

Security Now! #1070 - 03-17-26

CISA's Free Internet Scanning

This week on Security Now!

- The Security Now "Caption That Photo" contest.
- A mega social media company says "no" to strong encryption.
- WhatsApp to give parents more control.
- Consumer bandwidth proxying is becoming a big deal.
- Meta buys the Moltbook duo.
- The EU gives up and settles upon the status quo.
- When a ransomware negotiation is not what it seems.
- CISA compels federal agencies to submit their logs.
- Is that a VPN in your pocket or something more malicious.
- Be careful what you download thinking it's AI.
- A super-clever and super-simple A/V scanner bypass.
- Will AI write code for me?
- Another listener discovers the Joy of AI.
- Steve's CISA Internet scanning experience.

Security Now's "Caption That Photo" contest



Security News

No more E2EE for you!

Last week the news was that TikTok had decided and formally announced that it would not be adding end-to-end encryption to its already-controversial short-format video sharing platform. And then, somewhat surprisingly, last Friday The Hacker News reported that Meta had announced their somewhat similar plan to back encryption out of their Instagram product. The Hacker News wrote:

Meta has announced plans to discontinue support for end-to-end encryption (E2EE) for chats on Instagram after May 8, 2026. The social media giant said in a help document: "If you have chats that are impacted by this change, you will see instructions on how you can download any media or messages you may want to keep. If you're on an older version of Instagram, you may also need to update the app before you can download your affected chats."

When reached for comment, this is what Meta had to say: "Very few people were opting in to end-to-end encrypted messaging in DMs, so we're removing this option from Instagram in the coming months. Anyone who wants to keep messaging with end-to-end encryption can easily do that on WhatsApp." The American company first began testing E2EE for Instagram direct messages in 2021 as part of CEO Mark Zuckerberg's "privacy-focused vision for social networking." The feature is currently "only available in some areas" and is not enabled by default. Weeks into the Russian-Ukrainian war in February 2022, the company made encrypted direct messaging available to all adult users in both countries.

The development comes days after TikTok said it does not plan to introduce E2EE to secure direct messages on the platform, telling BBC News that the technology makes users less safe and that it wants to protect users, especially young people, from harm.

Late last month, Reuters also reported that Meta proceeded with plans to adopt encryption to secure messages in Facebook and Instagram despite internal warnings in 2019 that doing so would hinder the company's ability to detect illegal activities, such as child sexual abuse material (CSAM) or terrorist propaganda, and flag them to law enforcement.

E2EE has been hailed as a win for privacy, as it ensures that only communicating users can decrypt and read messages, thereby locking out service providers, bad actors, and other third parties from accessing or intercepting the data. However, law enforcement and child safety advocates have argued that the technology creates a safe space for criminals, as it prevents companies from complying with warrants to turn over message content – a problem referred to as the "Going Dark" phenomenon. This year, the European Commission is expected to present a Technology Roadmap on encryption to identify and evaluate solutions that enable lawful access to encrypted data by law enforcement, while safeguarding cybersecurity and fundamental rights.

I think this is interesting, and I wonder whether this signals the start of a gradual backing away from providing strong encryption to consumers on the mega-popular generic platforms. I doubt whether most lawful users of TikTok, Instagram or even WhatsApp really care all that much about encryption. Sure, if they can have it for free and if it's built-in, they'll take it. But is even a single person going to walk away if it's removed? I doubt it. While there was an initial rush on the part of the publishers to provide it, I don't think it's ever been shown that there was any consumer demand. Anyone who really wanted secure messaging could switch to Signal, which is also free, and where Meridith maintains unflagging vigilance at the gate.

So I suspect that the right solution to all the mess and pushback to the increasing prevalence of fully encrypting everyone's random messages on consumer platforms by default, is simply not to bother with it and no one will much care. I know this will make the privacy-at-all-costs people's heads explode, but, again, Signal is always available and is free for anyone who actually wants it. For those who worry about grooming and CSAM, removing always-on end-to-end encryption from the major platforms will eliminate opportunistic abuse.

WhatsApp adds parent-managed accounts

While we're on the topic, Meta also announced the addition of parent-managed accounts for WhatsApp. The accounts are designed for pre-teen children where access to account settings will be controlled by a PIN set up by the parent. While the message content on the pre-teen accounts will remain private, parents will be able to approve to whom their children may speak, what groups they can join, and review message requests from unknown contacts. This seems like a good thing. Parents are not forced to employ these controls. But they're now available if desired.

Everyone wants to use your IP

Last week we looked at the company "Bright Data" whose unfortunate business model involves arranging to offer end users lower costs for streaming and/or fewer advertisements in return for routing Internet traffic through their ISP's bandwidth and thus using their residential consumer IP address. As we noted last week, there's only one conceivable reason for doing this, which is to allow 3rd parties to hide whatever their purpose might be among the world's consumers.

The issue of consumer proxies was again in the news last week for another reason. Last week's Risky Business news opened by writing:

American and European law enforcement agencies have seized the infrastructure of a residential proxy provider named SocksEscort; the latest such crackdown against proxy providers over the past years. The service had been running since 2021 and rented access to more than 369,000 different IP addresses across its lifetime. According to the FBI, Europol, and Dutch Police, SocksEscort was a front for a malware operation that infected modems and home routers.

In other words, these were malware proxies. Maliciously installed without their hosts' knowledge or permission and forming a proxy botnet. We've talked about proxy botnets through the years and as IP-based blocking has grown the bad guys' need to bounce their traffic off innocent end users has grown. The article continues:

Lumen's Black Lotus Labs linked it to a botnet it discovered in 2023, named AVRecon. The botnet never grew to an extremely large size, but managed to maintain a healthy pool of IP addresses it could rent out to its customers, most of which were other cybercrime operations needing ways to hide their attacks inside the infrastructure of residential internet providers. Europol linked the service to ransomware deployments, DDoS attacks, and the distribution of child sexual abuse material (CSAM). It also estimated that SocksEscort operators made more than €5 million Euro from renting their infected IPs, which is quite the sum for such a service.

On the day of the takedown, the FBI published an advisory with tips on how telcos and consumers can protect their devices and prevent them from ending up as nodes in proxy networks. It also published advice on spotting and removing AVRecon from residential devices.

Over the past few years, the US has mounted a war against residential proxy networks after several reports concluded that foreign adversaries were using infected American routers to hide their tracks. Law enforcement takedowns have targeted both private proxy networks—called ORBs, or Operational Relay Box networks—but also residential proxy providers. The difference between the two is that ORBs are typically built and managed by the threat actors for their sole use, while a residential proxy provider is a service built for an operator's financial gain, typically rented out to whoever has the money.

Past proxy-using botnets that were taken down include 911 S5, Anyproxy, 5socks, RSOCKS, Flax Typhoon's Raptor Train, Volt Typhoon's KV Botnet, APT28's Moobot, VPNFilter, and others.

So our takeaway here is that while bad guys probably have little interest in the contents of any random person's internal network, there is substantial interest in using & abusing any distributed bandwidth they're able to obtain. We've also learned that "substantial interest in" equates to substantial pressure to get in. Keeping the bad guys out means resisting any temptation to rely upon a border router's authentication mechanisms. Any NAT router without any deliberately exposed WAN-side services is inherently bulletproof. In every case, it is only when consumers decide to deliberately expose external management, access, or other router-hosted services that authentication bugs in the router's firmware can be leveraged to install and maintain proxies.

Meta expected to close MoltBook purchase deal

In case anyone is wondering, MoltBook — that weird facility that was affiliated with OpenClaw which was where OpenClaw's autonomous AI agents went to talk amongst themselves where we lowly humans could only look on, gawking in wonder at the inter-AI-agent dialog — was just purchased by Meta. I assume Meta's entire interest is in obtaining the two creators of MoltBook, Matt Schlicht and Ben Parr, since they started working at Meta yesterday, March 16th in Meta's MSL which stands for "Meta Superintelligence Labs".

Matt Schlicht has been working on autonomous AI agents since 2023 and launched Moltbook in late January as an experimental "third space" for AI agents and Moltbook was built largely with the help of Schlicht's personal AI assistant, which he named Clawd Clawderberg. Matt's partner in Moltbook and now also at Meta is Ben Parr, formerly an editor and columnist at Mashable and CNET.

Apparently, Moltbook continues to be available though Meta indicated that they weren't certain what its future might be. The typical corporate-speak statement from Meta, as reported by Axios, was that "The Moltbook team joining MSL opens up new ways for AI agents to work for people and businesses" ... and I doubt that even they know what they mean by that. But that's how these sorts of acquisitions go where it's the people that are actually being acquired.

"Voluntary" Chat Control throughout the EU.

The good news is that the EU was unable to secure the votes needed to pass its most recent attempt to force all communication services to monitor their user's communications. So what we have is an extension of the previous "voluntary" chat control that's already been in place.

Last Wednesday, March 11th, Heise Online covered this news, writing:

The EU Parliament approved a renewed extension of "voluntary chat control" to combat child

sexual abuse in Strasbourg on Wednesday. After the initiative surprisingly failed in the responsible committee a week ago, MEPs are now attaching clear restrictions to the extension.

*The regulation creates a temporary **exception** to European data protection rules, **allowing** messaging services to scan chats for depictions of child sexual abuse. There is currently no agreement on a long-term solution, as desired by the EU Commission and member states.*

Providers of messaging services may automatically scan their platforms for digital traces of child pornography. The search for adults who prey on minors ("grooming") is also under debate. Because this violates the EU directive on the protection of privacy, the EU hastily created an exception regulation in 2021. This exception regulation, which has already been extended once, is valid until the beginning of April and was supposed to be renewed until April 2028 at the request of the EU Commission. Last week, however, the Commission's proposal surprisingly failed in Parliament's Committee on Civil Liberties, Justice and Home Affairs.

In a new compromise, Parliament has now agreed to an extension until August 2027. At the same time, MEPs voted for a clear limitation of powers to search for already known material, and only for users or groups suspected of concrete wrongdoing. Furthermore, encrypted chats should not be affected.

A spokesperson for the Committee on Civil Liberties, Justice and Home Affairs said: "This exception is a temporary, strictly limited instrument that allows providers to continue their voluntary detection measures under certain conditions. The extension must also maintain end-to-end encryption."

These restrictions correspond to Parliament's draft for a long-term solution. These will be the subject of upcoming negotiations with the Commission and member states. Only when an agreement is reached here can the renewed extension come into force. There is currently no majority in Parliament for far-reaching surveillance powers such as arbitrary chat control. The Council of Member States has also moved away from this after a long struggle. However, this does not make a permanent "voluntary" solution any easier, especially since it also affects the fundamental rights of EU citizens.

While the Commission and member states want to make the controversial exception regulation permanent, the EU Parliament insists on significant restrictions. For example, error-prone technologies such as AI should not be used in the search for child pornographic depictions. Scanning text messages for grooming attempts should also remain prohibited.

So, saner heads prevailed and things largely remain as they have been. Companies who wish to examine the content of EU citizens for the limited and express purpose of combating CSAM for known content may do so without running afoul of the EU's existing privacy laws. It is not at all clear how any of this will change in the future. And that's fine, too.

When ransomware negotiations are not what they seem

This next bit of news made me shake my head. I won't spend much time on it, but I didn't want to let it pass without comment. CyberScoop informs us that ransomware negotiators working for the ransomware negotiation firm DigitalMint were also the ransomware attackers they were negotiating with. CyberScoop wrote:

A 41-year-old South Florida man is accused of conducting at least 10 ransomware attacks and

*helping accomplices extort a combined \$75.25 million in ransom payments **while** he was working as a ransomware negotiator for DigitalMint. According to federal court records unsealed Wednesday, five of Angelo John Martino III's alleged victims hired DigitalMint, which assigned Martino to conduct ransomware negotiations on their clients' behalf — putting him in a position to play both sides, as the criminal responsible for the attack and the lead negotiator for his alleged victims.*

Martino allegedly obtained an affiliate account on ALPHV, also known as BlackCat – a criminal ransomware as a service group – and conspired with other former cybersecurity professionals to break into victims' networks, steal and encrypt data, and extort companies for ransoms over a six-month period. Prosecutors accuse Martino of providing confidential information regarding ransomware negotiations to ALPHV co-conspirators to maximize the ransom payment.

The five U.S.-based victims that hired DigitalMint and unwittingly tapped Martino to allegedly conduct ransomware negotiations with himself and his co-conspirators include a nonprofit and companies in the hospitality, financial services, retail and medical industries. All five of those victims paid ransoms.

CyberScoop's coverage of that continues at some length, but everyone gets the idea. On the one hand, this obviously puts the guy who's negotiating both sides of the deal in the position to know exactly how much ransom his victim will actually pay. The upside is that the negotiator is also in the unique position to know for sure whether the attackers – since that's also him – will actually honor their promise to restore the victim's data and delete any copies they might have.

CISA requires federal agencies to share their Cisco SD-WAN logs

Three weeks ago, during episode 1067, we covered the news of yet another horrific CVSS 10.0 in Cisco's SD-WAN product. This is the bug behind CVE-2026-20127, another critical authentication bypass in Cisco's Catalyst SD-WAN (it had another back in 2020) which allows unauthenticated remote attackers to gain admin-level access to SD-WAN controllers to compromise entire WAN infrastructures.

Last Wednesday, CISA revised their previous orders which we covered three weeks ago. CISA has now ordered federal agencies to upload logs from Cisco SD-WAN devices to its own cloud platform by next Monday, March 23. These Cisco Catalyst SD-WAN devices have been under attack using a 0-day since 2023 and a great many of Cisco's customers have done nothing about it. While CISA has no jurisdiction over private enterprises, it does over federal agencies. This uploading and aggregating of the logs on CISA's platform will allow CISA to investigate which agencies have been compromised. Agencies will also have to configure their Cisco SD-WAN to send future logs to the same Cloud Logging Aggregation Warehouse, also known as CISA CLAW.

When you download a VPN client, what else might you be downloading?

The past year has seen a huge upward trend in the use of VPN services for geo-relocation. This increase in VPN use has been driven by new regional legislation which forces providers of age-restricted content to block access based upon the geo-location of their would-be visitors.

Unfortunately, a new demand and a rush to something creates new opportunities for bad guys to take advantage of the inexperience of newbies who are entering a market that's new to them. We've previously noted that this has been happening with VPN add-ons for Chrome. Microsoft security has been tracking a group they identify as Storm-2561 which has been using SEO poisoning to provide malicious links to unwitting Windows users looking for VPN client software.

Microsoft writes:

In mid-January 2026, Microsoft Defender Experts identified a credential theft campaign that uses fake virtual private network clients distributed through search engine optimization poisoning. The campaign redirects users searching for legitimate software to malicious ZIP files on attacker-controlled websites to deploy digitally signed trojans that masquerade as trusted VPN clients while harvesting VPN credentials. Microsoft Threat Intelligence attributes this activity to the cybercriminal threat actor Storm-2561.

Active since May 2025, Storm-2561 is known for distributing malware through SEO poisoning and impersonating popular software vendors. The techniques they used in this campaign highlight how threat actors continue to exploit trusted platforms and software branding to avoid user suspicion and steal sensitive information. By targeting users who are actively searching for VPN software, attackers take advantage of both user urgency and implicit trust in search engine rankings. The malicious ZIP files that contain fake installer files are hosted on GitHub repositories, which have since been taken down. Additionally, the trojans are digitally signed by a legitimate certificate that has since been revoked.

This blog shares our in-depth analysis of the tactics, techniques, and procedures (TTPs) and indicators of compromise in this Storm-2561 campaign, highlighting the social engineering techniques that the threat actor used to improve perceived legitimacy, avoid suspicion, and evade detection. We also share protection and mitigation recommendations, as well as Microsoft Defender detection and hunting guidance.

In this campaign, users searching for legitimate VPN software are redirected from search results to spoofed websites that closely mimic trusted VPN products but instead deploy malware designed to harvest credentials and VPN data. When users click to download the software, they are redirected to a malicious GitHub repository (no longer available) that hosts the fake VPN client for direct download.

I'll note that while Microsoft keeps reinforcing that the malware has been taken down, they know as well as we do that no sooner will one set of malware be taken down than its replacement will appear. In fact, it's more often the case that multiple sets of redundant malware have already been staged in place on GitHub and are just waiting to be linked to. This allows that malware to age a bit on the platform to increase its appearance of authenticity. So, a takedown of one set, while certainly useful and necessary, should by no means suggest that the threat has been in any way diminished. This is a classic case of whack-a-mole and while it's true that the game must be played, it can never be won by playing catch up – another mole will always be ready to pop up somewhere else.

Microsoft continues:

The GitHub repo hosts a ZIP file containing a Microsoft Windows Installer installer file that mimics a legitimate VPN software and side-loads malicious dynamic link library (DLL) files during installation. The fake VPN software enables credential collection and exfiltration while appearing like a benign VPN client application.

This campaign exhibits characteristics consistent with the financially motivated cybercrime operations employed by Storm-2561. The malicious components are digitally signed by "Taiyuan Lihua Near Information Technology Co., Ltd."

The initial access vector relies on abusing SEO to push malicious websites to the top of search results for queries such as "Pulse VPN download" or "Pulse Secure client," but Microsoft has observed spoofing of various VPN software brands and has observed the GitHub link at the following two domains: [vpn-fortinet\[.\]com](http://vpn-fortinet[.]com) and [ivanti-vpn\[.\]org](http://ivanti-vpn[.]org).

Once the user lands on the malicious website and clicks to download the software, the malware is delivered through a ZIP download hosted at <https://github.com/latestver/vpn/releases/download/vpn-client2/VPN-CLIENT.zip>. At the time of this report, this repository is no longer active.

*When the user launches the malicious MSI masquerading as a legitimate Pulse Secure VPN installer embedded within the downloaded ZIP file, the MSI file installs **Pulse.exe** along with malicious DLL files to a directory structure that closely resembles a real Pulse Secure installation path: %CommonFiles%\Pulse Secure. This installation path blends in with legitimate VPN software to appear trustworthy and avoid raising user suspicion.*

*Alongside the primary application, the installer drops malicious DLLs, **dwmapi.dll** and **inspector.dll**, into the Pulse Secure directory. The **dwmapi.dll** file is an in-memory loader that drops and launches an embedded shellcode payload that loads and launches the **inspector.dll** file, a variant of the infostealer Hyrax.*

The Hyrax infostealer extracts URI and VPN sign-in credentials before exfiltrating them to attacker-controlled command-and-control infrastructure.

In other words, no one wants **any** of this **anywhere** near **any** of their computers. Microsoft noted that the files were all signed. As I've been saying, no code can get off the ground in today's world without being signed by someone. In this case, Microsoft explains:

The MSI file and the malicious DLLs are signed with a valid digital certificate, which is now revoked, from Taiyuan Lihua Near Information Technology Co., Ltd. This abuse of code signing serves multiple purposes:

- *Bypasses default Windows security warnings for unsigned code*
- *Might bypass application whitelisting policies that trust signed binaries*
- *Reduces security tool alerts focused on unsigned malware*
- *Provides false legitimacy to the installation process*

Microsoft identified several other files signed with the same certificates. These files also masqueraded as VPN software.

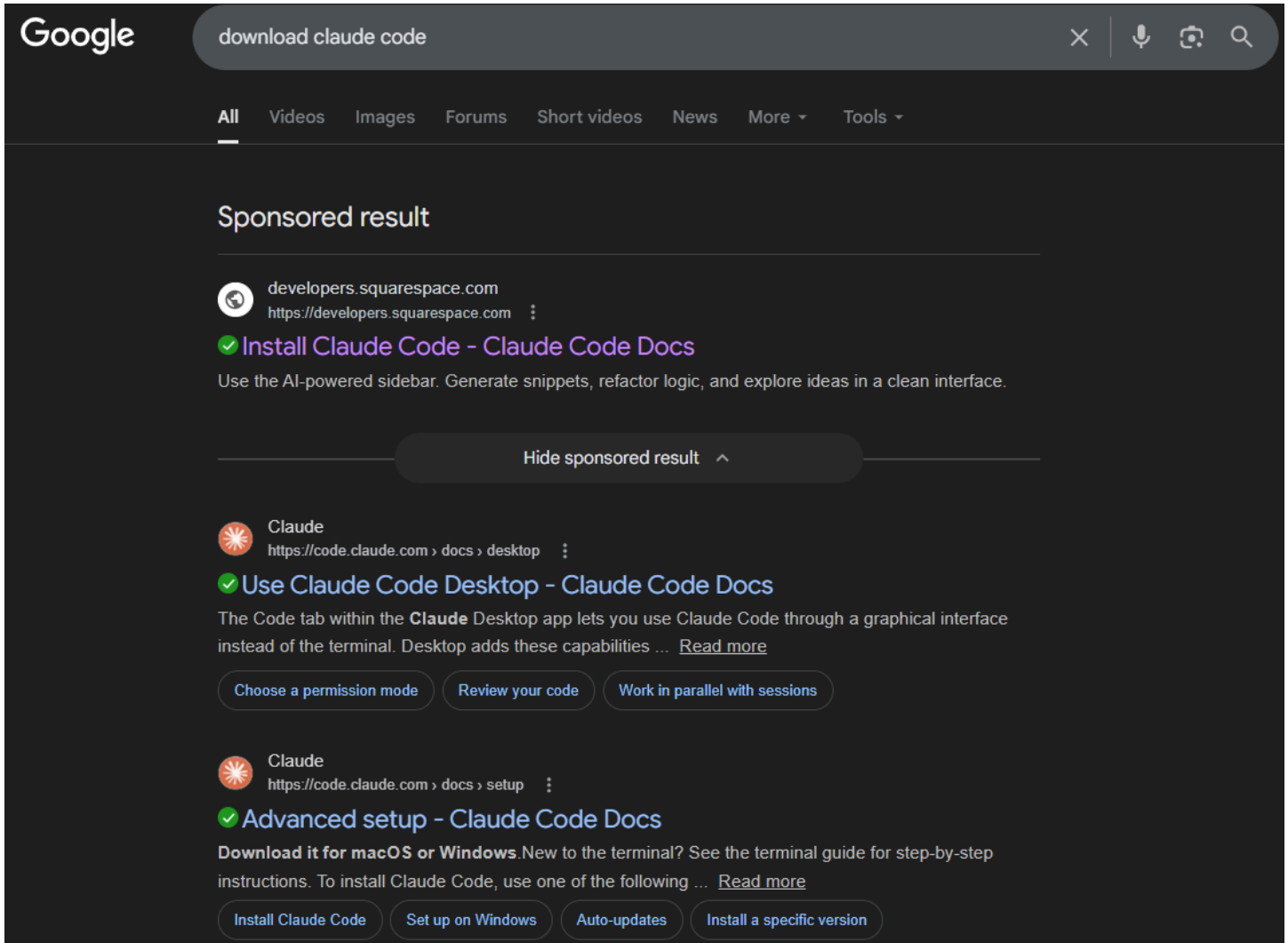
Microsoft described this as an "abuse of code signing". I suppose it's an abuse of the intent of code signing, but I'd be inclined to call it a failure of the code signing requirement to prevent the use of malicious software. The bad guys didn't "abuse" code signing, they "used" code signing to abuse the process code signing was designed to prevent. Perhaps I'm splitting hairs.

What we don't know and what Microsoft chose not to reveal is whether this "Taiyuan Lihua Near Information Technology Co., Ltd." is an authentic firm whose valid signing certificate somehow got loose, or whether the company was always a façade which bad guys used to obtain a valid code signing certificate. Microsoft also chose not to reveal who signed their certificate. It would be interesting to know which Certificate Authority allowed themselves to be spoofed, and how and where, exactly, the required chain of enterprise existence proof failed.

The only actionable takeaway we can have from this is the annoyingly diffuse imperative to remain ever vigilant. There are bad guys scattered all around the world focused upon taking advantage of our trust or any momentary lapse in our attention. All we really can be is as well informed and as careful as possible.

Everyone wants AI

While we're on the subject of bad guys taking advantage of the passion of the day, I wanted to note that Bitdefender, Kaspersky, and ThreatBook all recently posted independent examinations of the dramatic rise they all noted in malicious web pages offering instructions for installing AI agents like Claude and OpenClaw:



Because Google labels it as a sponsored result and the branding looks authentic, users tend to trust it. WE KNOW that Claude Code should not be sourced and downloaded from the website "developers.squarespace.com" but not everyone looking for it, OpenClaw, or whatever will be aware of that or will take the time. And those web pages actually deploy malware. No more needs to be said other than to be careful and to always go to the original source of anything you obtain from the Internet.

The Zombie ZIP vulnerability

A security researcher by the name of Christopher Aziz of Bombadil Systems has discovered a very very clever new technique that allows for the creation of malware-containing ZIP files that slide right past endpoint device security tools. In his testing, Christopher found that his ZIP format hack would evade 98% of antivirus engines.

When Chris packaged some known malware in a regular ZIP file it was almost universally detected by the A/V engines at VirusTotal. But when he then simply tweaked the ZIP file's header to simply claim that its file contents had been directly stored rather than compressed, nearly all existing A/V tools were fooled into believing that the contents was just gibberish. In other words, they didn't attempt to decompress the contents because the header said it wasn't compressed. It's almost too easy.

Christopher put up a page on his GitHub account to draw attention to this obvious-in-retrospect vulnerability: <https://github.com/bombadil-systems/zombie-zip?tab=readme-ov-file> He wrote:

How It Works

AV engines trust the ZIP Method field. When Method=0 (STORED), they scan the data as raw uncompressed bytes. But the data is actually DEFLATE compressed — so the scanner sees compressed noise and finds no [viral] signatures.

The CRC is set to the uncompressed payload's checksum, creating an additional mismatch that causes standard extraction tools (7-Zip, unzip, WinRAR) to report errors or extract corrupted output.

However, a purpose-built loader that ignores the declared method and decompresses as DEFLATE recovers the payload perfectly.

*The vulnerability is **scanner evasion**: security controls assert "no malware present" while malware is present and trivially recoverable by attacker tooling.*

As for the attack vector, this is not an end-user extraction vulnerability. It is a staged delivery / smuggling technique:

- *A malicious payload packaged in a Zombie ZIP with a modified header.*
- *The ZIP transits security boundaries (email gateways, network scanners, endpoint AV)*
- *Scanners read Method=0, scan compressed noise and report "all clean"*
- *A purpose-built loader or dropper decompresses the payload programmatically*
- *The payload materializes and executes*

This is consistent with established malware delivery patterns (ISO smuggling, HTML smuggling, CAB abuse) where attackers use custom loaders rather than consumer extraction tools.

What is affected?

- *50/51 AV engines on VirusTotal*
- *Microsoft Defender, Avast, Bitdefender, ESET, Kaspersky, McAfee, Sophos, TrendMicro, etc.*
- *Only Kingsoft detected it*


This just goes to show how some of the simplest hacks, even after all this time, can still be among the most effective. Sometimes there's no need to get overly fancy.




Will AI write code for Steve?

Our listeners continue to express their understandable curiosity over my own plans for AI coding. I'm sure this is partially due to my having made T-shirts that say "Born to Code" and also due to my having been completely open minded about a topic that has perhaps been more near and dear to me than anything else in my life. I have many times celebrated Leo's successes and experiences embracing Claude Code, and I've shared many of our listeners' similar, stunned, mouth-left-hanging-open experiences when AI-produced code for them that made their computers do things they never imagined they'd be able to obtain for themselves. Obviously, something huge has happened. The question remains, what that is, exactly.

As I settled down last Saturday morning to begin assembling today's podcast I decided to log into 'X' to see whether any of our listeners might have posted a candidate picture of the week there. Everyone has largely switched over to using email, as I have, but you never know. So it was serendipitous that when I happened to check, my feed contained several posts that were completely on-topic for the question of AI and coding. Presumably, Elon's 'X' system knows of my interest in the topic and therefore dropped those posts into my feed.

The first post I want to share was written by a guy named "Aakash Gupta" who posts frequently on Medium <https://aakashgupta.medium.com/> and whose short bio says he helps product managers, product leaders, and product aspirants succeed:



Aakash Gupta  @aakashgupta · 13h  

41% of all code shipped in 2025 was AI-generated or AI-assisted. The defect rate on that code is 1.7x higher than human-written code. And a randomized controlled trial found that experienced developers using AI tools were actually 19% slower than developers working without them.


Devs have always written slop. The entire software industry is built on infrastructure designed to catch slop before it ships. Code review, linting, type checking, CI/CD pipelines, staging environments. All of it assumes one thing: the person who wrote the code can walk you through what it does when the reviewer asks.


That assumption held for 50 years. It broke in about 18 months.

When 41% of your codebase was generated by a machine and approved by a human who skimmed it because the tests passed, the review process becomes theater. The reviewer is checking code neither of them wrote. The linter catches syntax, not intent. The tests verify behavior, not understanding.







The old slop had an owner. Someone could explain why temp_fix_v3_FINAL existed, what edge case it handled, and what would break if you removed it. The new slop has an approver. Different relationship entirely.

Aavid's right that devs wrote bad code before AI. The part he's missing: the entire quality infrastructure of software engineering was designed around a world where the author and the debugger were the same person. That world ended last year and nothing has replaced it yet.



Arvid Kahl  @arvidkahl · Mar 13

Devs are acting like they didn't write slop code before AI.

 109  159  1K  92K  

Aakash's post captures aspects of my discomfort with using AI to create code. So the answer to the question of whether AI will write code for me would be "not the AI we have today." Even before this AI coding revolution arose I should objectively have at least been using 'C'. But I'm so comfortable with assembly language, and I now have so much solid boilerplate written by me in assembly language that moment to moment the path of least resistance is to just keep using assembly.

When I face the possibility of using something to write code for me, I'm immediately brought up short, wondering how I can possibly know the code it creates is correct? The code I am writing and need is never for a lark. I'm always writing production code that I and others will depend upon. Either it's server-side code for GRC, or code that will form a product that bears my name. In either case, the code **needs** to be **as** correct as I am able to make it. It's true that I have strong perfectionist tendencies. I know that's one of the reasons people listen. I don't ever judge my work by whether it's "good enough". I judge it by whether it's as good as I am capable of making it. So, if I don't actually write the code I'm using, offering and selling, how can I ever definitively make that judgement? If no one or nothing "sentient" and personally responsible creates it, if the code just magically appears, and if there are large swaths of code that is never carefully inspected, how can I ever have confidence in what the code does?

Sure... I know, testing. Test, test, test. I get that. That is, as all of our many development testers know quite well, the development model that has evolved with the code that I currently author by hand. But is the appearance of the code working, or the code no longer being seen to fail, an adequate replacement for someone **one** actually writing the code for a purpose? I don't know.

But I do know that the entire world is objectively going nuts over AI-written code. Perhaps the reason for this is that there's tremendous pressure within the larger code-creating universe to create more code with fewer human coders.


So perhaps it's the fact that I truly love writing code myself, and that I feel very little pressure to produce more code faster? I've talked about days past when my little company employed many more people, many of whom I was jealous of, since they were getting to do the work I wanted to be doing instead of just managing them. If that's the case, then why would I want to have an AI producing code that I would not then have the joy of writing for myself? All of the foregoing suggests that the answer to the question "When will Steve be using AI to author his code?" is at least not yet.




Before we leave this topic for now, I want to share another 'X' post that appeared in my feed directly underneath that previous. It was written by someone who's well enough known to have a Wikipedia page created to capture and describe his life's work. His given name is Robert Martin though he goes by "Uncle Bob Martin." Wikipedia informs us that he's an American software engineer, instructor, and author who is most recognized for promoting many software design principles and for being an author and signatory of the influential Agile Manifesto. He's authored many books and magazine articles and was the editor-in-chief of the C++ Report magazine and served as the first chairman of the Agile Alliance. He joined the software industry at age 17. He's credited with introducing the collection of object-oriented programming design principles that came to be known as SOLID.

Wikipedia mentions that he has authored many books. Yeah. 13 books. Since I'm going to share his important opinion about the current state of AI-generated code, I want to first clearly establish why he might be someone worth listening to. Here are the titles of the 13 books he's authored across the past 30 years. These are real books published by Prentice Hall, Cambridge University Press, Addison-Wesley Professional and Pearson:

- Designing Object-Oriented C++ Applications Using the Booch Method.
- More C++ Gems.
- Extreme Programming in Practice.
- Agile Software Development, Principles, Patterns, and Practices.
- UML for Java Programmers.
- Agile Principles, Patterns, And Practices in C#.
- Clean Code: A Handbook of Agile Software Craftsmanship.
- The Clean Coder: A Code Of Conduct For Professional Programmers.
- Clean Architecture: A Craftsman's Guide to Software Structure and Design.
- Clean Agile: Back to Basics.
- Clean Craftsmanship: Disciplines, Standards, and Ethics.
- Functional Design: Principles, Patterns, and Practices.
- We, Programmers: A Chronicle of Coders from Ada to AI.

Here's what Uncle Bob Martin posted last Saturday morning:



Uncle Bob Martin  @unclebobmartin · 6h  

Two months ago, while working on my Empire game with AI, I had that quicksilver experience. When you push on a blob of mercury it slips out in some random direction.

Every time I added a new feature some older feature would shift behavior. This was true even after I added unit tests and acceptance tests. The AI always took the path of least resistance on the current feature, and was willing to sacrifice older features. It would change tests, including acceptance tests in order to get the latest feature done.

Telling the AI not to do that was ineffective. AIs are stochastic, and so are any rules you feed them. Rules `_bias_` their behavior, but do not absolutely constrain it. When I call them out on breaking rules, they apologize and swear they won't do it again; but they can't really make that promise. They are, in the end, liars and cheats.







The solution is to massively overconstrain them. Force them to write so many tests that changing a tested feature breaks many tests. They feel that force and retract the change. It's like peer pressure with a lot of peers.

At the same time I reduce the chances for collateral damage by continuously forcing the AI to partition everything into small decoupled units. That way it is not easy to break one feature while implementing another. It also keeps the AI from getting confused by it's own messes.

The final goal is semantic stability in the face of continuous development. The things that worked before keep working as they were; while newer things get added.

This is a continuous effort. Acceptance tests, unit tests, TDD. Crap analysis, and mutation tests are run after a reasonable batch of changes and are tasked with reducing crap below 8, covering any untested behavior, and killing all surviving mutants. The size of the batch of changes is a judgement call. Too big and the analysis and repairs take a long time. Too small and the verification effort overwhelms the development effort.

Side note. The mutation tests consume massive amounts of computer power. My cores are running full bore all the time; and that's even with differential mutation. There is something poetically just about this. The AIs require a massive amount of computer power to create. What they create for us takes a massive amount of computer power to keep stable.

 14
 12
 167
 9.1K
 

From the start of our discussion of AI I've been saying that I firmly believe AI will have a very bright future in coding. I still believe that's true. But **not** today's AI. Today's AI is still general purpose AI. It's like asking AI for that list of high quality random numbers. Doing that perfectly, which we now know how to do, requires specialization, not generalization. This is every bit as true when it comes to writing code correctly. The laughable catastrophic mess Bob describes in his posting – commonly referred to as attempting to herd cats – is no way to write code. These four sentences from Bob's posting say it all. He wrote:

- *AIs are stochastic, and so are any rules you feed them.*
- *Rules _bias_ their behavior, but do not absolutely constrain it.*
- *When I call them out on breaking rules, they apologize and swear they won't do it again; but they can't really make that promise.*
- *They are, in the end, liars and cheats.*

I believe that in those four statements Uncle Bob exactly and perfectly captures the state of play today. But that's only today. I'm always very careful to state that nothing we have or believe we have today regarding AI will hold tomorrow. The way I think this will shake out is that someday we will have many differing forms of application-specific AI. I suspect that's where the answer lies. As I understand today's AI operation, having a single super-genius AI that contains all knowledge and does everything perfectly is incredibly wasteful, as in way too expensive to contain and operate, if all you want is high quality code. Instead, employ the far more cost-effective services of a specialist code-generating AI whose model can be far smaller while also containing far more concentrated knowledge about code and only about code.

Now, having just said all that, here's the first piece of feedback from a listener that happened to arrive as I was assembling today's show:

Listener Feedback

Craig

The Subject of Craig's email was "Hard to describe." He wrote:

First I'd like to say, thanks for mentoring me throughout nearly my entire career. Now retired, I ran the IT department for a 50 employee DoD/DoE subcontractor. What I learned from you and implemented over the years made NIST 800-171 compliance easy. And I can proudly say that my company was NEVER hacked .. oh wait, aside from that warez kiddie who created a hidden ftp site on my public FTP server. Remember those days? lol. But aside from that, never once was my network taken down. I had weekly security awareness training for my users almost always from your show. I was tight a decade before anyone was even thinking of security. Thank you!

My entire career was hobbled by my poor coding skills. I never attended college for computers, just drinking and failing out. I learned everything building PCs in those box shops in the 90s. Netware Lite FTW, lol. Computer Shopper FTW! I used to tell people I can code, but I can't develop. I could write a simple script ... after hours of scouring Stack Exchange or Spiceworks to figure it out. The places I could have gone if I had properly trained as a developer.

Now all those tools I wish I had over my 30 years of career are at my fingertips. The best analogy I can give is that I spent my career in 2D black and white, and all of a sudden I can see 3D in color and Infrared and ultraviolet and x-ray.....

I now have an entire agent infrastructure team; CISO, Architect, Audit, Monitoring, Hardening, infrastructure, etc... managing my entire homelab. My kitchen module has an AI Chef running from local Ollama to help with the current recipe. I just got done having CISO build a 3D desktop for my platform inside of my Quest 2. It made downloading 20 years of google account and then organizing it into my own system easy... It's working on building out a complete voice system around my house. It can talk to my 3D printers. All of this is possible and I just have to ask for it in natural language. My jaw is still on the ground... I hate to say it, Steve, but commercial software is dead. I don't need to buy what I can have my agents write. All I need are the GPUs.

CMDottie Platform

Machines
10 machines

All Active Maintenance Inactive

Machine	Status	Hours/Day
Band Saw	Active	8h
Chop Saw	Active	8h
Creality Laser Cutter Engraver	Active	8h
Drill Press	Active	8h
FolgerTech Cloner BTT Manta M5P board upgrade 10.200.1.38	Active	24h
Jointer	Active	8h
Planer	Active	8h
Table Saw	Active	8h
Voron 2.4 CoreXY 3D printer, BTT Manta M8P board 10.200.1.24	Active	24h
Wood Lathe	Active	8h

Platform Admin

CISA's Free Internet Scanning

Last week I shared feedback from a listener who shared with us that his organization uses CISA's free Internet network scanner to keep an eye on his organization's network security exposure. He explained that when he first had CISA scan their network what they found was quite bracing and brought their other IT people up short. And as I also noted, his sharing that with me raised my curiosity about just who might qualify for CISA's periodic scanning.

It's formally known as CISA's Cyber Hygiene Service and its page says:

Reduce the Risk of a Successful Cyber Attack

Cyber threats are not just possibilities but harsh realities, making proactive and comprehensive cybersecurity imperative for all critical infrastructure. Adversaries use known vulnerabilities and weaknesses to compromise the security of critical infrastructure and other organizations. CISA offers no-cost cybersecurity services to help organizations reduce their exposure to threats by taking a proactive approach to monitoring and mitigating attack vectors.

By taking advantage of CISA's Cyber Hygiene services you can:

- *Significantly Reduce Risk — Organizations typically reduce their risk and exposure by 40% within the first 12 months. Most see improvements in the first 90 days.*
- *Avoid Surprises — Because the services look for assets exposed to the internet, they identify vulnerabilities that could otherwise go unmanaged.*
- *Sharpen Your Response — By combining the vulnerability insights gained with existing threat detection and risk management efforts, enrolled organizations can increase the accuracy and effectiveness of response activities. This means fewer false alarms and less chance of real dangers slipping through the net.*
- *Broaden Your Security Horizon — CISA's scanning is about more than pinpointing vulnerabilities; it's about expanding your organization's security boundaries. From basic asset awareness to daily alerts on urgent findings, you'll be in a better place to make risk-informed decisions.*

CISA's Cyber Hygiene services include:

- *Vulnerability Scanning: This service continuously monitors and assesses internet-accessible network assets (public, static IPv4 addresses) to evaluate their host and vulnerability status. In addition to weekly reports of all findings, you'll receive ad-hoc alerts about urgent findings, like potentially risky services and known exploited vulnerabilities.*
- *Web Application Scanning: This service deep-dives into publicly accessible web applications to uncover vulnerabilities and misconfigurations that attackers could exploit. This comprehensive evaluation includes, but is not limited to, the vulnerabilities listed in the OWASP Top Ten, which represent the most critical web application security risks. This service provides detailed reports monthly, as well as on-demand reports to help keep your web applications secure.*

I've brought all this up again because my experiment to see whether GRC's little, decidedly non-governmental, non-tribal, 16-IP network block might qualify to receive CISA's automatic, periodic background security scans and reporting was a resounding and surprising success. And based upon my experience, I would hazard to imagine that a great many of our U.S.-based listeners who are in charge of their own small, medium and large enterprise networks would be able to similarly qualify to receive this free service, as I have. And if so, why wouldn't everyone wish to avail themselves of this entirely sane, zero-cost service offered by an agency of our federal government?

I suppose I can imagine that it might make some listeners a bit queasy to invite Uncle Sam to scan and report on the state of their networks. But stop to consider that anything that might be discovered and reported is already public information. It's not as if we're making an exception for CISA, allowing them through our firewalls to rummage around inside our networks. They're on the outside attempting to look in, just like would-be attacking hackers in Russia, North Korea and China. The difference is that **CISA is on our side** with the goal of strengthening North American networks against attackers in Russia, North Korea, China and elsewhere. They email password-protected PDF reports that only its intended recipient is able to open and view. I don't see any possible down side, whereas I see a potentially huge upside.

So what happened with GRC?

That CISA <https://www.cisa.gov/cyber-hygiene-services> page invites candidates to indicate their interest and open a dialog by sending an email to "vulnerability@cisa.dhs.gov" with the subject "Requesting Cyber Hygiene Services" So I addressed an email and wrote simply:

To whom it may concern, I own a small commercial network which I would like to have scanned. Thank you! /Steve.

That was on the morning of Saturday March 7th, so no one was working at CISA. I received a reply to that email first thing Monday, after the weekend, at 5:32am Pacific, so 8:32am in the East. That email response said:

Steve,

Thank you for your interest in our Cyber Hygiene (CyHy) Vulnerability Scanning (VS) service. Enrollment in our CyHy VS service must be done by a person in your organization who has ownership or authority over the IP addresses to be enrolled. This individual should hold a position such as Chief Information Officer, Chief Information Security Officer, or a similar official capacity.

If you are in this role, please proceed to navigate CISA's Cyber Services: Cyber Hygiene Services, the beta version of our web-based enrollment system, to complete the following steps:

- 1. Create a Login.gov Account :Login.gov is our trusted partner for secure and private access to CISA's online services, including Cyber Hygiene. The Login.gov account must use the same organization (business) email that will be used to complete the remaining enrollment steps.*
- 2. Return to CISA's Cyber Services: Cyber Hygiene Services Page: After logging in, you will*

now be redirected to the CISA Services Portal for ReadySetCyber. Use the navigation ribbon to go to Cyber Services > Enroll in Cyber Hygiene to return to the enrollment process.

- 3. Complete Account Registration & Organization's Profile: Complete your organization's profile, enabling your organization to receive Cyber Hygiene and access other CISA services.*
- 4. Complete Enrollment Form: Once you have completed the Organization information page, you will be redirected to a Thank You page. Select the "Enroll Now" option to continue the CyHy VS enrollment process. This step includes collection of the necessary information to enroll in the CyHy VS service and serves as the authorizing document allowing CISA to perform the CyHy VS service for your organization.*

NOTE:For the IP Address validation process,

- a. You will need to input and successfully verify the formatting of your IP Address(s) before continuing to the next page*
- b. Multiple IP addresses must be separated by comma or line break*
- c. If there are errors with the formatting, the system will display a modal noting how many errors*
- d. You will have the option to either go back and correct the errors or download a CSV file for editing if you have input numerous entries.*

If you have questions during your enrollment, please reach out to us at this email address.

Best regards, Matt Leon

CISA Vulnerability Management Intake Team

Cybersecurity and Infrastructure Security Agency (CISA) | Cybersecurity Division (CSD)

Vulnerability Management (VM) | Ops and Plans

Email: vulnerability@mail.cisa.dhs.gov

So I went back to CISA and logged in at [Login.gov](https://login.gov), where I already had an account since I've used it for managing social security and renewing both my global entry certification and driver's license. I was then bounced back over to CISA where I filled out a modest and not very intrusive questionnaire.

Around ten minutes after completing that process I received another email with the subject "CISA Organizational Account Confirmation" and an invitation button to complete the sign-up process. I may have done something there, I don't recall. But either way, the email trail shows that 13 minutes later I received a final email with the subject "Cyber Hygiene Vulnerability Scanning Acceptance Letter". I thought "Huh! That was easy." The letter said:

Welcome To CISA's Cyber Hygiene (CyHy) Vulnerability Scanning (VS)

These people really do love their abbreviations. The letter continues:

Your CyHy VS Acceptance Letter has been processed and a copy of the letter has been attached for your convenience. Your organization has been placed in queue for inclusion into the CISA CyHy VS service. Scanning will begin as soon as your request file is processed in alignment with your requested scan start date.

If not otherwise requested the scanning begins immediately. The letter continues:

Please keep an eye out for traffic from the CyHy VS Scanning IPs which will signal to you that scanning has begun. You will receive your first CyHy VS report via email on the Tuesday following the initial scan which is based on your requested scan start date. The CyHy VS report will come from reports@cyber.dhs.gov.

Overview of CISA CyHy VS methodology:

Cyber Hygiene defines a host as having at least one open port and service. Scanning of hosts occurs continuously between each weekly report. Cyber Hygiene's scan prioritization is as follows:

- *Addresses with no running services detected (dark space) are rescanned after at least 90 days.*
- *Hosts with no vulnerabilities detected are rescanned every 7 days.*
- *Hosts with low-severity vulnerabilities are rescanned every 6 days.*
- *Hosts with medium-severity vulnerabilities are rescanned every 4 days.*
- *Hosts with high-severity vulnerabilities are rescanned every 24 hours.*
- *Hosts with critical-severity vulnerabilities are rescanned every 12 hours.*
- *A single host may have multiple vulnerabilities of varying severity, which informs the frequency that a given host is scanned.*

Need Assistance: If you need to make changes to the information submitted in the Acceptance Letter to include updating IPs to be scanned, or if you have any other questions pertaining to your CyHy VS service please email us at vulnerability@cisa.dhs.gov

Then, last Wednesday, the day after last week's podcast, I received my first "CyHy VS" report.

I'll admit I was actually somewhat surprised to see that CISA had not found anything critical to complain about. But that's not to say that CISA did not find anything. They did complain that GRC's web servers would still negotiate and accept SSL/TLS connections using old and deprecated 64-bit block ciphers – things like Triple-DES and Blowfish.

What caused my heart to initially skip a beat – or two – was that their report's headline was **"Urgent Vulnerabilities Detected"**. That certainly commanded my attention. Their report enumerates their findings by Vulnerability description, whether it's known to be exploited, whether ransomware is known to exploit it, its severity, the host IP address and port where they vulnerability was found and the data and time of its initial discovery.

In this case, all of GRC's web server IPs at the HTTPS port 443 share the vulnerability that CISA identifies as: *"SSL Medium Strength Cipher Suites Supported (SWEET32)"*.

It is not, however, known to have ever been exploited. The reason is that the SWEET32 vulnerability and attack is theoretical. It's called SWEET32 because the theoretical attack has a complexity of 2^{32} . The "Sweet" part comes from the pun "Sweet 16" because it's a Birthday attack.

The vulnerability has its own website at <https://sweet32.info/> which explains the nature of the attack, writing:

An important requirement for the attack is to send a large number of requests in the same TLS connection. Therefore, we need to find clients and servers that not only negotiate the use of Triple-DES, but also exchange a large number of HTTP requests during a single TLS connection (without rekeying). This is possible using a persistent HTTP connection, as defined in HTTP/1.1 (Keep-Alive). On the client side, all browsers that we tested (Firefox, Chrome, Opera) will reuse a TLS connection as long as the server keeps it open.

Okay. It says "*a large number of requests during a single TLS connection*". But exactly how large? In their own testing to recover a 16-byte authentication token – which might be an HTTP cookie – so two 64-bit encrypted blocks, they needed to keep a single TLS connection established for 18.6 hours during which their client pounded on the server with a storm of continuous small HTTP requests, transferring 705 gigabytes of data in the process.

In short, at least for GRC, this is not a real problem. But that does not mean there's any way for me to defend GRC's totally unnecessary support for this old and weak Triple-DES cipher today. So I very much appreciate the reminder nudge from CISA, and I've already tweaked the cipher suite configuration of GRC's various web servers so that the next time they're rebooted their support for that long-ago deprecated Triple-DES cipher suite will disappear. It hasn't been useful for a long time.

So that's the story of GRC's establishment of an ongoing free vulnerability scanning service courtesy of CISA. I cannot imagine why anyone listening to this podcast, who is responsible for anything more than a single IP home network and any sort of truly fixed pre-assigned IPs pointed to by DNS, would not wish to immediately avail themselves of CISA's free scanning service.

You won't know what might surprise you until you do. And even if you find nothing, that would be super useful. If you do find something it might be very important. And the more that's going on within a complex networking environment involving multiple departments and overlapping responsibilities and configurations, the more chance there is for unsuspected trouble to arise.

