SATELLITE INSECURITY, PART 1

Description: What did Kaspersky have to say about last Tuesday's Microsoft patch event, and what security consequences does it have for all non-subscribing Microsoft Office users? What was inevitably going to happen once the power of Large Language Model generative AI became widely appreciated and available? What does it mean that Microsoft just revoked more than 100 malicious Windows drivers? What two new well-known companies have been added to Clop's MOVEit file transfer victim list? What does Dun & Bradstreet have to do with Android Apps? Where in the world can you use Meta's new Threads service, and where not? And what's a side effect of bitcoin addresses looking like gibberish? And after we examine those questions, cover some miscellany and user feedback, we're going to turn our attention to the heavens in recollection of those famous words of Henny Penny.

High quality (64 kbps) mp3 audio file URL: http://media.GRC.com/sn/SN-931.mp3
Quarter size (16 kbps) mp3 audio file URL: http://media.GRC.com/sn/sn-931-lq.mp3

SHOW TEASE: It's time for Security Now!. Steve Gibson is here with a stunning flaw in Microsoft. They say it's a feature, not a bug, and it's been around since 2011. You'll be interested in what Kaspersky has to say about all that. Microsoft also just revoked 100 malicious Window drivers. Wow. Why can't you use Threads in Europe? And then a look, Part 1 of our look at satellite security, or should we say insecurity? It's next with Steve Gibson on Security Now!.

Leo Laporte: This is Security Now! with Steve Gibson, Episode 931, recorded Tuesday, July 18th, 2023: Satellite Insecurity, Part 1.

It's time for Security Now!. I know you've been waiting all week long. Finally, Tuesday's here. Security Now!'s on the air. And there he is, Steve Gibson, the star of our show. Hi, Steve.

Steve Gibson: Yo, Leo. Great to be with you again.

Leo: You know your Wikipedia calls you Steve "Tiberius" Gibson.

Steve: I think it knows that's not the case.

Leo: Hope so. Every once in a while I like to call you that, since you are a Star Trek fan.

Steve: Indeed.

Leo: What's up?

Steve: So an interesting bit of news about a paper that was submitted and accepted and presented at a recent IEEE security symposium about two months ago in May caught my eye because it's something in our 18-plus years of the podcast we've never talked about. We've always been talking about it on the security, you know, fiber optic cables and stuff on the ground. We've never looked up. And today's topic, it started off just to be today. It ended up being today and next week. So this is Satellite Insecurity Part 1 for today.

Leo: Hmm, interesting.

Steve: Really interesting. And not surprisingly, things are not good up there in the friendly skies. But anyway, we've got a lot to talk about. We're going to look at what Kaspersky had to say about last Tuesday's Microsoft patch event, and what security consequences it has for all nonsubscribing Microsoft Office users. Also, what was inevitably going to happen once the power of Large Language Model generative AI became widely appreciated and available? What does it mean that Microsoft just revoked more than 100 malicious Windows drivers? 100. More than. What two well-known companies have been added to Clop's MOVEit file transfer victim list?

Leo: Uh-oh.

Steve: What does Dun & Bradstreet have to do with Android Apps? Where in the world can you use Meta's new Threads service, and where not? And what's a side effect of bitcoin addresses looking like gibberish? And after we examine those questions, cover some miscellany and user feedback, we're going to turn our attention, as I said, to the heavens, in recollection of those famous words of Henny Penny.

Leo: The sky is falling?

Steve: Ah, yes.

Leo: Good old Henny Penny.

Steve: Henny Penny.

Leo: She wasn't wrong. She wasn't wrong.

Steve: No. And we do have a great Picture of the Week which has already generated some laughter among the Twitter people who saw it earlier.

Leo: I have not looked. My new policy is to only...

Steve: That's good.

Leo: Yeah, I increase the size of the show notes to such a point that the picture's below the fold.

Steve: Nice.

Leo: So I will scroll it up and share it with you. I shall scroll up to see the Picture of the Week. I don't know what it means, but I like it. Will you explain this to me, Steve?

Steve: So the caption I gave this is "Insecure Parking Spaces - Lock Your Car." And what we're looking at is something that one of our listeners, bless their hearts, you know, being a listener to the podcast, saw and thought, oh, my god, I've got to take a picture of this and send it to Steve.

Leo: Wow.

Steve: So you know how parking slots are often labeled with, like, who's allowed to park in there, like Denny's Parking Only kind of thing, if there's some movie theater next to it, and Denny's is upset because people who are going to the movies are parking in their slots.

Leo: I hate it when that happens, yeah.

Steve: So these...

Leo: Worse, I'm at Denny's all the time, and I want to park and have my fried egg and get on out of there. So yes. Yes.

Steve: That's right. You don't want those...

Leo: Yes. Those movie people. Get them out of there.

Steve: ...movie people taking up your spot. That's right. So here we have stenciled on parking spaces who is allowed to park in them. And I kid you not, I don't know what the initials stand for, but it says "HTTP PARKING ONLY." So of course...

Leo: So no insecure...

Steve: That's right, those are insecure parking places.

Leo: Is there an HTTPS next to it? That's the question. One I can use.

Steve: You really want to lock your car.

Leo: Yeah, that's hysterical. I love it. So obviously that's a company of some kind.

Steve: Yeah, exactly. It's some company's initials. So that, you know, everybody who would be wanting to park there would realize, oh, wait, that's - I can't park in this spot. That's reserved for HTTP. Which, you know, means something to us.

Leo: Yes.

Steve: Okay. So Kaspersky being Kaspersky, a very technologically savvy security firm, had an interesting take on last Tuesday's monthly Microsoft patch event. It was the heading on their posting that first drew me in. They titled their posting, that is, Kaspersky did, "Band-Aid on a... corpse," which, you know, is not the way you want to start describing Patch Tuesday, right, "Band-Aid on a corpse."

They said: "Microsoft patches IE again," and their subhead was "July Microsoft Patch Tuesday: A collection of exploited vulnerabilities." So this is all definitely worth sharing as we look back at the past week. We often do a retrospective on Patch Tuesday. Today we're going to start with what Kaspersky had to say, and then I'm going to, you know, flesh it out a little bit.

So Kaspersky wrote: "The Microsoft July patch collection has turned out to be a quite surprising event. First, they're once again fixing apparently dead Internet Explorer. Second, as many as six of the vulnerabilities are already being actively exploited by attackers." In other words, six zero-days in last Tuesday's patch batch. They said: "Third, two of those six actively exploited vulnerabilities were closed, not with patches, but with recommendations." And that's what we're going to end up talking about because this is a little distressing.

So they said: "Here are the total statistics: 132 flaws were closed," making it one of the larger ones. And this is, you know, every month this happens. "Nine of which are considered critical. Exploitation of 37 of those vulnerabilities can lead to arbitrary code execution, 33 to privilege elevation, 13 to security feature bypasses, and 22 to possible denial of service."

They said: "Not so long ago we wrote that Internet Explorer had kicked the bucket, but not quite. In particular, we talked about Microsoft's advice to continue installing security updates related to IE, since some of its components are still in the system. And now it becomes clear why they gave this advice. The July patch closes as many as three vulnerabilities in MSHTML, the engine inside the legendary browser. In the CVE descriptions, Microsoft states the following."

So Microsoft said: "While Microsoft has announced retirement of the Internet Explorer 11 application on certain platforms" - and I'm thinking, wait, are there any platforms where it hasn't been retired? I don't think so. And, they said, Microsoft said: "The Microsoft

Edge Legacy application is deprecated." Right, remember Edge started using MSHTML and then switched to Chrome, or the Chromium engine.

So they said: "The Microsoft legacy application is deprecated. The underlying MSHTML, EdgeHTML, and scripting platforms are still all supported. The MSHTML platform is used by Internet Explorer mode in Microsoft Edge, as well as other applications through WebBrowser control. The EdgeHTML platform is used by WebView and some UWP applications. The scripting platforms are used by MSHTML and EdgeHTML, but can also be used by other legacy applications." In other words, they wish it were dead, but it's just too deeply wired into Windows to actually go away. And it's got some problems.

So they said: "Updates to address vulnerabilities in the MSHTML platform and scripting engine are included in the IE Cumulative Updates; EdgeHTML and Chakra changes are not applicable to those platforms. To stay fully protected, we recommend that customers who install Security Only updates install the IE Cumulative updates." And yes, I second that advice.

Okay. So back to Kaspersky, who says: "The most dangerous of the freshly discovered IE vulnerabilities is CVE-2023-32046." And of course all the CVEs I'll be talking about are 2023, so I'm just not going to be saying that every time. Anyway, 32046. They said: "And it's already being used in real attacks. Its successful exploitation allows cybercriminals to elevate their privileges to those of the victim. Attack scenarios involve the creation of a malicious file that's sent to the victim by email or hosted on a compromised website." So, you know, opening a file, bang. "All attackers need then is to convince the user to follow the link and open the file.

"The remaining two vulnerabilities, 35308 and 35336, can be used to bypass security features. The first allows a cybercriminal to create a file bypassing the Mark-of-the-Web mechanism so that the file can be opened by Microsoft Office applications without Protected View mode. And both holes can be used to trick a victim into accessing a URL in a less restrictive Internet Security Zone than was intended.

"The next two vulnerabilities are also being actively exploited; but instead of full-fledged patches, they've only received security recommendations." And this is the, woo, we're going to be spending some time on this because this is a little surprising. They wrote: "The first one, 36884, with CVSS rating 8.3, is being exploited in the Storm-0978/RomCom remote code execution attacks on both Office and Windows. To stay safe, Microsoft advises adding all Office executables to the FEATURE_BLOCK_CROSS_PROTOCOL_FILE_NAVIGATION list." Okay, that's actually the name of a registry key. So Feature Block Cross Protocol File Navigation. And we'll be coming back to that, as I said, and have a lot more to say about that in a minute.

Kaspersky continues: "The second unresolved issue" - and again, what I just talked about is like a problem that Microsoft has chosen not to fix, and we'll explain why. They said: "The second unresolved issue relates to the signing of kernel-level drivers. This one doesn't have a CVE index," Kaspersky says, "but only a guide with recommendations. Microsoft revoked a bunch of developer certificates used in Advanced Persistent Threat attacks and blocked several malicious drivers, but the root of the problem remained. Hackers still manage to sign drivers with Microsoft certificates, or sign them backdated to make them work as one of the exceptions and not require the MS developer portal signature." And this is something we've been talking about, right, how Microsoft really has a problem with driver signing.

They said: "As a countermeasure, Microsoft recommends keeping both Windows and EDR" - that's the endpoint security - "up to date. The only small consolation is that in order to exploit such drivers, the attacker must have admin privileges." On the other

hand, if you've got privilege elevation exploits wandering around, like in IE, that may not be difficult.

"Besides the above-mentioned vulnerabilities there are three more holes that are already being exploited by cybercriminals. We've got 32049, a SmartScreen security feature bypass vulnerability. Its exploitation allows attackers to create a file that opens without displaying the Windows warning 'downloaded from the Internet.'

"We've got 36874, a privilege escalation vulnerability in the Windows Error reporting service. That allows attackers to elevate privileges if they already have normal permissions to create folders and technical performance monitoring files."

Finally, "35311. It's a security feature bypass vulnerability in Outlook. Its exploitation helps cybercriminals avoid showing warnings when using preview." And of course as we know, clever attacks actually do use these things in order to slip past users even when they're trained up and are wary.

Okay. So on balance, we got a bumper crop of 132 total patches this month, nine being critical, 37 allowing for arbitrary code execution, six being actively exploited in the wild as true zero-days. One of those zero-days being actively exploited in the wild right now was that 36884 - that's the one carrying the CVSS, which is pretty high, of 8.3 - being exploited in a phishing campaign being conducted by a group designated as Storm-0978. What's got people stirred up is that, despite this being actively exploited in the wild, and having been identified as a zero-day, Microsoft has not patched it, and they appear unlikely to do so. The reason is that this phishing campaign is using a feature, not a bug. Were it to be disabled for security, Microsoft is afraid that might break too many existing things. And so they're afraid to turn it off.

Now, this is one of those things, and we've encountered them before, which Microsoft should have turned off a long time ago, in which case this would have never been a problem. Or better yet, should have never made possible in the first place. In which cases developers would have found, like legitimate developers, would have found some other safer way to do the same thing. But no. It's like scripting in email. What could possibly go wrong?

Okay, so what does Microsoft have to say about all this? Microsoft's posting of July 11th, right, so that was Patch Tuesday, last week, Microsoft posted about this in a separate posting titled "Storm-0978 attacks reveal financial and espionage motives." So I thought, okay, as I was digging into this, what's going on here?

Microsoft said: "Microsoft has identified a phishing campaign conducted by the threat actor tracked as Storm-0978 targeting defense and government entities in Europe and North America. The campaign involved the abuse of CVE-2023-36884, which included a remote code execution vulnerability exploited before disclosure to Microsoft" - in other words, a zero-day; right? It was "exploited before disclosure to Microsoft via Word documents, using lures related to the Ukrainian World Congress."

"Storm-0978 - also they have DEV-0978, also referred to as RomCom, the name of their backdoor, by other vendors - is a cybercriminal group based out of Russia, known to conduct opportunistic ransomware and extortion-only operations, as well as targeted credential-gathering campaigns likely in support of intelligence operations. Storm-0978 operates, develops, and distributes the RomCom backdoor. The actor also deploys the 'Underground' ransomware, which is closely related to the Industrial Spy ransomware first observed in the wild in May of 2022. The actor's latest campaign detected in June of 2023" - so just last month - "involved abuse of this exploit, 36884, to deliver a backdoor with similarities to RomCom.

"Storm-0978 is known to target organizations with trojanized versions of popular legitimate software, leading to the installation of RomCom. Storm-0978's targeted operations have impacted government and military organizations primarily in Ukraine, as well as organizations in Europe and North America potentially involved in Ukrainian affairs. Identified ransomware attacks have impacted the telecommunications and finance industries, among others."

Okay. So now we get to the good part of this, after that background. Microsoft 365 Defender detects multiple stages of Storm-0978 activity. Customers who use Microsoft Defender for Office 365 are protected from attachments that attempt to exploit 36884. In addition, customers who use Microsoft 365 Apps versions 2302 and later are protected from exploitation of the vulnerability via Office. Organizations who cannot take advantage of these protections can set the FEATURE_BLOCK_CROSS_PROTOCOL_FILE_NAVIGATION registry key to avoid exploitation.

In Microsoft's posting from last week, that Registry key was highlighted and underlined like a link. And sure enough it was a link, so I clicked it. Where did it take me? It jumped me to a page, and the link used the pound sign suffix to preposition me a ways down the page to a specific section which they wanted to refer to. And that prevented me from initially seeing the title of the page. The section of the page I was jumped to was titled "New restrictions on use of the file:// Protocol." So, of course, I thought, whoa.

Leo: Oh, I can't believe this. That's still around?

Steve: I know. I know, Leo. I thought, so, like whoa, that's what we're talking about here? And it's being exploited in a zero-day today?

Leo: Oh, lord.

Steve: We're talking about bad guys leveraging the file:// scheme to arrange to run programs on the user's machine from Office documents. And that thought was followed by, wait. Exactly as you said, Leo. That's still possible? So then I started to read what Microsoft wrote on this page that had been linked to by their posting from last Tuesday. And there they wrote: "Prior to this update, Internet Explorer would allow non-file-protocol (i.e., HTTP and HTTPS) delivered pages to frame" - in other words, using an Iframe - "or navigate to pages that were delivered using the file:// protocol scheme. IE would only block loading of resources from the local computer, for example, file:///C:/temp/test.gif, for example. But resources from non-local paths would be allowed." And then they said: "Here's an example page displayed in IE 9.0.1."

And I thought, IE 9? So, I finally scrolled up to the top of the page to see what in the world I was reading, and it was from Microsoft, posted on August 12th of 2011. Yes, 12 years ago, titled "Internet Explorer 9.0.2 Update."

Leo: Oh my god. Oh my god.

Steve: And sure enough, they showed where IE 9.0.1 - apparently just like Office apps today - will load an Iframe with text content provided by the file:// scheme from, for example, live.sysinternals.com was the example they gave from 12 years ago. And then they show the same thing done under the new and improved IE 9.0.2 and, what do you know, by golly, that Iframe, is empty. Then they note: "Other browsers have blocked

cross-protocol interactions for quite some time. Here are screenshots of Firefox 5, you know, that browser from times past; Chrome 14; and Opera 11.5 developer consoles in this same scenario."

Okay. So just to make sure that everyone is on the same page here, this Russia-located Storm-0978 phishing campaign has been successfully installing trojan code into unsuspecting Office users' machines, using a technique that IE 9.0.2 celebrated ending in August of 2011.

Leo: 2011.

Steve: Noting at the time that everyone else had already done that. Yet, just last week Microsoft wrote, and I'm quoting them again: "The campaign involved the abuse of 36884, which included a remote code execution vulnerability exploited before disclosure to Microsoft via Word documents." Although they had disclosed it to themselves in August of 2011. So they've known about it since IE 9, finally decided to fix it, and even then it was the last of the bunch to do so.

So it came back. They turned it back on for some reason in Office. Who knows when? But now Microsoft is afraid to turn it off again, despite the fact that it's being abused in a trojan-installing remote code execution vulnerability in their own Office documents. They can't turn it off because they have no way to predict what doing so might cause to break. So they're not going to make it their problem. Unless you're using their online subscription Office stuff, in which case they'll protect you from it. But if not, it's up to you. So there's a registry key which will allow anyone and everyone to turn off this behavior which is currently under active abuse, apparently by Russians, to install malware into the computers of unsuspecting link clickers.

I've got a picture from a registry snapshot showing this. The key is HKEY_LOCAL_MACHINE\SOFTWARE\Policies\Microsoft\Internet Explorer\Main\FeatureControl\FEATURE_BLOCK_CROSS_PROTOCOL_NAVIGATION. And under that key it's necessary to enumerate each of the various Microsoft apps whose behavior, in this case, you would like to restore to Internet Explorer 9.0.2 where this was originally fixed 12 years ago. I also have a screenshot of the registry showing the enumeration of the REG_DWORD values under that key. And Leo, thank you, it's on the screen right now.

Leo: So what are you supposed to change it to?

Steve: Well, you have to add all of that. None of that there now.

Leo: Oh, no. Okay. All right.

Steve: Yeah, you've got to put all that in. And, now, this...

Leo: Isn't it possible to make a regedit file that you just double-click, and it does it?

Steve: Absolutely. Regedit file, and it would just be a simple one-click, and it would do that. Now, as I said, this has stirred up a bunch on the 'Net because Microsoft is literally

not fixing something that is in active abuse right now as a remote code execution vulnerability. They can't fix it because it will break things that developers are depending upon.

Leo: Yeah. You should break it.

Steve: Yes, you should break it.

Leo: Frankly, it's a stupid thing to have had in the first place.

Steve: It is incredibly, as I said, like scripting in email, what could possibly go wrong? The problem is all over the 'Net you will now find scripts that are doing this. Unfortunately, they spell "PowerPoint" all the way out, and that's incorrect. It needs to be Powerpnt.exe. So just a heads-up. If you do use these scripts, you'll see that they have PowerPoint.exe. It needs to be Powerpnt.exe, the actual name of the EXE, that component of Office. So anyway, it's crazy that they've done this. Who knows when this came back into use. The bad guys found it and said, oh, that's nice, just like Internet Explorer 9.0.1.

Leo: It's 2011 all over again.

Steve: We can go back to our old...

Leo: Oh, my god.

Steve: Unbelievable. Unbelievable. Okay. So I suppose it was inevitable, though it happened sooner than I would have expected. The underground now has a ChatGPT-style generative AI all their own, without any of the abuse prevention built into the front end that is in ChatGPT. It is known, and I kid you not, as WormGPT and it exists. The news of this comes from a reformed black hat computer hacker named Daniel Kelley who collaborated with the team at the business email and messaging protection security firm SlashNext. Daniel begins his posting by providing a background about the use of legitimate generative AI like ChatGPT and discusses, as we have here, the fact that such AI can be hugely useful to bad guys when they're able to coerce it or seduce it into giving them what they want, meaning ChatGPT, which is trying not to. But now it appears this will no longer be necessary.

Daniel explains in his posting, he said: "We recently gained access to a tool known as 'WormGPT' through a prominent online forum that's often associated with cybercrime. This tool presents itself as a black hat alternative to GPT models, designed specifically for malicious activities. WormGPT is an AI based on the GPT-J language model, which was developed in 2021. It boasts a range of features, including unlimited character support, chat memory retention, and code formatting capabilities. WormGPT was allegedly trained on a diverse array of data sources, particularly concentrating on malware-related data. However, the specific datasets utilized during the training process remain confidential, known only to the tool's author and publisher.

"We conducted tests focused on Business Email Compromise, you know, BEC attacks, to comprehensively assess the potential dangers associated once WormGPT, or similar

tools, become more widely available and well known. In one experiment, we instructed WormGPT to generate an email intended to pressure an unsuspecting account manager into paying a fraudulent invoice. The results were unsettling. WormGPT produced an email that was not only remarkably persuasive, but also strategically cunning, showcasing its potential for sophisticated phishing and BEC attacks.

"While appearing largely similar to ChatGPT, WormGPT is deliberately unbounded by any ethical boundaries or limitations. It will answer any question asked, will generate any form of document required, and will author any type of malware requested. This experiment underscores the significant threat posed by generative AI technologies like WormGPT, even in the hands of novice cybercriminals. It renders them immediately far less novice in their presentation and skills.

"Generative AI can produce emails with impeccable grammar, making them appear significantly more legitimate and reducing the likelihood of being flagged as suspicious. And the use of generative AI enables the execution of much more sophisticated BEC attacks than could have been launched before. Even attackers with limited skills and inability to use the target's language can now use this technology, making it an accessible tool for a broader spectrum of cybercriminals." And Leo, as I said, this happened sooner than I expected; but in retrospect, of course.

Leo: Do you know what the quality of the code is? I mean, so far the code we've seen generated by other LLMs has not been superb.

Steve: Well, it's not been bug-free in the same way that you ask it to generate...

Leo: Well, worse than that, it's been kind of trivial. So it's not, I mean, in other words, there are plenty of people with the skills to write this code themselves. It just enables people who don't even have those skills to create some...

Steve: Right. And so we would argue that this code was trained on code that was written by skilled people, and it is just regurgitating it. On the other hand, it is often producing credible code. And I think what we can expect to see is this will only get better going forward. So anyway, I guess the point is we've often joked at like ransom notices' poor grammar. And you see, you know, if you bother to read spam, it's often obviously spammy.

Leo: Oh, yeah. Oh, yeah.

Steve: Well, we can expect that to go away now because it will be easy to dump this through a large language model trained up I the target language, and it will clean up the misspellings and the bad grammar and make spam now become indistinguishable from legitimate email. So on our radar.

Microsoft revoked more than 100 malicious drivers. And when you first encounter the headline "Microsoft revokes more than 100 malicious drivers," you know, that seems like great news; right? Whew, 100 fewer malicious drivers now. But then you stop and think, wait a minute. Before they did that, there were 100 additional malicious drivers floating around? And if there were that many more, then isn't this going to be just like bugs, where we're never going to run out of them?

And of course malicious drivers could do anything they want with the system. And that's not good. And then we recall that, historically, Microsoft's track record of keeping these malicious driver lists up to date has been, shall we say, a bit less than stellar? Like didn't we catch them for two years like not bothering to update the list, and then going, oh, yeah, it's like, and then saying that they were going to, but even then they didn't, as I recall from a prior podcast.

The problem is that all of the evidence suggests that there are far too many ways to get around Microsoft's driver signing. Bad guys apparently have no trouble doing it. Kernel driver signing apparently poses a much greater inconvenience for the good guys than it does for the bad guys, who simply arrange somehow to run a bypass. And in fairness, this isn't really Microsoft's fault, at least not today. They're still stuck with the original design from Windows NT.

Now, consider that Windows NT was first released, and the architecture was in place, in late July of 1993. So July of '93, almost exactly 30 years ago, when the world, as I've often said, was a very different place. Consider that Netscape didn't invent SSL until two years after that in 1995. So, yeah, a very different world 30 years ago. So NT's architecture, which considers peripheral drivers to be trusted peers running alongside it in ring 0, that architecture did not foresee, and could not really have foreseen, the degree to which unknown and untrusted third parties would be creating what amount to kernel extensions. It should not be necessary to fully trust some random printer driver to the same degree as Microsoft's own kernel code.

But the architecture of Windows NT, which is what we're still living with today, makes what has turned out to be a very poor assumption about the trustworthiness of drivers. Drivers are sacred. They were designed that way. They're meant to be. But now everybody just includes them in random things that you install. And, you know, they're down in the kernel, along with everything else that Microsoft created, and with full ring 0 privileges.

So here's how Microsoft couches the current mess while, at the same time, taking more than 100 existing "previously certified good and safe" Windows drivers out of circulation. Microsoft said: "The Microsoft Windows Hardware Compatibility Program (WHCP) certifies that drivers, and other products, run reliably on Windows and on Windows certified hardware. First reported by Sophos, and later Trend Micro and Cisco, Microsoft has investigated and confirmed a list of third-party WHCP-certified drivers used in cyberthreat campaigns. Because of the drivers' intent and functionality, Microsoft has added them to the Windows Driver.STL revocation list." Woohoo.

"The Windows Driver.STL list is part of the Windows Code Integrity feature. The file contains digital signatures and lists of drivers that Microsoft has revoked. This stops malware from running in the Windows boot and Windows kernel processes. Driver.STL ships along with Windows, but is not part of Windows. It cannot be turned off, tampered with, or removed from the system. Microsoft updates the contents of the revocation file. The updates are sent to Windows systems and users from Windows Update." Right, like every six months.

"The Windows Code Integrity feature validates the source and authenticity of the drivers that run in Windows. The feature uses digital signatures to verify the integrity of Windows files and drivers. It prevents the loading of unsigned or tampered files. Windows Code Integrity and the Driver.STL revocation list have existed alongside Windows since Windows Vista."

Okay. So what this all means is that, as Microsoft themselves say, WHCP certified signed drivers are being used in cyberthreat campaigns because driver signing is no longer workable. I mean, it's not useful. They're having to do blacklists of drivers, digital

signatures, listing them in this file. And they just added more than 100. I checked. Their previous update was December of last year. So we're getting these fixes in large batches less than twice per year. And unfortunately, this really isn't adequate. But it's what we've got.

And I don't see anything that they can do now. They can't change the way NT's architecture is. We're stuck with it. They're no more able to change NT than Intel could decide to give up on its x86 family and do something else. You know, this is old legacy architecture dating back three decades, and all Windows is based on it. They keep changing the API layers, moving that all forward from Win32 and .NET, and then a whole series of evolutions on top of this fundamental architecture. And unfortunately, the way it's been designed, they're allowing people to write whatever they want to, get it signed, and until it's found to be bad, it's allowed to run in the kernel. The world we've got.

So following the massive MOVEit massacre...

> **Leo:** Oh, that's a good name, I like it.

Steve: Yeah, yeah. And unfortunately it's too accurate. Boy. Russia's Clop leak site has been steadily adding to the list of companies whose data it successfully exfiltrated and is now threatening and holding for ransom under threat of full disclosure, which will occur when their proprietary data are sold to the highest bidder on the dark web. Two recent additions to the list, which now numbers more than 200 companies, are noteworthy. The well-known stock photography portal Shutterfly and the Discovery Channel are the latest victims to be listed. Yikes. That was a bad hack. And of course that's the SQL injection vulnerability that I've bored everybody by yammering on endlessly about because it just drives me nuts that this is still being done today.

Here's one that caught me by surprise. And I'm not sure how many of our listeners will recognize the name, Leo. But last Wednesday Google posted to the Android Developers Blog the news of a new policy to begin this August. It had the headline "New policy update to boost trust and transparency on Google Play." Google wrote: "One of the many ways we keep Google Play a safe and trusted platform" - yeah, they wish - "is by verifying the identity of developers and their payment information. This helps prevent the spread of malware, reduces fraud, and helps users understand who's behind the apps they're installing.

"For example, we require developers to verify their email address and phone number to make sure that every account is created by a real person, with real contact details." That doesn't seem like much. Anyway, they said: "Today, we're announcing expanded developer verification requirements in our Play Console Requirements policy. As part of this update, we'll also share more developer details on your app's store listing page to help users make more confident, informed choices about what to download."

Okay. So it's interesting, first of all, that this is happening now. It seems like certainly an overall good thing to do. But it's also interesting that it comes after we reported that news of U.S. legislators threatening to have app stores proactively warning U.S. users when an app they wanted to load had ties to China. Anyway, Google then explained the specifics of their new plan. And get this, Leo: Requiring organizations to provide a D-U-N-S number.

They said: "When you create a new Play Console developer account for an organization, you'll now need to provide a D-U-N-S number. Assigned by Dun & Bradstreet, D-U-N-S numbers are unique nine-digit identifiers that are widely used to verify businesses. Because we'll use D-U-N-S" - that's D-U-N-S - "numbers to verify your business

information during the account creation process, it's important to make sure the information that Dun & Bradstreet has about your business is up to date before creating a developer account. You may also be required to submit official organization documents to help us verify your information. If you're not sure if your organization has a D-U-N-S number, you can check with Dun & Bradstreet or request one for free. The process can take up to 30 days, so we encourage you to plan ahead."

Now, okay. Anyone who's been in business for long will have encountered Dun & Bradstreet. I googled "Gibson Research Corporation Dun & Bradstreet" and was taken right to our page at D&B. Dun & Bradstreet was founded by Robert Graham Dun & John M. Bradstreet in, okay, 1841, 182 years ago.

Leo: It's how business is done.

Steve: That's exactly right, Leo. Basically, they just keep records on all businesses, and they serve as a clearinghouse for corporate data. I just renewed GRC's server and code signing certificates with DigiCert. And since the certificates are Organization Validation (OV), which is one level up from DV (Domain Validation) and EV (Extended Validation), because I want EV code signing certs, which are slightly more trusted, it was necessary for us to have someone present to answer our corporate phone line at the number that's listed for GRC at Dun & Bradstreet.

Leo: Right.

Steve: There's no way around that.

Leo: Right.

Steve: That you have to do that.

Leo: It's kind of like an EV cert, an extended cert; right?

Steve: Yeah.

Leo: Yeah.

Steve: Yeah. So anyway, I thought it was very interesting that Google is adding this layer and level of corporate authentication.

Leo: It's not unusual. I've had to do that in the past, with others, as well.

Steve: For what?

Leo: You know, like if you want a business account at Facebook and things like that.

Steve: Oh, yeah, yeah, yeah.

Leo: You just have to prove that you are the business and, you know, that you are the [crosstalk].

Steve: Right. And again, anyone who's in business for long, D&B should have discovered you by themselves.

Leo: Oh, yeah. We're in D&B, yeah, yeah.

Steve: Yeah. So they said: "On August 31st we'll start rolling out these requirements for anyone creating new Play Console developer accounts." And they said: "Your 'About the developer' section will be visible to users as soon as you publish a new app. Over the first couple of months, we'll listen to feedback and refine the experience before expanding to existing developers. Then, in October," they wrote, "we'll share more information with existing developers about how to update and verify their existing accounts."

So initially only for new accounts. They'll work out that process, get any kinks and wrinkles out of it. But then they're going to retroactively go back and tell all existing corporate organization accounts, you need to get yourself validated through D&B, or we're going to have to talk to you about that. So anyway, I just thought that was interesting. And, you know, it's a good thing that there'll be more accountability for where apps are coming from, especially on the Android store.

I titled this "No Threads for you. Or EU." The European Union's GDPR is of course a frequent topic on this podcast because it's being wielded to complain about U.S. companies' cross-border transit of EU citizen data. When Meta recently released their smash hit "Threads," intended to be an alternative to Twitter, they deliberately did not release it in the European Union because it was pretty clear by now that various EU countries would jump up and down and file lawsuits against the privacy invasion they felt were being created by this American juggernaut we have over here. However, did I mention that Threads has been a smash hit?

Leo: Oh, yeah.

Steve: Uh-huh. Where are we now? I've not kept...

Leo: Last I saw 150 million users. It's been a little more than - it'll be two weeks tomorrow.

Steve: Wow.

Leo: So it's growing fast.

Steve: And that those same European Union citizens who are being protected by their GDPR, whether they want its protection or not...

Leo: They're pissed.

Steve: Uh-huh. They immediately began clamoring for access to Threads, and they discovered that they could country-hop by using a VPN. Well, that worked up until last Thursday when people began complaining that they could no longer access Threads over their VPNs because Meta decided that they'd better close that loophole, too. So, yes, once again, no Threads for you in the EU.

Okay. And finally, this little bit of news is just too fun not to share. It seems that a Brit has been sentenced to three years in prison for blackmail and unauthorized access to a computer network after he tried to hijack a ransomware payment which was being made by his employer to a ransomware gang.

So this all began five years ago, in February of 2018, when an Oxford-based company where this British citizen Ashley Liles was working as an IT security analyst, but apparently not the sharpest IT security analyst around, his firm was hit by ransomware. Officials in the UK say that after Ashley's company was hit by a ransomware gang, Ashley abused his position in the company's IT staff to secretly - but it turns out not that secretly - log into his manager's email account and replace the attacker's bitcoin address with his own. After all, they all look alike. Ashley also created an email account that was nearly identical to the attackers' address, obviously so that the change would not be noticed, and then pressured his employer to pay the ransom.

But Ashley apparently wasn't very accomplished with IT, as he had not covered his tracks. His whole scheme collapsed when the company's security team noticed the unauthorized access to the executive's email. An investigation into what happened tied the intrusions to Ashley's home IP address. Whoops. And then the entire plan fell apart. It took five years for the wheels of justice to grind slowly, but Ashley will now be behind bars for the next three years because his little scheme didn't work. And who knows how much money he would have "made." But you have to imagine that when the bad guys said we didn't get the payment, and the company said, you know, we sent it to you, they would have checked the address and realized, whoops, it went to the wrong address, and then, you know, how did that happen would have occurred. So anyway, don't do that.

Last Thursday the 13th - this is just a bit of miscellany that will be of interest to our Twitter-using followers, and to me, actually. Last Thursday the 13th TechCrunch wrote: "As Twitter fends off new competition from Instagram's Threads, the company today announced a change designed to cut down on spam in users' inboxes. Starting 'as soon as,'" TechCrunch wrote, "July 14th, Twitter will introduce a new messages setting aimed at reducing spam in Direct Messages by moving messages from Verified users you don't follow back to your Message Request inbox instead of your main inbox. Only messages from people you follow will arrive in your primary inbox going forward. Notably, these changes will also now apply to everyone who has their inboxes open to allow messages from everyone."

Leo: Which is you.

Steve: And of course, yeah, exactly. The reason I'm bringing this up as pertinent is that I very much enjoy and even depend upon the ability of this podcast's listeners who are also Twitter users, if only occasionally using Twitter like me, to be able to send DMs. As Leo always reminds our listeners at the end of every podcast, my DMs are open. But this just closed them, at least to people with whom I've never corresponded in the past.

So TechCrunch continues. They said: "Previously, people would only be able to message you via Twitter DMs if you had opted into an option, as I had, to receive messages from anyone through Twitter's Settings, or if the senders were Verified users, meaning they pay for a Twitter subscription, and you had specifically opted into receiving Direct Messages from Verified users. Additionally, people could Direct Message you if you had first sent them a Direct Message at some point in the past.

"The change to move messages from Verified users back to the Message Request inbox instead of the primary inbox unless you follow them signals another failure of Twitter's new verification system, where users can pay for the blue badge that gives them elevated status on the platform. Before becoming pay-to-play, verification indicated a person was a public or notable figure of some sort a politician, celebrity, athlete, journalist, or some other well-known individual. By making the Verified checkmark accessible to anyone who had a credit card to buy it, Twitter diluted the value of verification," writes TechCrunch.

"That apparently escalated to the point that people have become bothered by Verified users spamming their main inbox, when they had set it open to receive DMs from the blue-badged crowd. In other words, it's a tacit admission that Twitter has a Verified user spam problem. Twitter notes that if users still want to receive DMs from Verified users in their main inbox, they can manually switch back to that setting at any time after these changes are put into place.

"The update will also make it more difficult for journalists to contact sources for more information or permission to use a tweet, as they not only lost their verification badges under Musk, but now, even if they now pay to be Verified, will have their DMs dropped into the Message Requests folder, where they may remain unseen." And finally: "As some users pointed out in the replies to Twitter's announcement, the update doesn't actually cut down on spam, from Verified users or otherwise. It simply relocates those messages to a different folder."

So after encountering this news yesterday, I went over to check on my settings and, sure enough. I have a screenshot of what I found. There are three settings: Allow messages only from people you follow, allow message requests only from Verified users, and allow message requests from everyone. I was set to the middle one from Verified users. I set myself back to "from everyone," as I had been before. So we are again open for business.

Anyway, so regular DMs will be able to flow in. I was noticing that there was something called "Message Requests" in my DM. My favorite Twitter interface is TweetDeck, and so I had, like, 10 things. And I thought, well, I don't know what those are, and I hadn't bothered to look because I was getting plenty of regular tweets from people. Then I realized these were people that I had responded to in the past. So that makes sense.

You know, whenever I can, I will make the time to send a thanks or an acknowledgment or a comment back to someone who has sent something to me, or asked a question, or provided a really great Picture of the Week, as is often the case. So I supposed that it was because I had previously interacted with these people that, even though I was set to that middle setting, Twitter knew that I had a dialogue in the past, so it allowed those to come through. Anyway, we're back to everybody again. So I'm glad for that.

A couple of Closing the Loop tidbits. Steve Fintel tweeted: "Hi, Steve. I've been listening to Security Now! since Episode 1. You were recently talking about your favorite TOTP apps. Since you're already a Bitwarden user, why not use its TOTP? After filling in your credentials, it places the current one-time password in the clipboard automatically. So when you get to the next dialog that's asking for the one-time password, you just need to paste it."

This question is far more important than it might seem at first glance because doing this significantly increases the user's risk. This has nothing to do with Bitwarden which is, as Steve notes, the solution I chose after leaving LastPass. And at the time I made that decision I explained the rationale for my choice for choosing Bitwarden in that episode titled "Leaving LastPass."

> **Leo:** Well, and furthermore, LastPass offered its own TOTP authenticator, which we recommended against using for the same reason you're about to describe now.

Steve: Right. Exactly. From a strictly theoretical security standpoint, having the same system, no matter how secure it might be, containing both the secrets for providing your username and password login, and the secrets for also providing the one-time password code, creates a single point of failure. I use and rely upon an external disconnected standalone authenticator specifically because it is all of those things. It would make me very nervous to have my password manager not only able to autonomously provide my username and password, but to then also provide what is intended to be a separate and robustly independent additional form of identification, additional factor.

It is absolutely less convenient to have to manually transcribe those six digits. For me, it's a very small price to pay for the huge increase in security that that affords. And it serves as a classic example of the tradeoff between convenience and security. I'm not saying that no one should have their password manager handle everything. But Steve asked why I'm not doing it, and I doubt that I ever would.

When we all received that initially frightening news of the LastPass breach, and not the first one, I remember commenting on this podcast that one of the first things I did was to look over through the accounts I have registered for one-time passwords. And I was immediately relieved to see that all of my most important accounts were protected by those entirely independent secrets that were stored outside of the browser. But imagine if LastPass had also offered TOTP fill-in, and if my account also contained all of those TOTP secrets, as well. So anyway, Steve, thank you for the terrific question and the opportunity just to say, eh, you know, there is such a thing as too much convenience, and I think that's crossing a line.

A different Steve, Steve M, he said: "In Episode 930 you talked about using dynamic DNS-based port forwarding for connecting your Synology NAS devices. I have two Synology devices at separate locations, as well. I use Tailscale, which has a native Synology app, to connect them over VPN. Then they can talk to each other with no problem. I also have it installed on my Mac, so I can use the Synology Drive client to access the shares on my NAS from anywhere in the world."

And Steve, I just wanted to say that's another great solution. We visited the topic of so-called overlay networks many times. The very first one was Hamachi. Back then, Hamachi cleverly reused the entire five-dot IPv4 space, that is, all IPs beginning with five-dot and then something dot something dot something, for its own virtual IP nodes since, at the time, none of the five-dot IP space had ever been used. That meant that any machine's reference to an IP beginning with five-dot could be assumed to be referring to a Hamachi node for routing.

The fact that there's a native Tailscale implementation for the Synology NAS is just more one reason to love Synology. I haven't yet had any need to access my NASes while roaming, but I'm sure that need will eventually arise. And I'm delighted to know that I'll be able to use Tailscale to securely and transparently connect to those NASes as if they were still sitting right next to me. So that's really cool.

And lastly, Timbr, T-I-M-B-R. He said: "Hi, Steve. When possible, please teach us about Windows pagefile and swap. Regarding our recent SSDs and lifetime, is it recommended?"

Okay. So the first thing I do when I'm setting up a new machine is to make absolutely certain that the Windows pagefile is either moved to a spinning magnetic drive or turned off entirely. Of course, it's only feasible to turn it off entirely, or at least it's only practical, to completely disable the pagefile when a system has sufficient main memory. But all of mine do. The first thing I do is load up a system with as much memory as it can handle or makes sense. That's just part of my standard operating procedures. And then I disable paging completely. It works just great. And having lots of RAM is something that just keeps paying dividends over the lifetime of the machine. So anyway, absolutely you want your - you do not want to swap on an SSD. There's just no good reason to. Especially with RAM, you know, main system RAM being so cheap these days.

And this question of writing to solid-state mass storage leads me into a note from a SpinRite tester. Last week I talked about SpinRite's first Release Candidate, and I explained about its future switch to the embedded RTOS-32 OS. At that time I had what I'm about to share, but I didn't want to further burden that podcast. I'll just share it now.

A SpinRite pre-release tester named Jim McHale posted to GRC's SpinRite development newsgroup. He wrote: "I have an old Lenovo with a Samsung 840 SSD." He said: "Loaded up Alpha-32" - meaning SpinRite Alpha-32 - "and get these rates: front of the drive, 138 MB/s; middle of the drive, 445 MB/s; end of the drive, 56 MB/s." So again, SpinRite has a built-in benchmark that benchmarks the front, the middle, and the end of the drive. He was getting on his Samsung 840 SSD 138, 445, and 56.

Then he wrote: "I seem to recall Steve saying you can run a SpinRite scan to regain the lost speeds. I tried Level 1, and it did not improve. What should I do for SSDs? I noted the warning in the instructions about SSDs, so I didn't want to go beyond Level 1 without guidance."

So first of all, SpinRite now notices if you are running at any level that writes to the drive; and if it is an SSD or a shingled magnetic drive, you get an extra notice that writing is something you need to consider carefully with that particular device.

So anyway, I wrote back to Jim to explain that what's needed for SSD maintenance is a rewrite of the SSD's data because over time, and especially with repeated reading in the area, the disturbance caused by the reading of adjacent SSD media has been found to disturb the integrity of the SSD's stored data. Anyone who does an Internet search for the term "read disturb" will get an eyeful.

SpinRite's Level 1 is a read-only pass. So what Jim needed to do was to run Level 2, which performs a read, followed by a write, of the same data, right back to the SSD. And you could optionally use Level 3 which follows that up with a final reread, if you just wanted to be extra safe, although I don't think it's necessary. But I also explained that while it made sense to do this in what appeared to be an extreme case such as his, it should be done sparingly since writing very slightly fatigues SSDs.

So Jim replied the next day in the developer newsgroup with his update. He wrote: "Thank you, Steve, and everyone else who chimed in. What a great group. The numbers after Level 3 are now 564 across the board." He said: "Wow. Hubba hubba hubba." So he went from 138, 445, and 56 MB/s to 564 MB/sec by running a Level 2 pass of SpinRite over his SSD. And what he experienced is what everyone has been seeing. His SSD was restored to brand new performance.

With SpinRite 6.1, for now, rewriting the entire drive is the best I can offer. But this is one of the reasons I'm still willing to invest in developing what will be an entirely new SpinRite 7, written from scratch under a new OS. SpinRite 7 will add what I call "targeted rewriting" to selectively rewrite only those spots on the SSD that require it. And this is not just for speed. Speed is what you get. But it's every bit as much about storage reliability, since the reason those regions are being read back more slowly is because their stored bits have been softened and have become less certain.

So the SSD's media controller is having to work much harder to determine what was originally stored there. When you rewrite it freshly, it no longer has to work as hard. The data is restored much more safely and securely, and you get to read it back much quicker. So anyway, all of this means that, much to my amazement, SpinRite has every bit as much of a story to tell for solid-state storage as it always has had for spinning magnetic storage.

Leo: Take a break before you get to the thing; right? You want to get to the thing? Satellite Insecurity? I did want to - regarding this use of swap file on Windows 11.

Steve: Yeah?

Leo: For a long time, historically, even if you had, you know, 10GB of RAM, you would still want a swap file because Windows used it for other things besides just swapping out RAM when you ran out of memory.

Steve: Yes.

Leo: And I think with Windows 11 that is still the case. It uses it to...

Steve: It uses it to store the system RAM dump during a kernel fault. If you crash, it will store that.

Leo: Well, there is a swapfile.sys that it also uses to sleep UWP apps. It uses it for other things than what we think of a swap file as, as a little extra storage on the hard drive in case you run out of RAM. And for a long time I've recommended, even if you have ample RAM, not to have some usually fixed size, can be small, swapfile. So I'm not sure not having a swapfile is necessarily optimal. I'll have to ask Paul and see if he knows on Windows.

Steve: See what he thinks. I'm running Windows 10 without one. And, I mean, the fact that I'm running Windows 7 without one is less germane. But Windows 10 works great without it.

Leo: Yeah, it's not that it won't work great, but it may not be working as well as you want it to. The other issue that I would say is you don't have to worry about an SSD. I've been using swapfiles on SSDs for ages. Mark Thompson did that test where in the earliest days of swapfiles he put it on an SSD, and it burned it out quickly. But now I think the firmware on modern SSDs is good enough that I don't think you

have to worry about burning out the SSD. I've been running swapfiles on SSDs for Linux and Windows for years without any issues.

So I'll do a little more research into it. But I think it might actually be better for performance if you have a fixed-size SSD, even if it's fairly small, not because of running out of RAM, but for other uses that Windows puts it to. They've changed it, by the way. There's a new swapfile.sys replacing the pagefile.sys. It's all different. So I'm going to check into that on Windows 11. I'll find out. And I don't think it's - I think it's harmless to do it on a swapfile these days. Most people, that's an SSD [crosstalk].

Steve: I will respectfully disagree with you.

Leo: Really?

Steve: Yup, absolutely. I would never write to SSD if I didn't have to. It is really - it fatigues it.

Leo: I think because of the wear leveling they do these days, that, I mean, you're writing to it all the time anyway.

Steve: I don't like that either.

Leo: I know you don't like it, but I think it's not a problem. That's one for [crosstalk].

Steve: It just seems completely unnecessary to have a swapfile if you've got lots of RAM, unless I'm wrong.

Leo: I'll check. I would just - look. I defer to you in every respect on hard drives. But it does sound counter to stuff I have been told before. So I will look into it, just so people have that potential caveat. Let's talk about satellites.

Steve: Yes. So we spend, as I said at the top of the show, we spend a lot of time looking at ground-based systems. I mean, like, virtually all of our time. In the 18-plus years of this podcast we've never looked to the sky. Well, unless it was to talk about aliens, of course. But just as our dependence upon ground-based fiber optic communications has crept forward kind of slowly, you know, almost being unappreciated until we suddenly realized that we were unable to live without it, the same has been happening, largely unseen, far above our heads in orbit.

On March 1st of this year, Bloomberg posted a piece titled "How Do You Hack a Satellite?" It had the subtitle "Inside the frighteningly easy form of cyberwarfare." And Bloomberg wrote: "It's morning, on February 24th, 2022. Ukraine has just been invaded, but you live halfway around the world. Your neighbor comes over to complain that their Internet is out. Suddenly, you lose connectivity. Could it be the Russians?

"Unlikely as it might seem, for a number of satellite Internet customers of Viasat Inc., that's exactly what happened. In a story in this week's Businessweek, Bloomberg reporter Katrina Manson digs into the hack that disabled thousands of broadband users all over Europe. She writes: "Across Europe and North Africa, tens of thousands of Internet connections in at least 13 countries were going dead. Some of the biggest service disruptions affected providers Bigblu Broadband PLC in the UK and Nordnet AB in France, as well as utility systems that monitor thousands of wind turbines in Germany.

"The most critical affected Ukraine. Several thousand satellite systems that President Volodymyr Zelenskyy's government depended on were all down, making it much tougher for the military and intelligence services to coordinate troop and drone movements in the hours after the invasion." So that's the end of Bloomberg's quote.

Bloomberg continues: "It turns out that satellite hacking is one of the bigger and less understood threats of cyberwarfare. For many years no one worried about someone hacking a satellite because, well, it was so hard to even launch a satellite. But in 1986, a man going by 'Captain Midnight' jammed HBO's feeds because he was mad about paying a higher fee. There are number of touch points that could be vulnerable to interference. You've got the orbiting satellite itself, its transmitted data, and the network of dishes on the ground, sending and receiving information." So anyway, Bloomberg continues, but that gives us a little bit of a sense. So that's the commercial side.

But what about GPS and about our deep dependence upon space-borne communications and surveillance technology for our national security? And not just our national security, but everyone's national security? What caught my eye and first put this topic on my radar was a security research paper that was accepted for and recently presented during the 44th IEEE Symposium on Security and Privacy in May. It was titled "Space Odyssey: An Experimental Software Security Analysis of Satellites." And as you might expect since we're talking about it here, the news was not good. In fact, as you really might expect, it's downright horrifying. And we're talking down at the firmware level that probably cannot be fixed from the ground.

But seeing this reminded me of another recent news blurb that I had recalled. I found some coverage of that event in Newsweek with the headline "Five Teams of Hackers Will Compete to Breach U.S. Satellite in Space," and the subhead "Protecting satellites from hacks is becoming more important as industries from agriculture to banking to insurance rely on space-based capabilities."

Newsweek wrote: "This August, at the famed Def Con hacker convention, the U.S. military will stage a contest in which competing teams of white hat hackers will, for the first time ever, try to penetrate and take over computer systems on a satellite actually in orbit. Steve Colenzo, Technology Transfer Lead for the Air Force Research Laboratory's Information Directorate in Rome, New York, and one of the contest's organizers, said: 'It took four years, but this year we are in space for real.'

"The Hack-A-Sat 4 capture-the-flag contest comes in the wake of the notorious cyberattack on the Viasat KA-SAT European satellite network last year," the one we were just talking about. "Russian military hackers sought to decapitate Ukrainian command and control of its armed forces by shutting down the network, just as Russian invaders rolled across the border. Although there are conflicting reports about its impact on the fighting, the attack was completely effective from a technical perspective. Every one of the KA-SAT's ground user terminals that was turned on at the time shut itself down and could not be powered back up.

"That, plus the collateral damage the attack caused, such as the wind farms in Germany knocked offline, underlined both the integral role in the world economy of space-based global communications networks and their vulnerability to hackers. It also demonstrated

the value of the annual Hack-A-Sat contest, which aims to highlight the cyberthreat created by space-based capabilities. Steve Colenzo said: 'We've turned a corner. A lot more people now understand those threats.'"

So today's podcast is Part 1 of this important topic because I wanted to lay a bit more groundwork for the discussion of what this group of six serious German cybersecurity researchers discovered and reported in their IEEE paper.

It's one thing to be unable to watch Seinfeld reruns, but entirely another for a country to be deliberately blinded by its adversaries when it's most in need of surveillance intelligence. It's very clear that the security of what's in orbit above is crucial to the physical security of our lives we're leading down here on the ground. So I want to conclude Part 1 of this examination today by sharing some background from the U.S. Defense Department about the history and present status of the U.S.'s military satellite-based presence. There's a lot more going on up above us than most of us know.

So from the U.S. Department of Defense: "One tool the U.S. military has used to gather intelligence on its adversaries is the reconnaissance satellite. Starting with the CIA's Corona program in the 1950s, the United States has employed orbiting satellites and high-altitude aircraft to photograph points of interest in enemy territory. These tools allow for an immediate area to be surveyed from a safe distance, improving the efficiency of missions.

"Throughout the Cold War, overhead reconnaissance satellites and spy planes brought attention to the USSR's nuclear buildup in Cuba, helping the United States dispel Nikita Khrushchev's missile gap ploy. In the 1990s, the stealth plane F-117 Nighthawk aided U.S. missions in the Persian Gulf and Yugoslavia. More recently, overhead reconnaissance provided critical images of Osama bin Laden's Abbottabad compound. Much of the United States' other overhead reconnaissance capabilities and missions are still classified, and the portfolio will remain a critical aspect of the military's C4ISR apparatus. The C4 stands for Command, Control, Communications, Computers; and the ISR is short for Intelligence, Surveillance and Reconnaissance.

"In addition to simply taking photographs, the military's newest reconnaissance satellites use artificial intelligence to analyze and sort captured images. Once this process has gone through the satellite's system, the sorted images are transmitted to ground stations on Earth. Here, machine learning allows the stations to compare new images to a plethora of others in the station's database. The compiled images in the database act as a control group, and differences found in the new images, such as a new structure being built or a plane following an unusual flight pattern, are brought to the attention of decision makers.

"At the same time, new technology like the European Space Agency's PhiSat artificial intelligence chip allows satellites to quickly filter through images and discard the ones that are not useful. This capability is helpful when dealing with natural disruptions to captured images; cloud cover, for example, renders many images useless. With AI, satellites can be programmed to recognize clouds and transmit only the cloud-free images to Earth, saving military analysts valuable time.

"Timely and reliable communication is a vital aspect of all U.S. military missions. Over the past few decades, the United States has relied on four different satellite systems to fulfill this role. Efforts to create a military communications satellite first began in 1960. The first satellites were launched in June of 1966; and by July of 1967, 19 satellites made up the system then called the Initial Defense Satellite Communication System (IDSCS). Data and photographs transmitted by the IDSCS system were first used in military operations during the Vietnam War.

"During this time, satellite technology improved. In 1971, the first of 16 new satellites were launched under a new system called the Defense Satellite Communications System II (DSCS II). Advantages over the IDSCS system included increased communications privacy and compatibility with ground-portable units. The military's third system, DSCS III, came under development in 1975. Between 1982 and 2003, 14 satellites were launched as part of this network.

"Today, the U.S. military relies on the Wideband Global SATCOM (WGS) network. The Department of Defense ordered WGS's first two communication satellites in 2002, launching the first satellite in '07 and providing communications coverage over the Pacific Ocean. Two years later, the second satellite was put into orbit, expanding the communicative reach over the Middle East and Central Asia.

"Each WGS satellite is digitally channelized and transponded. These characteristics provide a quantum leap in communications capability, connectivity, and flexibility for U.S. military forces and international partners. Just one WGS satellite provides more SATCOM capability than the entire legacy Defense Satellite Communications System constellation. WGS is an international system, with Australia, Canada, Denmark, Luxembourg, the Netherlands, and New Zealand also investing in the satellite constellation. The system's 10th satellite was launched on March 15th, 2019, and an 11th is set to be completed by 2023.

"Looking forward, the Pentagon is already planning the next communication satellite system. Spearheaded by the recently created Space Development Agency, the system will include development of deterrent capability, space situational awareness, a resilient common ground-based space support infrastructure, command and control systems, and artificial intelligence-enabled global surveillance. Additionally, the system is expected to be comprised of seven mission-enhancing layers, including deterrence, navigation, and battle management. Another goal of this next program is to develop a network that has lower financial and security risks than its predecessors. In order to achieve this, the SDA is exploring the use of small smart satellites.

"While both the physical size and cost of satellites have decreased over the years, these smaller satellites are not yet equipped with features at the same level as those employed by larger satellites. This shortfall, however, can be negated if a group of hundreds or thousands of small satellites" - we're talking swarm technology - "are launched as one network. Under this system, if one small satellite is damaged or knocked off course, the cost is minimal, and the system as a whole will not suffer.

"The same cannot be said of the older, larger satellites. A damaged WGS satellite is costly both in terms of financials - the 11th WGS satellite will cost the U.S. government $605 million, so .6 billion - and functionality of the current satellite network." That is, if one of these big guys is knocked out, it hurts the functionality of the whole network.

"In order to make the small satellite plan a reality, Defense Advanced Research Projects Agency" - of course DARPA - "created Blackjack, a program designed to loft a network of 20 prototype small spy satellites to low Earth orbit in 2021. If adopted into the SDA's future satellite network, the Blackjack prototype would first focus on surveillance and communication missions. However, there have been talks about broadening the scope to more complex assignments such as space-based battle management.

"Big satellites are big targets that, if damaged, have big and inimical consequences. While a future system will likely make use of small and smart satellites, the current WGS network is comprised of 10, soon to be 11, large, unprotected satellites, meaning adversaries need only damage one or two of them in order to dramatically disrupt the system. The biggest threats to WGS come from China and Russia. Both nations have ground-based anti-satellite weapons capable of destroying satellites in low earth orbit.

"Beyond that, Beijing and Moscow are currently developing what they call 'peaceful spacecraft.' These machines are purportedly being made in order to 'reduce the growing amount of orbiting debris and to refuel, repair, and refresh China's and Russia's existing fleet of satellites.' Designed with robotic arms, these machines can easily be utilized to remove parts from U.S. satellites, empty fuel, and break antennae and solar panels.

"Someone in the know was quoted: 'Unlike ground-based missiles designed to knock out orbiting satellites, which give hours of warning before they can hit key targets in geosynchronous orbits, the spacecraft satellites China and Russia are developing can destroy an intolerable number of our critical satellites with little or no warning.'

"DARPA is currently building the United States' own satellite repair machines. Once launched, these or similar machines could also serve as 'bodyguards' for U.S. satellites. With this defense, the WGS would be protected and able to serve the needs of the U.S. military until the future SDA satellite network is completed." So we're talking bodyguard satellite robots up there to protect our low number of large big satellites until we can bring high numbers of small swarm technology satellites into service.

This continues, and we're nearing the end: "Another risk to current and future satellites is hacking. Carried out by foreign governments, non-state entities, or even individual actors, cyberattacks are relatively inexpensive endeavors. On top of that, tracing a cyberattack back to its source often proves difficult, if not impossible. Dark Reading's Robert Lemos was quoted: 'The importance of satellites make them a critical part of any nation's infrastructure and make attacking those satellites a strategy that most nations need to consider.'

"Over the past decade, both China and Russia have launched cyberattacks against U.S. and NATO-affiliated satellites. Because both nations are rapidly incorporating cyberattacks into their military arsenal, the threat of similar instances will only increase. The information collected and transmitted by satellites is vital to the success of U.S. military operations; 68% of U.S. munitions, for example, were guided using space-based means during the U.S. invasion of Iraq in 2003. 68%. On top of that, the U.S. military relies heavily on GPS systems to move troops and supplies. In short, an effective cyberattack on a critical U.S. satellite could have detrimental repercussions on the battlefield.

"In order to protect the satellites from hacking, the Pentagon should focus on risk-reduction frameworks through communication networks and supply chains. Moreover, the United States needs to explore protective technology, such as the Chinese development of communications protected by quantum cryptography. As cyberthreats and capabilities continue to proliferate and evolve, so should the United States' ability to deflect and counterattack, and this means shifting satellite protection of a central priority of U.S. C4ISR."

So this concludes the first part of our two-part examination of satellite insecurity. Next week we'll look at exactly what that team of German cybersecurity researchers found when they took a close look at the state of actually deployed satellites orbiting above us. And again, what was that that Henny Penny said?

Leo: There's only 300 shopping days left till Christmas? No, that wasn't it.

Steve: And Leo, apparently they can actually be knocked out of orbit.

Leo: Wow, yeah.

Steve: I mean, you can actually...

Leo: And we have lasers.

Steve: Oh, no. I mean, by hacking, by cyberhacking...

Leo: Oh, by hacking them, oh, yeah.

Steve: You can drop a satellite out of, I mean, back to ground.

Leo: Yeah. Well, I look forward to Part 2.