Security Now! #992 - 09-17-24

Password Manager Injection Attacks

This week on Security Now!

What happened during Microsoft's recent Windows Endpoint Security Ecosystem Summit? And what, if anything, will probably result? How reliable is ANY form of digital storage when used for long-term archiving? What happened when an illegal Starlink Internet network was set up on a U.S. Navy ship? What's the best solution for securing the Internet-facing "edge" of enterprise networks? GRC has started notifying SpinRite 6 owners about 6.1. What's been learned about the challenge of sending email in 2024? Why might running SpinRite on an SSD cause the SSD to then appear to be running more slowly? Why is true secrecy so difficult to achieve, and how were most password managers leaking some of their secrets.

Not only does this truck have dual HDMI outputs, but great signal strength and is fully charged!

Security News

Windows Endpoint Security Ecosystem Summit

We recently noted that Microsoft was responding, as we (and they) knew they must, to what's now being referred to as the "Cloudflare Outage". Though the proximate cause of this global meltdown was a bad update to Cloudflare's kernel-level code, which forced a Windows' kernel panic and shut-down, the fact that 3rd-party vendors are (a) allowed to install their code into Windows' kernel and that (b) Windows lacked graceful resiliency that would have allowed it to arrange to get back on its feet using some form of rollback, meant that Microsoft received some measure of blowback themselves. Thus was held, last Tuesday's so-called "Windows Endpoint Security Ecosystem Summit." On Sunday, Microsoft posted about what happened, who attended and some of what was said, writing:

On Tuesday, Sept. 10, we hosted the Windows Endpoint Security Ecosystem Summit. This forum brought together a diverse group of endpoint security vendors and government officials from the U.S. and Europe to discuss strategies for improving resiliency and protecting our mutual customers' critical infrastructure. Although this was not a decision-making meeting, we believe in the importance of transparency and community engagement. Therefore, we're sharing the key themes and consensus points discussed during the summit, offering insights into our initial conversations.

We want to thank every one of our summit attendees for dedicating their time to participating in these meaningful discussions. The CrowdStrike incident in July underscored the responsibility security vendors have to drive both resiliency and agile, adaptive protection. And it was inspiring to see the engagement throughout the event's agenda and activities.

Together with our Microsoft Virus Initiative (MVI) partners—companies who develop endpoint protection and additional security products for Windows, covering client, server and IoT—we discussed the complexities of the modern security landscape, acknowledging there are no simple solutions.

A key consensus point at the summit was that our endpoint security vendors and our mutual customers benefit when there are options for Windows and choices in security products. It was apparent that, given the vast number of endpoint products on the market, we all share a responsibility to enhance resiliency by openly sharing information about how our products function, handle updates and manage disruptions.

In the short term, we discussed several opportunities to improve how we support the safety and resiliency of our mutual customers. First, we spent time going into depth on how we employ Safe Deployment Practices (SDP) at Microsoft and where we can create shared best practices as a community, including sharing data, tools and documented processes. We face a common set of challenges in safely rolling out updates to the large Windows ecosystem, from deciding how to do measured rollouts with a diverse set of endpoints to being able to pause or rollback if needed. A core SDP principle is gradual and staged deployment of updates sent to customers. Microsoft Defender for Endpoint publishes SDPs and many of our ecosystem partners such as Broadcom, Sophos and Trend Micro have shared how they approach SDPs as well. This rich discussion at the Summit will continue as a collaborative effort with our MVI partners to create a shared set of best practices that we will use as an ecosystem going forward.

Reading between those lines, CrowdStrike was being – I believe that the proper term is "bitch slapped" – for not evidencing **any** sort of staggered roll-out of their updates. This suggests a level of "can do no wrong " arrogance that did indeed come back to bite them. And the lack of any is impossible to justify in retrospect. There's no good answer to the question "why the heck weren't you deploying in any sort of staggered roll-out fashion?" Anyway, Microsoft continues:

Beyond the critical Safe Deployment Practices work, there are several ways we can enhance our support for customers in the near term. Building on the Microsoft Virus Initiative program we have today, we discussed how Microsoft and partners can increase testing of critical components, improve joint compatibility testing across diverse configurations, drive better information sharing on in-development and in-market product health, and increase incident response effectiveness with tighter coordination and recovery procedures. These are a sampling of the topics we plan to make rapid progress on, to improve our collective customers' security and resiliency.

In addition, our summit dialogue looked at longer-term steps serving resilience and security goals. Here, our conversation explored new platform capabilities Microsoft plans to make available in Windows, building on the security investments we have made in Windows 11. Windows 11's improved security posture and security defaults enable the platform to provide more security capabilities to solution providers outside of kernel mode.

Both our customers and ecosystem partners have called on Microsoft to provide additional security capabilities outside of kernel mode which, along with SDP, can be used to create highly available security solutions. At the summit, Microsoft and partners discussed the requirements and key challenges in creating a new platform which can meet the needs of security vendors.

Some of the areas discussed include:

- *Performance needs and challenges outside of kernel mode*
- *Anti-tampering protection for security products*
- *Security sensor requirements*
- *Development and collaboration principles between Microsoft and the ecosystem*
- *Secure-by-design goals for future platform*

As a next step, Microsoft will continue to design and develop this new platform capability with input and collaboration from ecosystem partners to achieve the goal of enhanced reliability without sacrificing security.

Finally, there are important steps customers can take today to increase resiliency in their current deployments. In addition to the important conversations summarized above, there are several practical, vendor-neutral steps enterprises can benefit from, including having business continuity planning (BCP) and a major incident response plan (MIRP) in place and backing up data securely and often.

It was clear from kickoff through closing at the summit that as platform and endpoint security providers, we are all focused on the productive conversations that need to be happening. ***We're competitors, we're not adversaries.*** *The adversaries are the ones we need to protect the world from. We are grateful for the support and input from this community and excited about the conversations in progress and work we have ahead.*

Vendors participating in the Windows Endpoint Security Ecosystem Summit offered remarks with further perspective:

Adam Bromwich, Broadcom's *CTO and Head of R&D for their Enterprise Security Group: "Organizations today benefit from a diverse, layered security defense. As a result, industry collaboration is vital to helping organizations stay ahead of persistent threats and remain resilient when unexpected business disruptions occur. As a long-time Microsoft Virus Initiative (MVI) Partner, Broadcom recognizes that working closely with Microsoft and other security vendors not only helps improve our customers' security posture, including endpoint protection, but also the greater global digital ecosystem."*

CrowdStrike's Drew Bagley*, VP & Counsel for Privacy and Cyber Policy: "We appreciated the opportunity to join these important discussions with Microsoft and industry peers on how best to collaborate in building a more resilient and open Windows endpoint security ecosystem that strengthens security for our mutual customers."*

ESET provided the statement: *"ESET supports modifications to the Windows ecosystem that demonstrate measurable improvements to stability, on condition that any change must not weaken security, affect performance, or limit the choice of cybersecurity solutions. It remains imperative that kernel access remains an option for use by cybersecurity products to allow continued innovation and the ability to detect and block future cyberthreats. We look forward to the continued collaboration on this important initiative."*

SentinelOne's Chief Product and Technology Officer, Ric Smith: *"SentinelOne thanks Microsoft for its leadership in convening the Windows Endpoint Security Ecosystem Summit and we are fully committed to helping drive its goal of reducing the chance of future events like the one caused by CrowdStrike. We believe that transparency is critical and strongly agree with Microsoft that security companies must live up to stringent engineering, testing and deployment standards and follow software development and deployment best practices. We are proud that we have followed the processes that Microsoft has discussed today for years and will continue to do so going forward."*

Sophos, CEO, Joe Levy: *"We are honored to be a part of the Windows Endpoint Security Ecosystem Summit. It was a welcome opportunity to join industry peers in an open discussion of advancements that will serve our customers by elevating the resilience and robustness of both Microsoft Windows and the endpoint security ecosystem. We were very pleased to see Microsoft support many of Sophos' recommendations, based on the collection of architectural and process innovations we've built over the years and present today on the 30 million Windows endpoints we protect globally. The summit was an important and encouraging first step in a journey that will produce incremental improvement over time, and we look forward to collaborating in the design and delivery of more resilient and secure outcomes to our customers."*

Trellix, Karan Sondhi, CTO *for the Public Sector: "Responsible security starts with vendor's architecture, coordination with the ecosystem and prioritization of resilience for all. The time for collaboration across our industry and government to stay ahead of our adversaries is now."*

Kevin Simzer, Chief Operating Officer for Trend Micro: *"I applaud Microsoft for opening its doors to continue collaborating with leading endpoint security leaders, to make our mutual customers even more cyber resilient. Looking forward to more collaboration."*

My feeling is that this was mostly for show. The photo that accompanied this blog posting was a perfect representation for this meeting. It shows a bunch of executives of various stripes sitting around a conference table in a stunningly opulent office building conference room setting. There's the requisite white board, a UI projected onto another screen and four world-time digital clocks visible for London, Moscow, Beijing and Sydney. Interestingly, the one laptop we can see clearly is distinctively a Mac.

My point is, nothing ever really gets accomplished at such meetings. This was all just for show, for the government and for Microsoft's shareholders. What's actually going to happen now will be a long, multi-year series of slow, plodding back and forth negotiations where Microsoft will present and may implement a next-generation set of userland hooks for use by their various 3rd-party vendors. The vendors will examine them and explain how what Microsoft is offering still doesn't give them the total freedom they really want, and can argue they need, which is only still available through true kernel-level operation. And so it will go, back and forth. Perhaps something will eventually come of it, but that's far from certain. So long as this meeting has been held and the parties are all now <quote> working on it together </quote>, face has been saved, lawsuits will trundle forward and the vendors will all work harder not to make another similar horrible mistake. My only hope was that since Mark Russinovich tweeted about this, if he's involved, I have somewhat more hope that something might actually change.

We know that in retrospect CrowdStrike now realizes it needs to be able to catch anything like this before it is rolled out to the entire world. That's trivial to do and we know they're going to be doing it. And this also means that everyone else ***must*** do the same and never fail at this trivial-to-implement requirement. No more cowboy developer jock behavior. The stakes are now far too high.

Aging storage media does NOT last forever

A listener forwarded this piece from ArsTechnica to me, titled: *"Music industry's 1990s hard drives, like all HDDs, are dying"* and the subhead is: *"The music industry traded tape for hard drives and got a hard-earned lesson."*

One of the things enterprise storage and destruction company Iron Mountain does is handle the archiving of the media industry's vaults. What it has been seeing lately should be a wake-up call: roughly one-fifth of the hard disk drives dating to the 1990s it was sent are entirely unreadable.

Music industry publication Mix spoke with the people in charge of backing up the entertainment industry. The resulting tale is part explainer on how music is so complicated to archive now, part warning about everyone's data stored on spinning disks.

Robert Koszela, global director for studio growth and strategic initiatives at Iron Mountain, told Mix: "In our line of work, if we discover an inherent problem with a format, it makes sense to let everybody know. It may sound like a sales pitch, but it's not; it's a call for action."

Hard drives gained popularity over spooled magnetic tape as digital audio workstations, mixing and editing software, and the perceived downsides of tape, including deterioration from substrate separation and fire. But hard drives present their own archival problems. Standard hard drives were also not designed for long-term archival use. You can almost never decouple the magnetic disks from the reading hardware inside, so that if either fails, the whole drive dies.

There are also general computer storage issues, including the separation of samples and finished tracks, or proprietary file formats requiring archival versions of software. Still, Iron Mountain tells Mix that "If the disk platters spin and aren't damaged," it can access the content. But "if it spins" is becoming a big question mark. Musicians and studios now digging into their archives to remaster tracks often find that drives, even when stored at industry-standard temperature and humidity, have failed in some way, with no partial recovery option available.

Koszela says: "It's so sad to see a project come into the studio, a hard drive in a brand-new case with the wrapper and the tags from wherever they bought it still in there. Next to it is a case with the safety drive in it. Everything's in order. And both of them are bricks."

Mix's passing along of Iron Mountain's warning hit Hacker News earlier this week, which spurred other tales of faith in the wrong formats. The gist of it: You cannot trust any medium, so you copy important things over and over, into fresh storage. Optical media rots, magnetic media rots and loses magnetic charge, bearings seize, flash storage loses charge, etc. Entropy wins, and sometimes much faster than you'd expect.

There is discussion of how SSDs are not archival at all; how floppy disk quality varied greatly between the 1980s, 1990s, and 2000s; how Linear Tape-Open, a format specifically designed for long-term tape storage, loses compatibility over successive generations; how the binder sleeves we put our CD-Rs and DVD-Rs in have allowed them to bend too much and stop being readable.

Knowing that hard drives will eventually fail is nothing new. Ars wrote about the five stages of hard drive death, including denial, back in 2005. Last year, backup company Backblaze shared

failure data on specific drives, showing that drives that fail tend to fail within three years, that no drive was totally exempt, and that time does, generally, wear down all drives. Google's server drive data showed in 2007 that HDD failure was mostly unpredictable, and that temperatures were not really the deciding factor.

So Iron Mountain's admonition to music companies is yet another warning about something we've already heard. But it's always good to get some new data about just how fragile a good archive really is.

This is only indirectly related to SpinRite, however there's good reason to believe that performing a periodic rewrite of either magnetic spinning or electrostatic solid-state mass media is an extremely useful thing to do. You don't need SpinRite to do that, though SpinRite makes that easy, and it provides a great deal of feedback about the state of the drive. (And if any sort of trouble arises during the rewrite, SpinRite is what you want to have in charge.)

In any event, we learned a few years ago that offline SSDs tended to lose their data more rapidly when stored at a higher temperature. Since SSD storage loss is about charge leakage, that makes perfect sense since higher temperature tends to weaken the strength of the dielectric insulation that isolates the charge bits.

How Navy chiefs conspired to get themselves illegal warship Wi-Fi
https://www.navytimes.com/news/your-navy/2024/09/03/how-navy-chiefs-conspired-to-get-themselves-illegal-warship-wi-fi/
The "Navy Times" recently blew the lid off an intriguing story of a U.S. Navy warship installing a secret Starlink network on board so that a select few of the upper echelon would not be deprived of their precious Internet connectivity while deployed. In there piece headlined "How Navy chiefs conspired to get themselves illegal warship Wi-Fi" the Navy Times wrote:

Today's Navy sailors are likely familiar with the jarring loss of internet connectivity that can come with a ship's deployment. For a variety of reasons, including operational security, a crew's internet access is regularly restricted while underway, to preserve bandwidth for the mission and to keep their ship safe from nefarious online attacks. But the senior enlisted leaders among the combat ship Manchester's gold crew knew no such privation last year, when they installed and secretly used their very own Wi-Fi network during a deployment, according to a scathing internal investigation obtained by Navy Times.

As the ship prepared for a West Pacific deployment in April 2023, the enlisted leader onboard conspired with the ship's chiefs to install the secret, unauthorized network aboard the ship, for use exclusively by them. So while rank-and-file sailors lived without the level of internet connectivity they enjoyed ashore, the chiefs installed a Starlink satellite internet dish on the top of the ship and used a Wi-Fi network they dubbed "STINKY" to check sports scores, text home and stream movies.

The enjoyment of those wireless creature comforts by enlisted leaders aboard the ship carried serious repercussions for the security of the ship and its crew. The investigation noted: "The danger such systems pose to the crew, the ship and the Navy cannot be understated."

Led by the senior enlisted leader of the ship's gold crew, then-Command Senior Chief Grisel Marrero, the effort roped in the entire chiefs mess by the time it was uncovered a few months later. Marrero was relieved in late 2023 after repeatedly misleading and lying to her ship's command about the Wi-Fi network, and she was convicted at court-martial this spring in connection to the scheme. She was sentenced to a reduction in rank to E-7 after the trial and did not respond to requests for comment for this report. The Navy has yet to release the entirety of the Manchester investigation file to Navy Times, including supplemental enclosures. Such records generally include statements or interview transcripts with the accused.

But records released so far show the probe, which wrapped in November, found that the entire chiefs mess knew about the secret system, and those who didn't buy into it were nonetheless culpable for not reporting the misconduct. Those chiefs and senior chiefs who used, paid for, helped hide or knew about the system were given administrative nonjudicial punishment at commodore's mast, according to the investigation. All told, more than 15 Manchester chiefs were in cahoots with Marrero to purchase, install and use the Starlink system aboard the ship.

The investigation said: "This agreement was a criminal conspiracy, supported by the overt act of bringing the purchased Starlink onboard USS MANCHESTER. Any new member of the CPO Mess which then paid into the services joined that conspiracy following the system's operational status."

Records obtained by Navy Times via a Freedom of Information Act request reveal a months-long effort by Marrero to obtain, install and then conceal the chiefs Wi-Fi network from superiors, including the covert installation of a Starlink satellite dish on the outside of the Manchester. When superiors became suspicious about the existence of the network and confronted her about it, Marrero failed to come clean on multiple occasions and provided falsified documents to further mislead Manchester's commanding officer, the investigation states. Unauthorized Wi-Fi systems like the one Marrero set up are a massive no-no for a deployed Navy ship, and Marrero's crime occurred as the ship was deploying to the West Pacific, where such security concerns become even more paramount among heightened tensions with the Chinese.

While Marrero claimed the Wi-Fi system was secretly installed for morale purposes, the investigation notes that such a claim "is undermined by the selective availability of the Wi-Fi and strict control of its access to the CPO Mess only."

The Manchester's secret Wi-Fi network was born in March 2023, when Marrero and a co-conspirator got to work buying and installing the Starlink system before the ship's deployment began the following month. The Starlink dish was installed on the Manchster's O-5 level weatherdeck during a "blanket" aloft period, which requires a sailor to hang high above or over the side of the ship. During a "blanket" aloft, duties are not documented in the deck logs or the officer of the deck logs, according to the investigation. It's unclear who harnessed up and actually installed the system for Marrero due to redactions in the publicly released copy of the probe, but records show Marrero powered up the system the night before the ship got underway to the West Pacific waters of U.S. 7th Fleet.

Marrero and her cohorts paid $2,800 for a Starlink High Performance Kit with a personal credit card, and contacted Starlink to expedite shipping so the system would arrive in time for the deployment. Starlink offers plans ranging from $90 to $5,000 a month, and allows users to control network settings via a cell phone app. The Navy is installing such authorized capabilities aboard some ships in the fleet. But that was not the case aboard Manchester, where Marrero set up payment plans for the chief's mess to pay for the system — either

$62.50 a month or a one-time fee of $375 — that the ship's Chief Petty Officer Association treasurer collected into a chiefs mess checking account.

Those involved also used the Chief Petty Officer Association's debit card to pay off the $1,000 monthly Starlink bill, and Marrero warned the chiefs to only use the network in their rooms. Marrero served as the gatekeeper of the system, records show, downloading and maintaining the Starlink app from her phone and naming it "STINKY." Only she could add others to the network, and would directly type the password into their devices, according to the investigation. After Manchester got underway from San Diego, Marrero and the chiefs soon realized the Wi-Fi signal didn't cover all areas of the ship, so the senior chief purchased signal repeaters and cable at the Navy Exchange store in Pearl Harbor, Hawaii, during a port visit in late April or early May, according to the investigation.

Little stays secret within the close quarters of a deployed ship, and shortly after getting underway, scuttlebutt started swirling among some sailors about the unauthorized Wi-Fi network, the investigation states. The ship's former executive officer, Cmdr. Matthew Yokeley, caught wind of the rumors in May and notified the commanding officer, Cmdr. Colleen Moore. Moore confronted Marrero about whether the chief's mess had an unauthorized Wi-Fi network that same month. Another unidentified crew member approached Marrero about a Wi-Fi network aboard the ship after finding available networks on a device that started with the name "STINKY." It's unclear who found the "STINKY" network, due to redactions.

In both instances, Marrero denied that such a Wi-Fi network existed. But she soon changed the "STINKY" Wi-Fi network name to another moniker that looked like a wireless printer — even though no such general-use wireless printers were present on the ship, the investigation found.

That's about 1/3rd of what the Navy Times printed, but that's sufficient to give everyone a good sense for what went down. With so many people seemingly unable to separate themselves from the Internet, it's foreseeable that there would be significant pressure to maintain connectivity while at sea. But at the same time we know how true it is that any form of Internet connection would need to be tightly filtered. Can you imagine how much any hostile foreign power would love to get into the smartphone or laptop of someone onboard who clicks the wrong link, to then pivot with that access and jump into the ship's internal networks?

adam:ONE named the #1 best Secure Access Service Edge (SASE) solution

A few months ago I shared how impressed I was after meeting with and learning about the technology that the guys at Adam Networks (adamnet.works) had created to help secure the Internet-facing border of enterprises. As I've noted several times, that's a daunting task and it's not a job I would want. So while I was delighted to see it, I was not surprised to discover that SC Magazine, a well known and reputable security industry publisher, after running a head-to-head competitive comparison and evaluation of the industry's many various solutions, picked Adam:ONE as the winner. SC Magazine wrote:

Adamnetworks has claimed the prestigious Best SASE Solution award at the 2024 SC Awards for its cutting-edge product, adam:ONE. In a cybersecurity landscape where traditional reactive methods often fall short, adam:ONE stands out by providing a proactive, zero-trust security solution designed to eliminate threats before they infiltrate networks. This recognition places Adamnetworks among the leading innovators in the increasingly competitive Secure Access Service Edge (SASE) market.

Since protecting the enterprise from all of the mischief that those inside the enterprise might get up to is no small task, and since the Adam:ONE folks appears to have the best handle on doing that job, I have the link to SC Magazine's announcement with many more details in the show notes. And it was episode #946, on Halloween (October 31st) of 2023 that I shared the results from my meeting with the Adam:ONE folks. I have a link to those show notes, too for anyone who wants a refresher in light of this best in class award: https://www.grc.com/sn/sn-946-notes.pdf https://www.scmagazine.com/news/sc-award-winners-2024-adamnetworks-best-sase-solution

Miscellany

Many of our listeners may have received email about the availability of SpinRite v6.1. Of course that won't come as news to anyone listening, but for anyone who purchased SpinRite 6.0, which was released 20 year ago in 2004, it would likely come as a welcome surprise. Over the weekend I receive a note about this from a listener name Patrick, he wrote:

G'morning sir. Quick note to let you know I've received an email from spinrite.news, but it was flagged as spam by Exchange and dumped in my Junk folder. Otherwise, thanks for the work on Spinrite 6.1 – I'll let you get back to work on 7.0 now. /Patrick

I replied to Patrick, writing:

Thanks for your feedback, Patrick. Since I'm mailing to all past SpinRite owners for the past 20 years, I'm sending those announcements through that domain you noted "SpinRite.News". Since that domain has not earned a reputation as a valid email sender, APPLE is bouncing =all= incoming email addressed to anyone @me.com, @icloud.com and @mac.com. And, as you note, Exchange is routing incoming mail to spam (but at least it's not bouncing the mail back.)

My primary goal for this is 2-fold: I **do** want to inform any non-podcast listeners of the availability of a free upgrade to v6.1, and I also want to remove all bad email addresses from the list for the future. This is why I'm sending from the previously unused domain "SpinRite.News" since I expect that the bounce rate will be high for the oldest 20-year-old mail addresses and I want to keep grc.com's email reputation as spotless as possible. Once I've managed to update GRC's creaky old SpinRite owner list, I'll be able to mail from grc.com, using its clean email reputation, and mail should then go into people's inboxes. :)

So… if any of our listeners happen to find a SpinRite upgrade announcement in any of their spam folders, marking it as "**not spam**" would be very much appreciated since it would inform your ISP that this was not unsolicited and unwanted commercial email, which would increase the chances of others actually seeing the message.

SpinRite

Speaking of SpinRite, the free upgrade email that's going out makes a point of noting that perhaps the most significant thing we learned during SpinRite's three and a half years of development was that SSD and presumably flash storage slows down dramatically over time. We believe that this is due to the phenomenon known as "Read Disturb" which any Internet search will reveal is a very real thing and a huge concern for solid state memory makers.

It turns out that in a similar fashion to how dynamic RAM memory is vulnerable to so-called Row Hammer attacks, due to the reading of adjacent memory regions, the electrostatic charges which are stored in non-volatile flash cells may also be disturbed by reading of neighboring data. Rewriting these regions would fix them, but the usage patterns of installed operating systems is to never rewrite most of the system's files or much of the file system's metadata.

Since this is behavior that has, until now, been flying under the radar, the email I've been sending to SpinRite 6 owners makes the point that SpinRite is not only for spinning magnetic mass storage drives.

In response to receiving this, to downloading his updated copy of 6.1 and running in, an owner named Michael sent a short three-line note:

> *Pre-level 3: Front: 131, Mid: 184, End: 185*
> *Post level 3: F, M, E: 120*
> *??*

So what Michael communicated was that running SpinRite over his SSD appeared to have slowed the entire thing down from a high of 185 megabytes per second to just 120 megabytes per second. So I replied to Michael by writing:

> *It appears that your SSD was mostly empty. So what happened is that those "pre" SpinRite benchmark readings were illusory and were not really returning results from reading from the drive's physical media. SSD's and spinning SMR (Shingled Magnetic Recording) drives are aware of whether anything has ever been written to individual regions of their media. If nothing has never been written, then there's nothing to be read. So they don't bother actually reading anything since nothing other than blank space – all 0's or all 1's – could possibly be there. They just return all 0's or 1's at lightning speed – at the full speed of the interface that connects the drive to its computer.*
>
> *But when SpinRite re-wrote the SSD's entire surface, the drive now believes that ALL of the media is now "in use" even though it may still only be storing all 0's or 1's. Now the drive believes **that** data is important to its owner. So when SpinRite's benchmark is run afterwards, what will be shown is the true reading speed from the media, which was exactly 120 megabytes per second everywhere.*
>
> *After running SpinRite and remounting the SSD in an operating system, the OS will "re-trim" the SSD. It runs through the entire SSD's "region in-use" table, marking all of the regions that*

are not actually storing any file system data as "empty". All current operating systems do this periodically, so this will cure itself. Under Windows this happens weekly, though you can run the Windows "Disk Optimizer" on the drive to cause Windows to do this on demand. And if you then re-run SpinRite's benchmark on the SSD you'll see results that can be compared with the "pre-SpinRite" benchmark result you had before.

I wanted to share this with everyone here since this is a common puzzlement. If an SSD is already mostly full, its surface will have been untrimmed from its storage of real data. But on mostly empty SSD's, the results can be confusing.

My favorite reaction to the weekend's mailing was:

Jim

Wow – talking about turning back time to the days of Madonna and a portable CD player that took 'D' cells! Love it. I bet if I dig long enough, I might be able to find a copy on floppy of SpinRite from early 1990 or so. Awesome. Thanks–you saved my tail many times in the past!

Closing the Loop

Chris Paetz

Just listened to your podcast and heard you say that you won't be able to find your email address anywhere online.... But perplexity.ai knows where to find it apparently FYI. I asked the question "What is Steve Gibson's email address at Security Now" and received the reply:

"Steve Gibson's email address for the Security Now podcast is securitynow@grc.com. This address is intended for podcast feedback and is mentioned regularly during the show, although it is not prominently displayed on the Gibson Research Corporation website"

I'll just say that I continue to be surprised by what we're creating. As we humans continue to push technology further and further, we create both new capabilities and new dilemmas. We're rapidly moving into a world that was only recently pure science fiction. It's going to be truly interesting to see what happens with AI.

Password Manager Injection Attacks

Today's exploration topic began with my receipt, a week ago, of a note which read:

> *Hi Steve, My name is Ben. I am a former Ben-Gurion University student and long-time listener of Security Now. You covered a few studies of mine in the past including: Lamphone (speech recovery from bulbs), Video-based Cryptanalysis (key recovery from a power LED), and Morris-II (the AI worm).*
>
> *In two recent studies that I co-authored and were led by Andres from Cornell Tech, we revealed attacks against E2EE applications, demonstrating the recovery of encrypted confidential data from backups of two messaging applications (WhatsApp and Signal) and ten password managers (LastPass, DashLine, Zoho Vault, 1Password, Enpass, Roboform, Keeper, NordPass, Proton Pass, and KeePassXC). We named these attacks "injection attacks" and the papers were published on USENIX Sec'24 and Security and Privacy'24.*
>
> *Attached are the links I believe that your audience will find interesting, as once again they prove that while E2EE is the best approach for applications, the devil is in the implementation.*

I had seen from Ben's note envelope that his full name was **Ben Nassi**, which I recognized immediately since we've covered a lot of his work and exploitation discoveries through the years as he noted. So, needless to say, I was interested in what new mischief Ben and his fellow security researchers had gotten themselves up to now.

Five or six researchers led by Cornell University's Andres Fabrega, from Cornell University and Cornell Tech, collaborated on two papers. One was titles *"Injection Attacks Against End-to-End Encrypted Applications"* and the other was titled *"Exploiting Leakage in Password Managers via Injection Attacks"*. To get us started on our understanding of what they have done, I'm going to share the Abstract from the paper about Password Manager injection attacks. It says:

> *This work explores injection attacks against password managers. In this setting, the adversary (only) controls their own application client, which they use to "inject" chosen payloads to a victim's client via, for example, sharing credentials with them. The injections are interleaved with adversarial observations of some form of protected state (such as encrypted vault exports or the network traffic received by the application servers), from this, the adversary is able to obtain confidential information.*
>
> *We uncover a series of general design patterns in popular password managers that lead to vulnerabilities allowing an adversary to efficiently recover passwords, URLs, usernames, and attachments. We develop general attack templates to exploit these design patterns and experimentally showcase their practical efficacy via analysis of ten distinct password manager applications. We disclosed our findings to these vendors, many of which deployed mitigations.*

Okay. So that's interesting. When they use the term "injection" they're referring to providing an unwitting target some information that the target will cause to be stored in their own instance of a password manager. And then by examining the target's encrypted vault or their network traffic presumably synchronizing their encrypted vault with the password manager's cloud backup,

they're able to learn as much as the user's secret username, passwords, URLs and attachments. We need to learn more about that.

But first I'll note that the second paper does something somewhat similar with end-to-end messaging, specifically WhatsApp and Signal. In the case of messaging, an attacker sends messages to a targeted user. Assuming that the attacker is somehow able to obtain and observe the targeted user's encrypted cloud backup, the researchers were able to demonstrate their ability to determine whether the target had received specific attachments or, for example, which of two messages the target had previously received.

Clever as these researchers are, I came away feeling better rather than worse about the safety of WhatsApp and Signal. My feeling was "Wow, if that's the most intrusion that these guys were able to achieve given their obviously serious skillz, then that says a lot about how good these apps are. But that said, since the goal is true zero leakage of any kind, I would not be surprised to learn that the app vendors had added some length fuzzing of things being stored specifically to thwart the leakage that these guys were able to induce. So the research was definitely useful.

But the password manager leakages, from what we've seen so far, appear to be somewhat more dire. So let's take a closer look there. They set the stage in their introduction, writing:

> *Password-based authentication suffers from well-known pitfalls, such as the fact that users tend to choose passwords that can be easily guessed by attackers. Password managers are often cited as the default solution to this problem, as users can offload to them the complexities of password generation, storage, and retrieval. Indeed, password managers have enjoyed a noticeable rise in popularity, placing them among the most ubiquitous security-oriented tools.*
>
> *Password managers have benefited from academic attention, which has helped understand and improve their security along various dimensions. The attacks uncovered by prior work broadly fall under two general threat models. First are attacks that use a client-side resource controlled by the adversary, such as a malicious website visited by the client, a rogue application in the victim's device, or the client's WiFi network. Second are adversaries that somehow acquire a copy of a user's encrypted vault, and exploit leakage from unencrypted vault metadata or by offline cracking attacks of a user's master password. State-of-the-art password managers are therefore designed to resist both kinds of threats and, notably, use slow cryptographic hashing to prevent cracking attacks for well-chosen master passwords.*
>
> *In this work, we consider a new kind of threat model in which an adversary (1) controls their own application client, through which they can send chosen payloads to the victim (for example, via the password sharing feature found in most modern password managers); and (2) can observe some form of encrypted state and associated metadata, such as the user's encrypted vault backups or network requests received by the application servers. Borrowing terminology from prior work in other domains, we refer to attacks in this threat model as injection attacks.*
>
> *The core idea behind injection attacks is that the adversary can use injections to trigger subtle interactions in the application logic between their data and target victim data (e.g., other passwords used by the target), which are reflected in their observations of ciphertexts (e.g., inspecting their lengths) and metadata in a way that allows recovering sensitive information.*

We argue that this threat model is increasingly important as password managers become more complex and feature-rich, which provides new avenues for injection mechanisms and vulnerable cross-user interactions.

To understand whether this threat model is of practical concern or not, we performed a security analysis of ten popular password managers that support sharing: LastPass, Dashlane, Zoho Vault, 1Password, Enpass, Roboform, Keeper, NordPass, Proton Pass, and KeePassXC. Together these reportedly account for over 30% of all password manager users.

We uncover a series of exploitable vulnerabilities that implicate all of the password managers investigated.

Our first class of attacks exploits the fact that a common feature of password managers is for clients to periodically log outside the device various metrics about the "health" of a user's vault, such as the number of duplicate passwords. We show how an adversary can leverage these benign-looking metrics to perform an efficient binary-search-based dictionary attack that recovers the target user's saved passwords. Our attacks do not require the adversary to know additional information about the victim's saved credentials beforehand (for example, URLs nor usernames). Five out of the ten applications are vulnerable to this attack. In most cases, the adversary must be a passive eavesdropper that observes these metrics directly (for example, by having a persistent foothold in the application servers), while for one application the attack is feasible by a passive network adversary that simply observes the HTTPS channels under which the E2EE data is transmitted. We note that both eavesdropping and network adversaries are within scope of the threat models under which password managers are designed, and the ubiquity of server-side breaches, combined with the difficulty of detecting such breaches, make it critical that password managers resist such attacks.

Our second class of attacks exploits another feature of password managers: clients often display a small identifying icon, such as a company logo, alongside each of a user's saved credentials. Importantly, such icons are only fetched once per URL, and subsequent credentials reuse the icon stored in the client. We show how this fact allows an adversary to perform an efficient dictionary attack on the URLs in a victim's vault. The attack always succeeds in our experiments, and mounting it requires no additional assumptions about the victim's saved credentials. Six of our ten case study applications are vulnerable to this attack, and in all cases exploitation only requires observations by a network adversary.

We then turn our attention to adversaries that have an encrypted copy of the entire vault, such as compromising a local password-protected database file or backup of it. In this case, we analyze the security of KDBX, which is a file format used by many password managers, notably KeePass and its derivatives. To optimize for storage, KDBX employs a variety of storage-saving techniques, such as file deduplication and compression. We show two attacks exploiting these features to recover URLs, usernames, and attachment contents. Compression and deduplication have led to attacks against other systems before (see Section 2), but our work is the first to show that these types of vulnerabilities also arise in the context of password managers. Our attacks target features of the underlying file format itself, and thus can potentially be leveraged against any application that uses KDBX. We implement a proof-of-concept for our attacks in the case of KeePassXC, and experimentally show that its accuracy is sufficiently high to make it a practical threat.

A summary of our attacks follows. They exploit common design patterns found in password managers, and as such other applications that employ these can be vulnerable to our attacks. Indeed, for each of our attacks, we describe a general template for it, which is agnostic to

lower-level application details, and that can be used to target any application that follows the relevant design pattern. More broadly, our findings uncover higher-level issues in password manager design, and we discuss the future work that will be required to provide generally applicable mitigations for injection attacks.

I want to clarify the nature of these attacks so that the vulnerabilities they found will make more sense. They identified three ways of attacking the security of today's password managers:

The first class of attacks, which the researchers refer to as "Vault-Health Logging" rely upon the newer features of application-wide metrics. These arise from the behavior exhibited by many password managers which compute various metrics, like the number of passwords their user has duplicated in their vault which includes both personal and shared vault entries. When these metrics are logged outside the device, such as by the application's cloud servers, an adversary can induce fluctuations in these metrics with injections, and observe how they are updated in the external location. To carry out these attacks, in addition to having the ability to inject credentials, this attack requires the adversary to have access to the location where the metrics are logged. So a foothold of some kind in the application servers.

LastPass, Dashlane, Zoho Vault, Keeper and NordPass are all vulnerable to these vault-health logging attacks. All of the password managers except for Zoho Vault require that server-side foothold. That's also effective against Zoho Vault, but it's additionally vulnerable to simpler passive network but for them, an attacker must arrange access to the vendor's cloud servers. However, the vault-health attack can be used successfully against Zoho Vault with only passive network eavesdropping.

The second class of attacks is "URL icon fetching." It arises from the fact that many password managers display a graphical icon next to each credential, identifying the website associated with it. This icon will only be fetched once from the application servers, and future entries for the same website reuse the image from a client-side cache. An adversary can use this to determine whether or not a target's password manager has previously obtained and cached a credential for a particular URL. If the attacker induces the target's password manager to get an icon that means there was no previous entry for that website. In addition to having a means of updating the target's credentials, probably through credential sharing, this attack requires the attackers to be able to observe the HTTPS requests that leave the victim's client, or have a foothold in the server from which the icons are fetched.

The researchers found that Dashlane, 1Password, Enpass, Roboform, Proton Pass, and NordPass are all vulnerable to disclosing their protected URLs through this passive network eavesdropping.

The third and final class of attacks only affects KeePassXC (among the top ten password managers tested). This arises from KeePassXC's storage file system. KeePassXC uses this system to decrease the size of its encrypted vault. The problem is, the presence of data compression in secure storage – which is used to reduce storage redundancy – can be used to reveal the data that's already stored in the compressed store. The size of the storage will increase when unique new information is stored. But if something is added that already exists in the storage, the compression will keep the total storage from increasing nearly by as much.

It's a clever side-channel that we're seen and covered in the past here. It's necessary to be very careful about compressing secret data before it's encrypted because this side-channel can be used to leak information about the content of the secret data. We know that post-encryption compression is never used because it's not possible to compress encrypted information since anything that's been properly encrypted is indistinguishable from completely random noise and completely random noise is, by definition, incompressible.

KeePassXC is vulnerable to this type of attack because it does exactly that. By examining fluctuations in the size of KeePassXC's encrypted vault, after injecting known information it's possible for an attacker to glean information about the vault's unknown contents. To pull this off, in addition to injecting credentials this attack requires the adversary to have persistent access to the victim's encrypted vault, either directly or my monitoring vault backups. In their work, the researchers demonstrated attacks against two such mechanisms: database compression and attachment deduplication.

After putting these ten password managers through the wringer, they contacted each of the password manager's vendors to share their results ahead of taking this work public. Here's what the researchers reported from that effort. They wrote:

We reported our findings to the ten vendors affected by our work, many of which proceeded to deploy mitigations.

LastPass adopted our suggested mitigation of separating vault-health metrics between personal and shared credentials, which disables the injection channel. They released an initial implementation of this fix in version 4.129.0, removing shared folders from the vault-health logs, as these lead to the most severe variant of our attack. Removing individually shared credentials from the logs is more technically challenging—and individual credentials lead to a less practical version of our attack—and thus has been deferred to later in their roadmap; their projection is to release this fix by the end of the year, which would complete a full mitigation to our attack.

Zoho Vault plans to adopt a similar fix, by implementing an option to separately compute vault-health metrics on personal passwords as of version 4.0. Dashlane opted for a partial mitigation instead, namely, increasing their rate limits on the sharing endpoints <quote> "as much as possible" </quote>. Given the fact that their vault-health metrics are only logged once per day, their tight sharing limits significantly affect the practicality and runtime of the attack. In addition, the resource limits on their web application and extensions prevent an adversary from sharing an unlimited number of credentials with a victim, which increases the runtime of the attack even more. As part of the disclosure, they informed us that incorporating shared passwords is a core feature of their vault-health metrics, and thus removing shared passwords would represent a notable disruption to this feature.

Then, to address our URL icon fetching attack, Dashlane implemented a new feature as of version 6.2415 that allows users to turn off fetching credential icons, which disables the side-channel for both an eavesdropping and network adversary, and thus provides a full mitigation to our attack. To address our attack even when URL icons are turned on, they additionally migrated their icon fetching tool to a new endpoint (api.dashlane.com), which is used by multiple parts of their application logic. As such, this would make it significantly more challenging for a network adversary to use traffic analysis techniques to identify whether an icon fetch request is included in the traffic sent to this top-level endpoint, due to the high

amount of noise from the other requests sent to this endpoint.

NordPass also adopted our suggested mitigation of separating vault-health metrics between personal and shared credentials, which disables the injection channel and is thus a full mitigation to our attack. This fix is planned to be deployed by the end of August 2024. Then, to address our URL icon fetching attack, NordPass added a feature to disable URL icons by default, which provides a mechanism to disable the injection side-channel. This fix is planned to be deployed before the end of 2024. In addition, they are currently exploring more robust mitigations for this attack, to protect users even when URL icons are turned on.

To address our attack on attachment deduplication, KeePassXC adopted our suggested mitigation of deduplicating files separately for every shared folder, which disables the injection side channel. Then, to address our compression-based attacks, they modified their file format by, every time the database is saved, picking a random length between 64 and 512 bytes, generating a random array of this length, and including this in a "custom data" field of their file format. We note that this is only a partial mitigation, as an adversary can potentially use statistical techniques to bypass the noise; this, however, would require a significantly higher number of injections. Both fixes were promptly implemented by the KeePassXC team, and have since rolled out as part of version 2.7.

Enpass already provides support for turning off URL icons (which is off by default), and thus users who disable URL icons are not vulnerable to our attack. As a first step towards more mitigations, Enpass added an option for organization admins to control this setting via an organization level policy. They have decided not to deploy mitigations at this time to address the attack even when URL icons are turned on.

1Password already provides support for turning off URL icons, and thus users who disable URL icons are not vulnerable to our attack. However, 1Password decided not to deploy additional mitigations to address the attack at this time, even when URL icons are turned on. Similarly, Proton Pass already provides support for turning off URL icons, and thus users who disable URL icons are not vulnerable to our attack. We shared suggestions for how to address this attack even when URL icons are turned on, but do not have details on their plans to deploy these.

Lastly, Keeper considers our attack on their system a very low severity issue and opted not to deploy mitigations, and instead have added changing this feature as a consideration for an upcoming platform update. As part of the communication, they shared that removing transferred credentials from the count of duplicate passwords displayed in the Admin Console would represent a notable disruption to a feature of the business product.

Okay. So where does this leave us?

I agree with Ben Nassi that this is interesting and potentially important work. But, it's clearly of more interest theoretically than practically. As a true threat it's out there on the fringe. It's not impossible that some extremely motivated attacker might somehow arrange to set themselves up in a position to pull off one of these attacks against a high value target, but I'd say it's safe to say that none of us had anything to worry about even before these obscure holes were plugged.

So the reason I chose to share these attacks on this podcast is for what we learn from them about the true challenges that are associated with truly protecting secret information.
It's so easy for the salesman to boast "Oh, don't worry, it's all military-grade encrypted **and**,

guess what! We're using a bazillion-bit key!! So no one will ever possibly crack that!" Right.

But the lesson taught by these injection attacks is that no one ever needs to crack that bazillion-bit key. The reason the password managers jumped to modify and improve their systems when they were informed of these subtle issues is that "subtle issues" may be all that's needed to infer the data that's being protected by those bazillion-bit keys.

Any mature and fully informed understanding needs to appreciate that encrypting something is far from being the end of the task – it's only the start. Twenty years ago there was a general lack of understanding of the gulf that exists between theoretical and practical "field-ready" security technology. But during these past twenty years this sort of research has opened the eyes of the people who are implementing these systems, and we all benefit.

For anyone who may be interested in digging deeper, I've included the links to these two research papers at the end of the show notes.

Exploiting Leakage in Password Managers via Injection Attacks
https://arxiv.org/pdf/2408.07054v1

Injection Attacks Against End-to-End Encrypted Applications
https://www.cs.cornell.edu/~ragarwal/pubs/e2e-injection-attacks.pdf