Reaper Redux

Description: This week we examine the source of WannaCry, a new privacy feature for Firefox, Google's planned removal of HPKP, the idea of visual objects as a second factor, an iOS camera privacy concern, the CAPTCHA wars, a horrifying glimpse into a non-Net Neutrality world, the Coinhive DNS hijack, the new Bad Rabbit cryptomalware, a Win10 anti-cryptomalware security tip, spying vacuum cleaners, a new Amazon service, some loopback Q&A with our listeners, and another look at the Reaper botnet.

High quality (64 kbps) mp3 audio file URL: http://media.GRC.com/sn/SN-635.mp3
Quarter size (16 kbps) mp3 audio file URL: http://media.GRC.com/sn/sn-635-lq.mp3

SHOW TEASE: It's time for Security Now!, the Halloween edition. Actually, every week is a scary show because we talk about security flaws. This week we've got of course the wide range of things to talk about, but then Steve's going to talk in more detail about the Reaper worm that's making its way through mostly routers, millions of routers all over the world. Actually, it's not yet a million, but it could be soon. Steve explains why this thing is written better than most IoT devices. That's coming up next.

Leo Laporte: This is Security Now! with Steve Gibson, Episode 635, recorded Tuesday, October 31st, 2017: Reaper Redux.

It's time for Security Now!, the show where we get together with this guy right here and chat about security, privacy, the Internet, how computers work, anything on his mind - Steve Gibson. It's really a rare opportunity just to sit down once a week and spend some time talking to Steve, and I always look forward to that. Hi, Steve.

Steve Gibson: Yeah, rare. We're in Year 13, Leo, and this is Episode No. 635.

Leo: Oh, I would do it every day if I could, so once a week seems rare to me.

Steve: And mostly what we talk about is the events of the previous week, that happened between the previous rare opportunity for us to talk and the current rare opportunity, which is pretty much guaranteed to happen since I think in the entire 13 or 12-plus years we've only missed one and got a lot of negative feedback for that.

Leo: Not allowed to miss a show anymore, no.

Steve: Not going to do that again. So No. 635 this week for Halloween, October 31st of 2017. I titled this "Reaper Redux" because many other security firms have had a chance to weigh in on their own observations. And whereas last week's initial reporting was maybe a little bit of hair-on-fire hysteria, we have now had a chance to examine it more, understand what Reaper is and, interestingly, is not doing to get a better sense for what the numbers are and what they mean. Depending upon who you ask, the two million number was inflated, or maybe it's just pending and more. So we're going to wrap up today's podcast by talking about that.

But we also have the attribution has finally landed for the WannaCry attacks from middle of May, earlier this year. We've got a new privacy feature coming for Firefox from the Tor version of Firefox, which is not the first time that it's happened. Google has announced they're removing an interesting high security feature from Chrome because it didn't really work out as well as they were hoping.

Leo: Oh, yeah, I was hoping you'd talk about that, yeah.

Steve: Yeah, the HTTP public key pinning that they're going to be taking out. I ignored this last week, although a bunch of our listeners sent me a note saying, hey, what do you think about this, the idea of using visual objects as a second factor for authentication. So I want to sort of address that as this specific bit of news and also as a concept, and why I think it fails.

The guy who did that really interesting login spoofing demo for iOS that you and I talked about a couple weeks ago, where you just couldn't tell that it wasn't the OS asking for your credentials, it was a malicious app, Felix, he's done an interesting new blog post about a concern over iOS's camera privacy that I want to talk about.

We've got an interesting escalation of the CAPTCHA wars. A horrifying glimpse into the world of non-Net Neutrality brought to us by an ISP in Portugal, where there is no notion of Net Neutrality. And it's just like, oh, no, please don't let that happen here. It's their page of services that they're selling. We've also got the relatively high-profile DNS hijacking of Coinhive, which occurred for sad reasons. You would hope that somebody like Coinhive would be doing their own security better than they were.

We've got a new cryptomalware has emerged known as "Bad Rabbit." We've got Ed Bott bringing us a Windows 10 anti-cryptomalware security tip, a concern from the Israeli security firm whose name is escaping me right now, it'll come back, over spying vacuum cleaners. LG, it turns out, had a problem with their whole home IoT system which would have allowed that. A new Amazon service which is interesting, which is just beginning to happen, with some feedback from our listeners if we have any time. And we will revisit the Reaper Botnet with the advantage of a week's more maturity in our understanding. And of course a fun Picture of the Week. So I think another great podcast for one of these rare episodes.

Leo: As always. You're the master of great podcasts, Steven. All right. Continue on.

Steve: So as we know, reliable attribution for cyberattacks is always problematical. It's made difficult by the ability for attacks to be looped through or bounced off of remotely located innocent machines. And, of course, I mean, it's a constant, it's a staple of detective novels, and you see it on TV all the time. Oh, they bounced it through 13 different countries before it arrived so we don't really know where the bad guy is. I mean, and that sounds questionable, but of course we know it's absolutely true. It happens. So this makes determining where things came from difficult. But careful forensic research and reverse-engineering of samples of the attack tools which are caught often reveals some clues. And we've talked about those over the years.

In the case of WannaCry, which as we know was a devastating cyberattack which leveraged a flaw in Windows SMBv1 which erroneously we believed it was affecting Windows XP, but it turns out that a later more careful analysis showed that a lot of Windows 7 machines and Server 2008, which is the equivalent, were also affected. So the U.K. and other nations have arrived at a consensus that the cyberattacks which we've now labeled as WannaCry were sourced by North Korea. North Korea is known to have about a 6,000-strong cyber hacker group which is doing these things. And there was no attribution until just recently where finally, after the analysis was done, it's like, okay, we've got to hang this one on North Korea.

So although the SMB attacks are still occurring, we're about to talk about, a little bit later, Bad Rabbit, which is using a different version of an attack which escaped from the NSA's control. WannaCry has been patched from Microsoft even in Windows XP, which surprised a lot of people. But it generated so much negative press for them that they had to do it. So but we know that it's very different to have a patch available and a patch applied. Which is why the point you made, for example, Leo, about the light.house taking responsibility for its own firmware, that's going to end up immediately being required behavior for IoT devices, much like the Ring Doorbell does and other things are.

For example, we'll be talking about - I'm blanking on the name - the Reaper and the fact that it is functioning by commandeering a whole bunch of routers which are high power, more power all the time as they become more capable. But typically routers are not auto updating. They offer you the option, but you've got to log into the management interface and say Check for Updates. That's not something they do autonomously, and that behavior has to change.

There has to be some means where the router figures out, okay, even though it means I'm going to have to go offline for a while - which is probably why they don't do that is it brings your whole network down and is very disruptive. Or maybe they need to come up with a way of being able to load another version of their firmware and then do a less interruptive update in order to bring a new kernel online. So anyway, some way this problem has to happen. So anyway, we now know with strong certainty, as it was stated, that North Korea were the perpetrators of this WannaCry attack, which made a lot of press because it was so damaging. And it was a cryptocurrency attack that was requiring ransom payment of bitcoin in order to get things decrypted.

In Firefox v58, which I think it's in January, yes, January 16th of 2018, so a few months from now, middle of January, we will be getting, in the normal public standard version of Firefox, a feature which has been four years in coming because it was noted as would be useful from the Tor version of Firefox. The Tor browser is an ESR, the ESR version of the normal Firefox mainstream, where they've taken that and added a number of privacy enhancements for packaging as part of the Tor solution, The Onion Router that is all about protecting your privacy. And we've also talked often about the problem of browser fingerprinting.

And I know, Leo, you've been talking about it on other podcasts recently. I mean, it is a big problem. But there are so many things about a browser which script running in the browser is able to send back to a site that wants to identify your browser uniquely. Traditionally it's been cookies. Cookies are a little controversial, of course, because especially in the EU we're beginning to see privacy requirements where sites have to disclose when they are just using cookies even the way they were intended to, just to create a persistent session on a first-party basis with their user. So we're often now seeing sites bringing up a little banner that you are asked to click Okay on, just to acknowledge that you accept that this site is using cookies.

So what's interesting about fingerprinting is it sidesteps those EU privacy concerns because they're not storing something that the server has given your browser to return, which is what a cookie is. Rather it's just a "passive," passive query of existing information about your browser. For example, it turns out that one of the things that tends to be very unique from one browser to another, even the same browser make and model on the same OS, is what particular plugins that the particular browser's user has chosen. I know I have a set that I use, and we've talked about them, things like uBlock Origin.

I have a YouTube downloader that I sometimes use. I have something that does a nice job of capturing screenshots, where even if the page scrolls off of the screen, this thing somehow, it really works well, is able to give me a screenshot of the entire scrolling page. I also use a session manager for Firefox, and of course I like tabs, so I've got the tab styling plugin. The point is that it's probably the case that nobody else maybe in the world has exactly my same exact set of chosen browser add-ons. Well, the browser add-ons, the plugins, is something that script running in your browser is able to enumerate. And so it provides a passive, yet unique, fingerprint for me, sadly. I mean, I'm not happy about it, but it does.

Another thing which tends to be sort of system specific are the fonts which your particular system has installed. And I know, for example, that over time I'll say, oh, I like that font, or I like this font. And so they tend to accumulate. And so while they may vary a little bit, they also give a unique snapshot, a unique fingerprint into a specific system.

Now, as it happens, in the interest of privacy, Firefox 52 blocked the enumeration of system fonts by JavaScript because it's really kind of hard to make a convincing use case for why script running in your browser needs to rummage around in your system fonts in order to get a complete list. The way the DOM works and the way CSS works is you're able in the CSS code to list in sequence of preference the font families that you would like the CSS and the browser to use and leave that up to the browser. Yeah, okay. Maybe it's interesting to make it more automated where the script is able to see what you've got and then rewrite the DOM on the fly. But we could live without that. And because system fonts are - it's hard to make a case for that. Firefox 52 blocked that.

So Tor four years ago blocked something else which is also very powerful. And that is, and it's something we've talked about in years past, the "canvas." The canvas is a drawing surface which I have played with. If you go to GRC.com/animation.htm, you will see a piece of my work. That is the canvas. That is JavaScript which I wrote using the canvas to create an animation of the way a hard drive converts data bits into flux reversals, and the way those are then picked up by the read head and turned back into data bits. And so that is a 640-by-480 canvas where JavaScript is animating that in order to make that happen.

Well, what happens is it turns out that you can draw on the canvas with script, and then you can scan the pixels of the canvas to get the RGB values of each individual pixel. And

it turns out that that's something super valuable for fingerprinting. As a consequence of browsers rendering details of text and lines slightly differently, like just the details of if the X and Y coordinates are even and odd, and the line is at a certain rate, how is the anti-aliasing performed where the line below begins to pick up some of the color from the line above as the line slowly crosses the raster scan, or the way the TrueType settings have been used in order to use the RGBness to anti-alias the actual pixels of text

And in the show notes here, I have samples from some of the coverage of this of the capital letter "T" rendered in different browsers on the same machine on exactly the same web content. So that "T" was being displayed essentially by the same page on two different browsers. And if you look closely, you can see slight differences in the coloration. The very first upper left pixel of the left-hand "T" is a little pink, where it's not at all on the right-hand "T." And in fact the whole left-hand bar of the left "T" is red, where it's more blue on the right-hand "T." So the point is that, if you were to render the two T's and then pull out the RGB and just hash the result, you don't care that it's a "T," you just hash the result, what you will get is a unique hash which is able to discriminate between those two browsers.

And it turns out that the OS, the GPU, if it's used for acceleration, the graphics driver, the display adapter, all the various things get involved in this and end up producing another unique fingerprintable thing which is not data being given to the browser, but just something that it's possible for script to read which is always unique. And the Tor browser said, hold on here. There is no good reason why JavaScript needs to read pixels from an offscreen canvas, which is what's being done is that a canvas is set up that you don't even see, where something is rendered, and then it is scanned and then sent back. And so the version of Firefox which the Tor guys took and then modified shut that down four years ago. That was just yesterday logged as a bug fix, which is just the terminology they use in their bug tracking, which we will see in Firefox 58 as a further advancement in Firefox privacy.

So that's something which is on the rise, that is, the use of browser fingerprinting is on the rise because it avoids the latest privacy legislation, which is beginning to shut down the traditional means of tracking people, and so they're switching to passive fingerprinting. And the good news is Firefox is beginning to sort of profile itself more as the privacy-protecting browser. It'll be interesting to see whether we see Chrome and Google following over time, moving in that same direction. We may see some differentiation between these two browsers over time, if Firefox continues in that direction and Google decides, well, we're going to focus on performance and security and not do anything to thwart tracking and profiling. We'll see.

But speaking of Google and Chrome, it turns out that not all RFCs end up gaining traction and being a good thing. There's a technology that we've touched on, but I've never really talked about too much because it always seemed problematical to me, and it looks like that's what the industry has decided. It will be leaving Chrome. It has been supported. It will be leaving Chrome early next summer, in May of 2018. And it's known as Public Key Pinning, or HTTP Public Key Pinning, or sometimes just by its initials, HPKP. And it's ended up being problematical. It uses a fun acronym, just because it sounds good, and that's TOFU, which is Trust On First Use, T-O-F-U.

The idea is that a website could offer in its reply headers some public key pins, meaning "pin" as in "pinning," meaning the SHA-256 signatures of the certificates that it uses, its own TLS certificates. So the idea is that this would be a means for a website to supply compliant browsers, that is, HPKP browsers, with the upfront knowledge of these are the signatures of the certificates that we use. Such a browser would then honor that assertion absolutely. And that's the problem, is if anything happens to break the validity

of that assertion, then a compliant browser will refuse to show you the site, period. That is, the problem is that in order for this to be useful, because it's going to be received the first time a browser goes to a site, and cannot be replaced because if it could be replaced with an update, then that's what an attacker would do.

So the only way these signatures get accepted is if there's a blank space for that domain in the browser. And once that blank space has been filled with the signatures, nothing can change them. Now, they do expire. But there again the max age has to be long, otherwise again they're not really of much use because, if they're expiring all the time, then they may have expired when a bad guy has performed a DNS attack or arranged to be a man in the middle or whatever, in which case again they're of no use. So they're only of use if they have a really long expiration date.

But while they are in place, you cannot change your certificate under any, I mean, you can't. There's no way to change this. And so it's like, yes, it's really good security until it's really bad security. And really bad means no one who's using a compliant browser who's ever visited you before can go to your site if your certificate's changed. And we know that, first of all, certificates expire, and a new certificate has to have a new signature by definition. There's all kinds of time and date stamps in certificates that is going to force a new certificate to be and to have a new signature. And we do know that sites sometimes have their certificates expire without them knowing it. So to use this mechanism responsibly and not have it bite you in the butt, it means you have to be very careful and cognizant and preemptive about bringing the expiration age down in anticipation of a planned whole Internet revamp of any PINs that exist in browsers.

And anyway, so what has ended up happening is it's too brittle. It's too fragile. It's too error prone. It was a good idea, but it was more trouble than it was worth. And it is going away. So not all ideas that surface as RFCs end up being a good idea, and this one is dying. It will be leaving Chrome. And it turns out it never really gained much traction. I remember seeing, as I was doing some research into this, something like 0.04% of the top bunch of Alexa's top million sites or something were using this. I mean, it just - you really have to have a need. And, boy, you've got to be responsible if you're not going to end up just creating your own denial-of-service attack on your own site by needing to change your certificates, but having pins not expired in browsers that are compliant. So, yeah, steer clear of that.

I mentioned that last week a number of people were asking me about this idea of an image as a second factor. This was a research paper that drove this, and the Verge has some coverage of it that I have a link to in the show notes, if anyone's interested. It was never meant to be implemented. It was just them experimenting. And so the idea is that, in their coverage and example, that somebody might have some fancy Aztec bracelet with lots of detail and some rhinestones or aquamarines or whatever, and they would use that as their second factor. So when in this concept, when something needed you to authenticate yourself, you'd go, oh, yes, here's my second factor. And so you'd use your smartphone to show it this bracelet that you're wearing, or whatever. Maybe you've got a particular wristwatch that you always have on your right or left wrist.

And so, again, you'd show your phone this wristwatch. And the idea is that, through some process of feature extraction, this optical second-factor technology would decide somehow that, okay, maybe it somehow turns it into a hash. But the problem is you're not going to be holding the image exactly the same distance the two times. I mean, we all know the image is not going to be the same. Focus is going to be coming in and out, as it tends to. There's going to be some image stabilization problem because you're just holding your hands up. It's not going to be the same size, the same orientation and so forth. So there are some big hurdles to overcome in turning this into something like a

hash. And you don't want it to just be a go-no-go, like is this the same thing I've seen before, because that wouldn't be secure.

But the biggest problem in my mind is replay attacks because nothing about this is replay attack-proof. That is, if something like a hash is being sent to, like as your second factor, then it's by definition not going to change. It wouldn't be known to an attacker initially. But if they captured it, then in the same way that a password is inherently static, the problem with a password is that it's inherently static. It's your password.

And so the problem people have is in repeating their password properly because it is a static thing. So if this is just a repeated hash of some feature extraction of an image, then that's just like a second password, which is better than only one password, but it's not nearly as robust, for example, as the existing infrastructure that we already have in place, which is a time-based one-time password, which is by its nature, and the reason it's time-based, is that it is replay-proof, the idea being that no server will accept the same token twice, and the token changes every 30 seconds. So that's much better than saying, oh, yes, here's something that I showed you before, and I'm showing it to you again; right? It's like, okay, but so was a password. And this just seems way fuzzier and less robust to me.

And the other - so a time-based one-time password is better because it's replay-proof. The problem it has, as we know, is that it does require that the server hold a secret, that is, there's a shared secret which does not go over the wire, but it is shared, and which is what allows the server to know the proper answer to the time-based one-time password. So your authenticator has the secret. The server has the secret. In fact, remember that the way these things work is the server gave you the secret, typically in the form of a QR code, which your authenticator then captured so that now you're sharing the secret.

So the problem there is that, if the server loses its secret, then that could compromise all of its customer base's one-time passwords - which, remember, is exactly what happened with the RSA breach back in 2011, where all of their RSA SecurID product line, which are six-digit time-based passwords, was compromised because that was the huge concern was, if their secret key database escaped, and in this case it did, that was bad news. So that's why the one step further is what the state-of-the-art authenticators use, like my own SQRL system, where we use a public key challenge/response system, where the server has the public key, not a shared private key.

And then the way the system works, the way you get replay protection is that the server generates a nonce, a unique challenge which it sends to the authenticator, which the authenticator signs using its private key; and then it returns the nonce and the signature and the server verifies, yes, that's the nonce I sent. And, oh, look, it was signed properly, which it's able to verify with the public key. That's the next generation which we are just in the process of beginning to move to. So that sort of creates a hierarchy of authentication quality. And unfortunately, interesting as the idea of showing your phone something, it doesn't have any of these characteristics that are replay-proof and ultimately proof against server-side compromise, which we really do want in this day and age.

Leo: On we go. Do you want to do the picture now?

Steve: Yeah, absolutely.

Leo: It's a very good one. Let me scroll back up here a little bit.

Steve: Yeah. I'm tempted to wonder whether it's true or a setup. But...

Leo: Yeah, it could be a setup; right?

Steve: ...it is fun. And, yes.

Leo: I think it's a little old, too, just judging by the reaction from the chatroom. I think a few of them have seen this.

Steve: Ah, okay, interesting. So, well, we know that, at least based on the photography, it involves an iPhone, so it's not - so what's that, it's been 10 years now?

Leo: No more than 10 years, yeah.

Steve: Yeah. So anyway, it shows a car's stereo system, and the iPhone pushed into the docking slot of the stereo system, with the caption: "My brother was upset because his car's 'docking station' for his iPhone wasn't working, and it was scratching his screen." And those old-timers among us will remember the days when we had cassette players. And it turns out that the iPhone is pretty much the same profile as the side of a cassette, yes. So not the eight-track big crazy cartridges, but the cassette tapes. And if we are to believe this, somebody thought, oh, look, a docking station for my iPhone.

Leo: Fits perfectly. Fits perfectly.

Steve: Yeah, that's right.

Leo: You might wonder why [crosstalk].

Steve: I'm not sure what happens if you hit fast-forward on the iPhone. Anyway, fun picture. So our friend Felix Krause, who has been sort of working at improving iOS security by poking at it, as researchers do, noted some concerns over the way iOS handles the camera. And our takeaway from this is it's worthwhile periodically doing a little privacy audit of your iPhone or iOS device - meaning I guess iPods and maybe, well, iPads - privacy settings. Under Privacy is Camera. And you're able to see which applications, over time, you have ever given permission to use the camera. The reason an occasional audit is useful is that those things never turn themselves off, so they tend to accrue over time. And you might find applications that you kind of wonder, wait a minute, why does that have access to my camera?

So the point is what Felix noted is that once any app has ever been granted access to your iOS device's cameras, it can access both the front and the back camera, record

what it sees anytime that app is in the foreground, take pictures and videos without notifying you in any way, do anything it wants to do with that content, upload the pictures and videos it takes immediately, perform face recognition or detect facial features and expressions. iOS 11 now builds some feature extraction, facial feature extraction into the underlying framework. And all of that without indicating in any way that your phone is recording you taking pictures, taking videos of you and your surroundings. No LEDs, no lights are shown, no indication of any kind. Which is not to say that apps are doing that, or that they would be doing so maliciously. And they can only do so when they have the foreground, when you're using the app.

But if you just - I think it's probably worth looking through, doing a little audit under Privacy > Camera, under your iOS device settings, just to see whether you still think it's a good idea for all of the apps that currently have that capability to keep them. And of course there are switches there. You can just flip them off, so to speak, if you decide that you don't want all of the apps that have access to your camera to retain that right because, as Felix notes, you are relying on the proper behavior of the application to which you have given - which if you ever gave it access, for example, sometimes an app may just want to, like during setup, acquire access to the camera to take a picture of you in order to establish an icon or an avatar for you. But then there's really no need for it to have enduring access, but by default it keeps it unless you say you don't need that anymore. So again, probably worth just keeping that in mind, just from a privacy standpoint.

Leo: You could do a little audit in iOS. You can go through. There's a permission section, and I do this periodically, go through all the permissions and turn off ones you don't think are necessary because, you're right, could just be let's - Snapchat does that. Let's just get a picture, well, I guess Snapchat continues to need access. But there are other apps that do that. And let's just get a picture of you or, hey, let me scan the QR code, that kind of thing. Yeah, don't, yeah - revoke it.

Steve: And I actually feel that way also about location services.

Leo: Same thing, yeah, yeah.

Steve: Yes. And as we know, the location system in our iDevices uses a lot of power. It's a big battery drain. And if you look through location services, it's like, what the heck? I mean, I want Google Maps, and I want Maps, and that's about it. But you'd be surprised how much crap in there wants to know where you are. And it's like, it's none of your business where I am. So location services is another very worthwhile thing to turn off. And you can say - there's never, always, and only when I'm using.

And so, for example, I have everything set for never except a couple of mapping apps where I have them only when I'm using them. And what you'll find is you can recover an awful lot of battery life just by keeping things from just being able to monitor where you are all the time. That's just - it's amazing. And of course Apple's always trying to turn that back on. And when you turn it off it's like, oh, my god, oh, oh. Not everything is going to be able to know where you are. It's like, yeah, it's none of their business. So anyway, another thing worth auditing, I think.

One of the problems we have, as we know, with the 'Net is bots and botnets. And on one hand it is super powerful to have them because, for example, Google revolutionized

finding stuff on the Internet by turning bots loose, spiders, to crawl around the Internet and find everything and then index it. So that's super useful. But, unfortunately, with all the good comes the bad. And we know that there are bots trying to get up to all kinds of mischief, creating accounts on sites. For example, we've talked about how bots have abused Ticketmaster, and so Ticketmaster has had to come up with ways to prevent bots from buying tickets and then reselling them because that isn't what they intend. They want humans to do that, not automation. And then there's bots that scrape sites and copy their content. I know that Craig's List has anti-bot scraping technology because, again, they want people to be viewing these pages, not automation to be abusing their intent.

So as we know, CAPTCHAs are the result of our attempt to determine what is a human and what is a bot. And the CAPTCHA itself is an abbreviation for Completely Automated Public Turing test to tell Computers and Humans Apart, CAPTCHA. And we've talked about CAPTCHAs and their evolution for years and how, for example, now Google is using much more of the intelligence they have about the history of you and your computer and your IP and the Google cookie that you're carrying so that you don't even have to solve a test anymore. You just have to click on the "Yes, I'm Not a Robot," and they think about that for a minute or two, and they say, okay, yeah, we agree you're not. But when they're not sure, then they will still give you some problem to solve, some text that you have to recognize.

If you have a problem with eyesight, then there's an audio option that you're able to use as part of Google's so-called "reCAPTCHA" system. Well, just in the last week some research has come to light where two different groups have managed to, with a high degree of success, spoof the state-of-the-art reCAPTCHA system.

One group has named their solution unCAPTCHA, and they've taken six different speech recognition systems - the Bing Speech Recognition, IBM's, Google Cloud's, the Google Speech Recognition which is distinct from Google Cloud, the Sphinx system, and the Wit-AI. They feed the audio puzzle from the audio flavor of Google's reCAPTCHA through all six of those and then aggregate the results and feed the most likely consensus answer back to Google with 85% success and far higher speed. In tests that they performed, their experimental system was able to break 450 reCAPTCHA challenges, with 85.15% accuracy, in a little less than 5.5 seconds, 5.42 seconds. Which is less time than one person could listen to and respond to a single reCAPTCHA audio challenge. They did it 450 times with 85% accuracy. So that's the audio side.

There's a different group of researchers who tackled the same visual CAPTCHA challenge. They don't have quite the level of success, but they managed to develop an AI vision bot that is able to break various CAPTCHA systems with varying degrees. Google's reCAPTCHAs where you say, okay, you're not a bot, Google's not sure, so then they give you the visual challenge? Their AI vision bot is now up to two thirds, 66.6% accuracy, it's able to solve the optical CAPTCHA. There's another type of CAPTCHA known as BotDetect which they can do with 64.4. Yahoo's seems to be tougher. That's at 57.4. And PayPal is about the same. Their image challenge they're able to solve at 57.1.

So what we're seeing, and this is I guess foreseeable, is that bots happened. We came up with a solution instead of puzzles that would stump the bots. That worked for a while. Then researchers said, okay, let's come up with more human solutions using existing technology and AI in order to push this bar further. And they have, to the point where unfortunately it looks a little bit like we're beginning to lose that battle for this kind of puzzle. Yet I really do like what Google has done where in the general case they're not presenting any puzzles. They're just saying, based on our aggregate analysis of your past behavior, we already know you're not a bot. So just click the "I'm not," and we'll move

on.

Leo: Although relevant to your earlier discussion, if they're not sure, they'll give you pictures.

Steve: Right.

Leo: Right? I always get the storefront. Is this a storefront? Click the ones that are storefronts. I guess that's hard for a computer.

Steve: Oh, that's interesting, yeah. I haven't seen that one.

Leo: You don't get that one?

Steve: No, well, for whatever reason Google's happy with me.

Leo: It always knows it's you, huh?

Steve: I don't move around much. Okay, now, Leo, look at this next picture. You're going to want to put this on the screen, too. This is just horrifying. This is an actual web page from a Portuguese ISP where Net Neutrality doesn't exist, showing how their customers are able to purchase which types of bandwidth they want access to. Messaging, and we see the icon for Skype and FaceTime and iMessage and a few others. And that's for, looks like, what, 4,99 euros per month, I guess? So that's the Messaging package. Or then you can also get the social engineering - the social engineering - the Social package, where we have Facebook and Twitter and Pinterest and LinkedIn and a few others, also for 4,99 euros per month. Or the Video package, where you get access to YouTube and Netflix and a couple others; or the Music package; or the Email & Cloud.

And anyway, the point is that in this scenario, this has all been broken apart, and unfortunately it's reminiscent of what we're subjected to with cable TV providers these days, where you choose which things you want, not a la carte, but in lumps, and always end up with things that you're paying for apparently but not using. And anyway, I just thought this was sort of interesting. A bunch of our listeners sent this to me, and I thought, yeah, this is not the world we want to live in.

Leo: This is a wireless provider, I'm guessing, because of the 10GB cap.

Steve: Right.

Leo: That's weird.

Steve: Ugh, yeah. Yikes.

Leo: Well, here's the question, which is not clear. Do they block these services unless you pay for them? Or does it, I mean, I wonder what they're offering here? Maybe someone from Portugal could tell us.

Steve: Right, who's able to read this. I assume what this means is that you pay them if you want access to these services. And so, yes, if you don't buy those, you don't get them.

Leo: That's just terrible.

Steve: I know. I know.

Leo: That's just terrible. Or is this offering zero rating? Which it kind of feels like it might be? In other words, if you pay for the video package, then none of the bandwidth you use for Netflix or YouTube or Periscope or Twitch goes up against your bandwidth. That's kind of what I'm getting from my imperfect Portuguese, is it looks like this is saying, if you want this - you still get access. But if you don't want it to count against your cap, you pay for it.

Steve: And in that move above it says something "Pago Unlimited."

Leo: Yeah, yeah. That's my guess. But I don't [crosstalk].

Steve: Well, let's just hope this doesn't ever happen to us because I just want to pay Cox for my connection, and I am super happy with it. Just let me do whatever I want to with it.

Leo: Would it be okay with you if ISPs just had speed tiers? Or, conversely...

Steve: Well, they do now.

Leo: They do now.

Steve: They do now, yeah.

Leo: Or, conversely, if they had, like speed tiers, but bandwidth tiers? Like if you want - how much bandwidth do you want? What if you just - in fact, what if you just paid by the byte?

Steve: The way we buy electricity. The way we buy electricity.

Leo: Yeah. That would be okay, wouldn't it?

Steve: The more we use - I think it would because, well, but...

Leo: Because then everybody's equal.

Steve: Yes. And I tend to be a low bandwidth user. I'm experimenting with cord cutting now for the first time in my life. But until then, I was just using cable TV. And so...

Leo: But even if you use, I mean, that seems more fair. Like if I'm 100GB or a terabyte a month user, it would make sense for me to pay a little more. Maybe give us all high speed. Or I guess you could still tier the speeds. And the main point here is access is equal. So you're paying the same amount per byte for Netflix that you'd pay for YouTube, that you'd pay for Skype.

Steve: And really, why shouldn't somebody who is downloading massive...

Leo: Pay more.

Steve: ...tens of terabytes a month - they should pay more.

Leo: That's not against Net Neutrality because that is completely neutral.

Steve: Correct.

Leo: It seems fair in the sense that you're paying for what you use, just like electricity or water. It's more like a utility. By the way, somebody in the chatroom, Dave, says he's Portuguese, and that's exactly what this is. You pay for zero rating for these services. So you pay 4,99 euros a month to get...

Steve: And then you're not billed.

Leo: ...unlimited, you're never billed for video. So if you watch a lot of Netflix, pay an extra five euros a month, that seems not too horrible. But again, I prefer...

Steve: It does mean that, yes, that does mean that they are getting involved in what you do.

Leo: Right, the zero rating, which we don't like because I'm not on - TWiT's not on

that list. So you might be more likely to watch YouTube than TWiT because, hey, it doesn't cost me anything to use YouTube.

Steve: Ah, right.

Leo: It might cost me something on TWiT. That why it seems to me paying per byte is the way we should work this out.

Steve: Yeah. I completely agree. Absolutely. I think that's, I mean, it just seems fair. And it fits the - you know, you want models that make sense. And so if your provider is going to run bigger pipes and fiber and so forth in order to be able to deliver, to beef up their own infrastructure, then they want remuneration. And it should be the people who are requiring them to make that investment be the ones who pay for it.

Leo: Yeah. It would still impact TWiT, of course, because say we have very large files. There'd be an actual cost for each show that you watch.

Steve: Well, and I wouldn't be surprised also if there isn't a quantity discount. That is, the more you - so there is a tiering system...

Leo: As there should be, yeah.

Leo: ...where big users - yes. And there's that way with high power users, too. Like with big electric users.

Steve: Oh, man, we get right up in that top tier every month. That's why we ended up getting solar. It got to be very expensive. In California electricity is extremely expensive.

Leo: I know.

Steve: That's why, as I was saying, actually it was Mark Thompson who explained to me that you can't do crypto coin mining anymore because you just can't pay for the power. You're literally paying more than your miners are able to produce.

So speaking of mining, in a bizarre, but I guess not that unexpected attack, the Coinhive guys who are the premier web browser mining provider that we've been talking about the last few podcasts, they got their DNS hijacked.

Leo: Ooh.

Steve: Get a load of how. Three years ago, back in 2014, we covered, and there was, a

major data breach at Kickstarter, which exposed a large number of passwords. When Coinhive recently set up their hosting with Cloudflare - wait for it - they reused the same password which had been leaked by the Kickstarter breach three years ago.

Leo: Well, that's why monkey123 never was a good idea.

Steve: It's never a good idea. And I don't even want to know what the password is. And they neglected to set up any other of the available second-factor authentication which would have prevented this problem. This allowed attackers to get into their Cloudflare account almost with their permission because it was like, hey, here's the password that's been floating around the Internet. What the heck? Commandeered their DNS to point the Coinhive.com domain to their own servers which effectively commandeered the entire Coinhive cryptocurrency mining operation in one fell swoop. So that suddenly all the websites that were hosting Coinhive's mining JavaScript were mining on behalf of the attackers and not on behalf of Coinhive or the sites which were hosting the script because Coinhive pays the sites for the success of their visitors' mining operations, the number of hashes per second that their visitors are performing while they're there.

And Leo, I thought you'd get a kick out of this. I happened in my research to run across the script which anyone who wants to can put in their web pages to invoke Coinhive. And it is simple as <script src="https://coinhive.com/lib/ coinhive.min.js"></script>. That's it. That invokes...

Leo: Of course you're loading JavaScript from Coinhive.

Steve: Yes, exactly. And so that's why, when the bad guys redirected Coinhive.com to their own servers, you were now loading their script rather than the script from Coinhive, which of course ran scripting, but in this case it accrued to the attackers' benefit. But that's just how easy it is. It takes a couple lines of JavaScript stuck in the browser headers in order to invoke mining. Although it's worth noting that we're seeing a strong backlash from the providers of adblocking, and even the AV folks are getting involved and stripping this stuff out of pages on behalf of their users who are saying, no, I don't want you monkeying around with my web page.

But as we've said here, I think that's a little bit misdirected, unfortunately. As we know, it doesn't represent a security breach. What it mostly means is that this is overt scripting. And whereas most of the scripting being done is not visible, if something pins your CPU, whoops, that kind of makes it visible. But there's a lot of scripting going on that is much worse than this, which is relatively benign, which is being used to track and aggregate and fingerprint and do all this other stuff, which people are just saying, oh, yeah, well, I don't care about that. They're saying, oh, yeah, but I don't want anybody mining cryptocurrency on my browser. Well, okay. Really not such a security problem.

Okay. Bad Rabbit is new ransomware which is rapidly spreading across Eastern Europe, a lot in the Ukraine and in Russia. Like WannaCry that we were talking about before, and several other recent families of malware, Bad Rabbit is leveraging one of the escaped NSA technologies, in this case EternalRomance, which was as we know released by the Shadow Brokers and was subsequently patched by Microsoft last March. But as is unfortunately the case, as we know, not all machines are getting patched. And in fact it's a little bit surprising to see how rampant this is.

EternalRomance was the remote code execution exploit that took advantage of a flaw, also another flaw in Microsoft's Windows Server Message Block, the SMB protocol, which allows, once it gets into a network, typically through a drive-by download, what the AV people are finding is that malicious Flash downloads - I'm just amazed this is still getting as much traction as it is. I mean, Flash is pretty much gone. But there are sites that say, oh, you need to update your Adobe Flash. And people say, oh, okay, and click on it. And what they're doing is they're installing a copy of Bad Rabbit, which then gets into their system. It is cryptomalware, which is requiring - remember the quaint days, Leo, when it was one bitcoin, and that was about $400?

Leo: Yeah.

Steve: Well, no. Now it's 0.05 bitcoin, which is about $285.

Leo: Well, that's a deal.

Steve: In current bitcoin...

Leo: A fraction.

Steve: Yes, valuation. So once it gets onto a single machine, then it uses the EternalRomance vulnerability to scan within the network and find other vulnerable machines and leap through an Intranet in order to infect all the machines within the network. So this brings us back to our standard advice, which is, number one, NEVER, in all caps, NEVER download anything offered to you by any website. And remember, even trusted sites may have been and often are transiently compromised. So it doesn't matter who, that is, what site is making the offer for something your system "needs," in quotes.

One of the other things we're seeing is sites saying, oh, you don't have the XYZ font. You need that font in order to properly, you know, click here to download that font into your system. No. They're offering you something. Never download anything offered to you by any website. And we learned this lesson back in the early email virus days. It didn't even matter if the email came from Mom because we knew that Mom may have gotten her computer infected, and the virus was digging around in Mom's address book and sending out replicating virus email to everyone Mom knows, including all of her kids.

So, similarly, you trust Mom, but you can't trust email that apparently your mom sent you. Similarly, you can't trust anything that any website tells you you need to download. Just plant that one as a big flashing red neon sign. Never download anything offered to you by any website. That is now the leading attack vector. It can be an ad. It can be the site itself. Just never. And then, I mean, that's advice you can tell everybody you know, and should tell everybody you know because it requires no technology. It requires self-control, but no technology. Then we know, people who listen to this podcast, that we want our networks behind NAT routing firewalls, which are NAT routers which are inherently firewalls, with Universal Plug and Play disabled wherever possible. That is, if it doesn't break something, turn it off, which is generally good security advice. And also, if possible, we want to establish and maintain network isolation.

So if you can have, like, many routers are now having guest networks. So you want, for

example, your IoT devices and visitors to be on the guest network and keep isolation with your more valuable computers on a separate network, to do that as much as possible. And, finally, keep up to date with all patching, not only your systems, which now patch themselves, but the IoT devices that aren't yet up to speed, that aren't yet patching themselves as really all future IoT devices are going to have to start doing. We really - that just has to be a requirement for next-generation IoT is that they are able to preemptively update themselves, and they're being offered by suppliers, by manufacturers who are responsible in after-purchase maintenance of these things because they're all computers.

There's something that I wanted to put on everybody's radar that will be appearing in the Windows 10 Fall Creators Update. And it's - I didn't write the version down. It's something 1702?

Leo: Nine, 1709.

Steve: Nine, okay, good. Ed Bott made a note of it, and I wanted just to bring it to everyone's attention because it's a very nice feature which is not currently enabled by default. And it has the feeling of something that Microsoft will eventually turn on once they have a sense of what it breaks and what they have to do. It's known as Controlled Folder Access. And it's a feature of Windows Defender, so Windows Defender has to be turned on. You have to be using it, and so it cannot currently be used with third-party AV. It's got to be Microsoft's built-in Windows Defender. Under Virus and Threat Protection, it's called Controlled Folder Access. Off normally, you want to turn it on. But do so - I wouldn't tell all your friends yet to turn it on. See how it works for you because the goal, Microsoft's goal is that it adds powerful and apparently effective anti-cryptomalware features.

What they're trying, what Microsoft is trying to do is to identify software which should not be reaching into what then become protected folders, which are the stuff that cryptomalware wants to encrypt because it's the unique stuff, your own documents and pictures and music and things that you may need to get back. So there are, after you turn it on, there are two other options below there. One is Protected Folders, which displays the list of folders whose contents are then being protected from tampering by malicious or suspicious apps. The default list includes data folders from the current user's profile and from the public profile, so that you're able to do some curation there, if you want to.

And then the second thing is which apps are allowed. So there's an "Allow an app through Controlled Folder Access." You may find that something that you know and that you trust, but that Windows Defender doesn't yet, is having a problem. So you can go in there in order to whitelist things. So this is, I mean, I can understand this not being turned on because it could upset things. I have a picture on the next page of the show notes of this page from the dialog from Windows 10 showing this thing switched on and those two other items that you're able to click in order to open and explore.

So I just wanted to make sure people knew it was there. Seems like a good thing. It'll be interesting to get some feedback over time. And Microsoft tends to do this, historically. For example, Windows XP was the first version of Windows to have an actual firewall built in, but it was turned off until, what was it, Service Pack 2 where they finally turned it on. So they generally roll these things out kind of gently to make sure they don't break something, let experts turn it on at their own risk, and then they get some feedback on how it's doing and allow it to mature a little bit. So a nice feature. And it would be nice.

Oh, and I did also see in my digging into this a little bit, several AV systems were being blocked as they should have been. I saw some reporting saying, whoa, well, this thing did stop us, and we're glad because our tests of ourselves demonstrated that it was catching unknown software trying to make modifications. So that's a good sign.

Leo: I like this Controlled Folder Access. It just underscores more and more why you don't need an antivirus because, well, you're breaking the antivirus; right?

Steve: Yes.

Leo: So, yeah, why should I let a third party with in some cases dubious security themselves access to my system?

Steve: Yes. I completely agree. I think we really have moved, I mean, again, another example of Microsoft sort of bringing out a feature and moving it slowly forward. What has it been, a decade since we're beginning to get - we have the little house down in our tray, and it's downloading updates. I mean, the writing has been on the wall. AV companies, you don't have an infinite opportunity here. It just doesn't make any sense at this point. And you and I have been singing this tune for quite a while now.

Leo: Oh, yeah. Oh, yeah.

Steve: So Check Point was the Israeli company that I couldn't remember the name of at the top of the show. They found a worrisome vulnerability in a widely deployed LG SmartThinQs, smart home devices. LG calls it the SmartThinQ, S-M-A-R-T-T-H-I-N-Q, smart home devices. Unfortunately, there was a design problem which Check Point found. Had it not been fixed before disclosure - and they worked with LG. LG, Check Point reported, was very responsible and immediately fixed this and has a system which was able to patch in place, which is what we want.

So after the fact, after this was fixed, and the fixes were pushed out, Check Point disclosed what they had found. And essentially what they found was what they called HomeHack, as they named their vulnerability, gave the attacker the potential to spy on users' home activities, among other things, via the Home-Bot robot vacuum cleaner, which has a video camera. And there are, as of the end of 2016, so about a year and a half ago almost, more than 400,000 of these camera-equipped Hom-Bot robot vacuum cleaners. And essentially an attacker located anywhere in the world could get into someone's home remotely, commandeer this vacuum cleaner, and turn on the camera and drive it around the house in order just to, literally, you would have a mobile camera wandering around.

And you kind of look at it thinking, well, it doesn't look like it's following its normal vacuuming path. No. Some bad guy is steering it, going wherever they want to go to see what they want to see. So anyway, sort of, I mean, but this was a true, a real-life example of an in-place active vulnerability that probably by now easily more than half a million homes had before Check Point found it, responsibly disclosed it, LG fixed it, and thank goodness they were able just to push a fix out to their entire install base of these connected devices and close this problem. This is a perfect example of why it's crucial

that IoT devices moving forward be able to manage their own security. It just has to happen.

And so Leo, get a load of this one. As I was doing the reporting on this, I'm thinking, okay, how long, count the seconds, before you have this ordered up? Because this is a new service.

Leo: Not at 250 bucks. No way.

Steve: Oh, okay. It is a little pricey.

Leo: Yeah. And, no, I can't get it here, either.

Steve: Oh, okay. So it's not available. Is it regionally exclusive?

Leo: Yeah, well, he's talking about the Amazon Key service, which lets the delivery person in. The reason it's only available in 37 markets, it has to be a market where Amazon deliveries occur from Amazon's own Amazon Logistics truck. They don't let UPS, FedEx, or the U.S. Postal Service do it.

Steve: Oh, that was a big question I had.

Leo: Yeah.

Steve: Okay.

Leo: And we're not in one of those areas, that's all.

Steve: I am, but I'm not letting them in. Unh-unh. No.

Leo: Why did you think I'd let them in?

Steve: Well, because, Leo, well, you know, you have...

Leo: You're home all day. This is why people would do it, if stuff's getting stolen from their porch.

Steve: Yes, yes, yes. And so having an Amazon bonded delivery person - okay. So let's explain what this is. New service from Amazon called Amazon Key. First of all, it relies on Amazon's new Cloud Cam and a compatible smart lock. The camera functions as the hub. It's connected to the Internet via of course the residential WiFi. It's positioned to show

the front door and entryway from inside the house. So it's showing the, what do you call it, the region inside the front door.

Leo: The foyer.

Steve: The foyer, thank you. The camera talks to the front door locks, either manufactured by Yale or Kwikset, over the ZigBee wireless protocol. So when a courier, when an Amazon courier arrives with a package for in-home or at-home, well, actually in this case in-home delivery, they scan the package's barcode, which sends the request to the Amazon cloud. If everything checks out, you've signed up for the service at this location and so forth, the cloud grants permission by sending a message back to the camera, which then starts recording the video. Once the video is started, the courier gets a prompt on their app and swipes the screen. The home's front door unlocks. The courier opens the door, puts the package in the foyer, closes the door, does something else on their scanner which causes the front door to lock.

The customer receives a notification that their delivery has been made, along with a short video showing the whole drop-off process from before the door was unlocked to after the door was relocked, in order to confirm that it was done. And of course the Amazon delivery person realizes that they're on candid camera so they'd better behave themselves. So it's as good as system as it could be where you want autonomous delivery and security, and everybody, all of the various components to be able to be held responsible. And as you said, Leo, if you're in an area where there's a high likelihood that the packages could be or have been or would be stolen from your front porch, then I can see this could make sense.

Leo: It's a big problem. I mean, even here in Petaluma it's a big problem.

Steve: No kidding.

Leo: Yeah.

Steve: Wow.

Leo: Because they say Amazon. They have a big smile. They're big.

Steve: It's going to be something good.

Leo: It's something good in there.

Steve: It's going to be a goodie, yeah. Interesting. So I just wanted to revisit the question, still unanswered: Who is Satoshi Nakamoto? As we know, we now believe this is a pseudonym for the mysterious and still anonymous and unknown original creator of the Bitcoin, the world's first cryptocurrency that we did a podcast on years ago [SN-287] because the technology was so cool, and I was just enraptured by it, whoever this was,

or whatever group did this. We've had, as we know, some false alarms about who this person is. I just wanted to note that, at this time, since this unknown person or persons is believed to have about 5% of the current bitcoins that have...

Leo: How much would that be worth?

Steve: $6 billion.

Leo: Holy cow.

Steve: Yes, my friend.

Leo: This is why I've always thought bitcoin and other cryptocurrencies are essentially a pyramid scheme because one of the features of a pyramid scheme is the person who starts the scheme makes the most money.

Steve: Yeah.

Leo: And the people who enter late, like if you bought bitcoin today, make the least money, or in fact no money, and end up subsidizing the guy who invented it. So good going, Satoshi.

Steve: That was a good algorithm. You got $6 billion.

Leo: But he can't cash it in without revealing who he is; right?

Steve: There are, I think there are ways. I mean, you could certainly buy stuff. You could transfer coins.

Leo: Coinbase is not going to give you $6 billion.

Steve: No. Coinbase doesn't have $6 billion.

Leo: Who could you cash that in with?

Steve: Yeah.

Leo: You could buy a couple of pizzas, maybe. I don't know.

Steve: Very expensive pizzas. All the toppings you could ever have.

Leo: I mean, seriously, it's really interesting.

Steve: Yeah.

Leo: That's, of course, how we have always known somebody could identify themselves as Satoshi is if they said, well, here's the key.

Steve: Yes, yes. If you demonstrated ownership of those first early coins, then it would be like, oh, there's only one person who has that, or one group, or one entity. So, yeah.

Leo: And either he's fabulously wealthy and really privacy focused, or dead, which some people think.

Steve: Oh, that's interesting. Sad. But, boy.

Leo: That six billion could be lost forever; right?

Steve: Well conceived, yeah.

Leo: There's no way to recover that if you don't have the key.

Steve: No, no. It's like my 50 that are wandering around here somewhere.

Leo: And my seven. Because I can't - I put the wallet on Dropbox or somewhere, and I keep loading it in different things, and it always says zero. I don't know what I've done wrong.

Steve: So I've talked about TunnelBear a couple times, and they continue to impress me. Their most recent blog post I just wanted to share, just to give a sense for this is the way you want your VPN. This is a philosophy you want them to be bragging about. Their most recent blog posting was how we avoid collecting your IP address on our website.

And just the first two short paragraphs read: "It might sound odd, but we pride ourselves on knowing very few things about our customers. In fact, the less we know about you, the happier we are. As we grow and learn more about the information we need to optimize our service, we find new ways to do things like anonymize site statistics in a privacy-compliant way. You see, we want to make sure that, even before you decide you need a Bear in your life, we're helping you maintain your privacy. Here are a few of the things we do on our website to limit the collection of your IP address."

And I'll stop reading at this point, but I dug into it, and I was impressed. They go on to

describe a feature that I didn't realize Google Analytics offered, which is an aid to help anonymize visitors who are otherwise being tracked by Google Analytics, where the last byte of the four-byte 32-bit visitor's IP address is deliberately recorded as zero. So every IP address ends in dot zero rather than what it really is, regardless of its actual IP.

So as we know, in practice this preserves the geographic data that might be useful and valuable, but hides an individual ISP's customer among or amid up to 255 others since that last byte could be any value from zero to 255, so any of a total of 256 possible values. But by zeroing it, you're never recording which one of you on that, what is it, 16 million different subnets. So you're one of 256 of that three-byte subnet. There is no recording of who you are.

So again, there have been - we've talked about TunnelBear a couple times just because a lot of our listeners have asked what do I think. I've checked them out and looked over their protocol and everything that they've published, and it looks like they're doing everything right. And, boy, I've got to say I sure like their attitude. It's the attitude that you want from someone whose services you are purchasing for privacy. It's like, yes, that's what we want, too.

A bit of errata from a Khyron, K-H-Y-R-O-N, who tweeted, he said: "Really enjoying Security Now!. Last week you mentioned bitcoin mining via JS." He said: "Most JS miners are using Monero, not bitcoin." And I keep making that mistake. So thank you for correcting me. Yes, I just - in my head I have it incorrectly. I should be saying cryptocurrency mining or Monero, not bitcoin. So thank you. For the record, that's what I meant. And Leo...

Leo: They're mining for bitcoin. They're using Monero to mine for bitcoin?

Steve: No, Monero is its own cryptocurrency; right.

Leo: How confusing is that.

Steve: Yeah, I know. And there are a bunch of them, as we know, that have sprung up. But this is the one that is being mined by the JavaScript miners.

Leo: So the JavaScript miners don't mine bitcoin. Oh, interesting.

Steve: Correct. And so thank you, Khyron, for correcting that. And Leo, are you guys finished with Season 2?

Leo: Not finished. We're about four episodes in. But I'm loving it.

Steve: I am, too.

Leo: Isn't it good? We're talking about "Stranger Things."

Steve: It is good. We are talking about the second season of "Stranger Things." There's one dud episode. You haven't encountered it yet if you're four in. But other than that, I thought it was really good. And I have a girlfriend who likes to binge on "Stranger Things."

Leo: I would marry that woman.

Steve: Quite happy.

Leo: Nothing like Netflix and chill, especially if it's "Stranger Things."

Steve: I couldn't believe it. She kept saying, "Oh, let's do one more, one more." I said, "You're kidding; really? Okay."

Leo: A woman after my own heart. Lisa's like that, too. We'll end up, you know, I think Saturday night we ended up in bed at, like, 1:00.

Steve: Yup, I know it. I know what you mean.

Leo: Very good stuff. Very good.

Steve: So I got a nice note from Rusty Burke, who actually had a question for us about Security Now!. But en route to that he said: "Before I get to my question, which is not actually sales related" - because he actually sent email through our sales email, which Sue forwarded to me. He said: "I just want to express my thanks to SpinRite."

He said: "It recovered a Sony Vaio laptop which was, with increasing frequency, inexplicably dying on me. So I ran SpinRite at Level 4; and, after rebooting, I haven't had a lick of trouble. Runs like new," he wrote. So Rusty, thank you. And I just wanted to remind people that it's not always just a crash that can demonstrate that the hard drive is really beyond trouble at that point. It's if something, you know, if it starts being flaky, if it starts going slower, that's a big indication. But also just hanging and flaky and stuff, just run SpinRite over it, and you may find those problems disappear. We know that lots of people have.

And then he asked: "My question is whether Steve ever did a series on networking, which he suggested might be coming while doing the How Computers Work series on Security Now!." He said: "I've combed through several sources for an answer to this question, but can't seem to find one. Thanks for a great product and great information. Rusty Burke."

And you know, Leo, I know that we spent a lot of time in the early days really nailing down how the Internet works, going from packets to routing and autonomous routing and then building, talking about the UDP and TCP and the various protocols on top of those. So I know that's all out there in our past.

Leo: The problem is you can't search for "networking" because every show has the word "network" in it, and so that's not going to...

Steve: I will try to find some time to go back through and just scan the topics and pull together the list of podcasts. I kind of think we did them in clumps, where we got onto a...

Leo: I feel like we did, too, yeah.

Steve: We got onto a routine where it was, okay, this week, you know, last week we talked about this aspect, this week we're going to talk about this aspect, and next week we'll talk about that one. But anyway, I'll see if I can find some numbers.

Leo: Everything's here at TWiT.tv/securitynow.

Steve: It's all there.

Leo: Or on your site, too.

Steve: Yeah.

Leo: But the problem is to look at all the episodes, I just clicked that button, wait for a while because it's going to take a while. And then I think we still paginate it because there's, well, 600-some episodes. So, yeah, there's 27 pages. Although you could start at 27 and go backwards. Because it would have been early days, I would think.

Steve: It would have been the early days.

Leo: Yeah, yeah.

Steve: Yes, those quaint early days when we were actually able to talk about something other than what happened this week.

Leo: Unfortunately, the earliest shows really don't have very good show notes.

Steve: That's true. Yeah, well, in fact I wasn't doing the show notes. They have transcripts, but not the show notes.

Leo: Yeah. Yeah, I don't think I - oh, here. Well, here Steve explains how VPNs can protect you. He also announces the WPA password generator. So, yeah, I guess you could go one by one. VPNs Part 2, so that's kind of a networking subject; right?

Steve: Yeah.

Leo: So I don't know. Yeah, you can go through them one by one. PPTP and IPSEC. The early ones really did have a lot of...

Steve: Yes, of networking stuff.

Leo: ...information about how networks work. OpenVPN, I remember you did a big, big thing on OpenVPN. So, yeah, go through it. You'll find quite a bit. It's all there somewhere.

Steve: So in closing the loop, Steven, whose Twitter handle is @Ozzyla, he said: "@SGgrc Just heard on your podcast about running SpinRite on an SD card. How the hell do you do that? Tutorial anywhere?" And actually it doesn't need a tutorial. I'll just explain that you need an SD card to be seen by the BIOS for Version 6 of SpinRite. At some point release beyond 6, I don't think it'll be 6.1, actually I'm pretty sure it won't be 6.1 or 6.2. But when I get to the native USB hardware support, that'll change.

But for now, if you use any adapter to allow an SD card to be connected to your computer, some PCs have an SD card slot, or laptops do, or you can use an SD-to-USB adapter. Plug it in when you power up the machine, or when you reboot the machine. That will allow the BIOS to see the card at boot time, and it will assign it a drive designation, a BIOS drive designation.

Then, when you subsequently run SpinRite, it will see the drive. And you just can run - remember you want to use Level 2, not Level 4, because you want to just do a read test on an SD card, especially on an SD card because they are the least robust. You know, SSDs are far more - they've got much more overcommitted and available spare space for dealing with troubles, and much more sophisticated hardware controllers on SSDs than SDs. So just a Level 2, but it'll smooth the card out and probably fix whatever might be going on wrong with it.

Leo: I don't know if this is what our correspondent was looking for, but Episode 25, "How the Internet Works, Part 1."

Steve: Ooh, that looks like a good place to start, yeah.

Leo: Be a good place to start. And you can hear what Steve and I sounded like back in 2006.

Steve: Back when we were youthful.

Leo: In our youth. Yes.

Steve: Robert D. Wilson asks - this is off-topic, but sort of ties into the show because everyone knows I'm a sci-fi person. He just asked: "Is 'Dark Matter' on Netflix any good?" And I would have to say yes. It's not top-of-the-line amazing fabulous Peter Hamilton sci-fi or anything. It's not as good as "The Expanse," which is amazing. But it's one of those made for the Syfy channel, and it's compelling. The characterization is good. It's kind of like a lot of it happens inside of a starship in dark setting. But I liked it. I've got a bunch saved up that I haven't gotten to. I just don't have time. There's just too much stuff to watch. But I would say you'll know in a couple episodes if it's for you or not. So I wouldn't universally recommend it. But it's above the normal, you know, it blows "Sharknado" away completely.

Leo: That's a pretty low bar.

Steve: That's a very low bar, yeah. I liked it. So I would say it's worth considering. I wouldn't subscribe to Netflix for it. But if you've got Netflix, check it out, "Dark Matter." I think you'll find - and it's like in its third or fourth season now. So there's enough content to keep you happy if you end up liking it a lot.

Leo: And we like "Black Mirror." I'm sure you've seen that.

Steve: Yes, "Black Mirror," yes.

Leo: It's on Netflix. But that's kind of not science-y sci-fi. It's more like dystopian future sci-fi.

Steve: And there's also - Amazon has one I haven't gotten to, based on Philip K. Dick short stories.

Leo: Oh, "Man in the High Castle."

Steve: There's that one. No, I'm thinking of - it's also an anthology series.

Leo: I don't know that. I love Philip K. Dick. But I've, with the exception of "Blade Runner," not been thrilled by a lot of the, you know.

Steve: Yeah. And it's Philip K. Dick's "Electric Dreams" on Amazon Prime, "Electric Dreams."

Leo: Of course that was the short story that "Blade Runner" is based on.

Steve: Yes, exactly. And "Blade Runner 2" was good, too. I liked it. It's a little more cerebral. I don't think it did that well in the box office, but I thought it was very good.

Leo: I'm waiting to see...

Steve: What is it, "Blade Runner 2049," was that the...

Leo: Yeah. It was too dark. I couldn't really - don't see it in 3D, if you can avoid it. It's way too dark.

Steve: Yeah, I don't. I didn't.

Leo: I'm going to watch it on my big screen at home when it comes out.

Steve: So two people. Jan Rademan asked: "Is there a test to see if a router is a Reaper zombie? Could ShieldsUP! be used to test for specific port?" And Dr. Brian of London asked: "Is there a way to see if my own ISP-provided D-Link router is part of the new mega botnet?" The answer is probably yes, if you were to get into a command prompt for the flavor of Linux and enumerate the processes, and really revved up your propeller hat and figured out what everything was. We'll be talking about it here in a second in more detail. But the best solution, first of all, everybody should know it does not survive a reboot. It does not make any permanent changes to the file system.

So one of the problems Reaper is having is that any rebooting of the IoT devices flushes it out of RAM. It is only RAM-resident. So rebooting your router guarantees you that it then is gone, and what you really want to do then is just make sure that you're running the latest firmware for your router. That's really - that's the best advice, the best generic advice is that the router manufacturers are definitely responding.

But remember that most of these existing exploits, there are a couple new ones that Reaper - Reaper is continuing to evolve, which we'll be talking about here in a second. But most of these exploits that it's using have been patched, in some cases as many as four years ago. So it's just that the routers aren't being kept up to date. So if yours is, and there's always a reboot following a firmware update, so just update your firmware to the latest, reboot your router, and you'll know that Reaper is no longer sitting around with the ability to look inside your network, in addition to, well, actually not doing anything, which we'll be talking about in a second.

WarrenKC asked @GibsonResearch: "In your opinion, should a user disable Windows Defender on Windows 10 if said user is careful on the Internet? Thank you." And I'll just say, well, I'm very careful, and I'm happy to have it running. Like you, Leo, I have never used in modern times any third-party AV. I just - I'm careful. In the case of Windows Defender, which is built into the OS, supported by Microsoft, being updated constantly, sort of has the sanction of the OS in terms of its ability to see what it needs to see, rather than needing to be reverse-engineered into the kernel as now all other AVs have to. I would recommend turning Windows Defender on and using it proudly. I do in my various Windows 10 machines. So I would say use it, and it doesn't get in the way because it's part of the OS.

And then, finally, two questions regarding DNSSEC and certs. Tim Chase says: "On SN-632 on DNSSEC said cert info could be kept in DNS. This would only be DV certs, not EV; right?" And then also related to that, Grant Taylor asked, actually in a bit of a Twitter dialogue, he said: "Why couldn't an EV cert be stored in DNS via TLSA" - which is the protocol DANE, the system for delivering certificates over DNS? He says: "Or are you implying DANE only implies DV?"

So there's some interesting issues, DV versus EV. EV, Extended Validation, really only makes sense in the context of a certificate authority and their assertion. So the whole point of Extended Validation is that you are not serving the certificate yourself over your own DNS; and, by having that channel be secured, thus assuring that the browser obtains the certificate you want it to, rather with non-DNS in the classic hierarchical PKI model, the certificate authority is using their signing of the certificate to provide an assertion of its authenticity. In the case of a DV cert, they're simply saying yes, you have proven ownership of this domain. That's all we're saying.

In the case of an extended validation cert, you're saying, oh, we had a conversation with you. We said hi. We called your phone number. We checked your Dun & Bradstreet rating. We did all this stuff. And so we are really sure you are the organization and everything you represent yourself to be when someone visits you. You are this company. This is your domain. That's a much bigger assertion. And so a third party, this certificate authority, has done all that research and is maintaining its currency in order to make that and to sign that assertion on the certificate.

So the notion of gradation in assertion only makes sense, only has meaning in the context of a certificate authority's signing and what that signature asserts. It doesn't have any meaning in the context of a certificate being delivered over DNSSEC using DANE where you're using secured DNS to acquire the certificate from the server of the domain to then use that to assert the identity and to provide encryption for your connections. So really two entirely separate delivery mechanisms, one where quality of certificate means something, the other where it doesn't.

And, finally, Reaper. In the week that has transpired since we discussed this somewhat breathlessly - the whole industry was last week - a lot more has come to light. Pretty much all of the interested and interesting security companies have had a chance now to assess this, to look at it, to do their own analysis. The one of all of them that I liked the best was the take that F5 Labs had, which they first posted last Thursday, so two days after, well, remember this all happened on Monday, so we were happy that it happened Monday because we had the podcast on Tuesday.

Two days later, F5 Labs came out with their look, which was much less frantic. And then today, this morning, on Halloween, October 31st, they revisited their last Thursday position, so five days later, to sort of further substantiate their position. I got the sense that they were being attacked a little bit by downplaying this. But I really think they got it right. First of all, the title of their posting last week I liked a lot. And in fact I tweeted the link to this, I guess it was last night. Actually it had a typo in it. Instead of "botnet" I said "bonnet," which led to many interesting funny responses on Twitter.

Leo: The Reaper Bonnet, my favorite.

Steve: The Reaper Bonnet, exactly.

Leo: That's my Halloween costume.

Steve: So they titled - so their posting they title "Reaper: The Professional Bot Herder's 'Thingbot.'" And first of all, I love the term "thingbot." We're going to adopt that, "thingbot," because that's pretty much what we're going to have from now on is Internet of Things bots. And so these are going to be Thingbots.

So last Thursday they started by saying, "This isn't your mama's botnet. This is a proper botnet. If you were the world's best IoT botnet builder, and you wanted to show the world how well-crafted an IoT botnet could be, Reaper is what you'd build." And this is nothing like what we heard during the first couple days. They wrote: "It hasn't been seen attacking anyone yet, and that," they write, "is part of its charm. But what is it doing? We've got some ideas."

And then they interrupted themselves for their morning update this morning, where they said: "The intentions of Reaper are as unclear today as they were a week ago. We hold to our position that the interesting aspect of Reaper is not its current size, but its engineering, and therefore its potential. From a pure research perspective, we're interested in how Reaper is spreading. Instead of targeting weak auth like a common thingbot, Reaper weaponizes nine, and counting, different IoT vulnerabilities. We think the current media focus on the numbers instead of the method is a tad myopic." Then they said: "See the next update section below for our clarification." And we will.

So in their original posting they said of size and position, they wrote: "Krebs puts the current size of Reaper at over one million IoT devices. We have data that suggests it could include over 3.5 million devices and could be capable of growing by nearly 85,000 devices per day. The reason Reaper has gotten so big and, honestly, the reason we're so impressed with its construction is that, unlike its predecessors, Mirai and" - and I don't know how to pronounce this. Persirai? P-E-R-S-I-R-A-I. It ends in R-A-I. Persirai? I don't know how to pronounce it.

Leo: Persirai. I say Persirai.

Steve: Anyway, whatever that is. Good. "Reaper uses multiple attack vectors." Again, remember, Mirai was just guessing usernames and passwords. "Reaper uses multiple attack vectors. Mirai used default passwords," they write. "Persirai used the blank username and password combo, which," they write, "frankly is such a dufus security error on the part of the manufacturer that we feel it barely deserves to have a CVE. Reaper," they write, "is almost showing off by not even bothering with password cracking, and instead just exploiting different vulnerabilities."

Leo: This is a nation-state. It's got to be.

Steve: Yes, yes, "remote code executions, web shells, et cetera, in nine different IoT vendor devices." And then they interrupt themselves again with their morning update from this morning. "Reports on the 'size' of Reaper vary. We've scanned," they write, "750,000 unique devices that match the nine vulnerabilities currently exploited by Reaper. We regularly scan 85,000 new [they used the term] 'Reaper-compatible' devices per day." Meaning F5 Labs has found three quarters of a million devices that match the

nine reported vulnerabilities currently incorporated by Reaper, and they're discovering an additional 85,000 new ones per day, which Reaper could also be doing.

They say: "We don't know which of them are actually infected, but there's no reason that Reaper itself couldn't infect them, unless its authors didn't want it to. Nine vulnerabilities," they write, "currently used by Reaper are fairly rudimentary as vulnerabilities go. If the thingbot authors were to include a few dozen existing vulnerabilities that fit Reaper's device-targeting profile, we think they could grow the thingbot by an additional 2.75 million nodes, if they wanted to. Adding that 2.75 million to the 750,000 that are currently Reaper-compatible gives us the number 3.5 million."

And they say: "Note: We will not be disclosing the additional CVEs as that would simply expedite the authors' exploits." And they have a nice little chart that I dropped into the show notes which, Leo, you just showed, showing us 750,000 plus 2.5 million brings us to 3.5 million. They said: "The actual size of Reaper is probably limited to whatever size its authors want it to be."

Leo: Exactly. They don't want command-and-control of too many servers.

Steve: Yes. And this feels developmental. This feels like, yes, that's the other thing.

Leo: Oh ho.

Steve: They're not using it to attack. They're using it to hone their system.

Leo: Interesting.

Steve: They said: "The actual size of Reaper is probably limited to whatever size its authors want it to be. Right now," they write, "it feels like its authors are experimenting, building and testing. Maybe Reaper is pure research. We don't know, and that's kind of why we respect it." And then they say: "Reaper has better IoT security. Unlike many of the devices it infects, Reaper has an update mechanism."

Leo: Of course it does. I'm telling you. And the Lua, and the commands. This is definitely a nation-state.

Steve: Yes, yes. And they said: "How impressive is that? If it weren't malicious, it might qualify to meet the standards of the new Internet of Things Cybersecurity Improvement Act of 2017 federal requirements." They wrote: "Heck, the authors could even make a distribution out of it, and it could become the default remote management platform for IoT." Because it does it better than anything else.

And they ask the question: "Is it malicious? So far Reaper has not been seen attacking anyone with massive volumetric DDoS attacks. Yes, that's a good thing. At least one of us" - because it had three authors - "thinks it might never be seen attacking anyone. If Reaper were to start being used as the ultimate Death Star weapon, that would cheapen its value. It would also result in an active takedown campaign.

"Remember how at least two strike-back bots were created to combat Mirai after it attacked Krebs, OVH, and Dyn? Brickerbot actively wiped the file systems of infected IoT devices, in many cases turning them into little more than bricks. Hajime was more polite and merely blocked ports and left a cute little note informing the device owner that their device was participating in attacks and please patch. If Reaper starts attacking people with DDoS, it will turn from a marvel of thingbot infrastructure engineering into - yawn - another volumetric attack tool. The bot herders would be hunted down by law enforcement, and the bot would be disassembled."

So they say: "What is it doing? Right now Reaper is an object lesson for IoT manufacturers and security researchers. It's like a giant blinking red light in our faces every day, warning us that we'd better figure out how to fix IoT security soon." And they conclude with: "Is there a lesson yet?" They said: "As predicted, we will continue to see more thingbots arise as we expect 'things' to be the attacker infrastructure of the future. Just because Reaper is the latest doesn't mean it will be the last. We've added Reaper to the list of botnets that we're monitoring. We suspect that entire existing botnets will get folded into it, whether they want to or not." So it's going to be subsuming other botnets, taking them over and acquiring them.

They write: "If Reaper doesn't attack anyone or give away its intentions, it may enter the same mythical space occupied by the Conficker worm of the late 2000s. At its peak, Conficker infected over 10 million Windows computers and caused great concern because it could have done an insane amount of damage. But it was never activated, and it remains a study in bot construction. The obvious lesson is that the state of IoT security is still incredibly poor, and we need to do a better job of threat modeling the Internet of Things." So a beautiful piece of work by F5 Labs. Thanks, guys.

Leo: And who knows?

Steve: And I think they got it exactly right.

Leo: Maybe John McAfee wrote it, and he's going to release it as a way to update IoT security.

Steve: That's definitely John's modus, yes.

Leo: The ultimate IoT update tool.

Steve: It is terrifying.

Leo: I love it that it has better security than the devices it's acting on.

Steve: Exactly.

Leo: It just, I mean, am I wrong, but it feels like a nation-state. This is too well done.

Steve: It's beautifully well done, yes. The fact that they've got a network of interacting command-and-control, it's deliberately just sort of percolating along without being in a hurry. I think these guys have characterized it exactly right. So we got something fun to keep an eye on, too.

Leo: We're going to keep watching.

Steve: Yup.

Leo: See what happens. Hope it's our side. We don't know, though, will we, till it takes off. Steve Gibson, you see, this is why you listen to the show, everybody, because this is where you learn the good stuff. If you want to hear it live as it happens, tune in on Tuesdays, right after MacBreak Weekly, about 1:30 Pacific, 4:30 Eastern. Next week it'll be 21:30 UTC, 21:30 UTC because...

Steve: Let's see. We're falling back.

Leo: We're falling back.

Steve: So we get an extra hour, yay.

Leo: So an hour later. Oh, it's confusing. UTC doesn't change, obviously. We do. And the reason I give UTC is so that you can calculate your UTC offset next week and know what time it really is. Just 21:30, and then add or subtract what you need to. I'm going to subtract eight because that's how we do it here in the TWiT Brick House or Eastside Studios. If you want to watch live, got to TWiT.tv/live. If you do that, please join us in the chatroom. Great bunch of people at irc.twit.tv. You can use an IRC client, but you can also just use the website. We have a web-based IRC client.

And you can also ask questions of Steve. He's on the Twitter, @SGgrc, or on his website, GRC.com/feedback. While you're at the website you might note you can download on-demand copies of the audio of the show and read the fine transcripts that Elaine Farris does for Steve. He pays for them and makes them available. That makes it also searchable, which is really nice. You can search for terms and so forth and find the podcast you want. GRC.com. And let's do Steve a favor. Everybody buy a copy of SpinRite. If you have a hard drive, SSD or spinning, you need SpinRite, the world's best hard drive maintenance and recovery utility. There's plenty of free stuff there, as well, including ShieldsUP! and his SQRL project.

Steve: Everything else is free.

Leo: And it's all free. GRC.com. We have copies of the show, audio and video, if you want to see Steve's smiling face, at our website, TWiT.tv/sn. That's where you can go back in time, too, find earlier episodes, all 635 of them. Steve has all 635, too. And also a subscription button there because really the best way to get this is to build your own collection of Security Now! shows. Just subscribe. That way you'll download them automatically when the show's over and we put it out. Takes a couple hours to edit it and get it out the door. But later on a Wednesday, or a Tuesday, I should say, or early Wednesday, depending on where you are, you'll get a copy of it, and you'll keep it on your phone or on your tablet, and you'll always have it, and that's a good thing. So subscribe.

I think that concludes this episode of Security Now!. I can't think of anything else.

Steve: We've got it covered again for another week, my friend. And we'll see what happens in the intervening seven days.

Leo: All right. And you've seen the whole "Stranger Things."

Steve: Yes, yes, yes.

Leo: Well, by next week I'll have seen it all, too.

Steve: Yes. It's good. I agree with you. It's good. There's just one episode, actually Lorrie noted that, I think, that it wasn't written by the same guys, was it the something brothers with a "D."

Leo: The Duffer Brothers.

Steve: The Duffer Brothers. And I think she spotted that the dumb one, it was number seven, it was like, okay, why, what's going on here? Anyway, but overall...

Leo: Seems like even the best, one of the best TV shows of all time, "Breaking Bad," had one dopey episode, the fly episode, where the entire episode is them chasing a fly in the meth factory. Like just dopey.

Steve: Yeah.

Leo: Sometimes you just have to, you know, you have to take it out of gear and coast for a day.

Steve: Anyway, it was definitely fun and worth watching.

Leo: Steve writes every episode of Security Now!. That's why there's no dead spots. Thank you, Steve.

Steve: No flies.

Leo: See you next time.

Steve: No flies on me.

Leo: No flies.

Steve: Bye, Leo.