Security Now! #978 - 06-11-24

The rise and fall of code.microsoft.com

This week on Security Now!

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?

Security News

Last Friday, **Mary Jo Foley / @maryjofoley** (our Mary Jo of Windows Weekly fame)

> *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, and the tweaks are slated to take effect by then, too:*

She was referring to a Microsoft Windows blog posting by the Corporate Vice President for Windows + Devices. It appears that "Windows +" devices are a category. so that would presumably mean CoPilot+. He 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 in ***any*** system that aggregates, for all time, everything a user does on their PC. I'll have more to say 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 – so that their attitude can be seen. Bear with me at the start, since it's pure Microsoft marketing-speak. The posting reads:

> *Today, we are 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 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 their context.*
>
> *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 are 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 are 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 out of 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.*
>
> ***Listening to and acting on customer feedback***
>
> *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 Recall (preview) ships to customers on June 18.** [Yes, one week from now, next Tuesday.]*
>
> - *First, we are updating the set-up experience of Copilot+ PCs to give people a clearer choice to opt-in to saving snapshots using Recall. If you don't proactively choose to turn it on, it will be off by default.*
>
>   [THAT is a significant change... though we've also seen how persistent, seductive and eventually forceful Microsoft can be when they want to push their users in a certain direction. It's not difficult to imagine that while the user might need to switch it on, Microsoft won't be cautioning the user about the system's inherent dangers. Rather, they will be promoting the benefits and touting encryption, locality and security. The upshot will be that users will turn it on just to see.]
>
> - *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.*
>
> - *Third, we are adding additional layers of data protection including "just in time" decryption, protected by Windows Hello Enhanced Sign-in Security (ESS) so Recall snapshots will only be decrypted and accessible when the user authenticates. In addition, we encrypted the search index database.*
>
>   [I'll bite my tongue on points 2 and 3 until we're through hearing from this guy.]
>
> ***Secure by design and secure by default***
>
> *In line with Microsoft's SFI 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 (ESS), 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.*

Right. Remember that last week Kevin Beaumont deliberately used known-to-Microsoft infostealer malware, but Windows Defender was so slow to recognize the threat that the infostealer had already successfully exfiltrated the user's entire Recall history before Defender woke up and shut it down.

> *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.

> *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.*

Right. 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.

> *Some notable examples of security enhancements include:*
>
> *All Copilot+ PCs will be Secured-core PCs, 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.*
>
> [Wow! That's got to be secure!]
>
> *Microsoft Pluton security processor will be enabled by default 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 the PC.*
>
> *All Copilot+ PCs will ship with Windows Hello Enhanced Sign-in Security (ESS). This provides more secure biometric sign-ins and eliminates the need for a password.*
>
> ***Protecting your privacy on Copilot+ PCs***
>
> *In our early internal testing, we have seen different people use Recall in the way that works best for them. Some love the way it makes remembering what they've seen across the web so much easier to find than reviewing their browser history. Others like the way it allows them to better review an online course or find a PowerPoint. And people are taking advantage of the controls to exclude apps they don't want captured in snapshots, from communication apps or Teams calls, or to delete some or all their snapshots. This is why we built Recall with fine- grained controls to allow each person to customize the experience to their comfort level, ensuring your information is protected and that you are in control of when, what and how it is captured.*
>
> *Snapshots are stored locally. Copilot+ PCs have powerful AI that works on your device itself. No internet or cloud connections are used to store and process snapshots. Recall's AI processing happens exclusively on your device, and your snapshots are kept safely on your local device only. Your snapshots are yours and they are not used to train the AI on Copilot+ PCs.*
>
> *Snapshots are not shared. Recall does not send your snapshots to Microsoft. Snapshots are not shared with any other companies or applications. Recall doesn't share snapshots with other users who are signed into the same device, and per-user encryption ensures even administrators cannot view other users' snapshots.*

> *You will know when Recall is saving snapshots. You'll see Recall pinned to the taskbar when you reach your desktop. You'll have a Recall snapshot icon on the system tray letting you know when Windows is saving snapshots.*
>
> *Digital rights managed or InPrivate browsing snapshots are not saved. Recall does not save snapshots of digital rights managed content or InPrivate browsing in supported web browsers.*
>
> *You can pause, filter and delete what's saved at any time. You're always in control of what's saved as a snapshot. You can disable saving snapshots, pause them temporarily, filter applications and websites from being in snapshots, and delete your snapshots at any time.*
>
> *Enterprise and customer choice. For customers using managed work devices, your IT administrator is provided the control to disable the ability to save snapshots. However, your IT administrator cannot enable saving snapshots on your behalf. The choice to enable saving snapshots is solely yours.*
>
> ***Empowering people with experiences they can trust***
>
> *We are on a journey to build products and experiences that live up to our company mission to empower people and organizations to achieve more, and are driven by the critical importance of maintaining our customers' privacy, security and trust. As we always do, we will continue to listen to and learn from our customers, including consumers, developers and enterprises, to evolve our experiences in ways that are meaningful to them.*
>
> *We are excited for the upcoming launch of Copilot+ PCs on June 18 and for the innovative new features and benefits this entirely new category of PCs will bring. We will continue to build these new capabilities and experiences for our customers by prioritizing privacy, safety and security first. We remain grateful for the vibrant community of customers who continue to share their feedback with us.*

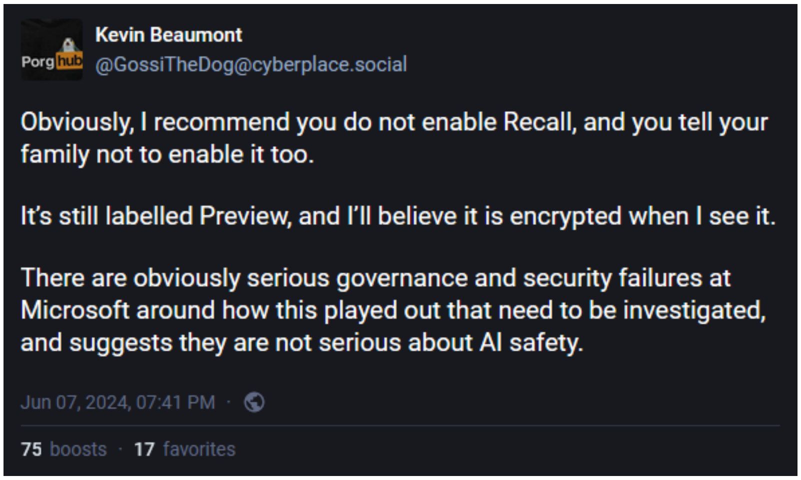
Okay. One thing I'm not sure I've made sufficiently clear about my own feelings about this is that it doesn't matter at all who is in charge of this data, how that data is stored, how it's encrypted, what access controls have been placed on it. The only thing that matters is that such an aggregation of this information exists at all. While it exists its owner's privacy is at risk. Period. Full stop. The existence of the information itself is the risk.

Microsoft has heard the industry screaming about this, and their answer is to tout the strength and thickness of the walls they've erected to contain this trove of secrets. It's all they can say, since they're clearly determined to collect, index and aggregate this trove of personal data. As I said last week, I'll bet that scrolling back through a timeline is only the beginning of what Microsoft plans to do with this data.

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. Because 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? 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. If so, I know that I'll call it, and Leo, I'll be sure you're not sipping coffee when I reveal its name, otherwise you'll need to be drying your screens.

Kevin Beaumont also weighed-in on Microsoft's revised explanation, he posted to his Mastodon:



Kevin Beaumont
@GossiTheDog@cyberplace.social

Obviously, I recommend you do not enable Recall, and you tell your family not to enable it too.

It's still labelled 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 AI safety.

Jun 07, 2024, 07:41 PM ·

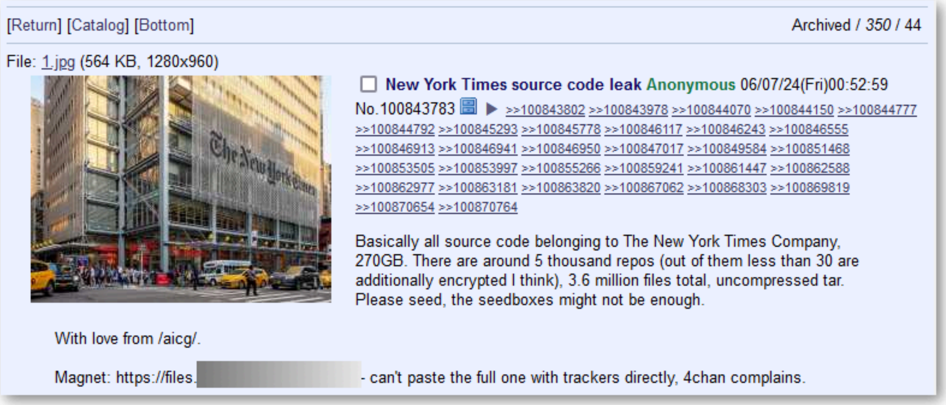
75 boosts · 17 favorites

Thanks for the "Memory"

I should mention that Google, with their ChromeOS, 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 ChromeOS, said that their "Memory" feature, as they're currently unofficially calling it, is different from Recall because users will have control of how and where the "memory" feature works. Uh huh. Right. Just like Recall will offer. So not so different from Recall, despite the fact that Google apparently already wants to distance itself from the stink surrounding the announcement of Recall.

New York Times (and Wordle) leak

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 – just like Microsoft is absolutely and positively certain they're going to secure their users' Recall data – nevertheless got loose. An unknown threat actor leaked the New York Times source code, as in, all of it. All 270 gigabytes 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 public website, mobile apps, and even its Wordle game. The 270 gigabytes of data being made available on the darkweb is mostly unencrypted. The hacker posted: *"Basically all source code belonging to The New York Times Company. 270GB. There are around 5 thousand repos (out of them, less than 30 are additionally encrypted I think), 3.6 million files total, uncompressed tar."*



[Return] [Catalog] [Bottom] Archived / *350* / 44

File: 1.jpg (564 KB, 1280x960)

□ **New York Times source code leak** Anonymous 06/07/24(Fri)00:52:59 No.100843783 ▶ >>100843802 >>100843978 >>100844070 >>100844150 >>100844777 >>100844792 >>100845293 >>100845778 >>100846117 >>100846243 >>100846555 >>100846913 >>100846941 >>100846950 >>100847017 >>100849584 >>100851468 >>100853505 >>100853997 >>100855266 >>100859241 >>100861447 >>100862588 >>100862977 >>100863181 >>100863820 >>100867062 >>100868303 >>100869819 >>100870654 >>100870764

Basically all source code belonging to The New York Times Company, 270GB. There are around 5 thousand repos (out of them less than 30 are additionally encrypted I think), 3.6 million files total, uncompressed tar. Please seed, the seedboxes might not be enough.

With love from /aicg/.

Magnet: https://files.[redacted] - can't paste the full one with trackers directly, 4chan complains.

The lesson here is that mistakes happen. We've seen stories of valuable exposed credentials sitting unnoticed for years. One real concern for the future is that there may soon be, if there aren't already, malicious AI-driven bots continually scanning and rifling through the Internet looking for any mistakes of value that anyone may have made. The world is changing... and I'm not sure the good guys are winning. This feels somewhat asymmetric.

Apple's own password manager app

During yesterday's WWDC, Apple's World Wide Developer Conference 2024 opening, Apple introduced their forthcoming Passwords app. Apple users have long been using their iCloud account to store and sync passwords, but what was going on wasn't super transparent. It just worked but without a clear and clean UI. 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 releases of their OSs, so iOS 18, macOS Sequoia and visionOS 2, will provide a UI for Apple's storage of this information. Since this is not a cross-ecosytem solution – it's Apple only – those of us who are also using Windows, Linux or Android platforms will likely remain with whatever cross-ecosystem solution we're using today. But this move does create an explicit and native password manager for Apple OSs 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 OTP-style authenticator. Since I'm currently using "OTP Auth" as my OTP authenticator of choice, I'll look at what Apple has once I upgrade to an iPhone that will run iOS 18 or later.

DJI drones on the defensive

I saw some talk a while back about some US congressional pushback on Chinese-made drones by DJI, which are by far the best drone technology available. In advance of the US 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 drone flight data to its servers and the option to sync U.S. drone data will be removed completely by the end of the month.

SlashData reveals some interesting developer statistics

I wanted to share some interesting results of a recent survey of more than 10,000 developers from more than 135 countries. The question put to them was "how has AI affected your workflow?” Let me first allow SlashData to introduce themselves. They wrote:

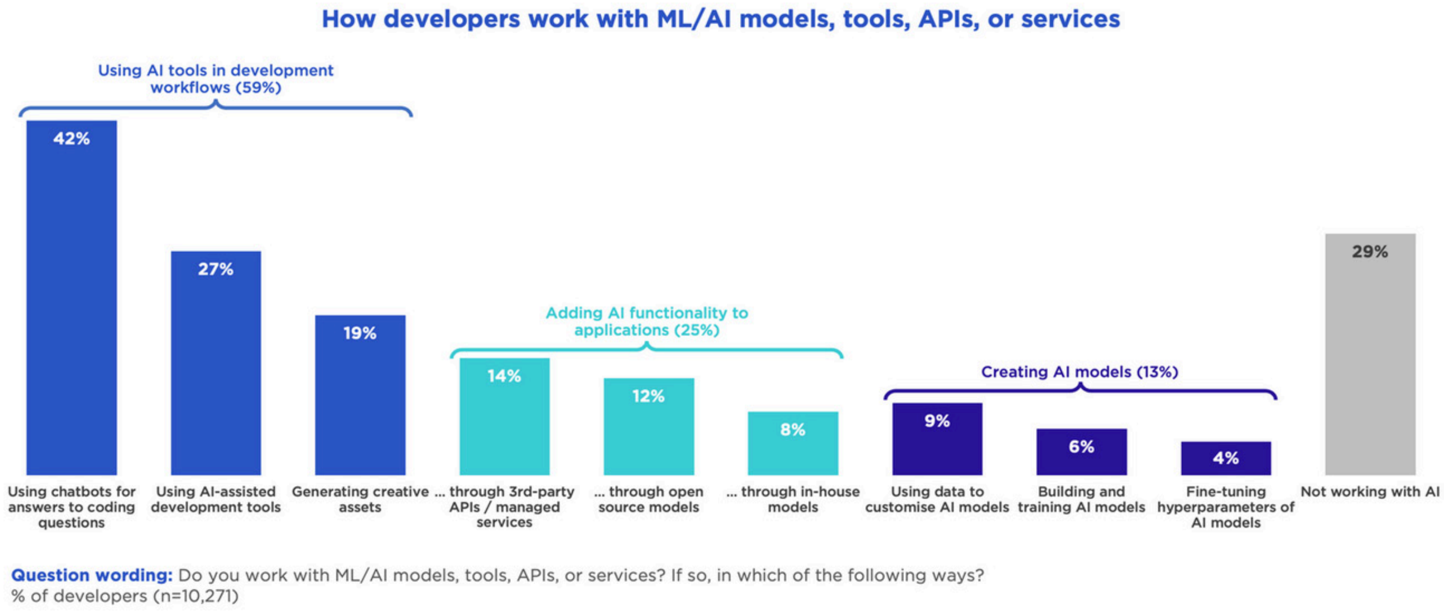
> *If this is the first time you heard about SlashData, I’m happy to share a few quick words. SlashData is a developer research company. Every quarter, SlashData runs a survey on the globe developer audience, to measure the pulse of the developer ecosystem and how they feel about new technologies, tools, platforms, the support from developer programs and more. Following the closing of the survey, our expert analysts work on identifying 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 2 of the 6-report series that are becoming publicly 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 to access, view, and download.*

Okay, so these 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:

> ***Developer Research Report: How developers interact with AI technologies***
>
> *Has AI taken over the world? Not yet. However, it has already achieved a takeover of all our discussions about the future, and 59% of developers reports that they are now using AI tools in their development workflows.*
>
> *This report investigates the current landscape of developers' work with artificial intelligence (AI) technologies and how this impacts their careers. We start by looking at the ways in which developers work with machine learning (ML)/AI models, tools, APIs, and services and highlight the key differences between professional and amateur developers.*
>
> *Following this, we focus on professional developers and explore the correlation between working with AI and self-perceived promotion opportunities at their current jobs. Finally, we take a closer look at the developers who are the most likely to express intent of quitting or changing jobs in the next 12 months.*
>
> *To understand the current landscape, we asked developers about the ways in which they work with ML/AI models, tools, APIs, or services. We find that 71% of all developers are actively working with AI in one way or another. The most popular use was AI workflows, where 42% of all developers reported using chatbots to answer coding questions.*

So on the AI front (see next page), we first have four broad categories: 59% report using AI in their own development workflows, 25% are adding AI functions into applications, and13% 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.

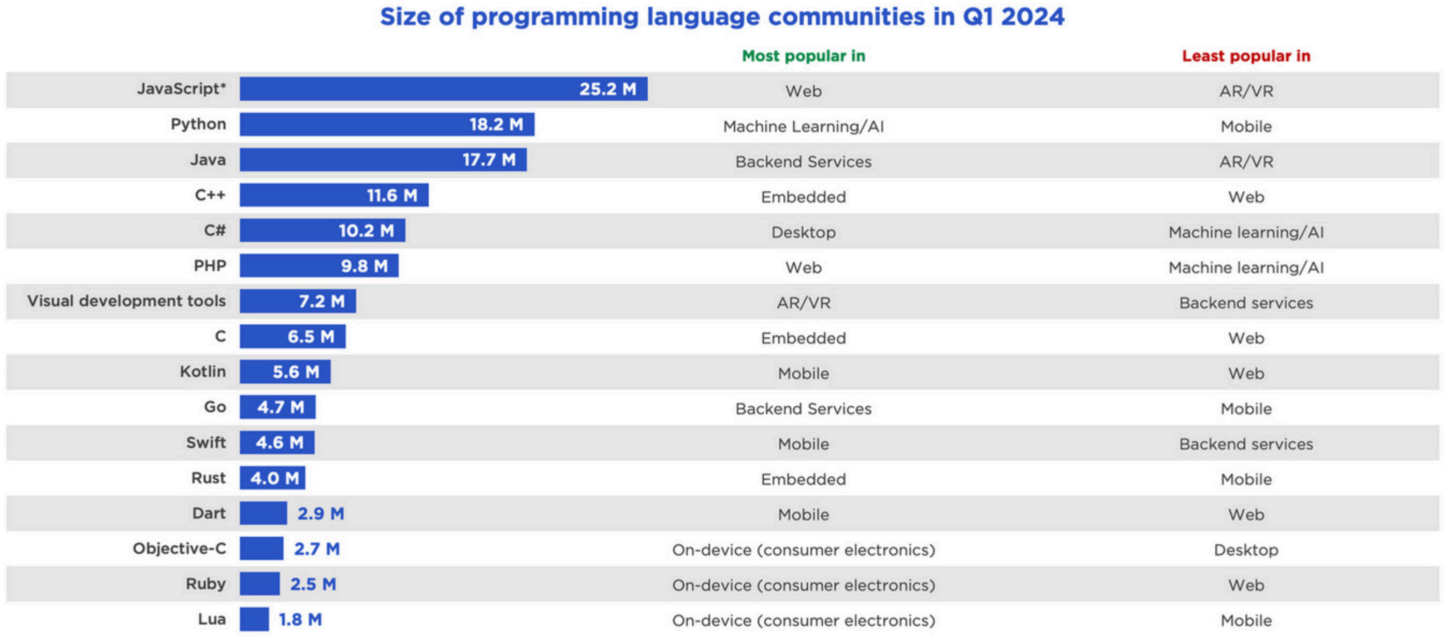
Among the 59% – so more than half and fewer than 2/3rds – but still many, who are now actively using AI tools in their development workflows, 42%, almost half, are using chatbots to obtain answers to coding questions, 27% are using development tools that have AI-assistance built-in, and 19% are using generative AI to help generate creative assets.



https://www.slashdata.co/post/59-of-developers-use-ai-tools-25-2m-javascript-users

If coding was a Monday through Friday, 9-to-5 job which I was doing to earn my living, where I was being judged by my productivity against my peers, then hell yes, I'd be quite happy to get quick answers to questions about how to do this or that from an AI chatbot. Rather than searching around the Internet looking for someone who has posted something similar to learn from I'd be happy to ask a smart bot what it had found from previously doing the same thing. There's no shame there. And it's clear that many coders agree.

Okay, so what's being going on with the use of programming languages?


Size of programming language communities in Q1 2024

		Most popular in	Least popular in
JavaScript*	25.2 M	Web	AR/VR
Python	18.2 M	Machine Learning/AI	Mobile
Java	17.7 M	Backend Services	AR/VR
C++	11.6 M	Embedded	Web
C#	10.2 M	Desktop	Machine learning/AI
PHP	9.8 M	Web	Machine learning/AI
Visual development tools	7.2 M	AR/VR	Backend services
C	6.5 M	Embedded	Web
Kotlin	5.6 M	Mobile	Web
Go	4.7 M	Backend Services	Mobile
Swift	4.6 M	Mobile	Backend services
Rust	4.0 M	Embedded	Mobile
Dart	2.9 M	Mobile	Web
Objective-C	2.7 M	On-device (consumer electronics)	Desktop
Ruby	2.5 M	On-device (consumer electronics)	Web
Lua	1.8 M	On-device (consumer electronics)	Mobile

https://www.slashdata.co/post/59-of-developers-use-ai-tools-25-2m-javascript-users

The survey revealed that by far the #1 language today is JavaScript. The current total is estimated to be 25.2 million JavaScript coders with that number having grown by 4 million over the past year. So, 25.2 million for JavaScript. In the #2 slot is Python at 18.2 million which is just a bit ahead of JAVA at 17.7 million in 3rd 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 million. Then, in steadily decreasing numbers we have Kotlin, Go, Swift, Rust, Dart, Objective-C, Ruby and Lua.

You know Leo, there's no sign of LISP or assembly language in this chart. What do you suppose it means that neither of the two languages which you and I have chosen to use, LISP and Assembler, respectively, are in the running here? Hmmm. I suppose it's that we're both able to choose the language we most want to code in, we don't have any boss telling us, or existing codebase to maintain in whatever language, and neither of us are part of a team that would certainly think we had lost our minds!

Are we going to turn programming over to AIs?

Coding appears to be something that AIs may be able to do. And it makes a sort of sense for code to be something that an AI might do well. 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? This new data point provides an answer."* 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 on 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. First-year applications to UC Berkeley's College of Computing, Data Science, and Society – CDSS – increased 48% this year. There were 14,302 (non-transfer) applications for these CDSS majors in the Fall 2024 incoming class, versus 9,649 the previous year. Whereas, for context, the number of first-year applications to UC Berkeley as a whole didn't change much from a year earlier. 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 John DeNero, a Computer Science Teaching Professor at UC Berkeley, to talk about this some more. 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.*

As this article is quoting his John DeNero guy, remember that 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: "Students express some concern that generative AI will affect the software engineering job market, especially for entry-level positions, but they are 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."*
>
> *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.*
>
> *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."*
>
> *This applies to what's happening at Lilt, which is building an AI platform for translators.*
>
> *Google Translate first came out 18 years ago. 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? 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."*
>
> *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 for Lilt."*

So this might be what we would think, and maybe even hope. But I think it's more than presumption. And it makes sense.

The Linux Kernel Project goes CVE crazy

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 kernel's code quality. Catalin Cimpanu ('ka -ta-lin Sim-'pa-new), the editor of the Risky Business Newsletter drew the facts underlying his recent editorial about this from across the industry. I’m explaining this beforehand since I wanted everyone to understand that this is far from being one grumpy person’s opinion. Here’s what Catalin wrote last Wednesday:

> *In February of this year, The Linux Kernel project was made an official CVE Numbering Authority (CNA) with exclusive rights to issue CVE identifiers for the Linux kernel. While initially this looked like good news, 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/platforms run by the likes of Trend Micro ZDI, Wordfence, and Patchstack. Ironically, this was a disaster waiting to happen, with the Linux Kernel team laying out some weird rules for issuing CVEs right from the moment it received its CNA status.*
>
> *We say weird because they are 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. Direct quote below:*
>
> > *"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 bugfix that they identify. This explains the seemingly large number of CVEs that are issued by the Linux kernel team."*
>
> *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 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.*
>
> *Instead, the Linux Kernel team appears to have adopted a simpler approach where it puts a CVE on* ***everything*** *and lets the software and infosec community at large confirm whether or not an issue is an authentic security flaw. If it's not, it's 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 CVEs for bugs in EOL Linux kernels, which is also another weird take on security. 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 until a patch has been deployed, meaning there will be no CVEs for 0-days or vulnerabilities that may require a longer reporting and patching timeline.*
>
> *The new rules also create a confusing process of validating, contesting, and rejecting CVEs. I'm not going to go into all of that 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. 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."*
>
> *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 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 backfiling 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. That 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.*
>
> > *"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."*
>
> *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 isn't helping NIST at all right now.*

Wow. Unfortunately, we depend upon CVEs to convey true problems that require remediation of some kind. Having the Linux Kernel project issuing CVEs for non-vulnerability bugs really is an abuse of the system.

Email @ GRC

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. It'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.

I wanted 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. So anyone who may have had difficulty initially registering with private domains fronted by gmail or other email anonymization services should no longer have any trouble and may do so.

Several people have been worried that they haven't ever received a single piece of email from GRC. 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 "dot 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 address correctly the next 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. The work I'm doing right now is in automating the process of receiving any immediate delivery attempt failures and holding that information for someone's second attempt. There are a surprising number of "mailbox unknown" or "mailbox over quota" bounces that I would like to be able to present to someone when they retry using the an address that just failed for that reason. Once that system is in place we'll begin actually sending email.

Closing The Loop

I got a kick out of this fictional dialog with an AI 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 is definitely worth sharing: **sentinalitqld / @sentinalitqld**

> ***Ordering a Pizza in 2024***
>
> *CALLER: Is this Pizza Hut?*
> *GOOGLE: No sir, it's Google Pizza.*
> *CALLER: I must have dialed a wrong number, sorry.*
> *GOOGLE: No sir, Google bought Pizza Hut last month.*
> *CALLER: OK. I would like to order a pizza.*
> *GOOGLE: Do you want your usual, sir?*
> *CALLER: My usual? You know me?*
> *GOOGLE: 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: 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?*
> *CALLER: What? I don't want a vegetarian pizza!*
> *GOOGLE: Your cholesterol is not good, sir.*
> *CALLER: How the hell do you know that?*
> *GOOGLE: Well, we cross-referenced your home phone number with your medical records. We have the result of your blood tests for the last 7 years.*
> *CALLER: Okay, but I do not want your rotten vegetarian pizza! I already take medication for my cholesterol.*
> *GOOGLE: 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, 4 months ago.*
> *CALLER: I bought more from another Pharmacy.*
> *GOOGLE: That doesn't show on your credit card statement.*
> *CALLER: I paid in cash.*
> *GOOGLE: But you did not withdraw enough cash according to your bank statement.*
> *CALLER: I have other sources of cash.*
> *GOOGLE: That doesn't show on your latest tax returns, unless you bought them using an undeclared income source, which is against the law!*
> *CALLER: WHAT THE HECK?*
> *GOOGLE: I'm sorry sir, we use such information only with the sole intention of helping you.*
> *CALLER: 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 is no phone service and no one to watch me or spy on me.*
> *GOOGLE: I understand sir, but you need to renew your passport first. It expired 6 weeks ago.......*

Yikes. If Microsoft Recall does evolve into a semi-smart personal assistant it better not start offering helpful advice.

Nathan Hartley / @Treestryder

> *I would love Windows Copilot on my work PC. Though, we have far more local admins, who have access to everything, than I am comfortable with. I will wait a bit for my personal PC.*

Nathan is suggesting that in a corporate environment having access to a comprehensive history of everything that had been done on a company machine might be useful. But he wonders what access to that information would also be available to local administrators. And I think that's another very good point. All indications are that in their enthusiasm for this idea, 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 to be answered.

Tom / infosec.exchange/@tomlawrence / @TomLawrenceTech

This was via a public Tweet which appeared in my timeline since he mentions @SGgrc.

> *I just had a great conversation with @DRtheNerd about AdamNetwork, @pfsense and their "Don't Talk To Strangers" system. I will be doing some testing, but for those who want to learn more right now, check out https://adamnet.works and @SGgrc episode #946 https://grc.com/sn/sn-946.htm*

@DRtheNerd is David Redekop, whom I first met when they were an early advertiser on this podcast with their Canada-based Nerds On Site. As we know, David is now part of the team at Adam Networks and, as I discussed during episode #946, they have some very interesting and mature perimeter security technologies.

Listener John Liptak asked:

> *Steve, I've been caught up in 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 have been unable to find the episode where you give your recommendation for a domain name provider. Can you remind me who you recommended? 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 from Network Solutions, who was GRC's original registrar, with whom I registered the GRC.COM domain back in December of 1991, a few months after the domain "microsoft.com" was first registered. I could not be more pleased with and happy to recommend HOVER as the place for anyone to hang their domain.

Steve in Tampa, FL — *re: Token2 keys*

> *I just wanted to let you know that after hearing you mention the Token2 keys on the podcast I ordered two to 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 login in every case. (USB A, 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.*

That's welcome feedback and I'm glad they were not a boondoggle. They really do look like solid solutions. The ones I ordered were backordered and they haven't shown up yet. But I'm not in a huge hurry.

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 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. Here's Bob's great reporting:

> *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 5 Gmail accounts and 2 Microsoft accounts I wanted to use with the Passkeys). I discovered a few indications that we have a ways to go before this is easy or ready 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 6 digit PIN whereas Yubikeys allow a more convenient 4-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 HW 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 etc. Amazon has the date (but not time) the key was enrolled, so there's no easy 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 5. 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 (e.g. iOS or iPadOS) but not another (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 (listing, deleting, editing, etc).*
>
> *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 Token2 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 trying to login. But I was able to use my Yubikeys without problem. Once I rebooted my laptop the Token2 key resumed responding. I don't know why the auth infrastructure would 'blacklist' a key but I'm going to keep my eye on it.*

(My guess is that the Token 2 Windows app froze and that's what the reboot cured.)

> *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. In fact most are still using SMS texted second factors rather than Google Auth or even the older U2F which could use keys for Multi-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 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, some terrific feedback about the current state of Passkey support. Thank you, Bob! All this suggests that today's optimal solution – driven by the fact that there are sites which will only accept a single Passkey enrolment – would be to enroll one or more (where possible) hardware dongles **only** for the highest security sites where that's what you need. 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 it's 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 15-20 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 3 developers able to sign the code. Thank you for all your work, long-time listener and a clip twit member, B E*

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, at 00:00 UTC, 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. This change 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. But now the industry is 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 them for the purpose of defraying and amortizing its cost. And it's not as if even 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.

What I expect will happen is that we'll begin to see code signing servers appear so that multiple members of a team located in physically distributed locations will be able to share a single HSM dongle. 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.

As we know, I wrote such a thing myself as part of SpinRite v6.1'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 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.

Wow. Yet another tax put on the good guys by malware.

Okay... what I didn't mention about GRC's email system is that its unveiling produced a flood of email. As of today at 10am, I've received 339 pieces of email from our listeners. You all know who you are, so one thing I wanted to do was to acknowledge that your messages have been received. I am determined to read every one of them, but I've run out of time and room to share more feedback this week because I want to share the backstory behind a domain that no longer exists after which today's podcast was titled: *"code.microsoft.com"*.

The rise and fall of code.microsoft.com

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**"*.

"The Deception Infrastructure" — What? Well, it turns out that the reader is not left to wonder for 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 that's true. I did an NSLOOKUP and that subdomain of microsoft.com has no name resolution.] *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."*

> *'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 AppService site that performed redirection thus preventing existing links from being broken. Then, sometime around mid-2021 the existing Azure AppService instance was shutdown 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 a subdomain that once pointed to a valid resource but now hangs in limbo. Imagine a subdomain like blog.somedomain.com that 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, this is now "dangling" and can't be resolved. 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. Now they control the content.*
>
> *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.*

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 it's 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 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 sub domain. 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?

Today it's relatively routine for MSTIC to take control of attacker-controlled resources and repurpose these for threat intelligence collection. Taking control of a malware Command and Control environment, for example, enables us to potentially discover new infected nodes.

Right. Because the infected machines will be phoning home to the mothership for instructions.

At the time of the dangling code subdomain 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.

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 deployed.

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 over 300 "pseudo-vulnerabilities" vulnerabilities can be triggered through the same exploit proof-of-concepts available in places like GitHub and exploitdb. Threat actors can communicate with over 30 different protocols and can even 'log in' and deploy scripts and execute payloads that look like they are operating on a real system.

There ***is*** *no real system... and almost everything is being simulated.*

Okay. So 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 have 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. He continues...

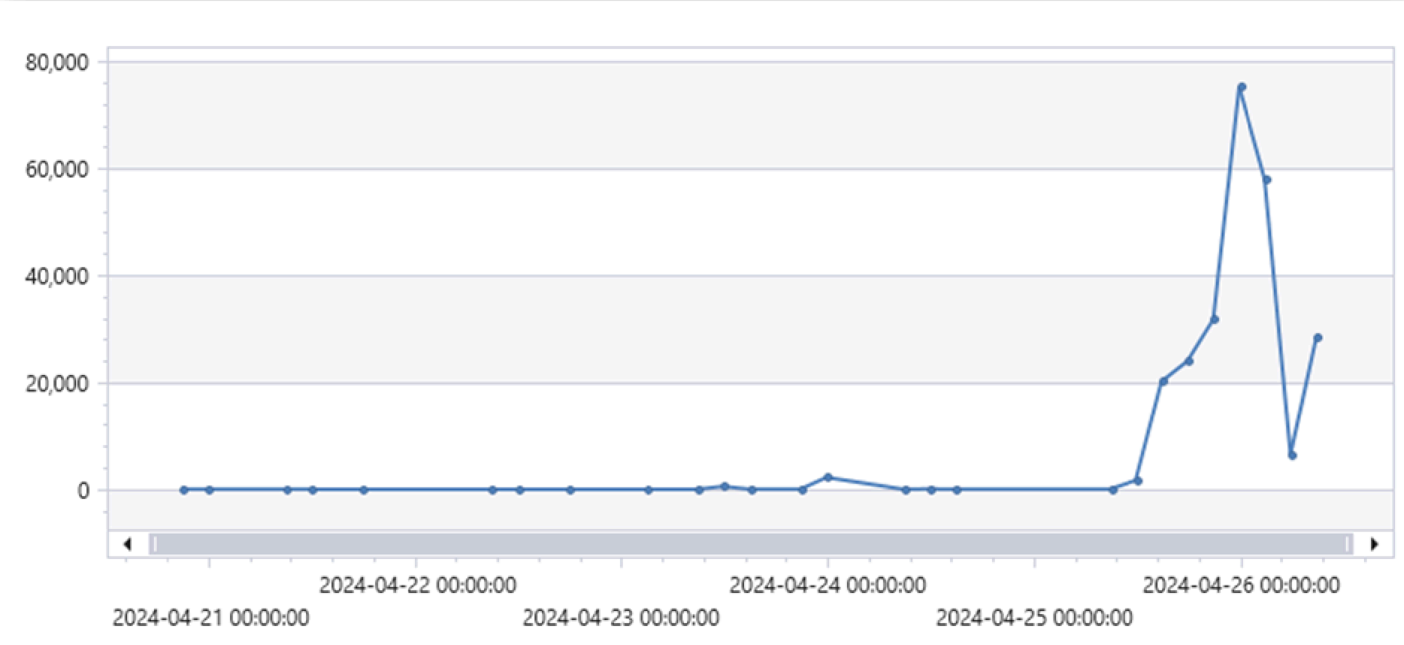
> *It’s important that in standing up a honeypot on an important domain like Microsoft.com it wasn’t possible for attackers to use this as an environment to perform other web attacks. Attacks that might rely on same origin trust. To mitigate this further we added the sandbox policy to the pages which prevents these kinds of attacks.*
>
> ***So… What have we learnt from the honeypot?***
>
> *Our sensor network has contributed to many successes over the years. We’ve presented on 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 0day and nDay 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 “productionising” 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.*
>
> *The team developing this capability also works closely with the MSRC who our 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 situational awareness enables us to give clearer advice to the industry, better protect our customers and integrate new threats we were 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.*

It’s very interesting that the announcement of a new vulnerability immediately triggers a mass frenzy as attackers, who are literally everywhere, scurry to take advantage of it before machines are patched. The author continues...

What happened next?

The "code" subdomain had been known to bug bounty researchers for several years. So whenever they would receive a report from someone who believed 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. And that time 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.



The timeline gives an order of events from our perspective. It's unknown exactly how the full exploit URL for 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 3 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.

In other words, it's very difficult to keep anything 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's bots are able to discover everything. In this case they somehow discovered a secret that Microsoft had been trying to keep 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 follows as to whether the domain is the Microsoft "code" subdomain. Six days later, on April 21st, Google trends show that many people are now searching for the "code" domains. Three days later, on the 24th, they start noticing a significant uptick in traffic to the subdomain. And finally on the 26th, they're hit with **126 thousand times** more requests than average. The write:

> *By 26th April we were handling ~160,000 requests per day, up from the usual 5-100. 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 URL's 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. As this was a simulation, this did not happen. 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 @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 Rick roll 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 has 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 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 wrong is happening. That's generally sufficient for most purposes. But 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.