Security Now! #1017 - 03-18-25

Is YOUR System Vulnerable to RowHammer?

This week on Security Now!

An analysis of Telegram Messenger's crypto. A beautiful statement of the goal of modern crypto design. Who was behind Twitter's recent outage trouble? An embedded Firefox root certificate expired. Who was surprised? AI-generated Github repos, voice cloning, Patch Tuesday and an Apple 0-day. The FBI warns of another novel attack vector that's seeing a lot of action. Google weighs in on the Age Verification controversy. In a vacuum, Kazakhstan comes up with their own solution. Was Google also served an order from the UK? Can they say? A serious PHP vulnerability you need to know you don't have. A bunch of great listener feedback, some Sci-Fi content reviews and… A new tool allows YOU to test YOUR PCs for their RowHammer susceptibility.

The Nature of Legacy Technology

Security News

Analysis of the Telegram Key Exchange

Our listeners possessing long memories may recall how repulsed I was by Telegram's design the first time I looked at it and we talked about it. It was a pile of made up nonsense. And since that was the general impression shared by the informed crypto community, eleven years ago, back in 2014, Pavel Durov's response to the community shunning his solution was to offer a $200,000 reward to anyone who could decipher an encrypted message sent between two Telegram users. Again, the crypto community was unimpressed because that was beside the point. By 2014 we already knew how to solve these problems correctly, and Telegram wasn't it.

For this reason, I was interested and I knew our listeners would be, when I saw that a team of actual cryptographers had finally taken a good hard long look at what can best be described as the "Ad Hoc" cryptography invented by Telegram. I use the phrase "actual cryptographers" because the first thing that becomes clear to anyone looking at Telegram is that its designers were not.

Five cryptographers, one from King's College London, two from ETH Zurich, one from Tel-Aviv University and the 5th from Amazon, last Monday published a paper containing their findings which was presented during the EUROCRYPT 2025 cryptography conference: https://eprint.iacr.org/2025/451.pdf Their paper's title was ***"Analysis of the Telegram Key Exchange"*** and its Abstract reads:

> *We describe, formally model, and prove the security of Telegram's key exchange protocols for client-server communications. To achieve this, we develop a suitable multi-stage key exchange security model along with pseudocode descriptions of the Telegram protocols that are based on analysis of Telegram's specifications and client source code. We carefully document how our descriptions differ from reality and justify our modelling choices. Our security proofs reduce the security of the protocols to that of their cryptographic building blocks, but the subsequent analysis of those building blocks requires the introduction of a number of novel security assumptions, reflecting many design decisions made by Telegram that are suboptimal from the perspective of formal analysis. Along the way, we provide a proof of the security for the variant of RSA-Optimal Asymmetric Encryption Padding+ used in Telegram, and identify a hypothetical attack exploiting current Telegram server behaviour (which is not captured in our protocol descriptions). Finally, we reflect on the broader lessons about protocol design that can be taken from our work.*

Then, one hundred and four pages later — this was not a short paper — they conclude under the poetic heading ***"The brittle monolith that is Telegram"***. But it's not just their heading that's poetic. Listen carefully to how beautifully they describe the way cryptographic protocols should be designed, versus what they found lurking in the heart of Telegram. They concluded:

> *In theory, the design of a cryptographic protocol has the sole purpose of achieving the protocol's security goals efficiently. In actuality, however, to achieve this goal it must also achieve the goal of allowing at least a sufficiently motivated expert to convince themselves that the protocol achieves these goals. In other words, the central insight of what is commonly referred to as "modern cryptography" is that a cryptographic design is also tasked with being easy to reason about. A fundamental paradigm of achieving this goal is modularity, where different components of the design can be reasoned about in isolation and then (generically) composed to establish overall security guarantees. This modularity is typically achieved by relying on building blocks that provide strong security guarantees on their own (as opposed to*

> *only and potentially in specific compositions) and by breaking the dependency between different components of a protocol by avoiding re-use of secret material. Telegram's failure to achieve this design goal is the root cause for the limitations and complexity of our proofs and our seeming need to reach for unstudied assumptions on cryptographic building blocks than would otherwise be necessary. We will now discuss these issues and highlight several of the main Telegram design choices and their effect on our proofs of security. We begin with mere complications, then move on to limitations and seemingly necessary ad-hoc assumptions. We finish by briefly recapping our hypothetical attack. We also discuss design choices that led to these issues and note that the same design choice often lead to several different difficulties for arguing for the security of Telegram, leading to necessary repetitions in what follows.*

And, a bit later, under the heading ***"Reliance on unstudied assumptions"*** they add:

> *In Appendix C we describe several unstudied ad-hoc and new assumptions that we used in our proofs. These assumptions could have been avoided if collision-resistant hash functions (e.g. SHA-256 or SHA3) had been used instead of SHA-1 and if proper key derivation functions had been used.*

In other words, the cryptographic design of Telegram is a mess at a time when "a mess" can, and for very good reasons should, be avoided. Telegram is likely secure enough for everything and everyone who's using and replying on it. But its design actively fights against that ever being proven.

The Twitter DDoS Outages last Monday

Those of us who watched the early rise of Twitter will recall the frequently seen "Fail Whale". Its appearance usually indicated that the service, which was struggling to grow fast enough to keep up with its exploding demand, was temporarily unable to do so. But those days are now long past.

Last Week, Twitter was on the receiving end of a widespread high bandwidth DDoS attack. And as we know, widely sourced very high bandwidth attacks are what's now required to take major sites and services down. In the case of last week's attacks, those who track such things saw massive traffic originating from IP addresses in the United States, Vietnam, and Brazil, among other countries. So I was annoyed when Elon Musk later told Larry Kudlow, during an interview on Fox Business Network, that the attack came from Ukrainian IP addresses.

What actually happened was that a group which offers DDoS attacks for hire named Dark Storm Team, took credit for Twitter's Monday outages. I don't have any problem when someone has a differing opinion. But Elon could have either said nothing, or said he didn't know where the attack originated or why it was launched. It would have been even better, and accurate, to simply say that like most modern attacks, they come from all over the globe. I get it that he's very busy. And he likely didn't actually have any information at all. He shouldn't be expected to know everything. But singling out and naming Ukraine as the source of the attack was not true – at least from a bandwidth standpoint which is knowable. And doing so appears to serve a current political agenda.

Firefox root certificate expiration

Last Friday, a critical Firefox root certificate expired. Earlier last week Mozilla wrote:

> *On March 14, 2025, a root certificate used to verify signed content and add-ons for various Mozilla projects, including Firefox, will expire. Without updating to Firefox version 128 or higher (or ESR 115.13+ for ESR users, including Windows 7/8/8.1 and macOS 10.12–10.14 users), this expiration may cause significant issues with add-ons, content signing and DRM-protected media playback.*
>
> *If you don't update, Firefox features that rely on remote updates will stop working, and your installed add-ons will be disabled. DRM-protected content, such as streaming services, may also stop playing due to failed updates. Additionally, systems dependent on content verification could stop functioning properly.*
>
> *This update is necessary for all Firefox users running versions earlier than 128 (or ESR versions earlier than ESR 115.13), including those using Firefox for Desktop on Windows, macOS and Linux, as well as Firefox for Android. If you were sent to this article through an in-app message in Firefox, it means your browser version is outdated and needs to be updated.*

Since I'm still using Firefox on a Windows 7 machine I was initially concerned. But I just checked and my ESR edition had already updated itself past that point. It's at v115.21.0esr. And in researching this further, it became clear that unlike those sites whose TLS certificate expirations catch them by surprise, Mozilla was not taken by surprise by this. The mainstream v128 edition and that ESR release which Mozilla said would be needed, v115.13, were both first made available on July 9th of last year, 2024. So anyone who hasn't updated their Firefox even ***once*** since then would have no one to blame other than themselves if something were to go wonky. This meant that Mozilla was just reminding everyone a few days before that certificate, which was formally retired nine months ago, that if for any reason someone was still running a Firefox from before last summer, various important things might stop working.

AI-generated GitHub malware repos

We knew it was going to happen. And it's also probably little surprise that it happened not long after AI became the big buzzword. An unknown threat actor has deployed a large number of malicious GitHub repositories which infect users with malware. Trend Micro says descriptions for the repositories have been generated using AI tools. The malicious repositories infect users with the SmokeLoader, which then deploys the LummaStealer malware to exfiltrate user credentials.

Wanna clone someone's voice? It's not difficult...

A Consumer Reports study found that Speechify, Lovo, PlayHT, and Descript made no efforts to ensure that users had consent to reproduce another person's voice. These four out of the top six most popular cloning apps incorporate no protections against abuse. They allow threat actors to easily clone anyone's voice. Consumer Reports study also found that voice cloning scams are seeing a wider adoption across the fraud landscape.

Microsoft Patch Tuesday

Last Tuesday Microsoft patched a modest 58 vulnerabilities among which were 6 that were actively exploited 0-days:

- CVE-2025-24983 — Windows Win32 Kernel Subsystem Elevation of Privilege Vulnerability
- CVE-2025-24984 — Windows NTFS Information Disclosure Vulnerability
- CVE-2025-24985 — Windows Fast FAT File System Driver Rem Code Execution Vulnerability

- CVE-2025-24991 — Windows NTFS Information Disclosure Vulnerability
- CVE-2025-24993 — Windows NTFS Remote Code Execution Vulnerability
- CVE-2025-26633 — Microsoft Management Console Security Feature Bypass Vulnerability

Apple zero-day

Apple also patched a 0-day in their WebKit affecting both iOS and macOS. Apple described the attacks as "extremely sophisticated."

FBI Warns of Online File Converter Scam

The U.S. Federal Bureau of Investigation posted a public service announcement:

> *The FBI is warning that agents are increasingly seeing scams involving free online document converter tools, and we want to encourage victims to report instances of this scam. In this scenario, criminals use free online document converter tools to load malware onto victims' computers, leading to incidents such as ransomware.*
>
> *FBI Denver Special Agent in Charge Mark Michalek said: "The best way to thwart these fraudsters is to educate people so they don't fall victim in the first place. If you or someone you know has been affected by this scheme, we encourage you to make a report and take actions to protect your assets. Every day, we are working to hold these scammers accountable and provide victims with the resources they need."*
>
> *To conduct this scheme, cyber criminals across the globe are using any type of free document converter or downloader tool. This might be a website claiming to convert one type of file to another, such as a .doc file to a .pdf file. It might also claim to combine files, such as joining multiple .jpg files into one .pdf file. The suspect program might claim to be an MP3 or MP4 downloading tool.*
>
> *These converters and downloading tools will do the task advertised, but the resulting file can contain hidden malware giving criminals access to the victim's computer. The tools can also scrape the submitted files for:*
>
> - *Personal identifying information, such as social security numbers, dates of birth, phone numbers, etc.)*
> - *Banking information*
> - *Cryptocurrency information (seed phrases, wallet addresses, etc.)*
> - *Email addresses*
> - *Passwords*
>
> *Unfortunately, many victims don't realize they have been infected by malware until it's too late, and their computer is infected with ransomware or their identity has been stolen. The FBI Denver Field Office encourages victims or attempted victims of this type of scheme to report it to the FBI Internet Crime Complaint Center at www.ic3.gov.*

I wanted to share this because it's not a source of malware infection or attack that we had ever talked about and it had never occurred to me. It makes sense that such services would be abused, since Googling "how do I convert format 'X' into format 'Y'" typically returns links to free online services that promise to allow their visitors to make such conversions without the overhead and hassle of installing yet another piece of software. Consequently, unwitting searchers are often directed to these sites by their favorite Internet search engines.

Meta loses its last appeal
The top court in South Korea rejected Meta's final attempt to dismiss a $4.6 million fine. Five years ago, South Korea's privacy watchdog fined Meta in 2020 for sharing the data of 3.3 million South Koreans with third parties without authorization. Meta lost and now they must pay. This is, of course, a drop in the bucket for Meta, but the principle matters since it sets precedent and the world is becoming less tolerant of mega corps doing whatever they want.

Google weighs in on the their side of the Age Verification requirements
And speaking of Meta, Google is reported to be extremely upset over Meta's sponsorship and push for that Utah age-verification bill that recently moved through Utah's legislature. As we know, it transfers the responsibility of the task of checking a suspected child account to the application stores rather than to individual apps.

Last week we looked at what Apple was doing, and last Wednesday Google posted their position under the title *"Google's legislative proposal for keeping kids safe online"* and in an indication of Google's annoyance with Meta, the tag line read: *"Legislation pushed by Meta would share kids' information with millions of developers without parental consent or rules on how it's used; we have a better way."* Google said the following:

> *Everyone wants to protect kids and teens online, and make sure they engage with age-appropriate content, but how it's done matters. There are a variety of fast-moving legislative proposals being pushed by Meta and other companies in an effort to offload their own responsibilities to keep kids safe to app stores. These proposals introduce new risks to the privacy of minors, without actually addressing the harms that are inspiring lawmakers to act. Google is proposing a more comprehensive legislative framework that shares responsibility between app stores and developers, and protects children's privacy and the decision rights of parents.*
>
> *One example of concerning legislation is Utah's App Store Accountability Act. The bill requires app stores to share if a user is a kid or teenager with* ***all*** *app developers (effectively millions of individual companies) without parental consent or rules on how the information is used. That raises real privacy and safety risks, like the potential for bad actors to sell the data or use it for other nefarious purposes.*
>
> *This level of data sharing isn't necessary — a weather app doesn't need to know if a user is a kid. By contrast, a social media app does need to make significant decisions about age-appropriate content and features. As written, however, the bill helps social media companies avoid that responsibility despite the fact that apps are just one of many ways that kids can access these platforms. And by requiring app stores to obtain parental consent for every single app download, it dictates how parents supervise their kids and potentially cuts teens off from digital services like educational or navigation apps.*
>
> *By contrast, we are focused on solutions that require appropriate user consent and minimize data exposure. Our legislative framework, which we'll share with lawmakers as we continue to engage on this issue, has app stores securely provide industry standard age assurances only to developers who actually need them — and ensures that information is used responsibly. Here are more details:*
>
> - *Privacy-preserving age signal shared only with consent: Some legislation, including the Utah bill, require app stores to send age information to all developers without permission from the user or their parents. In our proposal, only developers who create apps that may*

> *be risky for minors would request industry standard age signals from app stores, and the information is only shared with permission from a user (or their parent). By just sharing with developers who need the information to deliver age-appropriate experiences, and only sharing the minimum amount of data needed to provide an age signal, it reduces the risk of sensitive information being shared broadly.*
>
> - *Appropriate safety measures within apps: Under our proposal, an age signal helps a developer understand whether a user is an adult or a minor — the developer is then responsible for applying the appropriate safety and privacy protections. For example, an app developer might filter out certain types of content, introduce take a break reminders, or offer different privacy settings when they know a user might be a minor. Because developers know their apps best, they are best positioned to determine when and where an age-gate might be beneficial to their users, and that may evolve over time, which is another reason why a one-size-fits-all approach won't adequately protect kids.*
>
> - *Responsible use of age signals: Some legislative proposals create new child safety risks because they establish no guardrails against developers misusing an age signal. Our proposal helps to ensure that any age signals are used responsibly, with clear consequences for developers who violate users' trust. For example, it protects against a developer improperly accessing or sharing the age signal.*
>
> - *No ads personalization to minors: Alongside any age assurance proposal, we support banning personalized advertisements targeting users under 18 as an industry standard. At Google, this is a practice we've long disallowed. It's time for other companies to follow suit.*
>
> - *Centralized parental controls: Recognizing that parents sometimes feel overwhelmed by parental controls across different apps, our proposal would provide for a centralized dashboard for parents to manage their children's online activities across different apps in one place and for developers to easily integrate with.*
>
> *Google has demonstrated our commitment to doing our part to keep kids safe online. We're ready to build on this work and will continue engaging with lawmakers and developers on how to move this legislative framework for age assurance forward.*

With Apple and Google being the two gorillas in the market, they appear to be converging onto the same solution. Essentially, parents are able to group the phones of their family members and indicate which phones belong to their minor children. Once this is done, children wishing to download applications with mature ratings will require parental consent. Developers of restricted apps have no need to know anything about those who are downloading and installing their apps. The fact that they are able to do so means that they have permission – either by using an adult's phone, or because a parent or guardian gave a child permission.

Meanwhile, Kazakhstan has a different approach

The Kazakhstan government has introduced SIM cards specifically designed for the use of and by children. All parents will be required to buy and deploy the new SIM cards for use in their children's devices. The cards come with built-in filters to restrict access to dangerous websites and social media. The cards also report a child's location to parents through a special app.

It feels as though things are rapidly becoming a mess with random and uncoordinated legislation being created left and right. I lay this mess at the feet of Apple and Google who resisted taking the action they could and should have taken on this many years ago. As a consequence of a total lack of responsibility taking on the part of the major platform providers legislators have been left with no choice other than to take matters into their own hands. No one wins here.

Spain's government to fine unlabelled AI content

The Spanish government passed a bill last week to impose very stiff fines on companies that produce and dispense unlabelled AI-generated content. And when I say "stiff fines" we're talking up to €35 million or 7% of a company's global annual revenue. The law hopes to curb the spread of deepfakes and non-consensual adult content such as producing fake celebrity videos. Spain is the first country in the EU bloc to incorporate provisions from the EU AI Act into its national legislation.

Google and the Canary

Last Friday, "The Record" ran a piece that caught my eye. In the wake of what has become an extremely public withdrawal of enabling Apple's strongest privacy guarantees for iCloud backup in the UK, many have wondered – including elected members of US legislation – about Android and Google? What's their similar status relative to the United Kingdom of the even larger Android ecosystem being designed and managed by Google? So The Record gave their coverage of this question the headline: *"Google refuses to **deny** it received encryption order from UK government"*... and apparently they've been asked directly and pointedly. The Record wrote:

> *Google has refused to deny receiving a secret legal order from the British government, according to a bipartisan group of members of Congress who are concerned Westminster may have demanded that several U.S. technology companies provide its security services with a mechanism to access encrypted messages.*
>
> *It follows the British government reportedly issuing such a secret legal demand, officially known as a Technical Capability Notice (TCN), to Apple. Apple is believed to be contesting the demand at a closed court hearing on Friday.* [It's unclear which Friday they're referring to.]
>
> *In a letter published Thursday, the members of Congress complained about the secrecy of this court hearing, arguing it "impedes Congress's power to conduct oversight, including by barring U.S. companies from disclosing foreign orders that threaten Americans' privacy and cybersecurity."*
>
> *Despite widespread reporting of the TCN issued to Apple, the company is prohibited from confirming whether it had received such an order under the U.K.'s Investigatory Powers Act. In their letter, the members of Congress wrote that Apple had informed them "that had it received a technical capabilities notice, it would be barred by U.K. law from telling Congress whether or not it received such a notice."*
>
> *Companies who have **not** received such a notice are obviously free to state so.*
>
> *The group wrote: "Google also recently told Senator [Ron] Wyden's office that, if it had received a technical capabilities notice, it would be prohibited from disclosing that fact."*
>
> *Experts, including from Britain's own intelligence community, have said that the government's attempts to access encrypted messaging platforms should be more transparent. Academics described the Home Office's ongoing refusal to either confirm or deny the legal demand as unsustainable and unjustifiable.*

So what does this mean? I am here to formally let everyone who is listening to this podcast know that I have **not** in receipt of **any** such or similar demand from the UK government. And Leo, I imagine you're equally free to say the same thing. (Not that the UK government would be expected to have any interest in anything either of us may have encrypted.) So the point is, isn't this a reverse canary?

Doesn't Google's refusal to simply say, as I just have, that they're **not** in receipt of an order which compels them to not disclose such an order, automatically mean that they **are** in receipt of a similar order from the UK? And wouldn't that also make sense? Wouldn't we expect Google to be just as much subject to this as Apple? And if Google were not; if the UK only required Apple to comply; wouldn't that constitute unfair meddling in the direct commercial interests of these two commercial platforms? Forcing Apple to be able to decrypt the confidential and private information of their users, while not requiring exactly the same from others, would put Apple at a significant commercial disadvantage relative to its competitors.

It seems clear that whereas news of Apple's receipt of this leaked out, the same may have happened within Google but it hasn't leaked. And some have suggested that Apple's leakage may have originated within Apple itself as a means of opening this issue to the disinfecting light of day.

And this brings us to another piece of related reporting from The Record, which they posted last Thursday, the day before:

"Calls grow for UK to move secret Apple encryption court hearing to public session"
The Record wrote:

> *Politicians and civil society groups in the United Kingdom are calling for a secret court hearing expected on Friday about the British government's encryption demands on Apple to be held in public. It follows warnings from experts, including from Britain's own intelligence community, that the government's attempts to access encrypted messaging platforms should be more transparent. Academics described the Home Office's ongoing refusal to either confirm or deny the legal demand as unsustainable and unjustifiable.*
>
> *The Schedule for the Investigatory Powers Tribunal — the only court in the country that can hear certain national security cases — includes a hearing set to take place behind closed doors on Friday,* [Presumably last Friday] *featuring the Tribunal's president, Lord Justice Singh, alongside the senior High Court judge Justice Johnson. It follows Apple disabling the option for its British users to protect their iCloud accounts with end-to-end encryption last month, in the wake of a reported legal order from the British government requiring Apple provide it with access to encrypted iCloud accounts. The hearing is purportedly the company's attempt to contest this order, although it is unknown on what legal grounds that attempt is being made.*
>
> *The British government continues to say it neither confirms nor denies the existence of such legal demands. Apple has not confirmed the reason the encryption feature was turned off, and would be prohibited from doing so if it were due to a Technical Capability Notice, but stressed when it announced the move that* ***"we have never built a backdoor or master key to any of our products or services and we never will."***
>
> *In a joint letter on Thursday to Lord Justice Singh, a collection of British civil liberties groups asked him to use his discretion to open the hearing to the public, arguing that doing so would not prejudice national security.*

> *The campaigners wrote: "There is significant public interest in knowing when and on what basis the UK government believes that it can compel a private company to undermine the privacy and security of its customers." They argued that there are "no good reasons to keep this hearing entirely private" given that the existence of the secret legal order has been publicly reported and effectively confirmed by Apple's decision to remove its end-to-end encrypted service for British iCloud users.*
>
> *Politicians from opposition parties, including the Conservative Party, Liberal Democrats and Reform, have also called for more transparency from the Home Office. David Davis, a Conservative Party politician who has long campaigned to limit state surveillance powers, told Sky News the government needed to explain its case to the public if it wants "effectively unfettered access" to private data.*

So this is all good. This is what we need to have happening because this all needs to be decided one way or the other. And, importantly, since the delivery of privacy and confidentiality is a commercial competitive attribute, whatever the rules finally turn out to be must be applied to all parties equally and evenly. At this point, nothing about this process of secret UK government compulsion can become or remain the status quo.

Attacks using a serious PHP bug are ramping up

Before we get to some feedback from our listeners, I want to make absolutely certain that anyone who's responsible for any PHP-based Windows web servers – as I am now with GRC's web forums, our eMailing system, the GRC.sc link redirector, and so forth – is not vulnerable to a very very serious PHP vulnerability, the exploitation of which has recently ramped way up after its first disclosure last summer, the summer of 2024.

The good news is that of the several ways the PHP interpreter can be invoked, only the oldest original method of using the php-cgi.exe executable gateway (or if the php.exe itself is placed into the php-cgi directory) is vulnerable. None of the newer approaches including Mod-PHP, FastCGI, or PHP-FPM are vulnerable. However, on Windows the common use of the so-called XAMPP stack **is** vulnerable in its default configuration because it uses the php-cgi executable to invoke the PHP interpreter. XAMPP refers to the Apache web server, the MariaDB database and interpreters for PHP and Perl.

I breathed a personal sigh of relief at this, since all of GRC's many web servers have always been configured to use the FastCGI method of invoking PHP. Before I talk about this further, the only solution is to move to the current release of a supported PHP, which means PHP v8.1.29 or later, v8.2.20 or later, or v8.3.8 or later. Unfortunately, this leaves a massive population of publicly exposed PHP servers vulnerable to complete system takeover. So here's the backstory on this.

The news that put me onto this was also just published by The Record. They wrote:

> *Researchers said Friday that a vulnerability initially exploited mostly in cyberattacks against Japanese organizations is now a potential problem worldwide. Threat intelligence company GreyNoise said exploitation of the bug, tracked as CVE-2024-4577, "extends far beyond initial reports," referencing in particular a blog post published Thursday by Cisco Talos. The Talos team had said an unknown attacker was "predominantly targeting organizations in Japan" in January through the vulnerability, which affects a setup called PHP-CGI that runs scripts on web servers. A patch was issued last summer.*

> *Cisco Talos said the attacker's apparent goal was to steal access credentials and potentially establish persistence in a system, "indicating the likelihood of future attacks." GreyNoise said it observed similar activity beyond Japan, revealing "a far wider exploitation pattern demanding immediate action from defenders globally."*
>
> *There are 79 known ways to exploit the vulnerability and remotely execute code on a compromised system, GreyNoise said. The PHP scripting language is decades old and is widely used in web development. <quote> "Attack attempts have been observed across multiple regions, with notable spikes in the United States, Singapore, Japan, and other countries throughout January 2025." Cisco Talos said Thursday that the attacker it studied used a command and control (C2) server that deploys a full suite of adversarial tools and frameworks. The researchers said they believed the attacker's motive was to move beyond just stealing credentials. Researchers at Symantec had reported exploitation of CVE-2024- 4577 last August, against a university in Taiwan, not long after the patch was issued.*

The discovery of this is credited to an old friend of ours whom we haven't heard much from recently, good old Orange Tsai at Devcore. In just the previous four years he's won:

2021 - 28th of Top 100 Microsoft Most Valuable Security Researchers
2021 - Champion of Pwn2Own Vancouver
2021 - 3rd of Top 10 Web Hacking Techniques for Exchange Server RCEs
2021 - Winner of Pwnie Awards — "Best Server-Side Bug" for Exchange Server RCEs
2022 - Champion of Pwn2Own Toronto
2024 - 1st of Top 10 Web Hacking Techniques for research of Confusion Attacks
2024 - 4th of Top 10 Web Hacking Techniques for research of WorstFit Attack

So last June 6th, when Devcore published their Security Alert titled: CVE-2024-4577 - PHP CGI Argument Injection Vulnerability, it drew the security industry's attention. They opened with:

> *During DEVCORE's continuous offensive research, our team discovered a remote code execution vulnerability in PHP. Due to the widespread use of the programming language in the web ecosystem and the ease of exploitability, DEVCORE classified its severity as critical, and promptly reported it to the PHP official team. The official team released a patch on 2024/06/06. Please refer to the timeline for disclosure details.*

And in their published timeline we see the way this is supposed to go. For one thing, the PHP developers well understand the nature of critical bugs – can you say "interpreter" ?? And secondly, they all know Orange Tsai and Devcore. So when you get a universal scope bug report marked "CRITICAL" from those guys, your plans for the next several days just changed.

- 2024/05/07 - DEVCORE reported this issue through the official PHP vulnerability disclosure page.
- 2024/05/07 - (SAME DAY) PHP developers confirmed the vulnerability and emphasized the need for a prompt fix.
- 2024/05/16 - (9 days later) PHP developers released the first version of the fix and asked for feedback.
- 2024/05/18 - (2 days later) PHP developers released the second version of the fix and asked for feedback.
- 2024/05/20 - (and another 2 days later) PHP entered the preparation phase for the new version release.
- 2024/06/06 - PHP released new versions 8.3.8, 8.2.20, and 8.1.29.

Under “Description” they explained:

> *While implementing PHP, the team did not notice the Best-Fit feature of encoding conversion within the Windows operating system. This oversight allows unauthenticated attackers to bypass the previous protection of CVE-2012-1823 by specific character sequences. Arbitrary code can be executed on remote PHP servers through the argument injection attack.*

In other words, this PHP bug was originally found and fixed 13 years ago. But Windows employs its own “best-fit” UNICODE character conversion feature and Orange Tsai discovered that many many (apparently 79) other deliberately crafted UNICODE character sequences could be used to disable and bypass the original php-cgi command-injection vulnerability.

This thing is so bad that, for example, a single query remotely issued to any vulnerable Windows web server can cause it to fetch any remote file named in the query and then execute that file, no matter what it might be, on the vulnerable machine. That is NOT anything that anyone wants to have happen.

Under the “Impact” section they were very clear, writing:

> *This vulnerability affects all versions of PHP installed on the Windows operating system.*

They also noted:

> *Since the branch of PHP 8.0, PHP 7, and PHP 5 are End-of-Life, and are no longer maintained anymore, server admins can refer to the Am I Vulnerable section to find temporary patch recommendations in the Mitigation Measure section.*

And in that “Am I Vulnerable?” section they wrote:

> *For the usual case of combinations like Apache HTTP Server and PHP, server administrators can use the two methods listed in this article to determine whether their servers are vulnerable or not. It’s notable to address that Scenario-2 is also the default configuration for XAMPP for Windows, so all versions of XAMPP installations on Windows are vulnerable by default.*
>
> *As of this writing, it has been verified that when the Windows is running in the following locales, an unauthorized attacker can directly execute arbitrary code on the remote server:*
>
> - *Traditional Chinese (Code Page 950)*
> - *Simplified Chinese (Code Page 936)*
> - *Japanese (Code Page 932)*
>
> *For Windows running in other locales such as English, Korean, and Western European, due to the wide range of PHP usage scenarios, it is currently not possible to completely enumerate and eliminate all potential exploitation scenarios. Therefore, it is recommended that users conduct a comprehensive asset assessment, verify their usage scenarios, and update PHP to the latest version to ensure security.*

That was written last June at release time. Since then it’s been widely confirmed that this vulnerability can be exploited anywhere and on any vulnerable server regardless of local language configuration. Therefore, by far the safest and most recommended mitigation is to update to a version of PHP that once again fixes this problem. There are some web application filter mitigations, but "mitigation" is really not what you want in this instance and I'd be very nervous relying upon blocking all incoming attempts, since this entire mode of CGI operation is inherently very poor and very unsafe design.

The bigger concern is that last June is not that long ago... at least not in the time frame of most upgrade cycles. So it would not be very surprising to find that many systems were still running earlier and vulnerable versions of PHP, and in vulnerable configurations. Apparently, this is that attackers are also discovering.

Listener Feedback

Sam Miorelli / @SamMiorelli

> *@SGgrc : Hey Steve, on the applications thing: I run an industrial cybersecurity business. Last year before we all knew about these things we got an applicant (who we hired to work in person) who was incredible on the CV (lots of certs, including for Fortigate) and video interview. We foolishly ignored warning signs when the in person manager met him post-offer and pre-start and things seemed off. After he started it was immediately clear the CV didn't reflect his actual skills. E.g. googling how to apply firewall rules on modern GUI firewall admin interfaces…*
>
> *When I endorsed hiring him, I chalked up his strange conversation style during the video interview to be from his accent/cultural as he's from India. (And he had all the right answers and wow what a great CV!) In hindsight I'm convinced he was using an AI interview helper tool like @finalround_ai. Of course it's impossible to prove these things, so we're having to think harder about how we screen applicants in the future. Lots of phonies out there, not just the North Koreans!*

Ian Beckett / @ianbeckett

> *@SGgrc : Re: SN1012: Microsoft Sysinternals tools. These tools are so popular, it's astonishing Microsoft's engineers don't securely recode these little tools. The little SyncToy tool (download now removed from Microsoft's Sysinternals site) still provides just about the only way to simply do a regular Windows sync backup to external drives using a TRUSTED tool. The pitiful inbuilt Windows 11 backup tool's only purpose is seemingly to drive revenue to onedrive subscriptions. I really despair of Microsoft nowadays … unless it generates online services revenue they have little interest in user experience.*

Ian is, of course, referring to the DLL Injection vulnerabilities that were recently discovered to adversely impact the security of the SysInternals tools. Rather than loading the standard system DLLs from the system's well known directories, the tools have retained Windows' deliberate though extremely insecure design of first looking in the executable's own execution directory before looking elsewhere. This allows bad guys to drop their own malicious versions of these DLLs, perhaps even older versions of Microsoft's own signed Windows DLLs that contain long since patched vulnerabilities, allowing them to effectively turn back the clock to be exploited again. Microsoft reportedly said "tough beans", we're not planning to fix them, which seems irresponsible. And as we noted at the time, even if they were fixed there's a massive inventory of them already deployed in the world. And they never receive updates.

@TycoonTom

> *@SGgrc : Hi Steve, what's that networking App that shows you net traffic 🛜the company was from Australia?🤔*

NetWorx from SoftPerfect: https://www.softperfect.com/products/networx/ Free for 30 days, after which I would be surprised if you didn't want it forever for $15. It will easily monitor the local machine. But my favorite feature is that from a local machine it's able to monitor the real time usage of the entire network by watching the router's SNMP interface byte counters.

John David Hickin

> *I'm not sure if it even deserves a CVE. This may well be similar to the case of the WIN32 API (and it's DLL) vs. the (at least, at one time, undocumented) API of NT.dll. These ESP32 undocumented commands may not be guaranteed to survive the next chip re-design. Device driver writers beware! Cheers, John.*

John's of course talking about last week's "Backdoor" that wasn't a backdoor. As we said, they were some undocumented functions in the SOC – System On a Chip – hardware. He's 100% correct that no one should be relying upon them since, being unofficial and undocumented, the Chinese chip maker Espressif should feel free to change their function or remove them entirely at any time. And I also agree that assigning this a CVE was ridiculous, though I understand the discover's motivation behind doing so. They were advertising this as a big bad backdoor, which was the narrative that most of the tech press picked up. So, of course, ya gotta have a CVE for that!

Mark Goldstein

> *Thanks for sharing Roger Grimes' story on the North Korean hackers. You did an important public service. The recitation of the story was funny and compelling podcasting. I told Roger of your recitation.*
>
> *In 2009, I wrote a business plan for my company, America Online, to acquire LastPass. The CEO said we were not in the security business so my proposal was shut down (although one day I visited Joe and his team with dozens of ice cream sandwiches on a hot Washington, DC day).*
>
> *After the first breach at LastPass I searched for a new password manager. I read what cryptologists said. I read FAQs and everything on various password manager websites. Finally, I found that 1Password had written some technical papers including their security model. It explained their various security choices. I could not evaluate all the crypto but I understand their perspective of the vulnerabilities of password managers. I discovered that they knew users of 1Password could create easy-to-crack master passwords so they used the master password along with a strong certificate to create the security for each instance of the password manager on a PC, Mac, iPhone, etc. When I create a new instance of 1Password, it copies the strong certificate to the new device. If someone cracks my 16-character password, they still must crack the 64-bit certificate. Good luck.*
>
> *This is why I choose 1Password. Subsequently I use 1Password on my iPhone and Windows PC. Their cross-platform implementation of passkeys works great for me. Passkeys on 1Password is my security solution.   Regards, Mark*

Thanks for sharing your note and your experiences, Mark. And many of us agree that 1Password (a TWiT network sponsor) is doing a terrific job.

I should note that I've also always been a fan of 1Password's additional user-account entropy which 1Password introduces with a client-side blob. While it means that it must be duplicated across all of a user's devices, that's a one-time requirement that then creates and provides very strong additional enduring security forever. It makes sense to me.

-Anonymous-

> *Steve, Please keep my name confidential. I would like to explain to you what happened to lastpass a few years ago. I work for a major cloud distributor and this occurred during a meeting with their CTO at the time, since Lastpass was one of our vendors.*
>
> *I asked what happened and the CTO explained that the Dev at home was using Plesk on his personal Mac which was hacked due to a Plesk media server that had not been updated.*
>
> *But the primary issue was that he was logged into the LastPass network from his personal machine. I asked the CTO why was he able to log into LastPass's network from in personal machine since they had policies in place to prevent that. The CTO confirmed that they did not enforce their own in policies. Also the secret AWS keys where they stored their customer vaults was kept in Lastpass Corporate Secure Notes, so readily accessible to anyone.*
>
> *So your evaluation of the product wasn't wrong, it's a good password manager. But the Compgny itself is not well managed. Regards,*

There was a bit of additional insight there that we haven't had previously. Since we cannot know how and where crucial decisions were being made, there's really no way to assign specific blame. But one thing we do know is that LastPass really dropped the ball on the PBKDF iterations issue. And there's really no excuse for that. They just didn't care. We know that because once this was brought to the glaring attention of the industry they went to the trouble of autonomously updating everyone's iteration counts. This proves that they could have done so at any time but never bothered to.

As we know, I always draw a sharp distinction between policy and mistakes. The LastPass developer whose machine was doubtless targeted and compromised was not practicing good security hygiene. And LastPass was not managing the connections into their corporate network. So the developer made a bad mistake. But not bothering to ever retroactively update original or older PDKDF iteration counts was a policy or priority decision made somewhere, and that's unforgivable.

Jeff

> *Steve; Mandiant is reporting on an espionage campaign by China, exploiting Juniper big-iron routers.*
>
> https://cloud.google.com/blog/topics/threat-intelligence/china-nexus-espionage-targets-juniper-routers
>
> *"End of life hardware and software." Yeah, that's a thing I see all the time. (You don't want to know what I found on the network of my Fortune-500 defense employer last week...) It's a bit*

> *of a dog-bites-man story, but it's part of a pattern by China to infiltrate critical infrastructure and hold it at risk as part of their national strategy. / Jeff*
>
> *PS -- Ha! I forgot to use my GRC-registered email. I appreciate the instant bounce, since I could fix that and re-send in less than 2 minutes.*

Since Jeff referred to his Fortune-500 Defense contractor employer, I left off his last name, though it's familiar to me since he's been an avid provider of feedback through the years.

I was familiar with the news he linked to. Older Juniper routers have problems that have been resolved in later devices. And those older routers are no longer receiving updates. So they're stuck with older firmware that will never be repaired. Still, those routers are well built and running ... so it's difficult for any CIO to tell his CFO that we need some money (a bunch of money) to replace some aging network infrastructure equipment. The CFO replies *"Okay. What's wrong with it? Isn't it still working?"* and our responsible CIO explains *"Well, yeah... but it's old and it's no longer being maintained by its manufacturer. So it could have some security weaknesses that could possibly be remotely exploited by foreign hostiles."* And the CFO says *"So you're saying that as far as you know there's nothing wrong with it, and it's still working just fine. But there **might or might not** be something wrong with it and we wouldn't know?"* And our CIO, feeling that he's losing this one says: *"Yes, that's exactly right. We could be in danger."* And the CIO ends the discussion saying: *"I get what you're saying. I really do. But we have very very pressing needs, and they are not what ifs, they're real. It only makes sense for those to take priority."*

I don't know how that changes. Certainly every one of the C-suite executives appreciates the need for proactive security. That CFO would not blink at the need for an industrial strength firewall appliance to keep the bad guys out. And I'm sure that intellectually everyone also appreciates the need for security patches and updates. Everything around them is constantly being updated and patched and fixed – their phones and PC and now even their automobiles. And we're all being told that these measures keep problems from ever occurring. But we never actually see any of these supposed problems. So it's rather intangible and difficult to sell.

It feels like this is going to require a cultural change. And while I intensely dislike the "rental model" that the world is moving toward, in the case of keeping older gear secure there's real value being offered.

Where I believe that, for example, Juniper, has missed a trick is in choosing to allow their appliance to fall out of maintenance and to not tie its continued operation to an annual paid maintenance agreement. They're leaving money on the table by not keeping their older devices alive and maintained, in return for some cash. The very many companies with older and still working Juniper gear are not upgrading to newer devices because the older devices their customers already have are still working. But those customers DO truly need security maintenance for those devices and they would probably pay for it. Why abandon a customer and their ongoing need for security? Makes no sense to me.

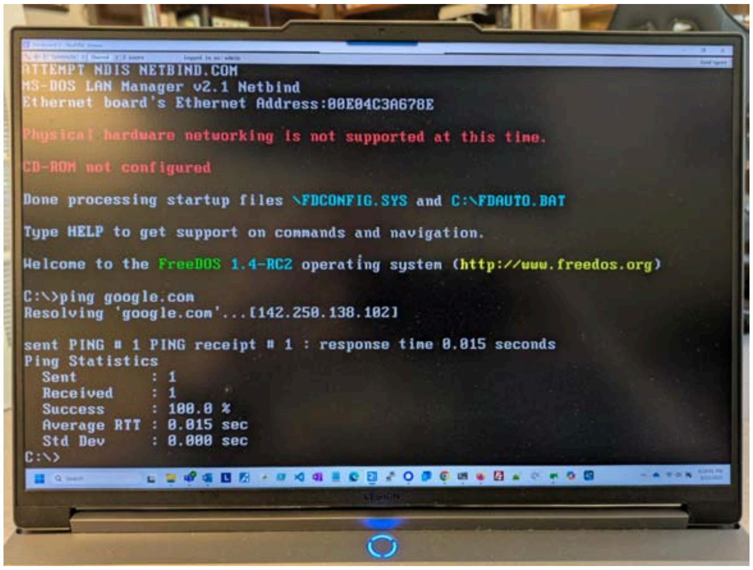
Bruce Olson

> *I wanted to make sure you knew about this claim being made by users on Reddit. It seems the organization behind ZimaBoards may be selling user information as some folks have started receiving marketing targeted at e-mail accounts given to ZimaBoard. Thanks for all the great work and always looking forward to the next episode! Bruce from Michigan*

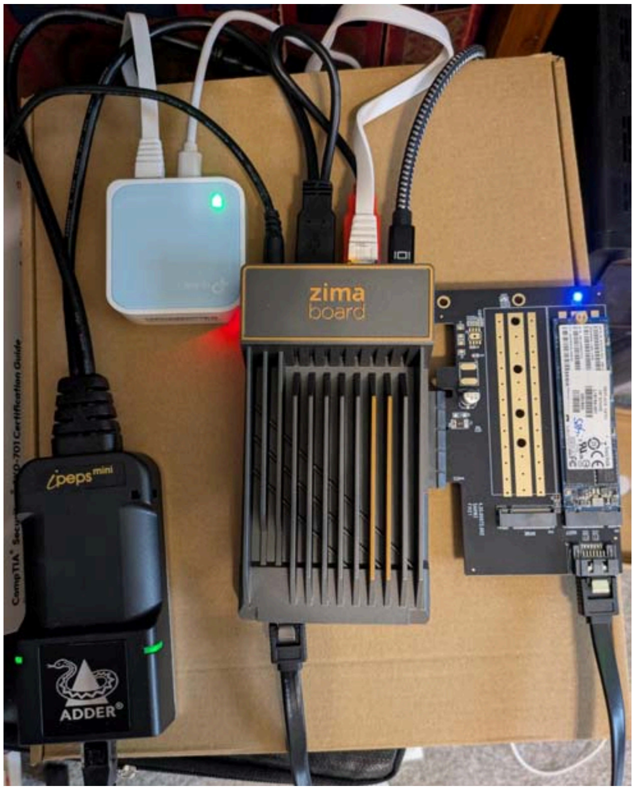
Well, that's disappointing. It's certainly a reason for using an eMail aliasing service so that this abuse can be controlled by the email recipient. And in the case of IceWhale, the ZimaBoard creators, I can't say I'm surprised. I receive a great deal of promotional email with all manner of special offers and come-ons. I just went over to their site and the top of the page has a bright orange scrolling banner saying "Sign up now and unlock up to $50 for new members." This argues for purchasing the boards through Amazon. But I suppose I would chalk that up to the cost of obtaining a perfect little single board PC with 2 network interfaces, 2 SATA ports, a PCIe expansion slot and Linux preloaded all for $90 dollars. It's still the best deal around, even if one does need to give them a throwaway email address.

Bill Allen / Subject: Loving my Zimaboard!
I promise that I did not plan this. As I was moving through my email feedback bag, the very next note that popped up after Bruce's note about IceWhale selling our contact data was this note from Bill Allen with the subject *"Loving my Zimaboard!"* Bill wrote:

Steve, I got started with a Zimaboard specifically to run Spinrite more easily on hard drives in my office, which it does very, very well. But it has turned into a bit of an obsession, and a really fun project platform! Here is my Zimaboard system:



```
ATTEMPT NDIS NETBIND.COM
MS-DOS LAN Manager v2.1 Netbind
Ethernet board's Ethernet Address:00E04C3A678E

Physical hardware networking is not supported at this time.

CD-ROM not configured

Done processing startup files \FDCONFIG.SYS and C:\FDAUTO.BAT

Type HELP to get support on commands and navigation.

Welcome to the FreeDOS 1.4-RC2 operating system (http://www.freedos.org)

C:\>ping google.com
Resolving 'google.com'...[142.250.138.102]

sent PING # 1 PING receipt # 1 : response time 0.015 seconds
Ping Statistics
  Sent        : 1
  Received    : 1
  Success     : 100.0 %
  Average RTT : 0.015 sec
  Std Dev     : 0.000 sec
C:\>
```



To its right is an outboard PCIe card carrier for the NVMe M.2 drive it is booting from. Upper left is a mini travel wireless router in client mode. Down and to the left is an ADDERLink IP KVM which is giving me keyboard, mouse, and video access to it across my local network via its internal VNC server. Currently running FreeDOS (as shown in the other photo). That Freedos install also has Spinrite 6.1 on it, of course. Thanks for pointing us to the Zimaboard! Best Regards, Bill, in Crowley, TX

I've received many similar reports through the years since my discovery of this lovely little device. It's not super powerful. I always purchased the smallest of the three available models since it was just going to be running FreeDOS which can be powered by a squirrel cage. But these little boards are the machines that built and tested SpinRite. I managed to get it on my local network and communicating with my Windows machines. With SpinRite v6.1 finished and SpinRite moving to Windows, I doubt I'll have an ongoing need... but I have fond memories.

Mark Jones
This one has some detailed lead-up, but I loved his story, which is a bit of a head shaker. The subject of his email feedback was *"AI and Microsoft Defender"* ... Mark wrote:

> *Dear Steve, Love the show, loyal listener since episode 1, and Club TWIT member. I really appreciate you and Leo.*
>
> *I encountered something new that illuminates some of the comments you've made recently about AI. I volunteer with an organization that has websites and a newsletter. About half our membership is employed by one of two big multinationals. Both are Microsoft shops. Both have lots of barbed wire wrapping their IT infrastructure. Microsoft Defender blocks questionable sites. The sieve is set pretty tight. At one point when I was still working there GRC.com got blocked.*

Just to insert a note here: For many years I was hosting known viral code for research purposes. The page containing the various archives was very clearly marked and everything was very RED and flashing. But all any search engine or trawling bot sees is ZIP archives containing known dangerous files. Since there is no longer much interest in that I removed them long ago.

> *I moved 25 years worth of our organization's newsletters to its own site 3 years ago. The site is only three PHP files, some XML for SEO, and a bunch of PDFs. I made the move after consultation with IT folks at the company I used to work for prior to retiring. They indicated that simpler was better at keeping out of the crosshairs of security suites. Sites that allow visitors to uploading files are particularly troubling to the corporate IT folks and our main site, over my protests, has WordPress plugins that accept uploads.*
>
> *Just recently, the site, midlandchemist.org, started being blocked by the corporate Microsoft protection. I went to an IT friend and asked how I could fix it. After 3 years of being OK, the site was suddenly being blocked. He was kind and connected me with someone responsible for the blocking. Here is where AI comes in.* [Get a load of this!] *The filters, are now AI-based, not rules based. He could not tell me why the site was being blocked because there was no rule being tripped. Something about the site triggered the AI algorithms. No reason could be given. It was just AI.*
>
> *Just as you described, AI makes connections that may elude human interpretation. The good news is that there is a way to whitelist sites provided I can find an employee willing to take responsibility. Regards, Mark*

Wow. You gotta love that one. *"We turned all site blocking over to AI, so it just does whatever it does. We no longer know how or what."* Welcome to the future, where we still don't have flying cars. (And thank god for that!)

PV

> *Steve, I was recently casting a line out into the sea of kindle unlimited suggestions. Unfortunately, I also ran into the "Artifact" book before you talked about it, but I also found a winner. The series is called "dumb luck and dead heroes" by Skyler Ramirez. It starts out a bit rough in the first book (both main characters are at a VERY low point in their lives and there's a lot of wallowing in that), but it picks up really fast, and there's a lot of crazy fun space adventure and just the right amount of humor.*

Besides the main books, he has a lot of little side stories that are the (strange but true) details behind one of Brad's stories, and there's also 3 books about his "best friend who's also a king's cross assassin" which are a bit different in tone, but fun as well. I generally am not a fan of side stories, but I enjoyed all of these. To 1100 and beyond, PV

I appreciate and I'm forwarding "PV's" recommendation without any of my own review.

While we're on the topic of Sci-Fi reading, for my part I am remaining ever-more-deeply hooked on Neal Asher's novels. I'm now into the 3rd of the first 5-novel "Agent Cormac" series and toward the end of the 2nd one I realized that I was really having a good time. I'm super-finicky about the quality of the writing, and these are fully satisfying in that regard. And he is building up some truly interesting characters.

It's still pulp. I'm not meaning to suggest otherwise. And it's not free. Unlike "PV's" discovery of those "Dumb Luck and Dead Heroes" novels which are available through Amazon's Kindle Unlimited, these Neal Asher novels are $7 each. But with a 5-shot Starbucks Latte now at $9.50, I'm easily obtaining way more than $7 worth of entertainment from each one. And given how much Asher has written, and the comments online that they get better and better, I'm going to be stuck reading everything he's written for a while.

And lastly, before we get to today's main question of just how susceptible any of the PC-compatible machines you have may be to RowHammer attacks, while I'm reviewing Sci-Fi stuff there's something Lorrie and I watched and enjoyed immensely Friday evening:

SciFi / Action

"The Gorge"

If someone who knew I had a subscription to Apple TV and that I enjoyed Science Fiction themes, recommended "The Gorge" to me, having just watched it Friday night, I would have been appreciative of their recommendation. So, having seen and enjoyed the movie immensely, I'm hereby making that recommendation to our listeners.

As the movie unfolded it had all the promise of being what I call "A perfect movie." They are rare, and this one was not as it turned out. But it certainly started out that way. At about 1/3rd of the way through it I said to my wife "So far, this is a perfect movie." And by that I don't mean that it's ever going to win any awards. But as the plot unfolded the movie was perfectly paced and in no hurry to get where it was going. Necessary facts were revealed as needed while much of the bigger story about what was going remained a mystery. And something clearly was going on. So the viewer was kept wondering and the journey was very enjoyable. Lorrie and I recognized the female protagonist as the actor who played the chess prodigy in "The Queen's Gambit" – another movie we both loved.

Now, I should say that later in the movie things became somewhat far-fetched, ridiculous and looking a bit like they were trying to create a video game tie-in. I could easily watch the entire front of the movie again, it was so well done, right up until they descended into the gorge.

No longer being 14, I'm not a fan of implausibly ridiculous over-the-top violence. But I would still strongly recommend the movie to anyone who has an Apple TV subscription and hasn't yet watched it. It was a lot of fun.

Is YOUR System Vulnerable to RowHammer?

It's rare that we're able to invite the listeners of this podcast to actively participate, themselves, in cutting edge security research. But this week a research team that has been looking into and questioning the actual dangers presented by RowHammer attacks is asking for as much breadth and depth of real world participation from the field as they can obtain. This amounts to downloading an .ISO file, writing it to a thumb drive, then booting and running the Arch Linux OS and RowHammer data gathering tests that it contains.

I immediately downloaded the 1 gigabyte .ISO file, used the latest RUFUS v4.6 for Windows to transfer that .ISO to a 32 gigabyte thumb drive, booted it on my ZimaBoard and let it run in the background while I worked on the podcast. But let's back up a bit...

We've been talking about the many various aspects and versions of the original discovery known as "RowHammer" since its first description in 2014. The essence of the problem is that in the inevitable quest to increase the density of main system dynamic RAM, you know, the RAM that's typically measured in tens of gigabytes, engineers squeezed every last bit of noise margin out of their designs. The RAM still worked. Systems booted and ran reliably. But then some clever researchers came along and asked a question no one else had before. They asked: "What if we were to hammer over and over and over on one row of RAM or on the RAM on either side of one row? Might that confuse any nearby bits?"

And we know the answer to that question. It turned out that, yes indeed, not only can neighboring bits be affected, but those effects can be powerfully weaponized to completely collapse and bypass the security boundaries and guarantees upon which all modern computing relies for its operational security.

During the decade that followed, these surprisingly prevalent and successful attacks have been elaborated upon and expanded by many groups of researchers across the globe. The attacks have been strengthened, optimized and sped up. Researchers have even demonstrated web-based exploitation via JavaScript code and network packets. And after the industry reacted to the initial news of these exploitable weaknesses with improved designs, these creative researchers even bypassed those enhanced protections.

Nearly four years ago, in May of 2021, Google's security blog posting was "Introducing Half-Double: New hammering technique for DRAM Rowhammer bug". Google's summary of their discovery is worth a review since it nicely lays out today's situation. They wrote:

> *Today, we are sharing details around our discovery of Half-Double, a new Rowhammer technique that capitalizes on the worsening physics of some of the newer DRAM chips to alter the contents of memory.*
>
> *Rowhammer is a DRAM vulnerability whereby repeated accesses to one address can tamper with the data stored at other addresses. Much like speculative execution vulnerabilities in CPUs, Rowhammer is a breach of the security guarantees made by the underlying hardware.*
>
> *As an electrical coupling phenomenon within the silicon itself, Rowhammer allows the potential bypass of hardware and software memory protection policies. This can allow untrusted code to break out of its sandbox and take full control of the system.*

> *Rowhammer was first discussed in a paper in 2014 for what was then the mainstream generation of DRAM: DDR3. The following year, Google's Project Zero released a working privilege-escalation exploit. In response, DRAM manufacturers implemented proprietary logic inside their chips that attempted to track frequently accessed addresses and reactively mitigate when necessary.*
>
> *As DDR4 became widely adopted, it appeared as though Rowhammer had faded away thanks in part to these built-in defense mechanisms. However, in 2020, the TRRespass paper showed how to reverse-engineer and neutralize the defense by distributing accesses, demonstrating that Rowhammer techniques are still viable. Earlier this year, the SMASH research went one step further and demonstrated exploitation from JavaScript, without invoking cache-management primitives or system calls.*
>
> *Traditionally, Rowhammer was understood to operate at a distance of one row: when a DRAM row is accessed repeatedly (the "aggressor"), bit flips were found only in the two adjacent rows (the "victims"). However, with Half-Double, we have observed Rowhammer effects propagating to rows beyond adjacent neighbors, albeit at a reduced strength. Given three consecutive rows A, B, and C, we were able to attack C by directing a very large number of accesses to A, along with just a handful (~dozens) to B. Based on our experiments, accesses to B have a non-linear gating effect, in which they appear to "transport" the Rowhammer effect of A onto C.*
>
> *Unlike TRRespass, which exploits the blind spots of manufacturer-dependent defenses, Half-Double is an intrinsic property of the underlying silicon substrate. This is likely an indication that the electrical coupling responsible for Rowhammer is a property of distance, effectively becoming stronger and longer-ranged as cell geometries continue to shrink. Distances greater than two are conceivable.*
>
> *Google has been working with JEDEC, an independent semiconductor engineering trade organization, along with other industry partners, in search of possible solutions for the Rowhammer phenomenon. JEDEC has published two documents about DRAM and system-level mitigation techniques (JEP 300-1 and JEP301-1).*
>
> *We are disclosing this work because we believe that it significantly advances the understanding of the Rowhammer phenomenon, and that it will help both researchers and industry partners to work together, to develop lasting solutions. The challenge is substantial and the ramifications are industry-wide. We encourage all stakeholders (server, client, mobile, automotive, IoT) to join the effort to develop a practical and effective solution that benefits all of our users.*

Everyone is worried about the possibility of what this would mean, but despite all the academic work that's been done, there have never been any reports of actual Rowhammer attacks in the wild. This is reminiscent of "Spectre" and "Meltdown". But it might also be more relevant to the Y2K worry where, despite the fact that the world didn't end, that may have largely been due to so much work going into making sure it wouldn't end. But in the case of all of the various RowHammer attacks, questions have been raised about the attack's true feasibility in real-world scenarios.

This brings us to the December 2024 presentation at Germany's 38th Chaos Communication Congress during which a trio of academics observed that the actual practical impact of these various RAM hammering attacks remains unknown and is still largely theoretical. They noted that past academic research used small, even relatively microscopic, sample sizes.

*The density of memory cells in modern DRAM is so high that disturbance errors, like the Rowhammer effect, have become quite frequent. An attacker can exploit Rowhammer to flip bits in **inaccessible** memory locations by reading the contents of nearby **accessible** memory rows. Since its discovery in 2014, we have seen a cat-and-mouse security game with a continuous stream of new attacks and new defenses. Now, in 2024, exactly 10 years after Rowhammer was discovered, it is time to look back and reflect on the progress we have made and give an outlook on the future. Additionally, we will present an open-source framework to determine whether **your** system is vulnerable to Rowhammer.*

In 2014, researchers reported a new disturbance effect in modern DRAM that they called Rowhammer. The Rowhammer effect flips bits in inaccessible memory locations just by reading the content of nearby memory locations that are attacker-accessible. They trigger the Rowhammer effect by accessing memory locations at a high frequency, using memory accesses and flushes. The root problem behind Rowhammer is the continuous increase in cell density in modern DRAM. In early 2015, Seaborn and Dullien were the first to demonstrate the security impact of this new disturbance effect. In two different exploit variants, they demonstrated privilege escalation from the Google Chrome NaCl sandbox to native code execution and from unprivileged native code execution to kernel privileges. Later, in 2015, Gruss et al. demonstrated that this effect can even be triggered from JavaScript, which they presented in their talk "Rowhammer.js: Root privileges for web apps?"

Now, in 2024, it is precisely 10 years after Rowhammer was discovered. Thus, we believe it is time to look back and reflect on the progress we have made. We have seen a seemingly endless cat-and-mouse security game with a constant stream of new attacks and new defenses. We will discuss the milestone works throughout the last 10 years, including various mitigations (making certain instructions illegal, ECC, doubled-refresh rate, pTRR, TRR) and how they have been bypassed.

We show that new Rowhammer attacks pushed the boundaries further with each defense and challenge. While initial attacks required native code on Intel x86 with DDR3 memory, subsequent attacks have also been demonstrated on DDR4 and, more recently, DDR5. Attacks have also been demonstrated on mobile Arm processors and AMD x86 desktop processors. Furthermore, instead of native code, attacks from sandboxed JavaScript or even remote attacks via network have been demonstrated as well. Furthermore, we will discuss how the Rowhammer effect can be used to leak memory directly, as well as related effects such as Rowpress. We will discuss these research results and show how they are connected. We will then talk about the lessons learned and derive areas around the Rowhammer effect that have not received sufficient attention yet. We will outline what the future of DRAM disturbance effects may look like, covering more recent effects and trends in computer systems and DRAM technology.

*Finally, an important aspect of our talk is that we invite everyone to contribute to solving one of the biggest unanswered questions about Rowhammer: What is the real-world prevalence of the Rowhammer effect? How many systems, in their current configurations, are vulnerable to Rowhammer? As large-scale studies with hundreds to thousands of systems are not easy to perform, such a study has not yet been performed. Therefore, we developed a new framework to check if **your** system is vulnerable to Rowhammer, incorporating the state-of-the-art Rowhammer techniques and tools.*

Thus, we invite everyone to participate in this unique opportunity at the 38th Chaos Communication Congress, to join forces and close this research gap together.

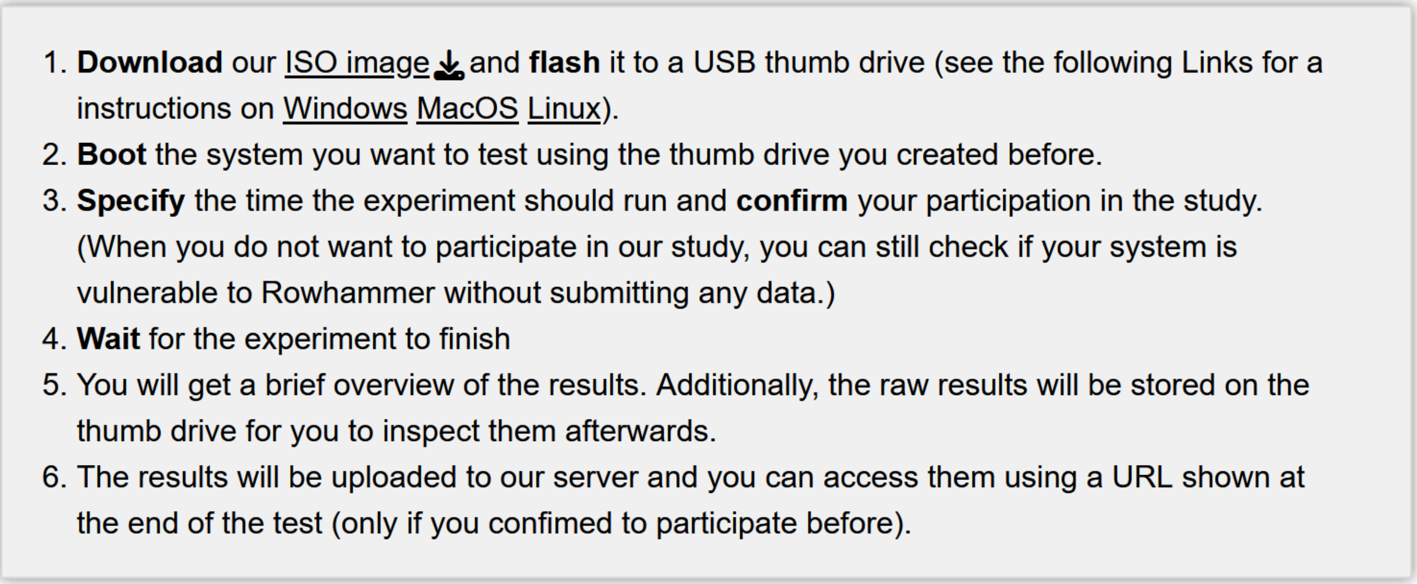
https://flippyr.am/

The website where all this lives, and where everyone who's interested should go to grab their copy of the open source test code is: https://flippyr.am/

Welcome to our FLIPPYR.AM Study. We want to analyze the prevalence of Rowhammer in real-world systems. Everybody can participate in our study. The entire source code is open-source and available via GitHub. You can either build the ISO yourself or run the entire study using Docker. However, we highly recommend using the ISO image:

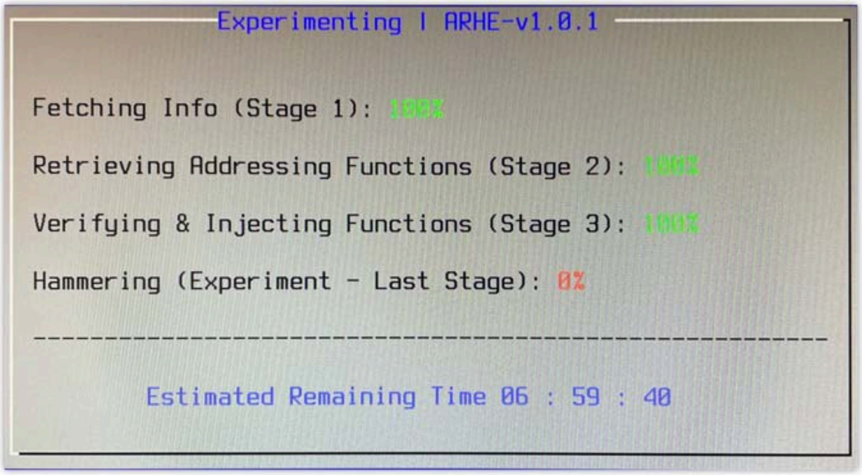
https://flippyr.am/hammeriso.iso

Simply follow these steps:



1. **Download** our ISO image ⤓ and **flash** it to a USB thumb drive (see the following Links for a instructions on Windows MacOS Linux).
2. **Boot** the system you want to test using the thumb drive you created before.
3. **Specify** the time the experiment should run and **confirm** your participation in the study. (When you do not want to participate in our study, you can still check if your system is vulnerable to Rowhammer without submitting any data.)
4. **Wait** for the experiment to finish
5. You will get a brief overview of the results. Additionally, the raw results will be stored on the thumb drive for you to inspect them afterwards.
6. The results will be uploaded to our server and you can access them using a URL shown at the end of the test (only if you confimed to participate before).

Okay. So the test defaults to running for 8 hours and it requires some patience. Unfortunately, there's nothing cool like a running total of RowHammer strikes, so no results are available until the test has completed. The testing moves through four stages, and on my ZimaBoard the first hour was spent getting ready to start the hammering. So some patience will be required:



```
                    Experimenting | ARHE-v1.0.1

Fetching Info (Stage 1): 100%

Retrieving Addressing Functions (Stage 2): 100%

Verifying & Injecting Functions (Stage 3): 100%

Hammering (Experiment - Last Stage): 0%

--------------------------------------------------------

          Estimated Remaining Time 06 : 59 : 40
```

The screen contains a time-remaining down-counter which updates once per second as it counts down. The entire screen has a blue background and on the ZimaBoard the once-per-second update causes the entire screen to flash. I have some next-generation not-yet-deployed new server hardware where I'll also be running this test as well.

Once the test finishes a complete report is written to the drive under a unique filename so that multiple runs will never overwrite previous results. And assuming that permission is then given, that file will be uploaded to the Mothership while a short summary of the machine's result will be shown on the screen. It's also possible to NOT have the results autonomously uploaded, but to instead examine the file that's left behind on the thumb drive, then return to the Flippy RAM site and perform the same upload yourself.

For what it's worth, I'll be uploading all of my results and would hope that others would as well since we would all like to benefit from the results of a large-scale RowHammer sensitivity and vulnerability test.

And, finally, though I doubt many of us would require much incentive to do this, these guys want to express their thanks and they really do want to see what's out there in the world. So they have an "Incentives" section which says:

> *As an incentive, the following two rewards can be won:*
>
> *When you upload a valid dataset, you will receive a cryptographic token. This token is generated by hashing random data, and when you upload your dataset, we will save this token separately in our database. This means the token is not associated with your dataset. This ensures that you can participate in the raffle without linking the token to your dataset. Please make sure to bookmark or save this token.*
>
> - *The first people to send us 10 valid tokens via e-mail (flippy underscore ram at hof minus university dot de) will receive a free flippyr.am t-shirt. We have 10 t-shirts to give away. First come, first served!*
>
> - *Everyone who sends us an e-mail with a valid token will participate in a raffle and have a chance to win a €10 Amazon gift card. The more tokens you send us, the higher your chances are.*

Two releases of the tool have been made so far, v1.0 and 1.0.1 and the SHA256 hashes of both of versions' .ISO files are provided on the site for anyone who wishes to manually verify that they have downloaded the real deal. I'll share the results I find and I'll be glad to receive and anonymously share during a future podcast any results summaries our listeners may wish to share.

At the bottom of the show notes I also have a link to the researcher's Chaos Communication Congress presentation in both video and audio and with multiple language tracks:
https://media.ccc.de/v/38c3-ten-years-of-rowhammer-a-retrospect-and-path-to-the-future#t=1453