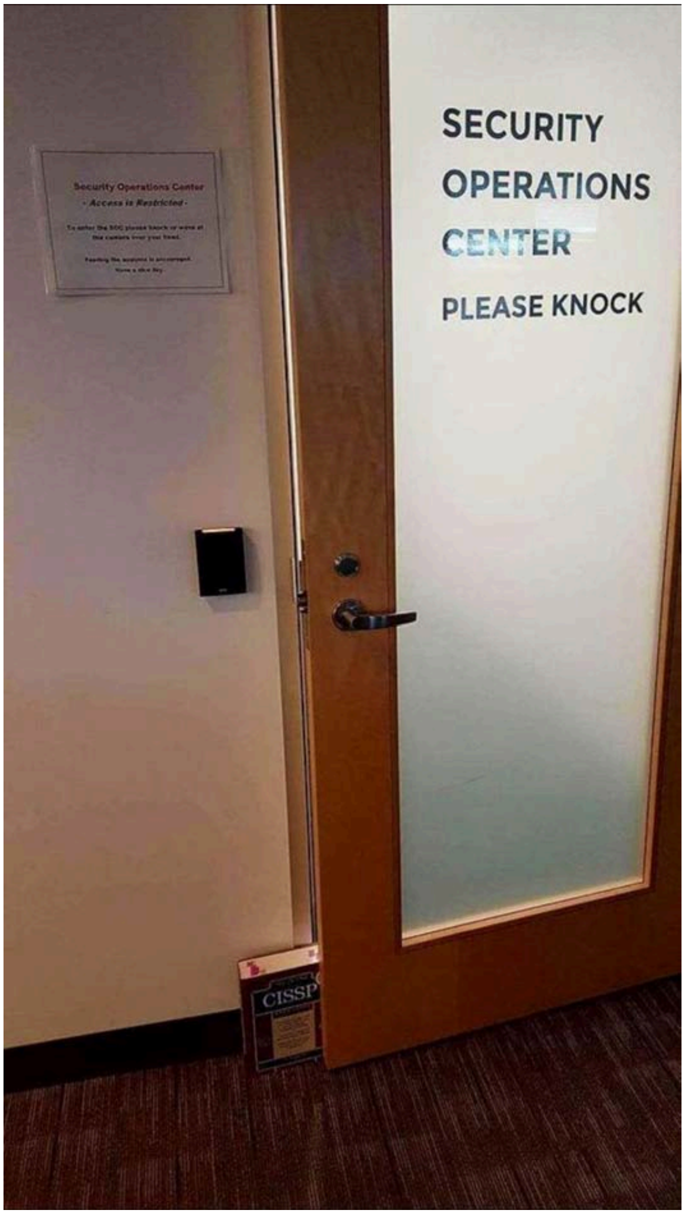
Security Now! #1022 - 04-22-25

Windows Sandbox

This week on Security Now!

• Enabling Firefox's Tab Grouping. • Recalled Recall Re-Rolls out. • The crucial CVE program nearly died. It's been given new life. • China confesses to hacking the US (blames our stance on Taiwan). • CISA says what Oracle still refuses to. • Brute force attacks on the (rapid) rise. • An AI/ML Python package rates a 9.8 (again!) • The CA/Browser forum passed short-life certs. :(• A wonderful crosswalk hack hits Silicon Valley. • Android to add force restarting ahead of schedule. Maybe. • The EFF is never happy. But especially now, about Florida. • Interesting research into ransomware payouts. • Windows Sandbox: The amazing gem hidden inside all Windows 10 & 11!

Why we will never have perfect security.

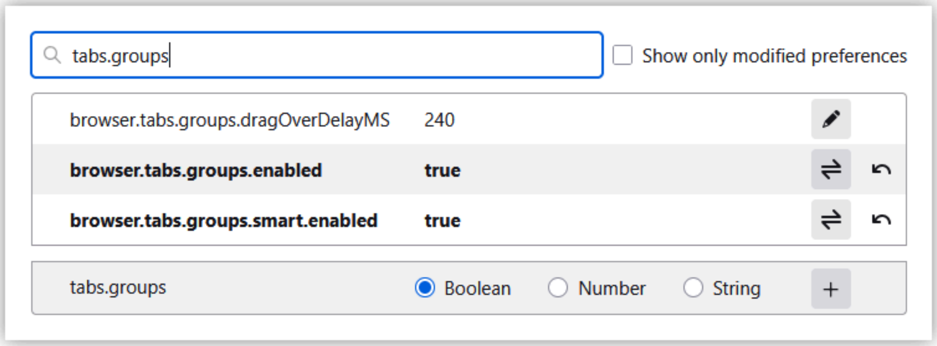

Security Operations Center
- Access is Restricted -
To enter the SOC please knock or wave at the camera over your head.
Feeding the analysts is encouraged.
Have a nice day.
SECURITY
OPERATIONS
CENTER
PLEASE KNOCK
CISSP

Security News

Enabling Firefox Tab Grouping

After hearing last week's note about Firefox Tab grouping and how I had been unable to get a pair of tabs to merge, a number of our listeners said "Uhhhh... Steve... it's probably there but just disabled." And sure enough, I not have the ability to merge tabs to create groups.

To enable this feature, place "about:config" in the Firefox address bar then search for "tabs.groups" which should return three entries:



tabs.groups

□ Show only modified preferences

browser.tabs.groups.dragOverDelayMS	240
browser.tabs.groups.enabled	**true**
browser.tabs.groups.smart.enabled	**true**

tabs.groups ● Boolean ○ Number ○ String +

The "dragOverDelayMS" specifies the number of milliseconds Firefox should wait while you're dragging one tab over another before it decides that your intention is to merge the dragged tab into the dragged-over tab or group. The other two settings are true/false Boolean settings which should both be set to "true" to enable tab grouping. Once that's done you'll be able to experiment with tab groups.

If we assume Mozilla was deliberate and correct in their "incremental rollout" announcement, it must be that everyone who already has Firefox 137 or later already has the code for this, and that what's being incrementally rolled-out is the autonomous enabling of this feature, which any user is free to do for themselves immediately if they wish.

And a big "thank you" to our listeners who immediately sent that helpful "Uhhhh... Steve..." feedback.

The re-Rollout of Recall

This news would have made it into last week's podcast, except that last week already broke the record for the longest Security Now! podcast ever. So there was no room available. And just so that everyone knows, it is not my intention for Security Now! to become a three hour podcast. I recognize that three hours takes up a lot of everyone's life and I heard the pushback I received after last week's marathon. It may happen from time to time when the time is required, but it's not some new goal or intention of mine.

The original announcement of a new release of Windows 11 to the Release Preview Channel was made on April 10th and that was for Build 26100.3902. But that release apparently had some issues which caused Microsoft to update to .3909, last Friday the 18th. And, after all, that's the inherent nature of Preview Releases; things are going to be discovered due to wider deployment and then fixed.

Because Microsoft now clearly recognizes that their CoPilot+ "Recall" technology is a big deal and a huge change in the operation of Windows, it was the first new feature noted for this Preview

Channel release. Once Recall makes its way into production I'm sure we'll give it another close look — as will the entire Windows 11-using world. But as Microsoft promised last year, when Recall is initially released it will be disabled by default and provide users with settings to manage its behavior. I'm glad to see that Microsoft has tempered its excitement and enthusiasm for this technology. It's a big deal and there's no hurry. It should be eased into the market gradually and voluntarily. Then, over time, as people have adjusted to its presence, they can decide how much push is appropriate to give it. For now, I doubt that it should not be pushed at all. And, again, what's the hurry?

The CVE program to switch from DHS-funded to a non-profit

For a few days last week it appeared that the incredibly important and useful Common Vulnerabilities and Exposures (CVE) program that's operated by the MITRE Corporation and has been funded by DHS, the US Department of Homeland Security, might become unfunded and shut down. The entire security industry breathed a collective sigh of relief with the news that CISA found some loose change available to keep it going. Last Wednesday, under their headline "CISA extends funding to ensure 'no lapse in critical CVE services'" BleepingComputer wrote the following:

> *CISA says the U.S. government has extended MITRE's funding to ensure no continuity issues with the critical Common Vulnerabilities and Exposures (CVE) program. The U.S. cybersecurity agency told BleepingComputer: "The CVE Program is invaluable to the cyber community and a priority of CISA. Last night, CISA executed the option period on the contract to ensure there will be no lapse in critical CVE services. We appreciate our partners' and stakeholders' patience."*
>
> *BleepingComputer has learned that the extension of the contract is for 11 months. The announcement follows a warning from MITRE Vice President Yosry Barsoum that government funding for the CVE and CWE programs was set to expire today, April 16, potentially leading to widespread disruption across the cybersecurity industry. Barsoum said: "If a break in service were to occur, we anticipate multiple impacts to CVE, including deterioration of national vulnerability databases and advisories, tool vendors, incident response operations, and all manner of critical infrastructure."*
>
> *MITRE maintains CVE, a widely adopted program that provides accuracy, clarity, and shared standards when discussing security vulnerabilities, with funding from the U.S. National Cyber Security Division of the U.S. Department of Homeland Security (DHS).*
>
> *After publishing our story, MITRE shared the following statement with BleepingComputer.*
>
> > *"Thanks to actions taken by the government, a break in service for the Common Vulnerabilities and Exposures (CVE®) Program and the Common Weakness Enumeration (CWE™) Program has been avoided. As of Wednesday morning, April 16, 2025, CISA identified incremental funding to keep the Programs operational. We appreciate the overwhelming support for these programs that have been expressed by the global cyber community, industry, and government over the last 24 hours. The government continues to make considerable efforts to support MITRE's role in the program and MITRE remains committed to CVE and CWE as global resources."*
> >
> > ❖ *Yosry Barsoum, Vice President, Director, Center for Securing the Homeland, MITRE*

> *Before CISA's announcement, a group of CVE Board members announced the launch of the CVE Foundation, a non-profit organization established to secure the CVE program's independence in light of MITRE's warning that the U.S. government might not renew its contract for managing the program.*
>
> *MITRE said in a Wednesday press release: "Since its inception, the CVE Program has operated as a U.S. government-funded initiative, with oversight and management provided under contract. While this structure has supported the program's growth, it has also raised longstanding concerns among members of the CVE Board about the sustainability and neutrality of a globally relied-upon resource being tied to a single government sponsor."*
>
> *Over the last year, the individuals involved in the launch have been developing a strategy to transition the program to this dedicated foundation, eliminating "a single point of failure in the vulnerability management ecosystem" and ensuring "the CVE Program remains a globally trusted, community-driven initiative."*
>
> *While the CVE Foundation plans to release further information about its transition planning in the coming days, the next steps remain unclear, especially considering CISA has confirmed that funding for MITRE's contract has been extended. The European Union Agency for Cybersecurity (ENISA) has also launched a European vulnerability database (EUVD), which "embraces a multi-stakeholder approach by collecting publicly available vulnerability information from multiple sources."*

So, whew! It's difficult to imagine a world without some common, uniform, system for ranking the dangers and threats of vulnerabilities. Lord knows, the US government probably obtains at least as much value and benefit from having this program as any other entity. CISA will provide an additional 11 months of federal funding to MITRE, making this a very valuable wake-up call for the rest of the industry and giving it time to arrive at a non-government funded alternative.

Announcing the CVE Foundation

And speaking of a non-government funded alternative. Also last Wednesday the industry was treated to a press release from the newly formed "CVE Foundation". The press release read:

> *FOR IMMEDIATE RELEASE / CVE Foundation Launched to Secure the Future of the CVE Program*
>
> *[Bremerton, Washington] – The CVE Foundation has been formally established to ensure the long-term viability, stability, and independence of the Common Vulnerabilities and Exposures (CVE) Program, a critical pillar of the global cybersecurity infrastructure for 25 years.*
>
> *Since its inception, the CVE Program has operated as a U.S. government-funded initiative, with oversight and management provided under contract. While this structure has supported the program's growth, it has also raised longstanding concerns among members of the CVE Board about the sustainability and neutrality of a globally relied-upon resource being tied to a single government sponsor.*
>
> *This concern has become urgent following an April 15, 2025 letter from MITRE notifying the CVE Board that the U.S. government does not intend to renew its contract for managing the program. While we had hoped this day would not come, we have been preparing for this possibility.*

> *In response, a coalition of longtime, active CVE Board members have spent the past year developing a strategy to transition CVE to a dedicated, non-profit foundation. The new CVE Foundation will focus solely on continuing the mission of delivering high-quality vulnerability identification and maintaining the integrity and availability of CVE data for defenders worldwide.*
>
> *Kent Landfield, an officer of the Foundation said: "CVE, as a cornerstone of the global cybersecurity ecosystem, is too important to be vulnerable itself. Cybersecurity professionals around the globe rely on CVE identifiers and data as part of their daily work—from security tools and advisories to threat intelligence and response. Without CVE, defenders are at a massive disadvantage against global cyber threats."*
>
> *The formation of the CVE Foundation marks a major step toward eliminating a single point of failure in the vulnerability management ecosystem and ensuring the CVE Program remains a globally trusted, community-driven initiative. For the international cybersecurity community, this move represents an opportunity to establish governance that reflects the global nature of today's threat landscape.*
>
> *Over the coming days, the Foundation will release more information about its structure, transition planning, and opportunities for involvement from the broader community.*
> *For updates or inquiries, contact: info@thecvefoundation.org.*

So https://www.thecvefoundation.org/home exists and depending upon how things look 11 months from now, it appears that one way or another, the valuable CVE reference system will survive and thrive.

China admits to hacking the U.S.
The Wall Street Journal carried the news under their headline: *"In Secret Meeting, China Acknowledged Role in U.S. Infrastructure Hacks"* with the sub-heading: *"A senior Chinese official linked intrusions to escalating U.S. support for Taiwan."* The Journal's story said:

> *Chinese officials acknowledged in a secret December meeting that Beijing was behind a widespread series of alarming cyberattacks on U.S. infrastructure, according to people familiar with the matter, underscoring how hostilities between the two superpowers are continuing to escalate. The Chinese delegation linked years of intrusions into computer networks at U.S. ports, water utilities, airports and other targets, to increasing U.S. policy support for Taiwan, the people, who declined to be named, said.*

So the attribution of these attacks to state-sponsored groups, specifically "Voly Typhoon", has been officially substantiated and we have further evidence of what seems to me like a bizarrely intertwined and complex relationship between our two countries.

CISA says what Oracle won't
As one security news reporter wrote: *"CISA has published an alert on the Oracle Cloud data breach before Oracle did—mainly because the company is still busy wordsmithing its way around the issue."* CISA's "Alert" published last Wednesday was titled: *"CISA Releases Guidance on Credential Risks Associated with Potential Legacy Oracle Cloud Compromise"* CISA wrote:

> *CISA is aware of public reporting regarding potential unauthorized access to a legacy Oracle cloud environment. While the scope and impact remains unconfirmed, the nature of the reported activity presents potential risk to organizations and individuals, particularly where credential material may be exposed, reused across separate, unaffiliated systems, or embedded (i.e., hardcoded into scripts, applications, infrastructure templates, or automation tools). When credential material is embedded, it is difficult to discover and can enable long-term unauthorized access if exposed.*
>
> *The compromise of credential material, including usernames, emails, passwords, authentication tokens, and encryption keys, can pose significant risk to enterprise environments. Threat actors routinely harvest and weaponize such credentials to:*
>
> - *Escalate privileges and move laterally within networks.*
> - *Access cloud and identity management systems.*
> - *Conduct phishing, credential-based, or business email compromise (BEC) campaigns.*
> - *Resell or exchange access to stolen credentials on criminal marketplaces.*
> - *Enrich stolen data with prior breach information for resale and/or targeted intrusion.*
>
> *CISA recommends the following actions to reduce the risks associated with potential credential compromise: For Organizations:*
>
> - *Reset passwords for any known affected users across enterprise services, particularly where local credentials may not be federated through enterprise identity solutions.*
>
> - *Review source code, infrastructure-as-code templates, automation scripts, and configuration files for hardcoded or embedded credentials and replace them with secure authentication methods supported by centralized secret management.*
>
> - *Monitor authentication logs for anomalous activity, especially involving privileged, service, or federated identity accounts, and assess whether additional credentials (such as API keys and shared accounts) may be associated with any known impacted identities.*
>
> - *Enforce phishing-resistant multi-factor authentication (MFA) for all user and administrator accounts wherever technically feasible.*
>
> - *For additional information for or on Cloud security best practices please review the following Cybersecurity Information Sheets: CISA and NSA Release Cybersecurity Information Sheets on Cloud Security Best Practices.*
>
> *For Users:*
>
> - *Immediately update any potentially affected passwords that may have been reused across other platforms or services.*
> - *Use strong, unique passwords for each account and enable phishing-resistant multifactor authentication (MFA) on services and applications that support it. For more information on using strong passwords, see CISA's Use Strong Passwords web page. For more information on phishing-resistant MFA see CISA's Implementing Phishing-Resistant MFA Fact Sheet.*
> - *Remain alert against phishing attempts (e.g., referencing login issues, password resets, or suspicious activity notifications) and reference Phishing Guidance: Stopping the Attack Cycle at Phase One.*

So, as we see, that advice could hardly have been more generic. That doesn't mean it's not obviously useful advice. But it does mean that in the absence of any confession from Oracle that's about as definitive as anyone can be.

While I doubt Oracle's irresponsible behavior will hurt them in the very short term, no one who's involved in the security industry is likely to forget this. It really should cause everyone to wonder *"if they will act this way, what else is their internal corporate and security culture likely to do?"*

Brute force attacks on the rise thanks to "FastHTTP"

I wanted to share a recent useful, important and thought-provoking piece from the security firm Rapid7. Their piece was titled *"Password Spray Attacks Taking Advantage of Lax MFA"* MFA, of course, abbreviating Multi-Factor Authentication. I've recently been encountering reports of significantly increased brute-force guessing attacks and so-called "credential stuffing". I recall us taking a close look at some problems McAfee had several years ago where bad guys were just pounding and pounding away at their login pages while McAfee was apparently blissfully unaware. And, of course, just offering MFA is not a guarantee of safety. We recently looked at Microsoft's mis-designed MFA system which was allowing massive MFA brute-forcing – enough to bypass the "million guesses required" barrier presented by any random 6-digit passcode. But the more factors that can be added without unduly inconveniencing the user, the better.

And as we've also seen, being smart about the deployment of MFA or even backup email loop confirmation, where connecting from a previously seen IP or carrying a known browser cookie can be used to shift the security of a login in the direction of increased trust. This can make those layers of additional authentication far less intrusive and annoying, while still being available when needed.

Here's part of what Rapid7 wrote:

> *In the first quarter of 2025, Rapid7's Managed Threat Hunting team observed a significant volume of brute-force password attempts leveraging FastHTTP, a high-performance HTTP server and client library for Go, to automate unauthorized logins via HTTP requests.*
>
> *This rapid volume of credential spraying was primarily designed to discover and compromise accounts not properly secured by multi-factor authentication (MFA). Out of just over a million unauthorized login attempts we observed, the distribution of* ***originating*** *traffic sources is similar to that previously seen just in January 2025. Some of the most prominent nations serving as points of origin for these attempts are as follows:*
>
> - *Brazil: 70%*
> - *Venezuela: 3%*
> - *Turkey: 3%*
> - *Russia: 2%*
> - *Argentina: 2%*
> - *Mexico: 2%*
>
> *Rapid7 has consistently highlighted MFA as a primary concern across several threat research reports. By the midpoint of 2023, data for the first half of the year showed that 39% of incidents our managed services teams responded to had arisen from lax or lacking MFA. Our 2024 Threat Landscape blog highlighted that remote access to systems without MFA was responsible for 56% of incidents as an initial access vector, the largest driver of incidents overall.*

> *The third quarter of 2024 saw 67% of incident responses involving abuse of valid accounts and missing or lax enforcement of MFA. This total sits at 57% for Q4 2024, in part because of a 22% increase in social engineering. Even without pausing to consider user agent-centric password spraying, this is a potentially dangerous combination for organizations not making the most of MFA-centric protection. If the brute forcing doesn't get you, a social engineering campaign might just do the trick.*
>
> *Why MFA Matters and the consequences of "We'll Set It up Later"*
>
> *MFA is a key component of an overall Identity Access Management (IAM) strategy. If you're not making use of it, then your overall defense is weakened against many of the most common threats out there, including:*
>
> - *Phishing: The very best password you can muster is made entirely redundant if your employee hands it over to a phisher, whether via a forged website or a social engineering attack. One way to mitigate against this is to use a password manager, which will only automatically enter your details on a valid website. But what happens if your password manager's master password is compromised, and all the logins contained within are exposed? One of the best ways to address this additional headache is MFA for all your accounts, including your password manager.*
>
> - *Malware: Do you know what malware, password stealers, and keyloggers, love more than anything else? Grabbing all of those passwords stored in web browsers, or (in more serious cases) plain text files on the desktop and email drafts. Do you know what they don't like? Having all of those perilous passwords protected with an additional layer of security. MFA could make the difference between compromise and data exfiltration versus, a last-minute save and a security training refresher.*
>
> - *Credential stuffing: An unfortunate by-product of years of data breaches (often with phishing as the launchpad), roll-ups of new and ancient login details published online are a constant threat. It's worth noting that it isn't just your current employees who could be on these lists—ex-employees with valid credentials are a cause for concern too.*
>
> *Here are some steps you can take now to improve your security posture and mitigate risk from attacks like these, courtesy of Rapid7's experts:*
>
> - *Implement multi-factor authentication (MFA) across all account types, including default, local, domain, and cloud accounts, to prevent unauthorized access, even if credentials are compromised.*
> - *Use conditional access policies to block logins from non-compliant devices or from outside defined organization IP ranges.*
> - *Ensure that applications do not store sensitive data or credentials insecurely. (e.g. plaintext credentials in code, published credentials in repositories, or credentials in public cloud storage).*
> - *Audit domain and local accounts as well as their permission levels routinely to look for situations that could allow an adversary to gain wide access by obtaining credentials of a privileged account. These audits should also include if default accounts have been enabled, or if new local accounts are created that have not been authorized. Follow best practices for design and administration of an enterprise network to limit privileged account use across administrative tiers.*

> - *Regularly audit user accounts for activity and deactivate or remove any that are no longer needed.*
> - *Whenever possible and aligned with business requirements, disable legacy authentication for non-service accounts and users relying on it. Legacy authentication, which does not support MFA, should be replaced with modern authentication protocols.*
> - *Applications may send push notifications to verify a login as a form of multi-factor authentication (MFA). Train users to only accept valid push notifications and to report suspicious push notifications.*
>
> *You can't go wrong with MFA*
>
> *Imagine a scenario where your network is under fire from a worryingly high number of brute force attempts from across the globe, targeting your insecure accounts until just one is compromised. Now imagine that same scenario where everything is blocked by default, regional restrictions are applied, logins from user agents aren't allowed, and all of your VPNs, your RDP, VDIs, and SaaS tools are secured with MFA.*
>
> *This may feel like an overreaction to what you may view as an attack that looks like an edge case; however, consider that ransomware groups, alongside more commonly found malware authors and phishers, will also find you a significantly harder target to break as a result of these countermeasures being put in place. Please don't end up in the inevitable percentage of organizations compromised due to missing MFA in our next threat research report; there's no better time than now to think about building out a stronger security posture.*

What all this amounts to is adopting a multi-layered security approach. Never assume that any single protection will be sufficient. If it's possible to practically do more, do more.

And some of the strongest security protections can be somewhat brittle and troublesome. Leo and I cannot login remotely to our SSH servers without a client having the proper private key to verify its identity. Could that cause some inconvenience? Sure, it could be. But no way am I willing to expose an unmonitored SSH server that's only protected by a username and password, no matter how secure they might be.

And, as another example, filtering some classes of remote connections by IP will mean that those filters will break when IP addresses change. I had that happen to me last week when a cable modem died and I needed to switch to another. My cable provider, COX, was wonderful throughout the process, but I wound up with a never-before-seen residential IP address and a great deal of my network infrastructure fell apart. But I was prepared for that. I had previously made notes of all the many places I had an IP-based blocking or permission filter that would need updating, and I had previously arranged to be able to do that remotely in this event. Of course, IP-based permissions is only one layer of my security, but it's an awesomely powerful layer and it is well worth the hassle when, every three to five years, my cable modem's IP may need to change.

So I think that the ultimate takeaway from Rapid7's posting is to appreciate that there really are extremely determined, anonymous and numerous attackers who are more or less continually pounding away, largely unmonitored, outside our gates. They couldn't care less who you are. It's no longer reasonable to say "Well, I'm nobody that anyone would want to hack." They won't know that until they're in. And then, once they're in, the least they'll do is arrange to establish persistence so that they can mine crypto or use your bandwidth to increase their next DDoS attack.

BentoML for Python carries a CVE of 9.8
There's a Python library known as BentoML. And as with pretty much anything "ML", the "ML" stands for Machine Learning. BentoML is a project (https://pypi.org/project/bentoml/) over at PyPi which bills itself as <quote> *"The easiest way to serve AI apps and models"*. Unfortunately, since it also carries a CVSSv3 vulnerability and exploitability score of the difficult-to-attain 9.8, if you're unfortunate enough to be using v1.3.8 through 1.4.2, it may also be the easiest way to have your AI-related service taken over by bad guys thanks to the presence of a critical remote code execution vulnerability.

BentoML's documentation page explains that it's: *"A Unified Inference Platform for deploying and scaling AI models with production-grade reliability, all without the complexity of managing infrastructure. It enables your developers to build AI systems 10x faster with custom models, scale efficiently in your cloud, and maintain complete control over security and compliance."*

Except that apparently it's the bad guys who get to have the complete control over security, or lack of any. Since it seems pretty clear that we're on the brink of a new renaissance in AI-based security threats and vulnerabilities, I figured it would be worth taking a brief closer look at this one. Here's what the security research group "CheckMarx" wrote:

> *A critical Remote Code Execution (RCE) vulnerability, CVE (Thank God for CVE's!) 2025-27520 with a CVSSv3 base score of 9.8, has been recently discovered in BentoML, an AI service helper Python library found on PyPI. This flaw allows unauthenticated attackers to execute arbitrary code by sending malicious data payloads as requests and potentially take control of the server. While the advisory specifies versions from 1.3.4 through 1.4.2 as being affected, Checkmarx Zero's analysis indicates that this issue affects versions 1.3.8 through 1.4.2. It is recommended that affected adopters upgrade to version 1.4.3 or later to repair the issue.*
>
> *You are potentially affected by this issue if you use BentoML (either directly or indirectly) to receive and process ML "payloads" (which are serialized data structures) from untrusted sources. Since this is a primary purpose of BentoML, the presence of a vulnerable version of this library should be considered a significant indicator of risk.*

In other words, arranging to provide BentoML with a malicious serialized payload will not be difficult since that's what BentoML is designed to take in. Checkmarx wrote:

> *CVE-2025-27520 is a Remote Code Execution (RCE) vulnerability found in BentoML, a Python library designed for creating online serving systems that are optimized for AI applications and model inference. The full GHSA advisory describes the vulnerability and exploitation, which we summarize here. The flaw, that originates from an Insecure Deserialization, enables adversaries to execute arbitrary code on the server by sending a specially crafted HTTP request. This issue exists because the deserialize_value() function in the "serde.py" file deserializes input data without proper validation, meaning attackers can inject malicious payloads that trigger execution of arbitrary code when they are deserialized.*

By now, any of this podcast's long-term listeners will perk right up when they encounter the term *"deserialization"* since we have previously encountered so many instances of deserialization gone bad. As we know, *"serialization"* is the process of taking a complex data structure and converting into a stream by bytes – thus *"serializing"* it. So *"deserialization"* is the reverse process that takes as its input a previously "serialized" byte stream and hopefully returns the original complex data structure. The reason we keep encountering security-related problems with deserialization is that the act of deserializing requires the *"interpretation"* of the meaning of the that the serialized byte stream, and “interpreters” are notoriously problematic to get perfect; and any imperfection can too often by leveraged to create an exploitable vulnerability. What’s even more unfortunate is that this is not the first time the BentoML has had this trouble. The NIST already had a listing last year in 2024 for for CVE-2024-2912 to which it assigned the rarest of rare CVSS base scores of 10.0.

And that’s not surprising when a vulnerability disclosure describes the problem by writing: *“The BentoML framework is vulnerable to an insecure deserialization issue that can be exploited by sending a single POST request to any valid endpoint. The impact of this is remote code execution.”* Then Checkmarx writes of the newly discovered flaw:

> *This flaw is essentially a reintroduction of CVE-2024-2912 , which had been previously fixed in version 1.2.5. Both CVEs deal with the same exact issue: an Insecure Deserialization vulnerability that can be exploited by sending an HTTP request to any valid endpoint to trigger remote code execution.*

At this point anyone using BentoML might reasonably question the wisdom of continuing to rely upon the developers of this package to keep them safe. The Checkmarx guys wrote:

> *To exploit this vulnerability, the first step is to craft a malicious “pickle”, a binary data serialization system commonly used with Python. This “pickled” payload contains Python objects that can contain executable code that gets run when the payload is deserialized for use by the application. Vulnerable versions of BentoML do not deserialize such payloads in a safe manner, meaning an adversary can send Python code which performs malicious actions — including executing system commands — under the authority of the Python application running on the server.*
>
> *In this case, an attacker can create a custom Python object (e.g. the Evil class) and override Python’s magic method __reduce__ with a tuple that tells Python to run the os.system function. The __reduce__ method is used to specify how the object should be deserialized or serialized and allows users to override default behavior with other meaningful actions. By calling os.system, an attacker can trigger system commands during the deserialized operation, such as initiating a reverse shell connection to this machine, as shown in the provided Proof of Concept.*

Hoping to understand the sequence of events that caused a previously resolved and quite serious problem to return, the researchers reconstructed the timeline of events. They wrote:

> *The vulnerability exists in BentoML versions 1.3.8 through 1.4.2. If you are running a version within this range, you are affected. The advisory reports versions as early as 1.3.4 are vulnerable, but Checkmarx Zero analysts determined that the vulnerability actually re-emerged in version 1.3.8. Looking at commit 045001C3, we found that a previous security fix — originally introduced to address CVE-2024-2912 — had been removed. This missing code was specifically implemented to prevent this exact deserialization vulnerability now tracked as CVE-2025-27520. So...*
>
> - *The original vulnerability finding was reported as CVE-2024-2912.*
> - *It was patched in version 1.2.5*
> - *The fix was later removed in version 1.3.8*
> - *The same issue resurfaced and was reported again as CVE-2025-27520*
> - *It has now been re-patched in version 1.4.3*

As I noted before, without some very clear accounting – and accountability – for these events, given the potential consequences of this library's direct exposure to the Internet so that a single HTTP POST query can be used to completely takeover a system, anyone using or considering the use of this library would be well advised to proceed with extreme caution.

On the off-chance that any of our listeners might be directly affected by this, I've included the link to Checkmarx's posting and analysis. There isn't anything really tied to machine learning or AI about this. It just appears to be a problematic Python library that appears to need better development management. We all know that mistakes can happen. That's the nature of the game. But if they are to be forgiven they should be followed by some learned lessons. Let's hope that has happened here.
https://checkmarx.com/zero-post/bentoml-rce-fewer-affected-versions-cve-2025-27520/

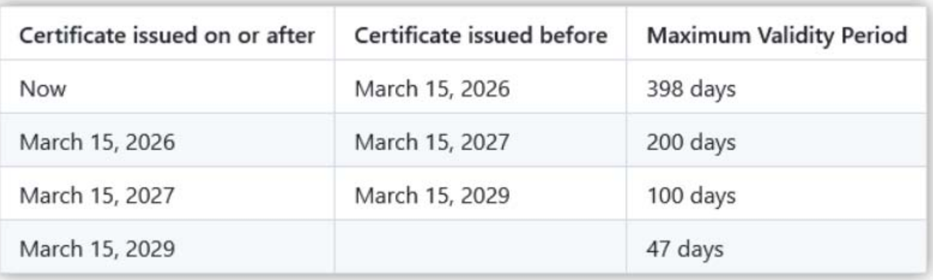
The CA/Browser forum just passed the planned reduction in certificate lifetime.
On April 4th at 12:30, Ballot SC-081v3 was posted and the voting began. 16 minutes later, at 12:46, Chris Clements posted: Google votes Yes on Ballot SC-081v3. The next day, Nick France posted: Sectigo votes Yes on Ballot SC-081v3. That was followed two hours later by Apple's Clint Wilson posting: Apple votes Yes on Ballot SC-081v3. The next day, Corey Bonnell posted: DigiCert votes YES on ballot SC-81v3. And the day after that, Ben Wilson chimed in with: Mozilla votes "Yes" on Ballot SC-081v3.

When the voting ended, of the 30 member Certificate Issuers, 25 had voted YES and no one voted NO, though there were five abstentions. Of the 4 member Certificate Consumers (Apple, Google, Microsoft and Mozilla) **all four voted YES.**

So what was it that just happened? This ballot was the formal adoption of a slightly toned down version of the quite aggressive certificate lifetime shortening proposal first made by Apple's Clint Wilson in October last year.

We talked about it at the time as I shook my head in bemusement. I don't understand it and I probably never will, because the proposal appears to ignore all of the trouble that this will cause, while also conveniently ignoring the fact that 100% privacy-enforcing browser-side certificate revocation has finally been made to work. Yet Clint's proposal just passed, and it did so handily. Clint's position is that **nothing** can be as certain as **never** issuing any certificates having long lives. And it's impossible to disagree with him on that because, factually, he's right. But I've seen no evidence to suggest that such an absolute level of certainty is warranted enough to offset the world of problems that it will also cause.

So what has just passed?



Certificate issued on or after	Certificate issued before	Maximum Validity Period
Now	March 15, 2026	398 days
March 15, 2026	March 15, 2027	200 days
March 15, 2027	March 15, 2029	100 days
March 15, 2029		47 days

Current certificate lifetime is a tolerable 398 days. So that's effectively annual renewal and replacement with some slack. This 398-day maximum lifetime will be operable for any certificates issued before next March 15th of 2026. I'll be reissuing all of mine the day before that because on March 15th of next year maximum lifetime will be summarily cut in half to 200 days – for no apparent reason that I'm able to devine. The year after that, on March 15th of 2027, lifetimes will again be cut in half just to 100 days. So this is essentially quarterly – requiring reissuance and renewal four times per year. At that point we either automate or we spend our lives fussing with certificates. And finally, in an apparent concession to reality, the annual march to certificate lifetime extinction receives a two-year break since the final drop to just 47 days is deferred until March 15th of 2029. But from then on, no certificate will be issued having a lifetime longer than 47 days. Why? I have no friggin' idea. But that's the way it's going to be. And everyone has just signed onto that.

What's clear is that anyone who is building any sort of device that needs to use public-facing certificates trusted by Chrome, Chromium, Firefox and Safari is going to need to add ACME automation to their appliance and they should start thinking about it sooner rather than later.

For those running web servers this shouldn't be any huge problem. There's a 'win-acme' client for Windows servers that I'll be able to use. Since I'll need to automatically issue a wildcard certificate I'll need to have it editing GRC's DNS, but that's easily enough done. I'm mostly annoyed because no one has made the case for why everyone needs to be so inconvenienced by this.

Of course, the flip side is that ACME is already being very widely used to dynamically generate the TLS certificates for 70% of all web servers worldwide. So it's really only the laggard 30%, of which I'm a part, that needs to get with the program.

Silicon Valley crosswalk buttons hacked to imitate Musk, Zuckerberg's voices

Last week, TechCrunch carried the news of a pretty wonderful hack that hit the crosswalks across the Northern California peninsula, commonly referred to a Silicon Valley. TechCrunch's headline was: *"Silicon Valley crosswalk buttons hacked to imitate Musk, Zuckerberg's voices"*. They wrote:

> Audio-enabled traffic control crosswalk buttons across Silicon Valley were hacked over the weekend to include audio snippets imitating the voices of Mark Zuckerberg and Elon Musk. Videos taken by locals in Menlo Park, Palo Alto, and Redwood City in California show the crosswalk buttons playing AI-generated speech designed to sound like the two billionaires.
>
> One crosswalk button, hacked to sound like Mark Zuckerberg, was heard to say: *"It's normal to feel uncomfortable or even violated as we forcefully insert AI into every facet of your conscious experience. I just want to assure you, you don't need to worry because there's absolutely nothing you can do to stop it."*
>
> Another crosswalk button, hacked to sound like Elon Musk, said: *"I guess they say money can't buy happiness… I guess that's true. God knows I've tried. But it can buy a Cybertruck and that's pretty sick, right?"* Elon's voice then adds: *"I'm so alone."*
>
> It's not clear why the sidewalk buttons were hacked, or by whom, but signs point to possible hacktivism. Palo Alto Online, one of the first outlets to report the hack, cited a Redwood City official saying the city was *"actively working to investigate and resolve the issue as quickly as possible."* According to the outlet, the tampering may have happened on Friday.

TechCrunch's report finishes by providing some additional background, adding:

> *Audio-enabled crosswalk buttons are widely used across the United States to allow those with visual impairments or accessibility needs to hear custom audio messages that play for pedestrians to know when it is safe to cross a street. In a video from last year, physical penetration specialist and security researcher Deviant Ollam explains how audio-enabled crosswalk buttons can be manipulated, often by way of default-set passwords that have not been changed. Polara, the company that makes the audio-enabled crosswalk buttons, did not respond to a request for comment when contacted by TechCrunch on Monday.*

That's what we need more of ... a bit of non-malicious good old fashioned techno-pranking. At the same time, that capability could have just as easily been used to produce extremely offensive audio messages instead of having some fun spoofing Zuck and Musk. So a tip of the hat to the people behind that one, which will have also had the side benefit of firmly closing whatever back door had been inadvertently left open, thanks to their benign prank.

Android to begin force-restarting (ahead of schedule)

Last week I noted that features similar to Apple's "Lockdown Mode" were expected to be announced during next month's Google I/O 2025. It appears that one of those forthcoming features could not wait. The features for Google Play services v25.14, dated last Monday (2025-04-14) listed under "Security & Privacy" the following. It wrote:

- Enables a future optional security feature, which will automatically restart your device if locked for 3 consecutive days.

I'm not 100% clear about what it means to "enable a future optional security feature". The optional part I get – that's fine. I have no problem with that. But what exactly does it mean to "enable a future feature" ?? It's apparently now been enabled, which is why it's been listed. But if so, then how is that a future feature if it's already happened? It sounds like some change was made that we cannot actually use today, but we will be able to, optionally, in the future. In that case, who the hell cares? Why tell us anything about a future security feature that hasn't actually been enabled yet while we're all still back here in the past? I don't know.

The EFF's not one bit happy about new Florida social media legislation

I want to share a write-up by the EFF over their – let's just say – *"extreme unhappiness"* over new legislation that's being proposed in Florida. I'm glad we have a well-funded Electronic Frontier Foundation staffed with lawyers who know Constitutional Law. Here's what the EFF wrote:

> *At least Florida's SB 868/HB 743, "Social Media Use By Minors" bill isn't beating around the bush when it states that it would require "social media platforms to provide a mechanism to decrypt end-to-end encryption when law enforcement obtains a subpoena." Usually these sorts of sweeping mandates are hidden behind smoke and mirrors, but this time it's out in the open: Florida wants a backdoor into any end-to-end encrypted social media platforms that allow accounts for minors. This would likely lead to companies not offering end-to-end encryption to minors at all, making them less safe online.*
>
> *Encryption is the best tool we have to protect our communication online. It's just as important for young people as it is for everyone else, and the idea that Florida can "protect" minors by making them less safe is dangerous and dumb.*

> *The bill is not only privacy-invasive, it’s also asking for the impossible. As breaches like Salt Typhoon demonstrate, you cannot provide a backdoor for just the “good guys,” and you certainly cannot do so for just a subset of users under a specific age. After all, minors are likely speaking to their parents and other family members and friends, and they deserve the same sorts of privacy for those conversations as anyone else. Whether social media companies provide “a mechanism to decrypt end-to-end encryption” or choose not to provide end-to-end encryption to minors at all, there’s no way that doesn’t harm the privacy of everyone.*
>
> *If this all sounds familiar, that’s because we saw a similar attempt from an Attorney General in Nevada last year. Then, like now, the reasoning is that law enforcement needs access to these messages during criminal investigations. But this doesn’t hold true in practice.*
>
> *In our amicus brief in Nevada, we point out that there are solid arguments that “content oblivious” investigation methods—like user reporting— are “considered more useful than monitoring the contents of users’ communications when it comes to detecting nearly every kind of online abuse.” That remains just as true in Florida today.*
>
> *Law enforcement can and does already conduct plenty of investigations involving encrypted messages, and even with end-to-end encryption, law enforcement can potentially access the contents of most messages on the sender or receiver’s devices, particularly when they have access to the physical device. The bill also includes measures prohibiting minors from accessing any sort of ephemeral messaging features, like view once options or disappearing messages. But even with those features, users can still report messages or save them. Targeting specific features does nothing to protect the security of minors, but it would potentially harm the privacy of everyone.*
>
> *SB 868/HB 743 radically expands the scope of Florida’s social media law HB 3, which passed last year and itself has not yet been fully implemented as it currently faces lawsuits challenging its constitutionality. The state was immediately sued after the law’s passage, with challengers arguing the law is an unconstitutional restriction of protected free speech. That lawsuit is ongoing—and it should be a warning sign. Florida should stop coming up with bad ideas that can't be implemented.*
>
> *Weakening encryption to the point of being useless is not an option. Minors, as well as those around them, deserve the right to speak privately without law enforcement listening in. Florida lawmakers must reject this bill. Instead of playing politics with kids' privacy, they should focus on real, workable protections—like improving consumer privacy laws to protect young people and adults alike, and improving digital literacy in schools.*

I sure hope that the US Supreme Court doesn’t mind working and being busy, since I don’t recall any time during my life when more important and fundamental issues surrounding the shape of our collective future are being pushed up our legislative hierarchy for their final examination, hopefully some useful discussion and judgment. I sure hope they get these things right.

Pay up, or else!

I found an interesting piece of reporting which I had Firefox translate from its reporting in Dutch. After examining more than 500 ransomware incidents occurring between 2019 and 2023, a Dutch researcher found that ransomware victims who are insured against the cost of cybercrime incidents pay, on average, 2.8 times larger ransoms than those who are uninsured. And the bad guys know this, so they make a concerted effort to research and determine the cyber-insurance status of all potential targets. The researcher wrote: *“As soon as they have gained access to a* *system, they actively look for documents with names such as 'insurance' or 'policy'. This additional information gives cybercriminals a better bargaining position, leading to higher ransom payments."*

The research also found that companies with a well-designed backup system pay 27 times less often in the event of cyber attacks. The researcher wrote: *"Cybercriminals who are in a victim's network consciously look for backups, and remove them. Just having backups is not enough. It is important to have backups that cannot be adjusted by unauthorized persons in your network. Offline backups are the easiest solution for that, but I've also seen cloud solutions coming by."*

And the research also found that most companies have no choice other than to pay. The researcher wrote: *"In only around 5 out of 100 cases in which payments are made, victims do have the opportunity to recover in a different way than to pay, but choose to pay anyway – for example, to recover faster or to prevent reputational damage. In the other 95 out of 100 cases, there is no other option to recover. In those cases, their entire IT infrastructure is broken and no longer recoverable, making paying a ransom the only option to prevent bankruptcy."*

I suppose that it's not really surprising that 95% of ransomware victims do not have a sufficiently comprehensive or attack-proof backup system in place and so have no choice other than to give the extortionists whatever they demand. It's either that, go out of business, or start over. And we know from our own years of looking at this that the bad guys will also actively look for and work to eliminate any backup systems and servers that they can find, since they're also aware of that 95/5 rule. They very much want their victims to have no other recourse than to pay.

Windows Sandbox

Often ignored or unknown to most users of Windows 10 and 11, but probably of tremendous value and interest to the followers of this podcast, is that built right into every Win10 and 11, 64-bit Pro, Enterprise and Education operating system, is a ready-to-use, extremely robust, virtual machine based **full security sandbox** inside of which Windows users can perform any experiments they may wish where everything they or their experiments do will deliberately be “sandboxed” from the enclosing host PC and will therefore be unable to affect or in any way damage the hosting PC. This valuable security feature has been right there, available and in front of us since 2018 with the release of Windows 10 version 1903, and most of us have been unaware of it.

Microsoft describes this sandbox, writing:

> *Windows Sandbox is a lightweight, isolated desktop environment designed for safely running applications. It is ideal for testing, debugging, exploring unknown files, and experimenting with tools. Applications installed within the sandbox remain isolated from the host machine using hypervisor-based virtualization. As a disposable virtual machine (VM), Windows Sandbox provides quick launch times and a lower memory footprint compared to full VMs.*

And for key features, Microsoft highlights:

> - ***Part of Windows:*** *Everything required for this feature is included in supported Windows editions like Pro, Enterprise, and Education. There's no need to maintain a separate VM installation.*
> - ***Disposable:*** *Nothing persists on the device. Everything is discarded when the user closes the application.*
> - ***Pristine:*** *Every time Windows Sandbox runs, it's as clean as a brand-new installation of Windows.*
> - ***Secure:*** *Uses hardware-based virtualization for kernel isolation. It relies on the Microsoft hypervisor to run a separate kernel that isolates Windows Sandbox from the host.*
> - ***Efficient:*** *Takes a few seconds to launch, supports virtual GPU, and has smart memory management that optimizes memory footprint.*

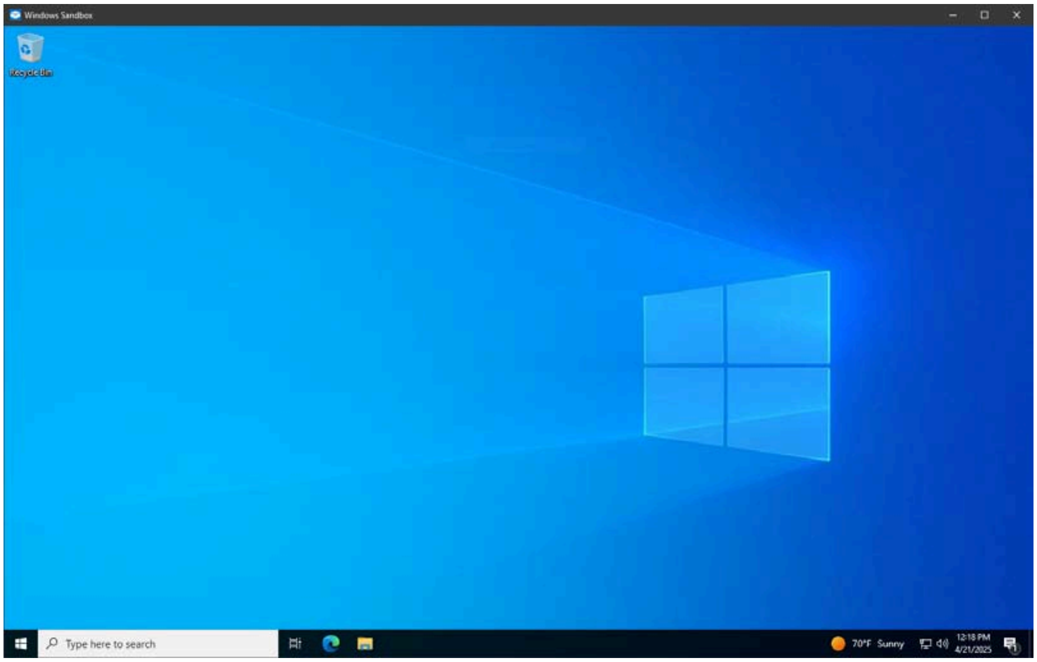
So this is clearly a win for anyone who might have any occasion to need a quick, safe, disposable instance of Windows. Perhaps you’d like to install something to see what it looks like, but it’s a big lumbering thing that’s likely to reconfigure and whack a big portion of your finely- tuned desktop, so you haven’t because the hassle of probably later uninstalling it and then perhaps recovering your machine from that experiment isn’t worth the trouble of satisfying your curiosity? With Windows Sandbox there’s nothing to uninstall. Just close the instance of Windows running in Windows and the monstrosity will be gone like it never existed the next time to use the sandbox. Or perhaps there’s a sketchy program you found on the Internet which you’d really like to run, but haven’t dared to on any machine that might be hurt by it. Or perhaps you need to poke into some particularly dark corners of the Internet, don’t want anything to poke you back, and want to leave absolutely no trace of ever having done so? It turns out that every non-Home edition of Windows has this capability built right in and ready to do all of those things.

And what’s so extra cool about this is that the Windows Sandbox is able to be far more efficient than a traditional full VM. It’s able to adjust its memory usage according to the demand and it doesn’t require an entire second installation of Windows since it’s able to reuse many of the host's read only operating system files. It’s really quite a slick solution. So where is it and how do you obtain the use of this little forgotten gem?

On the desktop, enter the search term “Windows Features” which will bring up the best match for “Turn Windows features on or off”. Selecting that will bring up a checkbox filled dialog with most features turned off. If you scroll down near the bottom of this alphabetized list you’ll see an item that you may have never noticed before, labeled “Windows Sandbox.” If you check the box to turn that feature on, then click the “OK” button to confirm your choices, Windows will spend a minute or two unpacking its bags and will then tell you that you need to reboot to finish things up.

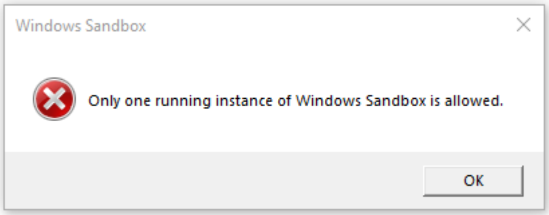
There’s a chance that you may find that feature greyed out and unselectable. If you hover your mouse over it, Windows will inform you that Windows Sandbox cannot be installed because the processor does not have the required virtualization capabilities. If that’s the case, you may be able to remedy that by rebooting, getting into your machine’s BIOS, and enabling the various processor virtualization features that are needed.

Once your main Windows is back from its reboot, scroll down to the “W’s” and you’ll find “Windows Sandbox” listed there. Click it and you’ll soon be presented with something you may dimly recall... your original Windows system before it was first touched:



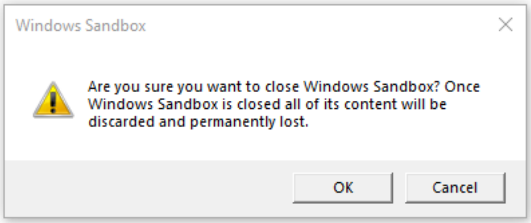
You’ll see a standard Windows window named “Windows Sandbox” with minimize, maximize and close icons in the upper right. I noticed that resizing the virtual machine Window was as smooth as any I’ve ever seen and maximizing the window completely replaced the underlying desktop and showed the “Remote Desktop Connection” bar at the top center. So Remote Desktop is the way the virtual machine’s desktop is being presented to the user.

The Sandbox has a C: drive with about 3 gigabytes in use and plenty of empty space, Internet access with a generic LAN adapter having a 172.17.*.* private IP, and a single user account named “WDAGUtilityAccount” where WDAG stands for Windows Defender Application Guard. Microsoft really appears to have done a nice job with this. I was curious to see what would happen if I attempted to launch a second instance of the Sandbox, and I was greeted with a dialog:


Windows Sandbox

Only one running instance of Windows Sandbox is allowed.

Out of curiosity, I tried clicking the upper-right close "X" and was told:


Windows Sandbox

Are you sure you want to close Windows Sandbox? Once Windows Sandbox is closed all of its content will be discarded and permanently lost.

And the second time the Windows Sandbox is launched its desktop pops right up, though that's somewhat misleading since Windows is not actually ready and does need a bit more time to actually boot. As the old timers among us will recall, at one point Microsoft was receiving so much flack over how long Windows was taking to boot that they deliberately engineered it to display its desktop at the earliest possible moment after booting and well before it was actually able to do anything. I always thought all that ingenuity would have been better spent actually making Windows boot faster, but no one asked me.

Before we dig under the covers to take a closer look at the technology behind this, let's look at some more of the surface details.

Windows Sandbox is also available on Arm64 from Windows 11, version 22H2 on. And starting with Win11 24H2, inbox store apps like Calculator, Photos, Notepad and Terminal are not available inside Windows Sandbox. The ability to use these apps is apparently coming soon.

A vGPU (virtualized GPU) is enabled on non-Arm64 devices.

As I noted, networking is enabled using the Windows Hyper-V default switch. Since this could potentially expose untrusted applications to the internal network it's possible to launch a Sandbox with networking disabled through the use of a custom .wsb file – as in "Windows Sand Box" configuration file. Audio input is enabled with the sandbox having access to the host's microphone input. However, video is not enabled. The sandbox doesn't share the host's video. Printer redirection is also disabled with the sandbox not sharing printers with the host. But clipboard redirection is enabled by default so that the host's clipboard is shared with the sandbox to allow text and files to be pasted back and forth.

It's also possible to change those defaults and many other aspects of the Sandbox's configuration: Windows Sandbox supports simple configuration files, which provide a minimal set of customization parameters for Sandbox. This feature can be used with Windows 10 build 18342 or Windows 11. Windows Sandbox configuration files are formatted as XML and are associated with Sandbox via the .wsb file extension.

A configuration file enables the user to control the following aspects of Windows Sandbox: The virtualized GPU can be disabled to cause the sandbox to use Windows Advanced Rasterization

Platform (WARP). Networking can be disabled. Mapped folders can be defined to allow the sandbox to see into the hosts's file system. A custom logon command to be executed when the Sandbox starts can be defined. The audio and video sharing defaults can be changed to disallow audio or allow video. The Remote Desktop Protocol's (RDP) "Protected client" mode can be engaged to place increased security settings on the Remote Desktop Protocol (RDP) session to the sandbox. Printers can be shared, the clipboard sharing can be disabled and the total amount of memory assigned to the Sandbox can be changed from its default of 4 gigabytes.

So now let's turn the clock back to December at the end of 2018 and look at what Microsoft shared about this terrifically useful innovation back then. The Windows OS Platform Blog posted, under the simple title "Windows Sandbox" that

> *Windows Sandbox is a new lightweight desktop environment tailored for safely running applications in isolation. How many times have you downloaded an executable file, but were afraid to run it? Have you ever been in a situation which required a clean installation of Windows, but didn't want to set up a virtual machine?*
>
> *At Microsoft we regularly encounter these situations, so we developed Windows Sandbox: an isolated, temporary, desktop environment where you can run untrusted software without the fear of lasting impact to your PC. Any software installed in Windows Sandbox stays only in the sandbox and cannot affect your host. Once Windows Sandbox is closed, all the software with all its files and state are permanently deleted.*
>
> *Since this is the Windows Kernel Internals blog, let's go under the hood. Windows Sandbox builds on the technologies used within Windows Containers. Windows containers were designed to run in the cloud. We took that technology, added integration with Windows 10, and built features that make it more suitable to run on devices and laptops without requiring the full power of Windows Server.*
>
> *Some of the key enhancements we have made include a* ***Dynamically generated Image***
>
> *At its core Windows Sandbox is a lightweight virtual machine, so it needs an operating system image to boot. One of the key enhancements we've made for Windows Sandbox is the ability to use a copy of the Windows 10 installed on your computer, instead of downloading a new VHD image as you would have to do with an ordinary virtual machine.*
>
> *We want to always present a clean environment, but the challenge is that some operating system files can change. Our solution is to construct what we refer to as a "dynamic base image": an operating system image that has clean copies of files that can change, but links to files that cannot change that are in the Windows image that already exists on the host. The majority of the files are links (immutable files) and that's why it has such a small size of around 100 megabytes for a full operating system. We call this instance the "base image" for Windows Sandbox.*
>
> *When Windows Sandbox is not installed, we keep the dynamic base image in a compressed package of around 25MB. When installed, the dynamic base package it occupies that 100MB of disk space.*
>
> ***What about memory management?*** *Memory management is another area where we have integrated with the Windows Kernel. Microsoft's hypervisor allows a single physical machine to be carved up into multiple virtual machines which share the same physical hardware. While that approach works well for traditional server workloads, it isn't as well suited to running*

> *devices with more limited resources. We designed Windows Sandbox in such a way that the host can reclaim memory from the Sandbox if needed.*
>
> *Additionally, since Windows Sandbox is actually running the same operating system image as the host, we allow Windows sandbox to use the same physical memory pages as the host for operating system binaries via a technology we refer to as "direct map". In other words, the same executable pages of NTDLL, are mapped into the sandbox as that on the host. We take care to ensure this done in a secure manner and no secrets are shared.*

Can everyone detect how utterly infatuated I am with this technology? It's genius. They're reusing all of Windows OS files. They're reusing all of the Windows' kernel's memory that's been loaded with static code. So any entirely separate and clean instance of Windows only requires around 100 megabytes of storage which is essentially a file system full of pointers into the host's file system. And rather that needing to create another full virtual machine with its own 4 gig or more of RAM, it also takes almost no RAM because it's able to map most of the host's actual physical RAM into its own virtual image. What a win!

And there's more, they write:

> *With ordinary virtual machines, Microsoft's hypervisor controls the scheduling of the virtual processors running in the VMs. However, for Windows Sandbox we use a new technology called "integrated scheduler" which allows the host to decide when the sandbox runs.*
>
> *For Windows Sandbox we employ a unique scheduling policy that allows the virtual processors of the sandbox to be scheduled in the same way as threads would be scheduled for a process. High-priority tasks on the host can preempt less important work in the sandbox. The benefit of using the integrated scheduler is that the host manages Windows Sandbox as a process rather than a virtual machine which results in a much more responsive host, similar to Linux KVM.*
>
> *The whole goal here is to treat the Sandbox like an app but with the security guarantees of a Virtual Machine.*

That's the genius of this. It really is running an entirely separate instance of Windows like an app on the underlying host OS. And remember how I mentioned that when I launched the Sandbox a second time it seemed to snap right up? This blog explains why I experienced that, too. It wasn't just my imagination (or infatuation!). They wrote:

> *As stated above, Windows Sandbox uses Microsoft's hypervisor. We're essentially running another copy of Windows which needs to be booted and this can take some time. So rather than paying the full cost of booting the sandbox operating system every time we start Windows Sandbox, we use two other technologies; "snapshot" and "clone."*
>
> ***Snapshot*** *allows us to boot the sandbox environment once and preserve the memory, CPU, and device state to disk. Then we can restore the sandbox environment from disk, loading it directly into the device memory, rather than booting it, when we need a new instance of Windows Sandbox. This significantly improves the start time of Windows Sandbox.*
>
> *And Graphics virtualization: Hardware accelerated rendering is key to a smooth and responsive user experience, especially for graphics-intense or media-heavy use cases. However, virtual machines are isolated from their hosts and unable to access advanced devices like GPUs. The*

> *role of graphics virtualization technologies, therefore, is to bridge this gap and provide hardware acceleration in virtualized environments. More recently, Microsoft has worked with our graphics ecosystem partners to integrate modern graphics virtualization capabilities directly into DirectX and WDDM, the driver model used by display drivers on Windows. Graphics components in the Sandbox, which have been enlightened to support virtualization, coordinate across the VM boundary with the host to execute graphics workloads. The host allocates and schedules graphics resources among apps in the VM alongside the apps running natively. Conceptually they behave as one pool of graphics clients.*
>
> *This enables the Windows Sandbox VM to benefit from hardware accelerated rendering, with Windows dynamically allocating graphics resources where they are needed across the host and guest. The result is improved performance and responsiveness for apps running in Windows Sandbox, as well as improved battery life for graphics-heavy use cases.*
>
> *To take advantage of these benefits, you'll need a system with a compatible GPU and graphics drivers (WDDM 2.5 or newer). Incompatible systems will render apps in Windows Sandbox with Microsoft's CPU-based rendering technology.*
>
> *And finally, Battery pass-through: Windows Sandbox is also aware of the host's battery state, which allows it to optimize power consumption. This is critical for a technology that will be used on laptops, where not wasting battery is important to the user.*

I've been spending a lot of time recently inside Virtual Machines. The DNS Benchmark that I'm currently working on needs to run under Windows 7, 8, 10 and 11, and those four operating systems span enough time that their behavior is all slightly different. So I'm routinely launching and running different OSes on different platforms. When I originally built my main old Windows 7 machine, I expected virtualization to be a thing, so I gave it a whopping 128 gigabytes of RAM. This was specifically so that I could fire up separate Windows virtual machines that would each need large chunks of RAM dedicated to them. And my Windows 10 machine has 32 gigabytes, which was the most that the Intel NUC could handle at the time. My point is that I've become quite accustomed to the feeling of virtual machines running on top of my desktop and I've never experienced as seamless and smooth operation of a Windows OS as is provided by this built-in Windows Sandbox.

Microsoft has really outdone themselves on this one. They've been very clever and they've done everything right. They've essentially figured out how to run an entire separate instance of Windows as an application on top of Windows. It's fast and lightweight and doesn't burn up disk space or RAM.

Toward the end of today's show notes I have a collection of links to additional resources to help everyone get the most out of their built-in Windows Sandbox, including all of the documentation about configuring and tweaking its operation, RAM, cross-host resource sharing, and everything else:

https://techcommunity.microsoft.com/blog/windowsosplatform/windows-sandbox/301849

https://learn.microsoft.com/en-us/windows/security/application-security/application-isolation/windows-sandbox/

https://learn.microsoft.com/en-us/windows/security/application-security/application-isolation/windows-sandbox/windows-sandbox-install

The Custom Config File:
https://learn.microsoft.com/en-us/windows/security/application-security/application-isolation/windows-sandbox/windows-sandbox-configure-using-wsb-file

Command Line Interface:
https://learn.microsoft.com/en-us/windows/security/application-security/application-isolation/windows-sandbox/windows-sandbox-cli

There are so many compelling use-cases for this slick technology that I wanted to make sure all of the listeners of this podcast, who use Windows as their primary desktop, knew that this little gem was hidden inside there, just waiting to come out to play.

I’m finally somewhat jealous of Windows 10, since this is a feature Windows 7 never even dreamed of having. For the first time ever, I’m considering making an image of my Win7 system – as an emergency fallback – then seeing how Windows does upgrading this old Win7 machine “in place” to Windows 10. I really dread the downtime required to set up a new Windows 10 machine from scratch and reinstall everything. But Windows Sandbox has been implemented so beautifully that it’s something I’d love to have in my desktop platform.