Security Now! #772 - 06-23-20

Ripple20

This week on Security Now!

This week we look at Microsoft's interesting decision to update Windows 7 desktops with their new Edge browser, Google's wholesale removal of 106 widely-downloaded malicious Chrome extensions, Microsoft's continuing drama over Win10 printing, a potentially critical remote code execution vulnerability in everyone's favorite VLC media player, an interesting move by RosKomNadZor!, Netgear's residence in the Dog House, a new and startling record in DDoS attack size, a bit of errata and the anticipated announcement of a new piece of spin-off freeware from the SpinRite project. Then we examine the ripple effects of the mass adoption of an embedded TCP/IP stack that is found to be horribly insecure many years after it has been quite widely adopted across the embedded device industry.

Guess who…??

Browser News

Edge for Windows 7

Starting last Wednesday the 17th, our long-dormant Windows 7 Update suddenly sprang back to life as Microsoft began rolling our an "Important" update for their Win7 64 systems. It's KB4567409 titled: "Microsoft Edge Update for Windows 7 for x64-based Systems". It is, of course, their new Chromium-based Edge web browser.

Microsoft's support bulletin states that this update is not being offered to enterprise devices, only to users running Win7 SP1 and Win8.1 Home, Professional, Ultimate, Starter, or Core editions.

They wrote: "This update is not intended to target Enterprise devices. Specifically, this update targets devices that run Windows 7 SP1 or later versions and Windows 8.1 or later versions that are either Home, Professional, Ultimate, Starter, or Core editions. Devices that run these editions on Active Directory or Azure Active Directory domain are also excluded from this automatic update."

This update is only available via Windows Update and not being offered as a stand-alone download from the Microsoft Update Catalog.

After being installed, the following changes will be made:

- Edge will be pinned to the taskbar and add a shortcut to the desktop.
  If the current version of Edge (if any) already has a shortcut, it will be replaced.
- The new Edge will not replace IE.
- And this update will not change the system's default URL handler.

This is interesting. It suggests that Microsoft would like to have a bit of a foothold in all of those non-enterprise machines that are still stubbornly running Windows 7. The latest market share have Win10 solidly in first place at 53.74% and Windows 7 solidly in second place at 28.35%. After that, 3rd is MacOS X v10.14 at 3.84% then Windows 8.1 at 3.65%. So together Win10 plus Win7 total 82.09% of all desktops.

I can see why Microsoft might not be willing to concede the desktop browser to Chrome when it likely feels that it finally has something -- thanks to Chromium, of course -- that's just as good.

Google removed 106 malicious Chrome extensions collecting sensitive user data.

The well-named cyber-security firm "Awake Security" identified the extensions as malicious in a report they published last Thursday titled: "The Internet's New Arms Dealers: Malicious Domain Registrars":
https://awakesecurity.com/white-papers/the-internets-new-arms-dealers-malicious-domain-registrars/

Awake explains that their report dives into the results of a multi-month investigation that uncovered a massive global surveillance campaign affecting millions of users. The campaign involved thousands of domains and more than one hundred malicious Chrome extensions with all

the activity tying back to a single internet domain registrar: Gal Communication (CommuniGal) Ltd (GalComm).

This campaign and the Chrome extensions included operations such as taking screenshots of the victim device, loading malware, reading the clipboard, and actively harvesting tokens and user input. In the wake of Awake's disclosure, Google has taken down the malicious extensions. However, this campaign was able to avoid detection by state-of-the-art security tools through a number of evasion schemes.

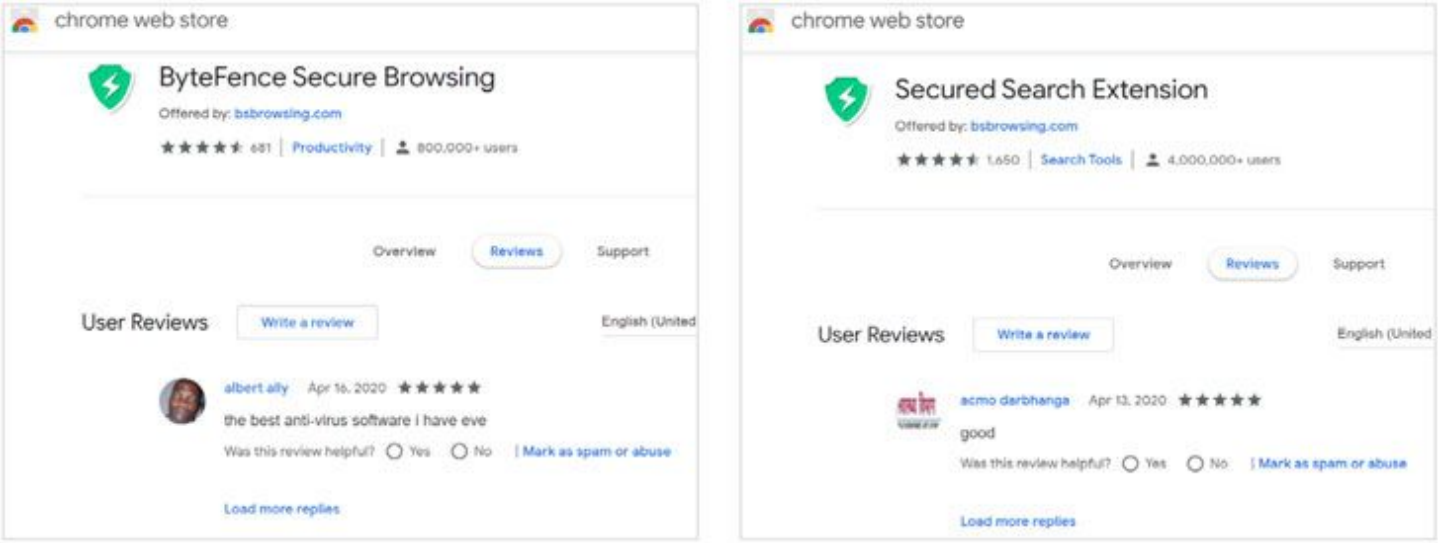
The report highlights that:

The attacker’s infrastructure including 15,160 malicious/suspicious domains and 111 malicious or fake Chrome extensions with approximately 33 million downloads
Connections between this campaign and a number of traditional malware families
The methods the attacker used to avoid detection by sandboxes, endpoint detection and response systems, web proxies and more.
And how Awake connected the dots all leading to a single domain registrar.

Awake said that the extensions posed as tools to improve web searches, convert files between different formats, perform security scan, and more. But the extensions contained code to bypass Google's Chrome Web Store security scans, take screenshots, read the clipboard, harvest authentication cookies, and snag user keystrokes including passwords.

Awake believes all of the extensions were created by the same entity though the company has not identified it yet. The primary connection between all the extensions was that they sent user data back to domains registered through the GalComm domain registrar.

Furthermore, Awake says that many extensions appeared to share the same graphics and codebase, with slight changes. This similarity made scanning for their siblings much easier. Some of the extensions even shared version numbers and the descriptions... So someone got lazy about covering their tracks.



The security-related extensions typically had hundreds to thousands of fake reviews.

And Awake said that by last month, when it reached out to Google, the 111 malicious extensions they had identified had been downloaded 32,962,951 times. (I suppose "If you build it they will come" is a truism.)

Based on internal telemetry, Awake says that some of these extensions have been found on the networks of "financial services, oil and gas, media and entertainment, healthcare and pharmaceuticals, retail, high-tech, higher education and government organizations." They are effectively acting as backdoors into private networks and espionage tools. Though there's no evidence to suggest they've been used as such, the capability is there.

Google has proactively deactivated 106 of the 111 identified Chrome extensions in every user's browser. The extensions are still installed, but disabled and marked as "malware" in the Chrome browser's extension section. It would be a little bit sobering to find that flag lit up. Go to "chrome://extensions" if you're curious to see whether you might have any of the 106 extensions installed.

And this is a good moment to just mention that add-ons of all types are the new front in the war on consumer trust. We know that Apple and Google invest heavily in the safety and security of the 3rd-party apps and add-ons they curate in their various app stores. But, nevertheless, stuff is going to slip past. Many people choose a Android device because they are typically less expensive and often because of the greater user-freedom that comes with a more open platform. But that same openness is available to the apps the user trusts to be safe.

The bottom line is that it's a numbers game. The chance that any one app might be malicious is likely small. But every additional app Orr add-on installed increases the probability that one of them is malicious. So, be a bit prudent and also don't let the crap accumulate. Occasionally page through what you have downloaded and installed and remove those things you no longer use. And afterward, reboot the device.

Security News

Zoom’s Encryption Story… Take III:
https://blog.zoom.us/wordpress/2020/06/17/end-to-end-encryption-update/

Last Wednesday, in an apparent effort to deal with his company’s recent spate of confused mixed-messages regarding Zoom’s plans for enforcing truly secure end-to-end encrypted teleconferences, Eric Yuan attempted to clean things up in a blog.

Eric wrote exactly the following:

> Since releasing the draft design of Zoom’s end-to-end encryption (E2EE) on May 22, we have engaged with civil liberties organizations, our CISO council, child safety advocates, encryption experts, government representatives, our own users, and others to gather their feedback on this feature. We have also explored new technologies to enable us to offer E2EE to all tiers of users.
>
> Today, Zoom released an updated E2EE design on GitHub. We are also pleased to share that we have identified a path forward that balances the legitimate right of all users to privacy and

the safety of users on our platform. This will enable us to offer E2EE as an advanced add-on feature for all of our users around the globe – free and paid – while maintaining the ability to prevent and fight abuse on our platform.

To make this possible, Free/Basic users seeking access to E2EE will participate in a one-time process that will prompt the user for additional pieces of information, such as verifying a phone number via a text message. Many leading companies perform similar steps on account creation to reduce the mass creation of abusive accounts. We are confident that by implementing risk-based authentication, in combination with our current mix of tools — including our Report a User function — we can continue to prevent and fight abuse.
Additional Information

- We plan to begin early beta of the E2EE feature in July 2020.
- All Zoom users will continue to use AES 256 GCM transport encryption as the default encryption, one of the strongest encryption standards in use today.
- E2EE will be an optional feature as it limits some meeting functionality, such as the ability to include traditional PSTN phone lines or SIP/H.323 hardware conference room systems. Hosts will toggle E2EE on or off on a per-meeting basis.
- Account administrators can enable and disable E2EE at the account and group level.

We are grateful to those who have provided their input on our E2EE design, both technical and philosophical. We encourage everyone to continue to share their views throughout this complex, ongoing process.

So the news, here, is that free/basic tier users -- presumably meeting hosts -- will have the ability to host secure E2EE Zoom teleconferences after first providing some level of deanonymizing authentication for their account. And once a meeting's encryption "cone of silence" has been activated and lowered over the conference, it cannot be withdrawn, thus protecting the meeting's participants from a bait and switch. To me, that sounds like a fair and equitable tradeoff, though at this point I will never trust Zoom again. After their initial out-of-the-gate stumbles they had one chance at redemption. Yet, as we know, through a bizarre lack of messaging coherence they totally screwed the pooch and, I think, lost any hope for ever convincing those who most have a need for easy-to-use encrypted "warrant proof" conferencing that Zoom is the place to get it. And you know, it's interesting that we have no similar concern about Apple. Apple has played the encryption issue **exactly** right from day one: Everyone knows that Apple's products are as absolutely and truly secure as they are capable of making them. Any mistakes that they or others find are immediately fixed. And Apple's very public fights with US law enforcement have only proven to enhance that well deserved reputation.

That said, and as has been observed in many of Leo's recent discussions of Zoom's encryption across TWiT, it's unclear how much Zoom's style of teleconferencing really needs uber-secure end-to-end encryption anyway. Zoom's post-COVID usage explosion is almost entirely replacing meetings such as yoga and classrooms that were publicly accessible and never particularly secure in the first place. So Zoom's quick addition of the platform's various new attendee management features was what was most needed and the world now has that. Zoom's encryption is nice for the prevention of casual eavesdropping, but someone would be a fool to trust its strength against a duly issued government law enforcement subpoena.

Windows 10 printing... not so much
Last week we covered some of the many problems people were experiencing after installing Windows 10's June updates. Among them was the "printer not turned on before starting windows" and the more significant "printing no longer works at all after the updates". As we were podcasting Microsoft was acknowledging some printing problems and issuing an out-of-cycle emergency update to fix ONE of the several problems... but, apparently and incredibly not either of the problems we talked about. This one is a crash in the print spooler that also prevents printing.

In the Windows Message Center Microsoft says: "An out-of-band optional update is now available on the Microsoft Update Catalog to address a known issue in which certain printers may be unable to print after installing updates released June 9, 2020."

The announcement noted that the issue could cause the print spooler to "generate an error or close unexpectedly when attempting to print, and no output will come from the affected printer."

And you "might also encounter issues with the apps you are attempting to print from, such as receiving an error, or the app might close unexpectedly," and the issue "might also affect software-based printers, such as when printing to PDF."

It's unclear to me what users who are not proactive are expected to do, because Microsoft recommends users to install the one of the three consecutively numbered updates: KB4567512, KB4567513, and KB4567514 which are flagged as "optional cumulative updates" for Windows 10 versions 1909, 1903, 1809, and 1803, respectively... but only if their devices are affected by this issue. They noted that "These optional Windows 10 updates are not available from Windows Update and will not install automatically."

And then since they broke printing across ALL versions of Windows, two days later, last Thursday, Microsoft dropped another collection of optional cumulative updates to address the printing issue on Windows 10 versions 2004, 1709, 1703, 1607, and 1507, as well as on Windows 8.1, Windows Server 2012, and Windows Server 2016. Wow.

But still nothing about the “be sure to turn the printer on first” problem.

And these things being optional where the affected user must go get the fix is puzzling to me. It sure seems that if Microsoft is going to automatically break someone’s Windows 10 machine they really should also automatically fix what they’ve broken, especially when they have given users no choice about whether to accept and install the forcible breakage in the first place. But that’s just me.

VLC Media Player 3.0.11 fixes severe remote code execution flaw
VideoLan has released VLC Media Player v3.0.11 for the desktop (Windows, Mac, and Linux). In addition to some random bug fixes and improvements, the main reason to update the next time you run VLC, if not now, is that this release fixes a security vulnerability that could allow attackers to remotely execute commands.

Following our theme of “interpreters are hard to make perfect and unfortunately perfection is

required", this vulnerability, tracked as CVE-2020-13428, is a "buffer overflow in VLC's H26X packetizer" which, if exploited, would allow attackers to execute commands under the same security level as the user.

So the good news is that it's running as a client in the user's account. The bad news is, you still really don't want to be running someone else's deliberately malicious code in your computer. According to their security bulletin, this vulnerability can be exploited by creating a specially crafted file and tricking a user into opening it with VLC.

While VideoLan states that this vulnerability will most likely just crash the player, they warn that it could be used by an attacker to execute commands under the security level of the user remotely. As we know, crashes are where exploits are born.

So... due to the potential severity of this vulnerability and the public disclosure of the problematic code, everyone should update their VLC to install version 3.0.11. Taking my own advice, I launched my copy and saw that I was running v3.0.8 and when I launched it a dialog was presented stating that a newer version (3.0.11) was available. The dialog added that: "VideoLAN and the VLC development team present VLC 3.0.11 "Vetinari". VLC 3.0.11 is a small update to VLC 3.0 branch, improving support for HLS, aac, Youtube, AV1 and fixes a minor security issue." (with the word "security" in red, which was a nice touch.) The update was 40 megs and definitely worthwhile.

An update from Roskomnadzor!
Roskomnadzor, which is the bureau serving as Russia's media watchdog, said that Telegram has agreed to help Russian law enforcement fight against extremist and terrorist content shared on its platform. http://rkn.gov.ru/news/rsoc/news73050.htm

So, the Russian government has lifted its largely ineffective 2-year ban on Telegram's instant messaging service. In a message posted on its website, Roskomnadzor said it lifted the ban after Russian prosecutors reached an agreement with Telegram's founder, Pavel Durov. Russian officials said Durov "expressed readiness to counter terrorism and extremism" content shared on his platform. However, the details about the collaboration between Telegram and Russian officials have not been made public at the time of writing.

As we know, Russia officially banned Telegram a little over two years ago, on April 13, 2018. The ban followed Telegram's refusal to cooperate with Russia's FSB Federal Security Bureau which is Russia's primary intelligence service. At the time, FSB investigators tried to obtain encryption keys from Telegram to decrypt conversations between two suspects that were under investigation in the 2017 Saint Petersburg metro bombing.

When Telegram refused to cooperate, the FSB filed a lawsuit, which (surprise!) it eventually won in the Russian Supreme Court in early 2018. Russian officials initially fined Telegram, but the Russian courts also ordered Roskomnadzor to ban the app inside Russia after Telegram continued to refuse to cooperate.

However, Roskomnadzor had had a difficult time enforcing the ban over the past two years. Telegram constantly changed its servers' IP addresses and also employed a technique known as

“domain fronting” to bypass the ban and allow Russian users to continue using its service.

And recall that in one famous (or infamous) botched effort to ban the service in Russia, Roskomnadzor applied a wide area blanket block on more than 19 million Amazon and Google Cloud IP addresses, blocking out countless legitimate services inside Russia, such as all of Google's services, online games, banking sites, cryptocurrency exchanges, and mobile apps. Russia also banned more than 50 VPN and proxy services that Russians were using to access the service. Throughout all of this, Telegram remained extremely popular in Russia, and despite the ban was often used by Russian politicians, with officials trusting the app to keep their conversations safe from FSB surveillance.

A few days prior to Roskomnadzor lifting the ban, the Russian news site Znak reported that Russian members of parliament had introduced a bill to have the app unbanned, though it's unclear whether the bill played any role in Roskomnadzor lifting the ban.

Doing a bit more digging I discovered that in April 2020, the Government of Russia started using the blocked Telegram platform to spread information related to COVID-19 outbreak. So perhaps that factored into the change?

I also found that Roskomnadzor had not indicated how the two organisations had been able to overcome the issue of Telegram's end-to-end encryption. It did not say whether it now had access to messages, or whether changes had been made to the platform. And neither Telegram and Pavel Durov – who regularly use Telegram to communicate with their users – have yet publicly commented on the lifting of the ban.

My own feeling, reading between the lines, is that the Russian government was secretly regretting their opposition to Telegram -- especially considering that it’s the instant messaging platform of choice for many of them -- and that perhaps communications during COVID-19 played some part in it. And also the fact that they weren’t actually able to block it.

Netgear in the doghouse

There have been various other reports of random routers, even bunches of routers, having exploitable problems. But those reports haven't risen to the level required to raise an alarm because their problems are invariably LAN-side issues rather than WAN-side. And while LAN-side issues are not nothing, they do not directly expose the router to external attack. They are only vulnerable if an attacker has already gotten themselves inside, behind the router, in the LAN and onto its network. And once any attacker has accomplished that it's pretty much the case that all bets are off.

However, when a router is weak enough on the inside, there is one troubling case where an equally weak attacker might be strong enough, and that's when a user browses to a malicious website.

When that happens, the attacker's code is running in the visitor's browser which exists on the LAN. We were recently talking about how JavaScript code was able to launch its own HTTP queries using so-called "AJAX" primitives. AJAX is the abbreviation of: “**A**synchronous **J**avaScript **A**nd **X**ML.”

These 79 Netgear models are **so** vulnerable (and we'll see in detail why in a moment) that just surfing to a malicious website could allow the code the browser downloads to blindly connect to the network's local Netgear router and cause it to open a telnet session, port and command prompt as root.

So let's step back a bit and start at the beginning:

An unpatched zero-day vulnerability exists in 79 Netgear router models. The vulnerability allows an attacker to take full control over any of 79 Netgear devices from within the LAN... Even from code running inside a user's web browser.

The vulnerability, which was independently discovered by two researchers, exists in the HTTPD daemon used to manage the router. In other words, in the router's web-based router management server. One of the two researchers released a detailed explanation of the vulnerability, a PoC exploit, and scripts to find vulnerable routers... So the hackers of the world are already clued in.

https://blog.grimm-co.com/2020/06/soho-device-exploitation.html
https://github.com/grimm-co/NotQuite0DayFriday/blob/master/2020.06.15-netgear/exploit.py
https://github.com/grimm-co/NotQuite0DayFriday/tree/master/2020.06.15-netgear/tools

Being a fully detailed technical blog, the full posting is quite detailed, so I'm going to excerpt from the full blog to hit the most interesting and important highlights:

Netgear R7000

SOHO Device Exploitation

After a long day of hard research, it's fun to relax, kick back, and do something easy. While modern software development processes have vastly improved the quality of commercial software as compared to 10-15 years ago, consumer network devices have largely been left behind. Thus, when it's time for some quick fun and a nice confidence boost, I like to analyze Small Office/Home Office (SOHO) devices. This blog describes one such session of auditing the Netgear R7000 router, analyzing the resulting vulnerability, and the exploit development process that followed. The write-up and code for the vulnerability described in this blog post can be found in our NotQuite0DayFriday repository.

The first step when analyzing a SOHO device is to obtain the firmware. Thankfully, Netgear's support website hosts all of the firmwares for the R7000. The Netgear R7000 version 1.0.9.88 firmware used in this blog post can be downloaded from this website. After unzipping the firmware, we'll use binwalk to extract the root filesystem from the firmware image.

While the router may have many services worth analyzing, the web server is often the most likely to contain vulnerabilities. [And I'll add that it's also always listening with a port open to the LAN.] In SOHO devices like the R7000, the web server must parse user input from the network and run complex CGI functions that use that input. Furthermore, the web server is written in C

and has had very little testing, and thus it is often vulnerable to trivial memory corruption bugs. As such, I decided to start by analyzing the web server, httpd.

As we're interested in how the web server (mis)handles user input, the logical place to begin analyzing the web server is the recv function. The recv function is used to retrieve the user input from a connection. Thus by looking at the references to the recv function in the web server, we can see where the user input begins. The web server has two helper functions which call recv, one used in the http parser and one used to read the responses from Dynamic DNS requests to oemdns.com. We'll focus on the former use, as shown below in the Hex-Rays decompiler.

[Now, bear with me here for just a moment. Unfortunately, this won't take long...]

After the call to read_content (the recv helper function), the parser does some error checking, combines the received content with any previously received content, and then looks for the strings name="mtenFWUpload" and "\r\n\r\n" in the user input. If the user input contains these strings, the rest of the user input after these strings is passed to the abCheckBoardID function. Grepping the firmware's root file system, we can see that the string mtenFWUpload is referenced from the files www/UPG_upgrade.htm and www/Modem_upgrade.htm, and thus we can conclude that this is part of the router's upgrade functionality.

In his blob posting, Adam then has the subheading: "1996 Called, They Want Their Vulnerability Back"

Following the user input, we'll next look at the abCheckBoardID function. This function, shown below, expects the user input to be the firmware file for the R7000. It parses the user input to validate the magic value (bytes 0-3), obtains the header size (bytes 4-7) and checksum (bytes 36-49), and then copies the header to a stack buffer. This copy, performed via the memcpy function, uses the size specified in the user input. As such, it's trivial to overflow the stack buffer.

[diagram showing the code]

In most modern software, this vulnerability would be unexploitable. Modern software typically contains stack cookies which would prevent exploitation. However, the R7000 does not use stack cookies. In fact, of all of the Netgear products which share a common codebase, only the D8500 firmware version 1.0.3.29 and the R6300v2 firmware versions 1.0.4.12-1.0.4.20 use stack cookies. However, later versions of the D8500 and R6300v2 stopped using stack cookies, making this vulnerability once again exploitable [against them as well]. This is just one more example of how SOHO device security has fallen behind as compared to other modern software.

In addition to lacking stack cookies, the web server is also not compiled as a Position-independent Executable (PIE), and thus cannot take full advantage of ASLR. As such, it's trivial to find a ROP gadget within the httpd binary, such as the one shown below, that will call system with a command taken from the overflown stack.

```
MOV             R0, SP  ; command
BL              system
```

The exploit in GRIMM's NotQuite0DayFriday repository uses this gadget to start the telnet daemon as root listening on TCP port 8888 and not requiring a password to login.

And...

As the vulnerability occurs before the Cross-Site Request Forgery (CSRF) token is checked, this exploit can also be served via a CSRF attack. If a user with a vulnerable router browses to a malicious website, that website could exploit the user's router. The developed exploit demonstrates this ability by serving an html page which sends an AJAX request containing the exploit to the target device. However, as the CSRF web page cannot read any responses from the target server, it is not possible to remotely fingerprint the device. So the attacker must know the model and version that they are exploiting.

Many SOHO devices share a common software base, especially among devices created by the same manufacturer. As such, a vulnerability in one device can normally be found in similar devices by the same manufacturer. In this particular case, I was able to identify 79 different Netgear devices and 758 firmware images that included a vulnerable copy of the web server. This vulnerability affects firmwares as early as 2007 (WGT624v4, version 2.0.6). Given the large number of firmware images, manually finding the appropriate gadgets is infeasible. Rather, this is a good opportunity to automate gadget detection.

[and he goes on... but everyone gets the idea.]

Adam notes at the end of his post that on 6/15/2020 (last Monday), Vietnam's ZDI (the Zero Day Initiative group) published an advisory by d4rkn3ss from VNPT ISC on this vulnerability. Adam's Grimm group discovered the issue independently and reported the vulnerability directly to Netgear on 5/7/2020.

Obviously, anyone owning and using a Netgear router of any model should start checking in with Netgear for news of new firmware for their particular router.

I've been assuming through all of this that NO ONE would be crazy enough to have enabled web-based Remote Administration on their router's WAN-side interface. If by some chance you have you must IMMEDIATELY disable it. There could be no better demonstration case for why enabling WAN admin is a bad idea. This is a ZERO AUTHENTICATION attack on an extremely insecure web server.

At the time of this podcast Netgear has only addressed this problem for only 8 of their 79 vulnerable router models. Their advisory was first published last Thursday the 18th and they have since updated it three times as they prepare and release additional firmware updates. I'm quote certain this is a nightmare for them and that they are doing everything possible to get their firmware fixed and released.

To make finding the advisory easy for everyone I have created a GRC shortcut:
https://grc.sc/netgear
https://kb.netgear.com/000061982/Security-Advisory-for-Multiple-Vulnerabilities-on-Some-Routers-Mobile-Routers-Modems-Gateways-and-Extenders

I have a Netgear cable modem that I love, and an Asus WiFi router. But the Netgear cable modem is outboard of my pfSense firewall router and the Asus is inboard serving not as a router but only as a WiFi access point. Having that professional grade pfSense firewall provides a great deal of peace of mind.

DDoS is alive and well... and growing...

We haven't touched on DDoS attacks much recently, because there hasn't been much news. But breaking a record is always newsworthy, and the record for the largest sustained DDoS attack was recently broken.

The 2nd-highest previous record was set at a whopping 1.3 Terabits per second -- that 1.3 TRILLION bits per second -- that's 1.3 thousand billion bits per second -- which hit GitHub back in February of 2018. But that was topped a month later at 1.7 Terabits per second, aimed at Netscout in March of 2018. Both the Netscout and GitHub DDoS attacks abused internet-exposed Memcached servers to achieve their massive bandwidths.

But now, although they didn't advertise it at the time, in an incident quietly disclosed in its AWS Shield Threat Landscape report, Amazon's AWS service disclosed that they had successfully mitigated the largest ever DDoS recorded, weighing in at 2.3 Terabits per second.

Amazon's report did not disclose the intended target, but they did indicate that the attack was carried out using hijacked CLDAP web servers, resulting in three days of "elevated threat" for its AWS Shield staff.

CLDAP (Connection-less Lightweight Directory Access Protocol) which is an alternative to the older LDAP protocol used to connect, search, and modify Internet-shared directories.

CLDAP has been abused for DDoS attacks since late 2016, and CLDAP servers are known to amplify DDoS traffic by 56 to 70 times its initial size, making it a highly sought-after protocol and a common option provided by DDoS-for-hire services.

Attacks this large are rare and surprising to those running attack mitigation services. There are far more much smaller attacks happening pretty much continuously. Cloudflare, who mitigated a 550 gigabytes per second attack during the first quarter of 2020 noted that 92% of all the DDoS attacks it mitigated during that same first quarter of 2020 were under 10 Gigabits per second. In other words, only 8% topped 10 gigs per second. And they also noted that 47% of all attacks were even smaller, under 500 Mbps.

As we know, DoS and DDoS are one of the consequences of the Internet's autonomous packet routing system which has served us so well from the start. So long as it's possible to query a remotely located public server with a UDP packet that does not require TCP's round-trip, and if that query's source IP can be spoofed without being blocked on its way out onto the public Internet, and if the remote server's reply to the query is much larger than the query, amplified DDoS attacks are going to be a feature of our global internetwork of networks.

Errata

"My how times flies"

Leo, our off the cuff (and inaccurate) discussion of how long we've been at this podcast generated some conversation over in GRC's "Security Now" newsgroup. The upshot of which was someone named Rob Allen summarizing our current position quite succinctly:

Rob wrote: *"We are currently in the 15th season, or Year 15, of the show, though it has only been 14 years and 10 months since August 18, 2005. On August 17th, the show will have been on the air for 15 full years, making August 11 the final episode of year/season 15. Season/Year 16 will start August 18, 2020 with episode 780."*

So there we have it. We've been doing this for two months shy of 15 years.

The gang over there mentioned "timeanddate.com" which is a pretty slick site for performing date math. I did a bit of quick computation about when our FINAL podcast would occur. Since we're at 772, and we run out of digits at 999, that's 227 remaining podcasts to go. At one podcast per week that's 1589 days, or 4.35 years. But since we'll have some holiday best-of's, that'll extend the end by another four weeks since we'll cross four holiday events. So around 1617 days from now... which timeanddate.com places at around:

Tuesday, November 26, 2024 ... which really isn't that far away!

SpinRite

Announcing: "**InitDisk**" / Google: "grc initdisk" / In GRC's main menu under Freeware/Utilities. Or go to: grc.com/initdisk

To aid the forthcoming testing of SpinRite's new technologies we needed a simply way to prepare a bootable USB drive. Because I always start from scratch at the beginning and build up from there, we would up with a uniquely capable new utility. During our testing a number of those testing found that USB thumb drives they had long believed to have died and be dead were immediately brought back to life by InitDisk. I encountered another report just this morning, posted by a frequent contributor who goes by "Obiwan":

> Tried it on a rather old TDK TF10 (8GB) USB stick which was "dead", that is, windows was unable to recognize it, at best it was recognized but with a very small capacity; found it inside a closet, not sure how long it was there, at any rate, downloaded and started initdisk, upon startup it asked me to insert the device, did so by inserting the "dead" key, initdisk recognized it, so I went on with the "NUKE" and at end, had the 8GB stick alive again :D Not bad, I think. Further checks performed using various tools reported it to be "ok", so... well, at least that "initdisk" tool may be useful to recover (to some extent btw, not pretending miracles) some USB sticks :)

I don't think that Obiwan had been following along with the groups work, so this ability of InitDisk to possibly bring dead devices back to life, which the group had already encountered several times, was likely news to him. :) So, the industry has a new bit of freeware from GRC.

It's the first of the widely public offshoots from the SpinRite work. The next thing, will be a very cool "bare metal" bootable mass storage device benchmarking tool. Given the fact that GRC's DNS Benchmark remains our #1 most downloaded utility (now at nearly 5.8 million downloads and at a rate of 3 thousand new downloads per day) I hope that this forthcoming mass storage benchmark will also develop a following.

The method to my madness, here, is that everyone who uses it will also be simultaneously testing and verifying the new hardware driver suite that will be moved into SpinRite. So the more early testing we can get of that, the better. And also, anyone wishing to verify that the next and all future SpinRites will work for them, on whatever hardware they have, will be able to use this free benchmark -- which utilizes all of the same technology --- to make sure. And if not, I want to know so that I can fix it now. So I'll also be launching a new set of GRC forums to make reporting and managing any problems much easier for the benchmark's users. :)

Ripple20

How much damage can a little 2-person company situated in Cincinnati, Ohio do the world? The answer to that question was likely revised this week.

The company "Treck, Inc." has a very up-to-date looking web site at: https://treck.com/. The site's homepage declares: "Since 1997 Treck has been designing, distributing and supporting real-time embedded internet protocols for worldwide technology leaders. The Treck TCP/IP stack designer and Treck co-founder has more than 20 years experience and is a leading expert of embedded internet protocols."

As we know, anyone can be forgiven for making a mistake. And, arguably, someone should probably have taken a good hard look at these offerings before allowing 23 years of embedded device adoption to have occurred. But now, is when we are, and as "The Hacker News" summed it up in their headline: "New Ripple20 Flaws Put Billions of Internet-Connected Devices at Risk of Hacking."

The Hacker News wrote:

Dubbed "Ripple20," the set of 19 vulnerabilities resides in a low-level TCP/IP software library developed by Treck, which, if weaponized, could let remote attackers gain complete control over targeted devices—without requiring any user interaction.

According to Israeli cybersecurity company JSOF—who discovered these flaws—the affected devices are in use across various industries, ranging from home/consumer devices to medical, healthcare, data centers, enterprises, telecom, oil, gas, nuclear, transportation, and many others across critical infrastructure.

https://www.jsof-tech.com/ripple20/

Switching now to JSOF's summary and some details...

The JSOF research lab has discovered a series of zero-day vulnerabilities in a widely used low-level TCP/IP software library developed by Treck, Inc. The 19 vulnerabilities, given the name Ripple20, affect hundreds of millions of devices (or more) and include multiple remote code execution vulnerabilities. The risks inherent in this situation are high. Just a few examples: data could be stolen off of a printer, an infusion pump behavior changed, or industrial control devices could be made to malfunction. An attacker could hide malicious code within embedded devices for years. One of the vulnerabilities could enable entry from outside into the network boundaries; and this is only a small taste of the potential risks.

The interesting thing about Ripple20 is the incredible extent of its impact, magnified by the supply chain factor. The wide-spread dissemination of the software library (and its internal vulnerabilities) was a natural consequence of the supply chain "ripple-effect". A single vulnerable component, though it may be relatively small in and of itself, can ripple outward to impact a wide range of industries, applications, companies, and people.

Ripple20 reached critical IoT devices from a wide range of fields, involving a diverse group of vendors. Affected vendors range from one-person boutique shops to Fortune 500 multinational corporations, including Cisco, HP, EMC, GE, Broadcom, NVIDIA, Schneider Electric, Intel, Rockwell Automation, Caterpillar, Baxter, as well as many other major international vendors suspected of being vulnerable in medical, transportation, industrial control, enterprise, energy (oil/gas), telecom, retail and commerce, and other industries.

Ripple20 is a set of 19 vulnerabilities found on the Treck TCP/IP stack . Four of the Ripple20 vulnerabilities are rated critical, with CVSS scores over 9 and enable Remote Code Execution. One of the critical vulnerabilities is in the DNS protocol and may potentially be exploitable by a sophisticated attacker over the internet, from outside the network boundaries, even on devices that are not connected to the internet.

A second Whitepaper, to be released following BlackHat USA 2020 will be detailing the exploitation of CVE-2020-11901, a DNS vulnerability, on a Schneider Electric APC UPS device. The other 15 vulnerabilities are in ranging degrees of severity with CVSS score ranging from 3.1 to 8.2, and effects ranging from Denial of Service to potential Remote Code Execution.

Most of the vulnerabilities are true Zero-days, with 4 of them having been closed over the years as part of routine code changes, but remained open in some of the affected devices (3 lower severity, 1 higher). Many of the vulnerabilities have several variants due to the Stack configurability and code changes over the years.

Ripple20 are the only vulnerabilities reported in Treck to date as far as we know, except for some general logical vulnerabilities referenced in the past which pertained to many stack implementations and usually had to do with RFC misinterpretations or deprecated RFCs.

Ripple20 vulnerabilities are unique both in their widespread effect and impact due to supply chain effect and being vulnerabilities allowing attackers to bypass NAT and firewalls and take control of devices undetected, with no user interaction required. This is due to the vulnerabilities being in a low level TCP/IP stack, and the fact that for many of the vulnerabilities, the packets sent are very similar to valid packets, or, in some cases are completely valid packets. This enables the attack to pass as legitimate traffic.

To give some sense for what has just happened here are the six vulnerabilities with the highest severities rated from 10 down to 9 (which is still "house on fire")...

- CVE-2020-11896 (severity score 10.0): Improper handling of length parameter inconsistency in IPv4/UDP component when handling a packet sent by an unauthorized network attacker. This vulnerability may result in remote code execution.

- CVE-2020-11897 (severity score 10.0): Improper handling of length parameter inconsistency in IPv6 component when handling a packet sent by an unauthorized network attacker. This vulnerability may result in possible out-of-bounds write.

- CVE-2020-11898 (severity score 9.8): Improper handling of length parameter inconsistency in IPv4/ICMPv4 component when handling a packet sent by an unauthorized network attacker. This vulnerability may result in the exposure of sensitive information.

- CVE-2020-11899 (severity score 9.8): Improper input validation in the IPv6 component when handling a packet sent by an unauthorized network attacker. This vulnerability may allow exposure of sensitive information.

- CVE-2020-11900 (severity score of 9.3): Possible double free in IPv4 tunneling component when handling a packet sent by a network attacker. This vulnerability may result in remote code execution.

- CVE-2020-11901 (severity score 9.0): Improper input validation in the DNS resolver component when handling a packet sent by an unauthorized network attacker. This vulnerability may result in remote code execution.

JSOF has attempted to be as responsible as possible. JSOF said that their "disclosure had been postponed twice as pleas for more time came from some of the many affected vendors, with some of the vendors voicing COVID-19-related delays. Out of consideration for these companies, the time period was extended from 90 to over 120 days. Even so, some of the participating companies became difficult to deal with, as they made extra demands, and some, from our perspective, seemed much more concerned with their brand's image than with patching the vulnerabilities."

We know that hundreds of millions of affected devices scattered all across the globe will never receive security patch updates to address these critical Ripple20 vulnerabilities. We are really in a world of hurt. In some despair, and not knowing what else to recommend, ICS-CERT recommended consumers and organization to:

- Minimize network exposure for all control system devices and/or systems, and ensure that they are not accessible from the Internet.
- Locate control system networks and remote devices behind firewalls and isolate them from the business network.

Besides this, it's also advised to use virtual private networks for securely connecting your devices to Cloud-based services over the Internet.

We talked last week about the dangers inherent in having a technological monoculture. Here, we have a highly defective TCP/IP networking library that has been on the market for more than two decades, riddled with just-discovered critical vuilnerabilities. There is just no way to update most of the affected devices. And most users will never even be aware that anything like this has happened. Yet their networks, and the network security they depend upon, has just taken a huge hit. You can absolutely bet that state level and other hackers for hire are rubbing their hands together. Lists of affected hardware is being assembled at this moment.

The Internet's already target-rich environment
just got a whole lot richer.