Security Now! #777 - 07-28-20

rwxrwxrwx

This week on Security Now!

This week we revisit the trouble with F5 Networks' Big-IP devices, we update on the epic Twitter hack and we look at a security update for GnuTLS. We also cover the big 5-day Garmin "outage" and Cisco's latest troubles. We'll point out a new Win10 debloater app and a bit of errata. Then I want to wrap-up by sharing some truly surprising and interesting results that are emerging from my work on the pre-SpinRite hyper-accurate storage benchmark.

Our PC Reality Today

Security News

F5 Networks "Big-IP" devices in Big-Trouble

So let's recall that at the beginning of the month a "maximum vulnerability" remote code execution flaw was disclosed in F5 Networks' Traffic Management User Interface (TMUI) of the BIG-IP application delivery controller (ADC). This came to light as F5 had published a patch, so the urgent call was for any affected users of these BIG-IP systems to immediately update with the highest possible priority.

It is known that F5's customers using BIG-IP solutions include governments, Fortune 500 firms, banks, service providers, and well-known brands including Microsoft, Oracle, and Facebook. As we noted at the time, F5's website boasts that "48 of the Fortune 50 rely on F5".

At the time of the disclosure more than 8,000 of these devices were found online and vulnerable to attacks designed to exploit this vulnerability. US Cyber Command urged F5 customers to urgently patch their devices, tweeting "patching CVE-2020-5902 and 5903 should not be postponed over the weekend. Remediate immediately."

F5 also offered some interim mitigation measures that they recommended for their customers who could not, for whatever reason, patch their BIG-IP equipment immediately. But it later came to light that the mitigation could be mitigated and bypassed which made emergency patching the only safe course.

Two days after the patches for this critical F5 BIG-IP vulnerability were released, security researchers started publicly posting proof-of-concept (PoC) exploits showing just how easy it is to exploit these devices.

https://us-cert.cisa.gov/ncas/alerts/aa20-206a

So that was then. Three weeks later, last Friday the 24th, the Cybersecurity and Infrastructure Security Agency (CISA) posted:

> CISA is issuing this alert in response to recently disclosed exploits that target F5 BIG-IP devices that are vulnerable to CVE-2020-5902. F5 Networks, Inc. (F5) released a patch for CVE-2020-5902 on June 30, 2020.
>
> Unpatched F5 BIG-IP devices are an attractive target for malicious actors. Affected organizations that have not applied the patch to fix this critical remote code execution (RCE) vulnerability risk an attacker exploiting CVE-2020-5902 to take control of their system. Note: F5's security advisory for CVE-2020-5902 states that there is a high probability that any remaining unpatched devices are likely already compromised.
>
> CISA expects to see continued attacks exploiting unpatched F5 BIG-IP devices and strongly urges users and administrators to upgrade their software to the fixed versions. CISA also advises that administrators deploy the signature included in this Alert to help them determine whether their systems have been compromised.

CISA has observed scanning and reconnaissance, as well as confirmed compromises, within a few days of F5's patch release for this vulnerability. As early as July 6, 2020, CISA has seen broad scanning activity for the presence of this vulnerability across federal departments and agencies—this activity is currently occurring as of the publication of this Alert.

CISA has been working with several entities across multiple sectors to investigate potential compromises relating to this vulnerability. CISA has confirmed two compromises and is continuing to investigate. CISA will update this Alert with any additional actionable information.

I've often spoken about the growing critical need for companies and for individuals who care to be certain that they have an open channel of communication for receiving vulnerability notices. I had been talking about eMail as that channel. But in thinking about this further, I think that Twitter likely makes more sense. As I noted last week, it really has become our global information dissemination platform. And no one imagines that the announcement of critically patched vulnerabilities is not going to be public anyway. I'm sure the bad guys have signed up to receive vulnerability announcement eMail from many providers.

So, the way this needs to work is for technology companies to create an authenticated vulnerability announcement Twitter account which =never= contains corporate promotional nonsense. The company wants to keep their channel clean so that their customers remain subscribed and following. The channel should be named and dedicated to the public disclosure of the availability of important updates to correct important security vulnerabilities. This should be seen as beneficial to its reputation, and a company's record of disclosure should be a point of pride for the company and seen as an aspect of security for its prospective and ongoing customers.

If an enterprise's entire IT security team subscribes to the set of vendors whose hardware and software they are using, tweets containing calls to action should be reliably picked up and acted upon in as close to near real-time as possible.

And Speaking of Twitter

It's obvious in retrospect that if high-profile accounts were compromised so that attackers were able to obtain login access, they would or could have nosed around in the normally private DM channels of those accounts.

The evening of the day after last week's podcast, Twitter updated their blob posting with some additional news.

https://blog.twitter.com/en_us/topics/company/2020/an-update-on-our-security-incident.html

The bit that was added was:

"We believe that for up to 36 of the 130 targeted accounts, the attackers accessed the DM inbox, including 1 elected official in the Netherlands. To date, we have no indication that any other former or current elected official had their DMs accessed." [Added on July 22, 2020]

Not surprisingly, the news that some of the world's most influential people probably had their personal messages read by hackers who are still unknown, at least publicly, will put additional pressure on Twitter to better protect its users. US Senator Ron Wyden, who is generally one of the more technically savvy politicians, said that he has pushed Twitter's CEO Jack Dorsey to protect direct messages with end-to-end encryption. Ron said: "Twitter DMs are still not encrypted, leaving them vulnerable to employees who abuse their internal access to the company's systems, and hackers who gain unauthorized access. If hackers gained access to users' DMs, this breach could have a breathtaking impact, for years to come." And from Twitter's standpoint, it would be a big feather in its cap if it could boast true end-to-end encryption for private DMs.

The idea, of course, would be that neither Twitter nor anyone else except the tweet's intended recipient would be able to read the tweets. I'd love to be given the job of designing that system, since it presents a number of interesting challenges. But probably the best person anywhere would be Matthew Rosenfeld, whom we commonly refer to as "Moxie Marlinspike". Matthew, or rather, Moxie and his crypto team at Signal would be best suited to designing E2EE into Twitter. I still recall how weirdly over-designed the Signal protocol appeared at first. But that morphed into deep technical appreciation once I saw what their crypto-ratchet and other mechanisms enabled. That's the sort of design expertise—and inevitable crypto mistake side-stepping—Twitter needs. If anyone out there at Twitter is listening, and if you have any interest in following up on Ron Wyden's E3EE suggestion, PLEASE please please don't roll your own ad hoc solution. I'll bet Moxie and his team would welcome a new and high-profile challenge.

GnuTLS

We've often referred to OpenSSL as the standard. But as we know, it's becoming quite long in the tooth. Professor of Cryptography, Bill Buchanan recently summed up the situation with OpenSSL writing on Medium: "OpenSSL has caused so many problems in the industry including the most severe with Heartbleed. The problem with it is that it has been cobbled together and maintained on a shoe-string budget." As we've discussed in the past, is exactly true. Many developers come and go. They're working on this or that extension to SSL, so they swing by the OpenSSL Project, graft on their new widget for live testing, then leave it hanging there without anyone to care for it. It's kind of amazing that it's done as well as it has.

Today, where there was once one, there are now many. Google forked OpenSSL to create "BoringSSL" and the OpenBSD project also forked OpenSSL to create "LibreSSL". Amazon took a different approach for securing the communications to their cloud services by creating a minimal subset of the whole, named "s2n" standing for "signal to noise". And more recently, Google has created and released "Tink", which is a multi-language cross-platform crypto library.

And then there's GnuTLS... the subject of this segment:

GnuTLS was first created a little over 17 years ago, in 2003, as a means for allowing GNU Project applications to communicate securely over SSL and TLS. Although OpenSSL existed at the time, OpenSSL's license is not compatible with the GPL. Therefore, software licensed under the GPL, such as all of GNU's software, could not use OpenSSL without making a GPL linking exception. The GnuTLS library was licensed originally under the GNU Lesser General Public

License v2, which included applications using the GNU General Public License.
Then, in August 2011, the library was updated to the LGPLv3. But then, once it was noticed that there were new license compatibility problems introduced, especially with other free software with the license change, in March of 2013 the license was returned to LGPLv2.1 in March 2013.

So, under one license or another, GnuTLS has been around since 2003 and, not surprisingly, it's found its way into a great many applications. Just to get your attention, I'll name a few: apt, cadaver (aka WebDAV), curl, wget and git, GNOME, CenterIM, Exim, Weechat, mariadb, mandos, Mutt, Wireshark, rsyslog, slrn, Lynx, CUPS, gnoMint, GNU Emacs, slapd, samba, Synology DiskStation Manager, OpenConnect, and various VNC implementations.

This is why it's worth taking note, and looking into the situation more closely, when the result of a recent audit of GnuTLS is summed up: "Be Afraid". Linux users need to determine how afraid they, individually, should be.

https://nvd.nist.gov/vuln/detail/CVE-2020-13777

NIST explains the problem much less dramatically by writing: "GnuTLS 3.6.x before 3.6.14 uses incorrect cryptography for encrypting a session ticket (a loss of confidentiality in TLS 1.2, and an authentication bypass in TLS 1.3). The earliest affected version is 3.6.4 (2018-09-24) because of an error in a 2018-09-18 commit. Until the first key rotation, the TLS server always uses wrong data in place of an encryption key derived from an application."

When the news of these audit results broke, cryptographer Filippo Valsorda (@FiloSottile), Google's security team lead for their GO language, tweeted:

> Don't rely on GnuTLS, please.
>
> CVE-2020-13777 Whoops, for the past 10 releases most TLS 1.0–1.2 connections could be passively decrypted and most TLS 1.3 connections intercepted. Trivially.
>
> Also, TLS 1.2–1.0 session tickets are awful.

https://anarc.at/blog/2020-06-10-gnutls-audit/

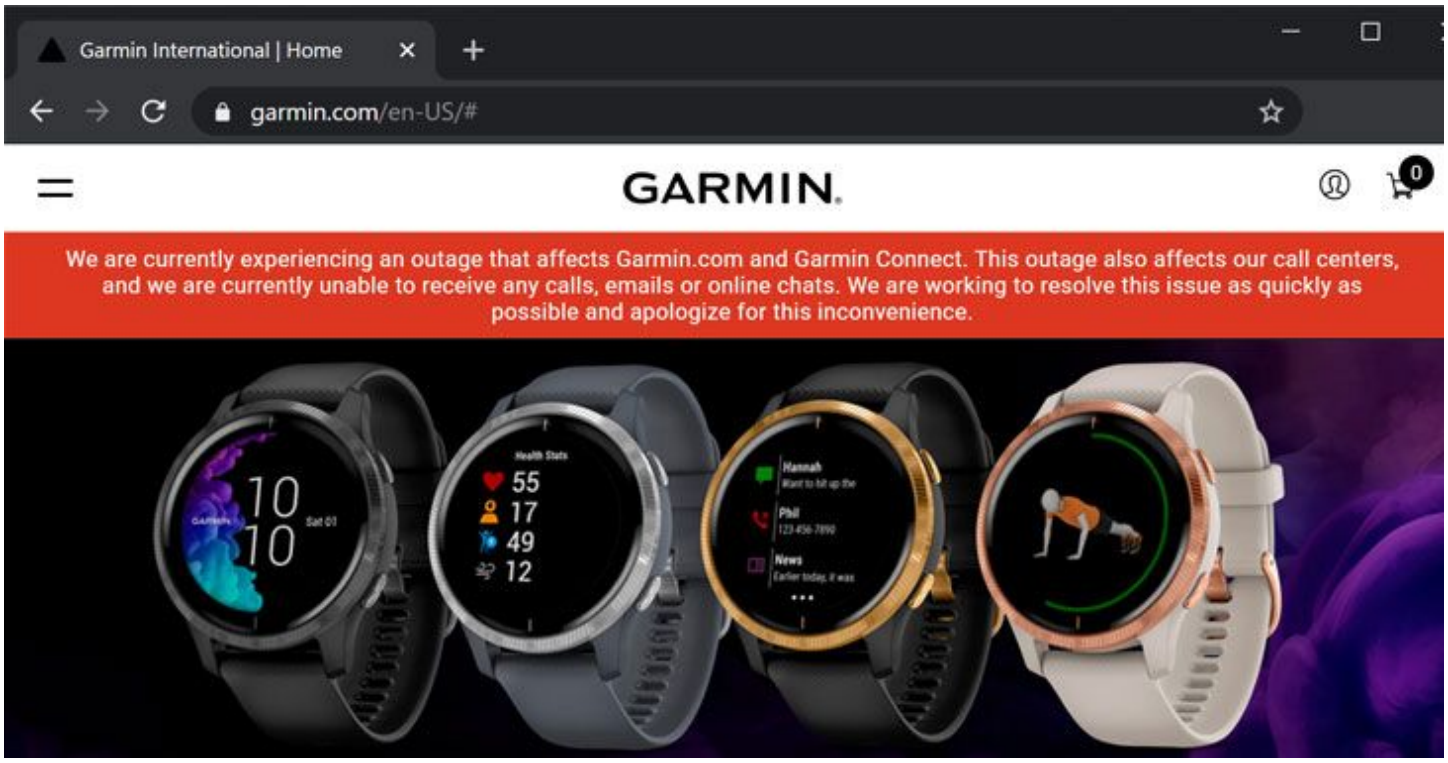
Someone who's hip to security quoted Filippo's tweet and wrote:

> You are reading this correctly: supposedly-encrypted TLS connections made with affected GnuTLS releases are vulnerable to a passive cleartext recovery attack (and active for 1.3, but who uses that anyways). That is extremely bad. It's pretty close to just switching everyone to HTTP instead of HTTPS, more or less. I would have a lot more to say about the security of GnuTLS in particular — and security in general — but I am mostly concerned about patching holes in the roof right now, so this article is not about that.
>
> This article is about figuring out what, exactly, was exposed in our infrastructure because of this.

A bit later, summing things up, he writes: “The impact of this vulnerability depends on the affected packages and how they are used. It can range from "meh, someone knows I downloaded that Debian package yesterday" to "holy crap my full disk encryption passwords are compromised, I need to re-encrypt all my drives", including "I need to change all LDAP and MySQL passwords".”

The link to this author's full blog posting is in the show notes for anyone who's interested in more detail. For now, just know that many Linux packages that are linked to the GnuTLS library will have been recently updated. So Linux users will want to be on the lookup for those.

The Garmin “Outage” ... THEN:



And Now...

https://www.garmin.com/en-US/outage/

After last year's overkill on coverage of ransomware, I promised to stop mentioning this scourge of the industry week after week. And I've been good since then, even though ransomware attacks form a constant background. And yes, when some random dentist in Hoboken needs to cancel his appointments — if, indeed, he's still in business after the novel coronavirus — I agree that more pressing matters need our attention.

However, when a high profile highly networked Internet-connected and Internet-dependent giant like Garmin gets its servers encrypted and needs to go dark, that's worth a mention.

Garmin's troubles began late Wednesday / early Thursday morning as customers reported being unable to use a variety of their services. This came as no surprise to Garmin. They Tweeted:

This service failure left their millions of customers unable to connect their smartwatches, fitness trackers, and other devices to servers that provided the location-based data required to make them work. Although many within the industry suspected exactly what happened, Garmin's post yesterday was the first time the company provided a cause of the worldwide outage:

> Garmin Ltd. was the victim of a cyber attack that encrypted some of our systems on July 23, 2020. As a result, many of our online services were interrupted including website functions, customer support, customer facing applications, and company communications. We immediately began to assess the nature of the attack and started remediation.
>
> We have no indication that any customer data, including payment information from Garmin Pay, was accessed, lost or stolen. Additionally, the functionality of Garmin products was not affected, other than the ability to access online services. Affected systems are being restored and we expect to return to normal operation over the next few days.
>
> As our affected systems are restored, we expect some delays as the backlog of information is being processed. We are grateful for our customers' patience and understanding during this incident and look forward to continuing to provide the exceptional customer service and support that has been our hallmark and tradition.

Screenshots and other data posted by employees suggested the ransomware was a relatively new strain called "WastedLocker". A person with direct knowledge of Garmin's response over the weekend confirmed that WastedLocker was, indeed, the ransomware used. The person spoke on condition of anonymity to discuss a confidential matter with the technical press.

WastedLocker first came to public attention about two and a half weeks ago, on July 10, when Malwarebytes published a "Threat Spotlight" profile.

https://blog.malwarebytes.com/threat-spotlight/2020/07/threat-spotlight-wastedlocker-customized-ransomware/

In their Spotlight Malwarebytes said that WastedLocker attacks are highly targeted against organizations chosen in advance. During the initial intrusion the malware conducts a detailed analysis of active network defenses so that subsequent penetrations can better circumvent them.

Pieter Arntz, a Malwarebytes researcher, wrote:

> "In general, we can state that if this gang has found an entrance into your network it will be impossible to stop them from encrypting at least part of your files. The only thing that can help you salvage your files in such a case is if you have either roll-back technology or a form of off-line backups. With online, or otherwise connected backups you run the chance of your backup files being encrypted as well, which makes the whole point of having them moot. Please note that the roll-back technologies are reliant on the activity of the processes monitoring your systems. And the danger exists that these processes will be on the target list of the ransomware gang. Meaning that these processes will be shut down once they gain access to your network."

The MalwareBytes posting also notes:

"WastedLocker" is a new ransomware operated by a malware exploitation gang commonly known as the "Evil Corp" gang. The same gang that is associated with Dridex and BitPaymer. The attribution is not based on the malware variants, as WastedLocker is very different from BitPaymer. What was retained, was the ability to add specific modules for different targets.

The attacks performed using WastedLocker are highly targeted at very specific organizations. It is suspected that during a first penetration attempt, an assessment of active defenses is made. And the next attempt will be specifically designed to circumvent the active security software and other perimeter protection which the initial foray found to be in use. This effort represents a new and clear escalation of the ransomware scourge. We're no longer looking at opportunistic attacks which ask for some fraction of a bitcoin. If reports are to be believed, including the US Dept. of the Treasury, the bad guys are now highly organized Russian cybercriminal gangs. They're not screwing around.

The name "WastedLocker" is derived from the filename it creates which includes an abbreviation of the victim's name and the string "wasted". For each encrypted file, the attackers create a separate file that contains the ransomware note. The ransom note has the same name as the associated file with the addition of "_info".

Once the WastedLocker gang have taken hold in a network, their demands typically range from $500,000 to $10 million. This is the new face of international cybercrime extortion. If hackers delete or steal a company's data, there's nothing to extort. But if hackers encrypt a corporation's data they're able to dangle the carrot of the decryption key. It's diabolical.

Garmin's notice yesterday did not employ the terms "ransomware" or "WastedLocker". But the description "cyber attack that encrypted some of our systems," all but definitively confirms that ransomware of some sort was behind the "outage." And we have disclosures from unnamed but presumably reliable Garmin insiders.

We all want to know whether or not Garmin paid up or restored from backups. And if they anted up, what did they pay? Sky News, citing a number of unnamed security sources, reported that Garmin **did** obtain the decryption key. And that report matched with what the person with direct knowledge told members of the tech press. Sky News said Garmin "did not directly make a payment to the hackers," but did not elaborate further. As we've discussed before, there are now middlemen agencies who negotiate on behalf of their ransomware victim clients. Payment may have been made through such an intermediary. Garmin's representatives declined to provide confirmation that the malware was WastedLocker and whether or not the company paid ransom.

There's no benefit to them in elaborating, and, in fact, it might cause them more trouble. On December 5th of last year, the US Department of Treasury officially sanctioned Russia's "Evil Corp", citing the Russia-based cybercriminal group as being behind the Dridex malware:

https://home.treasury.gov/news/press-releases/sm845

The US Treasury Department's announcement started:

> "Today the U.S. Treasury Department's Office of Foreign Assets Control (OFAC) took action against Evil Corp, the Russia-based cybercriminal organization responsible for the development and distribution of the Dridex malware. Evil Corp has used the Dridex malware to infect computers and harvest login credentials from hundreds of banks and financial institutions in over 40 countries, resulting in more than $100 million in theft."

So the US Treasury's action could complicate Garmin's position with respect to Evil Corp. It could open Garmin to legal action for paying the Russian crime gang in exchange for the decryption keys and thus access to their decrypted data. Today, Garmin's services are now mostly if not entirely back online.

Attacks are driven by motivation, and few things motivate like cold hard cash. Ransomware has emerged as an insidious but viable technique for the extraction of cash from those who have it and those who have been caught without adequate failsafe fallbacks in the event of such an intrusion. As our listeners know, when ransomware first appeared and was covered on this podcast, our reaction was ..."uhh oh"... because it was clear that in-place encryption, coupled with cyber-currency, enabled a significant new threat. To that, we now add tightly targeted attacks launched by international organized cybercrime groups. Staying current with security updates, keeping employees on guard against intrusion spoofs, and maintaining offline backups in case bad guys get in anyway is the order of the day.

Thus ends our ransomware reminder wakeup call for 2020. Now back to our regularly scheduled news...

Cisco's latest trouble

Speaking of going where the money is and limiting ingress to high value targets, we have the sad patching status of Cisco's most recent critical vulnerability in tasty enterprise-grade devices. And when I tell you that it's another directory path traversal mistake try not to roll your eyes.

https://tools.cisco.com/security/center/content/CiscoSecurityAdvisory/cisco-sa-asaftd-ro-path-KJuQhB86

Last Wednesday the 22nd, Cisco released their Security Advisory with a CVSS score of 7.5. But... uhhhhh... today it's seeming somewhat worse than that. Cisco's advisory reads:

> A vulnerability in the web services interface of Cisco Adaptive Security Appliance (ASA) Software and Cisco Firepower Threat Defense (FTD) Software could allow an unauthenticated, remote attacker to conduct directory traversal attacks and read sensitive files on a targeted system.
>
> The vulnerability is due to a lack of proper input validation of URLs in HTTP requests processed by an affected device. An attacker could exploit this vulnerability by sending a crafted HTTP request containing directory traversal character sequences to an affected device. A successful exploit could allow the attacker to view arbitrary files within the web services file system on the targeted device.
>
> The web services file system is enabled when the affected device is configured with either WebVPN or AnyConnect features. This vulnerability cannot be used to obtain access to ASA or FTD system files or underlying operating system (OS) files.
>
> Cisco has released software updates that address this vulnerability. There are no workarounds that address this vulnerability.
>
> ***Note:*** *Cisco has become aware of the availability of public exploit code and active exploitation of the vulnerability that is described in this advisory. Cisco encourages customers with affected products to upgrade to a fixed release as soon as possible.*

Over time this podcast has compiled a few "Golden Rules of Cyber Security." I may not have explicitly stated this one, but it clearly ranks among the most important: "Web interfaces are dangerous, don't use them." Oh... they're pretty. And they mean that the IT guys don't need to read yet another boring manual listing confusing-looking commands. No. A web interface means that you can just fire it up and poke around in the menus until you find the button you're looking for... then press it. Unfortunately, so too can the bad guys. One already well-established Golden Rule is that interpreters are incredibly difficult to make perfect. Yet, perfection there is required because the job being performed by most interpreters asks them to interpret untrusted content. And, the interpreter in any web server is right up there in complexity and exploitability with that of any multi-media codec.

Mikhail Klyuchnikov of Positive Technologies who was credited with independently reporting this flaw along with Ahmed Aboul-Ela of RedForce said: "This vulnerability is highly dangerous. The

cause is a failure to sufficiently verify inputs. An attacker can send a specially crafted HTTP request to gain access to the file system (RamFS), which stores data in RAM."

Last week at the time of the disclosure no attacks were known to be underway. But Ahmed Aboul-Ela of RedForce released a Proof-of-Concept (PoC) which demonstrated the vulnerability, and he's been tweeting up a storm of example ways to exploit the flaw ever since. The bad news is, it's horrifically easy to exploit, and also horrifically trivial to find vulnerable targets. Which brings us to the state of affairs as of today. It's not good.

As the update to Cisco's vulnerability announcement notes, attacks are underway. Radid7 jumped on this and took a look at what it meant:

https://blog.rapid7.com/2020/07/23/cve-2020-3452-cisco-asa-firepower-read-only-path-traversal-vulnerability-what-you-need-to-know/

In their report from last Thursday, they noted:

> Rapid7 encourages immediate patching of vulnerable ASA/FTD installations to prevent attackers from obtaining sensitive information from these devices which may be used in targeted attacks. There are no workarounds that address this vulnerability.
>
> Cisco has provided fixes for all supported versions of ASA and FTD components. Cisco ASA Software releases 9.5 and earlier, as well as Release 9.7, along with Cisco FTD Release 6.2.2 have reached the end of software maintenance and organizations will have to upgrade to a later, supported version to fix this vulnerability.

Rapid7's Project Sonar discovered just over 85,000 ASA/FTD devices, 398 of which are spread across 17% of the Fortune 500. Since it is difficult (if not impossible) to legally fingerprint Cisco ASA/FTD versions remotely, Rapid7 Labs revisited the "uptime" technique described in a 2016 blog post for another Cisco ASA vulnerability, which shows that only about 10% of affected Cisco devices have been rebooted since the release of the patch. Rebooting is a likely indicator they've been patched yet only 27 of the 398 detected in Fortune 500 companies appear to have been patched and rebooted.

In this case the vulnerable component is a WebVPN. So not exposing it to the public is not an option. And the extent of the danger appears to be information disclosure of the device's LUA files. So in this case it's not the end of the world. But this again highlights the crucial need for today's enterprises to have near real-time responses to newly announced vulnerabilities. It should be clear to everyone by now that a vulnerability is no longer a surprise exception. Just ask Microsoft on any Patch Tuesday. "Dynamic Update and Patch Management" could become a job title... and I'll bet it probably will.

Miscellany

"Bloatbox"

De-bloating Windows 10 after installation and before getting down to any serious business has become something that any serious user needs to do... unless you enjoy the idea of having your Windows Workstation turned into some sort of advertising videogame hybrid. I certainly don't.

Over time I've assembled a few tools to do this. Mostly notes about PowerShell commands with wildcards for "please remove everything." And then a few extras for things like "Connect" which is extra stubborn and requires some extra coaxing to leave. So far, the available utilities for accomplishing these tasks have left me unimpressed. And I know that the world would rather that I continue moving SpinRite along than do something that someone else will finally get around to doing right.

So, a newly released open source tool called "Bloatbox" looks like perhaps the best thing I've seen so far, and I wanted to bring it to our listener's attention. It's on Github, and just Googling "Bloatbox" will take you there: https://github.com/builtbybel/bloatbox

I have not had occasion yet to use it, so I am not vouching for it. And one concern is that it might be digging a bit too deep. In the sample screenshot I see options to remove various versions of the Microsoft.Net.Native.Framwork, the FluentXAMLThemeEditor and more. Things that probably ought to remain where they are. So I would advise caution and only remove things that are recognizable, in your face and annoying you. Still, "Bloatbox" might be worth a look. I'll be on the lookout for any Twitter feedback about it from any of our listeners who do check it out. And the next time I'm facing a start menu loaded with flippy animated tiles pushing Candy Crush Soda Saga, I'll give it a try myself.

Errata

David A. Wheeler @drdavidawheeler

Hi, you keep claiming in "Security Now" that the CVE number after the year is in sequential order. That hasn't been true for a long time. There are too many CVEs for one organization to assign them all. So there are now many "CVE Numbering Authorities" (CNAs), each of which is given a block of integers to assign. So it's not as simple as "number after year indicates order" or "indicates number of vulnerabilities this year".

SpinRite

The benchmarking software has evolved into a surprisingly accurate measure of performance. It's a bit like having access to a high-resolution microscope. And as a result we've been discovering some very interesting and surprising things:

For example, here's the result of seven separate runs of the benchmark against a system containing a 10 terabyte Seagate spinner and a half terabyte Crucial SSD.

The earliest hard disk drives maintained a 17 sectors (for MFM) or 26 sectors (for RLL) per revolution sector count. They had the same number of sectors around the innermost cylinder as the outermost. But that meant that the bit density of the bits around the inner cylinder set the bit rate for the drive and that the same number of bits were greatly spread out around the outer cylinder. That clearly wastes space. So all of today's modern drives vary the number of sectors around the track depending upon the track's length, and that varies with the tracks position on the drive. As a consequence, a modern hard drive's data transfer rate will also vary with the position of the track on the drive...

P	S	Size	Drive Identity	Location:	0	25%	50%	75%	100
3	2	10TB	ST10000DM0004-2GR11L		243.5	227.6	205.3	170.0	112.7
3	2	10TB	ST10000DM0004-2GR11L		243.7	227.6	205.3	170.0	112.7
3	2	10TB	ST10000DM0004-2GR11L		242.0	227.6	205.3	170.0	112.6
3	2	10TB	ST10000DM0004-2GR11L		244.0	227.6	205.3	169.6	112.7
3	2	10TB	ST10000DM0004-2GR11L		243.4	225.9	205.3	170.0	112.7
3	2	10TB	ST10000DM0004-2GR11L		244.1	227.6	205.3	170.0	112.7
3	2	10TB	ST10000DM0004-2GR11L		244.1	225.9	205.3	170.0	112.6

To look at this, I recently added position-dependence to the benchmark. As we can see from the table above, the benchmark determines the data transfer rate at five positions on the drive: At the front (0%), and at the 25%, 50%, 75%, and 100% locations. As we would predict, the actual data transfer rate drops off as we move in. In the case of this 10 terabyte Seagate drive, the back of the drive runs at about 46% throughput compared to the front of the drive. This suggests that for a spinning drive, moving the most often accessed data to the front can more than double the drive's actual throughput compared with data located at the end of the drive.

And notice something else I'm quite proud of: The remarkable run-to-run repeatability of the benchmark's results. These numbers are surprisingly rock solid — in many cases to 4 significant digits. At each location I'm performing a 1 gigabyte maximum-throughput data transfer. And I have an improvement in mind.

We also applied the same test to SSDs and many of us have discovered something surprising: The most highly used front-ends of many of our SSDs are markedly slower than the lesser used regions farther in...

P	S	Size	Drive Identity	Location:	0	25%	50%	75%	100
0	3	512GB	Crucial_CT512MX100SSD1		493.5	521.6	535.1	538.4	530.2
0	3	512GB	Crucial_CT512MX100SSD1		493.6	521.6	535.1	538.4	530.2
0	3	512GB	Crucial_CT512MX100SSD1		493.7	521.6	535.1	538.6	530.3
0	3	512GB	Crucial_CT512MX100SSD1		493.6	521.6	535.1	538.4	530.2
0	3	512GB	Crucial_CT512MX100SSD1		493.6	521.6	535.1	538.4	530.2
0	3	512GB	Crucial_CT512MX100SSD1		493.6	521.6	535.1	538.6	530.3
0	3	512GB	Crucial_CT512MX100SSD1		493.7	521.6	535.1	538.7	530.4

In the 512GB Crucial SSD above it's clear that the front of the drive is slower than the rest. This might be caused by the need for more error correction at the front of the drive through repeated

use. Or its cells might be getting fatigued despite the presence of wear leveling. We know that wear leveling is employed to average the write fatigue of SSDs, but it might be that wear leveling only functions on a relatively local scope and not globally. So the front of an SSD might still be aging at a faster rate than the back of the drive which is being used much less often, if ever.

And we've also seen something else. Now that we know that we have a highly-precise measuring instrument, we can take its results seriously. And a table like the one below is quite revealing:

P	S	Size	Drive Identity	Location:	0	25%	50%	75%	100
1	3	2TB	ST2000DM001-1CH164		161.7	187.9	172.7	142.7	92.9
1	3	2TB	ST2000DM001-1CH164		166.2	187.9	172.5	142.7	92.9
1	3	2TB	ST2000DM001-1CH164		161.6	187.9	172.7	142.7	92.9
2	2	2TB	ST2000DM001-1CH164		174.2	198.1	178.9	144.1	91.2
2	2	2TB	ST2000DM001-1CH164		174.3	198.1	178.9	144.0	91.2
2	2	2TB	ST2000DM001-1CH164		174.2	198.1	178.9	144.1	91.2
3	2	2TB	ST2000DM001-1CH164		167.5	190.7	171.4	141.0	87.4
3	2	2TB	ST2000DM001-1CH164		160.5	190.7	171.4	141.0	87.4
3	2	2TB	ST2000DM001-1CH164		167.5	190.7	171.1	141.0	87.4
4	2	2TB	ST2000DM001-1CH164		158.6	188.0	173.6	141.2	90.2
4	2	2TB	ST2000DM001-1CH164		158.7	188.0	173.6	141.2	90.2
4	2	2TB	ST2000DM001-1CH164		158.5	188.0	173.6	141.2	90.2

The table above is the result of three runs across four identical Seagate, 2-terabyte drives. You'll notice that, as usual, the speeds vary as a function of the position on the drive, and thanks to the precision of the benchmark, the same-drive performance repeats almost exactly across the three benchmark tests of each drive.

But look at the values for each drive. The drives are identical make and model, and their numbers repeat almost exactly within the drive, but the numbers from drive to drive are consistently different. Why?

What can account for the precise performance of identical drives staying consistent for each drive, but different from the others? What differs from one drive to the next? What has ALWAYS differed from one drive to the next? The number and location of physical surface defects. This benchmark is revealing the subtle transfer timing variations which result from physical sector remapping around defects. The location and number of defects differs from one drive to the next, but of course they remain fixed for any single drive.

At the moment I’m performing the benchmark as 32 consecutive back-to-back 32-megabyte transfers of 65,536 sectors each. I’ve proven that I’m eliminating all inter-transfer overhead. No revolutions are lost between blocks. So I’m streaming data off the drive at its maximum theoretical performance. And I have achieved timing resolution is down in the hundreds of picoseconds.

But I mentioned that I had an idea for improvement: Because these timing irregularities have raised some interesting questions, by next week's podcast I'll have changed the test to 33 consecutive back-to-back transfers, and I plan to snapshot the exact inter-block time so that we can see how the 32, 32-megabyte transfers flow. I'll be using 33 so that I can discard the first one. That way the benchmark will be able to eliminate any head seek time and rotational latencies from the start of each block. The benchmark won't start timing until the first discarded 32-megabyte block has been read.

As I've noted before, the idea of a mass storage benchmark started out as a bit of a Trojan horse: An inducement for our listeners to obtain some value in return for their effort of formatting and booting a USB stick to run the benchmark on their various hardware systems. In the process of doing that, they would be verifying that SpinRite's new suite of "bare metal" no-BIOS drivers, of which this AHCI driver is the last, are working for them, and this proving that SpinRite will, too. But it looks like this hyper-accurate storage benchmark is going to wind up providing some very interesting information for its users. My original plan for a companion web forum was to help field problems. But it's clear that we're going to also need a place to discuss people's interesting findings.