The rise and fall of code.microsoft.com

Description: How has Microsoft responded to the tidal wave of criticism over Recall? And what about Google? Who else recently lost control of their data? Apple devices will be getting a password manager? What about iCloud? Is that a drone recording a wedding, or a Chinese Communist Party surveillance device? What did SlashData's survey of more than 10,000 coders reveal about their use of AI and choice of language? And if AIs can code, what's the career future for programmers? Why has the Linux Kernel project suddenly begun spewing CVEs in great number? Will we be able to order pizza in the future? What did one listener discover when he attempted to register his new Passkey devices across the Internet? And how did a stunning mistake at Microsoft turn into a goldmine of attacker intelligence?

High quality (64 kbps) mp3 audio file URL: http://media.GRC.com/sn/SN-978.mp3
Quarter size (16 kbps) mp3 audio file URL: http://media.GRC.com/sn/sn-978-lq.mp3

SHOW TEASE: It's time for Security Now!. Steve Gibson is here. He has a rebuttal to Microsoft's rebuttal to last week's accusation, all about Microsoft Recall. He still says you might want to hold off on that one. Apple gets a password manager. Is it good enough for government work? We'll also talk about the use of AI and coding. Is AI going to kill the future of code? And what Microsoft did when they had a little error that's ended up becoming a very useful tool to go after bad guys. It's all coming up next on Security Now!.

Leo Laporte: This is Security Now! with Steve Gibson, Episode 978, recorded Tuesday, June 11th, 2024: The Rise and Fall of Code.microsoft.com.

It's time for Security Now!, the show you wait all week for. As Steve says, it's Tuesday, it must be Steve Gibson. Hello, Steve.

Steve Gibson: Hey, Leo. We were out walking with some friends around the neighborhood yesterday, and there's like three long legs of our community, and after doing two of them we all started down the third. And I said, whoop, nope, it's podcast night, I've got to get back to working on the podcast.

Leo: I love it. Aww.

Steve: So the walk was cut short. And Lorrie had been pushing me to start earlier on Mondays because I would be trying to get work done through the mornings, but invariably it would creep into the afternoon, and then it'd be like, oh, shoot, I should be

further along. So now I'm just not even trying to get anything done on Monday except work on the podcast. Now, that's the driving force. We're seeing the result of that today because we have 24 pages of show notes as a consequence of me having gotten an extra early start yesterday. Now, normally the people who listen to us at 1.5x do so because I kind of plod along. There'll be no plodding today because we have a lot of ground to cover. So if you are hearing me at 1.5x and wondering why I seem to be going a little faster than usual, you might want to consider dropping down to 1x and hearing it at normal speed.

We've got a bunch of things to talk about on today's podcast 978 for Patch Tuesday. And I am mourning the fact, and if one of our listeners is listening, someone tweeted me a couple weeks ago a picture of a pipe that just had patches all over it like crazy, and silver, like, screw-on straps. And it looked like this thing was on life support. I thought, that is the most perfect Picture of the Week for Microsoft Patch Tuesday. But I don't know what happened to the picture.

Leo: Oh, no.

Steve: So maybe if the person is listening you can send it to me again, and I'll hold it for July's Patch Tuesday. We're going to talk about something really interesting, which is - I titled this "The Rise and Fall of Code.microsoft.com," which everyone would recognize as a subdomain, the code subdomain of Microsoft.com. Essentially the way what happened started was with an architectural failure the scope of which I can hardly grasp. I have no idea what's going on up there in Redmond that what I will describe could have happened, but it did. But they turned it around. I mean, they still have some problems to solve, but they managed to repurpose this mistake into something that ended up creating a ton of intelligence for them, collecting a ton of intelligence. Anyway, we're going to have a lot of fun with that.

But first we're going to talk about how Microsoft has responded to the tidal wave of criticism that they've received over Recall, and what about Google. Who else recently lost control of their data? Apple devices will be getting a password manager? What? What about iCloud? I thought we already had that. Is that a drone recording a wedding, or is that the Chinese Communist Party surveillance device?

Leo: Whoa.

Steve: What did SlashData's survey of more than 10,000 coders around the world reveal about their use of AI today and their choice of language? And if AIs can code, which they seem able to do, then what's the career future for programmers? We have some feedback on that. Also, why has the Linux Kernel project suddenly begun spewing CVEs in great number? Will we be able to order pizza in the future, or should we just give up now? Or maybe order all your pizza today and, you know, freeze it. What did one listener discover when he attempted to register his new Passkey devices across the Internet? And as I said, how did a stunning mistake at Microsoft turn into a goldmine of attacker intelligence? So not a slow podcast today for the second podcast of June.

Leo: We know Microsoft never makes any mistakes, ever.

Steve: Oh, Leo. Wait till you hear this one.

Leo: Oh, god. Oh, god.

Steve: I just, when I read this, I thought - and this is from them. This is their own blog posting. Somebody confessed this, which first of all I thought, oh. But, like, really? Well, anyway, we'll get there.

Leo: We'll get there.

Steve: It's stunning.

Leo: I know we've got 26 pages, so let me hurry up here. All right. Back to, speaking of photos, your Picture of the Week, Steve.

Steve: So we've had endless fun with these bizarre pictures of, like, a gate blocking a path where there's open grass on either side of it. It's like, what? Just like what were they thinking? Anyway, somebody had fun over the Microsoft Recall issue and actually relative to their most recent update, this titled "Microsoft Recall," and then it shows us one of the pictures that we've shown before, the bright yellow gate in the middle of a walkway, but, you know, with actually some - it looks like there actually is footpath traffic on either side because the grass is patted down and a little brown. Anyway, at the bottom it says, referring to this gate, "Don't worry, it's encrypted."

Leo: Oh. Doh.

Steve: Yeah. Right, right. And we'll have, well, actually right now. Last Friday Mary Jo Foley, that's right, our Mary Jo of Windows Weekly fame, she tweeted: "Microsoft, bowing to growing security-centric criticism, is making some tweaks to its coming Windows 11 24H2 Recall app. The first Copilot+ PCs still are on track to start shipping June 18." Okay, now, today's June 11th. So that's one week from today, folks. And she said: "...and the tweaks are slated to take effect by then, too."

Okay. So she was referring, of course, to a Microsoft Windows blog, that is, when she said "Microsoft bowing to security-centric" concerns, posted by the Corporate Vice President for Windows + Devices. So it appears that "Windows +" devices are a category. So that would presumably mean Copilot+. Anyway, this guy posted under the title "Update on the Recall preview feature for Copilot+ PCs." And as Mary Jo noted, this is clearly in response to the security industry's reaction over the previous three weeks to the privacy implications that would be present, my contention is, and I'll be echoing this a couple other times today, in any system that aggregates, for all time, everything a user does on their PC. Okay. And I'll have a little more to say specifically about that in a moment. But first let's hear from Microsoft.

Since a lot of the danger Recall represents is reflected by Microsoft's attitude toward Recall, I want to share this VP's entire post. It's not very long. But it's important that their attitude be seen. So bear with me at the start, since it's pure Microsoft marketing-speak. Anyway, everyone will recognize it as such.

He wrote: "Today, we're sharing an update on the Recall feature in preview for Copilot+ PCs, including more information on the set-up experience, privacy controls, and

additional details on our approach to security. On May 20, we introduced Copilot+ PCs, our fastest, most intelligent Windows PCs ever. Copilot+ PCs have been reimagined from the inside out to deliver better performance and all new AI experiences to help you be more productive, creative, and communicate more effectively. One of the new experiences exclusive to Copilot+ PCs" - thank god - "is Recall, a new way to instantly find something you've previously seen on your PC. To create an explorable visual timeline, Recall periodically takes a snapshot of what appears on your screen. These images are encrypted, stored, and analyzed locally, using on-device AI capabilities to understand" - loosely termed; right? - "their context."

Okay. He says: "When logged into your Copilot+ PC, you can easily retrace your steps visually using Recall to find things from apps, websites, images, and documents that you've seen, operating like your own virtual and completely private 'photographic memory.' You're always in control of what's saved. You can disable saving snapshots, pause temporarily, filter applications, and delete your snapshots at any time. As AI becomes more prevalent, we're rearchitecting Windows."

Okay, really? Like there are dialogs from Windows 95 that pop up every so often in Windows 11. So, right. All we've ever seen is some paving over the previous pavement which was over the pavement before that and so on. So, okay, "...rearchitecting Windows to give customers and developers more choice to leverage both the cloud and the power of local processing on the device made possible by the neural processing unit (NPU). This distributed computing model offers choice for both privacy and security. All of this work will continue to be guided by our Secure Future Initiative (SFI).

"Our team is driven by a relentless desire to empower people through the transformative potential of AI, and we see great utility in Recall and the problem it can solve. We also know for people to get the full value of the experiences like Recall, they have to trust it. That's why we are launching Recall in preview on Copilot+ PCs, to give customers a choice to engage with the feature early - or not - and to give us an opportunity to learn from the types of real world scenarios customers and the Windows community finds most useful."

Okay. So under the subhead of "Listening to and acting on customer feedback," he wrote: "Even before making Recall available to customers, we have heard a clear signal that we can make it easier for people to choose to enable Recall on their Copilot+ PC and improve privacy and security safeguards. With that in mind, we are announcing updates that will go into effect before the Recall preview ships to customers on June 18th." So as Mary Jo said, right now, before next week.

"First," he said, "we are updating the set-up experience of Copilot+ PCs to give people a clearer choice to opt-in to saving snapshots using Recall." He said: "If you don't proactively choose to turn it on, it will be off by default." So that's a big change, and that matters. The flipside is, how much does it really matter? You know, we've seen how persistent, seductive, and eventually forceful Microsoft can be when they want to push their users in a certain direction. It's not that difficult to imagine that while the user might need to switch it on, Microsoft will not be cautioning the user about the system's inherent dangers. Rather, they will be promoting the benefits and touting encryption, locality, security, and all the rest.

I believe the upshot will be that users will turn it on, if nothing less, just to see what it's about because Microsoft will be making it very appealing. But still, if nothing else, having people turn it on probably gets them off the hook when things go wrong. After all, well, we didn't ship it with it on. You turned it on. And it's like, yes, because you told me to.

Okay. Anyway, "Second, Windows Hello enrollment is required to enable Recall. In addition, proof of presence is also required to view your timeline and search in Recall."

Okay. So these are all good things; right? They've created additional hurdles, barriers, requirements in order to gain access to this, and I'll just note, through the front door, like gain access the way you're supposed to. They're not talking about gaining access the way you're not supposed to. We'll see how that turns out.

And "Third," he said, "we're adding additional layers of data protection including 'just in time' decryption, protected by Windows Hello Enhanced Sign-in Security. So Recall snapshots will only be decrypted and accessible when the user authenticates." On the other hand - okay. Anyway. Finally, he said, "In addition, we encrypted the search index database." Which wasn't originally decrypted.

So finally he says: "Secure by design and secure by default. In line with Microsoft's SFI" - that's the secure initiative thing - "principles, before the preview release of Recall to customers, we are taking steps to increase data protection. Copilot+ PCs will launch with 'just in time' decryption protected by Windows Hello Enhanced Sign-in Security, so Recall snapshots will only be decrypted and accessible when the user authenticates. This gives an additional layer of protection to Recall data in addition to other default enabled Window Security features like SmartScreen and Defender, which use advanced AI techniques to help prevent malware from accessing data like Recall."

Now, okay, remember that last week Kevin Beaumont deliberately used known-to-Microsoft info-stealer malware, which Windows Defender was so slow to recognize that the info-stealer had already successfully exfiltrated the user's entire Recall history before Defender woke up and shut it down. So again, the problem is Microsoft's heart being in the right place doesn't help anybody because Windows, as we know, is - it's not an exaggeration to say "riddled with vulnerabilities" because, you know, more than a hundred are being fixed today, in today's Windows Update.

Okay. Anyway, he said: "We also know the best way to secure information on a PC is to secure the whole PC itself." Right, because that's been going so well. And he said: "We want to reinforce what has previously been shared from David Weston, vice president of Enterprise and OS Security, about how Copilot+ PCs have been designed to be secure by default and share additional details about our security approach." You know, in other words, unlike all of our previous Windows systems, which really weren't all that secure, even though we've always told you they were. But oh, baby, this time we really and truly mean it. Not like all those previous times.

So he said: "Some notable examples of security enhancements include: All Copilot+ PCs will be Secured-core PCs, bringing" - which, you know, doesn't matter if the Windows that runs on it isn't secure. But he said: "...bringing advanced security to both commercial and consumer devices. In addition to the layers of protection in Windows 11, Secured-core PCs provide advanced firmware safeguards and dynamic root-of-trust measurement to help protect from chip to cloud." And that's a new phrase that Microsoft is using, "from chip to cloud." But, you know, from, what, "cradle to grave."

Leo: That's it.

Steve: Right. "Also, Microsoft Pluton security processor will be enabled by default" - oh, goodie - "on all Copilot+ PCs. Pluton is a chip-to-cloud security technology, designed by Microsoft and built by silicon partners, with Zero Trust principles at the core. This helps protect credentials, identities, personal data, and encryption keys, making them significantly harder to remove from the device, even if a user is tricked into installing malware, or an attacker has physical possession of a PC."

Again, unfortunately, making sure that a buggy operating system isn't altered before it boots or while it's booting doesn't help you once the buggy operating system is running. But at least it didn't get compromised before it booted. Who cares? Anyway, he said: "All Copilot+ PCs will ship with Windows Hello Enhanced Sign-in Security. This provides more secure biometric sign-ins and eliminates the need for a password." Because, yeah, who wants those passwords when you could smile at it?

Okay. Under the headline "Protecting your privacy on Copilot+ PCs," we have: "In our early internal testing, we've seen different people use Recall in the way that works best for them." Blah, blah, blah. I'm going to skip all this because we don't have a lot of time, and this is just all same stuff. He's basically saying, okay, we heard you. We're going to turn it off by default. We're going to seduce people to turn it on. But if they do, it's their fault, not ours because, after all, they were the ones who turned it on. And oh, baby, you know, this is the most secure thing we've ever made. So again, as I said...

Leo: We've heard that before, haven't we.

Steve: ...we've always told you that. Remember Ballmer jumping around onstage about Windows XP, which turned out to be the worst security of any operating system to date that they'd had? Anyway. So basically they're saying we heard you, and here are all the reasons why we're going to keep doing what we were doing except we're going to turn it off by default as our "get out of jail free" card. So anyway, we know that users will be impressed by the sounds of all this security. And I have no doubt that users are going to want to have the power that this provides. Don't get me wrong. I mean, I get it. This is a seductive feature. And that's part of why this is a double-edged sword. You know, make no mistake about it, this is powerful. But it's because it's powerful that it's also so dangerous and brings the potential for great harm.

Will that harm come to pass? Well, we'll be here to see. I should also note that I've been asked by a number of our listeners whether I would consider creating some sort of utility that absolutely positively guarantees that Recall is not running on a machine. We'll see how all this goes. But I am inclined to do so. And if so, I know what I'll call it. And Leo, I will make sure you're not sipping coffee when I reveal its name.

Leo: Okay. I've put the coffee down. Okay.

Steve: Because, yeah. And you'll have to center yourself over your ball because you're going to love this one. Anyway, we'll see how it goes.

Kevin Beaumont also weighed-in on Microsoft's revised explanation. He posted this on Mastodon. He said: "Obviously, I recommend you do not enable Recall, and tell your family not to enable it, too. It's still labeled 'Preview,' and I'll believe it is encrypted when I see it. There are obviously serious governance and security failures at Microsoft around how this played out that need to be investigated, and suggests they are not serious about IA safety."

And I think that raises a really good point. It's like, you know, they announce this, and we saw Satya jumping up and down, talking about how great it was going to be, and the entire security community had a collective meltdown. So that tells you something about, like, why they need to have this in people's machines. Which again comes back to my theory about what they're actually planning, which is that this will be used to train some sort of high-power local assistant. And again, I get it. I mean, that would be so cool. But they've never demonstrated their ability to do anything like this safely.

I should mention that Google with their Chrome OS is also in on the "store everything that happens for possible later use" bandwagon. Everyone can sense that there's huge potential here somewhere, so no one wants to be left out. Last week, John Solomon, Google's VP in charge of the ChromeOS, said that their so-called "Memory" feature - okay, they maybe talked to Apple about naming things as that's what they're currently unofficially calling it, is different from Recall. Okay, but then he describes Recall. He says: "Because users will have control of how and where the 'memory' feature works."

Leo: Uh-huh.

Steve: Right. Just like Recall will offer. So not so different from Recall. And after all, if you turn it off, then you're not going to get it. So people are not going to turn it off. They're going to have it on, if they want it, and then suffer the consequences, if there are any. Anyway, Google apparently already wants to distance itself from the stink surrounding the announcement of Recall.

The New York Times. On the topic of is it possible to keep secrets, last Friday 270GB of data belonging to The New York Times, which I'm quite certain The New York Times wanted to keep secure and secret, and which those in charge of securing it were absolutely and positively certain was completely secure until it wasn't. You know, just like Microsoft is absolutely and positively certain they're going to secure their users' Recall data, until they don't. In the case of The New York Times, it got loose. An unknown threat actor leaked The New York Times source code, as in all of it, all 270GB of it, after one of the company's IT guys apparently left a private GitHub access token in a public code paste.

The leaked data includes the source code of the company's entire public website, mobile apps, and even, for those who are interested, its Wordle game. The 270GB of data being made available on the dark web is mostly unencrypted. The hacker posted: "Basically all source code belonging to The New York Times Company, 270GB. There are around 5,000 repos - out of them less than 30 are additionally encrypted, I think" - he said, "3.6 million files total, uncompressed tar." And I have a picture in our show notes of the screen that was posted on the dark web with the series of links so that you, too, can download 270GB and find out what The New York Times coders have been up to. The lesson here is that, unfortunately, mistakes happen. In fact, Leo, were we to rename this podcast, it would be "Mistakes That Happen."

Leo: Yes, that's the whole show, right there, in a nutshell.

Steve: You know? Yeah. We've seen stories of valuable exposed credentials sitting unnoticed for years. Right? Where, like, some hacker came along and saw that a credential had been posted publicly, but nobody noticed it until now. One real concern for the future, against the background of "mistakes happen," is that there may soon be, if there aren't already, malicious AI-driven bots scanning and rifling through the Internet looking for any fresh mistakes of value that anyone may have made. The point is, our world is changing right underneath us right now, and I'm not sure the good guys are winning. This whole thing feels somewhat asymmetric. Right? Because, I mean, as we know, security is about a series of links, and we keep seeming to add more links to the chain, any one of which being defective can break the entire strength of the chain. Again, it feels like an asymmetric fight that we are not clearly winning.

Leo: Pretty much losing, I think would be fair. All right. How about this? "You Can't Recall." Or "Recall Recall."

Steve: Oh, no.

Leo: No, better than that?

Steve: I mean, I almost have to do the app just so I can use this name. It is...

Leo: "Recall What?" Okay. I can't wait. It'll be fun.

Steve: Let's take a break, Leo.

Leo: Okay. All right, Steve. Let's continue on.

Steve: So, and I know you'll have something to say about this one, Leo. During yesterday's World Wide Developer Conference, or the kickoff, Apple introduced their forthcoming Passwords app. Now, of course, Apple users have long been using their iCloud account to store and sync their passwords among their devices. But what was going on wasn't super transparent.

Leo: No.

Steve: You know, it just worked, but without a clear and clean UI. It was, you know, it was necessary to dig down into the Control app to locate a sub-page. So the passwords app that will be included in the next major release of their OSes, so that would be iOS 18, macOS Sequoia, and visionOS 2, will provide a UI for Apple's storage of this information. Now, since this is not ever going to be a cross-ecosystem solution, you know, it's Apple only...

Leo: Oh, it's Windows, too.

Steve: Well, yeah. And I heard you say something about iCloud for Windows password app.

Leo: Yeah, yeah.

Steve: Anyway, I'll just finish and say that those of us using Windows, Linux, or Android will likely remain with whatever cross-ecosystem solution we're using today.

Leo: That's right. Yeah.

Steve: But this move does create an explicit and native password manager for Apple OSes for the first time. And if someone is 100% pure Apple-world, it likely offers everything anyone would need. It also incorporates clear Passkeys management and a built-in one-time-password-style authenticator. Since I'm currently using OTP Auth as my one-time-password authenticator of choice, I'll look at what Apple has to offer once I upgrade my iPhone to something that'll run iOS 18. I think I'm stuck back on 12 or something right now.

But so Leo, I hear you guys using the word "sherlocked." Where did that come from?

Leo: So way back in the day, I mean, I think this is 20 years ago, there was an app called Sherlock that let you find files on your device. It was really good. You could, you know, you'd make an index, and you'd sherlock, and you could find anything on your hard drive. Then Apple released something it calls Spotlight. And Sherlock was out of business overnight. And so ever after, when Apple introduces a product that duplicates functionality of a third-party product and essentially puts them out of business, we call it "being sherlocked." And there were, I mean, obviously 1Password, Bitwarden, and other password managers...

Steve: There was a sherlock festival yesterday.

Leo: ...may have been sherlocked by this Passwords, but it was just one of many, exactly, yes.

Steve: Now, now, we've seen Microsoft do the same thing, too; right? Like I was complaining that it took until Service Pack 3 for Windows XP's firewall to be enabled by default. Back at the time, remember ZoneAlarm was my favorite firewall. And there was a firewall industry...

Leo: Exactly.

Steve: ...for Windows PCs.

Leo: Sherlocked, yup, yup.

Steve: And then, well, Microsoft says, you know, we're going to put a firewall in, but don't you worry, it'll be turned off by default. Well, it eventually got turned on. And I was just talking to a friend of mine the other day who was asking me if she needed to be still using McAfee. And I said, oh, lord, no. I said, you know, Windows' built-in Defender is really all anybody needs. And I explained that Microsoft does this. They sort of create the capability, but they don't want to step on anyone's toes, so they sort of ease it into the world slowly. Microsoft, I would argue, was a little less caretaking about that. They say, yeah, now we're doing that. So.

Leo: Yeah.

Steve: Anyway. Again, it makes sense for this thing to get moved in. So there is - oh, and iCloud for Windows, how can you use a password manager for Apple under iCloud for Windows? Because, I mean, it just - I thought it was just folders.

Leo: Oh, no, no, no. iCloud for Windows lets you do a lot of things, and including I guess now access your passwords. Actually, I think that's been around. It is not as elegant as a password manager. And remember, this is going to do Passkeys.

Steve: Right.

Leo: But I don't know, and they didn't really say, I think the Passkeys are hardware dependent, which means it seems unlikely Passkeys would make it over to Windows. But they might do what they do now, which is show you a QR code, and then the phone that you have the Passkey on, you do the QR code.

Steve: Right.

Leo: So, you know, is it going to be as full-featured as a standalone password manager? Probably not. That's often the case with these things.

Steve: Well, and I'm multiplatform. So...

Leo: Yeah, well, you and I won't use it. I can't - I'm Android in Windows and Linux.

Steve: Right, right. But again, Apple...

Leo: A lot of Apple people - and you know what, I love this because it means a lot of non-sophisticated users will just do it because it's part of the operating system. It's built in.

Steve: Yes. And I think the fact that they're showing it as a separate thing, you know, helps to raise people's awareness of passwords and the various aspects of passwords, like Passkeys and one-time passwords and so forth. It just, you know, it brings it more to light, which has got to be a good thing.

Leo: Yeah.

Steve: I saw some talk a while back about some congressional pushback on Chinese-made drones by DJI. And those DJI drones are by far the best drone technology around. In advance of the U.S. Senate's planned discussion of the so-called "Countering CCP Drones Act," which would limit the use of Chinese-made drones in the U.S. on the grounds of national security, tomorrow, June 12th, DJI will be disabling the ability of users in the U.S. to sync their drone flight data to its servers, and the option to sync U.S. drone data at all will be completely removed by the end of the month.

So, you know, DJI is seeing what's going in with TikTok and this general sort of concern over what the Chinese Communist Party is doing with technology that U.S. consumers are excited about. And so I'm sure they don't want to lose this market. So they're saying, okay, fine, we're going to strip this out of our devices. So I don't know what a problem that will be for DJI users, if being able to sync drone flight data to servers is a big deal. But it'll be gone by the end of the month.

Okay. Another thing that, Leo, I think you're going to get a kick out of seeing, although you and I are not represented among these statistics. SlashData revealed some interesting developer statistics. They recently surveyed 10,000 developers from more than 135 countries. The question put to them was "How has AI affected your workflow?" Okay, now, let me first allow SlashData to introduce themselves. They wrote: "If this is the first time you've heard about SlashData" - did I say Slashdot? SlashData.

"If this is the first time you've heard about SlashData, I'm happy to share a few quick words," writes the person who posted this. "SlashData is a developer research company. Every quarter, SlashData runs a survey on the global developer audience, to measure the pulse of the developer ecosystem and how they feel about new technologies, tools, platforms, the support from developer programs, and more. Following the closing of the survey, our expert analysts work to identify key trends and translate raw data into actionable insights that professionals and companies addressing a developer audience can utilize to fine-tune their strategy and address developers' needs and wants.

"The 26th edition of the Developer Nation survey reached more than 10,000 respondents from 135 countries around the world. SlashData announces the first two of a six-report series that are becoming widely available to the world, showcasing and diving into key developer trends for 2024 and beyond. Each report focuses on a specific topic. All reports published under the State of the Developer Nation will be accessible under the freshly launched SlashData Research Space, free access for viewing and downloading."

Okay. So the first two chunks are interesting. The first is how AI has impacted development. And the second is the ever popular "Which programming language do you use?" So first off, AI. They said: "How developers interact with AI technologies. Has AI taken over the world? Not yet," they write. "However, it has already achieved a takeover of all our discussions about the future." Indeed it has. "And," they said, "59% of developers report that they're now using AI tools in their development workflows. This report investigates the current landscape of developers' work with artificial intelligence technologies and how this impacts their careers. We start by looking at the ways in which developers work with machine learning models, tools, APIs, and services, and highlight the key differences between professional and amateur developers." And they go on.

So on the AI front we first have four broad categories. And Leo, I've got a chart at the top of page 9 of the show notes. Four broad categories. 59% report using AI in their own development workflows, 25% are adding AI functions into applications, and 13% are actively involved in creating AI models. This leaves only 29% whose development work has not yet been touched by AI in any of those ways.

Leo: Wow.

Steve: Yeah. Among the 59% - so more than half and fewer than two-thirds.

Leo: I'm in that category, by the way. I have an AI that helps me with my coding.

Steve: Yes, yes. So 59% who are now actively using AI tools in their development workflows. 42%, almost half, are using chatbots to obtain answers to coding questions. This is globally, out of 10,000, more than 10,000 developers surveyed. 42% are using chatbots to obtain answers to coding questions, 27% are using development tools that have AI-assistance built-in, and 19% are using generative AI to help generate creative assets.

If coding was a Monday through Friday, nine-to-five job which I was doing to earn my living, where I was being judged by my own productivity against my peers, then yeah, I'd be quite happy...

Leo: Yeah, yeah.

Steve: ...to get quick answers to questions about how to do this or that from a chatbot AI. You know, rather than searching around the Internet looking for someone, like wherever on the Internet.

Leo: Stack Exchange or somewhere, yeah.

Steve: Stack Exchange is the name I was just trying to remember, a Stack Exchange, who has posted something similar to learn from, I'd be happy to ask a smart bot what it had found from previously doing essentially the same thing. There's no shame there. And it's clear that many coders agree. Okay.

Leo: I use it instead of flipping through manuals.

Steve: Yes.

Leo: Almost universally the stuff that's on Stack Exchange is useless. But I still have to, I mean, I don't code enough to remember every - and the language I use is massive, Common LISP. So this is in lieu of looking through manuals. It's very useful.

Steve: Yup.

Leo: Very useful.

Steve: Yup. So what's going on with the use of programming languages? I have a chart there at the bottom of page 9. The survey revealed that by far the number one language in use today is JavaScript.

Leo: Oh, yeah. That's for sure.

Steve: Yup. Web programming. The current total is estimated to be 25.2 million JavaScript coders...

Leo: Wow. Wow.

Steve: ...with that number having grown by four million just over the past year. So 25.2 million JavaScript coders. In the number two slot is Python at 18.2 million.

Leo: There's probably a lot of overlap, too. I mean, nobody...

Steve: Yes.

Leo: ...uses just one language.

Steve: Right, right. Python at 18.2 million, which is just a bit ahead of JAVA at 17.7 million in third place. Behind those top three is C++ at 11.6, C# at 10.2, PHP at a respectable 9.8 million, Visual development tools at 7.2 million, followed by plain old 'C' language at 6.5. Then in steadily decreasing numbers we have Kotlin, Go, Swift, Rust, Dart, Objective-C, Ruby, and Lua. And you know, Leo, there's no sign of LISP or assembly language on this chart.

Leo: We're old-timers.

Steve: What do you suppose that means that neither of the two languages which you and I have chosen to use, LISP and Assembler respectively, are in the running here?

Leo: We're just smarter than the masses. That's all there is to it.

Steve: You know, I think part of it is that we're able to choose the language we most want to code in.

Leo: We get to choose. That's right. That's right.

Steve: We don't have any boss telling us, or an existing code base that we're having to maintain in whatever language.

Leo: Or colleagues who have to be able to read our code.

Steve: Right. Yes. And neither of us are part of a team that would think we had lost our minds.

Leo: But also there are very good modern languages that aren't on that list either. I think it really comes down to more trends, but also what your business is demanding of you.

Steve: Yes. I think, I mean, that's - I mean, we already know for decades, coders rsums have listed all the languages that they can, you know, that they're proficient in.

Leo: Right. Right. Any coder should be able to write in any language, if they're any good; right? I mean, it's...

Steve: Or be able to pick up a new one.

Leo: Yeah, that's what I mean. You can...

Steve: Yes.

Leo: Yes, yeah. All the concepts are the same.

Steve: And that's where a chatbot can help you.

Leo: Helps a lot.

Steve: It's like, okay.

Leo: That's right.

Steve: I'm not proficient in Perl, but I need to solve a Perl problem. So what regex would you expect to use?

Leo: I've actually done that.

Steve: And it may not be right. But it's...

Leo: It's a good start.

Steve: It's a place to start.

Leo: I've taken some Python code that I didn't fully understand, given it to ChatGPT and said, what would this look like in LISP? And, yeah, it wasn't perfect, but it gave me a big head start on understanding what that code was doing.

Steve: Yeah.

Leo: Yup.

Steve: So the question is, are we going to turn programming over to AI?

Leo: No. Well, eventually, I guess.

Steve: Well, coding appears to be something that AIs may be able to do. You know, and it makes a sort of sense for code to be something that an AI might do well because, after all, it's talking to a machine. So that begs the question, what's going on at the university level with computer science education? Business Insider published a piece last Monday titled "With AI writing so much code, should you still study computer science?" And the subheading was "This new data point provides an answer."

Okay, now, I realize that many of our listeners are well past university age, but many will have children - or perhaps grandchildren - who may be wondering whether coding has been lost to AI. So the author of this piece writes: "One of the most persistent concerns around generative AI is whether the technology will put workers out of a job. This idea has particularly caught on in the context of software coding. GitHub Copilot can write a lot of code these days, so is it even worth studying computer science now? That's been a question on the minds of math-minded high schoolers since ChatGPT burst onto the scene in 2022. There's a new data point that helps answer at least part of this question: Students are still lining up in droves to take computer science in college."

Let's take the University of California Berkeley as an example, as this college is at or near the top for computer science, as it was when I was there in '73. First-year applications to UC Berkeley's College of Computing, Data Science, and Society (CDSS) - now, that's not the college I was in. I was in EECS, Electrical Engineering Computer Science. But we have CDSS, the College of Computing, Data Science and Society.

Anyway, first-year applications increased 48% this year. There were 14,302 non-transfer applications for these CDSS majors in the Fall of 2024 incoming class, versus 9,649 the previous year. So in one year, 48% increase, they said; whereas, for context, the number of first-year applications to UC Berkeley as a whole did not change much from a year earlier. So it was specifically the College of Computing, Data Science, and Society. This was announced last week by Professor Jennifer Chayes, the dean of Berkeley's College of CDSS during the Joint California Summit on Generative AI in San Francisco.

Afterwards, Business Insider got in touch with an interesting guy, John DeNero, a Computer Science Teaching Professor at UC Berkeley, to talk about this some more. Now, he's also chief scientist at Lilt, a generative AI startup; and he was previously a researcher at Google working on Google Translate, one of the first successful AI-powered consumer apps.

Okay. So at this point the article quotes this John DeNero guy. And remember, he's a teaching professor of Computer Science at UC Berkeley who has been working with AI at Google and is now the chief scientist at a generative AI startup. So the article continues: "In an email to Business Insider, John wrote" - so this is John speaking - "'Students express some concern that generative AI will affect the software engineering job market, especially for entry-level positions. But they're still excited about careers in computing. I tell them that I think many of the challenging aspects of software development cannot be performed reliably by generative AI at this point, and that I expect there will still be a central role for human software developers long into the future.'"

So this is a Comp Sci professor, teaching professor at Berkeley, who's also deeply steeped in AI technology. The article says: "DeNero explained that generative AI is currently very good at replicating parts of software programs that have been written

many times before. But what if you want to create something new? This is where smart human coders will still be needed. This makes logical sense as AI models are trained on data. If that information doesn't exist yet, or it's not part of the training dataset, the models often get in trouble." Or as we say, "They just make it up."

DeNero said: "Generative AI requires a lot of thoughtful human intervention to produce something new, and all consequential software development projects involve quite a bit of novelty. That's the hard and interesting part of computing that currently requires clever and well-trained people. Generative AI can speed up the more mundane parts of software development, and software developers tend to adopt efficiency tools quickly." So this applies to what's happening at Lilt, which is building an AI platform for translators. "Google Translate first came out 18 years ago," they write, "and human linguists still have jobs and are relied upon when translations are really important. For instance, you could use Google Translate to read a Japanese train timetable, but would you use the app to translate your business's most important contract without having a human expert check it out? Probably not."

John said: "To reliably produce publication-quality translations, human expert linguists are still at the center of the process. But by using Lilt's task-specific generative AI models, those experts are much faster, more accurate, and more consistent. As a result, more text gets translated at higher quality into more languages." And they finish: "He expects this same pattern to play out in software development. A small team of highly trained human developers will have an even greater capacity to build useful, high-quality software."

DeNero finished by adding: "And so future Berkeley graduates will have plenty of opportunities to use their computing skills to improve the world. Hopefully some more of them will come work at Lilt." And I got a kick out of that because where better to recruit people for your own startup?

Leo: Your classes.

Steve: Yes, than teaching them and culling from the herd those...

Leo: The best, yeah.

Steve: ...that you want to have working for you. And Leo, it really does make sense. You know, I'm weird; right? I mean, we already know I code in assembly language. At the moment, I am creating the scaffolding to access COM API objects in my email server from assembly language.

Leo: Oh. That sounds painful.

Steve: Because it is unbelievably painful.

Leo: There would be absolutely libraries galore to do that in any higher level language.

Steve: And there are none. No one, as far as I know it's not been done...

Leo: Not in assembly, yeah.

Steve: ...in assembler. But I like it. And I also...

Leo: That's all that matters.

Steve: And I also spend a lot of...

Leo: You like it.

Steve: Yes, I like it. And how many times have I written a super fast sort algorithm of one type or another? I've written them and I've rewritten them because I like it.

Leo: It's fun.

Steve: It's like somebody who loves - like a woodworker building chairs.

Leo: Yeah, yeah.

Steve: It's like, I'm going to make - the next chair I build is going to be better than the last one. But, you know, I'm coding because I want to, not because I have to. So it totally makes sense to me that generative AI could be producing a bunch of the crap code that people have already written, not the new stuff, which is where the fun really is for most people who, you know, don't like building chairs over and over.

Leo: Yes. You can always buy a chair, but there's a satisfaction in building your own, absolutely.

Steve: Yup.

Leo: Yeah. And I would imagine anybody who's in a computer science program I would hope is there because they enjoy it, because they like it. They're not just there to get a job skill. I mean, that's a nice side benefit.

Steve: It certainly is nice to be able to spend your life doing something you love.

Leo: Do something you love. You'll be glad.

Steve: Okay. So, and this one affects you, Leo, as a Linux person. In case any of our Linux users notice and worry about a sudden torrent of CVEs emanating from the Linux Kernel Project, I wanted to assure everyone that the problem is with the underlying

issuing policies and is not reflective of any sudden collapse of the Linux's kernel code quality. Catalin Cimpanu, the editor of the Risky Business Newsletter, did some editorializing, but he drew the facts underlying his recent editorial from across the industry. So this is strongly based in what everybody who's looking at this going, what the hell is going on, is talking about. So I'm explaining this beforehand since I wanted everyone to understand that this is, you know, not just one grumpy guy's opinion.

Here's what he wrote last Wednesday. He said: "In February of this year" - get this - "The Linux Kernel Project was made an official CVE Numbering Authority" - that's called a CNA, a CVE numbering authority - "with exclusive rights to issue CVE identifiers for the Linux kernel. While initially this looked like good news," he wrote, "almost three months later, this has turned into a complete and utter disaster. Over the past months, the Linux Kernel team has issued thousands of CVE identifiers, with the vast majority being for trivial bug fixes and not just security flaws.

"In May alone, according to Cisco's Jerry Gamblin, the Linux team issued over 1,100 CVEs, a number that easily beat out professional bug bounty programs and platforms run by the likes of Trend Micro's Zero Day Initiative, Wordfence, and Patchstack. Ironically," he says, "this was a disaster waiting to happen, with the Linux team laying out some weird rules for issuing CVEs right from the moment it received its CNA status. We say 'weird' because they're quite unique among all CNAs. The Linux Kernel team argues that because of the deep layer where the kernel runs, bugs are hard to understand, and there is always a possibility of them becoming a security issue later down the line."

He said: "Direct quote below." This is the Linux Kernel team. "Note, due to the layer at which the Linux kernel is in a system, almost any bug might be exploitable to compromise the security of the kernel, but the possibility of exploitation is often not evident when the bug is fixed. Because of this, the CVE assignment team is overly cautious and assign CVE numbers to any bug fix that they identify. This explains the seemingly large number of CVEs that are issued by the Linux Kernel team." Wow.

He says: "While this looks good on paper, the reality is that other projects also manage similarly sensitive projects, but they don't issue CVEs for literally every possible bug fix. You don't see Intel and AMD issuing hundreds of CVEs with each firmware update. These projects vet reports to confirm that bugs pose a security risk before issuing a CVE and triggering responses with their customers, such as inventory asset scans and emergency patch deployments." In other words, CVEs have actual real-world consequences. They're not just to be used casually.

He says: "Instead, the Linux Kernel team appears to have adopted a simpler approach where it puts a CVE on everything and lets the software and infosecurity community at large confirm whether or not an issue is an authentic security flaw. If it's not, it's then up to the security and vulnerability management firms to file CVE revocation requests with the Linux Kernel team that's responsible for the affected component.

"Linux's new CNA rules also prohibit the issuance of CVE for bugs in EOL Linux kernels, which is also another weird take on security." He said: "Just because you don't maintain the code anymore doesn't mean attackers won't exploit it and that people wouldn't want to track it. The Linux team will also refuse to assign CVEs [whoops] until a patch has been deployed, meaning there will be no CVEs for zero-days or vulnerabilities that may require a longer reporting and patching timeline."

Leo: I think they do not know what CVE means.

Steve: Leo, that's nuts. I mean, it's like, if we don't admit that there's a problem, then Google can't start a clock forcing us to fix it.

Leo: Yeah.

Steve: So we're not going to issue it for a zero-day or vulnerabilities that may take a while to fix.

Leo: Wow.

Steve: I mean, you're right, it's like they don't at all understand what CVEs are for. Catalin said: "The new rules also create a confusing process of validating, contesting, and rejecting CVEs. I'm not going to go into all of that," he said, "since the venerable Brian Martin did a way better job back in February. Open Source Security's Bradley Spengler shared a real-world example last week of why the entire process of analyzing, validating, and revoking Linux CVEs is now a giant clusterf**k of confusion and frustration."

Catalin said: "We quote him: 'To say this is a complete disaster is an understatement. This is why CVEs should be for vulnerabilities, should involve actual analysis, and should provide that information in the CVE description, as any other responsible CNA would be doing.'" Catalin said: "Linux maintainer Greg Kroah-Hartman tried to justify the team's approach to its new CVE rules; but, as expected, this has not gone down well with those in the infosec community. Criticism has been levied against the Linux Kernel team from everywhere, and there have been some calls for the Linux team to reconsider their approach to issuing CVEs.

"The new rules were criticized right from the get-go. The likes of Katie Moussouris, Valentina Palmiotti, Ian Coldwater, Bradley Spengler (again and again), Adam Schaal, Tib3rius, the grsecurity team, the GrapheneOS team, and a whole bunch more, foresaw the disaster that is currently unfolding. And if this isn't bad enough, the Linux kernel team appears to be backfilling CVEs for fixes to last year's code, generating even more noise for people who use CVEs for legitimate purposes.

"Some described the Linux team's approach as 'malicious compliance' after the project was criticized for years for downplaying vulnerability reports and contesting CVEs assigned to its code by other CNAs. This may not be the case, as the new approach has some fans who see its merits, such as forcing more people to upgrade their kernels on a more regular basis." Meaning even if it's not necessary.

"The Linux CNA" - this is quoting somebody, he doesn't say who. "The Linux CNA intentionally adopts an overly cautious approach and assigns a new CVE when in doubt. While this may surprise many, it is a perfectly legitimate and entirely honest strategy. In contrast, vendors of proprietary software often tend to take the opposite approach, minimizing the assignment of CVEs whenever possible. Effectively managing the substantial number of CVEs involves understanding your kernel configuration, having a clear threat model, and ensuring the ability to update the kernel as needed. I hope that other large projects will eventually adopt Linux's approach."

And Catalin finishes: "Unfortunately, all of this CVE spam could have not happened at a worse time. Just as the Linux Kernel team was getting its CNA status, NIST was slowing down its management of the NVD database, where all CVEs are compiled and enriched. NIST cited a staff shortage and a sudden rise in the number of reported vulnerabilities, mainly from the IoT space. Having one of every fifth CVE being a Linux non-security bug

is not helping NIST at all right now." So unfortunately, we depend upon CVEs to convey true problems that require remediation of some kind. Having the Linux Kernel Project spewing CVEs for non-vulnerability bugs really is an abuse of the system.

Leo: Yeah. And they're creating a lot of noise, which is obscuring the real security issues.

Steve: Exactly.

Leo: Yeah.

Steve: Exactly. It has a real potential of just causing people - it's crying wolf; right?

Leo: Right, right. False alarm.

Steve: You're going to end up blunting the effect.

Leo: Yup.

Steve: What is not blunted, Leo, is the power of our sponsors.

Leo: I like your thinking. I like where you're going there, Mr. G. Now, back to Mr. Gibson. Steve Gibson, yes.

Steve: Okay. So GRC's email system continues to mature, and I could not be more pleased with my decision to create a more convenient means for our listeners to send podcast feedback. Some listeners have noted that nowhere on GRC's website do I prominently display the email address "securitynow@grc.com." That's true, and that's also deliberate. It's clearly not a secret, since Leo and I will be mentioning it every week here. But to whatever degree is possible I'd like to reserve inbound email to that mailbox for podcast feedback. There will be a temptation to send things to me that I already pay Sue and Greg to handle. So I'd prefer not to short-circuit our traditional lines of communication. So once again, securitynow@grc.com. And I imagine everybody can remember that.

I did want to let everyone know that, after last week's podcast, I improved GRC's email registration system to also accept email that's registered against a user's "From" header. The moment I made that change, all false positive rejections stopped. We haven't had a single one since then. So anyone who may have had initial difficulty registering with private domains fronted by Gmail or some other email anonymization service should no longer have any trouble and may do so. So again, I was a little overprotective initially. That's fixed. It's easier now.

Several people have been worried that they haven't ever received a single piece of email from me. You know, they're expecting the flow of weekly podcast announcements. So I wanted to assure everyone that, so far, I have never sent one. I'm still working to finish up the front-end email registration bounce processing, which I expect to complete this

week. I always wondered about the practice of asking people to enter their email addresses twice. I understood that it was to catch typos, and when I designed GRC's e-commerce system back in 2003, that's what I had it do, too. But I did that mostly because everyone at the time was doing it. Now I know why. The email registration system I have does NOT do that, and it's somewhat surprising to see how many typos are present in email that cannot be delivered. It turns out that ".vom" is not a valid top level domain, and that the V key is right next to the C key.

The good news is that such typos only result in a brief stumble, since this is part of an immediate email confirmation loop. So anyone who doesn't receive a confirmation email returns to try again, and they will probably enter their email correctly, and maybe by typing it more carefully, you know, the second time. Since I think that asking everyone only once, because they receive immediate confirmation, is more convenient for most people, I'm going to leave the system as it is. I'm not going to ask everybody to enter it twice.

The work I'm doing right now is in automating the process of receiving any immediate delivery attempt failures from our email server and holding that information for someone's second attempt when they don't get the first email confirmation and then come back and try again. There are a surprising number of "mailbox unknown" or "mailbox over quota" bounce-backs that I would like to be able to present to someone when they retry using an address that just failed for that reason. So once that system is in place, I'll actually begin sending email, and the system will be up and running.

Okay. So I got a kick out of this fictional dialog with an AI which was titled "Ordering a Pizza in 2024." This was shared by a listener via Twitter. There's no indication of the dialog's origin, but it's definitely worth sharing. So the caller says, apparently into their phone, "Is this Pizza Hut?" "No sir, it's Google Pizza."

Caller: "Oh, I must have dialed the wrong number, sorry." "No, sir. Google bought Pizza Hut last month." "Okay. I would like to order a pizza." "Do you want your usual, sir?" "My usual? You know me?" "According to our caller ID data sheet, the last 12 times you called you ordered an extra-large pizza with three cheeses, sausage, pepperoni, mushrooms, and meatballs on a thick crust." Caller says: "Super. That's what I'll have."

Google: "May I suggest that this time you order a pizza with ricotta, arugula, sun-dried tomatoes, and olives on a whole wheat gluten-free thin crust?" The caller says: "What? I don't want a vegetarian pizza." Google: "Your cholesterol is not good, sir." "How the hell do you know that?" "Well, we cross-referenced your home phone number with your medical records. We have the result of your blood tests for the last seven years." "Okay, but I do not want your rotten vegetarian pizza. I already take medication for my cholesterol."

"Well, excuse me, sir, but you have not taken your medication regularly. According to our database, you purchased only a box of 30 cholesterol tablets once at Lloyds Pharmacy, four months ago." Caller says: "I bought more from another pharmacy." "That doesn't show on your credit card statement." "I paid in cash." "But you did not withdraw enough cash, according to your bank statement." "I have other sources of cash." "That doesn't show on your latest tax returns, unless you bought them using an undeclared income source, which of course is against the law."

Caller says: "What the heck?" Google says: "I'm sorry, sir. We use such information only with the sole intention of helping you." The caller says: "Enough already. I'm sick of Google, Facebook, Twitter, WhatsApp and all the others. I'm going to an island without the Internet, TV, where there's no phone service and no one to watch me or spy on me." Google says: "I understand, sir, but you need to renew your passport first. It expired six weeks ago."

So anyway, yes. If Microsoft Recall does evolve into a semi-smart personal assistant, it better not start offering helpful advice, or maybe people will think about deleting it. Or using my forthcoming freeware app. We'll see.

Leo: End All Recall? I keep trying.

Steve: Leo, like I said, I'll just make sure you don't have a mouthful of coffee.

Nathan Hartley tweeted: "I would love Windows Copilot on my work PC, though we have far more local admins who have access to everything than I'm comfortable with. I will wait a bit for my personal PC." Now, of course, Nathan is suggesting that in a corporate environment, having access to a comprehensive history of everything that has been done on a company machine might be useful. But he wonders what access to that information would also be available to local admins.

And I think that's another very good point. All indications are that in their enthusiasm for this idea, which is understandable, Microsoft failed to give sufficient thought to just how transformative a change it would be for a machine's entire usage history to be captured and stored in detail. We know that enterprise machines are owned and operated by their companies who oversee them and their security. So how does Recall fit into that environment? There do appear to be some questions still to be answered.

Tom, who's @TomLawrenceTech - this was shared via a public Tweet, so it appeared in my timeline because he referenced @SGgrc. He said: "I just had a great conversation with @DRtheNerd about ADAMnetworks, @pfsense, and their 'Don't Talk to Strangers' system. I'll be doing some testing, but for those who want to learn more right now, check out https://adamnet.works and @SGgrc Episode 946," and he provided a link. Now, of course @DRtheNerd, that would be David Redekop, whom I first met when they were an early advertiser on this podcast, I think right from the get-go, Leo.

Leo: Yeah, very early on.

Steve: They were the Canada-based Nerds On Site guys at the time. And of course, as we know, David is now part of the team at ADAMnetworks, and I discussed their work during Episode 946 and noted that they have some very interesting and mature perimeter security technology, which is definitely worth looking at.

Okay. Listener John Liptak asked: "Steve, I've been caught up in the Google Domains to Squarespace DNS migration, and due to Squarespace's terms of service I want to move. However, due to the number of security issues with DNS as well as your wonderful testing software, I've been unable to find the episode where you give your recommendation for a domain name provider. Can you remind me who you recommend? Thanks, John."

The name John is trying to recall is "Hover.com." They are my absolute, hands-down, favorite domain name registrar. They were also a TWiT sponsor, though that followed my switching to them away from Network Solutions, who was GRC's original registrar, with whom I registered the GRC.com domain back in December of 1991, which was a few months after the domain Microsoft.com was first registered. I could not be more pleased and happy to recommend Hover as the place for anyone to hang their domain. I mean, again, I know that, Leo, you and I both have a ridiculous number of domains just because...

Leo: It's fun.

Steve: ...each one seems inexpensive, and maybe we'll use it someday for something.

Leo: Right, exactly.

Steve: I can't even tell you, like, the nonsense I have.

Leo: Oh, me, too.

Steve: But what the heck. They're all there at Hover. Steve in Tampa, Florida sent me a note regarding the Token2 keys that we've talked about a couple times. He said: "I just wanted to let you know that after hearing your mention of the Token2 keys on the podcast I ordered two of the T2F2-NFC-Dual keys. I received them today. I immediately downloaded the Windows app from their website and entered in a PIN. I then tried them with Bitwarden. After entering them in Bitwarden under WebAuthn, I was able to" - in other words, the Passkeys - "I was able to login in every case - USB-A, USB-C, and NFC - using either the web app or an Android phone. Of note is that to activate the key you need to squeeze contacts together and not just touch the contacts. Regards, Steve in Tampa, Florida."

So that's welcome feedback, and I'm glad that those Token2 keys were not a boondoggle. They really do look like solid solutions. The ones I ordered were backordered, and they've not shown up yet, but I'm not in a huge hurry.

Now, yesterday a listener, Bob Grant, wrote through the new email system with some of the best on-the-ground feedback about the current state of Passkeys support that I've seen so far. What Bob had to share was of crucial importance because it clearly dispels the belief that all websites which support Passkeys support multiple Passkeys, thereby allowing multiple physical dongles to be used without restriction. That's not the case. So here's Bob's great reporting.

He said: "Hi, Steve. I've always enjoyed trying out the bleeding edge, and I've been using YubiKeys for over a decade. So I recently replaced one of my YubiKeys with a Token 2 key from Switzerland to get its 100-passkey support. I then went about registering multiple YubiKeys and my new Token 2 keys plus Bitwarden at multiple sites. For instance, I have five Gmail accounts and two Microsoft accounts I wanted to use with the Passkeys. I discovered a few indications that we have a ways to go before this is ready or easy for prime time.

"For security purposes, all the hardware keys require a PIN to unlock the key for each login to a site. This is as opposed to Bitwarden, which will do it for you while the vault is unlocked; or, if locked, can use a biometric authentication, which is pretty quick. Further, the hardware token operation requires an initial touch to bring up the PIN prompt followed by another touch after the PIN to perform the authentication. The Token 2 keys require the FIDO-recommended six-digit PIN, whereas YubiKeys allow for a more convenient four-digit PIN. As usual, security trumps convenience.

"Next, I found that a bunch of sites do not follow the FIDO recommendations. eBay, PayPal, and Lowes only allow a single Passkey to be registered. This of course means you have to use something like Bitwarden that can sync between devices rather than a single hardware key, which is a point of failure. Kayak, LinkedIn, Adobe, and Amazon do not

allow naming the keys as you enroll them. LinkedIn calls them Passkey 1, 2, 3, et cetera. Amazon has the date, but not the time, the key was enrolled, so there's no way to differentiate unless you enroll on different days. The effect of this is that, if you need to revoke a key that is lost, you don't know which enrolled key should be deleted from the site. All other sites I used allowed naming at creation, and some even allow later renaming of enrolled keys.

"Most sites allow quite a few keys, but LinkedIn only allows five. Surprisingly, Amazon AWS seems to only allow FIDO1-style U2F mode keys, not FIDO2 for Passkey login. Many sites allow keys from one type of device, for example, iOS or iPadOS, but not another, like Firefox on a desktop. Chrome seems to have better support, and I think MS Edge has good support, although I didn't test extensively. Chrome allows managing keys, Token 2 or YubiKey, from its Settings > Security menu within the browser, and so you can list, delete, edit, et cetera.

"This all suggests that it's still early days, but I still kind of prefer my YubiKey to my Token 2 key, and I'm doubtful I'll get to 100 Passkeys anytime soon. The Token 2 is fatter and more bulky and at least feels a little more vulnerable than the YubiKey. Also, at one point my T2 stopped responding and prompting for a PIN when I tried to login. But I was able to use my YubiKeys without a problem. Once I rebooted my laptop, the Token 2 key resumed responding. I don't know whether the auth infrastructure would blacklist a key, but I'm going to keep an eye on it." Now, for the record, my guess is that the Token 2 Windows app probably froze somehow, and that that's what the reboot cured. And of course USB is always, because it came along after Windows had already launched and a lot of it been written, USB has always been a little bit flaky.

So anyway, he finishes: "I'd like to see PayPal allow multiple keys so I could switch to using a hardware key for added security. But I'll need to use Bitwarden with PayPal until then. It's disappointing to me that banks, investment houses, and other high-value targets do not currently support Passkeys at all. In fact, most are still using SMS text second factors rather than Google Auth or even the older U2F which could use keys for multiple-factor authentication. Hardware keys can also be used for SSH authentication for more security for your SSH sessions. Each one takes the same type of slot as a Passkey and can also can store the SSH key info which allows it to move the public key from system to system. It's easy to see what an uphill battle SQRL faced when even given all the support behind FIDO2, its implementation remains spotty and uncertain."

So, wow, thank you very much, Bob. That's some terrific feedback about the current state of Passkey support. And all of this does suggest that today's optimal solution - driven by the fact that there are sites which will only accept a single Passkey enrollment, and you never know when you're going to hit one - would be to enroll one or more, where possible, hardware dongles only for the highest security sites where that's what you want; but to then otherwise use a cross-platform password manager such as Bitwarden, a sponsor of the TWiT network, and use that hardware dongle to, in turn, unlock Bitwarden, if you want more than Bitwarden's biometric unlocking. In that fashion, any site's single Passkey support won't present a problem since Bitwarden is able to present that site's single Passkey from any Bitwarden-supported device. And now that its support for mobile devices is shipping, it's on all platforms everywhere.

A listener using his initials B.E. surprised me. He wrote: "Hi, Steve. Thank you for the new email system since I don't use any social media. Regarding Code Signing HSMs: My friend and I are on top of the development of a hobby software, used by only 15 to 20 people. We used to share the code signing keys between us and one other developer. But when I went to renew the code signing certificate, I saw there is no longer an option to be able to sign any code without an HSM. Do you have an idea how we can still have the three developers able to sign the code? Thank you for all your work. Long-time listener and a Club TWiT member, B.E."

Leo: Yay, thank you.

Steve: Okay. So this was news to me. In a follow-up note, B.E. sent some links so that I didn't need to track them down myself. And sure enough, reading from the knowledgebase maintained by my favorite certificate authority DigiCert, under the title "New private key storage requirement for Code Signing certificates," they write: "Starting on June 1, 2023, industry standards will require private keys for standard code signing certificates to be stored on hardware certified as FIPS 140 Level 2, Common Criteria EAL 4+, or equivalent." In other words, an HSM, some sort of a hardware dongle. "This change," they write, "strengthens private key protection for code signing certificates and aligns it with EV (Extended Validation) code signing certificate private key protection."

Wow. This is actually troubling. First, as I previously reported, the enhanced trust that Microsoft was originally conveying to any code signed with the significantly more expensive EV certs, which have always required storage in an HSM, has been revoked so that there is no longer any benefit to having an EV certificate for code signing. No one cares. But now the industry has moved to requiring ALL code signing to be performed inside, and by a hardware dongle. It was already a problem that any code signing was becoming a requirement, which is what we've been seeing due to the increasing prevalence of malicious code.

The problem is that many open source projects are hobby projects like that of our listener, which would otherwise not need to be signed. So this general signing requirement was already imposing a burden on developers. But now the stakes are raised even higher, requiring the purchase of hardware for the storage of any code signing key. And as a side effect, as our listener notes, this also prevents small teams of hobby developers from sharing a single certificate among themselves for the purpose of defraying and amortizing its cost across multiple users. And it's not as if this is a one-time event, since certificates expire and require periodic renewal. The hardware won't need renewing, but an updated certificate will need to be installed.

So what I expect will happen is that we'll start to begin seeing code signing servers appearing so that multiple members of a team distributed physically, geographically, will still be able to share a single HSM dongle among themselves. And when that happens, I sure hope they get their security right, since there will be tremendous pressure from malware authors to also get their malicious software signed by those same code signing servers.

Now, as we know, I wrote such a thing myself as part of SpinRite 6.1's launch, since everyone's SpinRite download is unique and needs to be individually signed. And I commented here a few months ago, when we learned that EV certs were losing their special treatment, that I had apparently wasted my time doing that because my next certificate would not be EV and would therefore not need to be contained within an HSM. It turns out my time was not wasted after all. Everyone who signs code will need to use an HSM to do so as soon as their current non-HSM code signing certificate expires. Wow. Yet another tax put on the good guys by malware. It's unfortunate.

And again, certainly nothing prevents anyone from automating the code signing process. As I found, due to Microsoft's pathetic documentation for doing this, for me it was a heavy lift. I got it working. It's been surprisingly, which is to say utterly bulletproof, since I finally finished it, and I'm thankful for that. But boy, I'm sure somebody will do it for Linux and open source it, and then code signing servers will be something we start to see.

And for what it's worth, B.E., it is possible to rekey a certificate for installation in a second hardware dongle. So if you ended up purchasing two dongles, one for each location, you could still only purchase one certificate key. The process of installing it does not give you any control over it, and this is by design, so that it can only get installed in a single device. But it can be immediately rekeyed and then installed into a second device. So at least you won't need to be doubling up on purchases if you have two sets of hardware. But, boy, it's very clear this is what the industry has done. And it's going to, you know, it's attacks on open source software, and I think it's really unfortunate, Leo.

Leo: Yeah, I agree, hundred percent. All right. I want to talk about code.microsoft.com. I didn't know anything happened to it. But you're going to tell us all about it. All right. What happened to code.microsoft.com?

Steve: So the page I ran across at Microsoft, and I don't recall how it came to my attention, has the intriguing title: "Examining the Deception Infrastructure in Place Behind code.microsoft.com." Okay. The Deception Infrastructure? What? Well, it turns out that the reader is not left to wonder long since this piece starts out: "The domain name code.microsoft.com has an interesting story behind it. Today it's not linked to anything, but that wasn't always true." And as a matter of fact, yesterday I did an NSLOOKUP and the domain, and there's no name resolution. So they completely disconnected it.

He writes: "This is the story of one of my most successful honeypot instances, and how it enabled Microsoft to collect varied threat intelligence against a broad range of actor groups targeting Microsoft. I'm writing this now as we've decided to retire this capability." Okay, now, that's not the good part. The astonishing part is how this got started.

So here's what he wrote: "Code.microsoft.com was an early domain used to host Visual Studio code and some helpful documentation. The domain was active until around 2021, when this documentation was moved to a new home. After the move, the site behind the domain was an Azure App Service site that performed redirection, thus preventing existing links from being broken. Then, sometime around mid-2021, the existing Azure App Service instance was shut down, leaving code.microsoft.com pointing to a service that no longer existed. This created a vulnerability.

"This situation is what's called a 'dangling subdomain,' which refers to" - which as far as I know Microsoft just made up, never heard that before, a dangling subdomain - "which refers to a subdomain that once pointed to a valid resource, but now hangs in limbo." Again, never - limbo's not a term. That's something you normally do where you have to like lean over backwards and get underneath a horizontal bar. I don't know, you know, limbo. Okay.

So he says: "Imagine a subdomain like blog.somedomain.com that's used to handle a blog application. When the underlying service is deleted - the blog engine - you might update your page link and assume the service has been retired. However, there is still a subdomain pointing to the blog." What? "This is now 'dangling' and cannot be resolved." Okay. He says: "A malicious actor can discover the dangling subdomain." Except no. It's a subdomain of your own domain. So a malicious actor, what do you mean they can discover it? Anyway, he says - this is what he said. "A malicious actor can discover the dangling subdomain, provision a cloud Azure resource with the same name, and now visiting blog.somedomain.com will redirect to the attacker's resource." What? He says: "Now they control the content."

He says: "This happened in 2021 when the domain was temporarily used to host a malware command-and-control service. Thanks to multiple reports from our great community, this was quickly spotted and taken down before it could be used. As a response to this, Microsoft now has more robust tools in place to catch similar threats."

Okay. So first of all, let me just say "holy crap," and I hope that no one listening to this while driving just lost control of their vehicle because this is nothing short of insane that that could happen. I'm not trained up on Azure, and on how or why it might be possible for a so-called "dangling subdomain" of Microsoft.com to be casually commandeered by someone not Microsoft, by giving their own Azure cloud instance the same name as an unassigned Microsoft subdomain. All I can surmise is that there must be some serious architectural design problems over in Microsoft land for that to ever have been possible. That's just nuts.

But in any event, this author continues by posing the rhetorical question: How did it become a honeypot? He says: "Today it's relatively routine for MSTIC to take control of an attacker-controlled resource and repurpose these for threat intelligence collection." Right? Like they'll take over some domain that was an existing command-and-control server and run it in order to gain intelligence. He wrote: "Taking control of a malware command-and-control environment, for example, enables us to potentially discover new infected nodes." Right. In other words, because the infected machines will be phoning home to the mothership for instructions.

So he says: "At the time of the dangling code" - you know, code.microsoft.com - "subdomain," he says, "this process was relatively new. We wanted a good test case to show the value of taking over resources versus taking them down. So instead of removing the dangling subdomain, we pointed it instead to a node in our existing vast honeypot sensor network." He says, and just for anyone who doesn't know, but everyone does: "A honeypot is a decoy system designed to attract and monitor malicious activity. Honeypots can be used to collect information about the attackers, their tools, their techniques, and their intentions. Honeypots can also be used to divert the attackers from the real targets to consume and waste their time and resources.

"Microsoft's honeypot sensor network has been in development since 2018. It's used to collect information on emerging threats to both our and our customers' environments. The data we collect helps us be better informed when a new vulnerability is disclosed and gives us retrospective information on how, when, and where exploits are developed. This data becomes enriched with other tools Microsoft has available, turning it from a source of raw threat data into threat intelligence. This is then incorporated into a variety of our security products. Customers can also get access to this via Sentinel's emerging threat feed.

"The honeypot itself is a custom-designed framework written in C#. It enables security researchers to quickly deploy anything from a single HTTP exploit handler in one or two lines of code, all the way up to complex protocols like SSH and VNC. For even more complex protocols we can hand off incoming connections to real systems when we detect exploit traffic and revert these shortly after. It is our mission to deny threat actors access to resources or enable them to use our infrastructure to create further victims. That's why in almost all scenarios the attacker is playing in a high-interaction simulated environment. No code is run. Everything is a trick or deception designed to get them to reveal their intentions.

"Substantial engineering has gone into our simulation framework. Today" - get this - "over 300 pseudo-vulnerabilities can be triggered through the same exploit proof-of-concepts available in places like GitHub and Exploit DB. Threat actors can communicate with over 30 different protocols and can even 'log in' and deploy scripts and execute

payloads that look like they're operating on a real system. There is no real system, and almost everything is being simulated."

Okay. So, wow. Let me just say "props where it's due," and it's definitely due here. That is some seriously cool technology. They've created "The Matrix" - a simulated, deliberately vulnerable environment that's designed to lure bad guys into believing that they've successfully exploited any of more than 300 known vulnerabilities on a machine, while retaining the control that the actual exploitation of the vulnerability was designed to bypass. So it looks like a duck, and it quacks like a duck, but it ain't no duck. Very, very cool tech.

So he continues: "It's important that in standing up a honeypot on an important domain like Microsoft.com it wasn't possible for attackers to use this as an environment to perform other web attacks, attacks that might rely on same-origin trust." Meaning they had to make sure that bad guys could not originate their attacks from inside Microsoft.com because that's where code.microsoft.com was. You can imagine that everybody knows what Microsoft.com networks are, and it would not be a stretch to imagine that there are some enterprises that have whitelisted Microsoft networks after having to put lots of individual whitelist IPs - some guy just said, oh, forget it, let's just whitelist the whole /16 or whatever Microsoft has. So again, origin of trust. They could not allow the origin to be Microsoft.com. So he said: "To mitigate this further, we added the sandbox policy to the pages which prevents these kinds of attacks."

So, he writes: "What have we learned from the honeypot? Our sensor network has contributed to many successes over the years. We've presented these at computer security conferences in the past, as well as shared our data with academia and the community. We incorporate this data into our security products to enable them to be aware of the latest threats. In recent years this capability has been crucial to understanding the zero-day and n-day ecosystem. During the Log4Shell incident we were able to use our sensor network to track each iteration of the underlying vulnerability and associated proof-of-concept all the way back to GitHub. This helped us understand the groups involved in productionizing the exploit and where it was being targeted. Our data enables internal teams to be much better prepared to remediate, and provides the analysis for detection authors to improve products like Microsoft Defender for Endpoint in real time." That's MDE.

"The team developing this capability also works closely with the MSRC who track our own security issues. When the Exchange ProxyLogon vulnerability was announced, we had already written a full exploit handler in our environment to track and understand, not just the exploit, but the groups deploying it. This kind of situational awareness enables us to give clearer advice to the industry, better protect our customers, and integrate new threats we are seeing into Windows Defender and MDE. The domain code.microsoft.com was often critical to the success of this, as well as a useful early warning system. When new vulnerabilities have been announced, threat actors can often be too consumed with trying to use the vulnerability as quickly as possible than checking for deception infrastructure like a honeypot. As a result, code.microsoft.com often saw exploits first. Many of these exploits were attributed to threat actors MSTIC already tracks."

Okay. So it is very interesting that the announcement of a new vulnerability immediately triggers a mass frenzy as - you know, we've talked about this effect before; right? - as attackers, who are literally everywhere, scurry to take advantage of it before machines are patched.

Okay. So the author continues: "What happened next?" He says: "The 'code' subdomain had been known to bug bounty researchers for several years. So whenever they would receive a report from someone who believed that they had discovered a critical vulnerability for this domain, these would be closed to let them know they had found a

honeypot. We've asked these security professionals to refrain from publishing details of this service in an effort to protect the value we received from it. We've also understood for a while that this subdomain would eventually need to be retired once its existence had become too well known to be of value. That time finally arrived.

"On April 25th, a sudden uptick in traffic to the subdomain, and posts on Twitter, revealed that the domain was being investigated by broad groups of individuals. Since this discovery meant that the secret was out, and the subdomain had lost its value, we decided to fully reveal the truth and retire the system." I have a chart in the show notes that shows this, where they're basically ticking along at almost nothing, and then over the course of a couple days the traffic just explodes.

He said: "The timeline gives an order of events from our perspective. It's unknown exactly how the full exploit URL of our server ended up in Google search database, but it looks like this, and the associated discovery on Twitter/X culminated in almost 80,000 WeChat exploits in a three-hour period. It's unlikely the Google crawler would have naturally found the URL. Our current theory is that a security researcher found this and submitted a report to Microsoft. As part of this process, either the Chrome browser or another app found this URL and submitted it for indexing."

Okay. So in other words, it's very difficult to keep anything a secret on the Internet. It's easy to imagine that Google would have set up Chrome to feed URLs back to them for bot-crawling indexing. That way, users of Chrome are unwittingly providing Google with links to index as a means for assuring that Google bots are able to discover everything, even things that are not pointed to by anybody else, as in this case. In this case they somehow discovered a secret that Microsoft had been trying to keep quiet for several years.

The timeline showed that in March the WeChat exploit appeared in Google search results for the first time. On April 15th, a redacted screenshot of an exploit mitigation was posted online, and some debate followed as to whether the domain was the code.microsoft.com subdomain. Six days later, on the 21st of April, Google trends show that many people were now searching for the "code" domains. Three days after that, on the 24th, they start noticing a significant uptick in traffic to the subdomain. And finally, on the 26th, they're hit with 126,000 times more requests than average.

They write: "By the 26th of April we were handling 160,000 requests per day, up from the usual between five and 100. Most of these requests were to a single endpoint handling a vulnerability in the WeChat Broadcast plugin for WordPress (CVE-2018-16283). This enabled anyone to 'run' a command from a parameter in the URL. Looking at these URLs, we found 11,000 different commands being attempted. Most of these pushed a message by some group or another stating that the site had been hacked by them. So just ego. This was a simulation, so nothing happened. Removing these messages gave a clearer picture of the kinds of commands people were entering.

"Most commands entered were Linux recon commands. These attempted to find out what the system was; what files it contained; and, more broadly, what value it was to Microsoft. The next biggest group were running command. These ranged from basic Linux commands like 'whoami,' but a few enterprising folks went on to run scripts of various languages. Most people who interacted didn't get further than the WeChat exploit. Over the three busiest days, 63 different exploits in total were triggered. The biggest surprise was that most researchers stuck to HTTP. Only three groups probed the other ports, and even fewer logged into the many other services that were available.

"Some of the best investigation came from a Twitter handle @simplylurking2 on Twitter/X who, after discovering that the system was a honeypot, continued to analyze

what we had in place and constructed, first constructing a Rickroll and then a URL that, when visited, would display a message to right-click and save a payload.

"With so much information now publicly available, the value of this subdomain was diminished. On April 26th we replaced the site with a 404 message and are working on retiring the subdomain completely. However, our ongoing data collection efforts are undiminished. Microsoft runs many of these collection services across multiple datacenters. Our concept has been proven, and we have rolled out similar capabilities at higher scales in many other locations worldwide. These continue to give us a detailed picture of emerging threats." So that's the story of the rise and fall of a honeypot.

Leo: What a story.

Steve: Which Microsoft inadvertently created, but then managed to put to great use and advantage for several good years before its identity finally leaked and was made public, thus rendering it useless. We've also seen how the tip of the iceberg for a honeypot is that it can detect that something is wrong on the network. That's generally sufficient for most purposes. But as we see, this can also be taken far beyond simple detection with a sufficiently advanced vulnerability simulator to reveal exactly what bad guys will do when they're given more rope to hang themselves.

Leo: I love it.

Steve: Wow.

Leo: I used to go to code.microsoft.com all the time to download VS code. I had no idea that they had abandoned it. That is a wild story. Wow.

Steve: Yeah. And Leo, again, the idea that, like, not having something responding to code.microsoft.com, which used to be hosted by Azure, allowed somebody else to register that. Again...

Leo: That's amazing.

Steve: I hope that somebody's looking at this architecture because something is broken, if that's possible.

Leo: You shouldn't be able to create a subdomain if the domain is owned by somebody else. That doesn't make any sense.

Steve: I know. I know. It's insane.

Leo: Is it still that way? I mean, should I worry about TWiT.tv, people getting subdomains on our own...

Steve: Are you on Azure?

Leo: No. Whew.

Steve: I'm not.

Leo: Oh, what a relief.

Steve: No, I mean, no one's ever heard of this. It's just nuts.

Leo: No. Crazy.

Steve: The only thing that - so code.microsoft.com must have pointed to an IP in Azure.

Leo: Right.

Steve: And something is rotten in Denmark. I've got nothing against Denmark. But, like, this - wow.

Leo: Shouldn't you blame the registrar? I mean, isn't that - or the DNS resolver? Isn't that...

Steve: No. No. It's somebody somehow created their own Azure instance. And because it was named code.microsoft.com...

Leo: That's all it needed was a name.

Steve: ...somehow it glued itself to that subdomain, which they called "dangling," a "dangling subdomain."

Leo: A dangling subdomain.

Steve: No one's ever heard of a dangling subdomain. Subdomains don't dangle.

Leo: Head up. What a story. Well, at least they got some good out of it; right? I think that's...

Steve: Yeah, it did. They turned it around. And the fact that it was in Microsoft, and they had some seriously cool tech, I mean, again, what I think must be the case, as I was reading this to our listeners, I was thinking why don't we get the sense in general that Microsoft is this good? I mean, there are parts of Microsoft that are really good.

Leo: They're smart people.

Steve: They're just buried so deeply down in the infrastructure that, you know, you just talk to morons on the surface.

Leo: Well, I don't even think it's that. I think it's just so complex that things fall through the cracks. So David Redekop was in...

Steve: Oh, Leo, just go to answers.microsoft.com, and you will swim in moron, the most moronic nonsense you have ever seen.

Leo: The intern did it.

Steve: Oh, my god.

Leo: It's the intern's fault. David Redekop, who we were just talking about, is in our Discord. I guess you're a Club member, thank you, David. He says Azure is the authoritative DNS for an Azure tenant. And there's your problem right there; right? They've decided. We don't need no stinking DNS service. We'll do it. We'll resolve the domain. We can do that.

Steve, you've always been full of fascinating stuff. Today was no exception. Another great episode of Security Now! in the vault. 978 done, 22 to go, and we can begin a new era as 1000, Episode 1000. Yay. You'll find Steve at GRC.com. You can email him there, securitynow@grc.com.