Anatomy of a Ryuk Attack

Description: This week we examine the coming controversial changes to the WebExtensions API. We look at the revelations and fallout from last week's Patch Tuesday, and at Zoom's latest announcement of this week's rollout of end-to-end encryption. We make sure everyone knows about the latest horrific SonicWall vulnerability and Microsoft's pair of not-that-worrisome, out-of-cycle patches. We share a bit of miscellany and closing-the-loop feedback. Then we examine an actual Ryuk ransomware intrusion and attack, step-by-step.

High quality (64 kbps) mp3 audio file URL: http://media.GRC.com/sn/SN-789.mp3
Quarter size (16 kbps) mp3 audio file URL: http://media.GRC.com/sn/sn-789-lq.mp3

SHOW TEASE: It's time for Security Now!. Steve Gibson is here. We're going to give you the anatomy, step by step, of a malware infection, a ransomware infection. That's fascinating. We'll talk about the new Edge extension specification. Is it good for ad blockers? We also have some information on Windows 10 God Mode. I'll be installing that as soon as I install Windows 10. And Zoom crypto, have they finally got it right this time? It's all coming up next on Security Now!.

Leo Laporte: This is Security Now! with Steve Gibson, Episode 789, recorded Tuesday, October 20th, 2020: Anatomy of a Ryuk Attack.

It's time for Security Now!, the show where we gather together to celebrate security and privacy online. It's kind of Thanksgiving for security every Tuesday with this guy right here. I'm the turkey; he's the stuffing. It's Mr. Steven Gibson.

Steve Gibson: We're celebrating our continued existence.

Leo: Yes. It seems in some doubt.

Steve: By hook or by crook, against all odds, yes.

Leo: Hi, Steve. How are you?

Steve: Great. Great to be with you again for Episode 789 for this lovely October 20th of 2020.

Leo: Seven ate nine. That's why we went right to Windows 10, actually.

Steve: What?

Leo: Because seven ate nine.

Steve: We did skip 9. Did we ever get a reason for why 9 got passed over?

Leo: Because seven ate nine. Get it?

Steve: What? No.

Leo: You obviously never went to high school. All right, all right, all right. He skipped straight to college.

Steve: In following up on some interesting news, I discovered a beautiful forensic analysis of an actual ransomware attack where the Ryuk gang, also the Ryuk ransomware was employed - and isn't that also Sodinokibi? There's so much of it now, I'm getting kind of confused - was used to attack an enterprise. And we have exactly how it was done, with the tools that were used, how the person jumped from machine to machine. I just thought, okay, this is just too cool to pass up. So I think a really interesting deep dive into a ransomware attack.

Leo: And a cautionary tale, I mean, now that we can follow the tracks.

Steve: Oh, it's a little chilling, actually, when you actually - it's one thing to just say, oh, yeah, they got attacked. But, boy, when you look at what they did, it really helps to bring it home. But we're first going to look at some of the controversial changes coming to the WebExtensions API, which actually is now two years old. That is, Google announced their intentions two years ago. And Microsoft just said, yeah, we're going to be - we're, like, rolling that out this week in Edge. We're also going to look at the revelations and fallout from last week's Patch Tuesday. I enjoyed hearing Mary Jo say, yeah, it was only 87, you know, not such a big deal. But it was like, wait, we used to have 10. There used to be 10 problems.

Leo: Well, and Microsoft had an out-of-band update this week, too. So, I mean, crikey, yeah.

Steve: Yup, we're going to get to that, as well. We also have again, you know, I just - I've decided, Leo, that we need the CEO of Zoom not to be quoted during any of their press releases because he just - everything's cruising along fine, and then he says something.

Leo: And you go, what? Wait.

Steve: It's just like, oh, no. Well, just go take your stock options and stay away from the PR people because it was all great until he opened his mouth again. But they're still trying with this week's rollout of end-to-end encryption. We also want to make sure that everyone knows about the latest horrific SonicWall vulnerability, the second bad one this year. This one's really bad. And we also got the not-that-worrisome out-of-cycle patches from Microsoft, a little bit of miscellany, a little bit of closing-the-loop feedback. I found a really cool utility for Windows 7, 8, and 10. I don't know what happened to 9. Again, this is Security Now! 789, but I'm still confused about where Windows 9 went.

Leo: You'll wake up in the middle of the night, and you'll get it, I'm telling you.

Steve: Okay. I don't think so. And then we're going to examine this really cool analysis, the forensic walkthrough of a ransomware intrusion and attack, which concludes with every machine in the enterprise deeply encrypted. So yes, I think another great podcast for our listeners.

Leo: It is going to be barrels of fun and educational, to boot. Steve, I saw somebody tweet you this Picture of the Week. I was hoping you would use it. I love it.

Steve: So this is two photos from scenes from "Star Trek: Next Generation." The first one shows the bridge. And Data and Wesley are there at the front consoles, and we have Will and Deanna Troi bracketing Picard, and Worf in the background at his security station. Everybody who knows STTNG is aware of that familiar setup. And this shows Picard giving Data an order. He says: "Mr. Data, please bring the Windows 10 updates completely online."

Leo: Uh-oh.

Steve: And of course things don't go well, and the next frame is like everything has exploded on the bridge. We have sparks flying everywhere. I don't know what happened to Deanna. I think she already got thrown out of her chair.

Leo: I can even hear in my mind the [voicing alarm sounds]. I can hear it.

Steve: Yes, yes.

Leo: That's hysterical.

Steve: So, yes, the fact that people are creating these...

Leo: It does say something, doesn't it.

Steve: ...images just sort of tells you what the industry - about how badly the industry has been abused by Windows 10 and its updates. It's like, oh, my goodness. Well, okay. Edge is going to be updated with browser extensions known as Manifest V3. The

proposed changes to the WebExtensions API, which are sort of generically known as Manifest V3, or I guess shortly known, were first announced by Google two years ago, back in October of 2018. And this was for Chromium. This is what Google said, "This is what we're going to do." And we talked about this at the time.

Our listeners may remember these stated plans from Google did not go over very well with the industry. When they announced their planned changes, they explained, Google explained that the main intent of this Manifest V3 was to improve extension security, improve extension performance, and give users greater control over what extensions did and which sites they could interact with. Which, you know, all sounds great. But extension developers quickly pointed out that these Manifest V3 updates contained changes which would cripple the ability of ad blockers, AV, parental control enforcement, and various privacy-enhancing extensions to do their job as they had been.

And as a consequence, Google's announcement triggered a significant backlash from users, extension developers, and even other browser makers because, among other things, the extensions had the effect of limiting the power of ad blockers to block ads. Of course the non-Google community was unhappy to see Google, clearly an advertising-based company, moving to limit our ability to control the ads that our browsers would be subjecting us to. And as I've often mentioned, from time to time I will encounter a browser lacking a competent ad blocker. And I'm always shocked by the experience. I think, wait, whoa, buckle up. It's just - it's horrific. So I can imagine choosing a browser entirely based upon whether or not it allowed me to have control over just how obnoxious the ads were that I was being served.

And back at the time, browsers including Opera, Brave, and Vivaldi quickly distanced themselves from Google's plans, announcing their intentions to ignore these Manifest V3 updates and thus allow users to keep using the ad blockers that they already were using and liked. And Mozilla, which had implemented the WebExtensions API up to that point in Firefox in order to get compatibility with where the rest of the industry was going, also explicitly denounced Chrome's plans and said it would not be following Google's WebExtensions API to the letter and would instead be making its own changes to allow ad blockers to continue working as they always have.

Now, I would argue that Google had its heart in the right place, but that they did perhaps willfully underappreciate the importance of allowing for dynamic extension-based page filtering. Here's what happened at the technology level. The original Web Request API, and that's what it was known, the Web Request API, allowed developers of web extensions to install complete and powerful in-line filters, both in the query and in the reply loops, sort of encircling the browser's engine. A query filter would inspect and perhaps modify any browser queries leaving the browser on the way to remote web servers. And a reply filter would receive remote web server replies before the browser engine saw them. And this would allow the extension to make extensive edits of the received page, among other things blocking subsequent requests for secondary page assets like ads.

Google's V3 reengineered solution was going to discard all of that, and in fact has, in favor of what they called a "Declarative Net Request API." And Google explained that it would prevent extensions from inspecting web requests made on a page while providing much of the same functionality. And again, I'll say that I think Google's heart was in the right place because that pre-V3 filtering, which is what we've been living with up until now, was awesomely powerful.

Two years ago, at the time of the announcement, Simeon Vincent, who's the Developer Advocate for Chrome Extensions, said that 42% of all malicious extensions which Google had detected year to date, so from January 2018 until October 2018, 42% of all malicious extensions were abusing that API for nefarious purposes. He said: "With Web

Request, Chrome sends all the data in a network request to the listening extension, including any sensitive data contained in that request like personal photos or emails." He says: "Because all of the request data is exposed to the extension, it makes it very easy for a malicious developer to abuse that access to a user's credentials, accounts, or personal information."

All of that is true. Which is why I like the idea, if we can somehow arrange to get both; if we could have good blocking while somehow not allowing extensions that could misbehave to see everything coming and going, to and from the web browser. So with Google's Declarative Net Request API, which is what is in the V3 next generation, an extension preregisters rules that the browser reads and then applies to each web page before and after it's loaded. This hugely improves security and privacy since extensions never receive and see all of the page data, which they do under V2. And then the browser makes all of the modifications requested on behalf of the extension only when one or more of those pre-declared rules are met. And in addition to enhanced privacy and security, this allows Chrome's optimized processing paths to handle all of the actual web request filtering, rather than leaving this to an extension's possibly slow JavaScript code. So we get a big performance boost in addition to enhanced privacy and safety.

So the problem is these changes promise to create a number of problems. The first and most obvious was that this would be restricting what extensions were able to do. And I don't see any way around that. You're either going to give extensions unfettered full access to a web page; or you're going to say just tell us what things you're sensitive about, and we'll look for those for you and then take care of it. So, for example, at the time the developers of NoScript and uBlock Origin were not happy because they liked the power that they had. They made it clear that the new API's declarative rule system would not provide the same level of control.

But the most glaring limitation that arose at the time was the total number of rules that the new engine could accommodate. Google planned to allow what I would think would seem like plenty of rules, 30,000. But it was quickly revealed to be far insufficient for ad blockers. They often have to filter web requests for hundreds of thousands of ad-related domains these days. So during the debate which ensued, the stated requirements ranged from 90,000 to 150,000, some people even arguing that, look, let's not have a too-low limit that ad blockers could hit their heads on. So how about half a million? Anyway, Google compromised and did agree to raise their planned 30,000 to 150,000 individual rules. So that's where we are, and that brings us to today.

Manifest V3 changes are now being tested in Chrome's developer channels, and much of the post-announcement grumbling from two years ago has died down, although some ad blocker extensions, the devs have given up on their product's ability to reliably block ads once these changes reach stable versions of Chrome. And I think that may be some grumbling. I'm kind of hoping, frankly, that Firefox, which is still my primary browser, will stay where they are and say, you know, we're going to want to be the powerful web extensions people. They're arguably sort of the power user's browser, while Chrome continues blasting ahead, having become the mainstream browser.

And so the reason this is in the news is, aside from it being useful and important browser-side technology for us to keep abreast of, last Wednesday Microsoft said that the Manifest V3 changes would shortly be rolling out in Edge also. I mean, you know, it's going to be what Chromium, then the Chromium browsers, have. And anyone using Chromium would probably have to struggle to explicitly continue to support those. Or who knows. It might be that Chromium will disable V2, but keep it around.

Anyway, Microsoft said: "In continuation of our commitment to reduce fragmentation of the web for all developers" - meaning let's all keep with a single API - "and to create better web compatibility for our customers, we plan to support the Declarative Net

Request API and other changes proposed as part of Manifest V3." They said: "The decision to embrace Manifest V3 changes is based upon our dedication to enhance privacy, security, and performance for the benefit of our end users as well as to allow developers to extend and provide rich experiences in Microsoft Edge."

And I'll skip some of this because they did explicitly say that: "We recognize the value of content-blocking extensions and appreciate the role they play in honoring users' choice by blocking advertisements and enhancing privacy by blocking cookies, and we want developers to continue to offer these capabilities." They said: "After an extensive review of the concerns raised by content blockers and the community, we believe that a majority of those concerns have been resolved or will be resolved before Web Request API is deprecated." Meaning the way we've always had it until now. And frankly, with care, maybe the way we need to keep it for some users.

So my hope is that a workable compromise has been or can be reached. And frankly, I love the power of a full filtering ad blocker, though it does come with some serious security and privacy tradeoffs. Again, I don't think there's a way around that. You're either going to blind your extensions to the full content of the page and come up with like a second-level filtering rule-based approach, which is what Google has proposed two years ago, or you're going to allow extensions very powerful access and trust them. And given our curmudgeon friend Gorhill and just who he is, I would trust him and his work, which is the way uBlock Origin is created, completely. And I would argue that those of us who listen to this podcast know how to be cautious. But we're also a vanishingly small minority.

So I get it. I get what Google is trying to do. I've seen the extension burden that some Chrome users subject their browser to. I look at it, I think, what are all those things? Because it's like, oh, let's add this, and let's add that. Sooner or later you're going to add something malicious, if your approach is just to add everything that you encounter. So it'll be interesting to see how this falls out. I wouldn't be unhappy if Chrome deprecated V2, but then allowed you to turn it back on if you were a power user, and/or if Mozilla allowed the same capability in Firefox. We'll just sort of have to see how it goes.

Last Tuesday, as I mentioned, Microsoft issued fixes for 87 security vulnerabilities. So yeah, a slow month, but only when measured against everything year to date. Those 87 fixes included a pair of critical remote code execution flaws. Actually, I think there were 11 of them in total. But there were two that stood out. One was in the core Windows TCP/IP stack, and another was in Microsoft Outlook. And I'll come back to the rest in a minute, or come back to those two in a minute. There were also nine other critical flaws. There were 75 ranked as important, and one classified as only being moderately important. And of course they collectively affected Windows, Office, Office Services, Web Apps, Visual Studio, Azure, .NET Framework, Microsoft Dynamics, Open Source, Exchange Server, and the Windows Codecs Library. And the good news is none of those are known to be under attack. However, six of the vulnerabilities were listed as being publicly known at the time of release. So a good thing to apply.

Now, hopefully most of our listeners will have perked up or perhaps started digging a new bunker for themselves at the mention of a core flaw in the Windows TCP/IP stack. Those are never good. And did I mention that this one, once it is weaponized from being merely a Blue Screen of Death crash at the moment into an active remote code execution attack, everyone who has looked at it is saying that it is 100% wormable. So what that means is all of these instances of Windows which have any public presence on the Internet have a TCP/IP stack outwardly facing on the 'Net and are subject right now, prior to last Tuesday, to at minimum a crash. They call it "denial of service" because if your server crashes, your service is denied. But right now it's crashing. That's not difficult to do. Everyone is expecting this will mature over time. Rapid7 and McAfee both have good write-ups about it.

Here's how McAfee summarized the situation. They said: "Today, Microsoft announced a critical vulnerability in the Windows IPv6 stack, which allows an attacker to send maliciously crafted packets to potentially execute arbitrary code on a remote system. The proof of concept shared with MAPP (Microsoft Active Protection Program) members is both extremely simple and absolutely reliable. It results in an immediate Blue Screen of Death; but more so, it indicates the likelihood of exploitation for those who can manage to bypass Windows 10 and Windows Server 2019 mitigations." Meaning that the things that are necessary to mature this from a crash to a remote code execution, everyone is thinking it's probable.

They said: "The effects of an exploit that would grant remote code execution would be widespread and highly impactful, as this type of bug could be made wormable." They said: "For ease of reference, we nicknamed the vulnerability 'Bad Neighbor' because it is located within ICMPv6 Neighbor Discovery Protocol, using the Router Advertisement type." Now, nobody has published yet a proof of concept, but it's regarded as imminent at this point.

Rapid7 said: "If you're in the U.S. and were waiting for an 'October surprise,' look no further than CVE-2020-16898, which is a remote code execution vulnerability in the Windows TCP/IP stack, or what our own Tod Beardsley likes to call 'exploiting poor implementations of core IETF RFCs.'" They said: "The vulnerability arises when the TCP/IP stack does not properly handle ICMPv6 Router Advertisement packets. Successful exploitation requires sending specially crafted ICMPv6 Router Advertisement packets to a remote Windows computer and could give an attacker the ability to execute code on the target server or client." It carries a CVSS v3 base severity score of 9.8 because, yeah, it's that bad. It would be a 10 if it existed now in the wild. If there was public proof of concept code, it would be over. We're talking about any presence of a network, of a Windows network stack on the Internet.

So they said: "Our talented crew of Rapid7 vulnerability researchers have a technical analysis up on AttackerKB, and security firm McAfee has their own technical analysis. Their research and engineering teams note that the Microsoft-provided exploit is both extremely simple and perfectly reliable, and results in an immediate Blue Screen of Death."

They said: "Before we go any further, we would like to strongly encourage you to patch this vulnerability if you are running Windows 10, Windows Server 2019, Windows Server Core 1903, 1909, or 2004. You really don't want to mess around," they're saying, "when the word 'wormable' is being used and so many eyes are on the non-BSOD prize of a fully working RCE. If you cannot patch, consider disabling ICMPv6 Recursive DNS Server (RDNSS) as a workaround," he said, "which is unfortunately only available for Windows 1709 and above." On the other hand, that's all recent Windows.

And then they have a PowerShell command that I have in the show notes. Again, it's hard to imagine that somebody has not updated, but it's only a week old, and it's really important. And, I mean, this is a potentially Internet meltdown-scale vulnerability, if somebody figures out how to turn this into a full working remote code execution exploit. We know that reverse engineering is now being done on a monthly basis, as soon as Microsoft updates something which is really bad, like they tried to do with Zerologon a couple months ago, and that's been a disaster. So here we have something even potentially more serious because remember Zerologon you've got to already be in, and then you can use it to gain access to an enterprise's domain controller. This is any Windows instance, any Windows stack exposed publicly. Again, it's not immediately weaponized. But everybody's working on that right now because this carrot is so big and juicy.

They said - this is Rapid7. "As noted above, there are many folks who have access to the known BSOD exploit, and more who are currently burning through" - this is Rapid7 - "burning through cases of Mountain Dew while working to replicate the BSOD, then to weaponize the unpatched vulnerability. In the short term, and possibly long term," they're saying, "you should be more wary of disruption and distraction campaigns using this weakness, especially since IPv6 is very likely running on your internal network, where Bad Neighbor attacks are really most likely to occur without you being aware of it."

They finish: "You and your organization should really be prepared to have between one to five critical 'patch now' events each month for the foreseeable future. That may seem disruptive, but the spate of critical bugs in core business and remote access technologies has become the new normal, and the only way to handle it is to make it part of the plan."

I wanted to go all the way through that because I thought what Rapid7 said was significant. I mean, and Leo, it obviously caught your attention. Yes, I mean, if we objectively look at how bad and how frequent these serious problems have become, it is...

Leo: Shocking.

Steve: Really, it is, it is shocking.

Leo: Why would anybody use Windows? It's just ridiculous. Just stop. Stop the insanity. Holy cow.

Steve: Yeah, yeah. Okay. So I love that they're trying. I headlined this in the show notes: If at first you don't succeed, Zoom Zoom again. Last Wednesday, Zoom announced that this week their 30-day evaluation of end-to-end encrypted video conferencing would begin. And of course, as we know, they've pretty much blown the implicit assumption of trust we might have been willing to confer on them due to their constantly botched and incredibly miscommunicated response to their early problems and to their plans to sort of, maybe mostly, but we really want to encrypt, we think. Okay, we got it. But we wouldn't be fair if we didn't acknowledge what they announced last week because it does really, on its surface, sound exactly right.

Here's what Zoom said last week: "We're excited to announce that starting next week" - which is now this week - "Zoom's end-to-end encryption offering will be available as a technical preview, which means we're proactively soliciting feedback from users for the first 30 days. Zoom users, free and paid around the world, can host up to 200 participants in an E2EE" - that's the new acronym, end-to-end encrypted - "meeting on Zoom, providing increased privacy and security for your Zoom sessions. We announced in May our plans to build an end-to-end-encrypted meeting option into our platform, on top of Zoom's already strong encryption and advanced security features. We're pleased to roll out Phase 1 of 4 of our E2EE offering, which provides robust protections to help prevent the interception of decryption keys that could be used to monitor meeting content.

"To be clear," they wrote, "Zoom's E2EE uses the same powerful GCM encryption you get now in a Zoom meeting. The only difference is where those encryption keys live." Okay. This all sounds good so far. In typical meetings, Zoom's cloud generates encryption keys and distributes them to meeting participants using Zoom apps as they join. With Zoom's E2EE, the meeting's host generates encryption keys and uses public key cryptography to

distribute these keys to the other meeting participants. Zoom's servers become oblivious relays and never see the encryption keys required to decrypt the meeting contents. Okay. That's, like, perfect in terms of architecture. Unfortunately, then, Zoom's CEO had to stick his foot in an otherwise great announcement.

Leo: Something like, let me guess, unless there's child sexual material on there, in which case we keep the keys.

Steve: Well, he did that before.

Leo: Yeah. If law enforcement needs those, then they can have them.

Steve: Yeah. The announcement finishes. "Zoom's CEO, Eric S. Yuan, said: 'End-to-end encryption is another stride toward making Zoom the most secure communications platform in the world. This phase of our E2EE offering provides the same security as existing end-to-end encrypted messaging platforms.'" What? No, it doesn't. It's like, it was better until you said that.

Leo: You know what, I wouldn't say this is an admission of a flaw or anything like that. I think he just doesn't - he's not a technical guy.

Steve: No, again, just muzzle this guy. He always says the wrong thing.

Leo: Yeah, yeah.

Steve: You know, everything was great until he just downgraded this to being as good as everything else.

Leo: It's as good as Messenger.

Steve: Because everything else is not good.

Leo: Right. It sounds good.

Steve: Most of them are terrible.

Leo: But a lot of them are not; right.

Steve: Yeah. Oh, boy. So anyway...

Leo: If he had said, "Now we're as good as Threema," you'd go, okay. Okay, yeah.

Steve: Yeah. Unfortunately, now we're as good as the other services which manages your keys for you, after we just told you that we're not going to do that.

Leo: Oh, boy.

Steve: Anyway, so the good news is they're taking - if we believe all this, they're taking the key management out of the hands of Zoom's servers and moving it to the meeting host, who then generates keys and distributes individual keys to each participant using public key crypto.

Leo: Good.

Steve: That is a far more secure architecture than any centralized key distribution system which is what you get from Zoom if you don't ask for the fancy encryption.

Leo: Right.

Steve: So anyway, so this will be available as a technical preview, is now, this week. To use it, customers must enable E2EE meetings at the account level and opt into E2EE on a per-meeting basis. I've got a link in the show notes. Then, and I won't go into the details, but the posting continues with a little Q&A, asking and answering. But the questions asked are: How does Zoom provide E2EE encryption? We already know that. How do I turn on? You turn it on. When would I use it? When you want to. Do I have access to all the features of a regular Zoom meeting? And they answer that question. Obviously some things break because of this. We know about that. Do free Zoom users have access to end-to-end encryption? Yes.

Leo: Oh.

Steve: How is this different - oh, yeah, it's free and paid.

Leo: Nice. Very good. Very good.

Steve: Yeah, because that's one of the things, that's one of the mistakes they made is they said oh, no, this is only for paid users. Well, the whole privacy advocate world exploded at that point. And they said, oh, sorry. Eric spoke, and he wasn't supposed to. So anyway, how is this different from Zoom's enhanced GCM encryption? And, you know, it's better. How do I verify that my meeting is using end-to-end encryption? You get a green shield. How will you continue to provide a safe and secure platform? We'll keep trying. What is the rest of the timeline for E2EE? So, and then there's three other phases and whatever.

So anyway, for what it's worth, they're giving the host of meetings the ability to press a button, generate keys on their client, distribute them using public key crypto to the other clients. And that's as good as it can be. We don't know there aren't mistakes. We have to believe that's what they're doing. But that is the architecture. And the architecture is sound. So I think it's a step forward for Zoom. And even if they don't get it right the first

time, they're going the right direction. And we know that they've got a good team of actual crypto people looking at the stuff now, and this is the way it should be done. So props to them, and let's just put Eric on an island, and he can count his money.

Leo: Yeah. So it's not that there's something wrong with it. It's just there's something wrong with him. And in fact it sounds like they're doing it exactly as you would want them to do it. So that's nice.

Steve: Yes, yes.

Leo: Very glad to hear that. That's great.

Steve: Yeah. So Sonic is not having much luck this year.

Leo: Sonic the Hedgehog? Sonic the fast food operation? Sonic the Internet Service Provider? Which Sonic are we talking about?

Steve: That's right, the Internet Service...

Leo: The last one.

Steve: The firewall provider, the people who make the SonicWall.

Leo: Oh, that's different, the SonicWall, yeah, that's another - that's different from SonicNet, yes, okay.

Steve: Correct. Oh, yeah, yeah, yeah. Not the ISP.

Leo: That's our ISP, so we want - my ears perked up.

Steve: Oh, yeah. Those guys are good.

Leo: Yeah.

Steve: This is their so-called NSA, the Network Security Appliance.

Leo: Oh, yeah.

Steve: If any of our listeners - and apparently there are nearly 800,000 of these. I was thinking, wow, this is a pretty successful product.

Leo: Oh, yeah.

Steve: Unfortunately, it's in trouble right now. While we're on the topic of supercritical remote code execution network vulnerabilities with critical ratings above 9 out of 10, we need to make sure that any of our listeners who might be responsible for the operation of one or more of the nearly 800,000 vulnerable SonicWall NSA - that's Network Security Appliance - firewall devices that are currently exposed to the Internet have patched their devices. Just, I mean, like absolutely you have to patch.

The vulnerability was discovered by the Tripwire VERT security team and was given a CVE-2020-5135. It impacts the SonicOS, which is the operating system running on the SonicWall Network Security Appliance devices. As its name implies, the SonicWall NSAs are used as firewalls and as SSL VPN portals to filter control and allow employees to access internal and private networks. Maybe there are a bunch more of these things that have been deployed recently in order to set up VPNs to allow remote workers to connect back to the enterprise network.

In any event, the Tripwire researchers explained that the SonicOS contains a bug in a component that handles custom protocols. The vulnerable component is exposed to the WAN, of course, the public Internet, interface; right? Meaning that any attacker can exploit it remotely. And moreover, Tripwire teased that exploiting the bug is trivial, even for unskilled attackers. Oh, boy. In its simplest form, the bug can cause a denial of service and crash devices. But as we often see, a remote code execution exploit is probably feasible, they said.

They reported the bug to the SonicWall team, who released patches last Monday, so that's a week and a day ago. Tripwire announced the vulnerability two days later, last Wednesday, but blessedly declined to provide any further details. So props to them for not feeling like they had to spill the beans. Remember, of course, that the Zerologon vulnerability disclosure was delayed six weeks; and, of course, that didn't help in this case. And it really doesn't help here. If we know, we know if there are - actually I know the number - 795,357 currently exposed, and at the time of the release vulnerable, SonicWall devices on the public Internet, they're never going to get fixed. Some of them, yeah, people who have on-the-ball IT, hopefully listeners of the podcast who know about SonicWall, got the announcement, already are updated. This is old news for them. Most of these devices will never get fixed. I mean, that's just the way we know it is now.

So Tripwire said: "Tripwire VERT has identified a stack-based buffer overflow in SonicWall Network Security Appliance. The flaw can be triggered by an unauthenticated HTTP request involving a custom protocol handler. The vulnerability exists within the HTTP/HTTPS service used for product management as well as SSL VPN remote access. An unskilled attacker" - now, again, skilled to create the exploit, unskilled to use it. "An unskilled attacker can use this flaw to cause a persistent denial of service condition." In other words, easy to make a crash. "Tripwire VERT has also confirmed the ability to divert execution flow through stack corruption, indicating that a remote execution exploit is likely feasible. This flaw exists pre-authentication and within a component (SSL VPN) which is typically exposed to the public Internet. As of the date of discovery, a Shodan search for the affected HTTP server banner indicated 795,357 hosts." Oh, boy.

So this won't be an instant attack. I'm sure this is why Tripwire are biting their tongue. Probably once the proofs of concept appear, then they'll do a full disclosure, as would be their due because they did discover this and responsibly report it. They waited for SonicWall to produce the patch, gave it a couple days, still they'd not said anything further. But this thing is so juicy, 800,000 servers that are currently exposed, just shy of that, on the Internet.

So as we know, we were just talking about this, it won't be an instant attack. Some reverse engineering and skilled R&D will be needed. But now that we have a ransomware ecosystem containing highly motivated penetration hackers who know that they'll be able to sell their intrusions to high-bidding ransomware attackers, this will be far too juicy to be passed up. I mean, it will be a week or two, and we'll be talking about this. Yes, an exploit has been developed. SonicWalls that were not patched are all being taken down. All of the enterprises behind them are now compromised with ransomware, blah blah blah. We know how this drama plays out. We've seen it several times now.

And we also know that while, yes, some SonicWall devices will be updated before the inevitable attacks commence, it's just not going to be the case that, I mean, I would argue less than half of them are probably being actively maintained. They're deployed. They've been forgotten. They're just sitting there, waiting to get exploited. So such is the world we're in today.

The good news is Microsoft's out-of-cycle patches, which were released last Friday, are kind of yawners. They're just not that big a deal. One exists in the HEVC codec, which is not part of Windows 10 by default. You've got to go get it from the Windows Store in order to have the vulnerability. And then it has to be leveraged. A malicious image needs to be created, you have to display the image, and Windows and the Microsoft Store will be updating this HEVC codec anyway. So that doesn't seem like a big problem.

The other one is interesting. It was technically a problem with Visual Studio Code that, Leo, you and I were just talking about last week. Again, not a huge worry. It allows an attacker to craft a malicious package.json file which, when loaded by Visual Studio Code before the update, could execute malicious code. So, you know, if you downloaded an open source package that contained a malicious package.json file and ran it on an unpatched version of Visual Studio Code, that would be a way of a bad guy to run code on your machine. It turns out that these package.json files are regularly used with JavaScript libraries and projects. As of course we know, JavaScript and the server-side Node.js technology are arguably one of the more popular technologies on the 'Net right now. So anyway, anyone using Visual Studio Code should update to the current version, and you'll be okay.

This is just sort of a fun tip. There's been some talk going around about a so-called God Mode in Windows 10. It's sort of a weird kludge. You create, I mean, bizarre that Windows 10 does this, but you name a folder one of those wacky global unique IDs, you know, a GUID. And when you do that, that sort of enables this God Mode, which is a whole bunch of Windows setting things contained in the folder. Our listeners may have heard that Microsoft is attempting to "simplify," in quotes, and eventually eliminate the traditional Windows Control Panel that I and probably a lot of our listeners appreciate. It's nice to have all that stuff in one place. Microsoft, in their infinite wisdom, has decided, oh, that's confusing. So we're going to just chop it all up and move it around and make it more context aware or something. No. Anyway, that's what they're doing.

So I did some digging, and I found a site and an app, neither of which I knew about. If you google "extended god mode," you will find it. I've also got it in the show notes. A guy by the name of Peter Panisz, P-A-N-I-S-Z, has a site, WinTools.info. I'm very impressed. I don't know it's taken me so long to stumble over WinTools.info. But this guy has written a bunch of nice free/donation ware. For his description of his Extended GodMode, he said: "The original God Mode contains more than 200 items, depending upon your configuration and operating system version." He says: "Extended GodMode" - which is his zero-installation app which you just run - "complements these functions with the Admin Tools and Control Panel elements. It displays all setting options in a single interface and allows access to them grouped in several ways according to different criteria. Extended GodMode also includes a powerful search engine. Individual searches can be saved to create groups of settings.

"Extended GodMode supplements default God Mode with the following features: a quick search by item name; searches can be saved; you can have managed favorites; display recently used items; most used elements; integration of Control Panel and Admin Tools, which can be disabled; quick access to each setting," blah blah blah. And support 64-bit Windows 7, 8, not 9, but 10. He's got a beta for 32-bit versions in the works. Free of charge; no install required.

On my Windows 7 machine this morning it listed 340 individual problem-solving applets. Again, you don't have to install it. You just run the EXE. It's like 437K or something. I mean, this guy - and he's got a bunch of other goodies. So WinTools.info gets my top recommendation. And there's, like, four pages of freeware stuff that he's got. So I'll be very surprised if you don't find something tasty and delicious, and thank me.

And in fact this gadget is so nice that it inspired me to create a new permanent thread in my blog at the new GRC forums, forums.grc.com. I have a "My Favorite Utilities, Apps, and Services." I think I said goodies. Utilities, Goodies, and Services or something like that. I added Sync.com and Syncthing, in addition to this Extended GodMode. And over time I'll be - oh, and I also put an entry for pfSense and my preferred pfSense hardware little box, which I really like. So I'll be extending that thread over time. It's a place for me to sort of keep our listeners apprised of these things that I find and really think are tremendous. And it was funny, I noted that it was one year ago, Leo, on October 1st of 2019, was "The Joy of Sync" podcast.

Leo: I remember that, yeah.

Steve: Where I talked about both Sync.com and Syncthing. I'm still completely happy with Sync.com, and I put a link in there in my thread to the referral link, which will give our listeners a free gig in addition to the first five that are free, that you just get for setting up a free account. And I'm so dependent upon Sync.com. And of course you and I are both fans of Syncthing.

Leo: Love it, yeah.

Steve: Which is a non-cloud peer-to-peer synchronizer, which again, another example of just doing everything right. So, very impressed.

Leo: Yeah.

Steve: One piece of closing-the-loop feedback. When I was in Twitter this morning looking for a Picture of the Week, which as we know I found, I saw a question from Patrick. He said: "Hi, Steve. Question about ShieldsUp! and port 0." He said: "I recently moved, and when scanning with ShieldsUP!, port 0 shows closed instead of stealthed." So instead of that nice green field, he has a blue box in the first cell, blue for closed, and then of course red is worse. That's open. But he says it's showing closed instead of stealthed.

He says: "I'm using the same ISP and router as before, though I do think the modem is different. Just curious," he says, "if there is a way to stealth port 0 because, based on the few postings I can find, this is an ISP issue, and I can't do much about it. Thanks."

So port 0, I don't think we've ever talked about it, or if I ever have, not for many years. Back when I was implementing ShieldsUP!'s full service port scan, I encountered some reference to port 0 being a bit ping-like in that, unlike the other 1023 service ports, numbered 1 through 1023, port 0 was not assigned to any specific service. And at the Unix Network API level - of course all this began, you know, the Internet was born on Unix - a specification of a null port when obtaining a network socket is shorthand for asking the OS to pick an unused local port of its choice to use in an outbound connection. But at the network level, down at the packet bit level, those 16 bits in the packet of a port number can indeed be all zeroes. So even though it's unused, technically it's legal.

So I figured, with ShieldsUp!, what the heck. Let's send out a few TCP SYN packets with all their port bits set to zero, and see what happens. And as Patrick's question notes, sometimes you get back a reply. Which probably means that it is a means for an attacker to possibly unstealth someone - and I don't know how important that is, but it's kind of fun to be stealth - who might have overlooked port 0. Which is of course why ShieldsUP! sends out those packets. So what to do about it? Well, that depends entirely upon where those SYN packets are being bounced back from. Since port 0 is not technically valid, any jump along the way between GRC's ShieldsUP! testing server, or actually it's a client, and the user could be intercepting that SYN aimed at port 0 and think, port 0? What the heck. Let's just say no.

Of course saying no is not stealth. Saying nothing is stealth. But my point is those port 0 SYN packets might not even be making it all the way to the user. Your ISP certainly has no problem blocking port 25, ports 137 through 139, and 445, which of course were famously Windows disasters. So they might also be returned, those packets could be returned with a closed status for port 0. He did mention that he was using the same ISP. But if he moved, he might actually be coming to the ISP through a different route. So it still could be the ISP.

What you can do to stealth something which is not stealthed by default is set up a static port forward to nowhere. That is, most routers allow you to set up a DMZ. And if you can DMZ that port, port 0, send it to a nonexistent IP on your LAN. And what'll happen is, rather than responding with closed on port 0, if your router does, it'll forward that into nowhere, and no response will happen. I'm kind of skeptical that it's the router because it would seem odd to me for the router to say closed on port 0, but not to respond to any other ports. But again, you just have to experiment a bit. But if you are able to forward port 0 to nowhere, then that would be a way of doing so.

Leo: Ryuk, Ryuk, Ryuk. It's time to talk about ransomware.

Steve: However you pronounce it, it is spelled R-Y-U-K.

Leo: Well, it's a Japanese word, so it's not even spelled R-Y-U-K. That's the Anglicization of it.

Steve: Oh, okay. Well, that's how I'm spelling it.

Leo: Okay.

Steve: I do know you don't want it.

Leo: You don't want it. That's true. I'll grant you that.

Steve: You do not want that in your network.

Leo: No.

Steve: The DFIR Report site specializes in forensic analysis of ransomware, and also some other malware attacks, but predominantly ransomware. Two days ago, on Sunday, the DFIR Report site posted a fascinating step-by-step forensic walkthrough of - technically it was a five-hour Ryuk attack. So that would be five hours, 300 minutes, from the time that a phishing link was first clicked on to provide a callback to the malware operators who were initially, before that, completely unaware of the resources they had just been provided access to. Now, as we'll see, I think they waited 2.5 hours. My theory is, based on the time of day that that link was clicked, they waited 2.5 hours for all the employees to drain out of the organization and go home so that they could operate more unseen, and they weren't in a big hurry.

But anyway, I'll explain how all this thing unfolded along the way. In order for this report's blow-by-blow walkthrough to make sense, we first need to add details of three commonly used malware packages to our knowledge base. The first malware package is known as BazarLoader, B-A-Z-A-R Loader. It's the tip of the spear, the first thing that gets loaded and runs to provide the foothold that an attacker needs for then further penetration. CyberReason.com has been watching BazarLoader since its first sighting this past April. There's a quick summary of what they know about it, and I'll note that this research is about BazarLoader in general, not specifically about its use in this particular malware attack.

But CyberReason said: "Bazar can be used to deploy additional malware, ransomware, and ultimately steal sensitive data from organizations." Basically it's an easy entry backdoor. They said: "Bazar malware infections are specifically targeted at professional services, healthcare, manufacturing, IT, logistics, and travel companies across the U.S. and Europe." And of course that's sort of specific to their particular view of it. Now it's being used as a general purpose spear tip. They said: "Bazar leverages the Twilio SendGrid email platform and" - and this is important - "validly signed loader files to evade traditional security software in conjunction with a fileless backdoor to establish persistence.

"Over the course of their investigation," they said, "it is evident that Bazar is under active development. Recently, the active campaigns disappeared, to later reappear with a new version, suggesting that the group is under a development cycle." They said: "This stealthy loader evades detection by abusing the trust of certificate authorities, much like previous TrickBot loaders. This loader uses EmerDNS with the .bazar top-level domain," thus the name, .bazar, "for command and control, and is heavily obfuscated."

Now, okay. As an aside, what is EmerDNS? Get this. It's a completely decentralized blockchain-based DNS alternative that we've never talked about. What's blockchain-based DNS? Well, quoting from the EmerDNS site, they said: "Are you afraid your website could be suspended by authorities? With the screws tightening around the world, your fears might well be justified. EmerDNS is safe from any kind of censorship. No other user can modify your record. Only the record creator can manipulate its content." So, uh-huh. What could be better for malware command-and-control servers than to rely upon a decentralized blockchain-based DNS that cannot be sinkholed by authorities? Yeah. Unfortunately, once again, sort of something, I guess I would put that in the gray, bordering on darker than gray, side of the law.

Anyway, Bazar also uses anti-analysis techniques to thwart automated and manual analysis, and loads the encrypted backdoor solely in RAM. So that's one piece of what we'll be talking about is this BazarLoader. It's now part of the toolkit of whoever was using Ryuk in this particular attack. It is signed validly, runs in RAM. It avoids and evades AV and opens a connection using this Bazar blockchain-based DNS, gets the IP of where the command-and-control server is currently located, and phones home.

Okay. Next is Cobalt Strike. What's interesting is that Cobalt Strike is not hiding. It lives at CobaltStrike.com, where it is being offered for sale for supposedly legitimate Red Team penetration testing purposes. Uh-huh. The Cobalt Strike site says: "Cobalt Strike is software for Adversary Simulations and Red Team Operations. Adversary Simulations and Red Team Operations are security assessments that replicate the tactics and techniques of an advanced adversary in a network." And of course, if this thing is given to an advanced adversary, then it's not replicating them. It's actually perpetuating them.

Anyway, they said: "While penetration tests focus on unpatched vulnerabilities and misconfigurations, these assessments benefit security operations and incident response. Cobalt Strike gives you a post-exploitation agent and covert channels to emulate a quiet long-term embedded actor in your customer's network."

Leo: Yeah. Yeah, baby.

Steve: Okay, now, you know, if this was a legitimate Red Team, they really wouldn't need that; right?

Leo: No.

Steve: So it's like, okay. They said: "Malleable C2 (Command and Control) lets you change your network indicators to look like different malware each time." How convenient. "These tools complement Cobalt Strike's solid social engineering process, its robust collaboration capability, and unique reports designed to aid blue team training. To learn more about Cobalt Strike, watch the 'Red Team Operations with Cobalt Strike' course." Oh, and get this. "And how much does Cobalt Strike cost? New Cobalt Strike licenses cost $3,500 per user for a one-year license. License renewals cost $2,500 per user per year."

Leo: Just think of all the money you're going to make.

Steve: That's right. Now it's worth noting that despite the hefty price tag, Cobalt Strike has a trial version that's entirely useful and functional and that was in fact used in unlicensed trial mode for this successful Ryuk attack.

Leo: Oh. Why pay for something when you can get it for free? There's no honor among thieves today. That's terrible.

Steve: No, none.

Leo: Terrible.

Steve: Now, Malpedia describes Cobalt Strike a little more practically. Yes, Leo, there is something called Malpedia.

Leo: There is a Malpedia, okay.

Steve: There is a Malpedia.

Leo: Oh, yeah.

Steve: They said: "Cobalt Strike is a paid penetration testing product that allows an attacker to deploy an agent named Beacon" - and that's important, we'll be talking about that, so that's why I wanted to bring this up - "Beacon on the victim machine. Beacon includes a wealth of functionality to the attacker, including but not limited to command execution, key logging, file transfer, SOCKS proxying, privilege escalation, Mimikatz functionality, port scanning, and lateral movement. Beacon is in memory, fileless, in that it consists of stageless or multistage shellcode that, once loaded by exploiting a vulnerability or executing a shellcode loader, will reflectively load itself into memory of a process without touching the disk. It supports command-and-control and staging over HTTP, HTTPS, DNS, SMB named pipes, as well as forward and reverse TCP. Beacons can be daisy-chained. Cobalt Strike comes with a toolkit for developing shellcode loaders, called Artifact Kit. The Beacon implant has become popular amongst targeted attackers" - uh-huh - "and criminal users as it is well written, stable, and highly customizable."

Okay. The third and final tool we need to note, for reasons that will soon become clear, is a 100% legitimate bit of freeware known as AdFind, A-D-F-I-N-D, as in Active Directory Finder. AdFind is offered by a freeware developer located at www.joeware.net, J-O-E-W-A-R-E. And everybody, I must say that I was somewhat delighted to encounter a website that's still on the 'Net and being actively maintained which makes GRC.com look futuristic by comparison.

Leo: Oh, I've got to see this.

Steve: Oh, Leo. It is really something, www.joeware.net.

Leo: Guess Joe isn't all that up on the latest web technologies.

Steve: Actually, this has to be on purpose. Because this thing, AdFind, was written in C++ and compiled with - I know.

Leo: No, this is ironically tacky. Okay. Never stop exploring.

Steve: That's right.

Leo: Wow.

Steve: And then we come to an advanced page where we have something - I don't know how he avoided the spinning mailbox. That's probably somewhere on the site. I know.

Leo: This is very hackery, yeah. So do you think this is the person who created the tools? Or this is just somebody...

Steve: Oh, no, no. This is definitely...

Leo: This is Joe.

Steve: Joe is a legitimate - he has a very Active Directory-focused approach. And Active Directory, I mean, it just - it's a nice tool. It's not malicious at all. AdFind accepts a bunch of command line commands. For example, adfind.exe -f and then in quotes "objectcategory=person." And that dumps all the people objects that an Active Directory server knows of. Or you can say adfind.exe -f and then in quotes "objectcategory=computer," and it will find all the computers in the domain and tell you about them. Or "trustdump," and it will dump all of the trust objects in Active Directory. Or "subnet," and it will dump all of the subnets that that Active Directory server knows about. So it is a very handy tool for someone who needs something small and lightweight that they want to use to, starting from zero, to find out where they are after a penetration.

Okay. So we know about BazarLoader. We know about Cobalt Strike. And we know about its Beacon and this freeware AdList utility. So the attack began with Bazar being introduced to the victim's environment through a phishing email with a link to what looked like a PDF located in Google Docs. Of course it was trading on the targets, so whoever it was who got the phishing email, their knowledge and presumed trust of Google.com stuff. So the link was to a file containing a double extension EmploymentRecord.pdf.exe. And of course well-known extensions are often suppressed. So what the user saw was EmploymentRecord.pdf. And they didn't stop to wonder why that filename extension was not suppressed. And of course it also confirmed that, oh, look, it's a PDF. Those are safe to open.

Well, of course it wasn't. It was a .pdf.exe, which was actually an instance of Bazar. Signed, properly signed, trusted, obfuscated, so basically self-encrypted per instance so that AV wouldn't know what it was. It's got a signature. Okay, let's run it. The user wants to run it. Thus the BazarLoader achieved its first penetration and foothold on that user's machine. It immediately spawned a new instance of Windows Explorer process, into which it injected itself into that process's memory for continued stealthful execution, and then terminated its spoofed PDF executable. It just abandoned it. Now it's living in an instance of explorer.exe, known and trusted and perfect except, by doing a RAM injection, it's not part of that executable. It's just living in that executable's process space.

It then reached out to its designated command-and-control server at 3.137.182.114, establishing an encrypted TLS tunnel to that IP's port 443. So it just looks like any standard, okay, outbound HTTP/TLS connection, except it's not. It's this instance running in RAM of explorer.exe that has now made contact to the attackers.

Now, since you never know when a phishing attack is going to be successful, from the attacker's viewpoint, they're not always paying attention. In this instance, 2.5 hours passed before they first responded to that BazarLoader's instance incoming call. Whereupon its human controllers, wherever they were located in the world, took over.

However, since that initial penetration occurred, that is, the user clicking on the link in email, occurred at 5:01 p.m. local time, it's also very possible that the attackers waited, somewhat impatiently, for 2.5 hours to pass so that that company, if it was closing down at 5:00, all the employees would have left for the day, turned the lights off, locked the doors, giving them a greater opportunity to do what they want to do unnoticed.

Leo: Was it a 5:01 on a Friday? Because that's often the case. They do it on the weekend; right?

Steve: Ah, I didn't catch the day of the week.

Leo: Because they want to get the most amount of time to play before they come back to work.

Steve: Well, in this instance they do know that the employee clicked the link on their workstation at 5:01. So it would...

Leo: That's interesting. He was on his way out.

Steve: It was probably not a Saturday or - yeah.

Leo: Click here before you go home for the weekend.

Steve: Yeah, exactly. So in this instance the machine that that user was using was a domain user with no additional privileges. But BazarLoader used the built-in Windows utility nltest. And I looked. I've never had the occasion to type it, but I opened a command prompt and, sure enough, my Windows 7 machine has nltest, "nl" as in Netlogon. And among other things, this is a Windows utility, ships with Windows. Nltest will obtain a list of the network's domain controllers to provide an initial mapping of the domain.

So by using something that that workstation had, although there were no privileges, you don't need any in order to just make that query using the nltest command line executable. And that allowed the attackers to get the location, names and location of the domain controllers. Now we're in a world where we have the Zerologon vulnerability, which is now a reality and which is, as I said last week, because it is a protocol-level reality, it is going to be a long time before all those domain controllers get themselves updated.

In this instance, the attackers wasted no time. They were on a machine with no privilege, so they immediately reached out and used Zerologon exploit to reset the password of that network's primary domain controller, which nltest had identified for them. They then began moving laterally from machine to machine using initially SMB file transfers to duplicate themselves, and WMI to remotely execute the newly copied instance on the other machine. Once they got onto that primary domain controller, they executed this Cobalt Strike Beacon. The Cobalt Strike Beacon comes in both a portable executable and a DLL version. And what these guys were using was they would copy the DLL and then execute it using the rundll32 command, which is also part of all Windows instances forever.

So once on the primary domain controller, the attacker again moved laterally onto the secondary domain controller to establish a backup beachhead. And it is believed that maybe something didn't work the way they were expecting on the primary domain controller because they end up going back to it, as we'll see in a second, through a different channel. But in any event, again, they move laterally to the secondary domain controller to establish a backup. That instance of the BazarLoader, that's what I was trying to remember, that instance of BazarLoader reached out and connected to a command-and-control server located at 88.119.171.94, also over TLS port 443, looking like anybody's HTTPS query. And a second Beacon DLL was dropped and executed on that machine using rundll32. It, once it came out, reached out to 5.2.64.174, also to port 443.

So at this point the attacker performed additional domain discovery using just some standard net commands and the PowerShell Active Directory module. They then employed the default named pipe privilege escalation module on the server and used RDP to connect back from the secondary domain controller to the first domain controller using the built-in admin account. And again, as I said, the guys who were watching this believe that that was done because something may have gone wrong with their initial intrusion onto the primary domain controller.

So after moving laterally to the secondary domain controller, they then returned to the primary over RDP, which was guaranteed to work robustly because of course Microsoft provided it to everybody in order to allow these things to be done. So once on the main domain controller, another Cobalt Strike Beacon DLL was dropped there, and it was executed. It also reached out and connected to the same IP as the first Beacon, that's 5.2.64.174 over TLS 443, connecting to the same command-and-control server.

Okay. So now we have BazarLoader and Beacons loaded and running all in RAM on both the primary and the secondary domain controller. So now Joe's well-meaning AdFind (Active Directory Find) utility was used to perform deeper domain reconnaissance to list all the people and all their machines, all the subnets and all the trust relationships. So that took 90 minutes from the time they became active. After waiting 2.5 hours, another hour and a half was used to get all of that set up. Moving throughout the network, they were ready to launch their multi-tiered attack. They had an inventory of all of the machines that they were going to blast with the Ryuk ransomware.

They first RDP'd into the network's backup server, which they had identified thanks to AdFind, which had been located during their earlier reconnaissance. Of course, a commonsense tactic is to always first encrypt an organization's backup servers to prevent easy recovery of all of the other machines. So it's backup servers targeted first for Ryuk execution, followed by the organization's non-backup server servers, and then all of the workstations. After 2.5 hours of active work, and Ryuk now deployed and running on every machine except the primary domain controller, which was at that point hosting the attacker, the attackers concluded their work by also executing Ryuk on the primary domain controller, thus encrypting every machine within the organization.

After Ryuk finishes, it displays the ransom notice and wipes itself from RAM to remove all remnants and record of the attack. So from the time some user unwittingly clicked a link in phishing email, nothing happened for 2.5 hours. Then, 2.5 hours later, every machine in the network was encrypted and a ransom note left on the machines. And that's the way one of these things happens.

Leo: Oh, you're not going to tell us how they remediated it or anything? You're just going to let it hang there?

Steve: That's the end of the story, children. Another company hosed, completely hosed, by Ryuk.

Leo: But, but, but what happened?

Steve: Went right past AV. It was trusted because it was signed. It made connections back to command-and-control servers that are using a bizarre blockchain DNS that cannot be taken down by the authorities. That's the world we're in now. And now 800,000 new SonicWall VPN firewalls are now open to an exploit, and you have to know that bad guys are working feverishly to leverage that new patched vulnerability on all the firewalls that won't get patched as a way in to do exactly what we've just described.

Leo: Wow. Wowie, wowie, wowie.

Steve: It is a nightmare.

Leo: Well, yeah. It's a cautionary tale.

Steve: And on that happy note...

Leo: Yeah. But it's fascinating to hear how it happens. Not surprising; but, yeah, fascinating. That's Steve Gibson, man. This guy, he's our security guru, our hero. Every week we get together and talk about the latest security news. Lots of information. You'll find Steve at his website, GRC.com. That's where SpinRite lives, the world's finest hard drive recovery and maintenance utility. That's his bread and butter. But there's lots of free stuff, like the ShieldsUP! we were talking about earlier. Find out if your port 0 is stealthed. Can you stealth port 0?

Steve: Yeah.

Leo: How would you stealth it?

Steve: You just let it go. You just let anything coming in go, just like nowhere. No reply.

Leo: Right. And most routers you can say, oh, just stealth it. I guess. I'm all green.

Steve: Yes, exactly. You're all green, so your router's doing the right thing, baby.

Leo: It's a Ubiquiti, baby. Of course it is. Steve has lots of information about all sorts of great interesting subjects, including things like Vitamin D, at GRC.com. He also has this show. He has a 16Kb version for people who like to pretend they're living in the 18th Century. He has text-only versions. Actually, that's useful because you can read along as you listen, and it really helps with the understanding, I find. He also has 64Kb audio, all at GRC.com.

I have at TWiT.tv/sn, 64Kb audio and video, as well, so you can download it there. If you want to watch us do the show live, it's usually around 1:30 to 2:00 p.m. Pacific. That's 4:30 to 5:00 p.m. Eastern time, 20:30 UTC. The stream is at TWiT.tv/live. There's audio and video streams. You can pick a stream and listen. If you're doing that, chat with us at irc.twit.tv, irc.twit.tv. We have an asynchronous forum, just like Steve does. Steve has great forums. We have them, too, at www.twit.community. You're more than welcome there.

And of course you can always get on-demand versions of the show at TWiT.tv/sn, as I mentioned. On YouTube there's a whole Security Now! YouTube channel. Best thing to do, though, find yourself a podcast application. Subscribe. I'm sure you'll find it's everywhere. And once you do, you get every episode the minute it comes out. Thank you, Mr. G. Have a wonderful week. I'll see you next time on Security Now!.

Steve: Right-o.