Update Exim Now!

Description: This week we catch up with the continuing antics of SandboxEscaper. We give an update on the status of the still-not-yet-widely-exploited BlueKeep vulnerability, and also look at a new botnet which is pounding on RDP servers (but not yet using BlueKeep). The FBI has issued an interesting advisory about not trusting secure sites just because they're secure, so we'll examine that. The popular VideoLAN player receives an important update thanks to an interesting source, Microsoft's Edge browser takes another step forward, and Mozilla reorganizes a bit. Then I'm going to share my must-have Utility of the Week, a just-released sci-fi movie on Netflix, and a bit of closing-the-loop feedback from the Twitterverse which resulted from my, as planned, first formal full release of SQRL. We'll close with a look at the critical need for anyone running the Exim mail server to update immediately.

High quality (64 kbps) mp3 audio file URL: http://media.GRC.com/sn/SN-718.mp3
Quarter size (16 kbps) mp3 audio file URL: http://media.GRC.com/sn/sn-718-lq.mp3

SHOW TEASE: It's time for Security Now!. Steve Gibson is here. SandboxEscaper is back with, yes, yet another zero-day. Not too much to worry about. The NSA has an advisory. And it's the release of SQRL. What? It's all next on Security Now!.

Leo Laporte: This is Security Now! with Steve Gibson, Episode 718, recorded June 11th, 2019: Update Exim Now.

It's time for Security Now!, the show where we cover your security online and your privacy, how things work, how things don't work more often.

Steve Gibson: Yeah.

Leo: With Steve Gibson. He's right there. He's the man in charge at the Gibson Research Corporation, @SGgrc, the guy who discovered the first spyware. That's how old he is. Actually, that wasn't - how long ago was that, Steve?

Steve: That was a long time ago.

Leo: Ten years? Twenty years?

Steve: I think it was before the podcast, and we're closing in on the end of year 14.

Leo: Wow, wow.

Steve: So that was OptOut, which is what I called this thing because it was advertising. I guess it was spyware. Well, it was spyware because it was in people's machines. It was watching what they did. It was profiling - yeah, okay, it was spyware.

Leo: It was spyware. Anything that does that is...

Steve: Yeah. And they didn't ask for it. The idea was it was a way of monetizing freeware so that you would show ads in a window on like the UI of freeware. And, oh, boy, I mean, it caused quite a ruckus when, you know, what happened was I was experimenting with an outbound firewall with - shoot, I just had it, and I just lost it, my favorite firewall back then. Well, it has been a while. Anyway, it was the one that did outbound blocking. No other firewalls did that.

And so I set it up, turned it on, and then up popped a notice saying that something, Aureate or Radiate or something, wanted to connect to the Internet. And back then, those were still the days when you sort of had some idea of what was going on in your computer. I mean, we've lost all hope of that now. But it was like, wait a minute. What? I didn't put that in my computer. So I tracked it down, and I found out what it was.

Oh, ZoneAlarm is the name I was trying to come up with, the ZoneAlarm firewall that allowed you to do outbound control for exactly this reason, because something could be in your machine that you didn't know about. And of course now, again, as I was saying, actually I have a recommended Free Tool of the Week that's going to give our listeners a little bit more control over this. I think everyone's going to get a kick out of it.

But this is Episode 718 for June 11th. And we were talking before the show. There is the number one most popular email server on the Internet, meaning the server that is running on machines deliberately connected to the Internet, facing the Internet, to transfer email, an email server. Once upon a time it was Sendmail in the early days. There have been several successors since then. What is now the most popular one, it think that the number, I have it in the show notes, I think it's 57% of all email servers on the Internet are Exim, E-X-I-M.

And they're all, well, I can't say "all." But unless you've updated that email server in the last month, if you had any version in the last three years, they are all vulnerable - and we're talking, I think it was 570,000 of them - to remote command execution. Not remote code execution. We have to do some fancy return-oriented programming or overflow a stack or bust, you know, like have a buffer overrun or something. This is a means of actually executing Unix or Linux commands as root from, like, remotely. So this is really bad.

The reason, if you'd fixed it in the last month, you're not affected, is that they did a new release, and they didn't even know they had a problem. They didn't know they had this problem, and they fixed it just kind of, oh, by the way. But it's been there since 2016, for more than three - I think it was April of 2016 - more than three years. So anyway, that's going to be our main topic.

We've got a bunch of stuff going on in the past week. We're going to catch up with the continuing antics of SandboxEscaper. We also update on the status of the still-not-yet

widely exploited, but expected any moment now, BlueKeep vulnerability. We also take a look at a new botnet which is pounding on, unfortunately, RDP, the Remote Desktop Protocol servers, but not using BlueKeep to do so. It could, but that's not what it's doing. It's probably BlueKeep just is too new.

We also have the FBI issuing an interesting advisory about not trusting secure sites just because they're secure. Which I thought was really interesting, and we'll talk about it because it's sort of - it tells us that, you know, the FBI gets all these calls. They're understanding what people's problems are in some ways differently than, well, than I am. Leo, you also get those calls on the weekend.

Leo: Oh, yeah, absolutely.

Steve: So you probably know. But it's like the world, like the general population is about a decade behind. So we'll talk about that. Also the very popular VLC, the VideoLAN player, received an important update thanks to an interesting source, that we sort of foretold about at the beginning of the year. We'll come back to that. Also Microsoft's Edge browser takes another step forward. Mozilla reorganizes themselves a bit. I mentioned also before we began recording, but now I'll say it officially, I have a must-have Utility of the Week for Windows users which I went searching for when I was sort of reminded of something that I'll explain, that I think everyone's going to get a kick out of. I want to share with you a bit, without any spoilers, about a just-released on Friday really good sci-fi movie on Netflix.

Leo: Oh, okay.

Steve: And I want to warn our listeners, I'll just say it right up front, do not watch the trailer because I was very disappointed in how much spoiler was in the trailer. So we'll tell everybody about it, but don't watch the trailer.

Leo: Okay.

Steve: It's just there's too much there.

Leo: I hate it when they do that.

Steve: It was so annoying. It was like - and as the trailer was spooling itself into my brain I was thinking, no, no, no, I don't want to be seeing this now.

Leo: Don't tell me, don't tell me, don't tell me. Oh, that's too bad.

Steve: Yeah. Anyway, so we've got that. We also have a bit of closing-the-loop feedback from our Twitterverse which resulted from my, as planned, first formal release of SQRL, which has happened.

Leo: What?

Steve: Yes.

Leo: You buried the lead, dude.

Steve: Well, yes. Just my last five years.

Leo: Only five, yeah, six years in the making, okay.

Steve: Then we'll close with a look at the critical need for anyone running, or anyone who knows of anyone who is running, the Exim mail server to update immediately.

Leo: Yikes. Yikes, yikes, yikes. Well, lots to talk about.

Steve: The Picture of the Week, I didn't have anything more pressing, so it is the first page of the 17-page document, which is finished, as the beginning of the formal, the full SQRL specification.

Leo: Wow.

Steve: And so it's called "SQRL Explained." And it's written sort of to be a - it's not a user manual because the goal is that SQRL doesn't need a user manual. I mean, if you just use the app and take a moment to read the screens, it just tells you what you need to do.

Leo: So you don't even need a user manual. But this is for people who conceptually want to understand it.

Steve: Yes. Somebody who read this would come away understanding everything that SQRL is: how it's able, for example, if your identity got stolen, how it is able to prevent bad guys from changing it, and how you are able to get it back, to take back a stolen identity; how in a two-party system where there's no one to say, whoops, I forgot my password, how we perform password recovery without anyone to ask; and all of the different components that are part of the system.

And so, you know, GRC.com/sqrl.htm is a page where there are only two things. There is that document, the 17-page PDF; and a link to the client because I also released version 1.0 of the SQRL client for Windows, which is sort of the reference client. The other clients - there's one for iOS and Android. It also runs under Mac and Linux with Wine. But there's also a Firefox extension that runs with Chrome and Edge. And they're not as feature complete because they're sort of following along behind. But anyway, it is there.

There are, as I said, more than 1,300 accounts now over in the SQRL forums. So there's an active community of people who have been playing with this and helping me to test it and know it really well, if anyone has any questions. And so this is what I've been waiting for almost six years...

Leo: Oh, man. This is so exciting, Steve.

Steve: ...to tell everyone about, yeah.

Leo: Yeah. And there'll be one more landmark: the first site to use it; right? The first publicly...

Steve: Yes. That would be, well, now, yes. The SQRL forums, of course, don't really count because they're mine. And so of course, thanks to Rasmus Vind, who is the PHP coder who I worked with in order to add SQRL support to the XenForo forum software, of course we have SQRL login there. And I also have some other demo servers and sites that people can use. There's one written in Java. Jeff Arthur, who's doing the iOS client, he has one. So there are sort of test servers. But yes. From the beginning there's been a strong enterprise interest, I think because large enterprises need some way of managing their employee identity that makes sense to them. And so I got a lot of inquiries in the beginning.

My next piece of work is to follow on this 17-page explainer, you know, the idea is that this would be useful for people who want to understand what SQRL is sort of at the higher technical level. There's no crypto, but there's some pretty block diagrams, and I explain things the way I do the podcast so that I think people will be able to probably enjoy learning something at the same time. But then what follows will be the so-called "on the wire" protocol, that is, how you encode the data, base-64 URL, what data you encode, what you put where and so forth, so that somebody would be able to take that and actually create implementations of this that would be compatible with all the clients and all the servers that exist so far in order to start making this actually happen.

So anyway, we'll talk about this a little bit more down in our closing-the-loop section because I tweeted this yesterday, or maybe it was this morning, and got a couple pieces of feedback that were exactly what I would expect to have because there's something of a mixed blessing here that is just the case, which is there is - and I've mentioned this before. There is more upfront investment required than with usernames and passwords. Really not much more than with a login manager because this is sort of a variation on that.

But it's true that you sort of, in a sense, you get what you pay for, except that with SQRL all of the payment - well, first of all, it's free. But I mean payment in terms of effort. It's just getting it set up once. Once that's done, then you get to reap the benefits of this updated architecture and approach, potentially forever. And then all subsequent logins are like way easier than any other solution provides. But it is the case that, with usernames and passwords, there's nothing for you to do except invent a password.

On the other hand, you have to then have a good password. It has to be a different password for every site. You have to somehow memorize the password or record it or write it down. And then when the site gets breached, you have to change the password and blah blah blah blah blah. And of course also it may not be very secure if you don't do all those things. So even though there isn't much investment asked of you if you use a username and password, you also have ongoing pain forever as you use that. And SQRL eliminates all of that.

So anyway, we'll talk about this a little bit more in our closing the loop. And for what it's worth, this was a major milestone in this multiyear project to just basically create and propose a solution. I have no means to make the world accept it. The goal is that, or my hope is that it'll sort of seep out. Over time it will be understood.

For example, all we need is a WordPress plug-in. No one has written one. Rasmus wasn't interested. When someone does a WordPress plug-in, then suddenly, as we have said, more than half of the websites on the Internet could use this to allow their users a much - their visitors a much easier way to log in, if you wanted to post a reply to someone's blog, than having to create an account. And so there's some incentive, you could imagine, for offering something easier. And then when people kind of get used to that, it's like, wait, why doesn't everyone do this? Well, yeah, exactly. So anyway...

Leo: People are saying, why doesn't TWiT implement it? I would, except we don't have user accounts, so there's no login on TWiT.

Steve: Yeah. It's like GRC. I have no accounts at GRC.

Leo: You can't use it, either; right? Yeah.

Steve: But we do in the forums so people will be able to experience it.

Leo: Forums it makes a lot of sense, yeah.

Steve: Makes a lot of sense.

Leo: Yeah. And I would love to see a big name company like Twitter or Facebook or Google implement this just as a secondary way of logging in. And that's what you're going to need, I think, is somebody like that saying okay. I can see enterprise jumping on it.

Steve: Well, yeah. And because it's free...

Leo: That won't be public facing.

Steve: ...it's not difficult to implement. And, well, and that's the other cool thing is imagine that enterprises, some enterprises did. Then their employees would have created a SQRL identity, and they'd have a SQRL client on their various laptops and computers, and maybe on their smartphones. And they would be able to use that for their enterprise. But then if they happened to go to a website that offered SQRL, there's nothing more they need to do. They literally click "Sign in with SQRL," and they're done.

So, and it's just - it's so right when you start using it. I mean, and of the people who have experienced it, they're like, oh, this has to happen. We have to make this happen. What can we do to make this happen? And it's like, I know. We just, you know, one step at a time. It had to exist first. Finally it now exists. And it took long because it's really nailed. I mean, it is, as I've said before, and as I say when I talk about it, I have an answer, an answer exists for every possible "but what if." But what if this? But what if that? It doesn't matter what you ask that we have an answer for it.

And in fact there is a "what if" page at the SQRL forums, at sqrl.grc.com, that is that. I compiled a list. I solicited "what if" questions from everybody in the forum. If you scroll

through that, there is no "what if" that wasn't asked, and there's an answer for every possible thing that could go wrong. So anyway, obviously I'm a little excited, but this was a big milestone. So it is there at grc.com/sqrl, or you can do a .htm, /sqrl.htm. Again, you'll find the downloadable 17-page PDF and a link to the client. Which is, I think, 278K. It's multilingual.

Leo: Wow.

Steve: I know. It's so big, I hated it, Leo, because there were some...

Leo: You know hard drives come in gigabytes now, terabytes; right? You know that; right?

Steve: There were some crypto libraries that just didn't make sense for me to write from scratch, so I thought, okay, fine, I'll just add this big blob to my code. So it would have been a lot smaller, if not. But there were some places it did not make sense for me to do from scratch. So, yeah, it is cute and tiny. It checks in for updates and auto updates itself. In fact, it auto installs. There's no separate installer. You just run it, and it goes, oh, I'm not here yet. Would you like me to stay? And you say, oh, yes. And anyway, I think our listeners will get a kick out of it. So it's there. It exists. And onward.

Again, my next piece of work - I know everybody's waiting for SpinRite. I've just got to get the - basically the specs are written, scattered across some old web pages, and some that I have been keeping updated for the other developers who've been writing pieces. I just need to pull it all together. So that's the next phase is to finish this document that I have started with the full, over-the-wire implementation spec. And then it's time for SpinRite. So we're getting there.

Leo: Nice. Congratulations.

Steve: Thank you. SandboxEscaper dropped another zero-day. And the good news is we don't have to worry about this one because it's kind of a crock. It's a second bypass for a problem that Microsoft kind of already patched last April, a couple months ago. Microsoft described the problem they were fixing as: "An elevation of privilege vulnerability exists when Windows AppX Deployment Service improperly handles hard links. An attacker who successfully exploited this vulnerability could run processes in an elevated context. An attacker could then install programs; view, change, or delete data."

Then Microsoft said: "To exploit this vulnerability, an attacker would first have to log onto the system," meaning they would have to be local. "An attacker could then run a specially crafted application that could exploit the vulnerability and take control of an affected system." So, you know, it's a privilege elevation flaw. You have to be there. You've got to be able to run something. But when you do, you can break through Windows' what appear to be relatively soft barriers. And it's unfortunate that that's the case. In fact, she indicated that she's just finding these LPEs, these Local Privilege Escalations, everywhere.

So what we have is another - there was one way to bypass it that she produced a couple weeks ago, and now she has another. But what I'm beginning to think, based on the way this looks and from the quality of her previous work compared to this, this feels like, as I mentioned, sort of a crude hack that she stumbled upon while working towards a

different exploit. So I'm not going to go into great detail because I don't think it matters. It's kind of a funky way of obtaining this elevated privilege that involves deleting a subdirectory of Microsoft's Edge browser files, then launching Edge twice. The first launch, without these files present, results in Edge crashing. So it's, again, inelegant. Then the second launch, if it's done programmatically by clicking Edge's quick launch icon down in the start bar or the tray, causes a mistake in setting of Windows Access Control, which could then theoretically be exploited.

So, you know, it's really a mess. She's in her typically grumbly mood, as she so often is. And my guess is that she couldn't sell this. The guys at 0patch, you know, the micropatch guys, told Threatpost, who had some coverage of this bug, that it wasn't even critical enough to warrant one of their little micropatches, and that they had been unable to reliably reproduce it. They wrote: "We know of no one being successful at it," and they said, parens, "(it could just be really difficult to reproduce, or depending on some external factors that were not present in our testing environment)."

So this happened late last week. And of course today is Patch Tuesday for June. This is the second Tuesday of June. So certainly I haven't looked at what Microsoft did for today, but I would be very surprised if they had time or probably even sufficient concern, really, to look into this one and fix it. And her proof of concept code has once again been removed from GitHub. If you go to SandboxEscaper's account over on GitHub, it's wiped. I mean, it's got, like, submissions and downloads and blah blah blah blah, all the various - there's like six or seven different categories. I looked because I was curious yesterday, and there's just nothing there because Microsoft keeps cleaning it up. Just like, you know, of course they took over GitHub. So they just keep - she posts things. I don't know why she posts them there. But they just get wiped clean. So there's nothing there at the moment.

Anyway, it feels to me like this is such a crappy hack, and she is so good, I mean, I've complimented her in the past. Last year when we saw some of these earlier ones, they were like, whoa. This is a gifted hacker who is producing these things. And given what we have learned from her recently, that there is, you know, she mentions wanting to be able to sell these for $60,000. That's the number that she keeps using.

My guess is that we're not seeing her best work; that her best work, presuming that she is continuing to produce at that quality, is being sold to people with deep pockets and who have a need for high-quality working Windows exploits. And so what we're seeing are kind of debris that doesn't really make the grade, that she's unable to monetize, so she dumps them on GitHub just to watch Microsoft come along and sweep them up and scrape them off.

Anyway, and on the second page of the show notes I have a posting of hers from 5:19 this morning. She says: "This is the second bypass" - or I guess it was, like, not this morning. It was late last week. "This is the second bypass for CVE-2019-0841. I won't ever be part of the infosec community, people have made that clear to me a long time ago, in many different ways. I'm just a harmless crazy person now, sharing harmless LPEs. Whatever. Bye. P.S.: I have one more zero-day." Then she says: "*growls and wanders off into the arctic*." So there's an update on our SandboxEscaper.

As for updates, we're still watching BlueKeep. There was a newly published, fully working proof of concept exploit developed as a new Metasploit module for BlueKeep. And, now, I did say that it was fully published, except that I also know that he said he plans to keep the module private. So I guess it is an existent Metasploit module which has been demonstrated. It was developed by Zerosum. And unfortunately I don't have his actual handle in the show notes because the "e" and "r" of Zero spelled out, Z-E-R-O, they're actually upside down in his handle. And I didn't want to bother worrying about how to do whatever character-set Unicode was necessary to faithfully reproduce his handle.

But anyway, he is a well-known, respected reverse engineer. He's the guy who did the first non-destructive, that is to say, non-crashing probe for the BlueKeep update. And that's the code that Robert Graham at Errata Security then based his scan, his RDPScan on, in order to scan the whole 'Net that we talked about last week and the week before.

So now what we have, this new thing, which is it exists, but it is not - it's still private. It demonstrates how an unauthenticated attacker can achieve full access to a victim machine in about 22 seconds. What's different about this is it bypasses the one mitigation that Microsoft and the FBI and the NSA and everybody else has been saying would protect you, which is this NLA, this Network Level Authentication. He worked around that.

What he did was he combined a different Windows credential harvesting tool that we talked about a long time ago, Mimikatz, M-I-M-I-K-A-T-Z. Mimikatz was created a long time ago, in the early days of NTLM, Microsoft's own NT LAN Manager protocol, to demonstrate some of its security problems. And it's continued to evolve through the years, keeping up with LANMAN as necessary in order to track the changes that Microsoft has made. So by pairing the zero authentication RDP exploit with Mimikatz, he's ended up with an extra potent tool that he is not releasing because he feels it is too potent.

He tweeted: "Still too dangerous to release, I'm sorry. Maybe after first mega worm." Which is to say, after it sort of no longer becomes an issue because all the RDP servers on the planet have been compromised. Then he may let it go. Anyway, so he knows what he's doing. He has advanced the state of the art in this BlueKeep vulnerability. And so far we are still not seeing what people are talking about. And as I said last week, to me I don't think it's going to be a worm. I mean, maybe at this point it will be just because why not; and because everyone's saying that there will be one, someone will create one just to say, okay, yeah, I did one.

But last Tuesday, a week ago, while we were doing Podcast 717, the NSA got into the BlueKeep act, publishing their own security advisory. They said, from Fort Meade, dated June 4th, they said: "The National Security Agency is urging Microsoft Windows administrators and users to ensure they are using a patched and updated system in the face of growing threats. Recent warnings by Microsoft stressed the importance of installing patches to address a protocol vulnerability in older versions of Windows. Microsoft has warned that this flaw is potentially wormable, meaning it could spread without user interaction across the Internet. We have seen devastating computer worms inflict damage on unpatched systems with wide-ranging impact and are seeking to motivate increased protections against this flaw.

"This is the type of vulnerability that malicious cyber actors frequently exploit through the use of software code that specifically targets the vulnerability. For example, the vulnerability could be exploited to conduct denial of service attacks. It is likely only a matter of time before remote exploitation code is widely available for this vulnerability." That I completely agree with. They said: "NSA is concerned that malicious cyber actors will use the vulnerability in ransomware and exploit kits containing other known exploits, increasing capabilities against other unpatched systems."

And again, I think that is what we're going to see. And since this is Patch Tuesday, note that it was last Patch Tuesday that we learned about this. This is when all of the systems going back to even Windows XP, the older systems, got themselves patched, and Microsoft said, "Oh, by the way, there's something really bad that we just fixed. Everybody please update."

And as we know, a couple weeks went by. Robert Graham scanned the Internet, I mean, and really, really, really scanned it, looked at all things that answered on all the various ports that might be typically running RDP, weeded out those that weren't actually RDP.

Of those that were RDP, then further weeded out the ones that were actually vulnerable, and that's where we got the 950,000 confirmed vulnerabilities that aren't going to go away by themselves, no matter how many pleas from people are being made. These systems are just unattended.

I did see some tweet, actually, in the last week, from someone saying that, yeah, we're not happy about it, but we cannot take down the servers that we have, even to fix something this bad. And I'm thinking, what?

Leo: Oh, then I'll take them down for you.

Steve: What are you smoking? Yeah, exactly, Leo. And I guess this guy didn't know about the 0patch, the micropatch, because that micropatch fixes any server that's vulnerable without taking it down. So somebody, if anybody is hearing this and saw that tweet, I saw it go by, and I didn't have a chance at that time to catch it. But that guy needs to be told, and anybody else, you do not have to bring - you don't have to down a server and update it to fix this. You can apply the micropatch from 0patch.com in order to fix it. So do so.

Oh, and an update from Robert Graham. RDPScan, his tool, is now in its fourth release. And so you can find it. I have the link in the show notes. But on GitHub.com it's /robertdavidgraham, G-R-A-H-A-M. That'll take you to his page where you can then find RDPScan. And under the releases there are now four of them. He's just been fixing a couple little minor bugs that have cropped up over time. And there is a downloadable Windows binary which is super useful for scanning intranets to make sure that an internal vulnerability is not leveraged.

Leo: Stevie?

Steve: So there is a new botnet in town, which itself is not news. What's interesting is that this one has decided - and the researchers, the guys that have been tracking it, are not quite sure what's going on. It's in the wild. It is spreading. It is actively searching the Internet for either patched or unpatched RDP servers. Though as I mentioned, it's known as GoldBrute because it's written in Java, and the Java class is named GoldBrute. Who knows why, except that it is a brute forcing bot. Its being written in Java means that it needs to drag an 80MB Java runtime along with it.

Leo: Oh, geez. Aren't you glad you didn't write SQRL in Java?

Steve: Not exactly like, yeah, like my 287K SQRL client that is loaded with text and is multilingual and so forth. Anyway - or multilingual-able. Our listeners will remember that I found a site that would allow us to handle translation. I'm going to hold off on that a bit till we see if there is actually a demand for it in other languages. And, if so, then we will pursue that. All of the foundation is there for that. But anyway, this GoldBrute is, yes, maybe that's why it's called GoldBrute, because it is a brute, needing an 80MB Java runtime in order to run itself.

Anyway, here's what we know. We know that a search using Shodan turns up that original 2.4 million IP addresses reflecting machines that are reachable over the web that have some version of Remote Desktop Protocol enabled. If it's present, if RDP is visible, then we know that about a third of them, because 950,000, about a third of those will be

older versions that you don't even have to guess the credentials because the problem that Microsoft fixed last Patch Tuesday was the ability to just jump right onboard without any credentials. The other two thirds are newer, or patched, and so you do need to provide a logon username and password. You need to provide credentials. Thus you would need to brute force them.

Okay. So apparently the GoldBrute botnet is not discriminating. But as I mentioned, it does not currently avail itself of the RDP flaw. It's using a pure username and password guessing in attempts to break into all of these 2.4 million machines. And as we know from previous reporting, for whatever reason, many of these machines will fall to persistent brute force credential stuffing attacks. We've seen this before. There are lots of servers out there that have malware on them, presumably because somebody was able to guess right and install something.

So GoldBrute's scanning, its own scanning, has turned up 1.5 million RDP-serving machines. The code being loaded does not reveal the final purpose of the hacked Remote Desktop servers, which is to say it appears to only be - it only exists for the sake of its own existence. There is no persistence mechanism, meaning it doesn't modify the file system in any way. Simply rebooting the machine which has been infected by GoldBrute will remove all traces of it. So it exists and it lives entirely in RAM. So one theory is that the botnet is being used simply to compile a credential list of available RDP servers which might then be offered for sale on the Dark Web sort of as an access for service sort of thing.

The researcher who's been following this says there's only one command-and-control server, and he knows this both by watching it work in a honeypot, and also by the fact that it's Java. They've been able to do some decompilation and see what's going on inside. So there's one IP, which is 104.156.249.231. That IP is where all these bots are phoning home to, which is a location in New Jersey. And I was curious, so I did a little bit of spelunking, and that IP is under the control of a cloud services provider Vultr, Vultr.com. Www.vultr.com takes you to their home page, and they're just sort of a - they're just some cloud service provider. And so I guess somebody is renting some space at that IP address and has all of these GoldBrute bots connecting there.

What's there is a port 8333 service which these bots used to connect an encrypted web socket connection to. 8333 is commonly used for bitcoin connections, but this isn't a bitcoin transaction, so that's a little strange, except maybe it's considered now a well-known port, so the person thought, well, I'll just hide my command-and-control server behind port 8333 so I look like a mining pool or something. So it's like, okay.

So what happens is the individual bots scan the 'Net at random, find RDP answering servers, and they report them as found, that is, when found, back to the central C2 server. And so it's that server which has accumulated a total of 1.5 million of these reports from the bots. And then, after reporting the addresses of 80 of these potentially victimizable RDP servers, the command-and-control server then chooses a number of targets which that bot should brute force. So the new bot finds 80 new targets, phones those home, and then from the command-and-control server receives username-and-password pairs for it to attempt to use in logging into that site.

So standing back sort of from a distance, what this looks like is the central command-and-control server is using its botnet to coordinate a diffuse scanning operation, and also a diffuse brute force attacking operation, doling out usernames and passwords and IP addresses and saying, okay, give this a try. And then when it says, no, that didn't work, it's like, okay, give this one a try. Okay, that didn't work. But also try a different IP.

So what this does is it distributes the attack so that, from the viewpoint of the server under attack, it's getting different attempts from all over the Internet so that, if it had

logic to blacklist an IP that was pounding on it for doing a credential stuffing attack, then the fact that they're coming in from all over the Internet would prevent IP address blacklisting. So, I mean, that's the only thing that really makes sense based on what we're seeing. And once a successful credential attack has succeeded, then the remote RDP server then has the bitcoin - I'm sorry, I got myself confused because it's actually downloading a jar file named bitcoin.dll - I guess, again, to sort of pretend to be like something going on with bitcoin mining, although it's actually a Java jar file that is handed to the Java runtime in order to execute.

So it zips. It has the Java class and runtime zipped up. It transfers that to the newly infected target, which then unzips it, installs the runtime, and then runs this bitcoin.dll in order to bring up another instance in RAM of a new bot which then checks in with the mothership. It checks in. It begins scanning, tries to find new things to attack. And after it finds 80, it gets some new machines to probe. So an interesting sort of odd take at a botnet, but one which is apparently becoming successful and doesn't take advantage of the RDP exploit.

So the researcher wrote that: "After six hours," they wrote, "we received 2.1 million IP addresses from the C2 server, of which 1.596," so actually almost 1.6 million, "are unique." And he said: "Of course, we didn't execute the brute force phase." So anyway, so they're running a honeypot. They've seen what thing is doing. And we have sort of an interesting botnet whose mission is to work on brute forcing RDP servers that it's able to find.

And I mentioned this interesting letter, or I guess public service announcement, from the FBI. The FBI has reminded us not to place trust in a website just because it's secure. I have the link. It's ic3.gov. And they said: "Cyber Actors Exploit 'Secure' Websites in Phishing Campaigns." So this is the FBI sort of talking to the general public, saying: "Websites with addresses that start with HTTPS are supposed to provide privacy and security to visitors. After all, the 'S' stands for 'Secure' in HTTPS: Hypertext Transfer Protocol Secure. In fact, cybersecurity training has focused on encouraging people to look for the lock icon that appears in the web browser address bar on these secure sites." When is the last time there was a lock icon on a URL, Leo?

Leo: No, they still...

Steve: Oh, it is?

Leo: Yeah, there's a little padlock. And good cybersecurity training would encourage people to click on that lock, but I'm sure that's what the FBI's going to...

Steve: They said: "The presence of HTTPS and the lock icon are supposed to indicate the web traffic is encrypted and that visitors can share data safely. Unfortunately, cybercriminals are banking on the public's trust of HTTPS and the lock icon. They are more frequently incorporating website certificates," and then the FBI writes, "third-party verification that a site is secure, when they send potential victims emails that imitate trustworthy companies or email contacts. These phishing schemes are used to acquire sensitive logins or other information by luring them to a malicious website that looks secure."

And so the FBI had their boilerplate recommendations. "The following steps can help reduce the likelihood of falling victim to HTTPS phishing. Do not simply trust the name on an email. Question the intent of the email content. Second, if you receive a suspicious

email with a link from a known contact, confirm the email is legitimate by calling or emailing the contact. Do not reply directly to a suspicious email. Three, check for misspellings or wrong domains within a link, for example, if a domain should end in .gov, but ends in .com instead." Of course, that's sort of an FBI-oriented perspective. And they said: "Do not trust a website just because it has a lock icon or HTTPS in the browser address bar."

So of course none of this is news to us. We know that the only assurance actually being provided by HTTPS and TLS is that our connection to the web server is encrypted, and the server provided a certificate matching at least some of the domain name shown in our browser. In other words, barely anything of any consequence any longer. But the FBI, as I've mentioned, I think must be sending this out because they have their finger on the pulse of the public more than we might, or more than I might. As I said, Leo, I think you probably do from people you talk to on your weekend show all the time.

Leo: Yeah. I say this on the radio constantly. I don't even mention that you can't trust HTTPS. I say click the lock. Click the padlock. You've got to see who owns that certificate.

Steve: Right. Right. And so what I found interesting about this and wanted to share with our listeners is that it's interesting that the public understanding of things like this probably lags about a decade behind where we are.

Leo: Yeah. Yeah, probably.

Steve: You know, 10 years ago, before Let's Encrypt, before automated certificate issuance and the promiscuous use of wildcard certificates, you know, asterisk dot anything, having a certificate kind of meant something. But it's true that, during the past five years especially, its meaning has become quite watered down, which is an unfortunate consequence of the push for encrypting everything. After all, if you do encrypt everything, and then you make it easy for the bad guys to even encrypt their things, then yeah. You've got encryption, but now it doesn't mean anything special anymore.

Leo: I just checked Chrome and Firefox. Both still have padlocks.

Steve: Oh, okay.

Leo: Next to the site address, which...

Steve: Oh, yeah. And even Firefox.

Leo: Yeah, you remember that.

Steve: Yes, I see the little green padlock for me.

Leo: Yeah, and actually I don't know if Edge does. I'll have to check. But I just train my audience, I say, "Click that padlock link and make sure the site that you think you're on matches your certificate." Otherwise - and I even tell them, it's not just hackers. It could be your boss, could be your business, could be your Internet service provider. There's a man in the middle, and you don't want a man in the middle of your secure transaction.

Steve: Right.

Leo: Yeah. I think people get that. But clicking that padlock is something I'm trying to train my audience into doing.

Steve: I think that's good. And of course 10 years ago...

Leo: You didn't have to.

Steve: Back when certificates meant more, security experts were all jumping up and down, extolling the virtues of the unbroken key or the closed padlock.

Leo: Right, right. That's why people think this; right?

Steve: Exactly.

Leo: Yeah. Key's not broken, it must be good.

Steve: Exactly. So, yeah, it's secure, so I can trust it. It's like, no, unfortunately, it no longer means what it used to.

Leo: Yeah.

Steve: Which is too bad. This is sort of a cool piece of consequence. I mentioned at the top of the show VLC, the VideoLAN media player, which is very popular. I have it installed on my various machines because it's a very good media player. It received 33 security bug fixes, two of which are rated high severity. So first off, if you are also a VLC user, just run it. I ran it this morning, and it immediately popped up with a notice that it had an update and was fixing some severe - it was shown in red - security flaws. They said, the notice said: "VLC 3.0.7 is an important security update to VLC 3.0 branch, improving HDR, 10- and 12-bit rendering, and Blu-ray support, in addition to numerous security" - and that's where the word was in red - "issues fixed."

Back at the beginning of the year we mentioned that the EU had started sponsoring bug bounties in the hopes of improving the security of popular open source projects which their institutions were using and relying upon. And, it turns out, it appears to be working. The president of VideoLAN, Jean-Baptiste Kempf, said: "This high number of security issues is due to the sponsoring of a bug bounty program funded by the European Commission during the Free and Open Source Software Audit (FOSSA) program."

So that's very, very cool. The EU produced a bounty, said we're offering prize money for security problems found in VLC. And as a consequence, this broke a record. There have never been 33 problems found, including two high severity ones. So these two big problems and this release of 3.0.7 occurred last Friday. One of the two problems was an out-of-bound write vulnerability; the other was a stack buffer overflow bug. Developers behind the software said that the patches were two of the 33 being pushed out for the media player.

So this is not super crucial. The only likely threat would be from a targeted attack against an individual or an organization that was known by the attacker to be using VLC. So somehow an attacker would have to get someone to use a vulnerable version. And again, the moment it pops up now, it displays this notice. So I guess if you ignored the notice and hadn't already updated, then it could get you. But again, it's very unlikely.

Basically, a specially crafted malformed media file would have to be played by the susceptible version of the player and then, well, and then it would execute the attacker's code in your machine, probably in the context of the logged-in user. That is, I don't think you would get, depending upon where the codec was - and it's not clear to me that it would be running in the kernel. They are supporting GPUs, though.

So if there were drivers that were vulnerable, and NVIDIA did just update their drivers to fix some high-security vulnerabilities, it's conceivable that they could get elevated privilege. But this is, again, this is why user-level privilege escalation is a problem is that it does come in very handy in order to make attacks substantially more powerful, although even where the attack itself doesn't give someone significant attack posture.

Also, Microsoft's Edge browser has taken another step forward. The Microsoft Edge development builds, the ones that you can download in advance if you're interested, now allows, now supports a feature that we knew has been coming, which is the ability to essentially relaunch a page in IE11 mode. You would first go to a site and, presumably, it wouldn't work. Or maybe there will be a way eventually for the site to say, "I need IE11 mode."

Anyway, at this point it's under the menu. You open the Edge menu, go to More Tools, and then under More Tools, as of this latest Edge development build, there is an option, "Show this page in Internet Explorer," which will re-render that site in the old IE11 code, which is presumably useful for legacy sites maybe being used internally in corporations that have code that isn't under Edge and doesn't make sense for them to move to Edge. This, of course, makes it possible for them not to have to do that.

So we have that now at the development level. I'm sure, once Edge is officially released, it'll just be there. And maybe they'll have some way of flagging these sites as needing to come up under IE11 right from the get-go, or some way of causing that to happen. For now, you can just ask the site to be re-rendered that way.

And Mozilla has been reorganizing things. They've officially changed their logo to sort of an updated Firefox logo. So Mozilla is sort of more - they're kind of cozying up to Firefox because of course that's mostly how they're known anyway. There's a new Firefox logo; a new Lockwise logo. Lockwise was originally named Lockbox, and that's a password management service, which is free, which Mozilla is offering to allow Firefox users to synchronize their saved browser login credentials among iOS, Android, and desktop versions of Firefox. So that's now been named Lockwise and has a new logo.

They have their Monitor service, which we discussed back in September of last year when it was announced, which is a service that is being done in connection with Troy Hunt's HaveIBeenPwned service. Basically, you submit your email address to Monitor, and then they will periodically poll HaveIBeenPwned to see whether your email address appears in

any data breaches and proactively notify you if that has happened. And that's what's different about what Troy is offering.

And, by the way, Troy was in a London security conference last week and announced that HaveIBeenPwned is outgrowing his one-man-size shop. And so I'm not sure what that means, but it looks like it's increasing in popularity, and so we may be seeing some changes coming there, as well. And the fourth service is Send, Firefox Send. It also got its logo updated with this new look and style. And so all of this gets wrapped around services that Mozilla is offering under the Firefox moniker. So anyway, just some nice news there.

There was another piece of news that caused me to go find something which has been scrolling and sort of distracting me a little bit, actually, during the podcast because it's so cool.

Leo: Is that what I've been hearing in the background every once in a while?

Steve: No, actually that was my email, and I finally closed that. I had a couple pieces of email come in. It's actually a little more interesting than email. It is showing every DNS query that my system is making in the background. And our listeners know why that's interesting because anybody who's looked at their data is just like, what the heck is going on? I mean, our systems are noisy now. There's just a lot of stuff happening. And anyway, so in the news was that Mark Russinovich, our friend over at Microsoft who's now the CTO of Azure at Microsoft - and he of course used to be at Sysinternals. And everyone worried when Microsoft bought Sysinternals. The good news was they didn't do bad things to it. They kept it there, and all the tools are still free, and they've continued to update them.

Well, there's one tool, Sysmon, which logs things into the Windows Event Log. The news is that it's getting a new feature, which is the ability to log DNS queries into the system's event log. Well, okay. First of all, I don't think that's useful. For one thing, there's a lot of DNS queries that a system makes. And the event log is not a very useful place for them to go. I mean, it's not very accessible. But I remembered something that I remembered seeing from a prolific coder who codes in my style. You probably know the site NirSoft, Leo.

Leo: Oh, yes, of course, yeah.

Steve: Yes. N-I-R-S-O-F-T.

Leo: It's been around for decades.

Steve: Yes. Well, since early 2000s, actually. So a decade. Or, no, you're right, almost two. So the guy's name is Nir Sofer. N-I-R is his first name. Sofer, S-O-F-E-R, is his second name.

Leo: He's so prolific. This guy has so many tools.

Steve: Yes. And they are lightweight, tiny little things. So this thing is DNS Query Sniffer. And I recommend it for Windows users. It is just - first of all, it's one exe. I was a little jealous. It's a little bit smaller than my SQRL client. Of course it does actually much less. But it is neat. It simply shows you a running list of DNS queries that your system is making.

And there are a couple things you want to do, which I poked around at. There is a Resize Columns. You want to tell it to do that. But the most important one is Auto Scroll on New Line. That's a checkbox under the Options menu. And that way it does what its name sounds like. It continually auto scrolls. Ah. And I just saw sqrl.ver.grc.com. That was my installed SQRL client doing a once hourly DNS query to see if there's a new version. I did a very, very lightweight check for version. I used DNS in kind of a cool way. And I just saw it, just caught that happening.

Anyway, tiles.services.mozilla.com. I got some akamaiedge.net stuff happening, mozilla.com, safebrowsing.googleapis.com. There's some wallstreetjournal.net. Maybe I have an old tab open in Firefox, or maybe - anyway, it's just very cool. The idea is you sort of want to have an idea of why anything you see there is happening, very much like what happened as we were talking about at the top of the show which drove me to create OptOut, where ZoneAlarm popped up an alert for a program I didn't know was installed. Well, this would have worked. This would have done the same thing. Basically, as we know, you could hard code an IP address which would prevent making a DNS query. But nobody does that. So this is just a super lightweight way of getting a sense for what's going on on your network, thanks to Nir Sofer at NirSoft.com.

Leo: Dot net.

Steve: Thank you. NirSoft.net. And so it's called DNS Query Sniffer. And it showed all the various LAN adapters I had installed. It showed a bunch because I'm a user of VMware. So I saw the virtual LAN adapters that VMware workstation installs. I also have, of course, OpenVPN installed, so I saw the TAP adapter that OpenVPN installs and a couple others that are just on my motherboard. So I chose the one that was the IP address of my machine to my router, and I'm watching stuff happen. So it's very cool. And I commend its use to our listeners.

Speaking of commending, the movie whose trailer you do not want to see, but which I otherwise wholeheartedly recommend, is titled "I Am Mother."

Leo: I think I saw the trailer for that just by accident.

Steve: It is great.

Leo: It was a very short trailer, so maybe it didn't have all the spoilers in it. I think I saw an ad, a network ad for it.

Steve: Maybe. Anyway, it just came out on Netflix on Friday. Lorrie and I watched it Saturday and really liked it. So again, I commend it without reservation. It's two hours long. You've got to pay attention. There was one review that I really liked. Someone created a review, or created an account on IMDB, specifically because he had got a few things that it was easy to miss. I mean, so I should just explain, it's post-apocalyptic sci-fi, and well done. It got awards at Sundance. What's her name, the one-name star? Oh,

Rose Byrne voices the robot, does a great job of that. Hilary Swank is the most recognizable star in the film. But anyway, not super expensive. Good special effects. I just, you know, we talk about sci-fi here. We know I'm a fan of sci-fi. So for anybody else who is, "I Am Mother" on Netflix. Really nice...

Leo: Good. I'll be watching it tonight. That's great.

Steve: Really nice two hours.

Leo: Yeah, that's good.

Steve: So after tweeting about the release of SQRL, I have three tweet responses. Klaus Pinhack wrote: "Nice work," smiley face. "Setup of app and ID took me about 15 minutes. No problem for me, but too long for my neighbor." And again, this relates back to what I was saying is that I've thought about this a lot. There are things I could have done, there are things SQRL could be which would have removed features from it, which would have then made it susceptible to "what if" attacks, meaning like, okay, what if I'm crossing a border, and the border agent takes my phone and maybe forces me to give them my fingerprint and unlocks it so I no longer feel like I can trust the security of my identity. Okay. I'd rather have an answer to that.

Of course, I actually got a call from a good friend of mine who was once a Microsoft high-end, high high high-end developer, whose Google account got hacked, which allowed access to all of the usernames and passwords that he had been storing in Chrome, which was synchronizing through Google to any instance of Chrome that got installed. So he had a 100% loss, a complete compromise of all of his usernames and passwords. This happened a couple days ago. So the point is I wanted an answer for every possible thing that could happen.

So as a consequence of that, you do have to do a few things upfront to establish that kind of security, that kind of beachhead. And so I have something that cannot be any simpler than it is to also offer the features it does. My feeling is, once it's built into iOS, once it's built into Firefox and Chrome, once it's built into Windows, then it will be much less of a big deal. So it is in some ways a demonstration of the fact that it is possible to solve every problem. And, at this point, it's sort of all up to the user. And I'll be super interested to hear what the listeners to this podcast think.

Again, it's there now. Everybody can get it. So it'll be interesting to see what people think. But, you know, I understand. I completely get it that, yeah, it's not as simple as having really bad security. But anything less than this is really bad security, I mean, that things can happen to you that there's no recovery from. This allows you to recover from anything that can happen. And it's way more secure. For example, as I said before, SQRL gives websites no secrets to keep. The things SQRL gives a website is a public key. They can publish it. It doesn't matter. So when they get breached, nothing happens. You don't have to change anything. You don't have to change your password there, and so forth. And anyway, yes, I'm excited.

David Eckard wrote: "Downloaded the Android app. Setup less than fall-off-the-log easy." He says: "Ripe for a demo on YouTube for setup." And I hope people will make some. Please do. "Among other things, I learned that I need to be able to actually type the password on the client." And again, remember that that's so that someone doesn't take your phone and use your SQRL client, impersonating you to your SQRL client. But if your phone supports any kind of biometrics, then that's all supported, too. So when I present

SQRL, I just let the phone look at me and I'm logged in, which makes gasps from the audience like, oh, my god. It's like, yes, and this works anywhere, or will, as soon as other sites start supporting it.

And, finally, Yosef N. Berger. He said: "@SGgrc I just started messing around with SQRL, and I noticed there is a setting, Set Password Verify Time. I didn't see it mentioned in the FAQ and after a cursory watch of the forum patient - of the forum patient see any mention." I'm not sure what he meant there. "Could you go into what it does? Does it have any bearing on security?"

And so I'll just mention, our listeners will understand, that with SQRL - and actually it is explained in this 17-page explainer that's now public - you are able to say, "I want my password to take five seconds to brute force." And the idea is you only need to enter it once per session. And then after you can use just the first four characters, or the first N characters. You could make it just one character if you wanted because the idea is you're saying "I am still here," rather than "This is who I am." And you are able to set that time, if you don't want it to be five, if you think five seconds seems too long, you can turn it down to one second.

But what it would mean is that, if somebody were brute forcing your password, it would probably take them one second, or that is to say, one fifth as long if you changed it from five seconds to one second per guess of your password. So there's a modest security cost. On the other hand, it is very difficult to brute force and accelerate that one second because this is deliberately GPU and ASIC hardened. It requires 60MB per decryption, which no ASIC or GPU has per core. So it's not like running an SHA-256 50,000 times, which hardware has gotten very fast at. It is actively fighting against being accelerated. So there's a lot of brute force protection built in.

And, finally, I had a note to us about, tangentially, sort of about SpinRite. This guy said - this is Craig Clarke, who's the Client Engagement Officer, said: "Hi, Steve and Leo. Long-time listener of Security Now! since Episode 1, in" - okay, I don't know how to pronounce this place in Australia.

Leo: Adelaide.

Steve: Adelaide, thank you. That's the way it looked. I was sure I was going to mess it up.

Leo: He's the Client Engagement Officer at the Australian Taxation Office, just to be clear.

Steve: Ah. He says: "I have purchased SpinRite and have used it to restore many drives, floppy disks from my deceased father-in-law, and an iPod with a spinning hard drive." He says: "I'm very much looking forward to the next update to SpinRite." He says: "Just a reminder to change SQRL pages from being under your website's Research tab to its own tab for ease of finding."

Leo: Oh, good point.

Steve: Yeah, and I haven't done that. I forgot that.

Leo: Promote it. It's not research anymore. It's real.

Steve: That's right, "...and indicate that the research has been concluded."

Leo: Woohoo.

Steve: He says: "I'm sure that traffic to your website will soon spike to a very high level. Maybe you will need CacheFly."

Leo: Good.

Steve: "Please use personal email address for correspondence."

Leo: I know someone there. I can help you with that. That's awesome. Thank you, Craig. Congratulations, Steve. That's really great.

Steve: Well, we're getting there.

Leo: Yup. Steve, Exim, E-X-I-M. Talk about this.

Steve: So according to a recent survey of all mail servers visible on the Internet, 57%, that is, 507,389, are running Exim. And after seeing what Qualys found, I'm very glad I don't run Exim. I'm Windows-based rather than Unix or Linux based, and Exim is running on the most popular OS for the Internet, which of course is Unix or Linux. And if everyone listens no further, if you have in any way or if you are in any way connected to one of those more than half a million email servers present on the public Internet, you really should update to the latest release of Exim immediately.

The vulnerability, as I mentioned at the beginning of the show, was accidentally, but fortuitously, patched with the release of Exim version 4.92 back in February. It was four months ago, not four weeks ago. So back on February 10th, so almost exactly four months ago. It was accidental because the Exim team had no idea that they were fixing a major security hole. Qualys was just doing a review of the code, and in doing the code review they found a big problem.

Now, okay. The news hit last Wednesday, that is, Qualys went public with this. They sent a note to the Debian group because Debian's email server is by default Exim. So the news of this breach - and this was a full disclosure. So all the bad guys know exactly how to pull this off. And that's what I'm going to explain here in a minute. But what's interesting is that, due to the weird nature of this, in its default configuration it takes seven days to cause a vulnerable remote email server to execute commands under the control of the attacker. So if upon the release of this news last Wednesday someone began to attack your email server immediately, they're still waiting until tomorrow to be able to execute the commands that they've set up.

Leo: That's so weird. That's kind of unprecedented. I don't...

Steve: I know. It really is. It's because of an expiration in an unsendable piece of email.

Leo: Ah.

Steve: It defaults to a week.

Leo: Okay, yeah.

Steve: And so what happens...

Leo: It keeps trying for seven days, yeah.

Steve: Yeah. So what happens is you actually - you keep the connection up, and you send it a byte every four minutes because there's a different expiration after five minutes.

Leo: Oh, how funny.

Steve: And by sending it a byte every four minutes, you keep it waiting, and you do that for seven days. And then a different expiration kicks in which, due to this weird code path that they found, causes the - basically you put the commands you want to have it run as the email address before the @ sign on the domain. And what happens is the way this thing trips over its own feet is that, after a week of your patiently sending it a byte every four minutes, it will finally give up. And it ends up running this email address through a function which will accept the E-X-E-C, the exec function, and take the rest of the parameters in the address as commands at root, commands with root privilege.

So again, this is one of those problems that, because it was introduced back in - I've got it in my show notes here. I don't see it in front of me right now. I'll get to it. But it was introduced in 2016, early in 2016, so most of 2016, all of 2017, all of 2018, and up until the beginning of 2019. But the point is probably any system that came up then that has not been updated in the last four months would have had one of the vulnerable versions of Exim. And of all of the more than a million Internet email servers, 57% of them that are answering on port 25, SMTP, and whatever other ports may be susceptible, there are, we know 507,389 of those are vulnerable. So this seems bad.

And it also seems the idea that an attacker with some patience, you've got to have a little patience, you've got to wait a week, could then execute the commands of their choice as root on an email server. I mean, you know, these are going to be email servers of big targets, potentially. You know, IBM and Fortune 500 companies are going to be running Unix servers with Exim on it. And let's hope that they've been keeping them current, that is to say that they have updated, in the last four months, a server that went online sometime in the previous three years. Because, if not, I wouldn't be surprised if they got some commands being run on them as root.

So let's see. In my notes I have, yeah, Qualys wrote to the Linux distro maintainers that the vulnerability is "trivially exploitable" and expects attackers to come up with exploit code in the coming days. Exim 4, the affected version, is currently the default MTA, the Mail Transfer Agent, on Debian Linux systems. A large number of Exim installations exist,

especially within ISPs and universities in the U.K. Exim is also widely used within the GNU Mailman mailing list manager, and cPanel.

Wikipedia notes that "Exim's security has had a number of serious security problems diagnosed over the years. Since the redesigned version 4 was released, there have been four remote code execution flaws and one conceptual flaw concerning how much trust it is appropriate to place in the runtime user. The latter was fixed in a security lockdown in revision 4.73, one of the very rare occasions when Exim has broken backwards compatibility with working configurations." And of course now we have another biggie.

Qualys put out a security advisory, calling this "The Return of the Wizard: RCE in Exim," so remote code execution. And for anyone who's interested, this is CVE-2019-10149. And I pretty much covered this in summary, so I'm just going to try to find things that are important things that I didn't.

They wrote: "During a code review of the latest changes in the Exim mail server, we discovered an RCE vulnerability in versions 4.87 to 4.91 inclusive." And I tried to look to see whether the HELO message divulges the version of Exim. Sometimes the HELO message, or there's also a version of it, EHLO, on more advanced, more recent versions, sometimes says, you know, Exim version something. Which of course the bad guys would love to have because then immediately on answering any connection, TCP connection over port 25, the server would be basically waving a flag saying "Please hack me. I'm vulnerable." I don't know whether Exim does, so that would be one thing to find out. But of course, even if not, chances are very good that it would be vulnerable.

They said: "In this particular case, RCE means Remote Command Execution, not Remote Code Execution. An attacker can execute arbitrary commands with exec as root." No memory corruption, no return-oriented programming, nothing fancy, no buffer overflows and so forth is required. They said: "This vulnerability is exploitable instantly by a local attacker," so that's also worth noting, "instantly by a local attacker," they said, "and by a remote attacker in certain non-default configurations," that is, instantly exploitable, "by a remote attacker in certain non-default configurations. To remotely exploit this vulnerability in the default configuration, an attacker must keep a connection to the vulnerable server open for seven days by transmitting one byte every few minutes. However, because of the extreme complexity of Exim's code, we cannot guarantee," they wrote, "that this exploitation method is unique. Faster methods may exist."

They said: "Exim is vulnerable by default since version 4.87 released on April 6, 2016," they said, and then they have some code here, "when #ifdef EXPERIMENTAL_EVENT became #ifndef DISABLE_EVENT, and other versions may also be vulnerable if EXPERIMENTAL_EVENT was enabled manually." They said: "Surprisingly, this vulnerability was fixed in version 4.92 released on February 10, 2019." So anyway, I have a link in the show notes at this point to their comment; also a note at bugs.exim.org talking about this. I won't go into this in any much greater detail. In the show notes I have, for anyone who's interested, all of the details about what they go through.

Down at the very end of this they explain about default configurations, non-default configurations, local exploit and so forth. But under default configuration, which of course is the thing of most concern, I'll summarize this a little bit. They say: "We connect to the vulnerable Exim server and send a mail that cannot be delivered because we send more than" - then there's a setting, received_headers_max. They said: "We send more than that many received headers in the email envelope." So that causes a delivery fault.

They say the recipient address, as in the RCPT TO, of our mail is "postmaster," and its sender address, that is, the MAIL FROM is - and here's where the exploit is. It's ${run{, then the commands they want to have executed as root, then close both curly braces,

then @ sign, and then the domain where that's the domain that is under their control. So basically, so it's the MAIL FROM. So what happens is, as I mentioned before, it's when this vulnerable server finally tries to send back a bounce message that it encounters this, basically an executable account name at the destination domain. And, unfortunately, it executes the commands which are contained in the account name.

So then they said, step two: "Because our mail cannot be delivered, Exim connects to [the target domain's MX]," they say, "where we listen for and accept this back connection, and starts sending a bounce message" to that $run blah blah blah target. They say: "We keep this connection," that is, its attempt to send a bounce message, "open for seven days," basically preventing it from successfully sending the bounce message for a week. And they say seven days, the default timeout_frozen_after setting in the service, "by sending a byte to Exim every four minutes."

They said: "This works because Exim reads the response to its SMTP commands into a 4096-byte buffer," which is the DELIVER_BUFFER_SIZE, "with a five-minute timeout." So, that is, they use four to be conservative, to slip underneath the five-minute timeout, and that is reset every time a byte is read. So that just keeps getting reset to another five minutes every time a byte comes in. They just keep it trickling in.

And, finally, after seven days, step four: "After seven days we complete our lengthy SMTP response with a permanent delivery failure." For example, they return a "550 unrouteable address" response to Exim's attempt to deliver the delivery failure message. This freezes the bounce in a function known as post process one. "This function should actually discard the bounce instead of freezing it, which would prevent us from reaching the vulnerable code because it is older at that point than two days old, which is the default ignore bounce errors after timeout." But this doesn't happen. "In this particular case, the message age is not the bounce's real age, over seven days, but its age when it was first loaded from Exim's spool when it was just a few seconds or minutes old." So there's a mistake in the code path there.

And, finally, step five: "Exim's next queue run, every 30 minutes by default on Debian, loads the frozen bounce from the spool, sets process recipients to RECIP_FAIL_TIMEOUT (this time the message age is the bounce's real age over seven days), and executes the vulnerable code and our commands," which is in the original sender address, which is interpreted by "expand string," that will invoke exec as root. Oh, and they said: "Note: To quickly test this remote exploitation method, the days in Exim's default timeout_frozen_after and ignore_bounce_errors_after can be replaced by hours, and the default retry rule can be edited." So they're able to speed this whole thing up, basically in order to run this in the lab and see if this all works.

So what we have is we have what they described as trivial, I mean, yes, it requires a little doing. But it's going to be too tasty for hackers not to play with, the idea that - it's kind of cool almost that you send your commands to an email server, you accept its attempt to bounce it back, you hang the attempt for a week, and then it executes the command you gave it as root. It's going to happen. So unfortunately, it could happen to somewhere on the order of half a million, that is, more than half of the email servers currently accepting email on the Internet.

So for all of our listeners, you don't want you or your corporation to be among those. All you have to do, since if the attack began when this first came to light last Wednesday, you still have a day before a week has gone by if somebody started to attack your server immediately. So sounds like a good idea to update to the latest version of Exim, and you'll be okay.

Leo: Yeah. Did you see the story - it broke this morning, so you probably didn't have time to get it into the show - about the Customs and Border Protection losing the face recognition information to thousands of people who've crossed the border from Canada? Did you see that story?

Steve: I did not see that story.

Leo: One more reason to be really concerned about the use of face recognition in kind of non-safe ways. I mean, Apple's face recognition lives on the iPhone and stays on the iPhone. But the Customs and Border Protection says photos of travelers into and out of the country were accessed. And you're going to love how they were accessed. A subcontractor downloaded them, and they were later stolen from the subcontractor in a malicious attack. In violation of CPB's policies they downloaded - yeah, some policy. "Hey, don't do that. That would be bad."

But apparently they didn't protect it. They transferred copies of license plate images and traveler face images collected by the Customs and Border Protection to the company network. It's fewer than 100,000 people - which means it's 99,000, right? - who had gone through a few lanes at a single land border over a period of a month and a half. It's not from airlines. It's not passport or other travel documents. But it underscores how dangerous this is, to give an image of your face to anybody, especially to the government.

Steve: Yes.

Leo: Yeah. Can't change your face or fingerprint very easily. That's the problem. We've talked about that at Disneyland. They don't do it anymore, but they used to collect fingerprints. Somebody used his elbow, remember that, to get into Disneyland?

Steve: Right, yeah. We used to talk about using your knuckle instead.

Leo: Use a knuckle.

Steve: Just give it your knuckle.

Leo: Steve, always a pleasure. I know what I'm watching on TV tonight. I can't wait.

Steve: Oh, it's a really nice two hours. I think you will...

Leo: And pay attention, it sounds like, because there's a lot of...

Steve: Pay attention.

Leo: There's twists. It's complicated, yeah.

Steve: Yes. And unfortunately I don't think we can talk about it next week, you know, because I don't want to spoil it for anybody.

Leo: No, no, no. Yeah.

Steve: But we'll have to talk about it in a couple weeks because it was - a couple comments that were made at the very end are, like, oh. They're really good, yeah.

Leo: "Billions" had a big twist last night. I don't know if you watched it yet.

Steve: Oh, do you mean the season finale?

Leo: Yeah.

Steve: Yeah, I love that show.

Leo: They've now found their niche, which is - because they did it once before, but they didn't do it - they haven't done it for all the seasons, which is you set people up for the whole season, and then in the last episode everything you thought you knew is wrong.

Steve: Yeah.

Leo: Which I love. Big payoffs are always fun. Steve, there's another thing we have to set up, which is a time for you to come up here.

Steve: Yes. Yes, yes, yes.

Leo: Let's start exchanging emails because we want to do that. And you tell us who else you want to have on the show. We should find out when Father Robert's coming to town. I understand - this is sad news. He was supposed to go to DEF CON and Black Hat. And that higher authority that he works for has...

Steve: Oh, wait, how high?

Leo: Pretty high. All the way up. All the way up has nixed that trip. But we do know that he'll be coming back to town at some point. So I don't know if we want to wait that long. I'll tell you what. I'll contact Robert.

Steve: What really works is having an audience. And so I was thinking, I wanted to propose to you that we have...

Leo: Oh, do it somewhere.

Steve: Well, like if there's anywhere we could set up a whole bunch of chairs, and I do a presentation to as big an audience as we can fit in the studio.

Leo: We can get about 30 or 40 people in our studio there. I think we could - that'd be enough; right?

Steve: Yeah, yeah, just so there's sort of some crowd dynamic, and I can take questions. Because what really works is for me just sort of jumping around up in front of a screen showing pictures and describing the whole thing.

Leo: Oh.

Steve: And then having people say, well, wait, what about this and what about that? So I would imagine...

Leo: That would be really fun. Maybe have to book a hall for that. That sounds like something we'd like to do, and maybe down in San Francisco so we could get the SQRL show on the road.

Steve: As long as we have your professional photography. That's really the only thing that is missing. And I'll bring the popcorn.

Leo: I inadvertently just took delivery from Amazon of a five gallon bag of caramel corn, which of course I can't eat because I'm keto.

Steve: Ooh, ooh.

Leo: I meant to send it to my mom, but I'm going to bring it to the office and let these guys eat it.

Steve: Well, you've got a lot of young bucks there, so...

Leo: By the way, thank you for the keto suggestion. People, he's got a lot of great information about keto on his website. And I've been doing it now for three months, lost 15 pounds, my blood sugar is normal again.

Steve: Congratulations.

Leo: It's really been a great thing. My blood pressure is normal again. Yeah, I'm very happy. So you had said it all along. But what turned the corner for me is doing it under a doctor's supervision with accountability.

Steve: Yeah, yeah, nice, nice.

Leo: Because I measure my vitals every day, and so that's important.

Steve does this show every Tuesday. As you can tell, it's kind of a time for me and Steve to get together. But we're glad you listen, too. It's about 1:30 Pacific, 4:30 Eastern, 20:30 UTC. You can watch live or listen live at TWiT.tv/live. If you do that, join us in the chatroom at irc.twit.tv. I would love to have you download it, too. I mean, you can get your 16Kb version if you're bandwidth impaired, a 64Kb version if you're not, and very nicely curated transcripts at Steve's site, GRC.com. He's got it all there, including show notes.

By the way, if you're there, you might want to pick up a copy of SpinRite, world's best hard drive recovery and maintenance utility. Ever hear of that? Be very handy for you to have, I can promise you that. There's lot of other great stuff on Steve's site, too: GRC.com. He's @SGgrc on Twitter. That's where you can leave him a direct message if you have a question, a comment, or a suggestion. Or you can go to GRC.com/feedback, get the same thing done.

We have audio and video at our website, TWiT.tv/sn for Security Now!. And of course you can always subscribe. That's probably the best thing to do. Get yourself a podcast application and just subscribe to Security Now! so you'll get it automatically, every day, the minute it's available. Steve, thank you. Have a wonderful evening. See you next week.

Steve: My friend, I can't wait. Bye.