The Meris Botnet

Description: This week we're going to note the apparent return of REvil, not nearly as dead and gone as many hoped. We're going to look at a new and quite worrisome zero-day exploitation of an old Windows IE MHTML component. Even though IE is gone, its guts live on in Windows. We're going to share the not surprising, but still interesting, results of security impact surveys taken of IT and home workers, after which we'll examine a fully practical JavaScript-based Spectre attack on Chrome. I have a bit of closing-the-loop feedback to share, and a surprisingly serious question about the true nature of reality for us to consider. Then we'll finish out today's podcast by looking at the evolution of Internet DoS attacks through the years, which recently culminated in the largest ever seen, most problematic to block and contain, RPS DDoS attack where RPS stands for Requests Per Second.

High quality (64 kbps) mp3 audio file URL: http://media.GRC.com/sn/SN-836.mp3
Quarter size (16 kbps) mp3 audio file URL: http://media.GRC.com/sn/sn-836-lq.mp3

SHOW TEASE: It's time for Security Now!. Steve Gibson is here. ActiveX is back. And it's a zero-day for everyone. Congratulations. We'll talk about REvil the ransomware gang is back. And then it's the biggest DDoS attack in history. The details of the Meris botnet and why we all should be afraid. It's all coming up next on Security Now!.

Leo Laporte: This is Security Now! with Steve Gibson, Episode 836, recorded Tuesday, September 14th, 2021: The Meris Botnet.

It's time for Security Now!, the show where we cover your security, your privacy, your safety online with this guy right here, Security Now! maven Steve Gibson. Hello, Steve.

Steve Gibson: Leo, it's great to be with you again for this big Apple news day.

Leo: Oh, yeah. Sorry we're a little late because of that, yeah.

Steve: The iPhone 13. And I think I'll just stay with my 12. Seems fine.

Leo: Yeah, yeah.

Steve: It works. And the pads, I've got so many. I have, like, every pad they ever made from the beginning. So I was giving them away for a while. But now I've got, you know, pads I like.

Leo: And, you know, what's absolutely the case is there's not that much difference in these new models of any of this.

Steve: Well, yes. You guys were talking about that. And I was thinking as I was hearing that, that this is the evidence of a mature industry; right?

Leo: Exactly.

Steve: It's like, we're not having, oh my god, breathtaking breakthroughs now. Now it's like, oh, look, you know, my bandwidth went up by 5%. It's like, okay, but it didn't go up by 1,000% kind of thing.

Leo: Right. When's the last time you felt the need to get a new toaster?

Steve: And they were talking about 5G. It's like, wait, don't I already have that? Yeah, I do. So, yeah. Although, Leo, Leo.

Leo: Yes, yes. Steve. Steve, Steve, what? Yes.

Steve: When I realized that their whatever that was, camera technology, like when the guy turns away from the camera to look behind him, and it focuses on what's back there, it's like, oh, you're kidding me. And then he turns back to look at the camera, and it puts the focus back on him. It's like, okay. That's, you know, Apple schmaltz.

Leo: No, no, no, that's so funny. It's such a...

Steve: Anyway, what were you going to say when I so...

Leo: Oh, just that you haven't yet tweeted your show notes. So I just wanted to mention that.

Steve: Oh, I forgot that.

Leo: But that's all right.

Steve: I'll do it.

Leo: But you put it online and everything.

Steve: Oh, that's right. But you need that.

Leo: Oh, no, I have them. I have them. I get them. There's an email, as well, so I have them.

Steve: Oh, that's right, okay.

Leo: And including your Picture of the Week, yes. Which I love.

Steve: Okay. I forgot, we've already started.

Leo: Yes, the show's begun.

Steve: Hello, everybody.

Leo: Yes.

Steve: This is Security Now!.

Leo: The reason I want you to get going is because I have a fresh doughnut from Chicago's Stan's Donuts, and it's crying my name out loud. And I can't bite into this until you...

Steve: Yes. And remember to mute your microphone because it's wrapped in paper.

Leo: Yes, I shall.

Steve: And we'll be hearing you crinkling the paper.

Leo: Oh, that won't be all you'll hear. You'll hear [appreciative mumbling].

Steve: So I don't speak Latvian, or I would first of all have known how to help John put a flat bar over the "e," which is in the show notes, and it's everywhere, just because I was cutting and pasting, and it came with it, happily. But I don't know how to pronounce it. So maybe it's the Meris Botnet? Meris? Meris? I don't know. Anyway, it's Latvian for "plague."

Leo: Oh.

Steve: Yeah. And the topic is this botnet because it represents sort of essentially a steady evolution over time. We've now gotten to the point where we have distributed denial of service attacks which are incredibly difficult to block. And not only because of the nature of the attack, which rather than just like packet-level attacks, like down at Layer 3 and Layer 4, these are Layer 7 attacks. They're application layer attacks. But because we've, you know, it's in this back-and-forth, cat-and-mouse game of attack and block and new attack and new block and so forth, all that's done is escalate this.

Anyway, we're going to go back and quickly look at the evolution of attacks, remind what Layer 3 and 4 and 7 are about, and then look at this one in detail, which is being launched by this Meris botnet. But first we're going to note the apparent return of REvil, not nearly as dead and gone as many had hoped.

Leo: Yeah, I saw that. I thought that was interesting.

Steve: Yeah. Well, and it reminded me that we shouldn't ascribe any motivations that we don't actually know about to any of these culprits out there. I mean, when they disappeared shortly after the Kaseya attacks, people were like, oh, Putin must have stomped on them or something.

Leo: Right, right, right, right. And then we thought another botnet or ransomware gang was them.

Steve: Was renamed, exactly.

Leo: Maybe it was, but...

Steve: Apparently, yeah, who knows.

Leo: Who knows, is right.

Steve: Anyway, we're then going to take a look at - exactly, who knows. We're going to take a look at a new and quite worrisome zero-day exploitation of an old Windows IE MHTML component. Even though IE is gone, its guts live on in Windows. We're going to share the not-surprising but still interesting results of a pair of security impact surveys, one taken of IT professionals who are trying to do the securing, the other taken of home workers who are trying to get around it. So that had some interesting little tidbits and stats that came out of it.

After that we're going to examine a fully practical JavaScript-based Spectre attack on Chrome. It's not just theoretical anymore. Then I have a bit of closing-the-loop feedback to share, and a surprisingly serious question about the true nature of reality for us all to consider.

Leo: Ooh.

Steve: It might be possible for you to eat doughnuts with no consequences, Leo.

Leo: That I'm interested in.

Steve: I thought that might - yeah.

Leo: Now you've got me.

Steve: And we will then finish out the podcast by looking back at where we've come from the beginning of DDoS and where this has landed us, after we of course deal with our Picture of the Week. Okay. So I added this caption to this photo of a coffee mug that I thought was a kick. The caption reads: "What do you mean, you've been unable to create an account there? What's the problem?"

Leo: And here's the answer.

Steve: And here's the answer. The coffee mug reads: "Sorry, but your password must contain at least 8 characters, upper and lower case letter, a symbol or a number, a hieroglyph, a haiku, a musical note, the feather of a hawk, and a drop of unicorn blood." Otherwise...

Leo: To which we're going to add "and the letter 'e' with a line across the top."

Steve: Yes.

Leo: And if it's not there, forget it. You can't get in.

Steve: So, okay. Microsoft is warning of a newly discovered IE, believe it or not, sort of indirectly IE, zero-day being actively exploited, currently in targeted attacks using their Office apps. While the danger might not be extreme, especially if the use of this exploit remains targeted, this should remind us of our Picture of the Week two weeks ago, which was titled "Pandora's Inbox," where Pandora's depicted thinking to herself, it can't hurt to open one little attachment; can it?

And while I agree that it's unlikely to hurt any of us, we do know that once a zero-day has been observed being used, and it's become public, those highly targeted attacks likely become spray attacks. You know, the secret is out, and a patch will be forthcoming. Which means that the optimal strategy at that point, for those who wish to exploit what has now become a time-limited advantage, is to go from targeting individual people to spraying this thing far and wide to collect all of the curious and even the incurious Pandoras which may be possible. So my word to our listeners, don't be a Pandora.

When we hear that it's an IE zero-day, that's really a misnomer because the vulnerability, which is now being tracked as CVE-2021-40444, was found in Microsoft's MHTML component, that was also known as Trident, which is the IE browser engine. And while IE itself is no longer showing up, like when you click on a URL you get Edge, but that component remains in active use by Office documents, which use it to render any web-hosted content which has been embedded into Word, Excel, or PowerPoint.

So Microsoft has said that they're aware of targeted attacks exploiting this vulnerability, and that, quote, this is them: "An attacker could craft" - which means, you know, is crafting, attackers are crafting - "malicious ActiveX controls to be used by a Microsoft Office document that hosts the browser rendering engine." They explained that the attacks and the underlying zero-day were discovered by security researchers from Mandiant and EXPMON. I've never run across them before, E-X-P-M-O-N, all caps. And last Tuesday this EXPMON gang who maybe ought to come up with something pronounceable, tweeted: "EXPMON system detected a highly sophisticated zero-day attack in the wild." And apparently now we have a new acronym or abbreviation, rather sorry, abbreviation. Every time I say "acronym" by mistake, I get corrected by the loving folks on Twitter who say, you know, Steve, it's only an acronym if you can pronounce it. It's like, oh.

So this is "in the wild" - ITW is our new abbreviation - "targeting Microsoft Office users." They tweeted: "At this moment, since there's no patch, we strongly recommend that Office users be extremely cautious about Office files. Do not open if not fully trust the source." And I thought for a moment that Obi-Wan had given Luke bad advice, but I remembered that was "Trust the Force," so that's different. You don't have to worry about that. And yes, I did have a lot of coffee while I was waiting for the podcast to start.

Leo: I love it. Trust the source, Luke.

Steve: Trust the source, Luke. Yes.

Leo: Love it.

Steve: And since Microsoft's disclosure, the hacking community, this is what's really interesting, just has exploded over this thing over the past week. I think the disclosure was last Tuesday. A couple days later there was proof of concepts. There was stuff on GitHub. I mean, they just went into overdrive. Posts, guides, demo code, working proofs of concept, and everything anyone could need to design and launch their own attacks.

Leo: Oh, Jesus.

Steve: In fact, the guys at BleepingComputer used the available stuff to duplicate the attacks themselves. It's like, yes, it all works. All freely available since late last week. Even GitHub has been hosting complete exploit documentation.

Now, the good news is Windows Defender's knowledge base has been updated immediately to recognize known instances of the attacks. And it's worth noting that, even though patches tend to be slow sometimes to come out - and this is Patch Tuesday, here we are on the 14th, which as I mentioned last week makes it the latest Patch Tuesday that can occur.

And I don't know if something that Microsoft just announced last week on Tuesday, if it's going to make it into today's patches. We'll have to wait to see if that's happening today. But the good news is we are seeing that Windows Defender updates, which are occurring daily, if not more so, are being good about catching these things and blocking them if you're using Windows Defender. It's less clear what third parties are able to do, if they are able to get the advanced knowledge that they would need from Apple in order to be

able to offer that. Anyway, some protection is presently available to everyone using one of the multiple forms of Defender, so that's good.

There are, as I said, guides to implementing, well, there are guides to doing it. There are also - there's a bunch of mitigation advice. But that's been a moving target because, as I said, this has been highly active the past week. Attackers keep finding ways around mitigations as they are suggested. So hopefully this thing gets fixed. I would just suggest that, I mean, it's annoying to not be able to open attachments. But again, two weeks ago, Pandora's Inbox. You don't want that to be your inbox.

So just be aware that this thing may have exploded from targeted-by-a-sniper sort of attacks to shotgun, just spray everywhere and get who you can. With any luck this thing's been fixed, and you'll get patched today, and this won't be a problem. But it is bad.

Leo: Why is ActiveX still alive?

Steve: I know. I know.

Leo: It's still in Windows?

Steve: Well, it's because documents live on, and you wouldn't want that document not to be able to open a website now, would you?

Leo: Yeah.

Steve: Because, you know, you've got to have that in your PowerPoint.

Leo: I remember you specifically talking about what a threat it was to allow something downloaded from the web to run locally on your computer as an app.

Steve: Well, because - and it's bringing in JavaScript. What could possibly go wrong? So not only is your document scripting in order to bring in, in order to host a container which is then a web browser in your document, which has been given a URL to a foreign server, which can then load something in with JavaScript running and, like, do something. It's like...

Leo: There should be a way to just remove ActiveX. I'm stunned that it's still alive in there. That's crazy. That's an Internet Explorer component?

Steve: Yeah, yeah. Well, ActiveX is what COM evolved into.

Leo: Right.

Steve: So there was the Component Object Model (COM). And then they got so tired of doing extensions of it because they kept figuring out it could do more, that they said, okay, let's just kind of start over. So we'll call it - oh, and it was also a renaming. I remember that it sort of like - they didn't feel like it was exciting enough.

Leo: It's active.

Steve: ActiveX, yes.

Leo: But before COM was OLE; right? Which that's pretty good. OLE sounds like a bullfighter.

Steve: Right, like OLE. That was good, yeah.

Leo: It's still supported through Internet Explorer 11, even though it's been deprecated for years.

Steve: Right. And so this is invoking IE, an old IE control, through ActiveX in order to bring it back alive. So, yeah. And, I mean, notice we're not even talking about the fact that IE's MHTML control has a problem, because of course it does. Why would we imagine that a browser component would not have a horrible, easily exploitable flaw. Instead we're just talking about, oh, this is the way you invoke it because, yeah, embedded in Office documents.

Okay. So we also have - this seems to be abbreviation day. I also ran across WFH, which is the new abbreviation for Work From Home. That's now a thing. You WFH. You're WFHing. Anyway, last Thursday HP's Wolf Security Group published a new study which they titled "Security Rebellions & Rejections Report." Not very optimistic. It's a compilation of data from an online YouGov survey aimed at office workers who adopted Work From Home and global research conducted with IT decision-makers.

So as I said, on one hand we've got the IT guys on the front lines, trying to not have those lines move. And now all of this traditional in-office workforce which thanks to the pandemic has been shuttled to home, who don't really understand why things are more difficult to do from the comfort of their couch, as opposed to the inherent containment that you get when you have to go into the office, maybe even wear a tag around your neck to show that you belong there, and then can sit down in front of your desktop at the office. Like why is my kid's Chromebook at home any less secure than my IT-enforced machine on the desktop?

Okay. So in total, 91% of the IT people surveyed said they have felt pressured to compromise on security due to the need for business continuity during the COVID-19 pandemic. So a little over nine out of 10, right, 91%. 76% of those respondents said that security had taken a backseat, with 83% believing that working from home has created what was described as a "ticking time bomb" for corporate security incidents.

I have a chart in the show notes which shows the relative level of threat IT teams feel as a result of their office working employees increasingly working from home. This was Figure 3 from this report. And so 84%, the number one most occurring concern is ransomware, that home employees would let bad guys in who would then be able to

crawl up the VPN connection into the corporate network, and ransomware was the biggest problem at 84%.

In second place at 83, firmware attacks against laptops and PCs. I thought that was interesting, just like because this is - I guess the IT folks, they're used to having the equipment in the physical plant and not spread out all over hell and gone, where they don't have any control over it. So firmware attacks against laptops and PCs. Now, that's interesting, too, because that does suggest, and I'm suspicious of this, but okay, that the whole Secure Boot thing that we talked about last week really is important; that you really want to know that the BIOS or UEFI firmware has not been altered in a way that would allow these PCs to no longer be booting 100% legitimate OS.

Leo: Such a difficult attack, and requires usually on-prem attackers. I'm surprised that's ranked so high.

Steve: Yeah, yeah.

Leo: I really am. I don't think that's as big a threat as they think it is. Or maybe it is.

Steve: I don't think so. Also at 83, same threat level, exploited vulnerabilities in unpatched devices.

Leo: Well, yeah. Duh.

Steve: How does that even fit on this chart, Leo?

Leo: That's all of them.

Steve: Yeah. My lord.

Leo: You mean Windows? Yeah.

Steve: So then we have at 82% - these guys really, I hope they've got some anti-stress meds they're taking because they're really worried about this stuff - 82% is data leakage, which is generic, but okay. 81%, I mean, these bars are just staying up there. 81% account/device takeover, man-in-the-middle attacks.

Leo: They might as well just say, yeah, we're worried about everything, thank you.

Steve: You know, this probably was a multiple choice, like, survey.

Leo: Yeah, that's what it was.

Steve: And they said yeah, and yeah, and yeah, and yeah, and yeah.

Leo: Yeah, we're worried about it all, yeah. Why wouldn't we?

Steve: Yeah. Who wasn't worried about data leakage? I'm worried about anything that leaks. So that's not good. Especially as I get older. Okay. So IoT threats, that's 79%. That's sort of a little more amorphous than I guess their own local PC stuff. Targeted attacks, okay. So I guess probably that flows from their own experience of dealing with a large employee cohort who are opening things they shouldn't.

Leo: That's really what they should be worried about, I would say.

Steve: Yeah, yeah. And then down at 76% are firmware attacks against printers.

Leo: That's oddly specific. Wonder what they're talking about?

Steve: They didn't say green printers or anything, just said printers.

Leo: Printers, yeah, uh-huh.

Steve: Okay. So, now, the view from the suddenly remoted office worker, it does substantiate the worries of those who are responsible for securing their experience. Okay, specifically, according to the survey, young office workers in particular, you know, those pesky muskrats that don't really understand all the damage that can happen, they're more likely to circumvent existing security controls and safeguards in order to manage their workloads, with 48% of this younger cohort saying that security tools, such as all those pesky website restrictions or VPN requirements, half of them say they're a hindrance, with at least one third, 31% of them having at least attempted to bypass the restrictions.

Overall, 48% of office workers said that security measures waste time, and 54% in the 18- to 24-year-old bracket were more concerned with meeting deadlines than potential security breaches. You know, I guess it's not their problem; right? It's like, oh, well, you know, those pesky IT guys, they'll worry about that. I've just got to get my work done. And within that same group, 39% stated that they were unsure or unaware of their employer's security policies. Yeah, I just, you know, I cash my check. So it's like, okay.

Three other bullet points which the report highlighted were: 37% of office workers believe security policies are often too restrictive; 80% of IT teams experienced backlash from home users because of security policies; 83% of IT teams said the blurred lines between home and work life were making enforcement, and this was quoted in the survey, "impossible." So I guess it wouldn't surprise us that this is generically the case. But this really I think brings home just how much a problem was created by this rapid exodus from the office to everybody working at home. Everyone likes it. It's like, yay, this is great. We can conduct meetings. I can take my own dog out for a walk as necessary and so forth. And a lot of people don't want to go back; right? It's like, hey, this a good thing.

Joanna Burkey, HP's Chief Information Security Officer (CISO), she said: "CISOs are dealing with increasing volume, velocity, and severity of attacks. Their teams are having to work around the clock to keep the business safe while facilitating mass digital transformation with reduced visibility." She said: "Cybersecurity teams should no longer be burdened with the weight of securing the business solely on their shoulders." I don't know who she thinks is going to take it over.

She said: "Cybersecurity is an end-to-end discipline in which everyone needs to engage." Ah. So she's trying to say that people who could care less need to care more. Well, Joanna, good luck with that. But tell that to someone who's having trouble authenticating to their remote employer's VPN and who has no appreciation for the dangers that are lurking out there. It's just not going to happen.

Leo: I'm actually surprised it wasn't more of an apocalypse, to be honest.

Steve: Yeah, that's a very good point.

Leo: Yeah. I mean, I guess it is an apocalypse. But of course these people don't - they're completely vulnerable.

Steve: Well, and we're not talking about those ransomware attacks occurring 11 seconds; right?

Leo: Right.

Steve: They're getting into corporate networks somehow. And there has been a doubling of them in the past year. So, gee, maybe there's a correlation between all these employees having moved home. And so think about it, too, Leo. This is a survey from IT people who seem to have a clue about security. How many companies just said, oh, yeah, go to Fry's and get one of those VPN boxes and attach it to the network and let everyone connect? The point being there's no doubt easily half of the small offices and enterprises that had to send people home didn't have, don't have the ability, really, to bring full-strength IT security to the challenge. It just doesn't exist within a smaller group.

Leo: Yeah. I don't envy you guys. I know it's hard.

Steve: Thank goodness I don't have a big crew. I've got Sue and Greg, and they really - and even there, I will sometimes forward them some bits of these things just to keep them on their toes.

Leo: Yes.

Steve: Just to say this problem didn't go away, guys. So, yeah, knock on wood. As I've said, I no longer leave drives mapped to GRC's inner sanctum because, careful as I am, it's not even really that much anymore about being careful. It's like there's so much pressure to get in.

Leo: Yeah. A zero-day comes along, and you don't...

Steve: Right.

Leo: Yeah.

Steve: Yeah, it's zero for a reason.

Leo: Yeah, exactly.

Steve: So once again, quoting Bruce Schneier, attacks only ever get better. As we know, the beginning of 2018 was dominated by the concept, which is all it really was then, of Spectre attacks. Spectre, which was assigned CVEs 2017-5715 and 53, refers to a class of CPU performance optimizations which turned out to create previously unsuspected hardware-based vulnerabilities in many modern CPUs that break the isolation separating applications which allowed, theoretically, attackers to trick a program into accessing arbitrary locations associated with its memory space, thus abusing that program to read the contents of accessed memory, thereby potentially obtaining sensitive data.

Well, practical attacks are no longer theoretical. Academic researchers at the University of Michigan, University of Adelaide, Georgia Institute of Technology, and Tel Aviv University have designed a Spectre-based side-channel attack which can be weaponized to successfully overcome the Site Isolation protections Google added to Chrome and to the Chromium browsers to leak sensitive data in Spectre-style speculative execution attacks. They call it "Spook.js," "Spook" of course as in that original icon or the logo that Spectre had of being a ghost, so Spook.js. And of course "js," as the name suggests, a JavaScript-based attack.

The researchers said: "An attacker-controlled web page can know which other pages from the same websites a user is currently browsing, retrieve sensitive information from these pages, and even recover the user's username and password login credentials when they are autofilled." They added that an attacker could retrieve data from Chrome extensions, as well. Any data stored in the memory of a website being rendered, or a Chrome extension, can be extracted, including personally identifiable information displayed on the website, and auto-filled usernames, passwords, and credit card numbers.

Responding to this, Google said: "These attacks use the speculative execution features of most CPUs to access parts of memory that should be off-limits to a piece of code, and then use timing attacks to discover the values stored in that memory. Effectively, this means that untrustworthy code may be able to read any memory in its process's address space."

Chrome's Site Isolation rolled out in July of 2018, so like six months after we began talking about it in January, at the very start of 2018. And it was fully enabled, has been fully enabled, that is Site Isolation, since Chrome 67. Site Isolation, that group of technologies, were a series of software countermeasures designed to make these processor-based side-channel attacks more difficult to exploit. And we talked about these things at the time, you know, things like reducing the resolution, that is, increasing the granularity of the timer that could be read by JavaScript so it would be less sure about how much time had passed. When Site Isolation is enabled in Chrome 67 and beyond,

each website will be loaded into its own process. This would thwart attacks between OS isolated processes, and thus between sites.

But the researchers found scenarios where the site isolation safeguards do not separate websites and were able to get around the anti-Spectre protections. Spook.js exploits this design quirk to result in information leakage from Chrome and Chromium-based browsers running on Intel, AMD, and Apple M1 processors. So not just Intel.

Okay. So I dug into this a bit, and upon closer inspection it turns out that what Google did was only isolating at the second-level domain. So, for example, Chrome will separate 'example.com' from 'example.net' due to them having different top-level domains. And Chrome will also separate 'example.com' from 'attacker.com because those have differing second-level domains. But 'attacker.example.com' and 'corporate.example.com' are still allowed to share the same process. And it's this lack of absolute process isolation by domain that allows pages hosted under 'attacker.example.com' the opportunity to extract information from pages under 'corporate.example.com.'

Now, that may not seem like a big problem, but remember that, for example, on GitHub it is subdomains are used, subdomains under GitHub.com are used to create accounts. So, and this sort of thing happens many places. So it is the case that there are pages in a subdomain which bad guys could control the loading of JavaScript into which would then, even with Site Isolation in place, allow them to peek into places they should not go that they share with the same third-level domain as places where there are secrets.

So this was a very useful discovery since they were able to demonstrate through practical JavaScript-based attacks that the existing countermeasures that were in place are insufficient to protect users from browser-based speculative execution attacks. The Chrome Security Team's immediate response to this, and they received early access to the research last July, was to immediately extend Chrome's existing Site Isolation to ensure that browser extensions would no longer share a common process with each other. Which was interesting.

It was previously the case that extensions were all lumped together in their own process. They figured that was safe. Turns out no. You could get an extension that would launch speculative attacks against other extensions also being hosted by Chrome and which were sharing the same process memory. That's no longer the case. That Strict Extension Isolation is what got added. And since Chrome 92, that's been enabled by default. And the only downside is further proliferation of processes.

If anybody, by the way, has opened any modern browser and looked at what they are now doing in the Task Manager, there's like, there's a hundred Chrome.exes, you know, processes. And same thing for Firefox. It's like, oh, my goodness. But that's the price we're paying for having pushed - we saw first that browsers were just unable to protect themselves. And we talked about this not that long ago on the podcast, is that they basically turned over the responsibility to the operating system, leveraging the operating system's much more mature inter-process isolation. And also the processor or the OS running on the hardware processor does have like a stronger ability to enforce inter-process isolation.

So moving forward it's unclear whether Google will take further action, that is, this third-level domain problem is a problem. So they might feel that isolating at the second-level domain is sufficient. I doubt that's the case. The researchers explained that, as an immediate workaround: "Web developers can separate untrusted, user-supplied JavaScript from all other content for their website, hosting all user-supplied JavaScript at a domain that has a different top-level domain," that is, arrange not to share any third-level domain, like somehow get yourself a second-level domain and put stuff there. Like dot something else. Dot, you know, there's an amazing number of dot things out there

now. I just saw, was it Alex who uses dot something? I can't remember what it was. I just saw it on MacBreak Weekly.

Leo: You mean .dot? Dot D-O-T?

Steve: No, it was like - it wasn't, oh, well, we know that there's .coffee.

Leo: Oh, yeah, yeah, yeah. You were thinking of Mikah Sargent's chihuahua.coffee, yeah.

Steve: Right, right, right.

Leo: Oh, there's lots of them.

Steve: So you can get your own.

Leo: Yeah, TLDs, yeah.

Steve: Yeah, exactly. So you could get your own top-level domain in some other - or your own domain name in some other TLD.

Leo: Exactly.

Steve: And maybe split things up there. And that way Chrome would isolate you. But that's not your job. That's the browser's job. So I really do think that Chrome will end up probably stepping up.

Leo: I'm looking at all the Firefox processes running on my Linux box.

Steve: Oh.

Leo: One, two, three, four, five, six, seven, eight, nine, at least nine.

Steve: And how many tabs have you?

Leo: Very few.

Steve: Exactly.

Leo: I don't think it's the tabs at this point. It's one tab.

Steve: No, no, no. Yeah, well, no. If you are like me and have a hundred tabs, then...

Leo: Ooh, baby. I have child ID one, two, three, four, five. They're all child IDs. So, wow, there's just - so is that sandboxing the tabs? Is that what that is?

Steve: Exactly. It's giving each tab their own process.

Leo: It's own process, yeah.

Steve: And then Linux is doing the process isolation in order to keep the tabs from having any contact with each other.

Leo: Right, right. Which is great.

Steve: We hope.

Leo: We hope.

Steve: So apparently vacation is over. Emsisoft may have been the first to observe that REvil's "Happy Blog," as it's literally called, data-leaking site on the dark web had returned last week. At that point, their ransomware negotiation site had not yet been brought back online. But it has been now. BleepingComputer arranged to get some Russian postings translated into English. Remember that REvil's primary operator is known as "unknown," UNKN.

So the three paragraphs or sentences that they got translated read: "As Unknown (aka 8800)" - I don't know what that refers to. "As Unknown disappeared, we the coders backed up and turned off all the servers. Thought that he was arrested. We tried to search, but to no avail. We waited. He did not show up, and we restored everything from backups. After UNKN disappeared, the hoster informed us that the Clearnet servers" - Clearnet clearly being like the Internet the rest of us use, not Darknet, Clearnet is their term - "the Clearnet servers were compromised, and they deleted them at once. We shut down the main server with the keys right afterward. Kaseya decryptor, which was allegedly leaked by the law enforcement, in fact was leaked by one of our operators during the generation of the decryptor." And that was signed "REvil."

So this suggests that Unknown is the guy in charge. They're all keeping some arm's-length connections. I mean, maybe Unknown is actually unknown even to them; right? So it's like, you know, whatever they're doing, texting or using some secure comms or something in order to say, hey, where did you go? Are you still around? And they didn't know. They thought maybe he'd been arrested. Anyway, a dialogue apparently with this guy, Unknown, the REvil operator, that BleepingComputer was able to obtain reads this person saying: "Nothing happened. Took a break and now continue to work. I advise you to take breaks too," smiley face. Then the person who this conversation is with says: "Yeah, a break is always good. So are you an affiliate, or are you one of the REvil operators?" And the reply is "Operator."

So if we believe all this, the REvil operation just decided to take a break. I suspect that there's probably much more to it, but I'm pretty sure we're never going to know. So it's worth noting, however, and the reason I mention this news, is that the distressingly successful ransomware gang behind the attacks on JBS and Coop and Travelex, GSMLaw, Kenneth Cole, Grupo Fleury and others, of course, and of course Kaseya famously, is back in operation. Only time will tell what it means. And to me I thought the real useful takeaway here was let's be sure to note when speculation is all we have. You know, the whole security industry was like, whoa, REvil's gone. Oh, Putin must have stomped him out and so on.

We have no idea what's going on. We don't know, you know, we have no idea. So it's like, okay, well, they're gone. Good. And then we were looking, as you reminded us, Leo, for evidence that maybe some of the people had renamed themselves. And in fact maybe some did. This looks like the Unknown guy went offline. Some of his crew said, hey, where are you? Maybe they did sort of say, well, shrugged, this guy, you know, Unknown's unknown, and now his whereabouts are unknown. So let's go do something else. Who knows?

Again, it's like the tip of the iceberg is all we're seeing. We have no idea what the shape of the rest of it below the surface is. But again, initially the reports were that when their leaking site and their blog log of attacks came up, there was nothing new, nothing since I think it was July 8th was the last one, and they went dark five days later on the 13th. Now it looks like there is renewed activity. There is some leakage, for example, of new victims in order to demonstrate, I guess they're flexing their muscles. Yes, we really are back. So anyway.

Leo: Well, that's a great thing. I'm so happy.

Steve: Yeah, isn't that sweet. Isn't that precious. Okay. Two pieces of closing the loop. One, my talk with you, Leo, about assembly language a couple weeks ago, I guess it was just last week, it generated a lot of interest among our listeners.

Leo: Oh, good. I knew it would. People were interested, yeah.

Steve: I was surprised. David Cortesi, he tweeted: "'Zen of Assembly Language'" - that was the number one Michael Abrash text - "is hard to find on the used market, but it can be read free online at" - and there's a link in the show notes. It's somebody's GitHub page. And sure enough, the GitHub page says: "This is the source for an eBook version of Michael Abrash's 'Zen of Assembly Language: Volume I, Knowledge,' originally published in 1990 and reproduced with the blessing of Michael Abrash, converted and maintained by James Gregory." And James Gregory is the site at GitHub. It says: "The GitHub releases have EPUB and Mobi versions available for download."

Leo: They look really good, too. A lot of times you'll find these, and they're scans from the page, and they don't look good. This is - he did a good job of this. It's very clean, yeah.

Steve: Yeah. So he converted it to an HTML. So there's an HTML as a single page you can scroll through.

Leo: That's what I'm looking at, yeah. Looks great.

Steve: Yup. And there's also EPUB and Mobi versions. And even, like, all of the listings from the examples that Michael cites through his text are also there. And it's free. You can click on the link, and you'll be looking at an HTML page. If you don't want to spend much time on it, but you're even mildly curious, it's all there with Abrash's blessing. So very cool. Thank you for bringing that to my attention, David.

Leo: Yeah, yeah, really great. Wow. Makes me want to learn assembly language for about three seconds, yeah.

Steve: Yeah. I have, oh my god, I have a Picture of the Week that I've been meaning to get to. I will get to it. It's one of those spoofed O'Reilly covers.

Leo: Covers, yeah, yeah.

Steve: And it's "Web Dev in Assembly." And it says something...

Leo: Actually, believe it or not, there's something called WebAssem that is very popular right now, which is a low-level...

Steve: Well, because it's high-performance, yeah.

Leo: Yeah, yeah, yeah.

Steve: Anyway, the subtitle for "Web Dev in Assembly" says something like you might as well just shoot yourself right now.

Leo: Yeah.

Steve: I do all my web dev in assembly. I understand that I'm weird. Anyway, Mr. John Doe, as his Twitter handle goes, tweeted: "@SGgrc my Verizon G3100 router recently updated and now has both a Guest network, and a separate IoT network. I was astonished." He said: "I've heard you mention this type of network separation several times since I started listening to Security Now!. Thanks."

And I was curious, so I did a little googling. There is a Verizon FiOS G3100 Wireless 4-port switch GigE 802.11ax with MoCA 2.5 that looks - I'm just very impressed that a router would decide to update itself and add, like solicit the use of an IoT subnet for those devices. That's just, I mean, that's very, very cool.

Okay, now, Leo. This is where we get to whether you can eat doughnuts without any consequences or not.

Leo: Okay.

Steve: I have this next piece under Science Fiction, but is it fiction? The other day I googled "are we living in" that's as far as I typed. And I was frankly quite surprised when the suggested phrase completion offered was what I was planning to type anyway, which was "are we living in a computer simulation." As much as we all know I'm a science fiction enthusiast, it turns out that this is actually a question. Back in 2003, an Oxford University philosopher named Nick...

Leo: Bostrom, yeah.

Steve: Yes, Bostrom, published in the Philosophical Quarterly his paper titled "Are You Living in a Computer Simulation?" In that paper, Bostrom establishes a serious philosophical framework for considering the question. And while Elon Musk made news when he brought up his support for what's become known as "the simulation hypothesis," the well-known astrophysicist Neil deGrasse Tyson had the same question posed to him by NBC News, with Tyson giving it a "better than 50-50 odds" that the simulation hypothesis is correct. Tyson said: "I wish I could summon a strong argument against it, but I can find none."

And by the end of his carefully constructed and considered paper, which I won't go into here, but I have a link to it in the show notes, Nick Bostrom concludes that, if computer-using aliens exist, meaning if there are intelligent entities having the power to create computer-based simulations, then, Bostrom argued, "we are almost certainly living in a computer simulation."

Now, I don't want to spend too much time, too much more time on this, but I want to note that the question is truly being taken very seriously. On his podcast, StarTalk, when discussing this question, which was the topic of the entire podcast, Neil deGrasse Tyson observed that such a simulation would most likely create perceptions of reality on demand, rather than simulate all of reality all the time, in the same way that a videogame is optimized to render only the parts of a scene which are visible to the player.

Okay. And two last points. Last year in the "Space & Physics" section of Scientific American, a magazine that is serious - and Leo, you and I have a warm spot in our hearts for Scientific American. This issue or this article, "Space & Physics" section dated October 13, 2020, the article posed the question "Do We Live in a Simulation? Chances Are About 50-50." Link in the show notes for anyone who's interested in delving further.

Wikipedia has a page titled "The Simulation Hypothesis." And as I said, I have a link to Nick's paper in the show notes. There's even, for someone who really wants to delve into this further, a website, simulation-argument.com, which hosts all of the repercussions, on both sides of the argument, of Nick's paper which have been catalyzed by his publication of the paper, what, 18 years ago. If you think that the whole thing's just a big joke, you might want to spend a little time over at simulation-argument.com, where you will find some serious people giving this question some serious consideration.

Okay. Now, what got me onto this whole thing? It was the recently released trailer for the fourth movie in "The Matrix" series. One of the Wachowskis, Lana, is bringing back Neo and Trinity for another special effects extravaganza which opens Wednesday, December 22nd, and I can't wait. That'll be the day after our final podcast of 2021, and the trailer looks like it's going to be a lot of fun. And who knows? We all know how often science fiction predicts reality; right? So perhaps this is actually recursive. We're truly living in a Matrix-like pseudo-reality, and we're making movies about Matrix-like pseudo-realities.

Leo: On we go with however you spell it or say it. Meris.

Steve: However, yeah. It's Latvian for "plague."

Leo: I love that.

Steve: Meris. Meris. Meris. Yeah, we haven't talked about DDoS attacks for a while. But the recent series of headline-grabbing mega-attacks on the Russian multinational Yandex have recently been breaking all records to culminate in the largest single attack ever seen. And since the attacks originated from a new and frighteningly large botnet, which is apparently hosted by unpatched and compromised routers, and these attacks could be aimed at any target on the Internet, I thought we ought to check in this week and catch up on what's going on in big botnet land.

So just Yandex is large enough to probably be at least somewhat familiar to this podcast's followers, though perhaps more so outside the U.S. than in. It's one of the largest Internet companies in Europe, operating Russia's most popular web search engine, and Yandex is Russia's most-visited website. Its revenue last year was around $3 billion USD. And, you know, I could not resist converting that to rubles because you get about 1.4 cents per ruble. So they had revenue of 214 billion rubles. That's some serious rubles.

Anyway, it seems that apparently somebody has a grudge against Yandex, or perhaps they're just wanting to flex their new mega botnet's muscles to see what it can do. And that's actually my theory because the attacks have just been little pulses. When you think about it, I'm sure that someone running a huge botnet wants to know just how huge it is. But it's not really possible to know how powerful a botnet is unless it's turned loose against someone. And it's got to be somebody who just doesn't immediately collapse, but who is able, who's got defenses that are strong enough to be able to measure the incoming load on that network. And since the typical result of a large and powerful botnet being unleashed is the, as I said, the immediate meltdown of its target, you've got to choose a sufficiently large - aim your net at somebody sufficiently large.

Anyway, before we dig into the details, let's review a bit about the history and the various forms of denial of service attacks and distributed denial of service attacks because they didn't start out being distributed, and they didn't even start out being high-bandwidth. You didn't need much bandwidth. First of all, let me just say that this term "denial of service" (DoS) is itself extremely broad and generic. I'm always a little self-conscious when I use that in the other context because it's so strongly rooted in this notion of an Internet-based active attack of some kind, like something's really happening.

So when we've talked about newly discovered vulnerabilities which have not yet been weaponized into full attacks, the easier sort of junk attack is just to crash the service that receives some malformed packet. It takes much more skill to finesse that into executing the attacker's code. Way easier just to cause a crash. And unfortunately, that's also referred to within the industry as a denial of service attack, technically, because a crashed service will be denying its clients its service. So when we talk about it in the future, denial of service attacks, as being an exploit against something where one packet got sent, it's because it crashed something, not in the stronger original denial of service sense.

Okay. So DoS and DDoS attacks have evolved tremendously through the years. In every case we'll see that attacks can always be reduced to the consumption or overloading of some resource. That is, you know, these kinds of denial of service attacks. But before we get specific, we should briefly refresh our memory about the Open Systems Interconnection (OSI) model which describes and standardizes the roles of seven layers that computer systems use to communicate over a network. The OSI's meaning of "layer" will become a little more clear when we talk about what the layers are doing.

The model was first - it was the first attempt at any standardization of network communications, and it was adopted by all major computer and telecommunications companies back in the early 1980s. They were all hungry to adopt something, anything. There were no standards, as I said. And all we had back then was chaos. So this was needed, and it really served to bring some order to what was chaos. As it turned out, the Internet is only loosely based on the OSI model because it had no need for two of the upper layers. They're 5 and 6, as we'll see. But the others are all well identifiable in today's networking environment.

Layer 1 is known as the physical layer. It literally describes and specifies the voltages and currents and bitrates and bit encoding that will be used to form the bits for every layer above. So Layer 1 physical, the actual electrical layer, and the encoding of the bits.

Layer 2 is the data link layer. For most networks today, that means Ethernet protocol. That's where we have the 48-bit MAC addresses. And so Layer 2 is the formatting of the bits that Layer 1 tells us how to create in the first place. Layer 2 is the data link that says, okay, here's how we're going to move stuff, like the basic things around the network.

Layer 3 is the network layer. This is where IP lives, the IP protocol. And ARP, the so-called Address Resolution Protocol, is what's used to associate the IP addresses, IPv4 or IPv6, on Layer 3 to the MAC addresses, the 48-bit MAC addresses on Layer 2, so that you're able to route packets by IP, even though the routers route at the data link layer by MAC address. So you needed to have some glue between those two, and that's what ARP does.

The next layer up is Layer 4, the so-called Transport Layer. And this is where TCP, UDP, and ICMP and other transport layer protocols live. And just as Layer 3 provides the payload that Layer 2 transports, similarly, Layer 4 is the payload for what Layer 3 transports. In other words, TCP is transported by IP, which itself is transported by Ethernet. They're like nested envelopes. And so now we're at Layer 5 and 6, which don't exist in today's networking. They're the Session and the Presentation layers, and they just aren't used here.

The final layer, at the top of this OSI stack of layers, is 7, known as the Application layer. This is where protocols that run over TCP or UDP are found. So of course that would be HTTP, FTP, SMTP, DNS, RDP, all of those things that we talk about. They're being carried by TCP or UDP, which itself is at Layer 4, skipping 5 and 6. And so these application layer protocols are running on Layer 7. And we'll see that this ends up being a useful set of terminology. Both we and the entity that I'll be talking about think in terms of OSI layers.

Okay. So the very first attacks on Internet servers were, even though it's hard to imagine, low-bandwidth "SYN Trickle" attacks. Those could be generated by a single client on the Internet, which was able to simply send TCP SYN packets to a remote server. Just sending SYN packets, bloop bloop bloop bloop, not even very fast, would cause early TCP/IP protocol stacks in the Internet's web servers or whatever was answering TCP, so not necessarily web, FTP, SMTP, whatever, because in front of those protocol handlers - a web server, an FTP server, an email server - was the stack, the

operating system level stack. Sending just a series of SYN packets, not very fast, the stacks would take time and resources to get ready for connections to be made, which those TCP SYN packets were saying that remote client wanted to do. So the stack would allocate memory and get itself all ready for a conversation.

Not very long after those conversations never happened, that pending connection resource would fill up and run out, and new connection requests had to get ignored. Well, what that meant was that the real TCP SYNs from real users attempting to use the service couldn't get in. The stack was full. Can't accept any new connections. So in this way, just this simple SYN trickle, a server offering any form of TCP connection-accepting service would become unable to accept legitimate connections once all of its local connection-accepting resources have been tied up just by a single client that never had any intention of honoring its TCP connection requests. So the OSI model would place that attack at Layer 4 since it was specifically an attack against the server's implementation of its TCP protocol, and that's a transport layer.

Okay. Now, if that was the SYN Trickle attack, the next thing that happened was the SYN Flood. As we know, the Internet is essentially a data communications fabric, knit together by routers. Packets come in, their IP addressing headers are examined, and they're routed toward their destination by referencing a routing table. One of the parameters of routers is their maximum packet-handling rate. Packets on the Internet can be of variable length. But since packets have a fixed overhead for addressing, the maximum overall efficiency will be obtained using the longest packets which are practical since that will yield the greatest packet-to-header ratio.

Since it's unusual, therefore, for routers to encounter a high percentage of small packets, because those are very inefficient, the performance of a router's routing switch fabric will be scaled to handle the average packet rate that might be presented to it through all of its incoming interfaces. And this explains the success of SYN flood attacks. TCP SYN packets are among the tiniest packets possible. Whereas a typical Internet packet will be 1514 bytes of payload, 1514 bytes, TCP SYN packets weigh in at just 60. This means that 25 TCP SYN packets can fit into the physical space occupied by one normal-size IP space.

But physical space is the same as temporal space when you're talking about data flowing. So it also means that if nothing but TCP SYN packets are sent to a router, they can arrive at a rate that's 25 times higher than typical Internet packets. Since every one of those SYNs are valid IP packets, each one must be examined and routed to the proper interface.

But few routers are able to handle 25 times the routing rate demand that they were designed to handle. Routers have short buffers which are used to even out the incoming flow. They queue packets briefly, as needed. But when a router's switch fabric is overloaded, those buffers will quickly overflow with the majority of packets being dropped and lost forever. The original senders of legitimate packets will then retransmit the dropped packets, which just makes things worse. And as a consequence, the packets of valid traffic will be very likely to be dropped.

So once again we have a denial of service. Valid users, would-be users of the service that the router is in front of are unable to even reach the server. And in this instance, the resource being consumed by the TCP SYN flood is the network's routing capacity near the targeted web server. It's where traffic from across the Internet, in the case of a distributed denial of service attack, traffic is coming in from many different points across the Internet. And with each step through a router, it generally becomes increasingly concentrated, and at some point it gets aggregated enough to begin collapsing the routers that are near to the target.

As a consequence, the targeted web server never even experiences most of the attacking traffic because it can't even reach the server, and of course neither can legitimate traffic. So this type of SYN flood would be an attack that leverages the IP protocol, or a weakness in the routing of IP protocol, so that would be an attack at Layer 3.

TCP SYN flooding attacks are typically generated, as I've mentioned and we've talked about, by a fleet of bots widely spread across the Internet, each aiming at the same server and, over time, aggregating their traffic, thus a DDoS attack, a distributed denial of service attack. Now, over time, because these attacks have been going on for decades, TCP/IP stacks were redesigned to tolerate the Layer 4 SYN trickle attacks. It turns out it's possible to start a TCP connection statelessly, meaning without storing any local state. So it's possible to honor the TCP/IP protocol and not commit any resources from the receipt of a TCP SYN packet, still be able to emit a SYN/ACK, and then wait for the final, the third packet and a handshake, that ACK.

And actually it's by taking that ACK and the things it contains that you're able to quickly scurry around. You know it can't be a spoofed source IP because the sender had to, or the originator of the SYN had to receive the SYN/ACK and send back an ACK to you for the SYN/ACK that was sent. So anyway, it solves the problem in a clever way. And Internet routers were also upgraded so that they would be able to handle the high rates of small packets at Layer 3 without collapsing. Okay. Again, cat and mouse. More brute force methods needed to be devised.

Someone noticed that a very small query to a DNS server could result in a very large reply. Since DNS servers reply to the IP that queried, the Unix raw socket API was used once again to spoof the source IP of the query. And thus were born the so-called reflection/amplification attacks. Even one determined attacker with a high-speed connection might bring down a large website. The attacker would send tiny DNS queries as fast as their connection would allow. And because the queries were small, that meant they could get a lot of them out per second. Those would be sprayed all across the Internet's publicly available DNS servers.

So the DNS servers weren't being deluged because there are many of them, and it's easy for an attacker to spray their tiny little queries at high rate across many DNS servers, so the rate per server is kept low. In every case, that attacker would spoof their source IP to being that of the target server, which would cause all of those DNS servers who thought they were receiving a legitimate DNS query to send their much larger reply packets to that single spoofed IP. In this case it wasn't a router's switching engine fabric which would be overwhelmed. It was the total bandwidth of the site's connection to the Internet.

Most web surfing traffic is very bursty. We look at a page for a while before we click another link on the site. And many times the site doesn't have what we're looking for, so we'll just hit our browser's back button to step back to wherever it was we came from. And even when we do load a page, these days a huge percentage of a website's total page content is being delivered by other Internet web servers. Right? All these other content providers that are causing us so much grief most of the time.

The result of this natural burstiness and this source distribution of web traffic is that sites are not scaled to handle massive amounts of continuous traffic because that never happens unless the site is being subjected to a significant bandwidth attack. All that's needed is to overwhelm a website's scaling for normal daily traffic. And it turns out that's not too difficult to do. When aided by raw socket IP address spoofing, and servers like DNS that can be induced to amplify Internet traffic, it's often feasible to readily overwhelm a site's connection bandwidth. But over time this, too, was overcome by aggregating many sites to share much larger Internet connection bandwidth. 100 websites might share 100 times the bandwidth.

Individually, the economics still make sense, and they're still getting the same amount of bandwidth. But by pooling their individual bandwidths to create a much larger shared bandwidth aggregate pool, they're now able to transiently borrow from that pool as needed to thwart large bandwidth attacks while still remaining on the air. Cloudflare and other large providers have been quite successful in offering anti-DDoS protection by fronting for their website clients.

But today's modern websites are dynamic. They do not deliver static HTML pages which sit as text files on nonvolatile disk storage. When you see a site whose page URLs end in .php or .asp or .jsp, you're seeing a page which doesn't actually exist anywhere as an HTML web page. Instead, it was created by invoking a script. Many sites like those created by WordPress, any modern shopping or catalog site - eBay, for example, or Craig's List - or any web forum, only exist as collections of virtual pages which are created on the fly, on demand, from a collection of SQL database queries admitted by PHP, Active Server Pages, or Java Server Pages. And then the results of those queries are knit together to produce finished HTML.

So it turns out that running that scripting, serving those SQL database queries, and assembling those finished pages can be quite compute intensive. It's neat and flexible to be assembling pages on the fly from templated style sheets, but this approach inherently introduces a new bottleneck which can be attacked and which may prove significantly more difficult to filter and block. Whereas yesterday's attacks, being measured in packets per second or bits per second, were "transport layer" capacity attacks of one form or another, and of course those are referred to as Layer 3 or 4 attacks, the newest form of attack is an application layer capacity attack where the attack strength is measured in requests made per second (RPS). Similarly, these are often referred to as the OSI Layer 7 (application layer) attacks.

And this brings us to the topic of today's podcast, this Meris Botnet. We started this discussion by talking about Yandex. Like many large firms, they have their own internal network security people, while also employing the talents of external security firms who have an inherently broader scope and are able to provide more experience and specialization as well as network services to their clients. Yandex's exterior partner is Qrator Labs, spelled Q-R-A-T-O-R, Qrator Labs.

Qrator Labs is a big European Internet security and attack mitigation company. They maintain offices in Prague, which of course is in the Czech Republic; Dubai in the UAE; and in Moscow. Their page on DDoS attack prevention, which is a service they offer, they explain: "Automatic DDoS mitigation at all OSI levels up to L7 (application level) inclusive, on all the tariff plans, with no exception.

"All layers of protection solutions by Qrator Labs solution work together in a connected complex. This approach serves as base for neutralization of even the most complex attacks, which sometimes combines DDoS attacks on channel capacity with the attacks on the layer of web applications (L7), frequently accompanied by hacking.

"The system blocks not only high-speed network layer attacks (L3 and L4), but also low-frequency destructive L7 attacks which can only be identified through behavioral or correlational analysis and continuous traffic monitoring. The speed of reaction to DDoS attacks is reduced to the absolute minimum as the system works in fully automatic mode, which also ensures the uptime of clients' web resources 24/7."

Okay. So that's how they're describing themselves and one aspect of many services that they offer. Last Thursday, Qrator published a blog posting that surprised and worried many in the Internet community and generated many headlines. Its title was "Meris botnet, climbing to the record." And they wrote: "For the past five years, there have virtually been almost no global-scale application-layer attacks. During this period, the

industry has learned how to cope with the high-bandwidth network layer attacks, including amplification-based ones. It does not mean that botnets are now harmless.

"End of June 2021, Qrator Labs started to see signs of a new assaulting force on the Internet, a botnet of a new kind. That is a joint research we conducted together with Yandex to elaborate on the specifics of the DDoS attacks enabler emerging in almost real-time. We see here a pretty substantial attack force, dozens of thousands of host devices, growing. Separately, Qrator Labs saw the 30,000 host devices in actual numbers through several attacks, and Yandex collected the data of about 56,000 attacking hosts."

Now, I need to pause here for a moment to mention something that I forgot to. And that is the issue of identifying the attacking devices. In DNS, for example, reflection attacks against DNS servers themselves which appear to be swamping the target with unasked-for and unwanted DNS queries, the queries appear to be coming from the target. The fact is, they're being spoofed by one or more attackers generating, sending those queries out to DNS servers. So there's no way to identify by the payload in the packets, the IP addresses, who is actually behind this, what their actual IP is. And anytime IP addresses are being spoofed, that's the case. We just don't know how to identify them. But this is never the case whenever an full TCP connection is completed.

And I was just talking about that with respect to the stateless TCP connection initiation protocol. Any time you have a Level 7 attack, it is over TCP. The IP of the request that's coming in is somebody who's making that request. You could have proxies in the way. That's possible. But you still have an IP address that you know of that, for example, you could block if you had to. So in order for a malicious client to attack a website by issuing a web query, the site must receive as part of that query a valid IP for the other endpoint of the conversation. Layer 7 attacks cannot be blind.

Okay. So when they say, "We see here a pretty substantial attacking force, dozens of thousands of host devices growing separately," blah blah blah, 30,000 in one case, Yandex saw 56,000, they actually have enumerated those devices' IP addresses. So then they continue: "However, we suppose the number to be higher." And we get to that gruesome reality in a minute. They said: "Probably more than 200,000 devices, due to the rotation and absence of will to show the full force attacking at once. Moreover, all those being highly capable devices, not your typical IoT blinker," as they put it, "connected to WiFi, here we speak of a botnet consisting of, with the highest probability, devices connected through the Ethernet connection," meaning hard-wired network devices primarily.

They wrote: "Some people and organizations already call the botnet 'a return of Mirai,' which we do not think to be accurate. Mirai possessed a higher number of compromised devices united under command-and-control, and it attacked mainly with volumetric traffic." Meaning Mirai was a flooding botnet.

They said: "We've not seen the malicious code, and we are not ready to tell yet if it is somehow related to the Mirai family or not." And actually they said somewhere - oh, in fact it's here. I'll get to it in a second. "We tend to think that it is not, since the devices it unites under one umbrella seem to be related to only one manufacturer, MikroTik." They said: "Another reason we wanted to name this particular botnet, operating under elusive command-and-control" - which was interesting - "with a different name, Meris, which means 'plague' in Latvian. It seems appropriate and relatively close to Mirai in terms of pronunciation." So, okay, both of them begin with "M" and have five letters.

Okay. Specific features of the Meris botnet: SOCKS4 proxy at the affected device maybe. That's unconfirmed, although MikroTik devices do use SOCKS4. The use of, and this is going to turn out to be very significant, HTTP pipelining, which is to say HTTP/1.1, and we know HTTP/2 does it also, which is a technique for further amplifying DDoS attacks.

That's confirmed. Making the DDoS attacks themselves RPS, that is to say requests per second, based. That's confirmed. And open port 5678 is confirmed.

They said: "We do not know precisely what particular vulnerabilities lead to the situation where MikroTik devices are being compromised on such a large scale. Several records at the MikroTik forum indicate that its customers experienced hacking attempts on older versions of RouterOS, particularly 6.40.1 from 2017. If this is correct, and we see that old vulnerability still being active on thousands, actually hundreds of thousands of devices being unpatched and unupgraded, this is horrible news. However, our data with Yandex indicates that this is not true because the spectrum of RouterOS versions we see across this botnet varies from years old to recent. The largest share belongs to the version of firmware previous to the current stable one." In other words, not four years ago, but very recent.

And I didn't put - they had a big pie chart showing all of the RouterOS versions. Clearly, Yandex has a strong interest because they're receiving this attack traffic; they're receiving the attacking IP. They can probe that IP. And it turns out that RouterOS answers with its version number. So they've got a complete histogram breakdown of the RouterOS versions that are attacking them, and it looked like all of them to me. I mean, it's like, not a few.

They wrote: "There's a suggestion that the botnet could grow in force through password brute-forcing, although we tend to neglect that as a slight possibility. That looks like some vulnerability that was either kept secret before the massive campaign's start or sold on the black market." In other words, there's a vulnerability nobody has known about. These guys bought it or discovered it, and they leveraged it to grow themselves a botnet of serious size and capability.

They said: "It is not our job to investigate the origins, so we must move on with our observations. In the last couple of weeks, we have seen devastating attacks toward New Zealand, the United States, and Russia, which we all attribute to this botnet species. Now it can overwhelm almost any infrastructure, including some highly robust networks. All this is due to the enormous requests per second power that it brings.

"It's been in the news lately about 'largest DDoS attack on Russian Internet and Yandex,' but we at Yandex saw a picture much bigger than that. Cloudflare recorded the first attacks of this type. Their blog post of August 19, 2021 mentioned the attack reaching 17 million requests per second. We observed similar durations and distributions across countries and reported this information to Cloudflare."

I have a graph in the show notes showing the single largest attack ever recorded, which occurred on September 5th, 2021, so that was Sunday before last. They show a prior history on 8/9 of this year, 5.2 million requests per second. I'm sorry, on 8/7. Two days later, on 8/9, the 5.2 had grown to 6.5 mrps. Twenty days later, on 8/29, that was up to 9.6 mrps. Two days after that, on 8/31, 10.9 mrps. And that September 5 attack, Sunday before last, was the largest ever seen at 21.8 mrps. Okay. So 21.8 million HTTP valid GET or POST requests, each of which is going to attempt to invoke a script which will be interpreted on a web server, causing it to generate multiple SQL database queries in order to dynamically build a web page which the request has asked for.

I mean, it's just not possible. It's not feasible to imagine a backend that is able to handle that because nothing in nature generates 21.8 mrps, or even close to it. My little bandwidth at Level 3, I have a 100Mb connection, you know, 10Base-T, or 100Base-T. You know, and it just purrs along at a few megabits, typically, and every so often somebody downloads something, and there's a little spike in the bandwidth. That's what Internet traffic typically looks like. Nothing like this. I mean, not that I have a huge site. I'm sure Yandex's site, the most busy page in Russia, is way busier. But nothing like this.

Leo: And this is a zero-based graph, which I like.

Steve: Yes, yes.

Leo: Starts at zero, goes to...

Steve: And actually, thank you, Leo, I'm glad you mentioned it. Zero-based, and notice what the line looks like normally. You can see what their request per second was before this happened. We don't have enough resolution in the scale to estimate. But it's like nothing.

Leo: Nothing.

Steve: Compared to this 21.8.

Leo: Wow.

Steve: So Yandex's security team members managed to establish a clear view of the botnet's internal structure. You can imagine, they're desperately going to be poking and probing the things that are attacking them, trying to figure out. There are...

Leo: They mitigated it pretty quickly. I mean, it looks like it only lasted a few minutes.

Steve: Oh, no, no. That wasn't mitigation. That was a probe.

Leo: Oh. That was just a single thrust.

Steve: Yes. It was not mitigated. It was, yes, that was a single thrust.

Leo: Oh, okay.

Steve: That was so that...

Leo: That was just a couple of minutes on that one.

Steve: The attackers did that so this posting would result that would tell them how good their botnet was.

Leo: Oh, great. It's good, gang. Well done.

Steve: Yeah, stop. Good enough. Stop. You don't need any more. Okay. So the attacking source's IP addresses, which were not spoofed, seem to be predominantly the same trait. They have open ports 2000 and 5678. And they wrote that of course many vendors put their services onto those particular ports. However, the specific combination of port 2000, which was "Bandwidth Test Server," and port 5678, which is the "MikroTik" - and I remember I used to pronounce or mispronounce it my-crot-ic, Leo. Now I'm...

Leo: Which is probably right.

Steve: Correct pronunciation, yes. The "MikroTik Neighbor Discovery Protocol," they said, "makes it almost impossible to ignore."

Leo: Oh, great.

Steve: "Although MikroTik uses UDP for its standard service on port 5678, an open TCP port is detected on compromised devices. This kind of disguise might be one of the reasons devices got hacked unnoticed by their owners. So based on this intel," they said, "we decided to probe the TCP port 5678 with the help of Qrator.Radar. The results we have gathered were surprising and frightening at the same time." It turns out that there are 328,723 active hosts on the Internet replying to the TCP probe on port 5678. They said of course not necessarily each of those is a vulnerable MikroTik router. There is evidence that Linksys devices also use TCP service on port 5678. There may be more.

But at the same time, we have to assume that this number might represent the entire active botnet. 42.6% of those IPs are located in the United States. Just shy of 140,000 of those are MikroTik routers here. China is second, with just shy, like six shy of 62,000 of those at 18.9%. Then we have Brazil at 9,000, Indonesia at 7, India at 6, Hong Kong at 5. And then Japan, Sweden, and South Africa all with just shy of 5,000, and about 1.5%. So, okay. The pipelining that they got, that they mentioned before, right, they talked about HTTP/1.1 using pipelining. We've talked about this. This is a big concern. This indicates that these evil bots were well-designed for this application.

We've talked about the advances in HTTP that allow for multiple outstanding requests to be pipelined, meaning that multiple queries can be sent to the server for it to handle an answer as it sees fit, even if they're out of order. For example, there might be one query that is going to take it awhile, yeah, like any of these things using active content, whereas you might be also asking for a GIF, which is tiny, like the favicon, in which case you might ask for that third, but the server gets them all, and it sees one it can deal with quickly. So while the active content engine is busy trying to build a page, it sends the favicon back out to you. That's the beauty of pipelining HTTP.

But by using pipeline requests, we have something that's vaguely reminiscent of the earlier Level 3 and Level 4 flooding attacks, but far more devastating and difficult to block. The only upside is that the attacking IP addresses cannot be spoofed, so they could theoretically be blocked. But blocking a list of 328,723 remote IPs alone comes with its own challenges. How exactly does one do that? That's a lot of discrete IP addresses. And they're not all lumped together in one network where you can block them off by network. They're just everywhere.

Leo: That's why it works so well.

Steve: Exactly. And these are also HTTPS requests.

Leo: Oh, wow.

Steve: Meaning that no external agency is able to see into the queries until the targeted website provides their TLS certificate so that the filtering agency can accept and terminate them.

Leo: So Cloudflare, for instance, couldn't mitigate. What is the drop in the spike? Burke wants to know why there are two points on the spike.

Steve: You sort of see that from time to time in attacks.

Leo: They rev it up, and then it pauses, and then it goes.

Steve: Yeah. Or something crashed and then managed to get itself back online. I mean, these things are really being brutal to all the equipment that is involved.

Leo: It's amazing, yeah.

Steve: The other clever thing that HTTP pipelining brings is a reduced need for TLS connection setup. If multiple requests are sent down a single TLS connection, less time is spent establishing a connection. And the other confounding thing is these are not invalid requests. You can't, like, deny them or block them because they're bad. They're valid. So they look just like the requests that the site is trying to simultaneously honor from non-attacking IP addresses. I mean, it is really a mess.

Leo: Wow.

Steve: Yeah. So what we've watched over the years is a gradual evolution in attack technology. In those quaint old days, source IPs were being spoofed to hide the attacker's identity when trickling TCP SYN packets into a server to successfully bring it down. Then later, source IPs were being spoofed to bounce reflection and amplification attacks off of other machines on the Internet, causing them to indirectly attack the target with a pure bandwidth flood.

But today we have such a large volume of vulnerable Internet appliances deployed globally that it's not necessary any longer to hide. And these RPS (requests per second) attacks cannot be spoofed. They're also dauntingly difficult to block because, as I said, each individual request is valid. And it's only when a given IP makes too many requests within a short time that the client begins to look like it's probably a bot. But think about that, like what you would need in order to block this. You'd need to have something logging the IPs, counting each of the - and once upon a time in the quaint old days we only had 4.3 billion because it was a 32-bit number. So you could easily lay out a big RAM map of counters and count up how many queries you were getting by IP, and the rate at which that was happening. And at some point, if it hit a limit, you'd add that IP to the block list.

But now we have IPv6 and 128-bit IPs. So you're going to need a data structure in order to arrange to determine the rate at which individual IPs, 128-bit IPs, are making queries, and you're going to have to answer a bunch of them because they might be valid until you decide that they're not. I mean, and most people don't have any of this. This is serious next-gen "protect yourself from the worst botnet that we've seen so far technology" that most sites don't have. Which means anywhere this horrible thing is aimed is just dust. Just poof, just gone from the Internet.

Leo: Just amazing.

Steve: Just a crater left behind with some lost bits in it. So someone somewhere has built, assembled, and is in control of a horrifically powerful botnet, unlike anything seen before. It consists of upwards of a quarter of a million MikroTik routers, nearly half of them with IP addresses in the United States. And as I said, I'm sure those brief 60-second attacks against Yandex weren't meant to harm them. They were meant to give the attackers some sense for the scale of this new offensive weapon they have created. So they must be feeling quite pleased with themselves now. Nothing can withstand 21 million web requests per second.

Leo: And it still - the botnet still exists; right? I mean, it's just a matter of where it's aimed next.

Steve: Yup.

Leo: It's like the Death Star. This has been a demonstration.

Steve: Yes.

Leo: Wow. Scary thought. Is there a patch MikroTik's sent out?

Steve: Well, if you look at the distribution from their page of RouterOS versions, it's quite clear nobody ever...

Leo: Nobody cares.

Steve: ...patches their MikroTik routers. I mean, it looks like the entire history of - basically the RouterOS version that the device is sold with is the one that's running today.

Leo: I think that's true for all routers.

Steve: Yes.

Leo: I mean, we would hope for better, but I'm afraid that's true.

Steve: I know. And so what that means is that somebody discovered a way of hacking a MikroTik router which has been vulnerable since day one and is vulnerable still.

Leo: Yeah, because there are some recent versions, too; right? I mean, it's not...

Steve: Yes, yes, yes. The one before the current stable release is a vast number of infected machines.

Leo: Golly.

Steve: What's unfortunate is it's a shame that so much - think about the industry that's now going to have to go into providing protection from this. So much complexity and cost is being added to what was originally a beautifully simple system, the IP-based Internet that we all love. And it's all just to protect it from abuse. And abuse aided not only by people who want to abuse, who are hiding behind anonymity because if they didn't have anonymity they'd just get arrested, and the Internet provides that to them also as a consequence of this technology.

Leo: Oh, yeah.

Steve: But also unfortunately this is what I talked about when the other day somebody liked my quote and tweeted it back to me when I said "We are filling the world with a bunch of complex crap."

Leo: Complex hackable crap. But it just underscores why there aren't - people don't rob armored cars anymore, because why do something so high-risk when you can make big money without any risk at all?

Steve: You got doughnut crumbs there, Leo?

Leo: Yeah, there's a few. Gotta get the waiter in here with his special doughnut crumb cleaner.

Steve: No, but you're right. Go cyber and go free.

Leo: It's unbelievable. It's not, by any chance, one of those router exploits that you can reboot the router and clear out of memory?

Steve: Given that, I mean, well, okay. How often are routers rebooted?

Leo: Never, never. I'm just saying, you remember the FBI put out an alert a few months ago saying everybody reboot your routers because...

Steve: Right.

Leo: Maybe it was Mirai. It could be wiped out of memory.

Steve: Well, you could almost imagine, like, some, again, science fiction, where we're going to turn off the global electricity.

Leo: Ha ha ha, reboot everything.

Steve: Everything. The only way to get out of this is we're going to shut down the world. Everybody get out of your cars, turn those off because they're rolling computers now.

Leo: That's hysterical.

Steve: We're going to turn off the power by universal agreement everywhere, probably not North Korea, but everywhere else. Actually, I don't know...

Leo: That would be a good movie. You need to write that script. That's great. "The Day the Earth Went Dark."

Steve: "The Day the Earth Rebooted."

Leo: Yeah, rebooted. Reboot. Project Reboot Earth. I think that'll be - I'd go see that movie. Wow. Fascinating, as always. By the way, I guess MikroTik is Latvian, which explains the reason for the use of Meris. That's according to Jason Nash in our chatroom.

Steve: And you're right, I do remember that now that you say it.

Leo: Yeah, that would explain it, yeah. It would also explain why this affected Yandex more than, say - I mean, I know it's been seen elsewhere. But it's Eastern Bloc kind of more. I guess.

Steve: Yeah. Well, you blast somebody complete into oblivion in New Zealand, and they're not going to be able to tell you, like, how much. They're just going to be, like, what happened?

Leo: Wha? Something went wrong. Something went terribly wrong.

Steve: Yeah, they're not counting packets. They're trying to pour water on their servers.

Leo: It really is kind of the Death Star of attacks.

Steve: Yes.

Leo: It's pretty, you know, you're nuking it from space. All right. This show appears on your computer once a week, thanks to the great and generous Steve Gibson. He's at GRC.com. He has copies of the show on his server, 16Kb audio, 64Kb audio, and transcripts. Go to GRC.com. While you're there pick up a...

Steve: Ah.

Leo: What?

Steve: It stands for Generous Research Corporation.

Leo: The Generous Research Corporation. I like it. You also should pick up a copy of SpinRite, the world's best mass storage recovery and maintenance utility. Current version 6.0; 6.1 is on its way. The momentum is building, and you could participate in its development. And you'll get a free copy, too, for that matter, if you go right now to GRC.com and buy yourself a copy of SpinRite.

We also have 64Kb audio and video at our site. If you want to see Steve's shining face, all you have to do is go to TWiT.tv/sn.

Steve: My shining forehead.

Leo: You can also get it on YouTube. There's a YouTube channel devoted to Security Now!, and of course you can also get the podcast. If you subscribe, you get it automatically. Just subscribe in your favorite podcast client. Do us a favor, though. Spread the word. Leave a five-star review so others know about the great resource that is Steve Gibson and Security Now!.

We do this show every Tuesday, right after MacBreak Weekly. That's usually around 1:30 Pacific, 4:30 Eastern, 20:30 UTC. You can watch us do it live. There are live audio and video streams at TWiT.tv/live. People watching live like to chat live. You can do that in two ways. There's the free chat at irc.twit.tv.

Steve, have a wonderful week. We'll see you next time on Security Now!.

Steve: Thank you, my friend.