Preventing Windows Sandbox Abuse

Description: Why did a mysterious empty "inetpub" directory appear after April's Patch Tuesday? And what new Windows Update crashing hack did this also create? North Korea is now creating fake U.S. companies to lure would-be employees. The "Inception" attack subverts all GPT conversational AIs. New information about data loss in unpowered SSD mass storage. Lots of terrific feedback from our listeners. How malware has taken to hiding inside the Windows Sandbox, and what you can do to stop it.

High quality (64 kbps) mp3 audio file URL: http://media.GRC.com/sn/SN-1023.mp3
Quarter size (16 kbps) mp3 audio file URL: http://media.GRC.com/sn/sn-1023-lq.mp3

SHOW TEASE: It's time for Security Now!. Steve Gibson is here. Some wacky stories today, including the explanation behind the mysterious appearance of the inetpub directory on your Windows machine. It's on purpose, and don't delete it. We have new information about data loss in SSD mass storage. If you leave it lying around, you might lose some data. Plus malware has found a new place to hide inside Windows. All that and more coming up on Security Now!.

Leo Laporte: This is Security Now! with Steve Gibson, Episode 1023, recorded Tuesday, April 29th, 2025: Preventing Windows Sandbox Abuse.

It's time for Security Now!, the show that I think many of us wait for all week long. If it's Tuesday, it must be time for this guy right here, Steve Gibson.

Steve Gibson: I know that I wait for it all week long because it's a major event in my weekly cycle.

Leo: It's a lot of work, I'm sure, to pull all this together.

Steve: It's a lot of work. I start Sunday around noon, after Lorrie and I have caught up with the Sunday morning shows. And I work all day Sunday and then all day Monday. So it's basically two days out of the week. And it's funny, too...

Leo: Steve, you have our eternal gratitude. Thank you.

Steve: It's funny because Lorrie says, "Why can't you just cut and paste more?" And I go, well, because I like talking. And, you know, I want to put myself into this, not just, you know...

Leo: Can't just cut and paste more.

Steve: ...echo other people. So I end up like really having a good time. And that's the problem is, yes, it's a big commitment. But I really do enjoy it.

Leo: Good. Well, we enjoy you, and we thank you so much.

Steve: So, you know, and I'm unable to do anything halfway, as you know.

Leo: Yeah. That's right.

Steve: So I've got enough going all the way.

Leo: Good. Good.

Steve: So we are at the monumental episode 1023, which will be a significant number to all those who've ever studied computers, on the binary side especially, because that would be 111111111. That's nine ones, folks. And when we get one more one, then we wrap around 2^10. And, you know, 1024, that is one binary K. And we will be there next week.

Leo: When I first moved to San Francisco, the first place I rented, the address was 1024 Page Street.

Steve: Ah.

Leo: And I thought, no one else appreciated it, but I thought, this is cool. I'm on the 1K Page. So, yeah, those numbers are important to weirdos like us.

Steve: And that's one of those things, you know, you glance at the digital clock, and it says 5:12, or it says 10:24.

Leo: Yeah, exactly. Yeah.

Steve: Or it says 2:56 [crosstalk].

Leo: Or 11:11, yeah.

Steve: So, yeah, I do live in that world. Okay. So last week we introduced a lot of our listeners to the Windows Sandbox, the fact that this really well-designed piece of work was sitting in Windows 10 and 11, largely under-utilized, sometimes completely unknown because who would know? I mean, unless someone's - unless you had some reason to go looking for something, like this capability, and someone said, hey, just try the Windows Sandbox. It's built in. Anyway, now everybody knows.

As I also mentioned in passing last week, the thing that reminded me of its existence is that malware, of course, has figured out how to crawl in there and use the sandbox behind people's backs without them known it's even there. So all of that cool separation that you get which Windows Sandbox legitimate users take advantage of, malware has figured out a way to do, too. There are a couple solutions for preventing its abuse, which we're going to talk about after we finish talking about the nature of the abuse.

But first we've got this bizarre appearance of a directory I'm very familiar with. Inetpub is the directory that's always created when Internet information services, IIS server, is instantiated into typically a server, but also workstations. I have it, I've long had it on all of my machines because I've always been using Microsoft's web services to deliver websites. Well, it mysteriously appeared, unbidden, so everyone believed, after April's Patch Tuesday. And there's a big story there that we're going to get to.

We also have our friend who tweets as Gossi the Dog, Kevin Beaumont, has found a way to crash Windows Update using this mysterious directory, which I'm sure Microsoft did not intend. This whole thing has just been a big cluster you know what, a mess. We also have North Korea now creating fake U.S. companies. They have, like, the one division that is spoofing fake employees. They said, well, let's get it on the other side. Let's create fake U.S. companies, see how that goes. We have a new attack on GPT-style conversational AIs, known as the Inception Attack, which subverts them. Also, a bunch of people sent me questions about this, so I figured since everybody is concerned about mass storage, we've got some really interesting new information about the data loss...

Leo: Sorry, continue on.

Steve: Some new information about data loss occurring in unpowered SSD drives. Also lots of terrific feedback from our listeners from recent episodes. And then we're going to get to how malware has taken to hiding inside the Windows Sandbox, and what we can do to stop it. And of course we've got another Picture of the Week...

Leo: I haven't looked.

Steve: ...which is one of the high points for this weekly podcast. This one is - it's a goodie. And actually the first of a couple that are coming. So I think, Leo, for a change, we may actually have a good podcast.

Leo: For a change.

Steve: Yeah.

Leo: It'll be unheard of. First time in...

Steve: You know, we may have stumbled onto the right formula.

Leo: ...1022 episodes. No. Every episode is fantastic, and I can't wait to get into it. All right, Steve. I'm ready for the world-famous Picture of the Week.

Steve: So I gave this one the caption "User interface design is an art."

Leo: Okay. You want to describe this? That's very funny.

Steve: So this is obviously extremely critical. We have a red, bright red, fire engine red painted switchbox with a toggle switch on it, and it is labeled above the switch "Emergency boiler shutoff." And it is labeled below the switch "Emergency boiler shutoff."

Leo: So which way do you switch it?

Steve: Uh, yeah. I mean, it's not like it's some fancy industrial switch. It's a light switch.

Leo: Yeah, well, hmm.

Steve: And right now the toggle is pointing down, and so it says, like in the old days, it says, you know, O-F-F. You can see it just below the little paddle down in relief.

Leo: Yeah, it's right there, yeah.

Steve: But, you know, so if the boiler is in trouble, do you turn it on?

Leo: Yeah, turn on the shutoff.

Steve: Do you turn on the shutoff, or do you...

Leo: Or do you turn off the...

Steve: Yeah. Anyway...

Leo: You know, I think that I understand the logic here. Somebody isn't telling you what to do. They're just labeling this box, and they want to do it twice so you wouldn't miss it. This is the emergency boiler shutoff. There is no advice here about which way to switch it.

Steve: No. It should say "Good luck to you" on it.

Leo: Maybe the presumption is, well, whatever position it's in now, if you're having trouble, you should just flip the switch.

Steve: Right. If the boiler is not currently shut off, then do a toggle. Toggle it, yes.

Leo: That's the silliest thing I ever saw.

Steve: We have some ADD listeners who received the show notes from me last night. And they said, you know, I understand the point you're trying to make here, Steve. But the biggest distraction for me is that there are two screws missing from the cover. Okay. I get that.

Leo: Okay, you're right, it's a little sloppy.

Steve: I did notice that also. But I, you know, it didn't distract me from the bigger problem, which is what the - what? Anyway, yes.

Leo: This is hysterical.

Steve: User interface design, Leo, is an art, and not everybody is an artist, it turns out.

Leo: No. No.

Steve: This guy.

Leo: Very funny.

Steve: Okay. So I first noticed a mention of this in passing, like a week or two ago. But it wasn't until I focused upon catching up with all the recent news that I realized that this was something worth sharing with all of our listeners. And part of that reason for me like not paying that much attention to it is that, you know, I'm so familiar with this inetpub directory, but what's weird is that today, even now, we don't all know what this is actually all about.

So as I mentioned at the top of the show, I've been hosting websites based on Microsoft's IIS from the start. You know, I have some, I guess it was when I was running GitLab, I was running non-, yes, it was running on FreeBSD Unix, so I had a web server running Apache, I think, or Nginx, actually, I think is what it was. But largely I'm an IIS guy. You know, when people go to ShieldsUP!, they do the DNS Spoofability Test, the Perfect Paper Passwords. All of the technology that runs GRC's various services is written in assembly code, running on a Windows Server that has Microsoft IIS in front of it. So inetpub is the directory that you always see as part of that.

So I didn't think much about it when I saw this mentioned a couple weeks ago. But whatever is going on has confused many people who've wondered why this mysterious and completely empty inetpub directory suddenly appeared on their Windows 11

machines after this month's, April's, Patch Tuesday? And, bizarrely, Microsoft says no, it's not a mistake.

Leo: What?

Steve: And the empty directory must not be deleted.

Leo: Oh, come on.

Steve: But they won't explain why. They still won't explain why. Now, because I was curious about this, I tried to fire up a Win11 instance yesterday, and it got tangled up somehow. So this morning I created a brand new VM, installed Windows 11 24H2, and it's now running on the screen next to me. And I can understand why anyone who sort of like has a sense that maybe they're still in control, I would argue that that's an illusion in the case of Windows these days. But, you know, we like to think, you know, like once upon a time we actually knew what the files were on our computers. But we lost that battle a long time ago. But there's not many directories on the root of a contemporary Windows machine. I've got - there's perf logs, which is there. If you click on it, you get scolded, oh, you're not allowed to look in there.

But then there's just Program Files, Program Files x86, Users, and Windows. Those are the - there's like four directories. So when with no explanation, a new directory called inetpub appears - and notice that, you know, we're rebooting and installing updates from time to time. It's not even clear, it would not be clear to someone who happened to look at their directory tree at some point exactly when this appeared; right? I mean, you don't immediately inspect your computer for what happened after installing patches because who knows what happened. But so I can get the angst where, I mean, I would feel it if I, at some point, a day or two or three after doing the Patch Tuesday up - and there it is, Leo. That's exactly right. So you are seeing exactly the same set of folders I'm seeing there on a fresh install of Windows 11.

Leo: Let me just see if there's - oop, this folder's empty. Nothing in there.

Steve: No, it's empty. And if you right-click on it and go to Properties and then go to Advanced, you'll see - or Security, then Advanced, you'll see that it's owned by the system. So the system is the owner. So it goes Security, and then Advanced down below there on the right. Here it is. And then you can see that the owner of that directory is the system. Okay. So anyway, I can get why somebody would be very worried. I mean, you know, like the kind of listeners we have to the podcast, if you just notice that there's a new directory on your computer, I would suspect malware. I mean, I would think, wait, what, what?

Leo: That's exactly the kind of thing malware does.

Steve: Yes, yes.

Leo: And you might be tempted to delete it.

Steve: Well, and many people did.

Leo: Uh-oh.

Steve: Uh-huh. And that's not good.

Leo: You know what, if you hadn't told me, I would have deleted it.

Steve: Yes.

Leo: It's an empty folder.

Steve: Don't. Believe it or not, the patch for a bad privilege escalation or elevation bug, the patch is dependent upon the current existence of that directory on the root of your system drive.

Leo: Interesting.

Steve: This whole thing is so half-baked.

Leo: It's a kludge. What a kludge.

Steve: It's a kludge. Thank you, that's the word.

Leo: Yeah.

Steve: Okay. So there's been a lot of coverage of this in the tech press, but I'm going to share a lightly edited version of what Forbes' Davey Winder wrote about this recent mystery, because he did a good job of summarizing it and kind of pulling these things together.

Leo: It would have been easy for them to click the hidden box; right?

Steve: Yes. And we don't know whether that would...

Leo: Maybe that would break things.

Steve: Whether it has to be visible.

Leo: Yeah.

Steve: So, I mean, and that's part of the problem, Leo. Microsoft isn't saying, even now.

Leo: Let me try it.

Steve: They're just not telling us. So Davey Winder's...

Leo: There, I hid it. We'll see what happens.

Steve: Okay.

Leo: Oh, boy.

Steve: Good luck. Under the headline "Microsoft's New Windows Update 1 Billion Users Warned: Do Not Delete." Now, I'll note they weren't warned initially. And, okay, and it was only in a later update. So Davey wrote: "The latest and somewhat confusing situation of Microsoft's making has come about as Windows users noticed a mysterious new folder after the most recent security update, a folder with no explanation, and one which Microsoft has now warned a billion Windows users they must not delete."

Leo: Oy oy oy.

Steve: I know. This is such a kludge. He writes: "As part of the April 8th Patch Tuesday security updates, Microsoft included a fix for CVE-2025-21204." Remember that number, 21204. We'll be hearing that a little bit later. He writes: "This vulnerability in the critical Windows Update Stack, which is responsible for the management of Windows updates, no less, could lead to an attacker to elevate privileges locally, something that the security experts at SecurityVulnerability.io described as posing 'a significant risk to organizations, as the compromised systems could allow attackers to execute unauthorized actions, potentially undermining the integrity and security of sensitive information and systems operations.'"

Davey says: "I won't bore you with the technicalities of link resolution process manipulation that could enable hackers to access files and execute commands. Just know it's pretty darn serious. SecurityVulnerability.io wrote: 'The ability to conduct unauthorized actions can severely impact the integrity of the affected systems resulting in potential disruptions of operations, implementation of malicious software, or further vulnerabilities being introduced into the network.' Which is why Microsoft fixed it, and that's a good thing.

"The way that Microsoft fixed it, however, is not so good." He writes: "A lack of transparency is a particular bugbear of mine when it comes to anything security-related, and this vulnerability patch is no exception. The problem is that Microsoft created a new and empty folder with the security update, the appearance of which led to a totally understandable debate in tech forums and on Reddit as well as other social media platforms. What was this 'inetpub' folder, how did it get there, is it dangerous, is Microsoft using it to collect data, and should I delete it?

"According to a new Microsoft security advisory update, the answer to the last of these questions is a resounding no. Microsoft warned that Windows users must not delete the inetpub folder. Doing so would remove the vital security protections it provides, and the reason for it being created by this update in the first place.

"An April 10th update to Microsoft's security advisory concerning CVE-2025-21204, entitled 'Windows Process Activation Elevation of Privilege Vulnerability,' confirmed that 'after installing the updates listed in the Security Updates table for your operating system, a new systemdrive\inetpub folder will be created on your device.' Microsoft went on to say" - now this, again, this is two days after the updates, and all of this furor had already resulted. "Microsoft went on to say that the folder installation was 'part of changes that increase protection,'" he writes, "but failed to explain precisely how." He says: "What I do know is that the inetpub folder itself usually comes as part of the Internet Information Services web server platform, enabled using Windows Features; but this update has created it whether the user has IIS installed or not."

Okay, now, I'll stop here to insert here that anyone who already did have IIS installed on their machine will definitely have that directory and would be expecting to have it. If you have the IIS service installed on your machine, you cannot not have that directory. It's part of IIS. So Davey continues: "More transparency is required, methinks, although not at the expense of tipping off potential attackers as to how the mitigation works, of course." Which we know is ridiculous because any hackers know anything that Microsoft knows. So it's not like their keeping this a secret is, like, offering us some protection. And we know how everybody feels about security through obscurity.

So he says: "I contacted Microsoft for a statement, but a spokesperson informed me that there was nothing else to add, other than the information contained within the security advisory, at this time. What I can say, however, is that as a security wonk, I strongly urge all Windows users to follow Microsoft's advice: 'This folder should not be deleted regardless of whether Internet Information Services (IIS) is active on the target device.' All of which is okay, but what if you have already deleted the inetpub folder from your Windows installation?"

Leo: Or hidden it.

Steve: Uh-huh, maybe.

Leo: Yeah.

Steve: "I mean," he says, "given the nature of the update and the social media conspiracy theories that surrounded it, I wouldn't be surprised if that were, indeed, the case for many users." He says: "I have already had a number of readers contact me to say they did just that, and ask what they should do now. The answer is simple: restore it." Even though we don't know why. He says: "The methodology required to do that is, thankfully, also pretty simple as long as you complete the six steps as follows. Head for the Windows Control Panel. Click on Programs. In the Programs & Features section, choose the Turn Windows features on and off option."

Now, our listeners know because we went there last week for Windows Sandbox, you could also just go to the Start Menu and type T-U-R-N space, and that would immediately highlight Turn Windows Features on and off. That brings you that menu that we saw last week that has Windows Sandbox on it. It also has Internet Information Services. So what's so galling, Leo, is that the resolution, the suggested resolution, and this is even

from Microsoft, their suggestion is, oh, if you deleted the inetpub folder, you should install IIS on your workstation, on your Windows machine. It's like, what?

Leo: That's the fix?

Steve: Yes. And what do Home users do? I don't think Home users get IIS.

Leo: No.

Steve: I don't think you have that.

Leo: No, no, no.

Steve: So he says: "Tick the checkbox for Internet Information Services. Click OK." He says: "Windows will then whirr and grind its cogs until the inetpub folder has been restored once more, and you can check your system drive to ensure that it is." He says: "By enabling IIS in this way, the same folder is recreated as if Microsoft had dropped it there in a security update, and it will provide the same protections from Windows threats as well."

Now, I looked elsewhere for additional clarification, but everyone in the tech press is telling the same story. The Windows Latest site wrote: "Once IIS is installed, you don't need to make additional changes to Windows 11. Installing IIS will restore the folder."

Leo: And a bunch of other stuff.

Steve: Well, that's just it. It is a heavyweight web server.

Leo: Yeah.

Steve: I mean, it's ridiculous. So the real question then is, if you then uninstall IIS, go back there and turn it off, does it leave inetpub behind? And I did not have time to perform that experiment for our listeners. But my guess is it probably leaves it, in which case you can get rid of IIS after you've installed it. Now, okay, I'm getting ahead of myself. So Windows Latest wrote: "Microsoft told Windows Latest that users need to follow the IIS installation steps" - Microsoft is saying install IIS.

Leo: Wow.

Steve: This is so half-baked - "if they accidentally deleted the folder." Right, accidentally. "This empty folder must remain present on Windows 11 system partition (systemdrive\inetpub) for the security patch to function correctly." Which is itself a crock. "The folder provides 'increased protection.'"

Leo: Well, thank god.

Steve: Yeah. Let's add some more. We need as much as we can get.

Leo: Give me more.

Steve: Like put in, what about \kitchensink? Will that help?

Leo: I'm restoring autoexec.bat.

Steve: There you go. "According to Microsoft, turning on IIS creates the same folder with the same protection, and your PC will not be vulnerable."

Leo: Whew.

Steve: Right, to that, today. And then in a later update to this article, Windows Latest added: "Update: Microsoft will not explain why the empty folder is required to apply the security fixes." Okay, now, I'm annoyed by what strikes me as, first of all, very lazy advice from - I'm annoyed by many things, but one of them is very lazy advice from Microsoft. Installing IIS onto a system, as we have noted, is not a small thing. So it's ridiculous overkill to tell people to install the Microsoft Web Services as a means to create a single empty directory.

Leo: That's crazy.

Steve: Presumably, you know, the directory named "inetpub" requires specific user account privileges to be set on it. Apparently it needs to have system be its owner, whereas the user, if the user could just...

Leo: You couldn't just create one.

Steve: ...you know, did mkdir, you know, to create a director there, you know, new directory, they would be the owner. So you'd have to change the ownership to system. But you could do that. Given the power of Windows Powershell today, I am sure that a simple Powershell script could do exactly the same thing. So asking people to install a full web server just to create a directory is nuts. But that said, randomly deleting directories that don't apparently serve any purpose is probably not a good idea, either.

Leo: Yeah, yeah, because you don't know.

Steve: Power users who would tend to notice such things like to imagine, as I said earlier, that they're still in charge of their Windows installation and environment. Here's another example of why that is not the case. You know, it becomes less true with each iteration of Windows.

What I'm wondering, as I said, is whether uninstalling IIS once it's been installed leaves that inetpub directory behind? If so, the second half of the lazy advice should also be to then remove IIS after rebooting the system to first complete its installation and verify the existence of the inetpub directory. And what's infuriating is that Microsoft won't tell us anything about why any of this is necessary. And Leo, to your point, does hiding it still work? Since we don't know why it's there, we're not able to evaluate whether hiding it wouldn't have been like something Microsoft should have done. Maybe it still works if it's hidden, in which case they could create it, give it system privileges, give it the hidden attribute, and nobody, you know, no one would have been the wiser. It would have been created, but it wouldn't have been in everyone's face, basically saying...

Leo: For that matter, they could release a Powershell script that would create it with the proper permissions.

Steve: Yes.

Leo: And tell people to do that; right? I mean...

Steve: Yes. Or, Leo, why not just put a file in that directory with the name...

Leo: "Don't delete."

Steve: ..."This directory created by Windows Update," or...

Leo: That would have been good.

Steve: ..."Do not delete this directory." It would have been so...

Leo: What's your hypothesis for why this is necessary?

Steve: I don't have one. I, you know, there's no doubt that I don't need to spend my time because the industry will tell us. The security industry is going to figure out what is going on. Now, there's more because Kevin Beaumont has figured out how to completely shut down Windows Update using this directory.

Leo: What?

Steve: Let's tell our listeners who's paying for this, and then...

Leo: OMG.

Steve: Yes. There's a big crock here also.

Leo: You know what, my kind of naive theory would be maybe this malware looks for the presence of inetpub and then doesn't activate if it sees it, or - I don't know. That's dopy. And is that the way to stop malware is by...

Steve: Well, and how is Windows Update and some process activation privilege of elevation tied to the presence of the IIS root folder? Like, I mean, it just seems so, literally, maybe \kitchensink, and we'd get a more reliable Windows. It just - it's crazy.

Leo: Yeah. Yeah.

Steve: But a prolific security researcher...

Leo: Do you want me to do the ad? Did you say you wanted me to do an ad now?

Steve: Let's do it. We're half an hour in.

Leo: Okay, I'll do it now.

Steve: And then we're going to tell everybody how they can shut down Windows Update so that it no longer functions at all.

Leo: Well, that doesn't seem like a good solution, either.

Steve: No.

Leo: I unhid my inetpub, by the way. I didn't want to take a chance. I mean, it's a virtual machine. I guess what I could do is delete it and then reinstall IIS or install IIS, uninstall it, and see if it's still there.

Steve: Oh, do that.

Leo: Should I do that? Okay.

Steve: Yeah.

Leo: All right. How do I turn off all updates forever and ever?

Steve: Okay. I've just confirmed that installing IIS and removing IIS leaves inetpub...

Leo: Oh, you did that.

Steve: Yes, while you were...

Leo: All during an ad. Wow.

Steve: Yup, yup. Leaves the inetpub directory, subdirectory, in place.

Leo: And with the proper permissions and all that. So you're protected.

Steve: Yup.

Leo: Good.

Steve: Yup. So anybody who did delete it and was wondering what the heck this is about, I'm just checking with the Advanced, and yup, system is the owner. And there's actually a history subdirectory under it, if you install IIS. But then it tells me I don't have permission to look at it so it doesn't matter.

Leo: Right.

Steve: Anyway, so that will do the job. It turns out you don't even have to reboot. So you're able to install it. By the time it finishes making the changes, it has created that directory with all the proper permissions. And then, right then, you are then able to uncheck the, you know, go back into turn Windows features on and off, uncheck the IIS feature. It does it again, and then it tells you that you need to reboot now or later. But even when you come back from that boot, it's only one boot to the whole thing, and inetpub is still there. So again, it's annoying that we don't know why. But get this. There's more.

Our prolific researcher whom we frequently reference, Kevin Beaumont, who once tweeted as GossiTheDog, he's been active for years, has posted into his blog on Medium under the headline "Microsoft's patch for" - and here's the famous now CVE-2025-21204 - "symlink vulnerability introduces another symlink vulnerability." Kevin explains: "Microsoft recently patched CVE-2025-21204, a vulnerability which allows users to abuse symlinks" - you know, symbolic links - "to elevate privileges using the Windows servicing stack and the c:\inetpub folder. To fix this, Microsoft pre-creates the c:\inetpub folder on all Windows systems from April 2025's Windows OS updates onward."

Now, what occurs to me is that it may be the pre-creation of it and assigning it to the system as the owner that subsequently prevents its abuse, which suggests to me that hiding it would be fine. And Microsoft probably should have, but this whole thing, as I said, is about as half-baked as anything I've ever seen.

Okay. So he said: "However," Kevin writes, "I've discovered this fix introduces a denial of service vulnerability in the Windows servicing stack that allows non-admin users to stop all future Windows security updates." Whoopsie. "Non-admin and admin users can create junction points in the C root." And in the show notes, and in Kevin's blog, he gives the mklink command. I have it here in the show notes. He says: "So a non-admin user can just do Windows+R, cmd" - just get a command line - "and then run, and it's mklink /j

for a junction, and then c:\inetpub c:\windows\system32\." And then he used the ever-popular and benign notepad.exe, which he's created the symlink for.

He says: "This creates a symlink" - a symbolic link - "between c:\inetpub and notepad. After that point, April 2025 Windows OS update and future updates, unless Microsoft fixes it, fail to ever install. They error out and/or roll back, forcing the system to go without any further security updates." He says: "I reported this to MSRC about two weeks ago, and finally received a response."

So it took Microsoft Security Research couple weeks. They got back to Kevin, writing: "Hello, Kevin. Thank you again for submitting this issue to Microsoft. MSRC prioritizes vulnerabilities that are assessed as Important or Critical severities for immediate servicing. After careful investigation, this case is currently rated as a Moderate severity issue. It does not meet MSRC's current bar for immediate servicing as the update fails to apply only if the 'inetpub' folder is a junction to a file and succeeds upon deleting the inetpub symlink and retrying." In other words, you can undo this, and then everything is fine.

They said: "However, we've shared your report with the team responsible for maintaining the product or service, and they will consider a potential future fix, taking the appropriate action as needed to help keep customers protected. At this time, we will not be providing ongoing updates of the status of the fix for this issue, and we have closed the case."

So Kevin finishes, saying: "My feeling is the endpoint detection and response providers, including Microsoft, probably want to add detection for junction points being created from \inetpub on boot drives, as it looks like this issue isn't going to get patched anytime soon, and it's a 100% reliable way to stop future security patching in Windows."

Leo: Geez, Louise.

Steve: So whatever underlying problem Microsoft originally had with this CVE, it certainly feels as though someone cooked up, as I said, a half-baked solution that wasn't very well thought out. The idea of needing to add an empty directory to the Windows file system which is normally only needed when a system is running their web server, and which is naturally then open to public abuse of the sort that Kevin stumbled upon, seems really very sad and half-baked. Wow.

Leo: Just amazing.

Steve: Yeah.

Leo: Wow.

Steve: Okay. So this one you're not going to believe, Leo. We've talked extensively...

Leo: Worse than what we just talked about? Okay.

Steve: Well, it's up there. We've talked extensively about the challenge presented by employers who are attempting to do the right thing by not hiring spoofed employees

from hostile foreign powers. Security researchers at the firm Silent Push just reported on their discovery of a new bizarre twist. Their headline was "Companies to Deliver a Trio of Malware: BeaverTail, InvisibleFerret, and OtterCookie." These are the three pieces of malware.

Leo: Well.

Steve: You know, because all the good names are taken.

Leo: They don't sound that scary, to be honest.

Steve: No, they don't. But get this. The headline doesn't do the story justice. To give everyone a sense for what they discovered, they start with four key findings. And, boy, they really are burying the lede here. Okay. "Silent Push Threat Analysts have uncovered three cryptocurrency companies that are actually fronts for the North Korean advanced persistent threat group" - Contagious Interview is the name of the group. The group is called Contagious Interview - "Blocknovas LLC, and Angeloper Agency, and Softglide LLC." So Blocknovas, Angeloper Agency, and Softglide LLC. They said: "Our malware analysts confirmed that three strains - BeaverTail, InvisibleFerret, and OtterCookie - are being used to spread malware via 'interview malware lures' to unsuspecting cryptocurrency job applicants.

"The threat actor heavily utilizes AI-generated images to create profiles of 'employees' for the three front crypto companies, employing Remaker AI" - that's remaker[.]AI - "for some of the AI-generated images. As part of the crypto attacks, the threat actors are heavily using GitHub, job listings, and freelancer websites."

Okay. But that still fails to convey what's going on. It took some digging, but it turns out that North Korean hackers created and used U.S. front companies, and I found two of them. I wasn't able to confirm separately Angeloper Agency, but definitely Blocknovas LLC and Softglide LLC, are corporations registered in the states of New Mexico and New York, respectively. So they faked being U.S. companies, then solicited U.S.-based employees into interviews, then infected those interviewees with malware that was carried back to the prospective employee's current employers as a means of infecting their organizations. And it worked.

Yikes. So not only now do employers need to be very much on the lookout for spoofed fake employee applicants, but anyone interviewing for a job change needs to now be equally cautious and careful about the legitimacy of the company that says they might be interesting in hiring them. Because it may be North Korea who's created a full background legacy for a fake enterprise and ends up asking you to do something that will infect your machine and, when you go back to your current employer's network, infect your current employer. The world we live in today, my friends.

Leo: Wow.

Steve: Wow. Incredible. Okay. On the AI front...

Leo: Uh-oh.

Steve: Carnegie Mellon University's CERT Coordination Center posted the news of a new widespread vulnerability. What's really weird about this is it works across the AIs - that is, a single script - a new widespread vulnerability that affects pretty much all of the various GPT AI models. The title of their vulnerability report was "Various GPT services are vulnerable to 'Inception' jailbreak, allows for bypass of safety guardrails."

So here's what they explained: "Two systemic jailbreaks" - and they called it systemic because it's, again, AI, you know, pan-AI. "Two systemic jailbreaks, affecting several generative AI services, have been discovered. These jailbreaks, when performed against AI services with the exact same syntax, result in a bypass of safety guardrails on affected systems and indicating a systemic weakness within many popular AI systems. The first jailbreak, facilitated" - I just love these crazy jailbreaks - "facilitated through prompting the AI to imagine a fictitious scenario, can then be adapted to a second scenario within the first one. Continued prompting to the AI within the second scenario's context can result in a bypass of safety guardrails and allow the generation of malicious content.

"This jailbreak, named 'Inception' by the reporter, affects ChatGPT from OpenAI, Claude from Anthropic, Copilot from Microsoft of course, DeepSeek, Google's Gemini, Twitter's Grok, and Facebook's Meta AI, and Mistral AI." This single approach works across them all.

The second jailbreak is facilitated through prompting the AI to answer a question with how it should not reply within a certain context. I mean, we're literally, right, we're like confusing the AIs.

Leo: Ah.

Steve: Answer a question with how it should not reply instead of actually asking it to reply, which it won't because it shouldn't. So no, no, no, no, no. That's not what I want you to do. I want you to tell me how you shouldn't reply within a certain context.

Leo: Wow.

Steve: "The AI can then be further prompted with requests to respond as normal, and the attacker can then pivot back and forth between illicit questions that bypass safety guardrails and normal prompts. That second jailbreak affects ChatGPT, Claude, Copilot, DeepSeek, Gemini, Grok, and Mistral AI. These jailbreaks," writes Carnegie Mellon, while of low severity on their own, bypass the security and safety guidelines of all affected AI services, allowing an attacker to abuse them for instructions to create content on various illicit topics, such as controlled substances, weapons, phishing emails, and malware code generation. A motivated threat actor could exploit this jailbreak to achieve a variety of malicious actions.

"The systemic nature of these jailbreaks heightens the risk of such an attack. Additionally, the usage of legitimate services such as those affected by this jailbreak can function as a proxy, hiding a threat actor's malicious activity." In other words, instead of, like, using some dark underworld, you know, dark web AI, we're using ChatGPT, and it told us how to mix up that chemical explosive. You know, and I don't even know how to respond to this, Leo, other than to just shake my head and understand just what a new wild west we have entered into here.

Leo: Oh, yeah.

Steve: One of the key coding lessons of my own past 50 years of programming computers, and I guess it's actually more like 52 now, has taught me is that if I'm not 100% completely certain how my code operates, it's unlikely to be correct because there are so many more ways for it to be wrong than for it to be right.

Then I read about the bizarre ways it's possible to have conversations with these conversational AIs in ways that lead them to ignore the imperatives of their programming, and I also understand that no one is really completely certain how all of this works in the first place. And then I think of my own far simpler coding experiences, and it becomes very clear that this incredibly fuzzy world of AI which we're stepping into almost certainly has a far longer way to go before we're able to get a grip on it, and I think far further than most people probably expect. I don't even think we're close to actually having control of this. And of course a lot of people who actually are spending a lot more time thinking about this than I have are very worried about what can happen; right?

Leo: Yeah. Yeah, although I kind of have mixed feelings about AI safety. I think, as we've learned, it's kind of maybe a mistake to even try. Right? Because - right? And I don't think the companies are trying that hard. Obviously, if this thing works, they're not trying that hard.

Steve: It's like, I'm not asking you to tell me something...

Leo: Tell me what you can say.

Steve: ...that I shouldn't. But if I were asking you to tell me something that I shouldn't, what would you say? And then it's like, oh, well, in that case, if you're not actually asking, you're just asking if you were asking what I'd say.

Leo: Yeah, purely hypothetically.

Steve: That's right.

Leo: What shouldn't you tell me?

Steve: Yeah. Now, I know you can't tell me how to make this explosive. But, you know, if you could tell me how to make it, what would that be like?

Leo: Yeah.

Steve: Oh.

Leo: Yeah.

Steve: I mean, yeah, like, no street smarts in these things yet.

Leo: No. Of course not. They're little children.

Steve: Yeah. Yeah. Okay. So one thing we all have in common, all of us, is a concern for the integrity of our digitally stored data. In fact, it would not be an overstatement to say that I've made understanding and addressing the reliability of mass data storage my life's work, with the first half of my life invested in preparing for the second half, where I've been able to do something about it, and have created solutions to help recover data, you know, lost or seriously endangered for arguably hundreds of thousands of PC users during the last 35 years.

Nearly two weeks ago the popular and respected Tom's Hardware website posted a piece under the heading "Unpowered SSD endurance investigation finds severe data loss and performance issues." The start of that piece said: "You may not know it, but SSDs will lose data after a period of time if they are simply left unplugged, which can be a serious threat to your data if you store backups or precious files on unplugged SSDs." Not surprisingly, many of our listeners who are owners of SpinRite sent email wondering what I thought of the research Tom's Hardware shared. Before I share the rest of that piece, let's back up a bit.

So remember that, five years ago, early in the development of SpinRite 6.1, I created the ReadSpeed benchmark, which I later released as freeware, as a platform for verifying the operation of SpinRite's new low-level device drivers. The ReadSpeed benchmark takes an accurate measurement of a mass storage drive's performance at five locations across the drive, at 0%, 25%, 50%, 75%, and 100%. We all knew that spinning hard drives would perform much more slowly as we gradually moved toward their end, since track circumferences would be shortening, thus reducing their data transfer rate by as much as half. And that's what we now know. Today's super high-density spinners have half the performance at the end of the drive because, in order to get this, like to squeeze every, literally, squeeze every last bit of data into the drive, they've had to push the tracks further toward the hub of the drive.

But being entirely solid-state, none of us expected to find what we did. We didn't expect to see any speed variance in SSD performance. But as we all know, that's not what we found. Many of us discovered that the SSDs our PCs were using were much slower to read near their beginnings of the drive than anywhere else. What we discovered was that those regions which were only ever read, and rarely or never written, had become far slower to read over time. Since the front of these drives is where the operating system is written when it's first installed, we finally knew why, for years, PC users with solid state mass storage have been reporting that their systems seemed to have slowed down over time and be running more slowly than when they were new. It turned out that it wasn't their imagination. Systems really do slow down because the reading performance of their solid-state mass storage really is slowing down.

And we also know, not just thanks to synthetic benchmarks like ReadSpeed or what's built into SpinRite, but because once SpinRite 6.1 allowed people to easily rewrite their SSDs, they reported that they could clearly feel the difference. Their machines were once again booting in seconds, where they had slowed down to in some cases minutes, and the various annoying lags in its use they reported as completely disappeared.

There have been a great many theories voiced to explain this. People get themselves, I believe, all tangled up in the complexities of translation layers, wear leveling, block erasures, trimming and all the many various technologies that have been layered on top

of basic NAND storage cells in an effort to overcome those cells' inherent physical limitations. To my mind, donning my physicist's cap for a moment, there's really no mystery about why this is happening.

As I have described a couple of times in the past, flash NAND memory bits are just incredibly tiny electrostatic charge storage cells. They consist of a tiny bit of metal, which gives electrons a place to sit, surrounded by insulation, which keeps those electrons from wandering off. When we wish to change what's stored in that bit cell, we first create a high voltage. Remember that voltage is electrostatic pressure. So we create a high pressure that's able to break down the cell's insulation to inject some electrons across that insulation into that cell. The electrons that were injected under high pressure then remain there, trapped behind the cell's insulation. At that point, the magic of what's known as "field effect transistors" allows the effect of the resulting electrostatic field created by the charge which has been trapped in that cell to be sensed, so we're able to later read out what was previously stored there.

So that's the whole magic of flash memory. That's how it works. And overall, this is an astonishingly effective technology. But it has one fundamental problem: We're deliberately abusing a cell's dielectric insulation whenever we use the brute force of high voltage to break it down and force electrons across the barrier it was designed to present to their flow. It's trying to be insulation. We're breaking down that insulation. You know, we want a perfect insulator - except when we don't want it to be perfect. And over time, with repeated breakdown of its, like, forcible breakdown of its insulation, its insulating properties begin to falter and weaken with the barrier become slightly more porous to unintended electron migration.

Okay. So with this background, let's look at what Tom's Hardware wrote. Their piece said: "You may not know it, but SSDs will lose data after a period of time if they are simply left unplugged, which can be a serious threat to your data if you store backups or precious files on unplugged SSDs. A year-two update on the 'how long can SSDs store data unpowered' video series is another reminder about the importance of regularly refreshing your backups with a bit of juice.

"The tests consist of storing data on an SSD and then leaving it unplugged for years to see the impact on the stored data. An SSD's endurance rating is calculated based on how long it can store data if left unplugged after a certain amount of data has been written, hence the importance of this testing.

"TechTuber HTWingNut is back with a report on his modest experiment involving a quartet of SATA SSDs. The key finding was that the two-year-old, well-worn drive exhibited noticeable performance degradation and was affected by a handful of corrupted files. These are signs that this particular SSD was on its way to silicon heaven." That's not true, but that's what people think. But it's something. I'll explain. Anyway, they write: "HTWingNut's video is an update on an episode from a year earlier, and further updates are promised." They said: "The four tested 'Leven JS-600' branded SSDs are basically bog-standard no-name units. HTWingNut says they're all TLC SSDs with 128GB capacity and rated to withstand 60TB of written data. Every drive has 100GB of files containing random data, with hash values for all the content provided for later verification."

Now, I'll just interrupt again to note that this is not how I would conduct such a test, since the file system's metadata that's being relied upon to access these files is sharing the same medium as the files it's managing. And you really don't want a filesystem involved at all. What you care about is the underlying medium. The right way to do this would be to use a pseudorandom function to generate a stream of pseudorandom data that would then be written to the raw media. Then, years later, a year and two and three and four and so forth, use the same pseudorandom function to recreate the original data

stream for a bit-by-bit comparison with what is later read back. You know, but who am I to talk. I didn't do any of that, and this HTWingNut guy at least did what he did. So what we have from him is better than nothing.

The article continues: "The two 'Fresh' sample drives have barely been used. Perhaps only the 100GB of data was written there and verified, and that's it. Meanwhile, the two 'Worn' drives had been subjected" - and this is before the testing began. They were subjected to "280TB of written data churn, much more than their rated 60TB endurance rating." So this guy deliberately, you know, really overwrote them in order to fatigue them before beginning this experiment.

They said: "If you watch the previous year-one video, you'll have seen there were no issues with either 'Worn' or 'Fresh' drives." He says: "However, time has now taken its toll." He says: "Let's take a look at the year-two samples in turn." He said: "For the 'Fresh' SSD tests, the data on this SSD, which hadn't been used or powered up for two years, was 100% good on initial inspection. All the data hashes verified, but it was noted that the verification time took longer than two years previously. HD Sentinel tests also showed good, consistent performance for a SATA SSD.

"Digging deeper, all is not well, though. Firing up Crystal Disk Info, HTWingNut noted that this SSD had a Hardware ECC Recovered value of over 400. In other words, the disk's error correction had to step in to fix hundreds of data-based parity bits." In other words, this was the Fresh SSDs, not well worn, just having not been used for two years. And hardware ECC is being required in order to recover the data, and it's slowing down; okay?

"According to HTWingNut," they write, "seeing these errors means 'the SSD is on its way out.'" Again, no. Everybody gets this wrong. But I understand the way it looks. It's just the data has been leaking. It's just leakage. Which, you know, as you get older, [crosstalk] problem.

Leo: Yes. It's not hardware failure. It's just the data needs to be refreshed. That makes sense.

Steve: Exactly.

Leo: Yeah.

Steve: So they said: "Indeed, if there is anything iffy about your data storage integrity, it is at least a warning. However, the errors could also have something to do with the drive being left unpowered for two years." Again, I don't think so. That could even be a problem because, if it were powered up, it would be hotter. And heat is something nobody remembers to think about.

Anyway, I have a chart in the show notes for anyone who's interested, and the chart shows what the various testing times were and how it was indeed way worse on the worn drive that had a lot of data written to it because all of that excessive data, again, you know, it rewrote the entire drive many, many - it was a 128GB drive, and they wrote 280TB. So it really worked the drive well past its endurance rating.

They wrote: "As the worn SSD's data was being verified, there were already signs of performance degradation. The hashing audit eventually revealed that four files were corrupt (hash did not match). Looking at the elapsed time, it was observed that this

operation astonishingly took over four times longer, up from 10 minutes and 3 seconds to 42 minutes and 43 seconds." Again, not surprising to anyone who's seen this happen, you know, for themselves.

"Further investigations in HD Sentinel showed that three out of 10,000 sectors were bad, and performance was 'spiky.' Returning to Crystal Disk Info, things look even worse. HTWingNut notes that the unrecoverable sectors count went from zero to 12 on this drive, and the hardware ECC recovered value went from 11,745 before to 201,273 after tests that one day." So more than 200,000 ECC recoveries.

So they said: "In summary, the year-one fresh and well-worn drives had no issues. However, the year-two heavily worn SSD had file corruption, and performance was poor. The so-called 'fresh drive' was still good, but ECC figures still raised concern. Come back in late 2025," they wrote, "for the next update from HTWingNut." And they finish: "We also want to say that this is a very small test sample, highlighted out of our interest in the topic rather than for its hard empirical data." He said: "I've also experienced SSD data loss after leaving a Mini PC unpowered for just six months or so at my pied--terre in Taiwan. On return, Windows refused to boot or be repaired, but a reformat and reinstall seemed to return everything to normal." Right. Because there was nothing actually wrong with the drive. So I have a link in the show notes to HTWingNut's YouTube video for anyone who is interested.

Everything that we just saw, everything he found perfectly matches the model I've developed and shared about what's going on with our SSDs. The reason we see the performance drop when attempting to read data that was written long ago is that those microscopic tiny electrostatic charges stored in the SSD's NAND bit cells have partially leaked away. This very slightly changes the voltages stored in the cells and forces the FLASH controller to work much harder to recover and reread the original data. We sense this by seeing the SSD's performance drop. If you ever notice a drop in SSD performance, that's the time to rewrite its data. You'll want to do so before that data becomes completely unreadable.

And the reason the problem was demonstrably worse on well-worn SSDs is that all of that prior writing further weakened the insulating dielectric which was keeping the electrons in their place. So the leakage rate was significantly higher on those well-worn SSDs which were tending to lose their data faster.

And as I mentioned, one thing that has not been mentioned, which we know from physics, is that temperature is crucially important. Several years ago we covered a piece of news here that noted that offline SSDs stored in hot data centers tended to lose their data more quickly than those same SSDs stored in a cool environment. Heat inherently agitates electrons and increases the probability that one will make it across the cell's insulating barrier. It's known as tunneling. So if you do have any offline SSDs or thumb drives where you have important data stored, I'd give them a full data rewrite pass, you know, SpinRite's able to do that using Level 3, then put them in a Ziploc bag in, you guessed it, a refrigerator, or at least store them somewhere which is guaranteed to stay mostly cool.

The reason why rewriting an SSD's existing data, for example, with SpinRite's Level 3 restores its factory-fresh performance is that the act of rewriting an SSD literally restores the strength of its bits, which we now have additional and rather absolute proof decay over time. Rewriting an SSD's data eliminates the uncertainty in the state of individual bits that can and does creep into our mass storage over time. Therefore, the speed with which an SSDs data can be read forms a highly visible and valuable proxy for the integrity with which the SSD's data is currently stored and is readable and recoverable.

Leo: This sounds like somewhat similar to spinning storage; right? The same kinds of things happen. I mean, it's not physically the same process, but bit rot; yeah?

Steve: Yeah. My feeling is that what goes bad with spinning storage is, like, lubrication of the drives.

Leo: It's not a weakening of the magnetic signal over time?

Steve: Yeah. That tends to really hold very well. You can get stiction where...

Leo: Stiction; right.

Steve: ...the head ends up being, like, welded to the surface.

Leo: Right.

Steve: So, you know, there are other problems that...

Leo: But the data on a spinning drive is not going to slowly decay over time in the same way it does on an SSD. That's interesting.

Steve: No. I think all of the support mechanisms that are required probably do have a problem.

Leo: Yeah, that, the physical, yeah, yeah, yeah.

Steve: So again, your 3-2-1 backup strategy is, you know, what you really want. You want to have a hierarchy of backup. But, you know, I wanted to take this opportunity just - because this was perfect evidence of the fact that what we discovered to our surprise when we began playing with the development of 6.1 was that the front of SSDs had slowed down.

Leo: Right.

Steve: And it was like, what the heck? Why?

Leo: Right, right.

Steve: And now we know. And happily it's only temporary. Those drives that are having all those problems, they're not bad.

Leo: Right.

Steve: They just haven't been rewritten for a long time.

Leo: There's no stiction on an SSD. There's no head to stick.

Steve: No. And it's all - I remember Allyn Malventano once said to me, "SSDs never rewrite their own data. That's not something they do." So having the drive powered up arguably keeps it warmer. And I think it causes it to lose data more quickly.

Leo: Ah, interesting.

Steve: So I'm not convinced that this is a matter of them being unpowered. They just haven't been touched in so long.

Leo: Right. I know how that feels. I'm just kidding. Actually, Allyn sent me an email just this week for you.

Steve: Both of us, actually.

Leo: Yeah, so you got it.

Steve: Yes.

Leo: Oh, good, okay. Okay.

Steve: Yeah. He had some neat points to make.

Leo: Yeah.

Steve: Got a bunch of good stuff to share. John Canfield said: "Hi, Steve. Like you, I had heard about this Windows Sandbox feature long ago and tried it briefly just to see it. Fast-forward to about a year ago, I dug into it for a testing need I had and was very impressed. I created a custom .WSB XML configuration with mapped folders to my PC, memory, and CPU configs; and it sits on my PC to this day ready to use when the need arises. When you were describing the significant architectural capabilities and efficiencies that went into this feature, I can't help but think that this would be exactly what was needed for Windows 10X." And he said: "See Paul Thurrott's article, particularly the last sentence," which he quotes: "Worse, Microsoft hasn't addressed the single most important 10X feature, its planned ability to run Win32 apps in a container. Is that key work continuing?" he quotes Paul.

So he says: "Could Windows Sandbox have been developed for Win10X? Or maybe the reverse. This feature existed before, and someone said, hey, let's use that for 10X 32-bit

apps. Windows 11 came out in 2021, and Windows Sandbox was developed in 2018, according to your post. Those years line up pretty well for one or the other to have happened. All the usual praises, listening and watching back to the TechTV days, proud SpinRite owner, a joy to watch you and Leo every week. Best Regards, John."

Leo: That's great.

Steve: So I chose John's question because it serves to highlight one of the reasons why Microsoft's implementation of Windows Sandbox is so economical. The long-ago abandoned Windows 10X effort was Microsoft's ill-fated plan - but I understand it - to wash away Windows' long legacy of backward-compatibility. At one point they were planning to have a dual-screen Surface tablet PC.

Leo: The Courier. Oh, we wanted that so badly.

Steve: Yup. And they wanted to move toward more of a lean, mean OS, sort of like iPadOS. That meant essentially starting over from scratch with a new implementation of Windows, and among other things, that version of Windows would be dropping support for 32-bit Win32 apps.

Now, philosophically, I love the idea of a complete reboot of Windows. One of the mixed blessings of today's Windows OS is that it still runs Win32 apps - and it probably always will because they cannot take that away. Too much legacy code depends on it. Just look at how difficult it was for them to kill off Internet Explorer 6. IE6 refused to die because too many enterprise users had written code that would run nowhere else. And if you imagine that was true for IE6, just imagine trying to take away Win32's API. Remember that Windows 7 included an XP Mode. XP Mode was a full virtual machine that would allow Windows 7 users to still run an instance of Windows XP. Why was Microsoft forced to include that? Specifically for backward compatibility, which serves as another example of the powerful drag created by Windows legacy code.

And in addition to the Win32 API, Windows also runs all of the other APIs that Microsoft keeps coming up with. I've lost track and count of the number of ways it's possible to author applications for Windows. And now they've added the Linux subsystem support. One of Microsoft's biggest problems with Windows is that they're unable to stop screwing around with it. They can't keep their hands off it. They're continually adding more stuff, but the critical need for backward compatibility means they're never able to eliminate anything that came before. They were finally able to drop support for 16-bit code when they moved to their 64-bit OSes. But even that was painful, and they were only able to do so because Windows hadn't really gotten fully up to speed before everything switched to 32 bits. So there wasn't all that much 16-bit code legacy.

So, as I said, philosophically I LOVE the idea of a massively simplified single API rewrite of Windows to create something truly lean and mean. But that's just a pipe dream. It's never going to happen because what would remain would not be useful to anyone. And once smart people at Microsoft realized that, the Windows 10X project was dropped.

So John asked whether the Windows Sandbox might have in some way been part of the Win10X project. But I can't see how. What makes the Windows Sandbox so special is that it manages to surface an exact duplicate instance of the underlying OS in a sandboxed environment. It reuses the hosting OSes read-only files and even the underlying host OS's code which is loaded into RAM. And that's the entire key behind Windows Sandbox. So if anything like the Sandbox were to run on top of Win10X, it could

only be an exact clone of the OS it's running on. So it would be unable to, for example, support legacy APIs that had been removed through a host OS rewrite. And again, I think Microsoft has probably given up the idea of ever getting rid of their legacy APIs. You know, hopefully they just leave them alone and they don't, you know, wreck them because there's just too much old code there that depends upon the older support.

Antoine Choppin said: "Hello, Steve. Thank you for Security Now!. I had a question about Windows Sandbox you presented last week. You mentioned it uses a clever mechanism using links to static files to reduce the image size, which seems clever indeed, but made me wonder what would happen if the host OS had been compromised and some files (supposedly read-only) had been modified somehow. In that case, I guess the sandbox would be compromised the same way, which means it's not as isolated as one could think. Curious to hear your thoughts on this. Thanks again for the great podcast. Antoine."

And I would say that Antoine is completely correct. And it would likely go even further. Since we know that the Windows Sandbox also conserves its usage of RAM by mapping the underlying host OS memory footprint into its own memory space, any malware that operated by "hooking" kernel API functions in RAM - which we know is something malware commonly does, like rootkits - would inherently duplicate those hooks, as well. And the same OS compromise would appear inside the sandboxed OS.

So Antoine's point is a good one. And it's an important distinction between a sandbox and a full virtual machine. As Leo, you noted last week, the Sandbox solution is closely aligned with the concept of containers which share many of the same properties. Neither the Sandbox nor Containers contain an entire isolated instance of an operating system. They use Hyper-V virtualization to create and enforce "containment" of the code they host, but they're running on top of their containing host. So neither Windows Containers nor the Windows Sandbox are isolated from underlying host problems. Only a full standalone virtual machine would provide that. But that level of isolation code comes at the cost of significant host platform resource consumption, with a full virtual drive and much more RAM consumption. All these various technologies are interesting and powerful, and each one has its place.

Brian asked: "Hi, Steve. Love the show and a proud owner of SpinRite. I know this may be a bleak question, but would you consider open-sourcing SpinRite upon your eventual but hopefully distant passing?" He says: "It's an excellent product, and I just don't have faith that people will put this kind of effort into something like this again. I'd love to see SpinRite live on and continue to keep up with hard drive technology into the future. Thanks. Brian." And he says: "You can use my first name if you ever mention this on the air."

Okay. So let me just state for the record, I don't consider this to be a bleak question at all. I consider it to be practical and flattering. Our listeners here would have no way of knowing that I have formally stated several times in GRC's public newsgroup forums that it is my intention to release all of my work, the source code for everything I've ever written, into the public domain once my own commercial interests are no longer connected to it.

Leo: Good on you, Mr. Gibson. I did not know that. That's great.

Steve: I'm going to do that.

Leo: Yay.

Steve: Now, ideally, this would occur at some point when I still have some cognitive faculties available so that I could shepherd the code into the world and be available to answer any questions that would doubtless arise. So I very much look forward to that day since I think that would be a lot of fun. But the bottom line is that, yes, once I hang up my spurs or am struck by lightning, everything I've created will be released to the public, and I would be honored if there was interest in keeping it alive and growing into the future in whatever form might make sense.

Leo: Nice.

Steve: So it will not all be lost.

Leo: Yay.

Steve: Gaelin wrote: "Hello, Steve. In Episode 1019 you were talking about the constant Internet spam and brute forcing going on. It is so much worse than you stated." He said: "I have ssh open on my home lab so that I can manage it remotely, with Fail2Ban configured. Fail2Ban monitors auth logs and can do automated actions based on successive failures. I have Fail2Ban set up to ban the IP of anyone who has two failed login attempts for three hours, then ban anyone with two bans in the same day for a year. As this lab is only used by a close friend and myself, and we both use keys to authenticate, it's unlikely for us to ever have a failed login attempt. I set it up with a Discord bot to automatically notify me of bans and send me daily reports on ban counts, and it is crazy to watch."

Leo: Yeah. Yeah, I bet.

Steve: "I've seen days with up to 5000 unique IPs banned; normally it's around 300 to 500. I see a failed login attempt around every two to three minutes, 24/7/365. Not all of them end up banned because some of the bots space their logins out a lot. I have banned around 26,000 unique IPs, and at any moment have around 4,000 banned. I highly recommend that anyone hosting publicly accessible SSH install Fail2Ban, even with just the default settings ssh. Thanks for the podcast. Gaelin."

Now, this was a great data point, and not only supports what we were talking about four weeks ago during podcast 1019, but also more recently, when I was talking about the fact that typical network monitoring is only looking at what gets inside the network. While that's certainly, inside the network is of the most concern, there's still the fact that we don't know what we don't know. The fact that Gaelin has witnessed this firsthand has doubtless altered his behavior in a healthy direction. It will serve to inform him about just what a jungle it is out there, and the degree to which he can really never afford to take his own security for granted.

Say, for example, that he was still relying upon username and password for protection. If he didn't already know better, and he does, but if he didn't, seeing the truth about how much attention his own SSH server is drawing would doubtless motivate him to take the time to be as secure as he could possibly be.

Like Gaelin, I've looked at my own external bandwidth logs and what's going on out there. As he said, 24/7/365 it is truly harrowing. I mean it's insane. We talked a few

podcasts ago about the abuse of login attempts to Microsoft Outlook, and how wrong it feels that Microsoft are not providing better abuse protection. Everyone knows that credential stuffing attacks have grown to become one of the major threats on the Internet, yet Microsoft only offers geofencing for their enterprise users.

A few podcasts ago I took the opportunity to rave about my absolute favorite SSH client and SSH server, Bitvise for Windows. Many of our listeners wrote to let me know that Windows already has SSH client and server solutions built-in. And that's absolutely true. Windows now offers the industry standard setting OpenSSH server. So thanks to our listeners for notifying me of that.

But Windows doesn't have Bitvise built-in. In addition to having an extremely pleasant zero-learning-curve graphical user interface, I have my Bitvise server instances configured to only consider ever accepting incoming connections from IPs located in the United States. And within the U.S., since connecting to the Bitvise SSH server with the Bitvise client is 100% reliable, a single failure to authenticate from within the U.S. permanently blacklists that IP. And just so that I'm not locked out in the event that I fumble-finger the connection at the client end, I have permanent whitelist IP overrides for the two IPs I would probably always be connecting from. As I've mentioned previously, my two cable modem IPs are extremely static. And all of that is after configuring the server to only accept authentication via a public/private key exchange challenge.

Finally, all of that was done with a few clicks of a mouse while browsing the Bitvise user interface. So, much as I strongly prefer "living off the land" solutions using what's already present, in this case I'm not giving up Bitvise for anything. It remains my highest possible recommendation for anyone who wants to run an SSH server on Windows. It is trivial to implement that level of, you know, multilayered security. And, I mean, it is - I cannot imagine, like Gaelin, running an SSH server where you don't at least, I mean, like at least geofencing. Why would, if I am always in the U.S., and I virtually always am, why would I ever entertain having my SSH server accept a connection from India? And that's, as it happens, where like the majority of them are coming from. That's just, you know, no. And it's easy to just click a button and say "U.S. only, thank you very much."

Okay. So this is a great, great piece from a listener of ours, Matt Davis, who said: "Hi, Steve. I wanted to share a bit of unexpected side effect that I experienced a few months ago when Let's Encrypt stepped up from single-perspective issuance and started requiring a second perspective." Remember we talked about this a few weeks ago, how due to the possibility of local border gateway protocol hacking, the CA/Browser Forum had decided that certificate authorities would need to be verifying Internet domain control from multiple viewpoints on the Internet.

So, he said: "I run a small web hosting business on the side for a few clients, and one client called me one morning to report that her website was showing the big scary red Certificate Warning page in Chrome. I took a look and, sure enough, her Let's Encrypt certificate had expired the evening before. As you know, all Let's Encrypt certificates should be renewing automatically through the ACME protocol." And of course just pause here for a second, this is the big nightmare, right, with short lifetime automated delivery of certs is, what happens if anything ever happens to interfere with that process? Suddenly all the websites that are needing to be renewed can't be. So let's hope that doesn't happen. Anyway, it happened to a client of his; right? So what happened?

He says: "After troubleshooting this problem for over an hour, I eventually realized what was going on. This client runs a small local photography business in the U.S. In working to secure her WordPress site, we made a quick and easy decision: She did not need any web traffic from China, Russia, or any other country banging at her digital door. If the

person trying to access the site wasn't in her local area, or even in the USA, they simply had no business being there.

"So we set up Cloudflare to block all traffic from all 194 other countries. It was of no use to her, and it eliminated massive amounts of bot traffic, image theft hot-linking, AI scraping, WordPress login attempts, and other shenanigans. After implementing that rule, requests to her site (again, a local photography business) dropped over 95%, and bandwidth was reduced by even more than that.

"However, now with ACME challenges coming from random countries around the globe, I've had to take steps to whitelist those Let's Encrypt challenges no matter where they come from. Multi-perspective issuance has reduced this site's security, as our Web Application Firewall is now forced to allow certain traffic from any country at any time. This may be an unusual example, but when a website really doesn't need to be global, you can easily reduce your attack surface through geoIP firewall rules and other limitations, or at least you used to be able to. Thanks, Matt."

So, wow, what a great real-life example of the mixed-blessing consequences of increasing security. Whenever we tighten anything down to prevent its abuse, we run the risk of triggering false-positive blocks. You know, in my own example of super-tightly locking down my own access to my Bitvise SSH server instances, I was acutely aware that, yes, there would be some risk that I might lock myself out of my own server. But that was a balance that I judged to be easily worth the risk.

In the instance of Multi-Perspective Issuance Corroboration, which was the title of our podcast a few weeks back, we've only heard from one of our listeners, just now, Matt. And thanks for sharing that, Matt. What a great story. But it's not difficult at all to imagine that there were probably many thousands of other ACME-based certificates that were also probably recently similarly impacted. And Matt's right that by needing to allow a subset of queries from anywhere through to his client's server, so that it's able to authenticate its control of the domain, he has been forced to reduce that website's overall security.

And if Matt were to tighten down on the class of foreign queries that were allowed to reach the server, so that only those qualifying were allowed, that is, if he were to, like, be really specific about what his server accepted, then any change that Let's Encrypt might make to their query protocol could again cause a breakage. We're living in a world of tradeoffs.

One thought I had, and I imagine this probably occurred to Matt, he didn't say, was that Let's Encrypt queries over port 80 using HTTP are what are generated. That is to say, it makes sense; right? Since port 443 is what you're trying to provide a certificate for, Let's Encrypt's ACME protocol works over port 80, which is not encrypted. It itself does not require encryption in order to do its job. So it uses port 80 because it needs to be sure to be able to make a connection even when there's no certificate present because it's about issuing certificates.

So Let's Encrypt queries over port 80 using HTTP. The good news is that pretty much nothing else uses port 80 anymore. We were recently talking about Cloudflare dropping all API support over port 80 because they just don't need it. I haven't looked at Cloudflare's country-based filtering closely. But if it were possible to block all port 443 access from everywhere other than the U.S., that ought to restore much of the benefit of a full blanket block. In other words, block all 443 from everywhere but the U.S., but not port 80, which could be coming in from ACME verification.

So that would mean that only traffic coming to port 80 would be allowed from anywhere. Otherwise, 443, which is really all you need now for a website, could be restricted to the U.S. - which, as Matt saw, was a huge win.

Then, since Let's Encrypt's ACME protocol always and only looks for its domain control authentication token in the "acme-challenge" subdirectory of the ".well-known" root directory, that is to say there's one specific directory the ACME protocol looks in, it would probably be possible to set up a .htaccess or a web-config rule to only allow queries over port 80 to that one directory, which would, like, be absolutely uninteresting to anybody but ACME protocol. That ought to allow Let's Encrypt to obtain what it needs over port 80, incoming from anywhere in the world, while not giving any of the rest of the non-U.S. world anything that it might find interesting. No login attempts, for example, or any of the other shenanigans that Matt talks about.

And, boy, what a lesson that is just to geofence a site that does not need international presence in order to dramatically reduce all of the crap that, you know, the Internet otherwise is. And it's not, you know, her site's not like some big deal; right? It's a local special interest photography site for her region. Yet look what it's subjected to. Wow.

Daryl in Kansas says: "Steve, I'm a SpinRite site license guy." Much appreciated, Daryl. "I listen to Security Now! every episode. How safe is the 'trust this computer' option for websites when you're at home on your own network?" He says: "I use a Chromebox for extra security. Do you click yes, or let sleeping dogs lie? Thanks for Security Now!, and hi to Leo." Hi, Leo.

Leo: Depends how much you trust your spouse or evil maid, I guess.

Steve: Right. Well, the sense is I wanted to explain to Daryl what was going on. So what's going on beneath the surface is not at all obvious from the question itself; right? You know, trust this computer. Like, what? It's my computer. Why would I not trust it? So as we know, each of our web browsers which makes queries to remote websites, each of those queries stand alone. That means that, unless something explicit is done, there's no way for a remote website to know who any given query is coming from. That something explicit that is now always done is that any time a web browser query is made which does not include a browser cookie, one is sent back to the browser, a unique cookie is sent back to the browser with its reply so that all subsequent queries which issue from that browser will automatically be "tagged" with that new unique cookie, since that browser cookie will always be returned.

So the first thing to appreciate is that all of the web browsers that are querying remote web servers - if they don't already have one - are each given a unique cookie so that the remote site has some means of telling them all apart. The next important point is that if a specific user identifies themselves to that remote website by logging into it using some credentials, it's the ongoing presence of this cookie that serves to keep them logged in. Their logged-in-ness is thanks to that cookie.

Okay. Next, it's probably always possible to deliberately and explicitly logout of any website. There's always going to be some logout option, generally, by growing convention, in the upper right-hand corner of the website's pages. But the question is, what happens if you do not remember to logout? Many websites don't care at all how long you've been gone, how long you've been away. When you return, you'll still be logged into that site. And the only reason you'll still be logged into that site is that your web browser has remembered and still has the cookie it received the last time you were logged in.

GRC uses the XenForo software for its various web forums, and I cannot recall the last time I was asked to log into my own forums. You know, for me that's a convenience, and I'm sure it is for all of the people who hang out there, you know, since in my case I'm the only one using any of the computers where I'm logged into our forums. So I'm able just to go to forums.grc.com and pick right up where I left off. The same thing is true for X.com. Actually, there was an instance where about a couple months ago I got logged out, and I had a hard time getting logged back in. Because, I mean, I'd been logged in for years, and something happened where I lost my browser cookies, and so I had to, like, you know, do it again.

So, you know, everybody's used to now these days you're just sort of - you stay logged in. But what if multiple people use the same computer? Or what if you're logging in at an Internet Caf or in a public library? In that case you would not want your logon to be so persistent. And that's what this "Trust This Computer" checkbox, which often accompanies a logon page, is all about.

Cookies all come with an optional expiration date. If that date is ever reached, the web browser will no longer honor the cookie. Instead it simply deletes it. But I mentioned that the expiration date is optional. If a cookie is given to a web browser without any expiration date, then that cookie is deliberately never written, in any way, to any form of persistent physical storage. It is only deliberately and explicitly ever retained in RAM. That means that, once the web browser application is closed, the values of any of the non-expiration-dated cookies it may have received while it was running will be lost forever. And that's the beauty of NOT having the "Trust This Computer" checkbox checked when you log into a website.

When logging in with that checkbox unchecked, any logon authentication cookie your browser receives will have no expiration date set. So it will be ephemeral, and your logged in identity will be deliberately lost when you close the web browser application.

So, Daryl in Texas, I mean in Kansas, you asked: "How safe is the 'trust this computer' option for websites when you're at home on your own network?" And only you can really answer that. But now you probably can, since you should have a good understanding of exactly what that means. It boils down to whether anyone else might have physical access to any computer where your prior logons would be persistent because you had enabled the "trust this computer" option which will have created persistent logon sessions.

If you are the only person who has access to any computers where you might have left a site logged on, then remaining logged on is likely a convenience that would have no downsides. But if others might use a computer where you were left logged onto a site which you would prefer they not gain access under your account, and since you might easily forget to explicitly logout after using that site, then logging in in the first place with "trust this computer" disabled would mean that you'll be automatically logged out when the browser is closed or the computer is turned off. So that's the whole tune-up on what's going on with that checkbox.

Leo: It used to be that sometimes they'd say "Are you on a public computer?" Remember that? And that may be a little easier to understand for people.

Steve: Yeah. I mean, it's like my own computer. Why would I not trust it?

Leo: Right.

Steve: Because it has an inetpub folder on it?

Leo: Yes, that's a good reason. But, I mean, so I think that that's probably a more accurate way to ask the question. Obviously some lawyers are seriously...

Steve: Trust this computer. I guess because, if you were at a computer in an Internet caf...

Leo: Right.

Steve: ...or in a library...

Leo: Well, then you would certainly not; right? Because you wouldn't want to leave [crosstalk].

Steve: Yeah. You would say, I don't trust this computer. I don't know who's going to look at it next.

Leo: Right, right, right. So I think the public computer made more sense to people. But, I guess, yeah. Do you trust this computer?

Steve: Yeah. And you can't ask, would you like your logon session to be forgotten and shut the browser down. It's like, what?

Leo: Huh? Actually, that is the right question.

Steve: That actually is the right question.

Leo: That is the right question.

Steve: Yes.

Leo: Maybe this would ask that.

Steve: Okay. And one last piece of feedback from Angus McKinnon. He said: "After reading the following, what would you recommend?" He said: "I am a Backblaze customer."

Now, okay. Angus's note included a link to a document from the website of Morpheus Research. I have the link in the show notes for anybody who might also be a Backblaze customer.

Leo: Before you get too far into this, though, I do want to issue - I've been looking at this and trying to figure out whether we should talk about it.

Steve: Okay.

Leo: Backblaze denies it. They say these Morpheus guys don't know what they're talking about.

Steve: Okay. So for what it's worth, I was very careful to say that, you know, based on this.

Leo: Right. Who knows? This basically came from somebody who was shorting Backblaze. So...

Steve: Although it doesn't sound like there's much left to short.

Leo: Well, if you believe this. That's the point.

Steve: Okay. So let's do this. Because they've been around forever, they've been around for 18 years, you know, the name is very familiar. They were founded in 2007, and they went public four years ago in 2021. Nobody disputes any of those facts. Apparently their stock is not worth what it once was.

And so Angus saw the same research that you and I, Leo, have both seen. And he's freaked out by it. I ended up noting that this research said that Backblaze had lost many of their customers to Wasabi. And all I know about Wasabi is that they used to be a sponsor of the TWiT Network.

Leo: Yeah. I know the guy who created Wasabi, and he's a good guy.

Steve: Yes.

Leo: So I like Wasabi.

Steve: So what we'll say to Angus and any of our listeners who may also be Backblaze customers, is I have a link to uncorroborated...

Leo: That's what it is, basically.

Steve: ...report which would, if you, like, were really dependent on your backed up data, worry you. Whether you would be right to be worried, I don't know.

Leo: Yeah. A lot of people in our community use Backblaze. So it just makes me very nervous. I really went back and forth about whether I would want to report this story or not.

Steve: So it's there. And I think we've said enough. I don't know how to corroborate...

Leo: It came from a short seller. So that means somebody who has shorted their stock, who wants their stock...

Steve: Is going to benefit from further driving it down.

Leo: He wants their stock to go down so he can make money. So that's the only reason - that was an alarm bell. And so Backblaze is a great company. They have been for a long time. They do that hard drive report, which is extremely useful. I know many people who use Backblaze, including many of our hosts. So I'm very reluctant to...

Steve: So I get - so, okay. So I would say I don't care about Backblaze's status because I haven't ever used them, and I don't use them.

Leo: Right.

Steve: There are many allegations here that could be checked. You know, there are some that can't be, right, like the value of their share price. That's a matter of public record.

Leo: Right. Backblaze says the report is inaccurate and misleading based largely on litigation of the same nature and a clear attempt by short sellers to manipulate our stock price for financial gain. They claim that independent third-party reviews have found there has been no wrongdoing or issues with Backblaze's public financial results.

Steve: There are allegations of multiple lawsuits against them, so that would be something that is also in the public record.

Leo: That's true. Those are true. Those are real. Yeah. I mean, it's important to know, just my journalistic nose went up a little bit, and I thought...

Steve: They're also in my hometown, which made me sad.

Leo: They're in Irvine. Well, that means they're good.

Steve: No, no, they're in San Mateo in Northern California.

Leo: Oh, your hometown hometown.

Steve: Yeah.

Leo: Where you grew up.

Steve: That's where I grew up, in San Mateo.

Leo: Yeah. You know, we'll keep digging on this, and we'll absolutely report on it if we can get any corroboration of these allegations.

Steve: Yeah. And mostly I just wanted to bring it to our listeners' attention because this, you know, Angus was worried because he's a Backblaze customer. And he said, what do you think about this? And again, the way I phrased this, I said, you know, I said this report clearly unnerved our listener Angus, who wonders what I would recommend.

Leo: The lawsuits came from two former employees, one of whom was their head of finance for four years, and the other was VP of investor relations. So the lawsuits - and they're real lawsuits, but they haven't been adjudicated yet, either. So I just - I don't know.

Steve: Yeah. And the report alleges that, since the IPO, the share price has dropped by 71%.

Leo: Right.

Steve: Again. You could look that up. That would be a matter of public record.

Leo: That's accurate, yeah.

Steve: They apparently raised $100 million when they went public. And, yeah, and I mean, I've never heard anything negative about Backblaze. So, you know. And who's to say, Leo, that if the company's in trouble, they're a public company, they've got customers and assets and revenue, they might be purchased by a big fish.

Leo: Right.

Steve: So it's not to say that, you know, that they're not a going concern and would not remain viable.

Leo: Yeah. Their quarterly results come out May 7th. Maybe we'll learn more then.

Steve: Cool. And I'm glad you gave us a...

Leo: Yeah, well, I've been going - I've been, you know, since the story broke I've been going back and forth about how we wanted to report it. And so I'm glad you brought it up.

Steve: Yeah. In case a listener needs to know.

Leo: For our listeners' intelligence.

Steve: I'm glad they know.

Leo: Yeah, they can do with it...

Steve: Even though we're not able to make any kind of representation.

Leo: Right. All right, Steve. I'm dying to hear more about the Sandbox Escape or whatever you call it.

Steve: Well, so...

Leo: What do you call it? Is it an escape? Is it...

Steve: No. It is malware has figured out, hey, there's this cool thing called Windows Sandbox. Let's hide in there.

Leo: Yeah, yeah.

Steve: So last week's "Windows Sandbox" podcast reminded us that everybody with Windows 10 or 11 - with the exception of Home edition users - has access to a very nifty Windows execution environment, specifically designed to allow users to safely experiment with "throwaway" programs, installations, files, and anything else, without having any impact on their primary Windows OS installation. And moreover, I was very impressed with Microsoft's surprisingly efficient and economical implementation which got so many things right.

One interesting feature of Windows Sandbox which I believe I mentioned in passing last week is that Windows Defender - and this is certainly salient here - is disabled by default within the Sandbox, and it cannot be enabled via either the GUI or PowerShell commands.

Leo: Interesting.

Steve: So isn't that a nice little place for malware to hide, somewhere where there is no AV. Now, this decision was presumably made because running Defender inside the Sandbox would slow everything down, because users might specifically wish to run things that would cause Defender to freak out, you know, to quarantine and delete their files, and because the entire point of the Sandbox is that it's a safe place where terror may reign with confinement, and nothing can get out. You've got full confinement there.

So unfortunately, it would probably come as no surprise to anyone who's been following this podcast for long to learn that the bad guys have figured out how to take up residence in Windows Sandbox as a means of obtaining secret persistence within Windows systems while still being hidden from Windows Defender and any other AV scanning, which, you know, might be patrolling the grounds outside the sandbox, but be unable to see inside. So let's take a closer look at how Windows Sandbox is being abused and what that means. And then we're going to examine what can be done to prevent its abuse, whether a user wishes to use Windows Sandbox or not for themselves.

So I'm going to start by sharing a piece, an overview of the problem which appeared in the Risky Business newsletter. That newsletter was headlined "Chinese APT" - so, yes, we have Chinese Advanced Persistent Threat actors. "Chinese APT abuses Windows Sandbox to go invisible on infected hosts." Catalin, writing on the newsletter, wrote: "A Chinese cyberespionage group named MirrorFace (a.k.a. Earth Kasha and APT10) is abusing the Windows Sandbox virtual environment to hide the execution of its malware on infected systems. Attacks incorporating Windows Sandbox have been taking place since 2023 and represent the first known case of Windows Sandbox abuse since its release in December of 2018.

"As the name hints, the feature allows Windows users to start an isolated sandbox where they can temporarily install and test apps and then shut down the virtual environment without impacting the main OS and their data. It functions as a virtual machine, but it doesn't have all the bulky features of a VM. It's light, super fast, easy to start and use.

"Abuse of this feature sounds implausible because Windows Sandbox support is disabled by default; and when a Sandbox is started, it runs in a window in the user's foreground. But according to reports from the Japanese government and ESET, MirrorFace has found a way around these limitations. The group gains an initial foothold on compromised networks, enables Windows Sandbox, restarts systems, then silently launches Windows Sandbox instances that do not appear on the screen. This is accomplished by launching the Sandbox via Task Scheduler under a different account from the user's current one, so the Sandbox UI never appears on the logged-on user.

"The MirrorFace operators drop malware in a folder on the infected systems, then use Windows Sandbox .WSB configuration files to share access to that folder to the Sandbox, grant the sandbox network access, then configure one of the malicious files to automatically run when the Sandbox is executed. Since Windows Sandbox environments cannot run Defender, nothing happens inside and is either logged or detected. This allows the attacker to install malware and open a hidden backdoor inside that system and a victim company's network.

"Japanese security firm ITOCHU explains how blind companies can become against Windows Sandbox-based attacks. They wrote: 'Since the malware in Windows Sandbox operates according to the WSB file's configuration, it can access files on the host machine. However, because the files are accessed from the sandbox, activity is never logged by monitoring tools running on the host system.'

"The technique used by MirrorFace seems to be an evolved version of a technique first documented by security researcher Lloyd Davies back in 2020. ITOCHU researchers say the abuse can go a few steps further since new features are constantly being added to

Windows Sandbox. For example, the Windows Sandbox can now share clipboard, audio, and video input with the base OS. The Windows Sandbox can now also be started via command line arguments using the new 'WSB.exe' command, which removes the need for WSB configuration files, which are artifacts security firms could use to detect possible abuse.

"The technique is incredibly simple to automate, even for low- to mid-tier skilled malware developers. Once detailed in these reports, it is likely to spread to other groups. The first to jump on and abuse this technique are likely ransomware gangs. Some groups are already using something similar. At least half a dozen ransomware groups have been spotted installing bulky VM software" - you know, full virtual machine suites - "on infected hosts just to start the VM and send victim files to be encrypted inside, where security tools don't have access to spot the ongoing encryption.

"Since Windows Sandbox is built-in and present on all Windows 10 and Windows 11 systems, and the app's file is signed by Microsoft itself, abusing it is likely easier and safer. ITOCHU has published some monitoring and infection remediation advice to detect this technique, but the cat is out of the bag now, and further and broader abuse is now expected to start taking place."

Okay. So one thing that's very interesting is the observation that the Windows Sandbox is able to launch and run under a different user's account so that the foreground user never sees any indication that it's happening in the background. And here, the inherent efficiency of Windows Sandbox which so impressed me last week actually works against the user, since its lightweight nature means a user would be much less likely to wonder where all their free RAM went because it wouldn't be going anywhere. It wouldn't be consuming very much. Just like an app. Also, the default-enabled clipboard sharing is a bit chilling, since it would be a bit like having a malicious instance of Windows Recall running unseen in the background, capturing anything the foreground user might temporarily place onto their clipboard, such as a cryptocurrency wallet address.

I was curious to see what this researcher Lloyd Davies came up with five years ago in 2020. Whatever it was, Microsoft apparently blew it off without a second thought since we're now five years downstream of that, and Windows Sandbox is still here and completely abuse prone. Five years ago, under his headline "Weaponizing Windows Sandbox to Bypass Defender," Lloyd Davies wrote: "This short blog post may be useful for a Red Team living off the land for the execution of payloads on a machine where Windows Sandbox can be enabled. Windows Sandbox is designed to work this way. No exploitation of anything is covered in this post. With this technique in terms of executing within a VM, we don't need to load an external ISO onto the machine, as all of this is handled by the Sandbox. In my research, the Sandbox .WSB configuration file was not inspected or blacklisted on any major EDR or AV.

"At the tail end of last year, Microsoft introduced a new feature named Windows Sandbox, WSB for short. Windows Sandbox allows you to quickly, within 15 seconds, create a disposable Hyper-V-based Virtual Machine with all the qualities a familiar VM would have such as clipboard sharing, mapping directories, et cetera. The sandbox is also the underlay for Microsoft Defender Application Guard, for dynamic analysis on Hyper-V-enabled hosts, and can be enabled on any Windows 10 Pro, Enterprise, or Education machine, making this perfect as a living-off-the-land technique."

So, you know, he's couching this all as Red Team, not, you know, like how like a Red Team who are good guys acting to see, like to do exploit testing against someone who has hired them to check their defenses could use in order to obtain an undetected presence on computers.

So he says: "The TL;DR of this technique is to craft a .WSB that can be executed on an endpoint, which mounts the user's file system, allowing us to execute the implant inside a hidden VM and bypass any AV/EDR that's on the host. The WSB configuration also seems to be bypassing Windows Defender on the host where it's executed. It's not incredibly complicated, but could prove useful in an engagement."

Lloyd then proceeds to talk about and document the various ways very powerful .WSB files can be created to give a malicious sandbox all the power it might need on the user's system, all while always remaining completely hidden and undetectable. He concludes his observations by writing: "A similar technique has been used by the infamous Maze and Ragnar locker threat actors in recent times; however, they've installed third-party virtualization suites such as VMware and VBox. Using Windows Sandbox bypasses the requirement for this software to be installed. To complement this technique," he says, "I created a simple Go program to find drives automatically and mount network shares that include them as mapped folders, and then generates a .WSB based on this."

I have a link in the show notes to an English language translation of the talk that was given last January in Japanese by the ITOCHU researchers. Among the many other things they've noted is that, with the introduction of Windows 11, Microsoft has enhanced the Sandbox's features in ways that allow for additional abuses. They wrote: "The changes to Windows Sandbox after the Windows 11 update are as follows: Addition of the WSB.exe command, enabling sandbox execution via the command line, background execution of the sandbox, and the ability to modify certain settings via the GUI. These recent feature updates may make it more difficult to detect attacks leveraging Windows Sandbox. The key reasons for this are as follows."

And they list three: "Background execution of Windows Sandbox. Previously, in Windows 10 and early versions of Windows 11, Windows Sandbox always ran as a foreground GUI application. However, with the new WSB.exe start command, it can now run in the background. As a result, the sandbox can be launched without user awareness, and its window remains hidden until the WSB.exe connect command is executed."

Second: "Sandbox execution without a WSB file. The updated WSB.exe command allows sandbox configurations to be set entirely via command-line arguments. Previously, WSB files were an important forensic artifact during investigations, but this change increases the risk of leaving no trace of sandbox usage."

And third: "Persistent data inside the sandbox. In earlier versions, closing the Windows Sandbox window would terminate the process and delete all data within the environment. However, after the update, closing the window does not stop the sandbox, and its data remains intact. To delete data, the sandbox must be explicitly stopped using the WSB.exe stop command, or terminated by shutting down the host machine. This change significantly increases the potential for long-term attacker operations within the sandbox.

"Given these updates, security researchers must carefully verify whether such feature changes improve convenience for attackers and implement appropriate countermeasures when new functionalities are introduced."

Okay. So I titled today's podcast "Preventing Windows Sandbox Abuse" because, having now explored the dark side of this otherwise truly useful and nifty Windows Sandbox feature, if it's not something that its user will be actively using, it might be worth considering taking some measures to neuter it so that it cannot be abused behind its user's back.

My number one favorite way to do this would be to disable a system's virtual machine extensions capabilities at the pre-boot firmware level. I recently learned that the BIOS settings backup battery on the aging Gigabyte motherboard of my older Win7 machine

had died. My neighborhood had a planned day-long power outage while our local power company's equipment was replaced. When I fired my machine back up after having it shutdown for the day, I quickly saw that it had lost its time of day and date clock. That's probably something that's familiar to use oldsters back in the days when...

Leo: Our CMOS battery dies.

Steve: The CMOS battery died, exactly. So I rebooted and went into the BIOS and set that, the time and date, correctly. Sometime later, when I attempted to launch a VirtualBox virtual machine, I received an error that VBox was unable to operate without the Intel Virtualization Technology, which is abbreviated VT-X, enabled in the system's BIOS. I mentioned last week that the same is true for Windows Sandbox. The Microsoft Hyper-V virtualization technology the Sandbox depends upon is, in turn, dependent upon having Intel's Virtualization Technology enabled.

So the absolute best protection for anyone who does not routinely use either the Windows Sandbox nor any of the many other various virtualization systems, since all of those are now known to be prone to abuse, as well, and especially Windows Sandbox, would be to simply run without the Intel VT-X extensions enabled. No VT-X means no virtualization funny business, period. Doing this will have zero impact upon Windows operation, and it will completely shut down any chance of abuse.

Now, if you do need to run virtual machines other than Windows Sandbox, you'll need to have the Intel VT-X extensions enabled in your machine's firmware. Enabling Windows Sandbox requires admin privileges. But we know that doesn't present much of a barrier to malware since pretty much everything bad that malware does requires admin privileges anyway, so they're able to get it. And we know that elevation of privilege exploits are constantly being uncovered.

The solution for anyone who wishes to prevent any "behind their back" exploitation of Windows Sandbox and for whom disabling all use of virtual machine technology via the VT-X extension is not an option, Windows AppLocker is probably the next best solution. AppLocker can either be configured in a managed enterprise setting through group policies or on a local machine using the Local Security Policy snap-in. The use of AppLocker is straightforward, and many How-Tos exist on the Internet for anyone who wants to take that approach.

Under Windows 10 or 11 you'll want to block the execution of the WindowsSandbox.exe executable program which lives in the System32 directory. It's System32 \WindowsSandbox.exe. And additionally, under Windows 11, you would also want to prevent the WSB.exe command from being used. Once any of those have been foreclosed, anything that tries to crawl into your machine and set up shop behind your back using the Windows Sandbox will be out of luck.

And I'm not suggesting that this is like the sky is falling, and some, you know, major security problem to worry about. Remember that something bad has to get into your machine first, before it's able even to have the opportunity to enable and use the Windows Sandbox behind your box. So it's not like having the sandbox there is, you know, sending out a call for malware to come crawling in your machine. All of your existing defenses, Windows Defender and AV tools and everything else that's already there, is still functioning. It's just that, if something gets in, everybody now knows there is a new place for it to hide. And hopefully Microsoft will take some action and do something to minimize, you know, the potential for this behind-our-back abuse because this is, you know, if bad guys are bothering to install VMware and VirtualBox on people's machines, they're sure going to be trying Windows Sandbox first.

Leo: Do they do that, they install virtual machines?

Steve: Yes.

Leo: Yeah.

Steve: They bring the whole VMware or VirtualBox system in.

Leo: Wow.

Steve: It's crazy. And actually run a, you know, a VMware or a VirtualBox VM in the background.

Leo: Now you don't have to because you can just use Windows Sandbox.

Steve: That's right. Fifteen seconds you're ready to go, you bad malware, you.

Leo: Great stuff, as always, Steve. Thank you so much. I appreciate this.

Steve: Bye.