Security Now! #945 - 10-24-23

The Power of Privilege

This week on Security Now!

How do fake drives keep being sold by Amazon? If you don't already know it, is VBScript worth learning today? NTLM authentication is 30 years old; will it see 40? What startling flaw was just found in cURL, and what should you do about it? Vulnerabilities with a CVSS score of 10.0 are blessedly rare, but today the industry has another. And also, asked by our listeners, how should "lib" be pronounced? How is SpinRite's 6.1 pre-release run? Is passkey export on the horizon? Doesn't a server's IP address make encrypting the client hello superfluous? Is there such a thing as encryption preemption? Are fraudulent higher-end drives possible? What's Privacy Badger and why did I just install it? And finally, within any enterprise, few things are more important than managing user and device access privileges. As highlighted by the NSA's and CISA's experiences, we're going to examine the need for taking privilege management more seriously than ever during this week's Security Now! Episode #945 — The Power of Privilege.

Not a well thought out plan.

(I'd LOVE to see what the worn dirt path looks like in a year.)

Fake drives

As many of our listeners know, ValiDrive’s homepage shows a photo of the twelve 1 & 2TB drives I purchased from Amazon last month. That page also includes links to the Amazon purchase listing for each of those 12 drives. With some surprise, many people have wondered how this fraud can exist on Amazon. So I wanted to share the experience of one of ValiDrive’s pre-release testers who took the trouble to purchase a drive then to post her review of that just purchased drive. Leila wrote:

> *I posted the following review of a cheap 512GB USB drive together with a screenshot of the ValiDrive results:*
>
> *"This USB stick is a fake. Although labeled and formatted to appear to have 512GB of storage, it in fact contains only 53GB. Anything written beyond 53GB appears to write without error but cannot be read back. The attached picture shows the result of testing with ValiDrive, a recently-published freeware program for testing USB-attached storage.*
>
> *If you buy a high capacity USB stick or microSD card for peanuts, don't be surprised if it's a fake. Check out the prices charged by reputable vendors, such as SanDisk and Kingston for reference.*
>
> *In any event, it makes sense to test all such cards/drives before putting them in your camera, dash cam, etc. Even those which appear to be from reputable manufacturers are sometimes forgeries."*

Not long after she received Amazon’s reply:

> *"Thank you for submitting a review of doykob Memory Stick 512 GB USB Stick Fast Speed USB 3.0 Drive Mini USB Metallic Flash Drive Portable USB Memory Sticks for Data Storage and Transfer, with Keychain (512gb); we’re sorry you did not have a positive experience. We investigated your concerns about product authenticity, and the information we have indicates that the product you received was authentic. As a result, we removed the review you submitted. This ensures that customer reviews remain as accurate as possible for the benefit of future customers."*

So that answers that question. Unwitting users who are taken in by this fraud post: *“Holy crap! I just bought this 2TB thumb drive for $20! I plugged it into my computer to verify its size and it looks great! This is a super bargain for 2TB so I’m definitely giving it five stars!”* And now we also know that authentic negative & cautionary product reviews are never seen by prospective customers. So you betcha these drives all have terrific reviews. How could they not?

Security News

So long, VBScript!
Microsoft’s VBScript (proprietary scripting language based upon Visual Basic) is officially headed for the dust bin of history, and not a moment too soon. Believe it or not, VBScript has been around since 1998 – so for 25 years! It was originally Microsoft’s attempt to combat JavaScript. They didn’t want someone else’s scripting language to prevail.

Since PowerShell is a far more comprehensive scripting language, and since Microsoft had already essentially abandoned it, with its last significant release being VBScript 5.8, in 2010, the only ones who are likely to mourn its passing are malware authors, among whom it has remained a popular tool. Though even there, not as popular as it once was.

That said, VBScript still has a large fanbase among system administrators who have an established code base that’s already working and has been proven. The good news for those who do still need it is that it’s not being completely removed from Windows. Instead it’s being moved from “you can assume it’s always there by default” to FoD or “Feature on Demand” where those comparatively few environments that depend upon it will still be able to install it. But since Microsoft definitely plans to remove it entirely at some point, anyone who does depend upon it would be well advised to migrate to PowerShell.

And as for malware, VBScript’s removal is expected to have some impact on malware in the sense that another piece of lowest hanging fruit will, at long last, have finally been pruned from Windows. But it’s not as if malware will be hugely inconvenienced. Just as when Microsoft dropped support for macros in Office applications, VBScript’s new FoD status will trigger a more concerned move to PowerShell-based malware exploitation. The larger gangs have all been using PowerShell for a while now, but the older “copy and paste” malware developer who have continued using VBScript because “why not?” will need to up their game.

Anyone who is listening to this podcast would be well advised to plan for the eventual disappearance of VBScript. Knowing Microsoft it will likely survive in the “Optional Features” category for quite some time yet. But probably not forever.

And while we’re on the topic of Microsoft deprecations...

NTLM Authentication Protocol to be discontinued
Microsoft has also announced their intention to eventually disable support for the ancient and never really secure NT (as in “New Technology”) Lan Manager authentication protocol in a future version of Windows 11. And labeling anything “NT” after 30 years in Internet Time is really a stretch, anyway!

The protocol was first introduced in 1993 – yes, 30 years ago – with the release of Windows NT 3.1. It served as the primary user authentication protocol until Windows 2000 which introduced the Kerberos authentication system originally developed at MIT (the Massachusetts Institute of Technology). Ever since then NTLM has still been there in Windows as a backup authentication protocol for legacy purposes and in situations where Kerberos couldn't be used.

The problem, as we well know, with keeping legacy protocols around is that the “weakest link” principle applies especially when it’s coupled with the “well, let’s just leave it enabled in case someone might need it” philosophy of network administration.

But, superior as Kerberos is, even today it’s not a slam dunk drop-in replacement for NTLM. Kerberos has knows problems where NTLM works, in offline/local use or (segmented) network topologies where there’s no direct line-of-sight to a Domain Controller. Microsoft has indicated that it’s very well aware of these issues and that it’s going to work to fix Kerberos' shortcomings before completely removing NTLM. Gotta love those 30 year old legacy protocols.

To that end, Microsoft says it’s working on two new Kerberos features named IAKerb and local KDC that will allow Kerberos to work in more diverse network topologies and in offline/local scenarios where there's no Domain Controller around. And all of this will only affect Windows 11, not Windows 10, because it’s probably still going to take a while and, believe it or not, Windows 10 now has fewer than two years of support left.

Windows 10’s support is slated to end on October 14, 2025. That’s 103 weeks from now. And that will be coinciding with Security Now! podcast #1046. So it’s a good thing we’re going to keep on going past #999 since I would not want to miss out on the opportunity to complain a lot about the premature ending of Windows 10 support!

In any event, to help enterprises to wind down their dependence upon NTLM, Microsoft will be adding new tools and logging capabilities to monitor NTLM usage along with providing finer-grained group policies to disable and block NTLM. Then, as NTLM usage goes down, Microsoft says that it fully intends to announce a hard date when the protocol will be disabled in Windows 11 and to finally pull the plug. Quoting Microsoft, they said: *“Reducing the use of NTLM will ultimately culminate in it being disabled in Windows 11. We are taking a data-driven approach and monitoring reductions in NTLM usage to determine when it will be safe to disable.”*

Two new cURL vulnerabilities

Vulnerabilities in the massively widely used cURL command and library are always a concern since the library is so often included into other projects. So it’s significant when Daniel Stenberg, the cURL project’s lead developer calls one of two just-discovered vulnerabilities “probably the worst Curl security flaw in a long time." The two flaws which were found by JFrog, impact libcurl versions v7.69.0 (where they were introduced) through v8.3.0. Daniel wrote:

We are cutting the release cycle short and will release curl 8.4.0 on October 11, including fixes for a severity HIGH CVE and one severity LOW. The one rated HIGH is probably the worst curl security flaw in a long time. The new version and details about the two CVEs will be published around 06:00 UTC on the release day.

CVE-2023-38545: severity HIGH (affects both libcurl and the curl tool)
CVE-2023-38546: severity LOW (affects libcurl only, not the tool)

There is no API nor ABI change in the coming curl release. I cannot disclose any information about which version range that is affected, as that would help identify the problem (area) with a very high accuracy so I cannot do that ahead of time. The “last several years” of versions is *as specific as I can get. We have notified the distros mailing list allowing the member distributions to prepare patches. Now you know. Plan accordingly.*

The worst of the two problems sounds quite bad and easy to exploit on its face but fortunately its required preconditions will prevent the world as we know it from ending. The cURL maintainers wrote in their follow-up advisory: *"The flaw is a heap-based buffer overflow in the SOCKS5 proxy handshake. When Curl is asked to pass along the hostname to the SOCKS5 proxy to allow that to resolve the address instead of it getting done by Curl itself, the maximum length that hostname can be is 255 bytes. If the hostname is detected to be longer than 255 bytes, Curl switches to local name resolution and instead passes the resolved address to the proxy. Due to a bug, the local variable that means 'let the host resolve the name' could get the wrong value during a slow SOCKS5 handshake, and contrary to the intention, copy the too long hostname to the target buffer instead of copying just the resolved address there."* So we have a timing-based race condition that could lead to remote code execution via a buffer overrun. The cURL devs said that the overflow could be triggered by a malicious HTTPS server performing a redirect to a specially crafted URL.

The JFrog guys who found the problem said: *"Seeing that Curl is a ubiquitous project, it can be assumed with good confidence that this vulnerability* ***will*** *get exploited in the wild for remote code execution, with more sophisticated exploits being developed. However, the set of pre-conditions needed for a machine to be vulnerable is more restrictive than initially believed."*

So this feels like another of those vulnerabilities that doesn't explode the Internet. Instead, it will be entered into those massive exploit databases I've theorized about in the past. Something like that must be maintained by our NSA and, unfortunately, by all other sufficiently sophisticated malign actors on the global stage. There's likely a huge dependency tree that knows which version ranges of every utility and library that uses cURL, was built using any of those vulnerable cURL libraries. So next year – or ten years from now – when we or some hostile foreign power wishes to penetrate a network, that database might serve to remind such an actor that *"hey, you know, the set of circumstances you're facing just happen to perfectly fit the use of a long ago patched, but still present in the target system, flaw that we can take advantage of today!"*

As I've often observed, given the Swiss cheese that is our industry's legacy of interdependent flawed libraries and slow moving updates there's just no way that such capability does not exist.

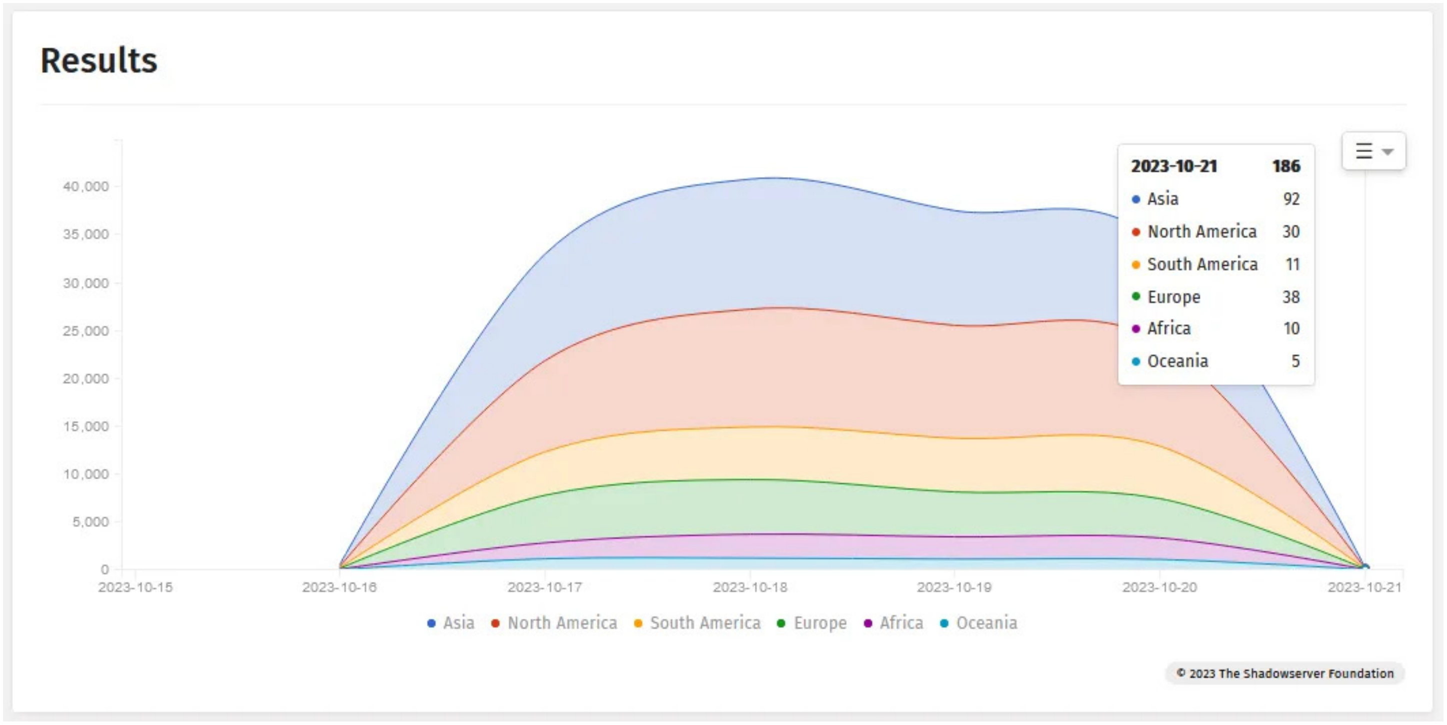
The CVSS 10.0 Cisco Nightmare

Meanwhile, Cisco has recently been dealing with a problem that's anything but latent. As we've often observed, really really bad vulnerabilities are typically awarded a CVSS score of 9.8. That's generally as high as it goes and even that is quite difficult to earn. I respect the reticence to go any higher, since we would like to reserve any assignment of 9.9 or 10.0 for those really just too horrible to believe problems.

So this means that Cisco's recent award of a CVSS of 10.0 really does mean something. What does it mean? It means that almost overnight, more than forty-one thousand of Cisco's high-end publicly-exposed Internet routers were compromised and taken over. Like I said, that's what

you want to reserve those 10.0's for.
The first known instances of attacks against Cisco's IOS XE-based routers and switches, which appear to have been initial proof-of-concept probing incursions, occurred at the end of last month on the 28th. Then more than three weeks passed before Cisco finally released the first fixes last Monday, October 16th. During those the intervening three weeks more than 42 thousand of Cisco's IOS XE-based devices were compromised, where compromise means full and effectively devastating takeover of the device. We know that it was 42 thousand devices because scanners were quickly created by security firms who wanted to track incursions. And in response to the visibility of their initial implants, the perpetrators of these attacks updated their malware to make it less visible. To all of this, the tech press responded as we'd expect. Just yesterday, SC Media lead with *"What the Cisco IOS XE zero-day reveals about potential risk management blind spots."* Help Net Security wrote *""Disappearing" implants, followed by first fixes for exploited Cisco IOS XE zero-day."* CRN wrote: *"Cisco Releases First In Series Of Patches For IOS XE Vulnerabilities"* A day earlier on Sunday, BleepingComputer's founder, Lawrence Abrams, wrote *"Hackers update Cisco IOS XE backdoor to hide infected devices."* Dark Reading wrote: *"Cyberattackers Alter Implant on 30K Compromised Cisco IOS XE Devices."* and The Register didn't disappoint with their headline: *"Cisco fixes critical IOS XE bug but malware crew way ahead of them. Initial fall in infected devices indicates evolution, not extinction, of attack code."*

Today's show notes contains a chart showing the initial infection rise, followed by the visible disappearance of the worldwide infection, not, as has been noted, because those 42 thousand Cisco Internet routers and switch were cleaned up, but because the malware was updated to burrow more deeply into the devices in order to hide:



In other words, what we have now is a real first class CVSS 10.0 mess.

As I mentioned, the news of this broke publicly for the first time on the 18th. The chart in the show notes shows that this corresponded with the peak of the detectable infections.

The problem that Cisco revealed was a zero-day, now CVE-2023-20198, which exists in the web administration panel of its IOS XE operating system. I could not count the number of times I've said on this podcast that all of our experience teaches us that it is **absolutely never safe** to expose any web-based administration to the public Internet. As an industry we simply do not yet know how to do that safely. If a network's management environment absolutely requires remote management access to a public router or switch, then use a VPN to connect to the private network behind the device, then access its web interface from the inside. NEVER expose any management-level web interface to the public Internet. There's just no reason to. "Ease of access" obviously also applies to the bad guys. And really, Cisco is at fault here in 2023. Not because they made a mistake – that can happen to anyone – but because it's **even possible** in this day and age to configure their devices to expose a web admin interface to any publicly routable network. It should not be possible. This is all enterprise grade high-end equipment. Anyone using it should have VPN technologies falling out of their pockets. So only exposing web admin to private networks would never impose a burden. And if Cisco **had** implemented such a policy, 41 thousand recently compromised routers and switches would not be, and Cisco could then safely make all the mistakes they might with their web admin authentication without it ever creating a global catastrophe.

And also note that this does not only apply to the enterprise and to enterprise class devices. It applies right now, today, to every one of us. It was an exactly analogous flaw in a Plex media server web admin interface, which was located in the home by a LastPass developer that was responsible for all of us having our LastPass vaults stolen and now being targeted for decryption. Unlike the Cisco case, it's probably asking too much for Plex to disallow Internet facing administration. But that individual, who was deep into the security space, should have done so him- or her-self. And everyone listening to this podcast should consider that, too. VPNs and Overlay networks are now so freely available that it's only a matter of taking the time to set them up. Okay, so enough preaching and ranting.

In today's Cisco case, the zero-day allowed threat actors to create an admin account with the highest level of privilege (level 15) on devices that had their WebUI panel exposed to the Internet. After some additional investigation, the presence of a second 0-day was discovered being used by the attackers to inject commands into the IOS XE filesystem that would execute with root privileges. That's never good. Cisco revealed that said the exploit chain was being used to inject a Lua-based backdoor on devices across the world.

Cisco initially believed that the second zero-day was the exploitation of a previously fixed bug (CVE-2021-1435), but in an update late last Friday, Cisco said the attackers had discovered and were using a new zero-day, now assigned CVE-2023-20273. But, believe it or not, both Cisco and CISA confirmed that the two-year-old patched flaw was **also** being exploited in the wild by a different threat actor in a different campaign. Wow. Like I said, there's never a good reason to expose all of this to the public Internet.

The other thing that was interesting was that the Lua backdoor that was originally placed on all 42+ thousand hacked Cisco IOS XE devices since late September started disappearing over last weekend. Estimates from both Censys and Shadowserver put the number of hacked devices at around 42,000 IOS XE routers and switches. As the chart I posted into the show notes shows, that number had remained steady until last weekend when it suddenly dropped to between 500 and 1,000. It’s believed that this change was made by the threat actor itself. The Lua backdoor was not a particularly strong persistence system, since it could be removed with a device reboot, and its public visibility was attracting too much attention to the attacker's operations. Several security researchers pointed out that Lua backdoor would likely have been replaced by something much more insidious, much as Chinese hackers evaded Barracuda’s patches earlier this year to entrench themselves so deeply into the compromised appliances that Barracuda ended up telling customers to replace their gear to be safe. At this point we don’t know that anything like this has happened but when vulnerabilities have been discovered which we know allows for the injection of commands into the device’s filesystem with root privileges, given sufficient time – and these attackers had three weeks – it’s clear that anything could be done.

So today, the Internet might well currently have something on the order of 42 thousand compromised Cisco routers and switches where their compromises have gone stealth after some threat actor somewhere finished burrowing in; and at this point who would ever know?

Once a device or a network has been compromised can it ever be fully trusted again? Today we finally have patches for both of the known 0-days that have made this possible. But applying the patches is only the first step. Even if some of the horses have left the barn, it’s still worthwhile to close the door. But you may still have some loose horses. Any enterprise whose Cisco edge gear may have been compromised would be well served to, unfortunately, perform a full and deep security audit into what may have happened on that device over the last month.

So far, no one in the infosec community has made even a tentative attribution. There’s no malicious actor who would not be interested in penetrating Cisco’s IOS XE gear, from hacktivists to APTs and from initial access brokers (you know, IAB’s) to botnet operators.

Giving Cisco credit where it’s due, it was they who discovered this new pair of 0-days during their investigation of a customer’s support ticket. Another vendor might not have bothered to engage a competent security team to dig into the trouble’s root cause.

The takeaway for each of us, individually, should be that our own networks should never expose anything publicly. You should never have your router’s web management interface enabled on the Internet regardless of your chosen username and password. And the same goes for Synology, Plex and everything else. Just take the time to setup a simple overlay network or your own VPN server to access those devices from the inside.

One of the things I’ve long wanted to do was to expand GRC’s service-port-only scan to check all 65,536 ports for it users. Back when I first wrote all of that code I was running GRC’s server from home over a pair of 1.54 megabit T1 connections. I was unable to offer to scan 64 times more ports. Today I could do that easily from Level3. Once I get to a place where I can turn my attention back to ShieldsUP! – which will be after SpinRite 7 has been released – I think it would be fun to boost ShieldsUP! to scan on demand all of everyone’s ports. :)

Closing the Loop

Stephen Lee / @AvidNetizen

Love your show! I was wondering: If you pronounce "lib" for library as rhyming with "vibe", do you pronounce "var" for variable as rhyming with "air"?

Tom / @TomFrillman

Hi Steve- I got the SR6.1 exe but Win 10 won't run it. How may I use it? Thanks for all you do. BTW, ValiDrive is great. I must be lucky, all 25 of my drives are fine except for one that won't even format...maybe Spinrite will help.

Last week's offer for SpinRite 6 owners to begin using the pre-release of 6.1 cause some confusion for exactly the reason that Tom mentioned. What we have today is the pre-release of the DOS component of SpinRite, not yet the hybrid Windows/DOS final. So what needs to be done is to insert an existing bootable SpinRite 6 bootable USB drive into Windows. The SPINRITE.EXE on it should be renamed to SR6.EXE and the SRPR.EXE that was downloaded should be copied to the USB drive and renamed to SPINRITE.EXE. I spent some time over this past weekend getting GRC's forums ready for SpinRite 6.1's release and I created a "How to run the SpinRite pre-release" page which can be found in the SpinRite FAQ section:
https://forums.grc.com/threads/how-to-run-the-spinrite-pre-release.1308/

And as for that one drive which Tom could not format. Rather than SpinRite, I'd let InitDisk take a crack at it. I'll bet it would get that drive going.

Doug Smith / @douglsmith

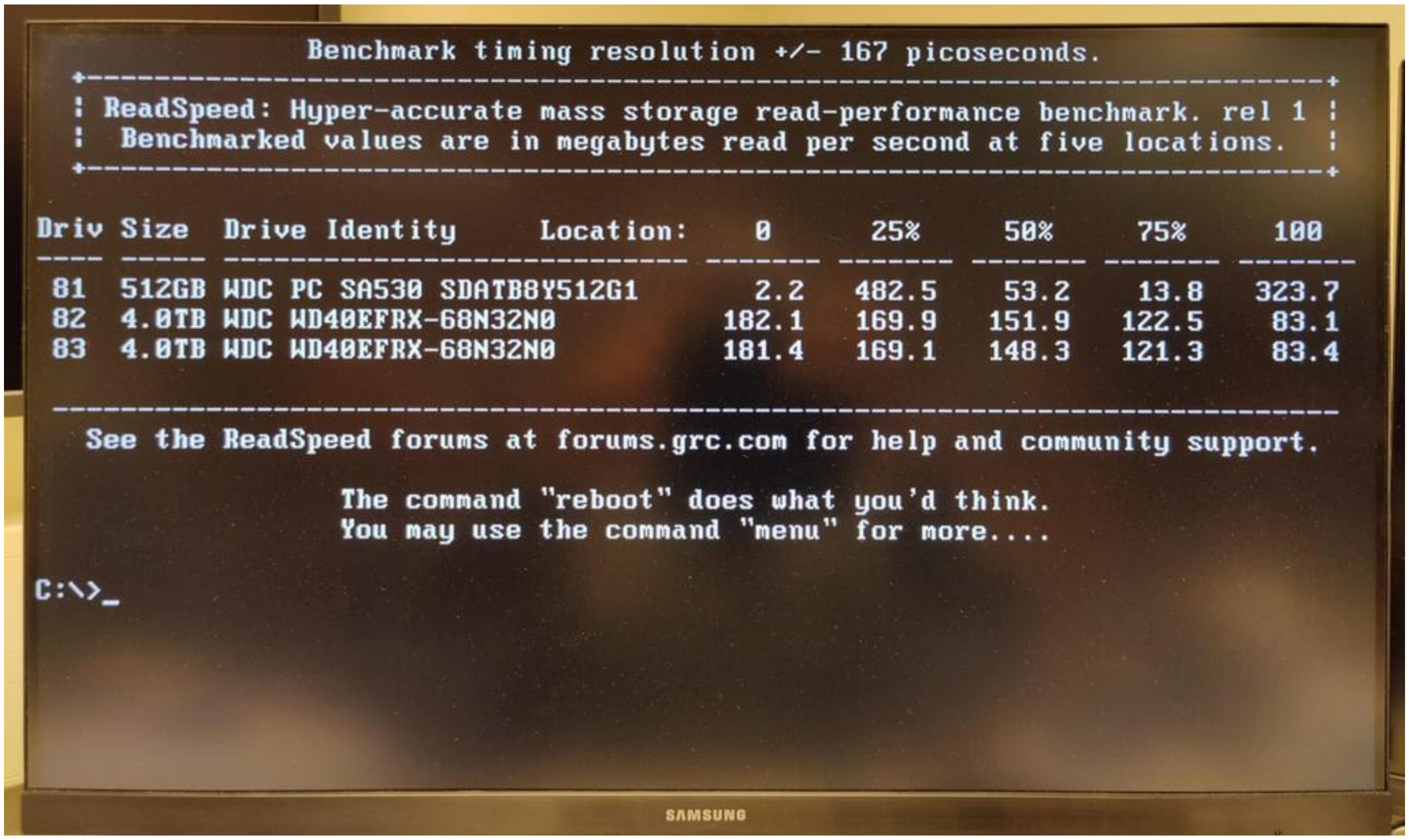
Regarding exporting and importing Passkeys, I saw this from a 1Password employee on their forums: "At this point in time, you cannot import or export passkeys from any managers like iCloud Keychain or 1Password. The good news is that we're working closely with platform vendors and other password managers through the FIDO Alliance to create a secure way to import and export passkeys. We believe it's your choice where to store and use your passkeys. Hopefully we'll have more to share soon."

Boy!! That's WONDERFUL news! It's just what we need. The user's experience would be to provide their own strong password to protect the passkey's export. This keeps it safe outside of the password manager. Then they would need to again provide it when they wish to import it elsewhere. That's what I did with SQRL and the technology to do this safely, unspoofably and verifiably is widely available.

Some of our listeners have suggested that users cannot be trusted to manage their identities safely; and we've certainly seen ample evidence of such mistakes. But I still think that restricting users' freedom is never the right course of action. I think users need education about where not to wander, rather than fences which limit their freedom.

Victor / @propellmestarn

I powered on a couple of years old desktop that had been unpowered for about a year. It took ages before the desktop was loaded, no errors anywhere but I decided to try your ReadSpeed. And look at those SSD speeds! Is it time to invest in SpinRite now? If SpinRite fixes this, I will try to encourage my employer to get a site license. Thank you mr. Gibson. Victor, Long-time SN listener, keep up the good work. To 999 and beyond ;)
https://twitter.com/messages/media/1714750052470526215



```
                 Benchmark timing resolution +/- 167 picoseconds.
   +----------------------------------------------------------------------+
   | ReadSpeed: Hyper-accurate mass storage read-performance benchmark. rel 1 |
   |  Benchmarked values are in megabytes read per second at five locations.  |
   +----------------------------------------------------------------------+

Driv Size  Drive Identity        Location:    0      25%      50%      75%     100
---- ----- ---------------------------- ------- ------- ------- ------- -------
 81  512GB WDC PC SA530 SDATB8Y512G1        2.2   482.5    53.2    13.8   323.7
 82  4.0TB WDC WD40EFRX-68N32N0           182.1   169.9   151.9   122.5    83.1
 83  4.0TB WDC WD40EFRX-68N32N0           181.4   169.1   148.3   121.3    83.4

  --------------------------------------------------------------------------
   See the ReadSpeed forums at forums.grc.com for help and community support.

                    The command "reboot" does what you'd think.
                    You may use the command "menu" for more....

C:\>_
```

Yes. Running any read/write level of SpinRite 6 or 6.1 will fix that immediately. The reason there will be a SpinRite 7 is that all we can do right now is rewrite the entire drive. We'd prefer not to do that, but today there's no choice. At least ReadSpeed shows the problem. What we need is surgical targeted spot rewriting, which is the new technology that SpinRite 7 will be bringing.

Andy Suarez / @AndyJSuarez

Hi Steve, great to hear you cover ECH. The whole time though I kept thinking that something missing from your coverage was that people should not be thinking that this is providing them cloaked internet access at all. Knowing what IP(s) are associated with what websites/services is definitely a thing. For anything other than the smallest websites there is absolutely a known correlation between ip addresses and the site behind them, and that is not in any way being obfuscated here. Services know the IPs for porn sites and political sites, and messaging services, and even http://www.grc.com (4.79.142.202). They may not be able to see the dns query (encrypted dns), or the header request (ECH) now, but they are certainly still seeing the IP of the webserver that you are sending and receiving packets from right? So won't tracking just change to keeping fresh lists of sites and what their associated IPs are?

Andy is right, of course. IPs are still IPs. But the more sites move behind large aggregators like Cloudflare, and thus share IPs, the less clear the user's destination becomes. But it's also unclear how this may change as the popularity of IPv6 grows. At the moment, the lack of IPv4 address space is forcing the sharing of addresses. But IPv6 promises to at least offer the opportunity of demultiplexing websites again. So I agree that Andy's reminder is a very good one. Not only was the Internet not initially designed to provide the security and privacy that we increasingly need, it was also never designed to hide where we go. TOR's concept of "Onion Routing" is the best solution we've come up with for obscuring our traffic. A trusted VPN is another solution. So, we would need no one to be able to know what domain IPs and ECH public keys we look up, Encrypted ClientHello to hide the domain within our traffic, and TOR routing or a trusted VPN provider to hide our traffic's destination.

New EOL / @EOLNew

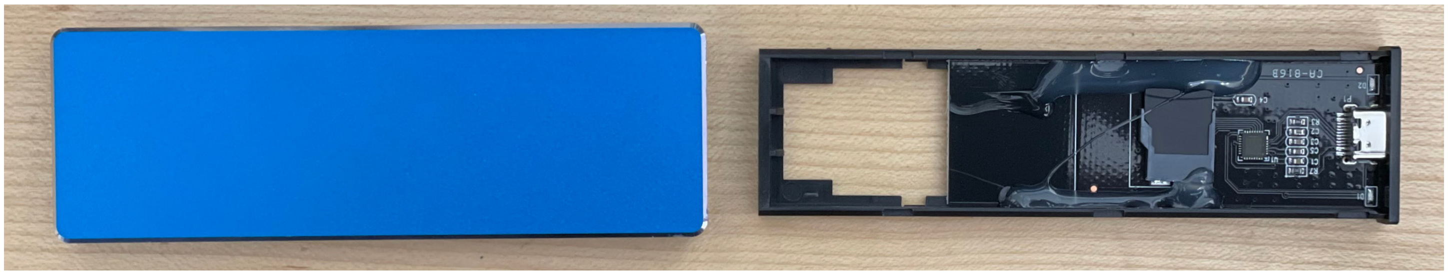
@SGgrc Will Ransomware be able to encrypt an AES256 encrypted 7-zip archive?

This one caught me a bit by surprise. But it was an interesting thought. This listener is wondering whether encrypting something **before** ransomware comes along to encrypt it, itself, could essentially "get there first" to prevent any further encryption. Unfortunately, it's definitely possible to encrypt something that's already been encrypted. In fact, that's something that the TOR onion routing relies upon. With TOR, a destination URL is encrypted using the public key of the last router in the chain. Then THAT once-encrypted URL is again encrypted using the second from the last router's public key. Then THAT twice-encrypted URL is again encrypted using the third from the last router's public key. And so on. Then the multiply-encrypted blob is sent to the first router. It decrypts the outer layer of the onion to obtain the identity of the next router in the chain, but since the destination URL is still triply-encrypted in the inner core of the onion, it has no idea where the user will eventually go.

Anyway, multiple encryptions do work and they even have some powerful and clever uses.

Daniel Hodgin / @dhhodgin

Hi Steve, Thank you for the new Validrive product and thank you for everything you guys do with Security Now. I decided to try it out this morning with a few USB keys and SD cards around our office. I then realized I wonder if this can work with SSD's as well with a USB connected cloning station. Sure enough i was able to test a 1TB Silicon Power brand SSD from amazon which turned out to be exactly as advertised. It made me wonder if the same kind of scam can be done with HDD's, SSD's, M.2 SSD's etc, which we are placing in corporate devices and assume our corporate data is being stored just fine once Windows reports the size as expected.



Two last feedback-inspired things I want to share...

Marcus Hitchins / @marcus7778

Privacy badger un-rewrites Google's rewritten links in search results

That caught my attention. Everyone knows how annoyed I am by Google's link-hacking where the link URL they show in the search results is deliberately not the link you get when you click it. They are tracking my choices by bouncing my browser through their server. The one beneficial reason I can see for doing this would be to use human feedback to help tune their search results. But we're all sure that they are also profiling everyone individually for profile building. They make all this noise about their privacy sandbox and how they want to do away with third-party tracking... but what about tracking every link I click on in their search results?

So, Privacy Badger is a creation of the EFF. They are people whose intentions we can trust. But I've often thought that it has an unfortunate name. Nevertheless, the idea that I could have something I trust stripping Google's URL link mangling out of Google's search results was too good to pass up. So I added it to Firefox and immediately confirmed that, sure enough, just as Marcus said, the links presented to me in Google's search results are exactly as they appear visually. No more subterfuge!

So I want to take this opportunity to touch on Privacy Badger a bit more. First of all, you can find it at **https://privacybadger.org/** (Badger) and it's also GRC's carefully numbered shortcut of the week, so just: **https://grc.sc/945**.

And I'm glad that I'm taking a longer look at it since what it does is interesting and unique. In describing it, the EFF writes:

Privacy Badger is a browser extension that stops advertisers and other third-party trackers from secretly tracking where you go and what pages you look at on the web. If an advertiser seems to be tracking you across multiple websites without your permission, Privacy Badger automatically blocks that advertiser from loading any more content in your browser. To the advertiser, it's like you suddenly disappeared.

Privacy Badger was born out of our desire to be able to recommend a single extension that would automatically analyze and block any tracker or ad that violated the principle of user consent; which could function well without any settings, knowledge, or configuration by the user; which is produced by an organization that is unambiguously working for its users rather than for advertisers; and which uses algorithmic methods to decide what is and isn't tracking.

As a result, Privacy Badger differs from traditional ad-blocking extensions in two key ways. First, while most other blocking extensions prioritize blocking ads, Privacy Badger is purely a tracker-blocker. The extension doesn't block ads unless they happen to be tracking you; in fact, one of our goals is to incentivize advertisers to adopt better privacy practices.

Second, most other blockers rely on a human-curated list of domains or URLs to block. Privacy Badger is an algorithmic tracker blocker – we define tracking behavior – and then Privacy Badger blocks or restricts domains that it observes tracking in the wild. What is and isn't considered a tracker is entirely based on how a specific domain acts, not on human judgment.

When you view a webpage, that page will often be made up of content from many different sources. (For example, a news webpage might load the actual article from the news company, ads from an ad company, and the comments section from a different company that's been contracted out to provide that service.) Privacy Badger keeps track of all of this. If as you browse the web, the same source seems to be tracking your browser across different websites, then Privacy Badger springs into action, preventing your browser from loading any more content from that source. And when your browser stops loading content from a source, that source can no longer track you. Voila!

At a more technical level, Privacy Badger keeps note of the "third party" domains that embed content – images, scripts and advertising in the pages you visit. Privacy Badger looks for tracking techniques like uniquely identifying cookies, local storage "supercookies," and canvas fingerprinting. If it observes a single third-party host tracking you on ***three*** *separate sites, Privacy Badger will automatically disallow content from that third-party tracker.*

As for their browser support, it's as universal as it could be. Their FAQ asks:

Will you be supporting any other browsers besides Chrome, Firefox, Edge and Opera?

We are working towards Safari on macOS support. Safari on iOS seems to lack certain extension capabilities required by Privacy Badger to function properly. Chrome on Android does not support extensions. To use Privacy Badger on Android, install Firefox for Android. Privacy Badger does not work with Microsoft Edge Legacy. Please switch to the new Microsoft Edge browser. Note that Microsoft Edge does not support extensions on mobile devices.

They also noted:

Privacy Badger sends the ***Global Privacy Control*** *signal, to opt you out of data sharing and selling, and the* ***Do Not Track*** *signal to tell companies not to track you. If they ignore your wishes, Privacy Badger will learn to block them—whether they are advertisers or trackers of other kinds.*

And I thought this was very interesting and cool from their FAQ:

I run a domain that uses cookies or other tracking. How do I stop Privacy Badger from blocking me?

One way is to stop tracking users who have turned on Global Privacy Control or Do Not Track signals (i.e., stop collecting cookies, supercookies or fingerprints from them). Privacy Badger will stop learning to block that domain. The next version of Privacy Badger to ship with an updated pre-trained list will no longer include that domain in the list. Most Privacy Badger users will then update to that list.

You can also unblock yourself by promising to meaningfully respect the Do Not Track signal. To do so, post a verbatim copy of EFF's Do Not Track policy to the URL https://example.com/.well-known/dnt-policy.txt, where "example.com" is replaced by your domain. Posting EFF's DNT policy on a domain is a promise of compliance with EFF's DNT

Policy by that domain.

If your domain is compliant with EFF's DNT policy and declares this compliance, most Privacy Badgers will see this declaration the next time they encounter your domain. Also, the next version of Privacy Badger to ship with an updated pre-trained list will probably include your declaration of compliance in the list.

Note that the domain must support HTTPS, to protect against tampering by network attackers. The path contains ".well-known" per RFC 5785. Also note that you must post a copy of the policy at each compliant subdomain you control. For example, if you wish to declare compliance by both sub1.example.com and sub2.example.com, you must post EFF's DNT policy on each domain.

To me, this reads like a solid piece of mature technology that I'm glad to have added to my browsing experience. So I wanted everyone to know about it.

Finally, recall the guy from last week whose Tweet asked about obtaining the pre-release of SpinRite? That was the tweet that prompted me to mention that it was available to everyone who was interested. This listener was Brett Russell, and as a reminder he tweeted:

Hi Steven, I realize this is a request out of the blue, but my hard disk is failing on my main machine. I own a copy of Spinrite (code =REDACTED=), but the disk is set up as GPT, which 6.0 doesn't support. Any chance you can give me a pre-release of 6.1 to run, please? There is nothing critical on the drive, but if it dies, it will take some time to bring it all back.

I replied with the instructions about obtaining the current SpinRite DOS executable pre-release. And then, when I was catching up on Twitter yesterday, I found three tweets from Brett:

Thanks Steve, much appreciated. Running it now.

Wow, it's fast. Feels a lot faster than 6.0. I like the new drive identification screen as well.

And, finally:

It worked, machine is up and running, faster than ever (although no errors were picked up, which I have seen before). Setting up a RAID mirror for when the drive does die. You saved me a lot of time and effort. Thank you.

Brett's comment about "no errors were picked up" is a common and sometimes mysterious experience for SpinRite's users. After running SpinRite, whatever was wrong is fixed, but SpinRite may not explicitly show that anything was done. The reason for this somewhat unsatisfying outcome is that SpinRite induces the drive to deal with marginal problems internally. Before SpinRite's deep recovery is needed, the drive will remove a sector from use and all will be well again. But drives typically don't report doing so. So all SpinRite and its user knows is that things are better afterward.

The Power of Privilege

https://www.cisa.gov/news-events/cybersecurity-advisories/aa23-278a

Today, we continue our examination of the NSA's and CISA's top 10 cybersecurity misconfiguration mistakes by drilling down deeply into their #2 misconfiguration: **Privilege.**

What strikes me most about this publication is that each item is quite beefy and worthwhile. Unlike some recommendation lists that feel like that were written by someone pontificating from an armchair, these resulted from their active successful penetration of networks.

So, here's what they had to say on the topic of "Improper Separation of User/Administrator Privilege"...

Administrators often assign multiple roles to one account. These accounts have access to a wide range of devices and services, allowing malicious actors to move through a network quickly after obtaining ***one*** *compromised account without triggering lateral movement and/or privilege escalation detection measures. Assessment teams have observed the following common account separation misconfigurations:*

- *Excessive account privileges*
- *Elevated service account permissions*
- *Non-essential use of elevated accounts*

Excessive Account Privileges

Account privileges are intended to control user access to host or application resources to limit access to sensitive information or enforce a least-privilege security model. When account privileges are overly permissive, users can see and/or do things they should not be able to, which becomes a security issue as it increases risk exposure and attack surface.

Expanding organizations can undergo numerous changes in account management, personnel, and access requirements. These changes commonly lead to privilege creep—the granting of excessive access and unnecessary account privileges. Through the analysis of topical and nested AD groups, a malicious actor can find a user account that has been granted account privileges that exceed their need-to-know or least-privilege function. Extraneous access can lead to easy avenues for unauthorized access to data and resources, and escalation of privileges in the targeted domain.

Elevated Service Account Permissions

Applications often operate using user accounts to access resources. These user accounts, which are known as service accounts, often require elevated privileges. When a malicious actor compromises an application or service using a service account, they will have the same privileges and access as the service account.

Malicious actors can exploit elevated service permissions within a domain to gain unauthorized access and control over critical systems. Service accounts are enticing targets for malicious actors because such accounts are often granted elevated permissions within the domain due to the nature of the service, and because access to use the service can be requested by any valid *domain user. Due to these factors, "kerberoasting" — a form of credential access achieved by cracking service account credentials—is a common technique used to gain control over service account targets.*

Non-Essential Use of Elevated Accounts

IT personnel use domain administrator and other administrator accounts for system and network management due to their inherent elevated privileges. When an administrator account is logged into a compromised host, a malicious actor can steal and use the account's credentials and an AD-generated authentication token to move, using the elevated permissions, throughout the domain. Using an elevated account for normal day-to-day, non-administrative tasks increases the account's exposure and, therefore, its risk of compromise and its risk to the network.

Malicious actors prioritize obtaining valid domain credentials upon gaining access to a network. Authentication using valid domain credentials allows the execution of secondary enumeration techniques to gain visibility into the target domain and AD structure, including discovery of elevated accounts and where the elevated accounts are used.

Targeting elevated accounts (such as domain administrator or system administrators) performing day-to-day activities provides the most direct path to achieve domain escalation. Systems or applications accessed by the targeted elevated accounts significantly increase the attack surface available to adversaries, providing additional paths and escalation options.

After obtaining initial access via an account with administrative permissions, an assessment team compromised a domain in under a business day. The team first gained initial access to the system through phishing, by which they enticed the end user to download and execute malicious payloads. The targeted end-user account had administrative permissions, enabling the team to quickly compromise the entire domain.

Okay. So there are two primary things I want to elaborate on and highlight.

The first is that idea of mission creep. They called it privilege creep and anyone who has been in charge of managing any complex environment of privileges will have seen and been fighting this.

I've often observed that code generally does not evolve well. By its nature, code is not designed to evolve. There will generally be a grand organizing concept for a project. And that vision will be realized in code. But that organizing vision influences the code at all levels. The way the large project is decomposed into smaller manageable components. The way those components talk to each other, what they are able to say, and how. The data structures those components use to capture and manage the project's state. Everyone one of those things is generally static, interdependent, and thus resistant to change.

I've also observed that it's often the case that once a project is finished it really should be entirely discarded then redesigned and reimplemented from scratch. The reason for that is not necessarily that the project itself evolved during its implementation – but that those who were implementing it did. They were changed by doing the work because the process of solving any sufficiently complex problem teaches those who are solving it how it should originally have been designed. For sufficiently complex projects several iterations of that sort can be required.

Unfortunately, outside of academic settings the luxury of multiple iterations is seldom available.

So now, let’s loop back to account privileges to see how this applies:

The primary message here is that the management of privileges provides a tremendous amount of power. And, as always, with power comes responsibility.

Any major enterprise’s account privileges hierarchy needs to be carefully designed. In a sufficiently complex environment the concept of “least privilege” is much more easily espoused than it is implemented. Mistakes in either direction result in problems. Users need varying levels of access, and differing groups of users may have both disjoint and overlapping regions of access at the same level. Autonomous devices also need their own access. And it’s difficult enough to get everything set up correctly given a static environment without any change. But when did nothing ever change?

The real test is the test of time, brought about by an enterprise’s evolving structures and needs. What this means is that any initial design will be almost immediately obsoleted once it faces reality.

The asymmetry of “privilege” is another factor: Too little privilege and things break, printers don’t print, users are unable to do what they need to and the boss is not happy. Too much privilege and everything works, but users can get up to mischief and things are made much easier for the bad guys... and, again, the boss is not happy. So assigning and managing privilege is a bit like walking a tightrope. Stepping off in either direction does not produce a good result.

I think the best advice that can be given is to clearly recognize and accept that the proactive management of “Privilege” is a truly important aspect of the enterprise’s entire operation. Designing and managing an enterprise’s account and access privileges should never be a grudging afterthought. Be sure to give it its due. Fortunately, it is never too late to repair an organization's no-longer-applicable privilege model. If this is your responsibility you likely already know what needs to be done. Don’t wait until it’s too late.

Next week we’ll continue moving through this excellent joint NSA/CISA document, as we keep up with the news of the week!