Post-Coinhive Cryptojacking

Description: This week we look at the mess arising from Mozilla's intermediate certificate expiration (the most tweeted event in my feed in a LONG time!), Google's announcement of self-expiring data retention, another wrinkle in the exploit marketplace, Mozilla's announcement about deliberate code obfuscation, a hacker who hacked at least 29 other botnet hackers, a warning about a very popular D-Link netcam, who's paying and who's receiving bug bounties by country, another user-agent gotcha with Google Docs, a problem with Google Earth on the new Chromium Edge browser, and a bit more about Edge's future just dropped at the start of Microsoft's Build 2019 conference. Then we take a look at the continuing and changing world of cryptojacking after Coinhive closed their doors last month.

High quality (64 kbps) mp3 audio file URL: http://media.GRC.com/sn/SN-713.mp3
Quarter size (16 kbps) mp3 audio file URL: http://media.GRC.com/sn/sn-713-lq.mp3

SHOW TEASE: It's time for Security Now!. Steve Gibson is here. Lots to talk about. Firefox broke on Friday night. Steve has a deep dive into what really went wrong. We'll talk about Microsoft and Google securing their browsers and why you might want to be a little bit careful about using a particular D-Link camera. It's all coming up next on Security Now!.

Leo Laporte: This is Security Now! with Steve Gibson, Episode 713, recorded Tuesday, May 7th, 2019: Post-Coinhive Cryptojacking.

It's time for Security Now!, the show where we cover your security and your privacy online. Thank goodness we've got this guy right here to explain it all to us, Steve Gibson of the Gibson Research Corporation. Hi, Steve.

Steve Gibson: Leo, it is great to be with you once again. This is the latest occurring first Tuesday possible of the month, being the 7th of, what is it, May, which is to say that next Tuesday will be Patch Tuesday, also as far into a month as possible. I don't know why that matters, but just a random observation. I guess what occurs to me is I'm always checking back to see what happened in the previous week. And so it occurred to me, it's like, oh, look, Wednesday was the first, so that means we're going to have a very late Patch Tuesday, and I don't need to worry about it this time.

Today's podcast is titled "Post-Coinhive Cryptojacking" because we discussed this months ago when the announcement was made by the ill-reputed Coinhive guy that he was shuttering his, well, he doesn't really have doors. His ports. Shuttering his ports. And that's like, sorry, we're going to be closing down. And of course we talked about it at the time, how the relative collapse in the previously very high valuation of cryptocurrencies

no longer really meant that it was worth the substantial reputational damage that this guy was taking from the fact that his service was so much abused by mal actors who were foisting his script onto unwitting users' browsers. So he closed his doors a month ago, on March 8th. Wait. April 8th. And so here we are - maybe it was March 8th. Maybe it's been two months.

Anyway, we are now after Coinhive. And the question was what are we going to see in terms of replacement strategy for, okay, we no longer have an online service, so what fills the vacuum? Well, we know what the answer to that is a few months downstream, so we're going to talk about that. But as I assembled the overall news of what happened in the past week, I thought, you know, this is sort of the web browser podcast because there's just a lot of browser news. And in fact the screw-up of Mozilla's over the weekend was one of the most oft-tweeted things into my Twitter stream that we've seen in a long time because Mozilla did something that messed things up for add-ons to Firefox.

Anyway, lots of stuff to talk about. We're going to look at this, as I said, the mess with Mozilla's intermediate certificate expiration. Google's announcement, welcome announcement, and I guess very surprising in retrospect since we've talked about what the Congress, the letter that had been written last week, their self-expiring data retention system, which is as far as I know not yet available, but due imminently. Another wrinkle in the exploit marketplace; the marketplace for zero-day exploits has surfaced.

We've got Mozilla's announcement about deliberate code obfuscation in their browser add-ons, essentially following Google's previous position. News of a hacker who hacked at least 29 other botnet hackers. So, as they say, sort of a variation on "no loyalty among thieves." A warning about a very popular D-Link netcam. There's little question that many of our listeners will have one of these. And so we need to sort of warn everyone. An interesting graphic that I ran across on a story that was sort of about other stuff, and I didn't care about the story, but I loved the graphic, which shows who's paying and who's receiving bug bounties by country, which was sort of interesting.

We have another user-agent gotcha with Google Docs. A problem with Google Earth on the new Chromium Edge browser. A bit more about Edge's future just dropped yesterday at the start of Microsoft's Build 2019 Conference. Then, as I said, we've got a little bit of miscellany, and then we'll take a look at this continuing and changing world of cryptojacking after Coinhive shut down their service. And this Picture of the Week, since there was really nothing else that jumped out at me, is just too fun. I don't know if you've seen it before, Leo.

Leo: It's kind of an old joke, but I like it that the guy actually built it.

Steve: Yes, exactly.

Leo: I'm very impressed.

Steve: Simple, clean, and I don't know if it was a science fair project or what it was.

Leo: A decision tree.

Steve: Yeah.

Leo: An engineering flowchart, if it will.

Steve: Exactly. The universal solve it problem flowchart.

Leo: This engineer was pretty smart. I like this picture.

Steve: Yeah, it's a great picture. Okay. So we'll describe it for those who do not have video. First the two "no problem" outcomes. So this is titled "Engineering Flowchart." And it's a simple two-layer binary tree. So does it move is the main issue here. Does it move? So if you follow the "no it doesn't move" branch, then of course the next question is, should it? And if you follow the no again - so does it move, no; should it, no - then you end up at the no problem output.

On the other side of the tree, we have does it move, and we go in the yes direction. And then the question is, well, should it? And if the answer is yes, then once again it moves, and it should, no problem. So then the other two outcomes of this two-layer binary tree deal with the problem case. So does it move, no; and then should it, yes. And of course that takes you to the can of WD-40 because it doesn't move and it should, it's supposed to. And then of course the engineer's universal fix-it-all is the fourth branch, the does it move, yes; should it, no. So that takes you to the roll of duct tape. If something is moving, put some duct tape on it and stop it.

Leo: I like it that this guy actually built this out of wood and with real duct tape and real WD-40.

Steve: Oh, it's beautiful.

Leo: Yeah.

Steve: Yeah, it's absolutely beautiful. Two of your TWiT people, Leo, have responded to my sending everybody this week's show notes because Google red-flagged it and refused to allow it to be downloaded, saying that this was potentially malicious and dangerous.

Leo: Oh, they've all been getting training. Okay.

Steve: So it turns out that it's probably my coverage of Coinhive and its successor because I have links to - I have URLs in the PDF that refer to those by name. And so Google said, "Agh, don't download this."

Leo: That's good. That's good.

Steve: So it's like, okay.

Leo: I think that's a good thing, yeah.

Steve: Cool your jets. That's all right. It'll be just fine. Okay. So the most oft-tweeted alarm sounded on Saturday due to a minor oversight over at Mozilla, entirely understandable. I mean, so, yes, it's the case that some communication broke down somewhere because someone should have been notified proactively about the expiration of a certificate. It's not clear to me whether - and I did do some digging, but I couldn't really get a detailed understanding. I wasn't able to, I mean, I guess I didn't go as far as I could have. But what happened was an intermediate certificate in the chain of certification that was signing a collection of still-in-use popular add-ons for Firefox expired.

So this is different than the problem I described that I ran into two weeks ago when I was forced to replace the code signing certificate that I've been using in good standing for three years. That trouble was that the replacement certificate was unknown to Windows and other third-party AV. So as we discussed many years ago, when first explaining the details of certificate-based security, certificates form a chain of trust with a so-called "trust anchor," which is typically a self-signed certificate - that is, it's vouching for itself - that's implicitly trusted due to its residence in the system's root certificate store.

So we want to protect entry of certificates into that store because anything there is implicitly trusted. It signs itself. So that's why we've talked over the years about the concern of the growth of that store, how many CAs' self-signed certificates now appear there. A new policy of Windows is that that store will only be populated on demand. So instead of there being what I discovered in a Windows XP machine years ago and talked about it with some alarm on this podcast, it's like there were 400-plus certificates in my store. And it's like, what happened? I remember when there were 12.

So anyway, so you start with this trust anchor. And it's possible for the working certificate, that is, the one at the end of the chain that is the one you care about, to be directly signed by the root certificate. But that's not typical any longer, nor really practical. We don't want our root certificates to be expiring often, for one thing, because in some settings it can be difficult, tricky, in some cases impossible, to securely replace those roots on the fly.

So we want the roots to be long-lived. But with long life comes the danger that they might get loose, that the corresponding private key for them could get loose, in which case it would be valuable, and it would have a long life of potential exploitation. So that's not what we want to do. Also, every time that root certificate private key is used, it has to kind of come out of hiding. It has to, if it's like being stored offline, as we hope it is. It should be not available electronically. So whenever it has to be used, it's got to go online somewhere in order to be used.

So each of those events creates that little, you know, no matter how small it is, diminishingly small opportunity for it to get loose. I'm sure it's in some sort of a hardware security module, an HSM, in order to protect it. But still, you just - you want to minimize using it. So for all of those reasons, what's normally done is that the root signs an intermediate certificate that has a longer life still than the certificates it will in turn sign, but less long than the root certificate. And it's just - it's sort of like a more handy common daily use cert.

So that's why in general today's certificate chains don't just have two certificates, the actual working certificate and the root that signs it. Typically there's at least three, with an intermediate certificate sort of being a compromise between short life and the very long life of the root, and also one that it could be revoked or killed, or if something happened at least it wouldn't be living as long as the root.

So what happened over this past weekend is that in the months leading up to Saturday, this last Saturday, no one at Mozilla had noticed that the intermediate certificate which was essentially the anchor or the signer of a bunch of popular Firefox browser add-ons still in current use, including uBlock Origin and LastPass, would be expiring. And unless the user was running the Canary branch or had their browser in developer mode, and there is also a way to turn off signing verification, which no users typically do, any of those things would override the verification of the signature of the add-ons. Then Firefox protects its users from malicious add-on impersonation by validating the signature of every add-on, every time they're loaded by the browser.

And in fact I ended up not having a problem because I famously never shut down my Firefox. I mean, I have to go, when I hear that there's like some update available, I go, oh, and go into the About Firefox, and then it kind of wakes it up and says, oh, wow, look, you really need to catch up. So then I do. But anybody who shuts their computers down or closes their browser and then opens it on demand, they would have been hit with the unwelcome news that a whole bunch of their add-ons were not being trusted by the browser because essentially the process of verifying the signature requires following this chain all the way back to the root. And in this case the certificate that was vouching for the add-on had expired. And at that point, sorry, we stop right there. We don't trust the thing that is signed at that point.

So there was a bunch of "what to do" advice flying back and forth over the weekend, including those measures I talked about, like getting the Canary release or putting the browser in developer mode. The one thing that any user could have done, if they were desperate, and so I'm sharing this now for the next time this happens because this can, a user could have backdated their system's clock by one day because the validity period of the certificates is checked against what the system believes is now. And Leo, in fact, I know that you often, when you get people calling your KFI radio show, your Tech Guy show on the weekends, and they say, "I can't get into any of my secure sites, they're all broken, my browser's blah blah blah blah," you say, well...

Leo: We know what that means.

Steve: Yes. What is the date on your computer? And they look, and they go, oh.

Leo: How did it get to 1970?

Steve: That's right. Because not only is there a not-valid-after, there's a not-valid-before on these certificates. And so if your computer, like you lost your BIOS setting for the clock in your BIOS, and your computer turns on, and it believes it's 1970, then none of your certificates are valid anywhere because they've all been issued since then. So anyway, the point is that it's possible to use this little hack. You turn your date back a day. Then you launch Firefox. It checks all of your add-ons. Everybody's happy. Then you put the date back to today. So that way you're not persistently running with a backdated date so that your other things aren't confused, timestamps on files and so forth. But just during that brief period when Firefox loaded the add-ons, you say yes, everything's fine, look, oh, that intermediate certificate's going to be expiring tomorrow. Well, no, it already did, but we fooled the system.

So anyway, that little hack works. And so all of our listeners can tuck it away for the next time this happens again. The Mozilla guys scurried around, fixed the intermediate certificate, but they needed to update Firefox. And so we went from 66.0.3 where the problem was - and, oh, by the way, all previous Firefoxes - to 66.0.4, which fixes the

issue. So anyway, an interesting little glitch. And I'm sure they'll figure out how did this, you know, how did we fail to get notified of this? Because, I mean, we've talked about this happening. It does happen from time to time if, you know, the email address that your certificate authority or certificate provider has for you expired, maybe it was going to somebody who was on vacation, I mean, who knows how it happened? But anyway, kind of easy to fix. But it certainly was a source of concern for everybody over the weekend.

Following up on last week's news about the Sensorvault at Google and the news that, as far as the reporting indicated, unless otherwise instructed, they've been retaining all of the detailed, highly accurate location information for basically just "hoovering" it, to use the verb, like everybody that uses their stuff anywhere for all time has just gone into some massive server farm somewhere so that - for purposes we're not really clear about. But as we discussed, and as the reporting on this discovered, they were honoring so-called "geofencing" warrants which were causing them to turn over successively more personal and accurate location information on people who were believed to be instrumental, either as perps or as witnesses, to crimes for law enforcement.

So what just dropped from the Google blog was the news that they're going to give people some choice. And they said in their blog: "Whether you're looking for the latest news or the quickest driving route, we aim to make our products helpful to everyone. And when you turn on settings like" - okay, they're on by default; right? "When you turn on settings like Location History or Web & App Activity" - again on by default - "the data can make Google products more useful for you, like recommending a restaurant that you might enjoy or helping you pick up where you left off on a previous search. We work," they wrote, "to keep your data private and secure, and we've heard your feedback that we need to provide simpler ways for you to manage or delete it." Well, they certainly heard Congress's feedback.

They said: "You can already use your Google account to access simple on/off controls for Location History and Web & App Activity." And as I'll describe in a minute, it's actually "on" or "pause." They don't actually do "off." They just do "pause" because they're optimistic that you're going to come back at some point. They said: "And, if you choose, to delete all or part of that data manually. In addition to these options, we're announcing auto-delete controls that make it even easier to manage your data. Here's how they'll work."

And they said, and I have a picture of the - apparently this is not yet available. They haven't, as far as I know, produced a date certain yet. But we have seen what it's going to look like. And, yeah. So Leo has it on the screen. We have the first option is "Keep until I delete manually." Second option, "Keep for 18 months, then delete automatically." Or "Keep for three months, then delete automatically." So they said: "Choose a time limit for how long you want your activity data to be saved, three or 18 months, and any data older than that will be automatically deleted from your account on an ongoing basis. These controls are coming first to Location History and Web & App Activity and will roll out in the coming weeks. You should always be able to manage your data in a way that works best for you, and we're committed to giving you the best controls to make that happen."

So that's their statement. I think that setting a three-month auto storage expiration policy is probably a nice compromise for users of Google services who still desire the promise of location-based enhancement of their online experience without the creep factor of everywhere they have ever been for the last 10-plus years being silently logged, retained, and searchable in a Sensorvault database somewhere. And for users who are sure that they want no retention right now, it can be paused, as I mentioned, as Google insists upon phrasing it, and then scrubbed through a manual process.

While signed into Google, you click your profile picture, then you click on the Google account button that clicking on your profile picture reveals. In the left-hand column there's Data & Personalization category. So when you select that, you'll find Web & App Activity and Location History. You select each of those in turn, then flip the toggle to Pause, and then confirm with Google that that's what you really intend. And then in that confirmation dialog under Web & Activity, it explains pausing this setting doesn't delete any of your past data. You can see or delete your data and more at myactivity.google.com. And there's a link there in that dialog. And similarly, from the confirmation dialog that comes up when you pause Location History, it explains pausing this setting doesn't delete any of your past data. You can see or delete your data and more in this case at maps.google.com/timeline. And again they provide that as a link.

So I'm not really much of an uber privacy nut, Leo. I feel sort of like you do, that, well, this battle has been kind of lost. And many of our listeners were unhappy recently to hear my capitulation on that front. But I have to say that going over to myactivity.google.com and scrolling back through days of stuff I've done, nominally in private, I mean, not like I was really worried about it, but I was by myself with no one obviously looking over my shoulder. When you scroll back through that, it's a little sobering, knowing that all that's being archived somewhere for god knows what purpose. You know, it's a little bit creepy.

So for what it's worth, if anyone's interested, while you're logged into Google, go to myactivity.google.com. I remember doing it before and kind of thinking, whoa. And it can be handy, I have to say. If I remembered that I was being watched that closely, there have been times where it's like, okay, what was that I was searching for the other day? Well, Google has it all there. They've sucked it all up, and they're keeping it.

So anyway, it's nice that Google is responding. And really I think a year and a half or three months, those are probably two good choices. And it'd be interesting to see. I'm sure Google will be watching with interest, as the news of this gets around, what percentage of their users choose which or neither of those options. I'd love to know what the demographics are. There is sort of this tease of like, ooh, well, maybe I am getting value from this, so do I really want to turn it off? I'm kind of in that camp. Although I certainly respect that people are like, no, I don't want to be followed or tracked at all. So the good news is you can certainly delete that and also have Google doing it on an ongoing basis.

And so I suppose when they show up to fulfill their request for Congress to get an update on this, they'll be able to say, well, yes, we're now offering these features.

So in the evolving marketplace of this crazy world of exploits, an apparently Ukrainian talented and prolific hacker has been found to be himself - if it's a he, and if it's a self, we're not really sure, at least that's the way this person is presenting themselves - has been selling zero-day exploits to various APT, Advanced Persistent Threat groups. Kaspersky Lab has been watching the network comings and goings of a prolific hacker whose name is believed to be Volodimir, a Ukrainian name, who goes by Volodya as a nickname. And this nickname often appears in the code that this guy sells.

For the past three years this hacker has been selling Windows zero-days to at least three different cyberespionage groups, as well as into cybercrime gangs. This as a consequence of Kaspersky's monitoring of I guess the dark web comings and goings. The hacker's activities reinforced beliefs that some government-backed - and I should mention this is high-end - government-backed, state-sponsored cyberespionage groups are not only developing these things themselves, but also regularly purchasing zero-day exploits from third parties on the dark side of the 'Net.

These APT groups, believed to be operating out of Russia and the Middle East, have often been spotted using zero-days which were developed by real-world companies acting as sellers of surveillance software and exploit brokers for government agencies. Kaspersky's recent reporting shows that these APT groups do not shy away from dipping into the underworld hacking scene to acquire exploits that were initially developed by what appear to be lone hackers and sold to cybercrime groups.

This Volodya character, who Kaspersky Lab characterizes as one of the most prolific vendors of zero-days, first came to light three years ago, which is why we know this has been going on for three years, back in 2016, when he first sold, or offered for sale, an exploit under the name BuggiCorp, B-U-G-G-I Corp. And that sounds familiar to me, so I think we probably talked about it on the podcast. At that time, this hacker's actions were in the news because he put a Windows zero-day for sale, not on the dark web, but on what was at the time an infamous Exploit.in cybercrime forum. And back then it was a surprise to see a hacker openly advertising a Windows zero-day in public, since those transactions typically happened in private.

So as a consequence, the world watched as this BuggiCorp dropped his initial asking price a couple times, from initially $95,000 down to $85,000, at which point he sold it to a cybercrime group and began to develop a reputation for producing high-value zero-days. Now he has sort of an exclusive, dedicated clientele and is continuing to sell other zero-days that he's developing privately, in some cases with prices reaching $200,000. Kaspersky has a group with the abbreviation GReAT, G-R-E-A-T, stands for Global Research and Analysis Team. They are the elite APT tracking unit which has been watching this Volodya, knowing that he's fluent in Russian, although they believe he has a Ukrainian origin due to his name. They believe he's the author of the recent Win32k elevation of privilege vulnerability, which was fixed in March and which we talked about. That zero-day which has since been patched was in use by a cybercrime group focused on financially focused threats.

And this zero-day is only the latest of this Volodya's accomplishments. That earlier one we talked about three years ago was also a Win32k privilege of elevation vulnerability that was also found leveraged in the wild and had been used by and apparently sold to that Russian Group, Fancy Bear, which we also know of as APT28, Pawn Storm, Sednit, Sofacy, and Strontium. No one could really agree on their name. Or maybe they were not known to all be the same group until later. Anyway, as we know, they are one of the infamous groups that were behind the 2016 attack on the Democratic National Convention headquarters, the DNC.

Kaspersky has been watching Volodya and has seen him selling zero-day exploits essentially continuously through the years to these high-end APT groups that they follow and noted also that, in additional to zero-days, he's also been selling one-days, which is to say, the moment something is patched that he did not develop, he quickly reverse engineers the patch and puts it on the market, recognizing that it has been patched. He's not going to get the kind of money that he will get for a zero-day, but it's been patched, but not everybody's applying patches immediately. So there is some short-term value to an exploit which is sufficiently valuable, but which is not yet seen, the patch for which has not yet been deployed in systems.

So as Kaspersky sums it up in their reporting of this, this Volodya appears to be making a profitable career out of selling zero-day and other exploits and building quite a portfolio. They said in their reporting that they don't know that Volodya is not the front for a larger group or a team of exploit developers, or even maybe an exploit brokering company that fronts for other independent brokers. With anonymity comes a lot of uncertainty.

But anyway, I wanted to share this because I thought it was interesting. This sort of fleshes out another wrinkle of the nature of the contemporary underground ecosystem for exploits. And this just, again, another piece of information that makes me shake my head. I keep flashing back on that very first scene from the first "Matrix" movie where Neo was selling a hack for something to his friends. That movie was released on March 31st of 1999, so just over 20 years ago. Back then, I remember, it seemed like such a stretch, like real sci-fi fantasy. But it turns out just to have been amazingly prescient. We're in that world today.

Leo: Well, we don't know if the machines are running everything. But other than that, yeah.

Steve: You know, there is a compelling book, apparently, that suggests, I'm not saying, but I think it's called "The Simulation..."

Leo: Elon Musk believes the simulation theory.

Steve: I know, I know. Like with a high degree of certainty. Like what is it, like...

Leo: It's a billion to one that it's not a simulation.

Steve: That it's like we're not in a simulation. It's like, okay, Elon.

Leo: Yeah. Elon's smoking something, I think.

Steve: Long as we all play by the same rules of the simulation, then it'll all turn out okay. So Firefox has announced that they're following Chrome in banning browser extensions containing obfuscated code. We talked about this decision that Google had made for Chrome last October. And the Chrome ban took effect on January 1st of this year, so right at the strike of midnight 2019.

Google's Analytics, as we discussed at the time, had determined that 70 percent of malicious browser extensions deployed mechanisms of deliberate code obfuscation. And that made their task of manually inspecting the code for malicious behavior so much more and unnecessarily difficult. It was like, why are we putting ourselves through all this? Like going to all the work of untangling code which has been deliberately obfuscated. They were, for a while. Finally they just said, forget this. The simplest solution is just to say no more deliberately unreadable code.

So last Thursday the Mozilla folks decided to follow Chrome's example, and they updated their policies which would go into effect on June 10th, so exactly one month after this notification. They said about their guidelines, they said: "Add-ons may contain transpiled" - that's a concatenation of "translated" and "compiled," so transpiled - "minified or otherwise machine-generated code, but Mozilla needs to review a copy of the human-readable source. The author must provide this information to Mozilla during submission, along with instructions on how to produce a reproducible build." So Mozilla can verify that the source that was provided equals the binary that the author also provided.

They wrote: "The provided source code will be reviewed by an administrator and will not be redistributed in any way. The code will only be used for the purpose of reviewing the add-on. Failure to provide this information will result in rejection. Add-ons," they continue, "are not allowed to contain obfuscated code, nor code that hides the purpose of the functionality involved. If external resources are used in combination with add-on code, the functionality of the code must not be obscured. To the contrary, minification of code with the intent to reduce file size is permitted."

So as an industry we're sort of creaking forward, step by step, figuring out how to do all of this. I think this just makes sense. If users are downloading add-ons that they're going to be getting from some sort of curated store, you know, add-on facility, which is what Chrome and Mozilla are both offering, due to the power of the add-on there has to be some mechanism for oversight. So I just say bravo to Firefox for doing the same.

Actually, Google had the guts to do it first. The Mozilla people probably waited to see how that was going to work out, allowed Google to take the arrows in their back if they were going to, or use their strength as the number one browser in the industry and to have sort of first mover position, and then to follow. So anyway, I imagine that will be the policy, which Google has set for Chrome, now Firefox, and I don't know where Microsoft stands. But certainly with this foundation it becomes the expected protocol for people who want to create add-ons for web browsers. And I think it makes a lot of sense.

So a researcher, Ankit Anubhav, is an IoT botnet researcher whose work we've looked at previously. Unfortunately, his name looks like you would say Ankit, A-N-K-I-T. I was very nicely informed by a Twitter follower that it's Ankit [pronounced un-ky] Anubhav. So thank you for providing the pronunciation help. He's the principal researcher at NewSky Security who arranged a discussion and interview with a hacker who calls himself Subby, S-U-B-B-Y. What transpired was interesting and entertaining.

So first of all, as we know, typical IoT botnets like Mirai and QBot rely upon obtaining access to their target devices, that is, the things that they're trying to attack and compromise using typically the device's weak or default credentials or some sorts of known problems with the router or smartphone, or not smartphone, smart camera, netcam and so forth. However, as it turns out, the hackers themselves behind these are not very security conscious or, it turns out, actually, very tech savvy. They are using very poor, weak, and often default passwords to protect their own command-and-control servers.

So this would mean that in theory another black hat could come along, figure out the IP addresses of the command-and-control servers, and then brute force those to obtain control of the command-and-control servers to basically commandeer the botnets away from the people who gathered the bots in the first place. Rather than building their own, they could just steal them from a different bot herder.

So as Ankit learned during his interview, this is exactly what happened in this case. The hacker hacker, who calls himself Subby, brute forced at least 29 different IoT botnet command-and-control servers, finding that they were in fact using extremely trivial credentials. I have a screenshot from the blog posting describing this interview that shows, for example, we've got in one case at 139.xx.xx.31 a botnet family named Frosty on port 8372 was using username "root" and password "root."

Leo: I like Emily/rawr. That's my favorite. Yeah, they're not trying to secure these. I see root/root a lot. It doesn't look like they're really trying to secure it.

Steve: Well, actually they are, Leo.

Leo: But that's dopey.

Steve: Well, wait till you hear how dopey these people are. I mean, it's a little shocking to understand the nature of them. So here is, sort of in Q&A form, the transcript of Ankit's interview with Subby. Ankit wrote: "I decided to ping Subby" - and English is not his first language, so we'll forgive him. He said: "I decided to ping Subby to know more answers besides the data, like why and how he is doing this, and what is the motive. Some of the excerpts from the interview are as follows." So Ankit asks: "What technique are you using for brute forcing the servers?" Subby replies: "I have a network of honeypots configured to capture binaries over Telnet SSH. The captured C2 IPs are then..."

Okay. So what he's saying is his honeypots capture the attempt to infect his honeypot as if it were an IoT device. He pretends to allow that to have happened, which allows him to then obtain the command-and-control server IP that that infection would then have attempted to reach out and connect to.

So Subby says: "The captured C2 IPs are then" - and of course C2 is command-and-control. "C2 IPs are then port scanned via NMAP to find the C2 port. For brute forcing I'm using a dictionary-style attack, coupled with a password list which has common username/password combos. In addition to this, each C2 undergoes a random-style password attack which continues up to six alphanumeric characters under the user 'root.' I changed the user to something specific if I have prior knowledge of the C2. Each cracked password is added to the password list used when brute forcing the C2s in the future."

So he obtains the IP from his honeypot that pretends to get infected. Then he goes out and he brute forces the login for that server. Oh, and the reason he's not obtaining the port is that typically the IoT's port, its connection port will be different than the port used to log into it for command-and-control. So he does an NMAP scan to find other open ports and then brute forces those.

Ankit then asks: "As you have found out, many of the credentials are very weak. Why do you think this is happening?" Subby says: "It's obvious as to why this is happening. A large percentage" - Leo, are you sitting down? "A large percentage of botnet operators are simply following tutorials which have spread around in the community or are accessible on YouTube to set up their botnet."

Leo: They're script kiddies.

Steve: Yes. "When following these tutorials, they do not change the default credentials. If they do change the credentials, the password they supply is generally weak and therefore vulnerable to brute forcing." Ankit asks: "How much total bot count you have achieved brute forcing these C2s?"

Subby replies: "Within the first week of brute forcing I surpassed 40,000 devices. This was quite an inflated number due to possible duplication. It is well documented that botnet operators like to artificially increase their bot count. I estimate the number to be closer to 25,000 unique devices. I was able to get a reliable network traffic graph produced of the traffic generated from all the botnets compiled, and it was just under 300 gigabits per second. This high number was achieved because of the vast amount of Digital Ocean servers on many of the botnets. It's well known that Digital Ocean are relatively slow in comparison to other hosts when dealing with abuse complaints. Since

then, the number of C2s vulnerable to brute forcing has lowered considerably," and he says, "(30-40%). This is likely due to how vocal I've been when brute forcing the servers. I have actively contacted botnet operators, letting them know that I managed to obtain access to their C2."

Ankit then asks: "Why are you doing this? Are you doing this for DDoS?" Subby replies: "The main reason I undertook this task initially was to see how well brute forcing would work on C2 servers and whether it would be an efficient way of getting access to devices, rather than having to use exploits or the usual loading onto devices with weak passwords via Telnet and SSH." He says: "Since Mirai was released, Telnet has slowly become saturated, and it's hard to get a decent number of bots." I assume what he means is that any previously exposed Telnet has already been used and then closed, or the authentication increased so that it's not obvious how to get in. Basically, by "saturated" meaning that large and easy opportunity has just been drying up.

And then, finally, Ankit writes in his conclusion: "In one previous case, we observed the SQL database of an IoT botnet having root:root" - meaning username "root," password "root" - "credentials before; but as we see now, the problem is bigger and not a one-off case. Pure novices in the field of IoT are increasing. We are not talking about script kiddies" - meaning that they're like a level above - "but such low skilled actors who are unable to set up a botnet from source, yet they want to launch a DDoS by doing nothing other than pressing a button. We also observed mistakes as novice as not replacing the botnet dummy C2 IP with their own."

And he cites an example: "Unstable, the Turkey-based author of Z3hir IoT botnet, has gone to the extent to release a video where he tells how to replace the dummy C2 (0.0.0.0) with the attacker's IP. When asked about the video, he said: 'Yes, these script kiddies are not changing IPs, and they are blaming me when the botnet does not work.' Recently [someone tweeting who goes by] @VessOnSecurity also observed a similar case where his honeypots found attacks with a dummy C2 of INSERT-IP-HERE, pointing to the fact that the threat actor forgot to change the dummy C2 IP to insert the real one, yet proceeded to attack the IoT devices, and subsequently the honeypots, with a non-functioning command-and-control server."

And so he says: "Interestingly, despite not knowing what they were doing, script kiddies often succeed, thanks to good support and tutorial videos by threat actors. In many cases having a secure password and updating the IoT devices can save one from these low-hanging fruit attacks." So, wow. Amazing that that's the nature of this. But again, you're going to have people who are highly skilled, operating in the dark web, selling zero-day exploits for $200,000 on one end of the scale. And on the other end you're going to have interested individuals who haven't done any of this sort of work before obtaining a kit off the shelf and really lacking any concept for how to even, you know, which buttons to push in order to make it go and make it do something. Amazing.

I wanted to bring this news from ESET researchers to our listeners' attention in case anyone had a D-Link DCS-2132L netcam. It is apparently a very popular camera, been on the market for many years. A ton of them exist in the wild. And they have never been secure. ESET's blog posting and disclosure at WeLiveSecurity.com was titled "D-Link camera vulnerability allows attackers to tap into the video stream." And the subtitle was "ESET researchers highlight a series of security holes in a device intended to make homes and offices more secure." So as I said, it's the DCS-2132L.

They wrote in their posting: "Many people are looking to improve the security of their homes or offices by installing smart cameras. With a direct connection to the Internet, their surveillance stream is just a few clicks away and available at any time. Yet this kind of convenience can quickly turn sour," they wrote, "if the camera suffers from a security vulnerability that opens the door to unauthorized actors. As shown by ESET smart home

research, this is the case with the D-Link DCS-2132L cloud camera, which allows attackers to not only intercept and view the recorded video, but also to manipulate the device's firmware. The most serious issue," they said, "with the D-Link cloud cam is" - get this - "the unencrypted transmission of the video stream."

That's right. You heard that correctly. It's an Internet-connected streaming video security camera that does not encrypt its video. That is, a passive eavesdropper of the camera's stream is getting an MJPEG or an H.264 unencrypted video stream just by eavesdropping.

ESET writes: "It runs unencrypted over both connections" - meaning, and this is also even further stunning - "between the camera and the cloud and between the cloud and the client-side viewer app," they wrote, "providing fertile ground for man-in-the-middle attacks and allowing intruders to spy on victims' video streams." And so in my words, in other words, they didn't really pay any attention to security at all.

And you could almost, I mean, if you wanted to forgive them for not encrypting the camera's feed to the cloud, making some excuse like, well, there's only a camera and a barely capable widget in there. Like we used up all of our budget just to create a WiFi connection. We didn't have any ability to do security. You could, if you wanted to forgive them for that, okay. But there's no way to forgive them for the stream from the cloud servers back to their app not encrypting. I mean, it's just amazing. So they just didn't bother with that, either, with no encryption.

The viewer app and the camera communicate via a proxy server connecting on port 2048 using a TCP tunnel which is based on a homegrown D-Link tunneling protocol. But none of it is encrypted. The actual payload, which is the sensitive contents, is in the clear so that all you have to do is sniff. And so they use the term, ESET uses the term "man in the middle." It's a man in the middle intercepting network traffic. It's worth noting, though, again, it doesn't have to be an interception. A passive eavesdropper is all this takes. So they responsibly notified D-Link.

Oh, there's other problems, too, I should also mention. There is, in addition to unencrypted audio and video, the device supports on-the-fly firmware updating, but offers no firmware update authentication mechanism. Which means that any sufficiently motivated adversary could look at what the protocol is - it's not documented, but it's trivial to reverse engineer, the ESET folks did - figure out what's going on, and essentially submit a POST to the camera which causes it to update its firmware from a source that the attacker can provide, allowing firmware to be changed for any purposes whatsoever. And there is no protection against a third-party maliciously supplying firmware to the camera.

So more than six months ago ESET notified D-Link. Apparently D-Link responded immediately, saying thank you for the news. We'll notify the proper parties and get right on that. Nothing happened for six months. The most recent firmware is dated 2016, so no indication that there has been any change being made by D-Link; and, as far as they know, none forthcoming. So the camera is currently on the market. I put a picture of a snap I took of the camera being sold on Amazon for 60 bucks. It's moderately well rated. It's being sold both by brick-and-mortar and online. And I'm just saying, if you own one of these, in fact what ESET said was be careful where you aim it. Be careful where you point it. You do not want this aimed at anything that is highly secure in your residence or in your business because it's not encrypting.

Leo: Are other cameras encrypting?

Steve: Many cameras are, yeah. I mean, even, you know...

Leo: Not all.

Steve: Well, we talked about even, long time ago, even baby cams that were doing sort of some encryption, sometimes it was sketchy, but they were making an effort. There's just none.

Leo: They don't have a lot of processor in these devices.

Steve: That is true; they don't.

Leo: I mean, you could, if they had the horsepower...

Steve: Although it's able to update its firmware. So, I mean, if it's able to update its firmware, you'd think, I mean...

Leo: Put a certificate on it of some kind.

Steve: And encryption is just, you know, simple symmetric encryption, even if they didn't do any fancy public key, private key encryption is just not difficult to pull off any longer.

Leo: At least do a ROT13. Do something.

Steve: Yes, exactly. Do a scramble. Wow. So I didn't have a story. This was part of a different story that didn't really make it onto the headline for the podcast. But I liked this graphic. This was a very cool graphic. This is a visualization of the flow of software bug bounty money, that is, offerings and payouts, from those putting up the bounties on the left to those collecting the bounties on the right. And of course our listeners who are not seeing the video and don't have the show notes can't see this. But essentially it shows the largest piece of the pie, 15, nearly just shy of $16 million over this period of time was sourced, bounties offered by those in the U.S.

In second place with a huge drop, at 1.2 million, is Canada. Third place, another drop nearly to a quarter of Canada. Germany comes in at 458,000, Russia at 308,000, Singapore 256,000, U.K. at 252. So that gives you a sense. U.S. at 16 million, Canada 1.2 million, then a quarter million for Germany. So that's the sourcing.

On the right side of the chart we have the nationality of the recipients of the bounties. So the largest, but not nearly as much as the offering, but the largest is the U.S. at 4.150 million. Number two, close to the U.S., is India at 3.1 million. Then Australia at 1.23 million, Russia at 1.3 million, the U.K. at 916. So on the offering side USA, Canada, Germany, Russia, in that order. On the recipient side, but more evenly spread, USA, India, Australia, and Russia. And it's interesting that Australia is so well represented there. There's nobody, they're not offering any bounties over on the left, but they're collecting. They're in number three position, collecting...

Leo: Ain't no fools, mate.

Steve: ...a 1.3 million chunk.

Leo: We're taking the money. We're not giving it.

Steve: That's right.

Leo: By the way, somebody, ScooterX in our chatroom, sent me a link to the Wyze Cam. You know, those are those, I mean, you can't get cheaper than them, $20 streaming video cameras. They probably have very, very simple processors. They use TLS, AES 128-bit encryption to protect the security of the live stream and playback data. Every device has its own secret key and cert so they can validate the identity during a handshake.

Steve: Wow. Wow.

Leo: I mean, they do it right.

Steve: That's what you want, yes.

Leo: And that's a cheap - there's no way anybody's using a slower, lower processor than the Wyze Cam.

Steve: Yeah, so rip those D-Link suckers out by their roots.

Leo: Yeah. Yeah, 20 bucks for a Wyze Cam and replace it.

Steve: Wow. Yes, yes, yes. So we have sort of another interesting user-agent gotcha with Google Docs and the new Microsoft Edge browser. We were just talking about this mess with user-agent headers and the idea that websites are expected to tune the code they produce for this or that browser. And frankly, my approach would always be to simply choose, well, if I were developing web-side stuff, the lowest common denominator feature set that all browsers share and write to that so that the same thing works independent of and across all web browsers.

But I understand that when what's being delivered is a cutting-edge web-based application, the lowest common denominator might be too low. And this is the situation with, for example, Google Docs, which has always presented a warning message when a web browser is not known to be capable of running its online app. Microsoft's new and forthcoming Edge browser, which as we know is based on the common, and increasingly common, Chromium engine would be expected to be able to run Google Docs without trouble. So observers and early adopters and probably a lot of our listeners - because we now know that Microsoft has made it officially available for download. It's officially

available under Windows 10. It also runs under Windows 7. And they will be making it available for Windows 7, which I think is great because it's a nice browser.

Anyway, so they observed, with some puzzlement, when Edge showed the message "The version of the browser you are using is no longer supported." Well, first of all, what? No longer? It turns out, upon examination and experimentation, that Google Docs maintains an explicit whitelist of known compatible web browsers. And at the moment, this new Chromium Edge is transmitting a slightly altered user-agent header that Google Docs' browser checker didn't recognize.

We were previously talking about how the user-agent header in the context of Microsoft's Edge was presenting different user-agents, depending upon the domain being visited. Remember Netflix did one thing, and some random site actually in Australia, as I recall, was doing something else. I mean, it was saying it was a bit of a chameleon. So it would presumably be simple for Microsoft to present a Google Docs-compatible user-agent header to Google Docs, or Docs could be instantly taught about the user-agent header which is being produced by Microsoft's forthcoming browser. And this is one of those weird things where, like, both of them could fix the problem. I wonder who's going to? I'll be mildly curious to see whether Google Docs just instantly fixes it, as they could; or Microsoft says, oh, well, we'll just pretend to be what Google Docs wants when we're at Google. Who knows?

And in something that's sort of related, Google Earth won't run under Microsoft's Edge browser, but not for the same reason. Turns out there's a very different and true incompatibility in the case of Google Earth. As we just said, with Google Docs it's like, eh, I mean, it will run, it's just that the user-agent header is presenting something that Google Docs isn't sure about. Okay. Turns out that when users try to launch Google Earth, which is supposed to work with Microsoft's new Chromium Edge, they get an error that, Leo, you probably have seen before because it rang a bell with me. The error is "Aw snap!"

Leo: Oh, yeah. Yeah, I've seen that many times, actually.

Steve: Yeah. "Aw snap! Google Earth isn't supported by your browser yet. Try this link in Chrome instead. If you don't have Chrome installed, download it here. Learn more about Google Earth." And so there's a few links, and they're obviously trying to move you over.

So Eric Lawrence, who is Microsoft's product manager for Edge, explained on a Twitter thread that the issue arises from the fact that the Chromium-based Edge browser does not ship with the Portable Native Client. That was the PNaCl. Remember there was Native Client, and then we went to Portable Native Client. That's the architecture, the Portable NaCl, PNaCl, is the architectural neutral version of Native Client, which was being used by Google when they converted Google Earth into a web app back in 2017. Google's plan is to move Google Earth over to the universally agreed upon now replacement for NaCl or PNaCl, which is WASM, WebAssem. But Google's running a little behind schedule on that project.

Jordon Mears, whose title is Lead Manager for Google Earth on Web, said: "It has always been our intention to bring Earth to as many people as possible, on as many browsers as possible. Parallel to our efforts in building the first iteration of Earth on Web, Google and the Chromium team have been active participants in the W3C process for WebAssembly." He continued, saying: "WebAssembly also has the advantage of being supported by the four major browsers: Chrome, Edge, Firefox, and Safari."

Okay. So what's a little weird is it's not yet supported by - it's not officially released yet for Edge. He said: "So since the April 2017 launch of Earth on Web, the Earth team has been working to port Earth on Web over to assembly language from Native Client." Yet they haven't yet finished that, despite the fact that a demo of Google Earth running in WebAssembly on Chrome, Firefox, and Chromium was presented by Google's leader of the Chrome Web Platform Team back at the Chrome Dev Summit in 2017.

So anyway, it's taking them longer than they planned. And I presume that it means that Microsoft will be waiting a while for Google Earth to be fully finished for operation on WebAssembly, and that they will not bother with the effort to bring the Portable Native Client (PNaCl) up on their version of Chromium Edge, since it's already a deprecated technology, which would sort of just be a throwaway. So it wouldn't really make any sense for them to do that.

And yesterday, while we're on the topic of Edge and browsers, we're currently in the middle - am I getting my months wrong? Are we in the middle of Microsoft's three-day Build? Or was that a month ago?

Leo: No, it's going on right now.

Steve: That's what I thought. Yes, May, right, right. So it was 6, 7, and 8. So we're in the middle of Microsoft's three-day Build 2019. And Microsoft's CEO, who we know is Satya Nadella, kicked off the 2019 conference with a keynote that touched on Microsoft's plans for its much-anticipated Chromium-powered Edge web browser. Among the list of goodies, the things that I pulled out that would be of interest to our listeners, Microsoft plans to have updated privacy tools. They'll be acquiring some of the new privacy-centric features to help users understand how their data is being used by sites across the web and to provide some controls over various features of the browser, including adding the innate ability to block trackers.

The privacy panel looks like it will allow users to select among different levels of information sharing, probably not being super granular the way Firefox's content blocking feature is, but offering users sort of a little more easy to use, you know, how much content blocking do you want to apply? There will also be, as we touched on briefly, an IE11 mode for enterprises that have software that won't run under Chromium that was written just specifically to Internet Explorer 11. So you'll be able to run the Microsoft Edge browser in IE11 mode in order to accommodate that need.

And then there will be advanced developer tools. Microsoft Edge will allow sort of a common set of tools for web content, progressive web apps, and WebView for developers. So anyway, I'm excited that I'll be able to get that on Windows 7 to go alongside Firefox and Chrome. That'll be nice.

And one quick note. I have been talking about, and I can't think of the name of it now, that really cool, there it is, Filemail, the Filemail site, and bullish about it because it was nicely built and offered an apparently free service. Well, I ran into, a couple days ago, a stumbling block with it that I wanted to note, and that is that they don't say anything anywhere until you hit the block. But you're only able to upload two files per day per upload IP. And there was something I was sending someone where I ended up needing to send them a third thing. Had I known, I would have put it all together, but I just didn't know. And so they said, no, sorry, can't do it unless you purchase their plan. And fortunately I remembered Firefox's Send system, which is free and has no such limit.

So I just wanted to just put that back in our listener's radar one more time: send.firefox.com. Without identifying yourself in any way to Mozilla, like if you're

otherwise a completely committed Chrome user, and you have no Mozilla account, you're able to create an identity with Mozilla. I use it for synchronizing Firefox across multiple systems. But without doing that, you're able to send files up to a gig. If you sign into Mozilla, then the limit increases to 2.5 gigs. So it's a bit of an incentive, if you need to move really large things, to identify yourself to them.

But I just wanted to remind our listeners: send.firefox.com. It is point-to-point encryption. It encrypts in the browser. It uses a technology such that Firefox is - no man in the middle is able, including Firefox as the intermediate, is able to decrypt it. You obtain a link that allows someone at the other end to download it, and then they're able to get it, with their browser doing the reciprocal decryption upon receipt. So anyway, nice service from those guys. And I just wanted to put it back on our listeners' map.

And Leo, that brings us to our subject, Post-Coinhive Cryptojacking.

Leo: Time to talk about Coinhive. We got the replacements for Coinhive. Everything you can use if you don't want to use Coinhive.

Steve: Actually, Leo, you should bring up the site that I ended up finding. Let's see, where did it go here?

Leo: Is this through Malwarebytes?

Steve: No, it's at the bottom of page 15. And our listeners can, too: www.webminepool.com. Yes, we have a service for you if you are missing Coinhive.

Leo: It's not popping up.

Steve: Oh, really, www.webcoinhive.com.

Leo: WebMinePool.com.

Steve: Yeah, WebMinePool.com. Interesting.

Leo: You killed it, Steve.

Steve: You may have some protection in place. I was able...

Leo: Oh, you know what? Of course I do. That's why I can't get there. We've got Cisco Umbrella sitting on us and our Sophos.

Steve: WebMinePool.com.

Leo: See if you can get there because I bet you that's our - we have all, you know, you know Russell. We've got the Sophos.

Steve: Yup, came right up for me. We'll see what the chatroom says for www.webminepool.com.

Leo: Russell's keeping me from going there, and that just shows you, that's probably Cisco Umbrella, one of many secure - I shouldn't say what our security stuff is. That's probably keeping us from going there.

Steve: Yeah, there's sort of a cool end-user thing. You can check out your current hash rate. You click on it, and it'll show you the rate at which your computer is able to mine. You can increase the number of threads which are mining and the percentage of your CPU that you want it to take up.

Leo: I kind of prefer that our employees don't do cryptomining on our hardware and network. Because, you know, this is a personal thing, you know.

Steve: Okay. So what happened after Coinhive shut down, aside from that. As we previously discussed, and as I said at the top of the show, the highly controversial Coinhive browser-based Monero coin mining facility voluntarily closed its doors. Actually it was two months ago. I said a month ago. It was on March 8, so it has been nearly two months. At the time, it was expected that the vacuum would quickly be filled with alternative mining solutions. So what has happened?

Interestingly, the head of Threat Intelligence at Malwarebytes, Jerome Segura, decided to answer that question and put up a blog posting at Malwarebytes titled "Cryptojacking in the Post-Coinhive Era." And he wrote: "September 2017 is widely recognized as the month in which the phenomenon that became cryptojacking began. The idea that website owners could monetize their traffic by having visitors mine for cryptocurrencies in their browser was not new, but this time around it became mainstream, thanks to an entity known as Coinhive."

He wrote: "The mining service became a household name" - and I was thinking, okay, I'm not sure whose house he's in, but it wasn't a household - I'm not sure that Coinhive was a household name, but within certainly our podcast listener community. "Became a household name," he wrote, "overnight, and quickly drew ire for its original API, whose implementation failed to take into account user approval and CPU consumption. As a result, threat actors were quick to abuse it by turning compromised websites and routers into a large illegal mining business."

He writes: "The ride was wild; but, as we came to see, short-lived, as Coinhive shut its doors in March 2019 following months of steady decline and loss of interest in browser-based mining." And I'll interject here, as I said, that as we talked about at the time, it was largely a function of the general collapse in the cryptocurrency valuations from their highs of a few years before that, which did a lot to take the wind out of the market for monetizing the arguably stolen CPU cycles of innocent web users. Remember that the browsers responded and started fighting back, and things like uBlock Origin and the like - that's interesting. uBlock did not block WebMinePool. Because I've got it running. It shows a block of something on my browser, but not that.

And as I said at the time, and I say now, with user permission, with a user's knowledge, I think this is an interesting idea for monetizing, for some sites that do it responsibly and want to monetize the time that their users spend. We've talked about ad blockers. We're seeing some pushback against them. From time to time you get a pop-up saying, "I see you're using an ad blocker. Would you please consider supporting the site?" Well, if using my processor while I'm rummaging around was an option to give revenue to them that they sanctioned, it's like, okay, as long as everybody agrees.

So anyway, what the Malwarebytes guy wrote, since what they have is they have a presence in many people's machines, and they are doing real-time traffic analysis in order to protect the machines, he wrote: "Interestingly, we still detect thousands of attempts for Coinhive-related domain requests, even though the service announced it was shutting down on March 8th. Over the past week, our telemetry recorded an average of 50,000 attempts per day. Digging deeper, we see that a large number of websites and routers have never been cleaned [of course], and the bits of JavaScript requesting the Coinhive library are still there. Evidently, with the service down, the necessary WebSocket that sends and receives data between client and server will fail to connect to the server, resulting in zero mining activity or gain."

Yet it's very much like Code Red and Nimda; you know? You put a sensor out on the 'Net, you still get Code Red and Nimda pings because there are still some of those viruses, those worms, still alive, trying to reproduce. Similarly, it's going to be a long time before every router that has a resident copy of a Coinhive miner in it gets rebooted and then doesn't get reloaded into RAM, presuming that it wasn't made persistent.

So this kind of stuff, these things have a very long tail before they finally go away. And for anyone who's interested, I have a picture from February 27 through the end of April. It's relatively, you know, it's declining from its peak. But the decline looks like it's leveled off to like those things that are going to probably be around for a long time.

And so he poses the question in his blog: "Is cryptojacking still a thing?" He says: "To answer that question, we go back to the early adopters of browser-based mining, which were torrent sites. Visiting a proxy for The Pirate Bay with our browser, we spot something familiar enough. Our system's CPU usage maxes out at 100%. So," he writes, "yeah. Apparently in some corners of the web cryptomining itself remains alive and well."

He says: "As we'll recall, this is what started the cryptojacking trend back in 2017 when users weren't told about this code running on their machine, let alone that it was hijacking their processor for maximum usage. In this instance," he writes, "the mining API was provided by CryptoLoot, which was one of Coinhive's competitors at the time. Malwarebytes reports that while they are seeing nowhere near the same level of activity as they saw during the fall of 2017 and early 2018, according to their telemetry, they are still blocking more than one million requests to CryptoLoot each day." So that says there is still an effort by mining to get its script loaded into people's systems for the purpose of mining.

There are a few other services out there, and it's worth mentioning CoinIMP, which, he writes, they've seen used more sensibly on file sharing sites. And he says: "Router-based mining still going. While the number of compromised sites loading web miners was going down in 2018, a fresh opportunity presented itself thanks to serious vulnerabilities affecting MikroTik routers worldwide. By injecting mining code from a router and serving it to any connected device behind it" - we've talked about this in the past where, for example, if you were behind an infected router, and you went to an unencrypted landing page like your ISP's DNS intercept, where it said, oh, the site you're looking for isn't available, brought to you by Cox. How about some of these? That would immediately pin your system's CPU because the router you got that message through took advantage of

the fact that the page you received was not encrypted in order to insert its JavaScript into your page header.

So he says: "By injecting mining code from a router and serving it to any connected devices behind it, criminals could finally scale the process so it was not limited to visiting a particular website, therefore generating decent revenues." On the other hand, they could only do this for any non-HTTPS page that the browser loaded. Once upon a time, 10 years ago, that would have been very lucrative. Now, not so much because pretty much everything is encrypted.

He says: "The number of hacked routers running a miner has greatly decreased. However, today we can still find several hundred that are harboring the old, now inactive Coinhive code, and have also been injecting a newer miner." Okay. So this brings us to WebMinePool.com, meaning that naturally it is being used for malicious purposes. WebMinePool.com is now being injected into people's systems by compromised routers.

And so anyway, he concludes with "Campaigns gone missing: Perhaps the biggest change in cryptojacking-related activity is the lack of new attacks and campaigns in the wild targeting vulnerable websites." He says: "For example, in the spring of 2018, we saw waves of attacks against Drupal sites where web miners were one of the primary payloads." And in fact we've addressed the observation that that's changing now previously. He says: "These days, hacked sites are leveraged in various traffic monetization schemes that include browser lock, fake updates, and malvertising. If the content management system is Magneto or another ecommerce platform, the primary payload is going to be a web skimmer." And we know that those are trying to obtain the user's credit card information as it's submitted to an ecommerce site.

He says: "We might compare cryptojacking to a gold rush that didn't last too long as criminals sought more rewarding opportunities. However, we wouldn't rush to call it fully extinct. We can certainly expect web miners to stick around, especially for sites that generate a lot of traffic. Indeed, miners can provide an additional revenue stream that is as concluded in this Virus Bulletin paper" - and they have a link to it that says - "dependent on various factors including, of course, the value of cryptocurrencies which historically has been volatile."

So I think that's where we are. The Coinhive died. That was the preeminent cryptomining tool. Everybody was using it. There were also-rans. They now have a larger share. And then there's this WebMinePool, which you guys in your network, Leo, you're probably glad are unable to access, thanks to your protection. It's alive and well. So I think, just like all of the worms that have existed in the past, the malware that's out there, there's, like, a place for and a diminishing presence of cryptojacking. And it's never going to go away. It'll sort of fade into the past. It doesn't make the money that it once did, so the world has moved to things like the ransomware in order to attempt to get into people's systems and extort money.

So anyway, sort of a podcast all about web browsers this week, just as a consequence of how much browser stuff is in the news. And of course we know that it makes sense for us to talk about that because that is the surface that we visit the Internet via. It is the attack surface that we expose. So we've got to keep our browsers secure.

Leo: It's better than spending every week talking about Adobe Reader, so that's - there's improvement.

Steve: Yeah, there has been some drift, hasn't there.

Leo: There's been a little drift. Well, that's what you see. You see how they started securing Windows pretty well, so the attacks moved to Adobe, Flash, and Reader. And now I guess, I think probably it's not that Adobe's secured it so well as that most people just stopped using those; right? Flash is dead, and you don't even need Reader if you have Edge. So now, well, let's go after the browsers. Of course, makes perfect sense, yeah. Although I think all the browser companies are doing a pretty good job of securing them. So you think making Edge Chromium-based will make it easier to secure?

Steve: Yes. And I really would love to know what's behind that. Have you and Paul and Mary Jo, like, have they provided some illumination into what's the thinking behind the scenes?

Leo: Oh, yeah.

Steve: Why, after this huge investment in their own HTML engine, they said, uh, okay.

Leo: No, and in fact they've talked more about it at Build this week. And it's pretty clear that the issue was Edge adoption was really lagging. And it just makes more sense to unify behind a single engine.

Steve: Yeah.

Leo: And Microsoft doesn't - it's a very different Microsoft. I don't think they care about proprietary lock-in as much as they used to.

Steve: Wow.

Leo: I know.

Steve: I'm taking a while to get used to that change.

Leo: You saw they're putting a Linux kernel in Windows.

Steve: Yes.

Leo: Did you see that?

Steve: A full Linux kernel, yeah. And I thought it was a joke. Someone tweeted it to me this morning. They're jumping up and down because Windows is going to have a console with tabs.

Leo: Oh, yeah, a nice terminal, a tab terminal.

Steve: I mean, I thought it was a joke, like wait a minute, what?

Leo: No, it's such a different Microsoft than the one you and I are used to. And I think that's a good thing. They're smart. I mean, it's turned them into trillion dollar company. I mean, they're clearly - the market's embracing what they're doing, and I think it's smart. And this is just part of it. Why spend a lot of energy on something really nobody was using? I mean, it was in a single-figure percentage adoption, even though it was the default browser.

Steve: Wow.

Leo: I know.

Steve: So a deliberate attempt to say, okay, I don't want this, I want Chrome.

Leo: Yeah. Or something, yeah.

Steve: Wow.

Leo: So, yeah, I think it was the right thing to do. And you know there's still a lot of questions. I think Microsoft's smart to build an Internet Explorer mode in, so that way they can eliminate that additional install plus serve all those people running legacy code that runs in IE8 only, you know. But I also think that they - I think it's a different company. They used to say, well, the big change was first we want to be everywhere our customers want to be. Nadella would say this. But the best experience will always be in Windows. Now they don't even say that. They've moved on.

There was a - I'm trying to find it, but I don't see it. But it's worth searching for, if you're interested, an interview I think Nadella or the Edge team gave about the decision, this last couple of days at Build. And it makes sense. We did talk about it on Windows Weekly.

Steve: Yeah, cool.

Leo: Steve, we done. Thank you for a very compelling and interesting show, as always. People who are interested in this matter, and I think we all are, should tune in every Tuesday, around about 1:30 Pacific, if we don't have a Google I/O to get in the way. That's 4:30 Eastern, 20:30 UTC. The live streams, audio and video, are available at TWiT.tv/live. You can also ask your voice device, you know, if you ask your Echo, listen to TWiT Live, it'll start playing whatever's going on. Or you could say listen to Security Now! podcast, and it'll play the most recent version of the Security Now! podcast. For some reason, I don't know, I'm a little weird, I say "listen." Most people say "play," like they're commanding it to play. For some reason I think "I want to listen to."

Steve: Yeah.

Leo: It's just me, I guess. I'm weird. So say "play," it's okay, that works, too. You can also download a copy of the show from Steve's site. So you have a regular 64Kb version, but you also have a super small MP3 for people who don't want to waste bandwidth.

Steve: Actually, no. I have...

Leo: You used to have a 64Kb version; right?

Steve: Yes. I have the super small one. But the standard size one just links to yours.

Leo: Right, right.

Steve: And so it's just a convenient length for that. And I do bounce through Podtrac, even for the small download, so that we get credit for the fact that the podcast was downloaded.

Leo: Thank you. We asked you to do that, and I really appreciate it, yeah, because we want to make sure everybody knows how many people listen, including our advertisers.

Steve: Yeah.

Leo: GRC.com. He does have something that you can't get anywhere else. That's those great transcriptions written by Elaine Farris. So if you like to read along while you listen, and I know for a lot of people it helps in the comprehension, GRC.com. While you're there, get a copy of SpinRite 6, the industry's number one hard drive recovery software. It's good for maintenance, too. That's Steve's bread and butter. Everything else there including ShieldsUP! is free. I'm just looking at the site: 102 million shields tested, and counting.

Steve: Yeah.

Leo: It's kind of amazing. GRC.com. You can leave feedback for Steve or questions at GRC.com/feedback. Or tweet him. He accepts direct messages at @SGgrc. That's his Twitter handle: @SGgrc. We have audio and video of the show at our website, TWiT.tv/sn. And of course the best thing to do would be subscribe. Then you don't even have to think about it, but just get the new Security Now! the minute it's available very Tuesday evening.

Steve: Yup.

Leo: Thanks, Steve. We'll see you next time on Security Now!.

Steve: Thanks, Leo.