train our fingerprint readers on our smart phones by giving them lots of samples of our thumb many times, and it builds up a composite image, you could train whatever you wanted as the trigger word by saying it over and over and over a number of times, in a number of different situations, different distances, different volumes and pitches.

Leo: It's not the future, it's the past, unfortunately. The Moto X allowed you to do that.

Steve: Oh, good.

Leo: You could have any arbitrary trigger word. And so I actually, I forgot what mine was, something like "Open the pod bay doors, Hal." But I, you know, made it long and complex and not likely to happen by accident.

Steve: Right.

Leo: But they did it, and they don't do it anymore, and no one else has done it. So I don't know. It's more challenging, obviously. You have to train it.

Steve: Yes. And I remember seeing you sometimes not getting it to respond.

Leo: Right, right.

Steve: And so that's what you don't want. You don't want false negative responses. Neither do you want false positives. So I think it's that situation where we didn't yet - we didn't quite have the horsepower needed to pull it off well enough for it not to be more trouble than it's worth. So with today's horsepower, they're able to build very good recognition for a few words so that it will very reliably trigger. My guess is that downstream, once we're on the A27 chip from - I guess that's Apple, not Echo, but the equivalent - then we'll have more horsepower, and you'll probably be able to just say, "Hey, this is what I want; this is the phrase I want to use in order to wake you up." And then we'll be in much better shape. It'd be like everybody having the same password right now. It's just not a good idea.

So speaking of passwords, some listener - thank you - sent this to me. I wouldn't have stumbled on it myself. You can google "invisible reCAPTCHA," and you will find it. It is in beta. It is not yet released. The show notes have a nice picture of the screen you get when you bring up invisible CAPTCHA or reCAPTCHA. And following from what we talked about, it was the end of our podcast last week, this notion of how can just checking a checkbox differentiate you between human and a bot? And what we understood was that the original concept of a self-contained test was what was at fault in the very first

CAPTCHAs because, unfortunately, computers learned to read. And if they couldn't, then you could outsource this to sweatshops in other countries, and you'd have human beings reading for you in order to get these little problems solved.

So the proper solution, which we discussed last week, is if you are a company in a position to have a huge amount of background associated with a user, and Google if nothing else is that company, then you're not just bringing, like, solve this little puzzle to bear. You're bringing, oh, you're logged into Gmail in the background, or any Google property. We've got your cookie. We see your IP from which the query is being made. We know everything you've been doing all morning, all of the previous day, and so forth. Oh, and we know that this is a site that you tend to visit. You've done so 26.5 times, because once you were in a hurry and you didn't bother scrolling, in the last year.

So the point is, with all of that knowledge, why even bother with a checkbox? And they've been incrementally moving in that direction. And it's in beta now. Developers can use it. We'll be seeing it before long. They simply remove the CAPTCHA completely from the screen. And it's called the "invisible reCAPTCHA."

They have three ways to invoke this. You can have a web page that loads automatically bind the "challenge," as they call it, to your own button, or you can programmatically bind the challenge to a button, both of which just mean that you can, when you're designing your page, you can have your own "press here to confirm you're not a bot." And so it's your own button, not the fancy rendered reCAPTCHA logoed thing, just your own. And behind the screens that button is invoking a snippet of JavaScript, which is sourced from Google, and sends the query to Google that allows it to do its work, to determine if it believes this assertion you've just made.

The other thing you can do is you can programmatically invoke the challenge, meaning that just loading the page can run the JavaScript, perform the query, and return to code on the page a go-no-go assertion about whether the person who is viewing this page is or is not a bot without the user doing anything. Which is the ultimate of what we want. So essentially this is a perfect example of us having sort of passed through a problem that over time, over the course of, what, maybe six or seven years, continued to evolve until other, well, essentially network externalities were able to be brought to bear so that the whole problem just went away.

Now, it's, again, not everyone can do this with the reliability that Google is bringing because not all remote properties have the opportunity to create this kind of knowledge. Arguably, Facebook probably also does. But you want somebody that's going to have enough depth that they're going to be able to do this. And let's remember also that there is something Google is getting in return for this. For a while, when we were solving images for them, especially when we were doing the one known word and the one unknown word, that's the first time we talked about CAPTCHAs, back in the day, was we were actually digitizing eBooks for Google.

Remember that the idea would be they would verify that we were doing a not-a-nonsense problem solve by presenting a word they did know the answer to, and then also would present one they did not know the answer to. And we typed in both words and, in the process, performed a little bit of OCR for this big project that was running behind the scenes. So anyway, essentially this problem is gone now.

Oh, but I was going to say that remember that our browser is telling Google where we are and that it is we who are there. So in the same way that Facebook's Like button is, while it's convenient, it's also an explicit tracking beacon, so is the ubiquitous presence of this very nice reCAPTCHA technology. It's all going back as essentially a tracking flag,

although lots of pages already have Google Analytics on them anyway. So they're getting that information that way, even if they weren't performing this nice service for us. So I expect that, in a while, we'll just stop seeing explicit requirements to declare or prove that we're human, and somehow the system will just know. And of course now we know how that's happening.

I found an interesting post that reminded me that I needed to clarify something from our discussion last week about brute-forcing passwords. Remember that we talked about Brutalis, which was this monster multiple-GPU triple-redundant power supply, 3U-high, rack-mounted, industrial-strength cracking machine that this company was selling throughout the world to governments and law firms that wanted access, basically anybody who wanted to be able to brute-force passwords for whatever purpose. And the argument there was that the guy who had designed this, who had very deep understanding of the difficulty of password cracking of high-entropy passwords, was sort of bemoaning the fact that he was seeing people complaining that 16 characters wasn't enough. And in fact, by the way, he was aware of our discussion and sent me a tweet saying, hey, you know, that's for the shout-out about our work.

So this post asks the question, and answers it, why are more than, or he says 12-plus, 12 or more character passwords vulnerable? And he writes: "Practically speaking, people who manually create passwords above 10 characters for the most part use common words or phrases. Why do they do this? Because remembering the password 'horsebattery123' is way easier than 'GFj27ef8%k$39.'" Now, I read that on purpose because also notice how much more difficult it is even for me to say it than "horsebattery123." The fact that I can say "horsebattery123" as quickly as I can demonstrates its lack of entropy. It's the fact that horse is a thing.

Leo: It's organized. It's organized.

Steve: And battery - yes, exactly. Battery is a thing. And 123...

Leo: I love him, but I wish he'd never done that cartoon.

Steve: Xkcd? Yes.

Leo: Becomes it comes up all the time.

Steve: I know, I know.

Leo: Everybody's convinced because he's usually right. Randall, I forget, is his name Randall? I can't remember. He's usually right, but this one's not.

Steve: So he says, the author of this post: "It's just simple human behavior exhibiting path of least resistance that will always exist; and, until auto-generating password managers gain mass adoption, this vulnerability will always be around." He says: "I agree that xkcd's password strength cartoon for four random words is sound, but only for non-fast hashing algorithms like bcrypt." And then he finishes: "In this article we will

demonstrate combo and hybrid attacks using Hashcat that will expand your cracking knowledge toolkit. These examples will show how an attacker can efficiently attack this larger keyspace, with modern hardware, and make these so-called 'strong passwords' succumb to his cracking methodology," "his" meaning the attacker's cracking methodology.

And I won't go any further. I have the link to this in the show notes. But what he does is he goes through exactly these examples, like "horsebattery123." And he uses Hashcat with parameters because it includes a dictionary that has, for example, the word "horse" and the word "battery" in it.

And so you can parameterize the invocation of Hashcat to tell it how you want it to guess. And he does this with all manner of the typical passwords that people are using, which, even though they may be 12 or more characters long, his demo, he shows you screenshots with the timer. In five hours, bang, got that one. In 3.5 minutes, bang, got that one. In a few hours, bang, got that one.

So what I wanted to make sure people understood was that that second example, that took me 30 seconds just to utter, as opposed to "horsebattery123," that one, that's part of the requirement of 12 characters, or certainly 16, being enough. That is, and I did say it, but I wanted to make sure it was heard, it has to be a high-entropy password. That is, it has to be something that either you roll dice, literally, in order to choose, or you went to GRC's passwords.htm page, or you used your password manager's random number generator to produce a string. People just can't do it reliably. We have an inherent bias. There will be a pattern. Even if we try to generate 10 in a row, they will get linked.

So anyway, I just wanted to make sure that, in discussing why passwords didn't have to be 128 characters long to be good, they could be shorter, but they absolutely need to be high-entropy. And I also did not mention that the use of, on the server end, on the backend, a password-based key derivation function is crucial. That's the thing that is deliberately time-consuming. It takes what the user provides and doesn't just do an MD5 or an SHA-1 or an SHA-256. Even though SHA-256 is very nice, we now, thanks to so many crypto currencies now using that, we've got hardware that just cuts through SHA-256. Which isn't to say 256 bits isn't a lot. That's still a lot. But it does allow you to do brute-force guessing at incredibly high speeds.

So the point is you can't just do a single iteration of SHA-256. You have to iterate it. You have to arrange to make it difficult to accelerate. And of course that's what password-based key derivation functions are all about doing. So high entropy, doesn't have to be super long, and we want the server on the backend to protect us. The problem, of course, is we don't know, we have no control as users on what's going on over on the backend of this. So all we can do is choose something that's pure gibberish and have it be as long as we're comfortable with and as the website will accept, and hope for the best.

For a couple weeks now I've been seeing a mention of something that I hadn't picked up on. But we had time this week, so I wanted to note it, a chat protocol which is in many ways the granddaddy of chat protocols, XMPP. It's been around for almost 15 years. I believe the author began working on it in the late 1990s. It was originally known as Jabber and then got renamed XMPP, which stands for eXtensible Messaging and Presence Protocol. The reason it's on Security Now!'s radar is it has recently received an extension called OMEMO, which gives it end-to-end encryption and support for encrypted group messaging. So for those who don't know, it's an XML-based, so text-based, messaging platform.

That's sort of a mixed blessing. The text-basedness means that it is easy to extend, and in fact that it's been extended crazily. There's a whole bunch of standards. There's, like, four anchoring RFCs that define the protocol, which because of its age have been replaced and extended several times by successive RFCs. It enables near real-time exchange of structured, yet extensible data between any two or more network entities. It's not strong on binary sharing because it is a text-based protocol, which would require that you base64-encode any binaries that you want to send from point to point, which of course makes them about 50% larger, which means that you're going to have - it's going to be slower than a protocol that incorporates an understanding of native binary attachments or content as part of it.

XMPP/Jabber doesn't have that. But it was designed to be extensible. And as I mentioned, the extensions are numbered. And you almost - it's hard to count them. There are just so many of them. I mean, which is really a nice aspect of this for something that we want to be able to evolve over time. And in fact OMEMO is an extension added to the existing framework without the underlying protocol needing to be changed at all. And so it's in active use. There are about 10 million users of this as of 2003. So I don't know if it's gone up or down since then.

Leo: Down, because unfortunately it was a protocol for Google Talk.

Steve: And they abandoned it, yes.

Leo: And they abandoned it. So I was really disappointed. I had very high hopes for XMPP. We wanted to use XMPP for all sorts of things, for our chat and so forth. But when Google deprecated it, it was the end of the line. I mean, Jabber is still used from time to time, but it's just - it's sad because that would have been a universal protocol.

Steve: Yes.

Leo: And now we just have all these silos that are incompatible with one another.

Steve: Exactly.

Leo: In fact, I'm really sad because now that they've added encryption, it would have been a really good choice for chat and stuff.

Steve: Yeah. Well, and it does exist. There is a nice client called Conversations for Android. And, I mean, there are clients for all platforms. OMEMO is a recent - essentially OMEMO took OTR, which is the Off The Record protocol that we've discussed in the past, and added to that multi-user chat and multi-device support, meaning devices owned by a single user. And OMEMO is inspired and based upon Open Whisper Systems - I practiced pronouncing this before, and now here at the moment I can't, the Axolotl...

Leo: Yeah, Axolotl, yeah.

Steve: ...Axolotl protocol, which we've talked about extensively, which was developed for Signal. So it's based on the Axolotl Ratchet, which is the way you do this kind of protocol securely. It's not backwards-compatible with OTR, but it is, that is, OMEMO is being standardized as the new end-to-end encryption mechanism on XMPP. So that's what everyone's going to be using. You need no server-side support. That is, the XMPP server just sort of serves as the hub for Conversations. And it's only the clients that need to know how to understand this extension. It's XEP-0280 is the number. So it needs no server-side support. And there are other chat protocols, other secure chat protocols in development, OTRv4 and one called n1sec. But it looks like this one is here now and has a lot of client-side support. There's Conversations that I mentioned on Android. There's, is it Gajim, G-A-J-I-M?

Leo: I don't know. Sounds like gag 'em.

Steve: Gag 'em, probably. Cryptocat, of course, we've talked about, on iOS, Android, Mac, Linux, and Windows; ChatSecure on iOS and Android. These all support XMPP. These are XMPP clients that now support OMEMO. Monal, or monal, on iOS and Mac; the Tor Messenger on Linux, Windows, and Mac; Instantbird; Jitsi, which is a Java-based platform; Let's Chat; and Pidgin on Windows and Linux. So anyway, I just sort of wanted to put it on our listeners' radar. I know that within our group there are people who still refuse to be on Twitter and would probably like the idea. You can set up your own XMPP server. Jabber.org exists. There are both free and very inexpensive commercial services where you can get an account.

And then, if you have a reason for doing group chat, XMPP is there. And now it can be secure. It's got the same problems that we're always going to have with authentication. So you need to go through that step of making sure that the key of someone you think you're talking to is actually their key, but it has provisions for that. And once you get it up and going, it's apparently very easy to use, and solid. So anyway, just for if anyone wants to roll their own, essentially, and also use an RFC-based, standards-based, open - everything's there to inspect. Everything's on GitHub. There are links to GitHub in many of the clients that I was talking about. So it's another way to go and still have security.

Leo: Yeah. The guys who did, I think, XChat, two of the chat clients, IRC chat clients are also talking about doing their own encryption scheme. Everybody realizes the importance of encryption now, and they want to put it everywhere.

Steve: Well, I just think we're in early days, Leo.

Leo: That's right.

Steve: This will end up getting sorted out.

Leo: Yeah.

Steve: And it'd be like somebody trying to come along and do HTTPQ or something.

Leo: Right, right. Not now.

Steve: No.

Leo: Not now.

Steve: Sorry.

Leo: Let's not. Can you believe $28,000?

Steve: Mom will be 90 in March.

Leo: Don't send her your CDs.

Steve: No. She would put them in the DVD player and try to watch them.

Leo: You've got good genes. She's 90 next March? That's awesome.

Steve: Yeah, still going strong.

Leo: That is awesome.

Steve: In fact, she was complaining that she's had a little bit of a problem with macular degeneration.

Leo: Oh, it's hard to read.

Steve: And she's been using Kindles that I provided her with years ago. So at Christmas I said, "Mom, I want to show you an alternative." And so I brought up the Kindle software on my iPad. She says, "Oh, honey, I don't know what Warp Drive is, but I can read that."

Leo: Oh, nice.

Steve: So anyway, that's going to be her birthday present is I'm going to go up and get her set up with an iPad for reading because basically she just loves to read. She does that.

Leo: I set up Mom with two Echoes. And she now listens to audiobooks, which she loves.

Steve: Nice.

Leo: Yeah, yeah.

Steve: So I got a couple tweets that I wanted to share. Our friend of the show, Simon Zerafa, shared a tweet, both with Paul Thurrott and me, from someone named Kevin Beaumont, who said - he's laughing. He says, "Ahahaha, just found a Windows 10 install ISO on BitTorrent..."

Leo: Oh, that'll be good.

Steve: Uh-huh, "...which includes a scheduled task to download and run a ransomware EXE after 90 days."

Leo: Nice. Let's get some data on there and then encrypt it. Smart.

Steve: Exactly. Exactly. Wait three months and then lock it up.

Leo: Wow.

Steve: Ugh, yes. So, and again, Microsoft will pay you to download their Windows 10 install ISO, so don't go get it somewhere else. And actually the same thing is true, I have a little comment later about GRC's DNS Bench because somebody apparently found, was trying to figure out what was going on and did some googling, and there was a malware site that showed the file data, ostensibly from DNS Bench, that had all the fields empty, like file version, filename, original name, publisher, and so forth. Well, of course my software has all the fields filled in and is digitally signed.

So the problem is that when something becomes popular, bad guys take the opportunity to bundle malware as that thing and get people to run it without knowing any better. So be just, you know, especially with the death of download sites. They've just pretty much become worthless. Again, it's a nice era we went through, but it's died.

Another tweet raised a good point. I just wanted to follow up also from last week. James P. tweeted: "When I see password length limits on websites, it makes me wonder if they're not hashing the password on the backend." And I did forget to mention that last week. Again, people are complaining when websites say "Enter your password, no more than X characters." And so on one hand I immediately jumped on that as, oh, well,

they're upset because it's not allowing them longer passwords, when in fact a very reasonable concern that James raises is that, well, the fact that they even care about the length does worry us that they have a fixed amount of space in their database record for the user's password in the clear. So they're not doing anything with it except storing it.

Which reminds me of a little giggle I had when I was up with Yubico, demonstrating the SQRL, doing the whole run-through of the SQRL technology and of course showing them a demo, because of course I had to have a password length limit of 256 characters. And it says that on the field, you know, username - because I support traditional login and then side-by-side SQRL login. Anyway, the point is that they noticed, in a light gray type below the field, I said, "Password up to 256 characters." And one of them said, "Who has a password of 256 characters?" I said, "Well, nobody. But the point is it can be anything."

Leo: It's like encouraging to make it longer; right? Yeah.

Steve: Yeah, exactly. And I of all people have to have a virtually unlimited password length because everyone should.

Leo: So the point being, once something's hashed, the result of the hash is always the same length.

Steve: Yes.

Leo: That's short.

Steve: Yes.

Leo: Or relatively short. How long is it?

Steve: Well, yeah, 256 bits.

Leo: Oh, okay.

Steve: But if you divide that by eight, what do you get, 32? So it's only 32 characters.

Leo: It's not bad, yeah.

Steve: Yeah. Okay. And lastly, Aaron Bishop provided some interesting addition to our dialogue about Apple's ATS security issue. Remember we talked last week how Apple had announced at the WDC 2016 last summer that there would be a sunset on the ATS workaround, where with iOS9 it was there, and everybody should use it, but developers could turn it off. And they said we're going to stop allowing you to submit new iOS apps

that turn it off at the beginning of 2017. They backed away from that.

Well, Aaron wrote: "I'm an app developer who links to third-party websites and have had issues with ATS on some sites because they don't allow any ECDHE cipher suites." Okay, that's Elliptic Curve Diffie-Hellman Ephemeral. And we've talked about that a long time ago. Elliptic Curve, of course we know, is the faster, shorter keys, strong security. Diffie-Hellman is the so-called Diffie-Hellman Key Agreement, which is the way you negotiate in public a private key where a bad guy can see a whole dialogue going back and forth but still can't figure out what key you guys agreed to, even though he saw all of the conversation.

And then Ephemeral is really the key. That's why Apple is enforcing this. And that is that it is not a persistent, long-lived key. It is ephemeral, as the word says. It is being negotiated on the fly on a per-connection basis. Which gives you perfect forward secrecy, meaning that if anyone were ever to get any one key, it wouldn't help you either with future or previous, to decrypt future or previous conversation.

So he goes on: "ATS only allows for ECDHE, and my guess is that the sites don't allow it, either because they haven't updated and modified their cipher suites, or they're afraid of lawsuits for using elliptic curve crypto." And we covered that about a year ago, that there was a patent troll that had sued a whole bunch of large sites, saying that they had a patent on elliptic curve crypto. It's like, okay, well, we haven't had any - I've been looking for more news of that, but I haven't seen anything.

Anyway, he finishes: "Just thought you might like to know since I haven't heard anything about ATS requiring Elliptic Curve Diffie-Hellman Ephemeral with regards to patent lawsuits." Right. "I don't know if ECDHE falls under that patent, but the key words are enough for a chilling effect." So I certainly agree with Aaron on that point. And I agree - I didn't realize, and this is why I wanted to share it and bring it up, is that ATS enforces ECDHE. Well, that is relatively new. And it is the case that not all sites are supporting it, even when they do support TLS 1.2 or 1.3 and lots of the other very recent cipher suites.

So I would argue that that was a little aggressive on Apple's part. It would have been nice maybe if they'd staged that, if they'd said you've got to have TLS, and we'll give you another year for it to be ECDHE. Because after all, most of the world is not on ECDHE yet. We're still using earlier cipher suites which are universally available. So anyway, Aaron, thank you for providing that from a developer's standpoint.

And two last points. One is that I've been meaning to mention this for a couple weeks because this is a few weeks old now. Many people have noted this; and I finally, when I was putting this together, I said, okay, I'm not going to forget again. This was via a Twitter DM. A listener said: "Hi, Steve. Just got an email from Cox." He said, "Seems to be from them, based on the header." So he was skeptical. Good. He says: "As I write this, it might be your DNS Benchmark triggered this." And in fact I'm sure of it. Cox has an email they're sending out to people who have used GRC's DNS Benchmark. It reads: "Cox has identified that one or more of the computers behind your cable modem are likely infected with the Zeus Trojan bot, also known as Zbot." And it goes on, but that's the headline.

So he says: "My first question is how can Cox see what's behind our modem? We're running NAT routers. Second, the only thing that changed over the past day is that I replaced our previous routers with Linksys running dd-wrt," and he says, "a.k.a. Tomato. Could dnsmasq be causing this false positive?" And he says, "Oh. I also ran your DNS Nameserver Benchmark the other night and rebuilt our list of nameservers. Perhaps that unusual traffic was it. Third, when did Cox start caring if a key logger is installed on one

of our computers? Again, how could Cox know? Is Cox seeing a bunch of data coming out of our router all of a sudden?" And then he ends saying, parens, "(DNS Benchmark would do that.)" And he's absolutely right.

A number of people have reported that they've received such an email from Cox. Clearly there is a one-to-one correspondence between using the DNS Benchmark and receiving this note. So what has apparently happened is about a month ago, just judging, or maybe at the beginning of the year - no, it's longer than that, I think I got some at the end of 2016 - Cox decided to get proactive, which is a good thing, in looking at their subscribers' traffic. And again, I just think that's all for the better, that we wish more ISPs would be more proactive.

Unfortunately, running the Benchmark does look like, if you didn't look too carefully, like you are generating a DNS reflection attack because a DNS reflection attack sends a bunch of little queries off to a whole bunch of different DNS servers, and with a spoofed source IP. Now, of course the source IP is not spoofed. So they could be a little smarter about this and see that in fact what's actually happening is a bunch of valid DNS queries spewing out from a given client. So the only one they would ever be DoSing would be themselves. And of course the Benchmark is very careful about metering those so that you don't saturate your own bandwidth because one of the things that the Benchmark does is check the reliability of the DNS servers. So I wouldn't want to saturate the connection, or we'd cause packets to be dropped and get false positives on low reliability, that I was careful not to do.

So anyway, for anybody else who has received this letter from Cox and has shot me a note, but I never responded, I'm absolutely sure that there is a correlation. And it doesn't seem that anything goes further. Nobody's had their traffic cut off. Cox is just providing them with a warning. And that does demonstrate that, on a per-subscriber basis, Cox has deployed technology to notice if the behavior of their subscribers indicates that they may be infected. So although it's a false positive that occurs when you run the DNS Benchmark, it does say that we're seeing some positive movement in this direction from a major Internet connectivity provider.

And then this is just random, but I thought this was interesting. The first country is beginning to phase out FM radio. This is from the "The Way Things Were" department. Leo, you're probably - you're a couple years younger than me, but you probably remember when FM was new.

Leo: Yeah, oh, yeah. Big deal, yeah.

Steve: Yeah. And it was...

Leo: And it sounded so good.

Steve: Oh, so much better.

Leo: Yeah, it was stereo.

Steve: So, yes, exactly. And so of course we have AM, which stood for Amplitude

Modulation, where a high-frequency carrier has its amplitude modulated, that is, the amplitude fluctuates, and that fluctuation is the audio signal that rides on the carrier. The problem with that is that all kinds of other things could cause - like interference could cause the amplitude to be modulated for other than the audio. That's why AM is not as good as FM.

FM is Frequency Modulation, where instead of changing the amplitude of this high-frequency carrier, you subtly tweak the frequency, which actually amounts to the instantaneous phase of the sine wave that is being broadcast. And although you can have problems with reflections, frequency modulating is far more noise-resistant than amplitude modulation between the transmitter and the receiver. Thus FM was, like, way better.

Two years ago Norway announced they were going to formally phase out FM radio. And at this point about 70% of the population has the replacement already, DAB, Digital Audio Broadcasting. And I guess for whatever reason FM, I mean, like for reason of the terrain, actually, in Norway, FM is just - it's difficult to get coverage.

Leo: Oh, that's interesting. So that would make sense, yeah.

Steve: Yes. Apparently there are only five FM stations as is, and DAB allows them, within the same bandwidth, to get 20. So there are 20 DAB broadcasts and only five terrestrial FM.

Leo: It's very different here. I mean, this is a very different terrain, yeah.

Steve: Yes. And so the Norwegian government estimated then, and has updated their estimate, saying they expect that, I guess, I don't know how many stations, how many radio station corporate entities there are. But they expect $23.5 million to be saved annually by the stations switching over. People are, of course, reluctant to do this. Sixty percent of the people recently polled said, no, we'd rather just keep what we have. But, sorry. So this week the first FM station went dark. They're going to be blanking them out geographically over the course of the year. And they will be FM radio-free by the end of 2017.

And really the only losers are - there are about two million autos on the road in Norway that still have FM radios and don't yet have DAB, this Digital Audio Broadcasting. So there's a bit of concern that lack of broadcast communications for highway-traveling vehicles could pose a safety hazard. Cars should have some way of receiving notification of problems. But everybody's going to have phones and things now, too. So I don't know that that's a big problem.

Radio.no, if anyone is curious, is a Norwegian web page for the association and the country. And it's all in Norwegian until you get about three quarters of the way down, and then there's sort of like a read English, finally. I can't tell most of what its saying. But this is happening. And the U.K. has talked about doing this, too. And so it does look like in the long term we may be seeing FM go the way of the dodo bird. But probably not soon.

Leo: You know most phones sold outside the U.S. have FM radios built into them. And the same hardware in the U.S. could do it, but it's disabled by the carriers, who would far prefer you used data and pay them than listen to free radio.

Steve: Free music? What a minute.

Leo: Yeah. There is a movement from the National Association of Broadcasters in the United States, trying to force these companies to turn those FM radios on for safety reasons, if nothing else.

Steve: Right.

Leo: Internet goes down, your cell network goes down, you'd still be able to receive a radio broadcast. I'd hate to see - I don't think that that's - I think it's a nonstarter in the U.S. Although it reminds me a lot of how the FCC forced television stations to go from analog broadcasts to digital...

Steve: Precisely.

Leo: ...to conserve spectrum.

Steve: Right.

Leo: And resell that really precious spectrum they owned and make billions of dollars.

Steve: Right. In fact, I sort of assumed that's what was going on. It was like, you know, why force existing infrastructure to go dark when it's already there, and it's been amortized, I'm sure, and paid for, except maybe to recover the bandwidth. But it looks like they're repurposing it with a digital technology that will give them...

Leo: More on the same frequencies.

Steve: Exactly, more for the same bandwidth, yes.

Leo: Analog spectrum is limited, obviously.

Steve: Yeah.

Leo: But digital use of a spectrum probably is not unlimited, but it's effectively much, much - it's almost unlimited. Much better.

Steve: Right. Well, look, think about what comes over a cable.

Leo: Right.

Steve: It's just astonishing.

Leo: Right, right.

Steve: It's just astonishing.

Leo: You could see why they would want to do that.

Steve: Oh, yeah. So I got a nice note from a follower, Glasair Pilot, with an interesting - and something yet again we haven't ever talked about, about SpinRite. He said: "Hey, Steve. A SpinRite story for you. I built a RAID 10 recently." Okay, now, RAID 10, that's a combination of a one and a zero. So a RAID 0 is where you span a volume across two drives. A RAID 1 is where you mirror a volume across two drives for redundancy. And of course spanning gives you size. So a RAID 10 is both. It is essentially two drives are spanned in RAID 0, and then duplicated to another two drives to give you RAID 1 for each drive, resulting in RAID 10, you know, RAID one zero, sort of both.

So he said: "I built a RAID 10 recently using four identical brand new Western Digital Black Caviars. To my surprise, the RAID went critical twice in two weeks shortly after." And then he said: "Interestingly, the drives didn't have a problem rebuilding." He says, "Therein lays a clue. Naturally," he says, "I was disappointed since the drives were new. Since I thought I might have to fight Western Digital over RMA's" - return merchandise authorizations - "I decided to run SpinRite so I could document bad sectors or any other problems. I started out running Level 3, but SpinRite reported that would take three days at that level. So I ran Level 2 to cut it down to one.

"Surprisingly, no bad sectors reported. And SMART data was good, too. But more importantly, after running the drives through SpinRite at Level 2, no more critical RAID errors. I think what's happening is similar," he writes, "with what SSD guys are finding. If I have it right, a RAID Controller waits a certain amount of time for a drive to acknowledge a write is complete. If the drive takes too long, say, due to a sector relocation, then the controller assumes the drive has failed, and takes it offline, and marks the RAID critical."

He says: "Ostensibly, the WD Red NAS drives mitigate this in their firmware. By running a drive through SpinRite at Level 2, any questionable sectors are exercised out. And voila. The RAID has not failed since." So yet another happy and somewhat inexplicable way that SpinRite makes hard drives better. From my standpoint, you can imagine I'm a little frustrated because drives have become so sophisticated that they have become black boxes. The manufacturers consider the details proprietary, with good reason, I

think, because they are magic in the amount of storage that you're able to get in such a small space.

But as the SpinRite developer, I would dearly love to have visibility into exactly what's going on. I could do so much more, as I did once upon a time, back when there were no controllers in drives, when the controller was separate, and it had to be documented in order for software to talk to it. So essentially what's happened is SpinRite has evolved into a very intolerant exerciser of the lowest common denominator. That is, okay, if all we can rely on is this is a black box, and we're going to give it data and get it back, we're going to be really, really, really picky about that.

And so of course SpinRite does put the drive into a bunch of special maintenance modes to turn off retries, to shut down error correction, and to force the drive to operate without a lot of the bells and whistles, then makes it do that and helps to show the drive it has problems that it was rather kind of preferring to ignore. Once you run it through that process, then turn all that stuff back on again. You're back to a drive with all the bells and whistles, but also with some of the fluff and debris and dust bunnies brushed off to the side. And it works.

Leo: All right, Steve. Let's talk malware.

Steve: Okay. So, yeah. Kevin's article in the Daily Beast started with a whole bunch of sort of background stuff that we pretty much know. So I didn't want to drag everyone through it because I wanted to focus on the second portion, where he really sort of pulls this down to Earth and shares some details. So, for example, he writes - oh, and I should say the title is "How the U.S. Hobbled Its Hacking Case Against Russia and Enabled Truthers." And so he writes, jumping down to the middle:

"The department released" - and he's talking about the Department of Homeland Security and the FBI, this was their joint creation - "released 876 Internet IP addresses it says are linked to Grizzly Steppe" - which we talked about last week - "hacking, and urged network administrators everywhere to add the list" - okay, 876 IP addresses - "to their networking monitoring. Lists of IP addresses," Kevin writes, "used by hackers can be useful 'indicators of compromise' in network security. Admins can check the list against access logs, or program an intrusion detection system to sound the alarm when it sees traffic from a suspect address. But that assumes that the list is good, carefully culled, and surrounded with enough context that administrators know what to do when they get a hit." Meaning not just here's a list of 876 bad IP addresses.

Kevin writes: "The DHS list is none of these things, as Lee, founder of the cybersecurity firm Dragos, discovered when he ran the list against a stored cache of known clean traffic his company keeps around for testing. The results stunned him. He said: 'We had thousands of hits. We had an extraordinary high amount of false positives on this dataset. Six of them were Yahoo email servers.' It turns out that some, perhaps most, of the watch-listed addresses have a decidedly weak connection to the Kremlin, if any at all." Kevin writes: "In addition to the Yahoo servers, about 44% of the addresses are exit nodes in the Tor anonymity network."

Now, I have seen a different number quoted elsewhere. I saw a number of 15%. But still it's ridiculous to think that Tor exit nodes mean anything. I mean, we know what they are. They're general-purpose servers where the traffic is finally decrypted after its last hop through the Tor network and emerges onto the public Internet. So, yeah, bad guys use it, but so do all kinds of good people who just would like to have the privilege of

anonymity on the Internet. So what I got a kick out of was that, thanks to Kevin's work, we did get a bit more information about what triggered that Vermont electric utility concern that we discussed last week.

He writes: "The consequences of the over-inclusive list became apparent last week, when a Vermont utility company, Burlington Electric Department, followed DHS's advice and added the addresses to its network monitoring setup," as DHS said to. "It got an alert within a day. The utility called the feds. The Washington Post soon broke the distressing news that 'Russian hackers penetrated the U.S. electricity grid through a utility in Vermont." Well, of course we debunked that already last week because it was clearly not the case.

And then Kevin says: "The story was wrong. Not only was the laptop in question isolated from the utility's control systems, the IP address that triggered the alert wasn't dangerous at all. It was one of the Yahoo servers" - the Yahoo email servers - "on the DHS list, and the alert had been generated by a Burlington Electric employee checking his email. The Post article was later corrected, but not before Vermont Senator Patrick Leahy issued a statement condemning the putative Russian attack." Oh, good lord.

So anyway, the good news is there's some meat here. Kevin says: "But to analysts in the computer security industry, the hackers are old, familiar adversaries." I skipped a big bunch of stuff that we sort of already understand, where they were just talking about how there was way more smoke than substance in this. But so he says these hackers are old, familiar adversaries that they, meaning the security industry, have been watching under a microscope for the better part of a decade.

"The first group, called Fancy Bear or APT28, has been active since at least mid-2007. The group typically begins its attacks with targeted spearphishing emails" - oh, my god, Leo, I forgot to mention. On Sunday I heard Reince Priebus. And I didn't verify it, but he did say on national television that John Podesta's email account's password was "password."

Leo: Yeah, I don't think that's true.

Steve: Okay.

Leo: I think that's more fake news.

Steve: Okay. Good. Because...

Leo: I'd like to see the verifying document on that.

Steve: Yeah. So we know what spearphishing is. "Then the group installs backdoors controlled through a cloud of command-and-control servers deployed around the world. Its targets have included NATO, several U.S. defense contractors, the German Parliament and, after Russia's doping scandal began, the World Anti-Doping Agency. One of the command-and-control servers used in the DNC hack was reportedly also used in the Bundestag intrusion."

Leo: Bundestag.

Steve: Thank you.

Leo: That's the German Parliament.

Steve: "The other group, commonly called the Dukes or APT29, was first spotted operating in Chechnya in 2008. Stealthier and more cautious than Fancy Bear, the Dukes have nonetheless been detected infiltrating the White House, the State Department, and the Joint Chiefs of Staff. Known for innovation - one attack campaign used Twitter as a command-and-control channel - they have their own fleet of customizable malware, including a program called Seaduke [S-E-A-D-U-K-E] that they only bring out for the really important targets, and which was found again on the DNC's network.

"Security companies," Kevin writes, "can tell you much more about these groups, their code, their infrastructures, and their methods. F-Secure has an excellent 34-page write-up of the Dukes, and FireEye has a deep dive into Fancy Bear, among many other reports from different companies on the 'Net. From analysis of the dozens of malware packages used exclusively by these hackers, researchers can tell you that they're usually compiled on machines with the language set to Russian. Both groups operate during working hours in Russia and take Russian holidays off." And these are of course attributes that we've talked about in the past as being signals that are used to give some sense for what's actually going on.

"Their targets are radically different from those of for-profit criminal hackers in Eastern Europe or anywhere else - no banks, no retailers with credit card numbers to steal, always governments, companies, journalists, NGOs, and other targets that the Russian government would be interested in. In other words, these hackers don't operate like 14 year olds" - or the 400-pound person on the bed. "They sometimes use off-the-shelf hacking tools, but more often they deploy industrial-scale malware no teenagers have access to. They hit targets of interest to spies, not kids. And virtually all the public analysis of these two groups concluded - well before it became a political issue with the DNC hack - that they are likely controlled by the Russian government.

"The evidence, then, that Russia interfered with the election is already solid" - well, okay, to whatever degree that it mattered. Well, of course DNC is part of the election, but okay - "and is supported by years of work by the security industry." Lee again, from Dragos, notes: "If you've been following along, all the evidence that matters is already public. This is one case out of hundreds that they've investigated involving the same hackers. It's all very, very consistent. It all makes sense. It's all very, very solid," he says. "It's just that the government is now confusing everyone."

And so in my notes I wrote, you know, how are we to understand what happened here? And all I can think is perhaps it's a mixture of Internet-illiterate politicians, coupled with the demands of bureaucratic management. It must be that we have, although this would shake anyone's confidence, probably deep within the NSA, skilled professional hackers who know all of this, who are no more happy than the security industry and informed Internet-savvy people are with the nature of this disclosure, which has just mucked everything up. And this must have been, I don't know, like calling most companies and speaking to technical support. We know you're not talking to an expert when you talk to the frontline tech support person. You wind up talking to someone who thinks IP is

something that happens when you wake up in the middle of the night, rather than an Internet Protocol address.

Leo: That's a good line. I like it. I'm going to steal it.

Steve: That's original. So anyway, so that's the story with the report. Now, part of that was this, oh, we found some malware. Okay, well, get a load of this. First of all, it's publicly available. The version that they reported is known as "PAS." And the DHS talked about v3.1.0, which is relatively old. There is 3.1.7 is freely available, and they're already at a v4. And it's sort of generic PHP-based malware.

So, okay. So let's step back for a second. We've talked around a lot about PHP and other server-side technologies. Essentially there's, just to give you a little bit of background, there are sort of three ways to skin the cat of enhancing a web page. Without any of that, you end up with a Tim Berners-Lee, click a link, there's a document. And you scroll through it, and it has other hypertext links, which you click, and then it takes you to other linked documents. That was the original web.

Of course then we said okay, we've got to make it do more. So there's the server-side approach, which has two different flavors. There's native code, which is my style. For example, ShieldsUP!, GRC's ecommerce system, the Perfect Passwords page, the SSL Fingerprints, the Cookie Forensics, the DNS Spoofability, all of those things, those services that I've created over the years, are enhancements to the web server. So they're extensions to the web server.

In fact, many people have seen ne.dll in the URL. That's my own DLL. NE stands for Net Engine. And so that's that thing. It's my own extension to Microsoft's IIS server, which has been growing. And the idea is that it can intercept things coming in. So, for example, it sees passwords.htm. And on the way to the server it converts the URL into an ne.dll something in order to access that DLL service on the backend, which then uses a very high-quality pseudorandom number generator, and then presents a static page - and this is the key - a static page customized for that particular instance. So the code running on the backend on the server is involved in dynamically creating the page. It's not just dumping an HTML file from the server's drive out onto the user. So it's involved in doing it on the fly.

Then the second approach is to move the dynamism to the client side. And that's, of course, what scripting, client-side scripting gives us. And of course I've done that where necessary. The Off The Grid tool that generated Latin squares locally, on the user's browser, and the Password Haystacks is a perfect example where, as you're typing in, the haystack calculator is showing the alphabet size and the length of time that password, current to that keystroke, would be updated. So that's JavaScript running in the browser, which is the way you create the greatest level of interactivity. I could have done that, but I would have had to have a roundtrip to GRC and back and be like constantly updating the whole page, which just isn't the way to do it.

The third approach is sort of a compromise. And it's an interesting solution. It's code on the server side which is interpreted sort of on the way to the user. So there's a static page that the server delivers to this middle layer, which it could be Perl or PHP, Ruby, Java, Python, or even JavaScript on the server side. And so the idea is that something like PHP, it scans the page, looking for something that it has to do. There will be a specific escape character sequence that is the key that says, "The following is PHP code." So when the PHP interpreter scanning the page sees that, it reads that as code and then

executes the code and typically replaces that code with the result of the code execution. So, for example, it might make a sequel database backend query, or a bunch of them, or do any amount of work.

And so the idea is, as the page is leaving the website, this scripting language, essentially, an interpreted scripting language on the server takes a static page and executes the code in the page. And then what the user gets is the result of that. There's no more PHP anymore on the page. It's been replaced with whatever that PHP code resolved to. So those are sort of the - those are the different ways you bring pages alive today.

So the server-side scripting is a very powerful architecture, and it's clever. I mean, I've had to write native code, well, because I want to, in my case in assembly language. Apache modules are another example of native server-side extensions. And in fact those interpreters are implemented as server-side modules of the various servers, if you want to include them in your server. So it's a powerful architecture and, again, a nice compromise because it allows you to have a nice purpose-built language made for, in the case for example of PHP, made for expressing web page content, rather than a more general purpose language that would provide less help to the author. But it comes with a great responsibility because now the web page content is being scanned by this extension to the web server for things to do, which it then does on the fly.

So this is exactly the same as with all the problems we've talked about with SQL databases over the years, where, for example, web pages would, well, intended to embed commands to retrieve and display data from backend SQL databases. But if attackers were ever able to get their own submitted SQL commands to be displayed and interpreted by the web server, commands like "drop table" can inflict significant damage. So you see what's happening here. The idea is that the way the scripting interpretation works is that the page that's being served contains invocations of the interpreter which evoke the interpreter to do something.

But the danger is, the responsibility is that it's just web content. So sites that aren't really careful about how they display the web content, if a site blindly takes a blob of text, for example, a forum submission, and displays it to the user in front of them, and that display hasn't been filtered for the escape characters to invoke a PHP interpreter, then the PHP interpreter will be invoked and run the code that the bad guy submitted to the forum posting. So while it's powerful, as we always see, with that great power comes great responsibility.

And so this is what's tricky about PHP. Now, it's also why it's an attack vector, because if any mistakes are made somewhere, then that provides an opportunity for a bad guy to run their code. And PHP is a powerful language. There is a working implementation of SQRL in PHP, the server-side SQRL. So you could do a lot of things with it.

Okay, so specifically digging into this PAS 3.1.0, having already said there's not that much special about it. What it is, is a PHP function which contains an encrypted blob of text which is decrypted when the attacker supplies the decryption password. After the PHP blob is uploaded to a web server, the attacker accesses the file somehow - and we'll get to that in a second - through a web browser, provides the password, which is then stored as a cookie so that it no longer needs to be reentered because PHP is able to look at the site's own cookies as they come and go, and so that allows the user to put the password in only once and then use this freely as essentially a function-enhanced web page. So the blob is decrypted with a password and then executed on the web server. That is, remember, it's executed by the PHP extension, which has a lot of power on the server.

So in the case of this PAS 3.1.0, it is what's known as a "web shell," which is a multipurpose, well-known type of toolkit that is often found in forensic examinations. It contains in this case a file browser and explorer, allowing a remote attacker to just bring up the contents of the drives that the system has access to and browse around. It's got a file search function, a database client to download the contents of the site's database, network tools including a port scanner and the ability to bind a service to a port. So essentially it's able to create a server on the fly. Hooking a service, it will then accept incoming traffic, assuming that it's able to get to the server.

A tool to brute-force attack passwords on FTP and POP3 email, a command line client to run arbitrary operating system commands, I mean, you could do anything you want with this thing. And a utility to view the server's configuration info in order to glean more information about the site and the way it's configured. And it is freely available on the Internet, with an optional bitcoin support donation. The site that offers it has a form you fill out. You put in the password you would like the blob encrypted with, and it then says, okay, here's your download link. You download that, and it is preencrypted for you. You arrange somehow to get a site to display it. And when it prompts you for the password, you enter that, and you then have all of those tools at your fingertips.

So the remaining question is how does such malware infect a PHP site, like WordPress, which of course is PHP-driven. So that's unclear. Nothing in the document that the government provided, provided that information. And, I would argue, maybe they know. Maybe what we're looking at is the result of massive redaction so that, like, all of the good stuff was taken out. Because there was, as we know, a private meeting that both President Obama and President-elect Trump had with the security agencies, which was not made public, and maybe more, like more compelling information was available.

But the point that Kevin was making was that we already know all of this. That is, this was just not an attack by these people only on the DNC. It has been an ongoing campaign for a long time with known groups where there is a strong reason to believe that they are based in Russia. We have to draw the connection to who supports them and gives them their marching orders. But all of those things, all of the facts line up.

So how does it get in? The guys who have looked at this a lot say that, well, WordPress website owners might have other malware installed on their workstations, and that malware attempts to install PAS, P-A-S, this malware, on their WordPress websites. Maybe a related cross-site request forgery, we've talked about those before, CSRF vulnerability is used to install the malware. Maybe unwitting users are voluntarily installing this on their own websites after downloading it from a malicious website, thinking it's safe. That is, like, oh, look, it's password-protected. I can do whatever I want to with it. It'll allow me to do things remotely on my own server. Bad idea. Or attackers could be compromising websites through some other means, and then using compromised credentials obtained to manually sign in and install PAS with a standard browser. And those sign-ins could be partially or fully automated.

So anyway, the bottom line is there is nothing whatsoever particularly impressive or unique about this particular piece of malware. Anybody who wants it could download it by the end of the podcast. It's not clear why DHS and FBI provided it. Maybe they felt they had to provide something.

Leo: Well, it was in hexadecimal. It looked really good.

Steve: It's gibberish. Leo, it was encrypted. It must be, you know, encryption, all that

encrypted stuff.

Leo: Yeah.

Steve: Yeah. So that's what we have. I think we have bureaucracy. Maybe the intelligence community is protecting its sources and methods, and we want them to do that. The presumption is, from what we learned from Snowden, there's a lot of stuff with crazy code names that is like, you know, freaky technology that actually exists. We saw pictures of it, and we saw slides. So none of that is represented by this ridiculous report portion that was made public. And it was unimpressive. But what I think this really says is that, yeah, there is ongoing cyber, I don't want to say warfare, but cyber intrusion. We are probably - we the U.S. are probably doing it every bit as much and as successfully as teams in Russia and teams in China and teams in the U.K. and wherever else. I mean, that just seems to be something going on.

And, you know, in the press, I've been thinking about this, in the press there's been a lot of, oh, what does this mean? Is this going to be an escalation? Is this the new arms race? And I was thinking, well, you know, the nuclear powers all have nuclear missiles aimed at each other. And no one has fired them because there just sort of seems to be this mutual deterrent effect. And meanwhile, we've become globally, thanks to the Internet, really closely knit together. I mean, there is strong financial ties now. And even though Russia may not need us, well, China buys a lot of Russian energy products, and we transact a lot with China. So even if it's one step removed, we're connected. Nobody wants the economic side to come tumbling down.

So I think countries are establishing this technology. They probably, you know, we call them "implants." We probably have implants throughout the critical infrastructures of these other countries. We're doing our best to thwart theirs, but they probably have them, too. And this is just sort of the way it's going to be. And meanwhile the script kiddies do their script kiddie stuff on an entirely different level than what the state actors are doing. And we saw sort of a snapshot of it, but it certainly wasn't very impressive.

Leo: Well, there you have it. Thank you, Steve Gibson. GRC.com. Everybody should go there and get the latest copy of SpinRite, the world's finest hard drive maintenance and recovery utility. That's Steve's bread and butter.

Steve: A couple people did while I was talking.

Leo: I heard the yabba dabba doos.

Steve: So thank you, whoever you were.

Leo: Keep those yabba dabbas coming. You can also get the podcast there. He has audio and transcripts at GRC.com, and lots of free stuff. I mean, it's a great site for, I mean, it's just a treasure trove. You can just browse your little heart out there: GRC.com.

We have audio and video at our site, too, of course, TWiT.tv/sn. And we put it on YouTube. By the way, we're on YouTube Live now.

Steve: Nice.

Leo: If you go to YouTube.com/twit, you can watch the live stream there as well as on our website, as well as using those apps. There are so many apps. There's like five or six apps on Apple TV, but all of them have live streaming. Or get the show after the fact, watch on YouTube or download. I think downloading is the best thing to do. Whatever. You know, we don't care how you get it, just get it. You don't want to miss a single episode of this show.

We record on the air, so you can watch it if you want live, every Tuesday, Wednesday, Tuesday at 1:30 - I have to think, what day is this? Tuesday at 1:30 Pacific, 4:30 p.m. Eastern time, 21:30 UTC. You just go watch the live stream, and you'll see a little bit different version of the show than the one we put out, but it's pretty much the same thing. You can also, let's see, what else? Get the app and subscribe. You'll have a collection. There's lots of ways to play. Steve, back next week. And we're going to do questions; right?

Steve: I'm not sure. I've got a bunch of topics to talk about. I'm sort of - a couple of people have suggested I sort of fold some in, as I did this time.

Leo: Oh, we could do that.

Steve: Some feedback from our listeners. So when it's topical and it makes sense. But maybe we'll do a Q&A. I'll check the mailbag. And if it's compelling, that'll sell me.

Leo: Okay. The mailbag really isn't a mail or a bag. It's you go to GRC.com/feedback, and you can leave a comment there. Or really I think most people now use Twitter. Steve's Twitter handle: @SGgrc. @SGgrc. Thanks, Steve. We'll see you next time on Security Now!.

Steve: Thanks, Leo.