Microsoft's Email Extortion

Description: In this week's grab bag question collection we wonder: What happened, and who cleaned up during last week's elite 2023 Pwn2Own competition? What happens when GitHub inadvertently exposes their own private SSH RSA key? Are all DDoS-for-hire sites legitimate, and is legitimate ever a word we can apply? Just how bad has the malicious open source registry package problem become? And how is it that Russia's presidential staff are still using iPhones? After its rocky start in the limelight, how has Zoom's security been faring these past few years? And what benefits can be derived from the sum of two sine waves along a logarithmic curve? What new feature is Microsoft exploring for their already feature-encumbered web browser? And in one of my blessedly rare rants we're then going to learn what new "revenue harvesting" measure Microsoft has just announced which seems deeply ethically wrong to me.

High quality (64 kbps) mp3 audio file URL: http://media.GRC.com/sn/SN-916.mp3
Quarter size (16 kbps) mp3 audio file URL: http://media.GRC.com/sn/sn-916-lq.mp3

SHOW TEASE: It's time for Security Now!. Steve Gibson is here. There's lots to talk about. An amazing amount of bug bounties paid by Zoom over the years. Steve gives them high praise. We'll talk about Pwn2Own. They just had it at CanSecWest in Vancouver. And a big winner this year, amazing winner this year. And we'll talk about 144,000 malicious packages published in open source software registries. That, and a stinging rant against Microsoft, all coming up next on Security Now!.

Leo Laporte: This is Security Now! with Steve Gibson, Episode 916, recorded Tuesday, March 28th, 2023: Microsoft's Email Extortion.

IT's time for Security Now!, the show where we cover the latest in security, privacy, computers, science fiction, vitamins. Whatever it is Steve wants to talk about, I'm game. Steve Gibson is here, our host. Hello, Steve.

Steve Gibson: Yo, Leo. This is our sendoff for you.

Leo: Yes.

Steve: We're going to be missing you for, what, I guess three episodes?

Leo: Yeah, I think Ant's going to take over next week. And Jason is in Costa Rica right now. But when he gets back, he'll come back in. It's a mish-mosh of people hosting the show.

Steve: It's a potpourri.

Leo: But it's really Steve's show. You're all that matters on this show.

Steve: Well, in this week's grab bag collection, we wonder what happened, and who cleaned up during last week's elite 2023 Pwn2Own competition? What happens when GitHub inadvertently exposes their own private SSH RSA key? Are all DDoS-for-hire sites legitimate? And is "legitimate" actually a word we can even apply to DDoS-for-hire sites? Just how bad has the malicious open source registry package problem become? And how is it that Russia's presidential staff are still using iPhones?

After its rocky start in the limelight, how has Zoom's security been faring these past few years? And what benefits can be derived from the sum of two sine waves along a logarithmic curve? What new feature is Microsoft exploring for their already rather feature-encumbered, shall we say, web browser? And in one of my blessedly rare rants we're going to learn what new "revenue harvesting" measure Microsoft has just announced, which to me seems deeply ethically wrong. Today's podcast is titled "Microsoft's Email Extortion."

Leo: Oh, boy.

Steve: Yeah. And of course...

Leo: Well, I look forward to the answers to all those questions and yet another ridiculous Picture of the Week. It's time for the Picture of the Week.

Steve: For the weekly photo conundrum. So for those who are not looking, imagine that you have a tall wall, very tall. And for some reason you have a need to get to the top of it occasionally. So what do you do? Well, you run a staircase up the side of the wall.

Leo: Yeah.

Steve: Because, you know, then you could climb the stairs to get to the top of the wall. But now you have a problem because you decide you don't want everybody to be able to climb to the top of the wall. You only want some people to do it. So what do you do? You put a gate with a lock so that only people who have a key to the lock are able to open the gate and then climb to the top of the wall.

But here's the problem. In this instance, what we see is this gate that extends across the stairs is on the third stair up. So like you go up three steps, oh, and there's a gate. Well, okay. Now, there is nothing to keep you from swinging around the outside of the gate because the gate's open on one end; right? On one side of the gate it's against the wall, as are the steps going up the side of the wall. But you can just, like, step around the

gate because it's open to the air on the other side. And there's a conveniently placed handrail on the other side for those who do manage to swing around the gate.

Leo: We wouldn't want you to fall.

Steve: Basically ignoring it. Anyway, the picture's wonderful because you just think - in fact, I gave it the caption, "Well, that'll stop 'em," because, you know, it won't. And again, this is one of those situations where a lot of time and industry and thought went into this. Now, the good news is, on this gate, unlike some of the other gates we've seen, the bars of the gate are vertical rather than horizontal. We have encountered in the past gates where they made horizontal bars, creating a ladder out of the gate. So yes, you could, if you didn't want to swing around the side, and the bars were horizontal, you could just use it like a ladder in order to climb over. But in fact in that case you could just probably go up to the top of the wall from the top of the gate. But anyway. Once again, thank you to our Twitter followers who are now understanding what sort of picture to send me to make history because we're getting a lot of great pictures from our listeners.

Leo: It's just amazing how many poorly designed facilities there are out there. I'm expecting as I wander around ancient Europe, Rome, Barcelona, I might find a few of these. And I will send them right along.

Steve: Please have your camera ready, Leo, because, you know...

Leo: I shall.

Steve: ...they are a constant source of entertainment.

Leo: I'll get Lisa to take a picture of me ascending one of these.

Steve: It does make one wonder about humanity a little bit. It's like, well, maybe we should be taken over by the AIs that might actually be intelligent.

Okay. So speaking of making history, Synacktiv's name first came up, well, actually we've talked about them way in the past. But they also came up last week when the firm GRIMM was asked to double-check and verify the result of Synacktiv's forensic reverse engineering of DJI's drone-controlling software. Since Synacktiv's report could have been seen as highly inflammatory, depending upon how much one believes that we actually have any true security to start with, it appeared that GRIMM was brought in to obtain the classic "second opinion." And as we know from last week's topic, they concurred with everything that Synacktiv found.

Okay. So if we didn't already know that Synacktiv clearly knows their stuff, their breathtaking performance during last week's three-day annual Pwn2Own hacking contest, held in Vancouver, Canada, would stand as testimony to that.

Through the years of this podcast we've been tracking these Pwn2Own competitions, both because they're a lot of fun and because it never hurts to have an occasional reality check to remind us that whenever skilled hackers take aim at some technology pretty

much, it seems, any technology which everyone believes to be secure, that belief is quickly proven to be merely wishful thinking.

There are three Pwn2Own hacking contests that take place throughout the year. One is dedicated to smartphones and IoT devices, and we recently covered some of the results from that one. Another is dedicated to industrial equipment, which we also recently talked about. And the third is last week's CanSecWest event that's dedicated to, probably more interesting to most of us, desktops, servers, and smart cars. This competition is widely regarded now as the premiere, most prestigious of all the hacking contests. And this was not Synacktiv's first Pwn2Own win, since they took the title two years ago after the competition in 2021.

Okay. So during the multi-category three-day event, the Synacktiv team successfully demonstrated, first, a heap overflow vulnerability and an out-of-bounds write error in a Bluetooth chipset. And this is what's interesting is that, you know, we were talking a couple weeks ago about the baseband modem, the bugs that Google found, four of them being the worst possible, where no user action - this was in a Samsung baseband cellular modem chip, and how that little chip was able to result in a complete compromise of the phone. Well, here we have a Bluetooth chip that allows you to take over a Tesla.

So this is an out-of-bounds write error in a Bluetooth chipset which allowed them to break into Tesla's infotainment system. And from there they were able to gain root access to the rest of the car. That bit of wizardry netted them a cool, get this, quarter million dollars and Pwn2Own's first-ever Tier 2 award, which is a designation the contest organizers reserve for particularly impactful vulnerabilities and exploits.

They also demonstrated an attack, secondly, known as "TOCTOU." That's an abbreviation in the industry which stands for time-of-check/time-of-use. The less fancy term for this, like the original term, is a "race condition." That's what we always used to call it. Now it's a TOCTOU. Okay. So a race condition where there is a time-critical sequence of events that can be used in some manner to slip past a system's defenses, like you query for some status, then you do something that the designers didn't anticipate rather than waiting for the answer, for example. In this instance they pulled off an attack on Tesla's Gateway energy management system. They showed how they could then, among other things, open the front trunk or side door of a Tesla Model 3 while the car was in motion. That less-than-two-minute attack earned the researchers a new Tesla Model 3. Talk about Pwn2Own.

Leo: But they won't want to drive it because it's dangerous.

Steve: Yeah. Be sure you have your seatbelt on and tie down the front hood. And they also got a cash reward of $100,000. Next, they pulled off a three-bug chain against Oracle's VirtualBox with a host elevation of privilege to earn themselves $80,000, compounding that to their rapidly growing winnings. They used another TOCTOU bug to escalate their privileges on macOS, earning $40,000. By leveraging an incorrect pointer scaling, they were able to elevate their privileges on Ubuntu's Desktop Linux to win $30,000. And, finally, they leveraged a use-after-free flaw against Windows 11 for another $30,000.

Leo: Wow, they made out like bandits.

Steve: They really cleaned up. Overall they took home more than half a million dollars, $530,000, and a shiny new Tesla Model 3 which earned them the largest award ever

raked in by any one contestant in Pwn2Own's history. And it's a good competition. Star Labs, which was the runner-up, took home a $195,000, which was not bad either. So that's a contest to pay attention to. And what's interesting, of course, is remember that China pulled their hackers out of Pwn2Own. So maybe things would be different if China's hackers, which have proved themselves to be extremely skillful many years in a row, if they'd been there. But that's their, you know, China's keeping them home now.

Okay. Here the moral of the story for GitHub is "Mistakes happen." We were just talking about the benefit of having GitHub repositories continuously scanned for any inadvertent leakage of secret data. You know, it's like keys, which should never be published. So it was interesting that this just happened to GitHub themselves, causing them to rotate their primary SSH RSA key. Here's what GitHub explained.

They said: "At approximately 05:00 UTC on March 24th, out of an abundance of caution, we replaced our RSA SSH host key used to secure Git operations for GitHub.com. We did this to protect our users from any chance of an adversary impersonating GitHub or eavesdropping on their Git operations over SSH. This key does not grant access to GitHub's infrastructure or customer data. This change only impacts Git operations over SSH using RSA. Web traffic to GitHub.com and HTTPS Git operations are not affected." So to clarify, they said: "Only GitHub.com's RSA SSH key was replaced. No change is required for Elliptic Curve DSA or Ed25519 users."

So then they explained a little bit more. They said: "This week we discovered that GitHub.com's RSA SSH private key was briefly exposed in a public GitHub repository." They don't ever tell us, like, how that happened, but certainly they dug into it and figured it out. They said: "We immediately acted to contain the exposure and began investigating to understand the root cause and impact. We have now completed the key replacement, and users will see the change propagate over the next 30 minutes. Some users may have noticed that the new key was briefly present beginning around 02:30 UTC during preparations for this change." They probably, like, quickly made sure that it would work, and then they pulled it back and then got ready to do the whole stage.

So they said: "Please note that this issue was not the result of a compromise of any GitHub systems or customer information. Instead, the exposure was the result of what we believe to be an inadvertent publishing of private information. We have no reason to believe that the exposed key was abused, and took this action out of an abundance of caution." So anyway, just sort of a, you know, a reminder that even somebody that's taking every precaution, who is being safe, mistakes happen. And so all you can do is say "Whoops" and then look at what the consequences of those are and fix them. Which they promptly did.

So I titled this "DDoS for Hire, or Not." And I love this idea. It just makes so much sense to me. The UK's National Crime Agency says its agents have created several fake DDoS-for-hire services that are up and running today. You know, so these are - they look legitimate, right, from the outside. So they're on the Dark Web. You can get to them through onion routing with some funky URL that gets passed around among hackers.

Now, okay. So such a site - so for the National Crime Agency to create a fake DDoS-for-hire service, the point is they're trying to catch people, right, who are wanting to hire these DDoS-for-hire services. But they actually have a dual purpose. They can catch those in the act of attempting to hire DDoSers, as well as frightening away others when the nature of the sting operation is revealed. And that's what caught my attention and I thought was so clever.

To serve that second agenda, the National Crime Agency chose last week to reveal one of its previously popular fake DDoS sites by replacing the site's previous homepage with a

splash screen announcing the chilling truth. And I've got the screen in the show notes. It says: "This site was created and controlled by the National Crime Agency."

Leo: This looks so fake. This looks completely fake.

Steve: Yeah.

Leo: They really could have done a better job making this realistic. Doesn't it? I mean, it looks, I mean, "Operation PowerOFF."

Steve: Yeah.

Leo: Come on, man. This is like from the '80s.

Steve: They have the Europol seal, and a seal for the NCA. But so what they're wanting to do is, this says: "The National Crime Agency collaborated under Operation PowerOFF to target users of criminal DDoS services. DDoS attacks are illegal in the majority of countries. The National Crime Agency has collected substantial data from those who have accessed our domain. We will share this data with international law enforcement for action. Individuals in the U.K. who engaged with this site will be contacted by law enforcement. The National Crime Agency has been and will run more services like this site. Operation PowerOFF has already resulted in the arrest of numerous individuals and continues to ensure that users are being held accountable for their criminal activity."

So the point of the second phase is, you know, imagine you're a miscreant who wants to, you know, who's annoyed with somebody else and wants to DDoS them. So you go to this site that, you know, you may know of. Maybe you've used it in the past. And now you're greeted with the news that this was a sting. The whole thing was a sting. So what that does is it chills the entire enterprise of DDoS for Hire because certainly the word will spread through the underground that law enforcement is erecting fake sting DDoS-for-hire sites. And how would you know?

So anyway, I just think that's a clever repurposing of the concept. Run it for a while, collect lots of names, and then flip the thing around to show people that, well, maybe you shouldn't be using DDoS for hire because we're running many of these, and you don't know which other ones you may be inclined to hire are actually ours. And then we're going to get you.

Okay. So I saw a statistic that was somewhat sobering. We've been looking at the new challenges facing online open source repositories which are increasingly being poisoned by a flood of malicious package uploads. These fall under the umbrella of supply-chain attacks. The developer security firm Snyk, S-N-Y-K, says that it recorded more than 6,800 malicious libraries uploaded on the npm and PyPI portals since just the start of this year, so not yet three months, 6,800 individual specific malicious libraries.

The number that caught my eye was that Snyk said that this recent batch brings the grand total of specifically identified malicious packages to more than 144,000 published to open source software registries, published and identified, found and removed, over the past several years since unfortunately this growing problem was identified. I mean, it's good that it was identified. It's unfortunate that it's a growing problem. And wow, you know, it's not clear how we're going to solve this problem.

Okay. So, the Russian news publication Kommersant reports that the Kremlin's security team has instructed Russia's entire presidential staff to discontinue all use of iPhones by April 1st, April Fools' Day. Kommersant reports that employees were told to get an Android device, either from a Chinese vendor, not surprisingly, since China's now Russia's friend, increasingly so, or one running Rostelecom's Aurora OS. And I loved this. The Kremlin officials cited security considerations as being behind their decision, claiming that iPhones were "more susceptible to hacking and espionage by Western experts compared to other smartphones."

Huh? Okay, well, that doesn't correspond to anything we know. But it is certainly another of the recent examples that we've looked at here. In an environment of increasing mistrust and hostility, it really doesn't make any sense for anyone to be using a closed device sourced from an entity on the other side of the dispute. And Apple's iPhones are certainly far more closed than Android devices. So, yeah, if Russia is planning a long-term split from the West, then discontinuing all use of Western-sourced tech is the only sane long-term strategy.

In our industry's apparently eternal quest to rid ourselves of mistakes made in the creation of software, one of the more effective strategies that's been found is the idea of paying good guys to find and report those flaws before bad guys can find them and use them against us. Thus bug bounty programs have become a mainstay. The COVID-driven work-from-home boom quickly put a lesser-known video conferencing system, Zoom, on the map. But as we know, not everything went well from the start. As we've seen time and time again, many more bugs exist than are known.

So Zoom's pre-celebrity confidence in its own software was quickly shaken when bad guys began looking more closely at it than ever before and discovered all sorts of ways that its benefits could be subverted. As we covered at the time, this came as quite a shock to Zoom's management, and they did stumble a little bit out of the gate. But to their credit they quickly hired some experienced right-thinking true security experts, and the establishment of a functioning bug bounty program was near the top of their list.

Okay. So we're now several years downstream. How has that been going? Here's how Roy Davis, Zoom's Security Manager, described this effort in a blog posting last week. Roy wrote: "In security, it's all about who gets there first. We race to identify bugs and issues before the bad guys do, so we tap the ethical hacking community to help us get ahead. We source this help through our Zoom Bug Bounty program, which lets us connect and engage security researchers that help us proactively mitigate risk and create a safer environment for our customers. And we've accomplished a lot as a community in the past year. Here's a look," he says.

"We test our infrastructure every day at Zoom, but we know we're not immune to edge-case vulnerabilities. So we call in backup. The ethical hacker community can sometimes detect bugs that may only be discovered in certain circumstances. That's why our bug bounty program focuses on recruiting skilled, effective researchers. In 2022, we sent additional invitations to researchers to join our HackerOne program with a focus on attracting active security talent. We also like to go beyond our program to find talent, so we tapped into the community via industry events like H1-702," which I'll talk about in a second, which was a HackerOne event, H1 standing for HackerOne.

He wrote: "These researchers work hard to help us, so we strive to celebrate successful report submissions accordingly. In the fiscal year 2023" - and I don't know how their fiscal calendar is aligned, but presumably it just closed. He says: "We awarded" - and this number surprised me - "$3.9 million in bug bounties to hundreds of researchers and over $7 million to date since the program began." So props to Zoom.

He said: "Beyond identifying vulnerabilities, outside researchers' support has helped us make other forms of progress at Zoom. We used these reports to demonstrate items that needed attention, flag root-level causes for issues, create better cross-functional alignment, and find potential threats before they become a problem. As a result, our time-to-resolution for bug bounty reports has significantly improved over the past two years. At the start of this year, we restructured our team and developed updates for the program for FY24. We evaluated the researchers currently in our program to make sure everyone is active and contributing. We want to put the right foot forward in the new year, and that all starts by working with high-caliber, effective researchers.

"Zoom's Bug Bounty program is also implementing a brand new Vulnerability Impact Scoring System to help researchers do their best work yet. While we will continue to use the industry standard Common Vulnerability Scoring System (CVSS) to score reports, we're evolving our program to add a companion scoring system called the Vulnerability Impact Scoring System (VISS) that analyzes 13 different aspects of impact for each vulnerability reported as they relate to Zoom infrastructure, technology, and security of customer data. With the implementation of VISS, Bug Bounty can focus more on measuring responsibly demonstrated impact, rather than the theoretical possibility of exploitation."

So then he finishes with the road ahead: "As the Zoom Bug Bounty program has grown over the past year, we're continuing to evolve and mature our processes, bug bounty awards, and testing scope. We're very excited to see the impact of our new scoring system and all the good our researchers can do in 2023. If you're interested in helping to make Zoom more secure, email your HackerOne profile name to bugbounty@zoom.us or visit the Zoom Careers page to review the open positions within the Trust and Security team. Happy hacking." So I am very impressed.

Leo: It's, I mean, the large number is good; right? It means...

Steve: Yes. Yes.

Leo: ...they're finding it.

Steve: And they're being actively proactive.

Leo: Yeah.

Steve: This is what being proactive about security looks like. You know, yes, we all know, since we chronicled those early failures that they made, that Zoom was initially caught flat-footed when their platform took off. But today Zoom's security team is actively, not passively, managing their bug bounty program. And I think that's clearly making a big difference. Not only are they clearly paying well for bugs that are being found, and they're willing to shell out so far $7 million, but they're not just passively listed over at HackerOne and claiming for the sake of a bullet point on a presentation slide that, oh, yes, we offer bug bounties. No, they're serious about tightening up their platform. And, you know, it was Alex Stamos that they hired, or they brought in as a consultant, as I recall; right?

Leo: Consultant, yeah, yeah, yeah. Oh, but not just that. I mean, they really did - they had bought a crypto company that I use, and I was unhappy that they bought it. But some of the best cryptographers in the world are now working there. I mean, they did the right thing. I think sometimes, you know, you hear these numbers like, oh, look at all the flaws that were found. And I think it makes people think, oh, it must be insecure software. But really that's a good thing, to find the flaws and fix them. Everything has flaws; right? I mean, not of course SpinRite. But everything else has flaws.

Steve: Well, okay. So, yes. So what we keep seeing, we see example after example where something looks great until you look at it more closely.

Leo: Yeah.

Steve: And that's what it takes. As soon as you start scrutinizing, as soon as anybody, a good guy or a bad guy, starts scrutinizing it, you're going to find problems. So yes, they're saying that they have paid out for hundreds of bug bounties. But those are hundreds of fewer bugs that are there now.

Leo: I think a lot of companies would be reluctant to say those numbers because they would assume people are going to say, wow, you really have, you know, your product's like Swiss cheese.

Steve: Right. And the point is it once was, and now it is way less so.

Leo: Now it's not, yeah, yeah.

Steve: So, you know, it is by examining those things. So this other cool thing, that H1-702 event that Roy referred to, that was a multi-day HackerOne event held in Las Vegas, Nevada last August. And Zoom was one of two corporate sponsors of the live hacking event on the 4th of August, during which more than 100 security professionals - about 70 of them were in-person and 40 were virtual - from 29 countries hacked the Zoom web and desktop client, the APIs, Zoom's Marketplace apps, and any of the binaries that Zoom distributes. Five individual awards were distributed, and overall Zoom paid roughly $480,000 in bounties in that one day. They said that they feel this is a reflection of the importance of this industry best practice, meaning paying bounties for responsibly reported bug discoveries.

They have come a long way from where they started when they first popped onto our radar, and back then it was in less than stellar fashion. So today I say bravo, Zoom. I really think they're doing, you know, this is the way to do it. They understood, if they wanted to hold onto their position as like this suddenly popular video teleconferencing system, they needed to fix their security. You know, it just hadn't been looked at that closely. So I NUIT.

Leo: Uh-oh.

Steve: Now, that's not as "I knew it." That's NUIT, which in French means nighttime.

Leo: Nuit. Nuit.

Steve: Nuit. Okay. The knights who say nuit. It's an acronym in this case for Near-Ultrasound Inaudible Trojan.

Leo: Oh, boy. That doesn't sound good.

Steve: Uh-huh. Not good.

Leo: Does not sound good.

Steve: And so, yes, some clever researchers are again going to entertain us with their out-of-the-box thinking. Researchers from the University of Texas at San Antonio and the University of Colorado at Colorado Springs recently published a paper for presentation, or I should say submitted because it's not published yet, submitted a paper for presentation during the upcoming USENIX Security 2023 conference being held in April next month. It demonstrates a novel inaudible voice Trojan attack which exploits vulnerabilities of smart device microphones and voice assistants like Siri, Google Assistant, Alexa, Cortana and so on.

The researchers used their Near-Ultrasound Inaudible Trojan (NUIT) to attack different types of smart devices, bridging from smart phones to smart home devices, or sometimes just within the smart phone itself. The results of their demonstrations show that NUIT is effective in maliciously controlling the voice interfaces of popular tech products, and that those tech products which are currently on the market are vulnerable.

Leo: Oh, this is really bad.

Steve: It's not good. It takes advantage of the fact that digital assistants use microphones which accurately pick up sounds that are inaudible to the human ear. NUIT plays sounds in a near-ultrasound frequency range from 16 to 20 kHz, which enables it to give voice commands to both close and more remote smart devices.

Leo: Yeah, because that travels really well, that high frequency stuff. Wow.

Steve: It does. Now, their research demonstrated that NUIT-style near-ultrasound commands can be embedded pretty much anywhere. An attacker could direct - and the demos, they've got like you just play a YouTube, and your phone lights up and does something. It's really...

Leo: Oh, it's terrible.

Steve: It's freaky. So an attacker could direct a victim to click a link to a website that would play some audio, or a YouTube video that would then play the inaudible voice commands. The researchers demonstrated that NUITs also work when playing from one

phone which controls another, over Zoom calls, playing on a phone to control a smart speaker or another IOT device, or even embedded into files that have background music, and it'll still work through that.

Once they have unauthorized access to a device, hackers can send inaudible action commands to reduce a device's volume and prevent the voice assistant's response from being heard by the user before proceeding with further attacks.

Okay. So I was unable to find their full research paper online; and the USENIX conference, as I mentioned, isn't until next month. So some puzzles remain. In some summary coverage published by their universities, they're quoted saying that to wage a successful attack against voice assistant devices, the length of malicious commands must be shorter than 0.77 seconds.

Leo: Oh, that's pretty quick.

Steve: So that, yeah, so three quarters of a second, but we don't know why that's the case until their formal paper is published. They did add that the vulnerability is created due to the nonlinearity of the microphone design, which the manufacturer would need, they said, to address. And the researchers said that out of the 17 smart devices they tested, Apple's Siri devices alone needed to capture and reuse, that is, replay their user's voice, while other voice assistant devices were activated by using any voice or a robot voice. They also pointed out that the attack could be surreptitious because it was possible to silence Siri's response since iPhones maintain separate volumes for Siri and non-Siri output.

And of course as anyone knows who's been around voice assistants, users of voice assistants experience odd triggering events, right, where it didn't appear that the system was being addressed when it suddenly woke up and said, you know, "Hey, boss, what do you want?" So we know that these are the result of their microphones hearing and responding to a much wider range of frequencies than humans do, and constantly listening for that trigger. So there's still a lot that we don't know about the mechanism of the attack, but there is an interesting opportunity for a bit of science and math conjecture here.

There was the comment made that the attack is due to nonlinearities in the operation of these microphones. And that provided the clue for me. That almost certainly means that the instantaneous response to air pressure sound waves is not linear. Now, if the response was nonlinear at normal operating volume, the result would be unacceptable distortion. That's what we call distortion. But the nonlinearity is likely to be extreme at very low volume levels where that nonlinearity doesn't matter.

Any time you have a nonlinear response, the addition of two inputs along that nonlinear response curve is wonderfully turned into multiplication. And although this may initially be counterintuitive, this is the principle of logarithms, and it's the way a slide rule, which adds linear lengths, is able to produce multiplication. The scales of a slide rule are logarithmically nonlinear. So when you're adding linear lengths on a slide rule, you're performing multiplication.

This means that, if we had two sine waves at very low volume, their summation by the device's microphone having a nonlinear response at low volume would have the effect of multiplying their values in real time.

Leo: Instead of adding, they'd multiply.

Steve: Instead of adding, exactly. So next we add one of my favorite trigonometric identities - don't we all have a favorite trigonometric identity? - which states that the product of two sine waves is equal to the sum and difference of their frequencies. And Leo, you have your amateur radio operator's license.

Leo: I do, yeah.

Steve: So you know of this as heterodyning. In radio, heterodyning is the way a radio's local oscillator is able to bring a radio frequency signal down into audible frequency range. What we hear is the difference between the two frequencies, neither of which are audible. And that's exactly what's happening here in this attack. The researchers are generating a pair of near-ultrasonic frequencies whose difference is the voice signal that they're using to control other devices. We don't hear anything. But the microphones in those devices, which are always straining to hear our commands, believe that they're hearing our voice because inaudible sine waves are being made to heterodyne.

Leo: Wow. That's quite clever.

Steve: Isn't that cool?

Leo: Yeah.

Steve: Yeah. And of course a problem because now regular audio stuff, audio material from wherever, could be containing commands that we can't hear. So, and their demonstrations are really chilling. So anyway, I'll keep my eye out for the paper when it's published next month. And if there's anything more, we'll loop back to it.

Okay. So the news is that Microsoft has started testing a cryptocurrency wallet which they are planning to build into their, let's just say "increasingly versatile" Edge browser.

Leo: Wow.

Steve: I know. I loved Ars Technica's take on this. Their headline read "Microsoft is testing a built-in cryptocurrency wallet for the Edge browser." Then it had the subhead "Crypto wallet would join coupons, cash back, and 'buy now, pay later' add-ons." And in the show notes I have two screenshots which Ars showed. So Andrew Cunningham is Ars Technica's Senior Technology Reporter whose take on this is, I think, spot-on. So here's how Andrew explained and characterized this new find.

He wrote: "Microsoft appears to be testing a built-in cryptocurrency wallet for Edge, according to screenshots pulled from a beta build of the browser. The feature, which the screenshots say is strictly for internal testing, was unearthed by Twitter user @thebookisclosed, who has a history of digging up present-but-disabled features in everything from new Windows 11 builds to ancient Windows Vista betas."

He says: "This is one of many money and shopping-related features that Microsoft has bolted onto Edge since it was reborn as a Chromium-based browser a few years ago. In late 2021, the company faced backlash after adding a 'buy now, pay later' short-term

financing feature to Edge. As an Edge user, the first thing I do," he writes, "in a new Windows install is disable the endless coupon code, price comparison, and cash-back pop-ups generated by Shopping in Microsoft Edge." And then he says, in parens, "(Many settings automatically sync between Edge browsers when you sign in with a Microsoft account. The default search engine and all of these shopping add-ons need to be changed manually every time)." Meaning they're deliberately apparently not synchronizing.

He says: "According to the screenshots, the crypto wallet is 'embedded in Edge, making it easy to use without installing any extension,' and it can handle multiple types of cryptocurrency. It will also record transactions and the value of your individual currencies as they fluctuate. An 'explore' tab offers news stories relevant to cryptocurrency, and an 'assets' tab will let you stare lovingly at your NFTs."

Leo: Which is all you can do with them. So, good, good, I'm glad, yeah.

Steve: Yeah, exactly. "The wallet is 'non-custodial,' also called 'self-custodial,' meaning that you have sole ownership of and responsibility for the passwords and recovery keys that allow access to your funds. Microsoft won't be able to let you back in if you lose your credentials."

Leo: Good. That's how it should be.

Steve: Yup. "Whether you find these kinds of add-ons useful, annoying, or predatory is a matter of perspective. Given the prevalence of crypto scams, there may be some value in having a 'trustworthy,'" he has in quotes, "built-in option that doesn't require the installation of dodgy third-party extensions. But the feature could also encourage casually interested users to begin exploring the world of cryptocurrency, which is, again, rife with scams.

"It's also yet another example of Microsoft building a not strictly browsing-related feature into its web browser. Many of these features can be disabled, and competing browsers like Chrome and Firefox all attempt to add value and earn money by building-in access to new niche features and third-party services. But Microsoft's moves can still have an outsize impact that deserves extra scrutiny. Edge is an installed-by-default, non-removable component of every Windows 10 and Windows 11 PC." And Leo, I'm endlessly entertained by you and Paul talking on Windows Weekly about its refusal to go away. And Lorrie, my wife, is like, honey, why does Bing keep coming back?

Leo: Oh, god.

Steve: I know. Anyway, he says: "And the operating system pushes you to switch to Edge with some regularity. And once in Edge, the browser pushes you to use Bing and other Microsoft services." So he finishes: "Microsoft may not ship the crypto wallet to Edge users. The company regularly tests features in Edge, Windows, and its other software that never end up making it into the general-release versions. We've contacted Microsoft for more information and will update if we receive a response."

So, you know, I don't know. Not long ago I tried to use Bing. I had become annoyed with Chrome because I noticed that every time I opened it the fan on my little Intel NUC would spin up to dissipate the heat that Chrome was, for some reason, causing the whole system to produce. And this was with no tabs loaded, just Chrome itself. And suddenly,

whirr, you know, spin up the propellers, we've got to cool this puppy off. You know? So Chrome had become bloatware.

And I didn't know whether Bing might be any better, but I did need left-side tabs, so Bing's built-in support of that feature drew me in. I figured that being first and foremost a Chromium-based browser, I'd at least get good compatibility. But then I found as I was using it that some web pages would not open or display in Bing. So back to Firefox I went, where I am once again completely happy. So if Microsoft decides to embed a cryptocurrency wallet in Bing, you know, I do hope it works better than their email solutions have, which we will be talking about next.

Leo: Oh, boy.

Steve: So that we don't break my rant in half...

Leo: Yes, we'll do an ad, yeah.

Steve: Let's tell about why we're here, and then, oh, boy.

Leo: The rant is on the way, baby.

Steve: At least it's not a faux rant. It's real.

Leo: We had a caller on Ask the Tech Guy on Sunday whose computer, without his approval, upgraded to Windows 11 and turned on some sort of weird security mode that he could only install stuff from the store, so he couldn't install Chrome or Firefox. It just drives me nuts. It just drives me nuts.

Steve: Well, we will shortly see another example of them throwing their weight around.

Leo: Yeah, yeah.

Steve: And it almost makes sense that this is where they would have gone; right? When you're that big, and you need to keep your shareholders happy, you just take advantage of the fact that your users no longer have a choice.

Leo: Cory Doctorow calls it "eating your seed corn." You know, at some point in the company, you know, you just start devouring everything for profit. All right. Put your rant helmet on. Your goggles.

Steve: So I should start out by noting that it's been quite some time since the listeners of this podcast have heard me really get upset about anything. It doesn't happy very often. And I don't recall the last time it happened, but it's happened before. When I dwell on this one, I'm pretty sure my blood pressure rises because I have a real problem with injustice and bullying. Someone at Microsoft has had a very bad idea.

When I first encountered this yesterday, I did a double-take. Really. I thought that I must have misunderstood what Microsoft meant. But unfortunately, no. Microsoft has formally announced that they are going to begin blocking incoming email to their Exchange Online cloud instances, which includes all of Office 365 and Outlook.com, if that incoming email originates from other private so-called on-premises Exchange servers which, while they may be functioning just fine, are nevertheless past their end-of-support life. Like, wow, what?

That's right. They're saying that they are going to begin blocking incoming email from older version instances of their own Exchange Server software. No one who purchased Exchange Server 2007, 2010, or 2013 was told at the time in fact they've not been told until now that in the future, the software they purchased, paid for, and have continued to happily use, would become less useful to them because Microsoft's various online services were going to unilaterally begin refusing to accept email from those otherwise perfectly functioning servers.

And, yes, I use the term "perfectly functioning" in the context of Exchange Server with a bit of tongue in cheek because, after all, it is Exchange Server. But many hundreds of thousands of instances of it are still functioning, and everyone else in the world will be able to receive the email they send except for Microsoft's services because Microsoft has apparently decided to punish their previous customers and extort them for additional licensing revenue.

Now, interestingly, the headline on Microsoft's own announcement doesn't quite 'fess up to this fully. It reads "Throttling and Blocking Email from Persistently Vulnerable Exchange Servers to Exchange Online." But they define the term "persistently vulnerable" as meaning "servers that are unsupported or remain unpatched." And just wait until you hear their rationale for doing this. My breath is still a bit taken away, but I want to repeat my summation so that you don't need to hit rewind or replay to be sure you heard it right.

Microsoft is essentially saying that they are going to use their market dominance in online email services to force their previous software customers to upgrade their own instances of Exchange Server by refusing to accept email sent by such servers as a means of extorting additional licensing fees from those prior customers. Microsoft is going to effectively begin reducing the functionality of their previous Exchange servers by refusing to accept their email unless and until the licenses for those Exchange servers are renewed, and their software is updated.

I can't think of a precedent for this in our industry. So I suppose that means that this is an unprecedented action. Okay. So here's how Microsoft couched this extortion. They wrote, this is them: "As we continue to enhance the security of our cloud, we are going to address the problem of email sent to Exchange Online from unsupported and unpatched" - yes, Leo, you're laughing, I know. "The problem."

Leo: We are not making enough money from legacy customers.

Steve: Yeah, we realize there are some people that we haven't squeezed yet enough. So anyway, they said: "We're going to address the problem of email sent to Exchange Online from unsupported and unpatched Exchange servers. There are many risks associated with running unsupported or unpatched software, but by far the biggest risk is security." Especially with Exchange. "Once a version of Exchange Server is no longer supported, it no longer receives security updates; thus, any vulnerabilities discovered after support has ended don't get fixed." Oh, the horror. "There are similar risks associated with running software that is not patched for known vulnerabilities. Once a

security update is released, malicious actors will reverse engineer the update to get a better understanding of how to exploit the vulnerability on unpatched servers."

Okay, yeah. As we know, all of that's true. But next comes the pivotal paragraph, the fundamental flaw in the logic upon which this entire extortion effort rests. Microsoft continues: "Microsoft uses the Zero Trust security model for its cloud services, which requires connecting devices and servers to be provably healthy and managed. Servers that are unsupported or remain unpatched are persistently vulnerable and cannot be trusted, and therefore email messages sent from them cannot be trusted." They said: "Persistently vulnerable servers significantly increase the risk of security breaches, malware, hacking, data exfiltration, and other attacks." Okay. So repeating...

Leo: To put this in perspective, it's as if your modern iPhone would no longer accept text messages from iPhones that were out of date.

Steve: Yes.

Leo: Like, oh, you can't send text messages. You have an iPhone 6. Sorry.

Steve: Right. We can't trust that anymore. We're not supporting that.

Leo: In their defense, is that, I mean, it could be hacked; right? Because they haven't patched it. And Exchange servers are notoriously bad.

Steve: Oh, yes. In fact, I have an update on that, too. Okay. But here's the problem. They said: "Servers that are unsupported or remain unpatched are persistently vulnerable and cannot be trusted. Therefore, exactly as your example is, the email messages sent from them cannot be trusted."

Leo: And those could be validated. I mean, they should just be text; right? I mean, they're not...

Steve: Okay. I have in the show notes here, I wrote: "This is what's commonly known as a load of crap."

Leo: Oh, yes, that.

Steve: That, yes. And I had two phrases. That was the more polite of the two. It's important that we pause here for a minute because, as I said, it is upon this fundamental logical fallacy that Microsoft is hanging their entire campaign. We know that the truth is that, yes, through the years Microsoft's Exchange Server has had a particularly difficult relationship with security. In short, it's pretty much been an utter disaster. And not just for a while. It's inexplicable that they've had this problem, especially when email is a trivial protocol. I mean, there's hardly anything simpler than email. But Microsoft has managed to make it deadly.

On January 3rd of this year, Bleeping Computer's headline read: "Over 60,000 Exchange servers vulnerable to ProxyNotShell." And few months earlier, on October 27th, Wired's headline, and I'm not making this up, was "Your Microsoft Exchange Server Is a Security Liability" was their headline, with the subhead "Endless vulnerabilities. Widespread hacking campaigns. Slow and technically tough patching. It's time to say goodbye to on-premise Exchange." As an aside, I'll remind everyone of just how conscientious Microsoft has been about Exchange's security.

In that article, describing the constant struggle over Exchange vulnerabilities, Wired wrote, they said: "The latest reminder of that struggle arrived earlier this week, when Taiwanese security researcher Orange Tsai published a blog post laying out the details of a security vulnerability in Microsoft Exchange. Tsai warned Microsoft about this vulnerability as early as June of 2021. And while the company responded by releasing some partial fixes, it took Microsoft 14 months to fully resolve the underlying security problem.

"Tsai had earlier reported a related vulnerability in Exchange that was massively exploited by a group of Chinese state-sponsored hackers known as Hafnium, which last year penetrated more than 30,000 targets by some counts. Yet according to the timeline described in Tsai's post last week, Microsoft repeatedly delayed fixing the newer variation of that same vulnerability, assuring Tsai no fewer than four times that it would patch that bug before pushing off the full patch for months longer. When Microsoft finally released a fix, Tsai wrote, it still required manual activation and lacked any documentation for four more months.

"Meanwhile, another pair of actively exploited vulnerabilities in Exchange that were revealed last month still remain unpatched after researchers showed that Microsoft's initial attempts to fix the flaws failed. Those vulnerabilities were just the latest in a years-long pattern of security bugs in Exchange's code. And even when Microsoft does release Exchange patches, they're often not widely implemented, due to the time-consuming technical process of installing them."

Okay. So let's first just be really clear here. Any security problems that Exchange Server has are directly Microsoft's fault. Right? So now they're saying that they're afraid of receiving any email that previous versions of their software might send to their current versions. It's true that the poor users of those previous versions are likely risking life and limb to continue running older and out-of-date versions of Exchange. But that's their choice. Or it used to be. And it's they who are being put in danger by running Microsoft's perennially insecure offering, not people who receive the email it sends. That's utter nonsense. But it's the utter nonsense upon which, as I noted, this entire campaign rests.

So here's how Microsoft continues. They said: "We've said many times that it is critical for customers to protect their Exchange servers by staying current with updates and by taking other actions to further strengthen the security of their environment. Many customers have taken action to protect their environment, but there are still many Exchange servers that are out of support or significantly behind on updates."

Okay. So Microsoft has coined a new term of art for this which will be entering our industry's lexicon. They are calling what they will be employing their "Transport-based Enforcement System." They said: "To address this problem" - which of course they've invented - "we are enabling a transport-based enforcement system in Exchange Online that has three primary functions: reporting, throttling, and blocking. The system is designed to alert an admin about unsupported or unpatched Exchange servers in their on-premises environment that need remediation, upgrading or patching. The system also has throttling and blocking capabilities so, if a server is not remediated, mail flow from that server will be throttled - delayed - and eventually blocked."

Leo: Wow.

Steve: "We don't want to delay or block legitimate email, but we do want to reduce the risk of malicious email entering Exchange Online by putting in place safeguards and standards for email entering our cloud service." And then they said: "We also want to get the attention of customers who have unsupported or unpatched Exchange servers and encourage them to secure their on-premises environments."

That's right. "Encouragement" is what we call it here at Microsoft when your email server's outbound mail starts being refused, not due to any misbehavior on its part, which would be your problem in any event, but only because you have chosen to continue using an older version of Exchange Server - which you paid for - whose email output we have since decided we no longer want to accept. Oh. But if you purchase a new license, then all will be forgiven, and we'll happily receive your email.

They also explain about the actions they'll be taking about, you know, progressive amounts of delaying and how they'll send back an SMTP 450 message delayed message and then ending in an SMTP 550 error to the sender.

So after all this, the announcement then contains an FAQ to clarify, in Q&A format, what this all means. The answers to a few of the questions they ask themselves are a bit chilling. So they ask themselves: What is a persistently vulnerable Exchange server? Answer: Any Exchange server that has reached end of life - for example, Exchange 2007, Exchange 2010, and soon, actually next month, Exchange 2013 - or remains unpatched for known vulnerabilities. For example, Exchange 2016 and Exchange 2019 servers that are significantly behind on security updates are considered persistently vulnerable.

Is Microsoft blocking email from on-premises Exchange servers to get customers to move to the cloud? No. Our goal is to help customers secure their environment, wherever they choose to run Exchange. The enforcement system is designed to alert admins about security risks in their environment - because their email can't get delivered anymore - and to protect Exchange Online recipients from potentially malicious messages sent from persistently vulnerable Exchange servers. Because you know how bad the email is that's sent from those old Exchange servers.

Okay. Why is Microsoft only taking this action against its own customers, customers who have paid for Exchange Server and Windows Server licenses? Whoa. We are always looking for ways to improve the security of our cloud and to help our on-premises customers stay protected. This effort helps protect our on-premises customers by alerting them to potential significant security risks in their environment. You know, created by our software. We are initially focusing on email servers - oh, initially focusing? We are initially focusing on email servers we can readily identify as being persistently vulnerable, but we will block all potentially malicious mail flow that we can.

Will Microsoft enable the transport-based enforcement system for other servers and applications that send email to Exchange Online? We are always looking for ways to improve the security of our cloud and to help our on-premises customers stay protected. We are initially focusing on email servers we can readily identify as being persistently vulnerable, but we will block all potentially malicious mail flow that we can.

If my Exchange Server build is current, but the underlying Windows operating system is out of date, will my server be affected by the enforcement system? No. The enforcement system looks only at Exchange Server version information. But it is just as important to keep Windows and all other applications up to date, and we recommend customers do that. However, we haven't figured out how to make you do it yet.

Delaying and possibly blocking emails sent to Exchange Online seems harsh and could negatively affect my business. Can't Microsoft take a different approach to this? Microsoft is taking this action because of the urgent - that's right, we just made it up, but now it's urgent - because of the urgent and increasing security risks to customers that choose to run unsupported or unpatched software. Over the last few years, we have seen a significant increase in the frequency of attacks against Exchange servers. You bet.

We have done, and will continue to do, everything we can to protect Exchange servers; but unfortunately there are a significant number of organizations that don't install updates or are far behind on updates, and are therefore putting themselves, their data, as well as the organizations that receive email from them, at risk. We can't reach out directly to admins that run vulnerable Exchange servers, so we are using activity from their servers to try to get their attention. Our goal is to raise the security profile of the Exchange ecosystem.

Why are you starting only with Exchange 2007 servers, when Exchange 2010 is also beyond end of life, and Exchange 2013 will be beyond end of life when the enforcement system is enabled? Starting with this narrow scope of Exchange Server lets us safely exercise, test, and tune the enforcement system before we expand its use to a broader set of servers. Additionally, as Exchange 2007 is the most out-of-date hybrid version, it doesn't include many of the core security features and enhancements in later - oh, so it's less bad, even though it's way older. Oh, no, it's more bad, more bad. The newer ones are less bad. Restricting the most potentially vulnerable - oh, it's the most potentially vulnerable - and unsafe server version first makes sense. That's right.

Does this mean that my Exchange Online organization might not receive email sent by a third-party company that runs an old or unpatched version of Exchange Server? Possibly. The transport-based enforcement system initially applies only to email sent from Exchange 2007 servers to Exchange Online over an inbound connector type of on-premises. The system does not yet apply to email sent to your organization by companies that do not use an on-premises type connector. Our goals are to reduce the risk of malicious email entering Exchange Online by putting in place safeguards and standards for email entering the service and to notify on-premises admins that the Exchange server their organization uses needs remediating.

In other words, the answer is yes to this question. If you're an organization using Exchange Online, you will stop getting email from these scofflaws who have decided to continue using the software they purchased a while ago.

How does Microsoft know what Exchange version I'm running? Does Microsoft have access to my servers? No, Microsoft does not have any access to your on-premises servers. The enforcement system is based on email activity, for example, when the on-premises Exchange Server connects to Exchange Online to deliver email. Anyway, it's in the headers, the version of Exchange and the service pack and update level and so forth.

And in this posting they have some additional back-and-forth between some people who feel pretty much as I do and a Microsoft guy. But everyone's got the sense for this.

So the world is apparently full of past-end-of-life and non-updated Exchange Server instances that are day-in and day-out working perfectly well for their users who once purchased the software from Microsoft and who have felt for whatever reason no need to purchase new licenses. The server's online and working for them the way they want it to, without problems. And for whatever reason they have chosen not to upgrade their software. Is it not their choice and their right to run the software that they have purchased as long as they wish and as long as it is serving their needs? That software was not a subscription. It did not come with an expiration date. Its license's clear assertion was that they would be able to use it for as long as they wished.

The idea that email originating from out-of-date and unsupported email servers is itself inherently dangerous is utter nonsense. There's no awareness out in the world that email from certain versions of this or that email software is known to be dangerous. We would know that on this podcast if that was the case. Microsoft just invented that out of thin air to suit their commercial purpose. To the extent that there is any danger, it's to the organization running such software, not to those receiving its email output.

This is the shaky premise upon which Microsoft's policy rests, and it doesn't hold up. But that doesn't matter because Microsoft is able to claim it, and will soon begin enforcing it. Those perfectly good servers have been tirelessly working, apparently without problems, for many years. But now that's about to change, not because they suddenly represent any actual danger, but because Microsoft figured out how they can extort some new revenue from their old customers.

Because most organizations will take the path of least resistance, which will be to upgrade, Microsoft surely knows that the net effect of this will be to generate additional revenue for itself. Those wayward users have fallen off the Microsoft gravy train, and it's high time that they be brought back into the fold. What better way than to create a new inability for those old servers to send mail to Office 365 and Outlook.com users whom Microsoft already utterly controls? It's diabolically brilliant. And so that it doesn't appear to be the bald-faced revenue extortion scheme that it is, couch the whole thing in their need to protect their paying customers from those evil email messages being generated by their own presumably now highly dangerous software.

And finally, it's worth noting that this is not a one-time event. No one using any Microsoft Exchange Server software will ever again be able to fail to keep it updated, nor to avoid the purchase of future licenses, forever. Now, I celebrate that idea moving forward, since keeping software updated, especially Exchange Server, is a good thing, and since new licensors will be aware that they will never again have any choice other than to be paying Microsoft forever under whatever terms and conditions Microsoft may choose.

The problem I have is with Microsoft's effectively unilateral revocation of the open-ended licenses they previously sold. That's a clear violation of trust, and that's never going to be okay, even though it's going to happen.

Leo: Yeah, I mean, I'll have to ask Richard Campbell about this. He is an Exchange Server admin, is abandoning Exchange Server for his home system. Anybody who has been using Exchange Server has been tortured long enough.

Steve: Yeah.

Leo: But this seems like a really unseemly way - because you're basically breaking email.

Steve: Correct. You are. You are saying, here is a valid piece of email with a letter from your mom that has no malware on her system. But because it was sent to an Office 365 or Outlook.com user through an old version of Exchange Server, they're not going to accept the email.

Leo: That's not okay.

Steve: They're just going to say, oh, no, we're not going to accept the email. I mean, Leo, it is so wrong.

Leo: Yeah.

Steve: Talk to you in four weeks.

Leo: All right. Thanks, Steve.

Steve: Okay, buddy.