SECURITY NOW!

Transcript of Episode #838

Autodiscover.fiasco

Description: This week we examine a new pair of zero-days which have forced emergency updates to their respective products. We examine the growing annoyance of those who are reporting bugs to Apple, Epik's belated confirmation of their mega data breach, Windows 11's further progress toward its release, and its new and much more useful PC Health Check tool. We look at some additional fallout from this month's ever-exciting Patch Tuesday, and take notice of a clever new approach for bypassing antimalware checking under Windows. And after a quick check-in about the first two episodes of Apple TV's "Foundation" series, we settle in to examine the week's most explosive, worrisome, and somewhat controversial disclosure of yet another huge Microsoft screw-up which caused this week's episode to be given the domain name "Autodiscover.fiasco."

High quality (64 kbps) mp3 audio file URL: http://media.GRC.com/sn/SN-838.mp3
Quarter size (16 kbps) mp3 audio file URL: http://media.GRC.com/sn/sn-838-lq.mp3

SHOW TEASE: It's time for Security Now!. Steve Gibson is here. Great show for you. We have the 12th, yes, 12th zero-day in Chrome this year alone; a Windows 10 emergency update; Steve's review of "Foundation"; and then we'll find out who owns Autodiscover.wtf and why. It's all coming up next on Security Now!.

Leo Laporte: This is Security Now! with Steve Gibson, Episode 838, recorded Tuesday, September 28th, 2021: Autodiscover.fiasco.

It's time for Security Now!, the show where we cover your security online with this guy right here, Steve Gibson. A good day, sir.

Steve Gibson: Good day to you. This is...

Leo: You were waiting for me to say something else, like, "I said, good day, sir."

Steve: I don't know what I'm doing.

Leo: Good day, sir.

Steve: This is Episode 838, the last episode of September. We bid September adieu.

Leo: Bye-bye.

Steve: This is the first podcast ever to be named or to be given a domain name, which you and I before we began recording were lamenting the fact that .fiasco is not actually a valid TLD because...

Leo: Oh, it'd be so good.

Steve: ...couldn't we have some fun with that. I titled this "Autodiscover.fiasco" for reasons that will become painfully clear to everyone as they have become clear to Microsoft in the past week. Bu we'll get to there in a minute. I forgot to query about the Picture of the Week, Leo. I figured you'd be picking yourself up off the floor, thinking that it was particularly clever. We will do that in a minute.

Leo: I don't want to give anything away, but it's very good, yes.

Steve: We're going to examine a new pair of zero-days which have forced emergency updates to their respective products. And I need to scold the industry yet again about the overuse of the term "zero-day." That just is - everyone's in love with it. But it's like, no, things that are vulnerabilities are not automatically...

Leo: Are not zero-days.

Steve: Yes, exactly. And it's like, oh, no, here comes a zero-day. It's like, okay, no. Anyway, we're also going to examine the growing annoyance of those who are reporting bugs to Apple. I didn't talk about it last week, but it just - there's another instance of it this week, and I thought, okay, we just have to touch on this. I picked up something on MacBreak Weekly about - that sounded like you guys were mentioning that.

We also have Epik - remember Epik.com - their belated confirmation of what everybody else knew, their mega data breach, which is remember the thing that caused Have I Been Pwned to alert me. We've also got Windows 11's further progress towards its release and a new and much more useful PC Health Check Tool from Microsoft, which actually is this week's shortcut of the week for the podcast, if anyone wants to jump ahead and put in - oh, no, wait. I didn't give it the number. I gave it "slash check" so that it would have a longer life, grc.sc/check. Anyway, we'll get to there.

Also we look at some additional fallout from this month's ever-exciting Patch Tuesday, and take notice of a clever new approach for bypassing antimalware checking under Windows. And then, after a quick check-in about the first two episodes of Apple TV's "Foundation" series, we're going to settle in, as I said, to examine the week's most explosive, worrisome, and somewhat controversial disclosure of yet another huge Microsoft screw-up which caused this week's episode, as I said, to be given the domain name "Autodiscover.fiasco."

Leo: Aw. I love it. All right. We will get to all of that and, yes, a stellar Picture of the Week. A visual pun, if you will.

Steve: So I don't know that I can actually describe this adequately. It was easier to talk about messy cabling closets because people could just, you know, you had a model to go from there. I titled this, I think sort of correctly, "A Logic Gate." It's actually a metal gate in sort of a narrow alley. And its vertical bars are made from beautifully crafted, I mean, this is a work of art, logic gate symbols, where for example, you know, a NAND gate or an AND gate has a shape to it. It's got typically two inputs and an output.

Well, so there's two vertical bars going up to a beautifully rendered metal shaped AND gate, and then one bar coming out of it. We've got inverters. We've got some, I don't know what that thing is, it's supposed to be an OR, but it's only got one input. So maybe it's not. And unfortunately the balls which are used to perform the welding, they don't all have balls, so it wasn't necessary because that's the typical inversion symbol in digital logic notation. Anyway, just I've had this in the pile of pictures for some time. Thank you, whoever it was who tweeted. I really appreciate this.

Leo: Chickenhead21 says the gate looks closed, but the gates are inverse, so it's really open. This is the ultimate logic gate. Do you think it's actual? It's not really an actual circuit. I don't think it could be.

Steve: No, no. No, no, no. No. Well, and you can sort of see the center bar, we've got some inputs that appear out of nothing, where there's nothing actually feeding them and so forth.

Leo: Yeah, yeah, right, right, yeah.

Steve: But still just, I mean, a beautiful piece of work. So somebody had a sense of humor.

Leo: I love this.

Steve: I don't, you know, what would be really cool, to know what the back story is, like, okay, where did this come from?

Leo: Yeah, that's a lot of work, yeah.

Steve: It is a lot of work, yeah. And it's going to fall on deaf eyes. Wait. Deaf ears? Blind eyes?

Leo: Most people will look at it and have no idea.

Steve: They're going to go, huh?

Leo: Yeah.

Steve: Yeah.

Leo: But not this crowd.

Steve: Not our group. As is the case for the news of the 12th zero-day so far this year.

Leo: Oh, agh.

Steve: We know what that is without any additional description. Last week Chrome's emergency zero-day update, that is to say, last week when we talked about it left us at 930.0.4577.82. But that one didn't last long. Last Friday, Chrome was updated to 94.0.4606.61, with a fix for a single high-priority update for, yes, yet another zero-day that Google says, thus zero-day, they are aware of being exploited in the wild. Now, one thing we're going to be hearing a lot about this week because boy are they becoming prolific, and that's Google's so-called TAG team, their Threat Analysis Group. They're responsible for this one. They found it in their own product.

It's being tracked as CVE-2021-37973. It's a use-after-free, which we know means that some way was found to access memory after the garbage collector had released some memory back to the system, believing that it was no longer necessary. There are languages where you explicitly, you the programmer, explicitly ask for and receive an allocation of memory from the operating system. There are languages where memory is allocated statically so your program just always has it, typically within its own memory space. And then there are so-called automatic languages where you just start to use some memory, and something underneath says, oh, and quickly creates some memory to hold your use.

And the idea is that it's supposed to figure out when you're no longer going to use that anymore because it's, depending upon what you're doing, how long this program's going to be running, the system can't just keep automatically allocating memory every time it sees that you're about to need it, without it also on the backend figuring out, okay, he's done with this now, I can let it go. And there are many instances where, like if memory is allocated inside of a subroutine, when you exit the subroutine, the presumption is, okay, anything you had allocated while in the subroutine should not be accessible outside the subroutine due to what's known as "scoping" in languages. So as soon as you go out of scope, then the operating system could say, okay, he's done with this, and let it go.

Anyway, it is complicated. It's another one of those things that is a great convenience for the programmer. And it comes at some cost to the system of providing that convenience. And it can be a little error-prone. So we keep seeing these use-after-free, which means that after the memory was freed, there was still some means for a malicious programmer to get access to that memory. And who knows, I mean, the memory could be reused by the operating system, and so they'd have access to some memory that was now in use by another process. And that's a big no-no because that breaks interprocess isolation.

So, you know, the problem is we keep finding these, yet we keep writing new code that introduces new problems. So we're the hamster on the little wheel that is never-ending. Anyway, the Google TAG team found the problem. They discovered and reported the flaw to the Chromium people. And it turns out they found it because a bad guy had found it already and was actually exploiting what I just described, tricky as that is, found a way to exploit it in the wild for their nefarious purposes. So quick, push out a new version of Chrome, one more use-after-free problem is now gone. Let's hope we don't see any more soon.

And, you know, Google doesn't give us any more information about this because they want everybody to update before they talk about it. That means that it ends up becoming sort of historical interest only. And then does anyone care anymore? No. So they're probably never going to tell us. And again, no one will care. But for those keeping score at home, this brings a total year-to-date zero-day tally for Chromium to 12.

Leo: Wow.

Steve: So since we're still in month nine until next podcast, you know - and again, I give Google serious props for being as responsive as they are. We're going to be talking about a - in fact, this podcast is named after a real mess that Microsoft has known about for some time, and as the case with the ProxyLogon Microsoft Exchange problems, and the printer nightmare problems where they've known for three months or nine months respectively and did nothing about it. Google jumps on this stuff and gets them fixed in a matter of days. So I'm glad that so many browsers have decided to go the Chromium route, and especially Microsoft's browser.

What was interesting was that the flaw appeared, remember I talked about how they keep writing new code, well, this is in new code. It was in the so-called "Portals API." And I thought, what's the Portals API? Turns out it's a new web page navigation system that enables the user's current web page to show another page as an inset thumbnail. And then, upon some action, maybe you click on it or you move your mouse over it or who knows what, it performs a seamless transition of that little thumbnail to become the next page by smoothly zooming the thumbnail to full size to replace its parent page and to become the new top-level document.

And I have to say, there's a bunch of examples. If you go to web.dev/hands-on-portals, so https://web.dev/hands-on-portals, you'll see some examples of this. And it's kind of slick since it's the sort of effect that we're used to seeing on fancy OS platform UIs. And once it catches on, I'm sure we're going to be seeing it all over the place, often. And maybe more than we want to since it is a bit cutesy-poo. But, you know, as I was looking at it thinking, okay, here's another bit of eyewash for us, it's just going to make GRC look even more Stone Age than it already is. But that's okay, too.

Once I get caught up with SpinRite, probably after 7.2, where we'll have operation on BIOS and UEFI and native support for all drive technologies, I might go for a change of pace and spend a little time on the website. Maybe. I have a feeling, though, that given the weird timing anomalies that we've detected with SSD reading, that that's going to be too much for me to resist, and I might just switch over to work on SpinRite 8.

Anyway, in our zero-day watch we also have Apple. There were urgent Apple iOS and macOS updates which were just released to fix actively exploited zero-days. The day before Google pushed out that most recent high-priority update to Chrome they were just talking about, Apple released security updates to fix multiple security vulnerabilities appearing in older versions of iOS and macOS. Apple says, because Google told them, again this TAG group, that they've been detected in exploits in the wild, thus, yes, true zero-days. Last Thursday's updates also expanded some earlier patches for a previously resolved vulnerability that was being abused by the NSO Group's Pegasus surveillance tool, which is used in targeted attacks on iPhone users.

The most worrisome was a type confusion flaw, another type of mistake that can again be sort of - it crops up in code where some object is referred to as being of a different type, like a string versus an integer, or a floating point value referred to as a string or something. And that can cause all kinds of problems. In this case, it's in a kernel component, the XNU kernel component. And it was being exploited within a deliberately developed malicious application and used to execute arbitrary code with the highest privileges. iOS client applications are never allowed to have kernel root privileges, which they were obtaining through this vulnerability.

And as I said, it was uncovered by Google's TAG team, which said that it had detected the vulnerability being used in conjunction with remote code execution targeting WebKit, which of course is Apple's web engine that they use in Safari and elsewhere. The patches are available for devices running macOS Catalina and iPhone 5s, iPhone 6, 6+, iPad Air, iPad mini 2, mini 3, and iPod Touch 6th generation running iOS 12.5.4. So this was basically fixing older versions of some of these things that turned out to also be under attack.

And speaking of Apple, back on September 9th the Washington Post ran a story titled "Apple pays hackers six figures to find bugs in its software, but then it sits on their findings." And, you know, the implication being that they're sort of buying the silence of hackers who are responsibly reporting problems. Thank you for paying us, but are you going to get around to fixing them? Because this thing was subtitled "Lack of communication, confusion about payments, and long delays have security researchers fed up with Apple's bug bounty program."

Now, as I was assembling the podcast that followed that story, which was discussing the mega Meris or Meris - we were never really sure how to pronounce that - botnet, I read what the Washington Post had to say. Basically it was grumbling from researchers over how Apple's security team was leaving bug reports unsolved for months, shipping incomplete fixes, low-balling their monetary rewards - in their opinion - or banning researchers from their program when they complained about the way Apple was treating them.

Now, so all of that is worrisome. But I decided that it was mostly sort of generalizations that didn't have a lot of meat, like not enough for the podcast. This week, however, we won't be asking where's the beef. Last Thursday, a Russian security researcher named Denis Tokarev, who uses the handle "Illusion of Chaos," having finally given up waiting for Apple to acknowledge and repair three of the four vulnerabilities he had reported to them in April, published full details and proofs of concept on GitHub for three vulnerabilities that Apple had not addressed. And that includes even with the just-released iOS 15 from earlier last week.

The three problems are: a vulnerability in the Gamed daemon that can grant access to user data such as Apple ID emails, names, authentication tokens, and grant file system access. I have a link in the show notes to his discussion of that and his proof of day. Any app installed from the App Store may access, he writes, the following data without any prompt from the user: Apple ID email and full name associated with it; Apple ID authentication token which allows to access at least one of the endpoints on *.apple.com on behalf of the user, meaning an authentication breach; complete file system read access to the Core Duet database, which contains a list of contacts from Mail, SMS, iMessage, third-party messaging apps and metadata about all user's interaction with these contacts, including timestamps and statistics and also some attachments like URLs and texts; and also complete file system read access to the Speed Dial database and the Address Book database including contact pictures and other metadata like creation and modification dates.

The researchers noted that they had just checked on iOS 15, and this last one is now inaccessible, suggesting that it was quietly fixed, although Apple had not acknowledged that to the researcher.

We also have a vulnerability in the nehelper daemon that can be used from within an app to learn what other apps are installed on the device. So that's a breach of security we know other platforms have fixed in the past. And an additional vulnerability in the nehelper daemon, the same one, that can also be used from within an app to gain access to a device's WiFi information.

Denis also published his proof-of-concept code for the fourth issue. Oh, and I should mention each of those three has proof-of-concept code up on GitHub. He also published the proof of concept for a fourth issue which affects the iOS analyticsd daemon. This was the fourth of the bugs he reported to Apple in April and was the only one of his four issues patched, and that was in iOS 14.7, back in July.

His blog posting last Thursday was titled "Disclosure of three zero-day iOS vulnerabilities and critique of Apple Security Bounty program." I have a link to his full posting in the show notes for anyone who wants it. I'm just going to grab the top of it where he begins: "I want to share my frustrating experience participating in the Apple Security Bounty program. I've reported four zero-day vulnerabilities this year between March 10 and May 4. As of now, three of them are still present in the latest iOS version (15.0), and one was fixed in 14.7."

Now, I'll interrupt here to note that this is a perfect example of where calling these zero-days is not correct. They are vulnerabilities, yes. They're not good. No. But what makes zero-days special, as I said before, is that they are discovered being in use in the wild. You know, we want to have some term that gives extra oomph to vulnerabilities, and "zero-day" is that term. Except now everyone's just using it for everything. And, you know, if we don't require that we use them correctly, then the term is going to lose all significance, except to make them sound more scary, which isn't fair.

So anyway, he continues: "But Apple decided to cover it up and not list it" - meaning his findings - "on the security content page. When I confronted them, they apologized, assured me it happened due to a processing issue, and promised to list it on the security content page of the next update. There were three releases since then, and they broke their promise each time.

"Ten days ago," he wrote, "I asked for an explanation and warned them that I would make my research public if I didn't receive an explanation. My request was ignored, so I'm doing what I said I would. My actions are in accordance with responsible disclosure guidelines." And he says, parens, "(Google Project Zero discloses vulnerabilities in 90 days after reporting them to vendor; ZDI in 120.)" He says: "I have waited much longer, up to half a year in one case." He says: "I'm not the first person who is unhappy with Apple Security Bounty program. Here are some other reports and opinions." Whereupon he lists eight publications and three tweets, and one of the publications is iMore. So some respected publications. Then yesterday he blogged an article with the title "How malware gets into the App Store, and why Apple can't stop that." Again, a link to his blog post in the notes.

He just starts out, he says: "Only after I had published a post detailing three iOS zero-day" - okay, which aren't - "vulnerabilities and expressing my frustration with Apple's Security Bounty Program, I received a reply from Apple." And he quotes it: "We saw your blog post regarding the issue and your other reports. We apologize for the delay in responding to you. We want to let you know that we are still investigating these issues" - okay, six months later; right? - "and how we can address them to protect customers. Thank you again for taking the time to report these issues to us. We appreciate your assistance." Even though they're not demonstrating any appreciation. "Let us know if you have any questions."

And so he takes them up on it. He says: "Indeed, I do have questions. The same ones that you have ignored. I'm going to repeat them. Why was the fix for analyticsd vulnerability quietly included in iOS 14.7 update, but not mentioned on its security content list? Why did you promise to include it in the next update's list, but broke your words not once, but three times? Why do you keep ignoring these questions?"

Okay. So given that there's been a lot of coverage of this recently, I wanted to give it some attention. I suspect we're dealing with the collision of egos and busy companies. Researchers doubtless work quite hard to find problems. I mean, these are not easy problems to find, especially in iOS, which fights against allowing anyone to look anywhere. And once found, they feel, the researchers feel possessive of them and want them to be acknowledged and to be fairly compensated for their work. Which after all they are promised, you know, ahead of time. And while the problems that Denis found and reported may not be remote code execution, information disclosure problems of the sort he detailed can be significant, especially with Apple increasingly begging us to trust them, allowing them to carry our purchasing cards and other information and to acquire real-time health data about us.

And I'd also wager that the signal-to-noise ratio among all of the reports of problems that Apple receives probably makes wading through an endless stream of non-problems, looking for true problems, annoying and fatiguing. I get that. This is where we are today. But we're seeing that example after example, and boy are we going to have one at the end of the podcast, of incredibly cash-rich companies like Microsoft and Apple, not appearing to be budgeting the resources that perhaps they should. You know, they get to a position of, like, super-strength, where nothing can touch them any longer, and then maybe inevitably it's like, eh, you know, thanks for the report. We'll let you know maybe. Maybe not. Maybe we'll fix it, and maybe not.

Epik, this slimeball registrar and host, has confirmed their hack. Last week we talked about the email I received when some GRC domain email accounts were obtained from the domain registrar and web host Epik. At the time, Epik was denying that anything had happened. It took them a week to acknowledge what all the evidence showed. They finally did. So I wanted to quickly follow up. Threatpost's updated coverage of this quoted the CTO and cofounder of Blue Hexagon. He explained - and this guy has nothing to do with Epik, but they quoted him to get some context for why this was happening.

And so this CTO and cofounder of Blue Hexagon was quoted by Threatpost saying: "This has happened to a lot of the right-wing outlets, Parler and Gab for example, because they have been brought up in record time to capitalize on current events like the election, vaccines, voting, and deplatforming to be able to fundraise or get traction quickly."

He said: "Unfortunately, this usually means that security takes a backseat due to business pressure." Right? It's like, hey, get us a website. Get us online. And as we know, security takes extra effort. So it's not surprising that security is lagging and lacking. And he says: "Which can result in breaches. Usually, hacktivists are not known to be as sophisticated as nation-state groups or the big-game ransomware operators. But nowadays a lot of tools and malware are for sale and can be used by anyone who is reasonably technically adept at penetrating networks."

And of course last week's topic was about Cobalt Strike, which is precisely that sort of out-of-the-box, turnkey, off-the-shelf hacking tool. So that's what's going to be happening if you're in a big hurry to get yourself online; and, unfortunately, if you paint a target on yourself. And you know, there was so much news this week I didn't have a chance to get to all of it. But I did keep checking back in with VoIP.ms, and this attack continued. At one point a couple days ago they thought they were back online. And then in an updated tweet they said, oh, apparently our U.S. points of presence are back down. And now there's yet another VoIP vendor, Bandwidth.com, which has been suffering another DDoS attack.

Leo: And a TWiT advertiser, by the way.

Steve: Oh, no kidding.

Leo: And by the way, huge because they're the backend for Google Voice and a bunch of other stuff. So when Bandwidth.com goes down...

Steve: Boy.

Leo: Boy, this is a tough ransomware play. You know? I mean, geez.

Steve: Yeah. Yeah.

Leo: It's hard to defend against this stuff.

Steve: Right. As I said, when you're looking at request-based attacks, you need filters for those requests. And nobody's built filters for VoIP requests. And because it's a real-time technology, unlike a web query, where you can afford to kind of bounce it around a bit and make it jump through some hoops, VoIP may be a tricky protocol to protect. And this may just be the beginning of new protocol-specific attacks that we're going to be seeing. So, yeah, I agree with you completely, Leo. It's going to be tricky.

Microsoft is moving forward with Windows 11, getting it closer to release. And it's going to be October 5th is the release date, which is next Tuesday. So that's apparently slated as Windows 11 launch date. Up until now, the Windows Insider Release channel was only offering users Windows 10 21H2, which is their version 19044, which is expected to be released next month. But as of last Thursday, Microsoft is now offering Windows 11 as an optional download within Windows Update for users with compatible hardware. And as for compatible hardware, we'll be talking about that in a minute.

The Win11 build being offered in the Release channel now, as of last Thursday, is 22000.194, which is the release that became available to users in the Beta channel the previous week, on September 16th. Even though the last few beta builds have just been feature-stable bug fixes, to me this seems pretty quick, given how flaky some of those earlier Win11 releases have been. I know that rounding off some pointy corners and changing the look and feel of the Start Menu is no big deal. We're going to see how this comes off. It does sort of feel rushed. I hope I'm wrong.

On the downside is the fact that a bunch of popular longstanding Windows features have been removed from Windows 11, which has upset many of the early users of it who are asking to have them restored. The currently missing features include a Taskbar context menu, the ability to drag and drop files onto Taskbar applications, the ability to move the Taskbar to the top or sides of the screen, and the ability to ungroup running applications. So it feels as though Microsoft is trying and succeeding in dumbing down the user interface. I guess change is good; right?

But apparently I won't be needing to worry about anything changing anytime soon because they also released a newly updated PC Health Check Tool which, because I thought a lot of our listeners would probably want to have access to it and run it, you can get it at aka.ms/getpchealthcheckapp. Or you can use my shortcut which will always be there, grc.sc/check, grc.sc slash C-H-E-C-K. And that just redirects to the same URL.

I was somewhat excited to run it on Windows 7. And it told me, oop, sorry, only runs on Windows 10. So you can't use it to see if your Windows 7 machine will be able to run Windows 11. You have to go to 10 first. So I waited until I got to my other location yesterday evening. And to my surprise, my super-snappy Intel NUC containing a quad core Intel i7-6770HQ running at 2.6 GHz, plenty of memory, plenty of everything else, fails to make the grade. In the show notes I have what the PC Health Test showed. And it said: "This processor isn't currently supported for Windows 11."

Okay. So anyway, I wouldn't probably have wanted to use it anyway because I'm using one of those big curved Dell kind of semi-wraparound screens that's extremely wide. And so I run my Windows Taskbar on the left edge of this very wide panel in order to get all of the vertical space that I can. There's no reason to have the Taskbar taking up space from all of the apps. And you can't do that under Windows 11 because, you know, they made it better. So I'll be sticking with Windows 10. For anyone who wants to see what Microsoft's latest is, however, grc.sc/check.

So it turns out that the failed network printing troubles we covered last week were not the only problems caused by September's Patch Tuesday. Microsoft has stated that users may experience app freezes, app crashes, and the inability to launch an application. So pretty much what an operating system is supposed to do, it might decide not to, after being made better with those updates.

Apparently, the trouble affects users who employ Microsoft's Exploit Protection Export Address Filtering, the EAF, Export Address Filtering feature. It's used to detect dangerous operations used by malicious code or exploit modules. Which is sort of generic-sounding. Microsoft said that: "After installing KB5005101 or a later update on devices using Microsoft Exploit Protection Export Address Filtering (EAF), you might have issues with some applications. You might be experiencing this issue if apps fail to open, fail to open files, or you might receive a white window when attempting to log in."

Microsoft also said: "This issue is resolved using the Known Issue Rollback." That's something new, KIR, the Known Issue Rollback. They said: "Please note that it might take up to 24 hours for the resolution to propagate automatically to consumer devices and non-managed business devices. Restarting your Windows device might help the resolution apply to your device faster." You know, the terms "might" and "maybe" are what you get once all of the actual science has been removed from the computer science. Yes...

Leo: It's just computer now.

Steve: We don't know. Yeah, you might, you know, you might try this.

Leo: Give it a shot. Give it a shot.

Steve: You might try that. Maybe it'll work. Yeah. You know? But hopefully, I was thinking, by now, two weeks after this month's exciting Patch Tuesday, all of the "mights" and "maybes" will have had the chance to work themselves out, and things will be working again for everyone, just in time for next month's Patch Tuesday adventure. Stay tuned. Maybe.

Okay. So okay. This is under the category of "This is just too clever to believe." Google's increasingly prolific TAG team has spotted and reported a diabolically clever new scheme being used by malware to avoid detection by third-party antimalware tools running on Windows. The attackers figured out a way to create a malformed code-signing certificate which would be seen and treated as valid by Windows, thus allowing the code to run without any trouble and with reduced scrutiny because, after all, it has a code-signing certificate, while at the same time being undecodable and thus unchecked by third-party antimalware systems which almost universally use the OpenSSL codebase to perform their various crypto operations. That's just too clever.

So Windows likes it, but third-party antimalware, which uses OpenSSL, hits a tiny little glitch in the parsing of the certificate that causes it to go, oh, well, this is invalid, and to stop further checking. This new technique was observed being exploited by a notorious family of unwanted software known as OpenSUpdater. It's used to download and install other suspicious programs on compromised systems. Most targets of this campaign are users located in the U.S. who are prone to downloading cracked versions of games and other sort of grey-area software.

And while adversaries have previously relied upon illegally obtained digital certs, and we've talked about this for years, it's uncommon, but it happens, to sneak adware and other unwanted software past malware detection tools - because again, as I said, if you have a valid cert, it kind of gives a little bit of a green light - or they've also embedded the attack code into digitally signed, trusted software components by poisoning the software supply chain, OpenSUpdater stands out for its intentional use of malformed signatures to slip through defenses. And again, I don't often give bad guys a tip of the hat, but that's just pretty clever. Leo?

Leo: Steve?

Steve: I'm not going to give Apple a tip of the hat.

Leo: Oh.

Steve: I titled this little piece "A Shaky Foundation."

Leo: I made Lisa watch both episodes because I said, "Steve's going to talk about it on Tuesday. I have to be ready for this."

Steve: Okay. So before I describe that I think so far about Apple's "Foundation" series - although I've already sort of tipped off our listeners, yeah.

Leo: Shaky, that's the word, yeah.

Steve: I wanted to share two paragraphs from Mashable's context-setting posting. And there will be no spoilers anywhere.

Leo: No, no.

Steve: In any of this discussion. We don't do that.

Leo: We don't do that.

Steve: But Mashable did prepare me, because I read this beforehand, they said: "An adaptation of Isaac Asimov's classic science fiction novels, 'Foundation' is less interested in following its source material to the letter than it is in creating a story within Asimov's universe that would make good TV." And unfortunately I'm not sure I agree with even that. They said: "The basic plot remains the same: Mathematician Hari Seldon, played by Jared Harris, foretells the fall of the Galactic Empire, thanks to his theory of psychohistory. Knowing the fall is inevitable," you know, because he's a mathematician and he computes it, "he establishes the Foundation in order to preserve knowledge and, hopefully, civilization in the years to come."

Kind of a cool concept; right? It's like, oh, you're convinced this thousands of years sophisticated Galactic Empire is going to, like, fall into Dark Ages. You want to preserve the knowledge. So you create this Foundation for that purpose, to shorten the fall. They continue.

"'Foundation' takes this story and tweaks it in some pretty big ways, which makes sense when considering the scale of Asimov's work. The Foundation books are collections of interlocking stories and novellas whose events span hundreds of years, not to mention an entire galaxy. Characters who appear in one story may be long dead in the next, and so much happens in between stories that we never fully 'see' on the page." So they finish. "These elements make creating a completely faithful TV show rather challenging, which explains several of showrunner David S. Goyer's choices to deviate from the books."

Okay. So my own take is that, so far, it has definitely not been amazing. So naturally it feels to me like an expensive lost opportunity. "Foundation" has often been called the story that's impossible to bring to the screen, and so far I think we're seeing this play out. There are several problems that I've seen. I'll point out a few. For one, I think the series is having a problem with pacing. It seems to swing between moving quickly and moving slowly.

And Leo, we discussed this before, and I was surprised but pleased to hear that apparently everybody is saying the same thing. One of my biggest complaints is that the dialogue soundtrack is often muddy and unintelligible. The very first scene of the series has four young friends playing outdoors, bundled up against the cold. They're talking meaningfully, like it's important. Yet what they're saying is unintelligible. I paused, backed up, turned up the volume, and still all I heard was "blub blub blub blub." And it seems so inconsiderate of the audience, and I wonder how difficult it could be in this day and age not to have everyone appropriately mic'd up and articulating their lines clearly. You know, and we could just agree that that's something everyone does in the far future, is like articulation has become a thing.

And unfortunately this continued intermittently throughout the two hours, with major characters mumbling to each other, where we're clearly supposed to be listening and obtaining information. And I get it that there's a sense of it being more real and realistic if someone turns to someone standing next to them and mumbles to them. But if we're not supposed to hear what's said, then just give the other person a meaningful look and not leave us pissed off that we don't - because we really, we're trying to care about this; right?

And the other thing, it's really interesting, and you experienced the same thing, you said, Leo, Lorrie and I were unable, I mean, much of a sci-fi fan as I am, I mean, we never stopped watching "Stranger Things." It was like, you know, 3:00 a.m. and Lorrie said, looked at me, "Can we do one more?" Okay, we were unable to watch both episodes of "Foundation" back to back because there's something heavy and oppressive about it. It's not the story. It's the feel. We felt somewhat drained and exhausted after each hour. And maybe it was just from straining to hear what was being said. But whatever it was, an hour of this was a lot.

And speaking of straining, what the heck happened at the end of the second episode? I mean, WTF? I had to go online the next day...

Leo: You're supposed to say that, and tune in to Episode 3.

Steve: Oh, my god. I had to, like, read speculation. And thank god it wasn't just me.

Leo: Oh, just hang with it.

Steve: Nobody had any idea.

Leo: No, it's called a cliffhanger.

Steve: It's called bad. Oh.

Leo: Okay. It didn't bother me. I knew there was something up. You know. Well, we'll find out next week. It's a little frustrating because they only did two episodes in the first week. Now we have to watch week to week, which I've never been a fan of.

Steve: Yeah. And if they wanted it to be a cliffhanger, they should have hung us off that cliff about five minutes sooner. I can't say what happened. I won't spoil anything for anybody. But...

Leo: It's a shocker. It's a big, big, big twist. And I have some speculation. I didn't read any of the articles about it. But I have some thoughts about it.

Steve: I read the articles. I think they're right. And it puts into context an argument that was had a little earlier over a meal. And it's like, okay.

Leo: That's right. That's what I thought. Okay, good. That confirms my - I said to Lisa, I said, they were setting us up a little earlier, yeah.

Steve: Yup. Yup. So anyway, sorry that we don't have another "Expanse," and we don't have another "Stranger Things," something that's like really fun.

Leo: And I'm going to tell you my opinion, which is I couldn't finish "The Expanse." I keep trying. I'm actually really enjoying "Foundation." So people should give it a try.

Steve: Good, good, good, good. Yeah, don't be - I would maybe wait a little longer before subscribing to Apple TV Plus if you don't...

Leo: That's the problem, you've got to pay to see it.

Steve: Yes, yes. And based on the previews, I found myself thinking, you know, next month maybe just a straight-up guns-blazing alien invasion series is going to be fun because that's coming...

Leo: You like, let's be fair, because your taste in science fiction tends to this as well, you like combat. You like space combat. And this didn't have a lot of...

Steve: But I'm a big Star Trek fan. And it was all about social situations and people. And we had our phasers.

Leo: The thing that was very interesting to me, and I mentioned this to Lisa, is to remember this predates pretty much all of the Star Trek-style stuff. This is from the '50s. And Star Trek was very influenced by it. You know, there's - she said, well, that's straight out of Star Trek. I said, no, no, Star Trek is straight out of that.

Steve: Yes.

Leo: And so that's a good reason. You know, I hadn't read it in so long that I wasn't too worried about it not following the actual novel.

Steve: Oh, and I don't care at all.

Leo: I understand the problem of making it, you know.

Steve: It doesn't have to - yes, yes. It doesn't, in no way does it have to follow the plot. You know, "Arrival" was like, just curled our toes.

Leo: Right. Great, a great movie.

Steve: Loved, loved, loved that.

Leo: I agree with you on the sound. Scott Wilkinson also brought that up, that the sound was a little muddy. I think these 5.1 Dolby, and Apple especially because they're really all in on spatial, these things really are starting to require Dolby Atmos systems. And you know what I do, because I do have a 5.1 surround system, if you have that capability, to boost the center channel. That in the past has solved a lot of these kind of muddy dialogue problems for me. I didn't have too much trouble. The other thing us old folks often do, we just turn on the closed captioning and read along.

Steve: Hey, does that work?

Leo: Oh, hey, yeah. Oh, are you kidding? In fact, here's a really nice tip on the Apple TV.

Steve: I think you've just saved it for me, Leo.

Leo: Yeah. If you press the Siri button, and you say "What did she just say?" I'm not kidding, it's the best feature of the Apple TV. It skips back 30 seconds, turns on captions, plays the same thing over again, and then goes back to normal. Try that. "What did those kids just say? What are they talking about?" It's a great feature.

Steve: That's kind of cool.

Leo: It's a great feature. I love it. I do it.

Steve: Get off my lawn.

Leo: It is, I do it all the time. I do it all the time. I was, you know, we watched the first episode on our 4K HDR screen, and it's beautiful, with stereo speakers. And that was easy.

Steve: Oh, Leo. Visually...

Leo: It's visually gorgeous.

Steve: Yes, visually. And for what it's worth we are going to apparently be getting some big space battles, which is okay, you know.

Leo: I'm wondering, if it's CGI, I'm wondering how they did it, if they're doing a Mandalorian-style LED screen, or how much of it's practical effect. I'm very curious as to - I want to see when they release "The Making of Foundation."

Steve: The sets are beautiful.

Leo: It really is beautiful, yeah, yeah. And very powerful, you know, that's - well, I don't want to go into it because we don't want any spoilers.

Steve: No spoilers. In order not to break this last topic, let's take our last break now.

Leo: Good idea. Good idea. Now's a good time.

Steve: And then we're going to talk about, oh, boy. For those who are not familiar with the term "fiasco," it's a particular favorite of mine. It's defined as a thing that is a complete failure, especially in a ludicrous or humiliating way. Thus...

Leo: It's a good Italian word.

Steve: Auto fiasco. And you know, Leo, when I was looking up the definition, I found that there's an Italian restaurant named Fiasco. And I thought, you know, I'm not sure that I'd want to eat at Fiasco.

Leo: The only extra thing I'd add to that is there's a - I don't know if you're familiar with the show and podcast "This American Life," which is a wonderful one. Probably, if you've never listened to any of "This American Life," find the episode called "Fiasco" and listen to it because it's a series, it's stories about a series of different kinds of fiascos. And it is among the funniest things I've ever heard. It is really, really good. If you want a good definition of "fiasco." On we go with the show and the Fiasco of the Week. I think we should have a whole feature: The Fiasco of the Week. There's plenty of them.

Steve: Just love that word. I love the way it feels.

Leo: Fiasco. Italians know how to do it. They really do.

Steve: Okay. So a recurring problem in security occurs when attempts are made to make complex and secure things less complex. It's often the case that they also become less secure. A classic example was the creation of Universal Plug and Play, which defined, as we know, a means for essentially bypassing the crucially important firewall protection being created by NAT routers. Why? Because those pesky NAT routers were getting in the way, exactly as they were designed to and as any savvy user wanted them to. So Microsoft and Intel defined an entirely unauthenticated protocol by which any device on the LAN could map external traffic through to bypass NAT protections.

That was bad enough, since attackers figured out how to get our fancy web browsers inside the LAN to send UPnP requests to the LAN's NAT router on behalf of the attackers, thus allowing the attackers right into the network. Couple that with the fact that mistakes are invariably made, and we saw many routers which mistakenly also bound their UPnP service to the WAN interface. I immediately, as our listeners will recall, I think it was Episode 300 and something, added a test for that to ShieldsUP!. That was 55,145 positive tests ago. And that number should be zero. But no.

Now, UPnP is not today's topic, but it serves as a prime example of the absolute sacrifice of security in the name of convenience. And that is the moral of today's topic. You take a bad and fundamentally flawed idea, mix in the inevitability of human error and even, if you can believe it, hubris, and what you get is Microsoft today frantically contacting domain registrars across the world to preemptively register hundreds, if not thousands, of domains in every TLD before the bad guys can get them. What? What is this about?

Okay. The original idea here was to make it much easier for users to configure their email clients without any need to bother with all those pesky email setup ideas, you know, all those details, you know, things like the server's full domain name, which port to connect to, and what sorts of authentication is supported and so on. The idea is that the user would only need to provide what they know, their email address and their password. Then the email client would use that email domain as a starting point and emit a series of HTTP, and later HTTPS, queries, searching for a server that would answer that query and thus provide the needed configuration information.

The trouble was that domain names themselves are not well formed. Sometimes the user's email might be @example.com, and sometimes @mail.example.com, and perhaps @mail.example.co.uk. And in what Microsoft refers to as "configuration resiliency" - you really can't make this up - they encourage clients to try everything, to throw everything at the wall to see what might stick.

So the crux of the problem is that one of the many things thrown at the wall is to emit queries to a subdomain of the email domain called "Autodiscovery." So an HTTP query, or HTTPS, although believe it or not when S doesn't answer, the fallback is to HTTP, an HTTP query is sent to Autodiscovery.mail.example.com. But again, because we're dealing with an ad hoc and weakly defined specification where Microsoft's stated goal is maximum resiliency - and that comes right off of their spec page - Microsoft said that it's not necessary to have the Autodiscovery subdomain be a subdomain of the user's email, since that might not be convenient. So instead of Autodiscovery.mail.example.com, for example, the Autodiscovery service might also be at just Autodiscovery.example.com." Sure. So let's try that, too.

And you know there are all those tricky domains like example.co.uk where no one is really sure where the enterprise's domain name stops and the public domains begin. And, after all, we want to be resilient. So we wouldn't want to miss an available Autodiscovery server because we didn't try hard enough or look everywhere. Therefore, since we're not sure how far back up the domain hierarchy we should go, and I'm not making this up, we'd better keep going until we either find an Autodiscovery server or we run out of domain name.

And believe it or not, that's what Outlook does. Whereupon we all say in chorus, what could possibly go wrong?

Leo: What could possibly go wrong?

Steve: How about, believe it or not, querying the domain Autodiscovery.com with an HTTP or HTTPS query, and here it is, containing the user's email address and their unencrypted email password, where those credentials are the same as they use to authenticate to their company's entire network domain? It's unbelievable, but it's true.

Guardicore's Amit Serper [A-M-I-T S-E-R-P-E-R] published the result of their research titled "Autodiscovering the Great Leak." Now that we have this bit of background, Amit's Executive Summary will make sense. For his Executive Summary he wrote four bullet points: "Autodiscover, a protocol used by Microsoft Exchange for automatic configuration of clients such as Microsoft Outlook, has a design flaw that causes the protocol to 'leak'" - and he's got "leak" in quotations because it's not, you know, it's like screaming it - "web requests to Autodiscover domains outside of the user's domain, but in the same TLD, for example, Autodiscover.com," believe it or not.

"Guardicore Labs acquired multiple Autodiscover domains with a TLD suffix and set them up to reach a web server that we control." In other words, they got Autodiscover.il, Autodiscover.is, Autodiscover.sc, a bunch of them. "Soon thereafter, we detected a massive leak of Windows domain credentials that reached our server. Between April 16th of this year, 2021, to August 25th, 2021 we have captured 372,072 Windows domain credentials in total; 96,671 unique credentials that leaked from various applications such as Microsoft Outlook, mobile email clients, and other applications interfacing with Microsoft Exchange server."

Third bullet point: "This is a severe security issue, since if an attacker can control such domains or has the ability to sniff traffic in the same network, they can capture domain credentials in plaintext, HTTP basic authentication, that are being transferred over the wire. Moreover, if the attacker has DNS-poisoning capabilities on a large scale, such as a nation-state attacker, they could systematically siphon out leaking passwords through a large-scale DNS poisoning campaign based on these Autodiscover TLDs."

Finally: "Additionally, we have developed an attack [they call] 'the ol' switcheroo'" - that's literally what it's called - "which downgrades a client's authentication scheme from a secure one, OAuth or NTLM, to HTTP Basic Authentication, where credentials are sent in clear text."

Okay, now, just as an aside so I don't forget to mention it later, the way this credential security downgrade attack works is to simply have the intercepting Autodiscover.whatever web server reply to the first query over OAuth or NTLM with an "I don't support that fancy protocol" reply, that's an HTTP 401 response from the web server, whereupon the client will reissue under HTTP basic authentication. So not that tough.

So I'll just share the first three paragraphs of this much longer and more detailed report to reiterate the nature of this danger. Amit writes: "As a part of the ongoing security research efforts by the Guardicore Labs team, we have discovered an interesting case of credential leak affecting a large number of people and organizations worldwide. The credentials that are being leaked are valid Windows domain credentials used to authenticate to Microsoft Exchange servers. The source of the leaks is comprised of two issues: First, the design of Microsoft's Autodiscover protocol and its back-off algorithm, specifically; second, poor implementation of this protocol in some applications."

They said: "As mentioned, Microsoft's Autodiscover protocol was meant to ease the configuration of Exchange clients such as Microsoft Outlook. The protocol's goal is to make an end-user able to completely configure their Outlook client solely by providing their username and password, and leave the rest of the configuration to Microsoft Exchange's Autodiscover protocol. It is important to understand," Amit wrote, "that since Microsoft Exchange is a part of the Microsoft domain suite of solutions, the credentials that are necessary to log into one's Exchange-based inbox are in most cases also their domain credentials. The implications of a domain credential leak in such scale are massive and can put organizations in peril. Especially in today's ransomware-attacks-ravaged world, the easiest way for an attacker to gain entry into an organization is to use legitimate and valid credentials," which this is leaking on massive scale.

Okay. Then skipping way down to some interesting and important details, Amit explains: "The protocol" - and this is also directly from Microsoft's page, which I also read. "The protocol has several iterations, versions, and modes. Their full documentation can be found on Microsoft's website. However, in this article we'll discuss a specific implementation of Autodiscover based on the [happily named] POX XML protocol." That's P-O-X.

"In order to truly understand how Autodiscover works, we need to know what happens behind the scenes. First, the client parses the email address supplied by the user." And he gives this example, amit@example.com. "The client then tries to build Autodiscover

URLs based on the email address with the following format." Okay, the first one: https://Autodiscover.example.com/Autodiscover/Autodiscover.xml. If nothing answers there, HTTP, meaning just drop SSL/TLS completely. Let's just try it bare. Okay. Nothing answers there. So then we go to https://example.com, that is, forget the Autodiscover, go to example.com/Autodiscover/Autodiscover.xml over HTTPS. And if that doesn't answer, drop the S: http://.

He says: "In the case that none of these URLs are responding, Autodiscover will start its back-off procedure. This back-off mechanism is the culprit of this leak because it is always trying to resolve the Autodiscover portion of the domain, and it will always try to 'fail up,' so to speak," he writes, meaning the result of the next attempt to build an Autodiscover URL would be http://Autodiscover.com/Autodiscover/Autodiscover.xml. This means that whoever owns Autodiscover.com - Autodiscover.com, literally, really - will receive all of the requests that cannot reach the original domain. He says: "For more information about how Autodiscover works, please check out Microsoft's documentation."

And then comes the proof of concept. Amit writes: "In order to see if the Autodiscover leak scenario is even a viable one, we have purchased the following domains." And there's a list of them: Autodiscover.com.br, for Brazil. Autodiscover.com.cn for China. Autodiscover.com.co for Colombia We have .es for Spain, .fr for France, .in for India, .it for Italy, .sg, .uk, .xyz, .online, .cc, .studio, .capital, .club, .company, .jp, .me, .mx, and .ventures. Autodiscover in all of those.

He says: "Later, these domains were assigned to a web server in our control, and we were simply waiting for web requests for various Autodiscover endpoints to arrive. To our surprise, we started seeing significant amounts of requests to Autodiscover endpoints from various domains, IP addresses, and clients. The most notable thing about these requests was that they requested the relative path of /Autodiscover/Autodiscover.xml with the Authorization header already populated with credentials in HTTP basic authentication," meaning cleartext.

He says: "Now, as you might imagine, Microsoft" - oh, no, I'm sorry, he's not. I'm saying. That's the end of my quoting him. As you might imagine, Microsoft is a little bent out of shape by this surprise revelation. After Guardicore released their report, Microsoft issued the following huffy statement. This is Microsoft: "We are actively investigating and will take appropriate steps to protect customers. We are committed to coordinated vulnerability disclosure, an industry standard collaborative approach that reduces unnecessary risk for customers before issues are made public. Unfortunately, this issue was not reported to us before the researcher's marketing team presented it to the media, so we learned of the claims today." Signed Jeff Jones, Senior Director, Microsoft. And as you know, well, you don't know.

Leo: But you're going to tell us, I hope.

Steve: But I'm going to tell you what you're going to know. That would be okay, I mean, we could understand that, if the presentation given on Friday, March 31st, 2017, yes, more than four years ago, 4.5 years ago at 3:30 p.m., during Black Hat Asia 2017, had not been "All Your Emails Belong to Us: Exploiting Vulnerable Email Clients via Domain Name Collision."

Leo: Oh, my goodness.

Steve: Uh-huh. They knew about it.

Leo: Oh, my.

Steve: I have the link to the full PDF. The abstract of that paper explains: "The Autodiscover HTTP Service Protocol provides a way for Autodiscover clients to find Autodiscover servers. This protocol extends the Domain Name System and directory services to make the location and settings of email servers available to clients. In this paper, we take a closer look at the Autodiscover protocol and identify its threat model. We analyze Autodiscover client implementations in two mobile built-in email clients to discover flaws which allow remote attackers to collect user credentials through domain name collision. We discover how many clients have vulnerable implementations by collecting and analyzing HTTP request information received by our servers, registered with specially crafted domain names." Sound familiar? Uh-huh.

"We make our analysis based on data we collect from 25 different domains. Our dataset contains information on about 11,720,559 requests" - that's approximate - "and we observe 9,726,026 requests containing authentication information. We identify 2,173 different email clients which use vulnerable Autodiscover client implementations. Finally, we propose different mitigation techniques for users, enterprises, and application developers to improve their email clients."

In other words, this recent work by Guardicore was a reminder to Microsoft of a directly related issue that was fully documented and disclosed 4.5 years ago which has never been fixed. Apparently, because it directly wasn't aimed at you, Microsoft, and you were not given specific instructions about how to fix it, you just ignored the whole issue until now. Your entire Autodiscover concept has always been an insecure bad idea, and nothing has changed during the intervening 4.5 years to make it any better or to resolve its fundamentally broken design.

Now, I realize, Microsoft, that you've got important work to do. The world has been clamoring for Windows 11, after all. So what is Microsoft doing? As I mentioned at the top, Microsoft is now frantically registering the "Autodiscover" domain in - and Leo, you might want to put this page 12, the bottom of page 12 and top of page 13 on the screen - frantically registering...

Leo: Good, because I don't want you to write it, I mean read it.

Steve: No, in every top level domain they can, just as fast as they can. Friday, BleepingComputer's Lawrence Abrams wrote: "At the time of this writing, BleepingComputer has confirmed that Microsoft registered at least 68 domains related to Autodiscover, which are listed below." I have them in the show notes. If we ever needed, I mean, so it's just like Autodiscover.everything. I wished that there was an Autodiscover.wtf, but apparently we don't have that because that would have been perfect.

Leo: There's wf. That's close enough.

Steve: That's, yeah, that's what gave me the idea. If we ever needed a clear example of a kludge, here it is. This is Microsoft registering the domains which their clients, their email clients are querying while providing the username and password, which is typically their domain credentials, which can be used to log into their domain at their enterprise.

Microsoft registering those domains so that they will receive the query or blackhole it, rather than a hacker beating them to it. And lord knows how many hackers have.

And understand, this really doesn't fix the problem because many of these domains are in foreign countries' controls. DNS servers are in foreign countries' controls. It hasn't been very useful for a foreign adversary to compromise a DNS server because the browser would check the web certificates. Clients don't do that. If a client is trying to autodiscover and nothing answers at https, it tries http. Which means all a foreign adversary needs to do is intercept DNS and change the resolution of the autodiscover.* to a different IP, and that individual or entity will begin collecting domain credentials for all of the users who are autoconfiguring.

Lawrence continued: "BleepingComputer also knows of 38 other domains registered since September 22nd whose owners are hidden behind privacy or WHOIS restrictions, that were also likely registered by Microsoft, researchers, or potentially threat actors." Who knows? "The actual number of registered domains is likely far larger," he writes, "as BleepingComputer has seen Microsoft register multiple Autodiscover domains for the same TLD, such as Autodiscover.com.es and Autodiscover.org.es." In other words, I mean, this is a catastrophe. You have to do Autodiscover dot whatever might be underneath it, or you'll miss one that will be leaking valid credentials. One domain, Autodiscover.ch, has been registered since at least 2015, which is interesting, and uses Microsoftonline.com as its DNS servers, but it's not clear who owns it.

And Lawrence finishes: "While registering Autodiscover.tld domains will block some of the leaks, Microsoft will need to issue fixes for the poor Autodiscover implementation in their Microsoft Outlook and Office 365 mail clients to resolve the issue further. As other non-Microsoft applications also have faulty protocol implementations, Microsoft will also have to release guidance on how to properly create Autodiscover URLs so that credentials are not sent to untrustworthy domains."

In their coverage of this, Threatpost reached out to Alicia Townsend, who's the technology evangelist for OneLogin. Alicia told Threatpost that it seems "incredible" that a product would send a user's username and password to an untrusted endpoint. "The fact that this is happening with an incredibly popular Microsoft product such as Exchange is even more disheartening." She pointed out that it's not clear how long this design flaw has been around - its like design has been around - given that the Exchange Autodiscover feature was introduced in Exchange 2007. But regardless, it doesn't shine a good light on Microsoft. "Whether the oversight was on the part of early developers or was introduced by more recent developers, it is clear that Security First was not their primary objective."

Right. You can imagine that some large enterprise customer came to Microsoft shortly after 2007 and the introduction of, oh, look, your clients will now configure themselves, and complained that Exchange's Autodiscovery was not working for them due to their domain's naming structure. So some genius over at Microsoft thought, "Ah, no problem. Let's just have Outlook try everything until it finds something that works." What could possibly go wrong?

And, you know, what if Amit had instead quietly and privately reminded Microsoft of something crucially important about the fundamental design of one of their protocols that they presumably already knew about? Would they have taken action? And if so, when? Late last year and earlier this year they waited until it was too late, until their Exchange Server was being violently attacked, to begin fixing it, and then that fixing went on through the spring. And as we noted last week, we're now at nine months since Microsoft was informed about the more serious of the many PrintNightmare problems, which has still not yet been resolved today.

Microsoft's official response to Amit's public disclosure was to say: "We are committed to coordinated vulnerability disclosure, an industry standard collaborative approach that reduces unnecessary risk for customers before issues are made public." Except of course this was made public 4.5 years ago. But at what point does this become a means of stifling researchers' voices, shaming them into keeping quiet while the software publisher spends their time rounding off the pointy corners of windows and changing the Start Menu's alignment?

I have some personal experience with that myself. As many of us recall, I tried mightily to explain to Microsoft that repackaging their NT-derived Windows 2000 operating system as WinXP, while leaving its unneeded raw socket API in place in a consumer OS, would be a real mistake. It wasn't until their own WinXP operating system used its unneeded raw socket API to blast them with a devastating DDoS attack - the so-called MSBLAST worm - that they finally understood what I had been trying to tell them all along and then limited that API's ability to do harm in a subsequent XP service pack.

Given the evidence, we're seeing that Microsoft has become a company that only responds to force. They must be forced to fix the things that are broken. "Responsible disclosure" is just a courtesy, after all. It's one that the industry might consider withdrawing if publishers do not honor their side of the implicit agreement to fix what's been responsibly disclosed.

Leo: Right on.

Steve: This is just a catastrophe.

Leo: Yup. And by the way, somebody did register Autodiscover.wtf on September 22nd. So I'm thinking it was probably Microsoft. But we don't know because it's, you know, they hide the name of the contact. It's just somebody in California.

Steve: No kidding. Interesting. I wonder what it resolves to.

Leo: Yeah, well, hmm. Yeah, it goes through Cloudflare. So that's their name server. Want to try, just out of curiosity? It's probably Microsoft. It's all happened in the last week, so, you know.

Steve: Unbelievable.

Leo: Mr. Gibson is at GRC.com. That's the place to go, the Gibson Research Corporation, where you will get, not only the world's best mass storage recovery and maintenance utility, SpinRite, currently 6.0, but 6.1's on the way, and you can help in the development of it. If you get your copy now, you'll get a free upgrade. He also has this show, 16Kb audio, 64Kb audio, fully human-written transcriptions thanks to Elaine Farris. That's all at GRC.com. Leave Steve feedback there at GRC.com/feedback, or on his Twitter, @SGgrc. His DMs are open, so you might want to stop by there.

Of course we have copies of the show at our website, TWiT.tv/sn. There's a YouTube channel with a video of every show. And of course any podcast app should be able to find this in our 17th year. If it's not in your podcast directory by now, I guess you just don't like us. Go ahead and get any podcast app. Subscribe. Do me a favor. If they allow reviews, leave Steve a five-star review, and let the world know about your love for Security Now!. I think the world needs to hear a little bit more Security Now!; don't you?

We do the show, if you want to hear us live, do it live, we do the show every Tuesday, about 1:30, right after MacBreak Weekly, so 1:30 to 2:00 Pacific. That'd be 4:30 Eastern, 20:30 UTC. The live streams, audio and video, are at TWiT.tv/live. People who listen live or watch live often like to chat live with our IRC server, irc.twit.tv. That's there all the time, 24/7, as is our Discord for Club TWiT members. And you're invited to hang out there with them if you're watching live. Steve, wonderful, as always. I will see you next week. Are you going to watch Episode 3 of "Foundation" this week?

Steve: Oh, yeah. Oh, yeah.

Leo: Give it a shot?

Steve: I'm, you know, I'm...

Leo: Got to know what happens. Got to know what happens.

Steve: I just - now my expectations have been set appropriately.

Leo: Yeah. It's just pretty to watch.

Steve: It is very pretty. And knowing that I can have subtitles, yay. I mean, I'm afraid - I hate subtitles, but that's the solution.

Leo: You know, I get used to it because a lot of things are hard to understand these days, and I just turn on subtitles. I ignore them until I need them, and then I look down and say, what did he say? Or just press that button, say what the hell did he just say?

Steve: Ah, love that.

Leo: That is a nice feature.

Steve: Okay, buddy.

Leo: I think that's the single best feature of the Apple TV. Thanks, Steve. Have a great week.

Steve: Bye.