Security Now! #758 - 03-17-20

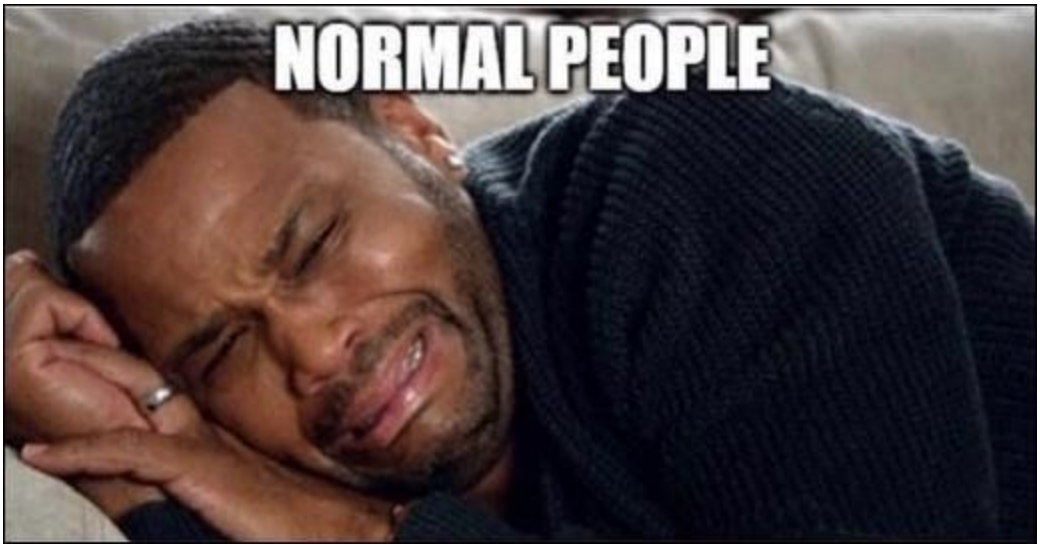
The SMBGhost Fiasco

This week on Security Now!

This week we take a deep dive into the many repercussions preceding and following last week's patch tuesday (Wouldn't it be nice to have a quiet one for a change?) But first we look at a nice list of free services being maintained by BleepingComputer's Lawrence Abrams, we look at a recent report into the state of open source software vulnerabilities, and at new and truly despicable legislation aimed at forcing social media companies to provide "lawful access" to their customers' encrypted content.

Coronavirus Lockdown Rules:

*Do **NOT** travel. Do **NOT** socialize. Remain inside!*



Security News

Patch Tuesday Redux

So far this year, every single Patch Tuesday (okay, there've only been three of them so far) has resulted in a flurry of aftereffect scrambling of one sort of another. Last week's Patch Tuesday did not break the pattern. It seems that, increasingly, as Microsoft works to fix one problem another one springs up. I was immediately put in mind of one of the famous Three Stooges episode titled "A Plumbing We Will Go"...

The Three Stooges - "A Plumbing We Will Go"
https://www.youtube.com/watch?v=wpIdOgxQSpg

There's a famous scene were curly (who, of course, is bald) tries to close up a leaking bathtub shower head by adding a length of pipe to it. He's initially quite happy with this solution since water is no longer getting loose at the wall... until he turns around and realizes that the length of pipe ended in a "T" junction and now there are two leaks. From that point, general mayhem ensues because rather than recognizing the source of his error, he assumes that what he was doing was correct, but just incomplete. Therefore, what's needed is just more of the same. Any similarly to the Windows Update process I will leave to our listeners to judge for themselves.

There was also mayhem ensuing after last week's Patch Tuesday, which we will, of course, get to momentarily. But let's first examine what Patch Tuesday brought us...

This month, Windows received updates to repair a whopping 117 vulnerabilities (or 116 or 115 depending upon who you ask). But in any event, 25 of them are rated critical, all of them enabling remote code execution and in some cases also privilege elevation. There were also 91 rated as important and a lone moderate.

The top 20 of the 25 critical vulnerabilities were most interesting.

Believe it or not, Windows is still having problems with .LNK files. (Remember Windows 95?) CVE-2020-0684 describes a remote code execution vulnerability in Windows occuring when a user opens a specially crafted and malicious .LNK file. This file could be presented to the victim on a removable drive or remote share, and when opened, would execute a malicious binary embedded in the file. So it's a sort of self-contained buffer overflow in a Windows link file. What's significant here is that since .LNK files are non-executable they are often passed over by any channel-monitoring A/V system in the interest of saving time. So a .LNK file has been, until last week, a clever way to sneak executables past network protections.

Then we had 4 memory corruption vulnerabilities in Microsoft's Media Foundation. Any of those 4 could allow an attacker to gain the ability to install programs, view, change or delete data or create new user accounts on the victim machine. Whoopsie! And, worse, a user might have run afoul of this merely by accessing a malicious file or web page. Attackers are most likely to try and exploit this vulnerability via spam emails with malicious links and attachments. I didn't dig in any further. But it sounds like 4 surviving buffer overflows in media content interpretation. We've often seen how difficult it is to get that all exactly right.

Next up -- and I had to count'em -- I counted to 10. So, half of the top 20 of the important vulnerabilities were all found in the way the Microsoft ChakraCore scripting engine -- the engine Microsoft wrote from scratch for the first attempt at its illustrious brand new Edge web browser to finally replace the creaky old Internet Exploder. In every one of these ten different instances, an attacker could successfully corrupt the victim machine's memory in a way that would allow them to execute arbitrary code in the context of the current user. Given that our web browsers ARE now the way we reach out onto the Internet and expose ourselves, I'm sure everyone is happy with Microsoft's decision to simply put their own window dressing around the open source, community developed and maintained, Chromium browser engine.

Then we had two additional critical remote code execution vulnerabilities fixed in the VBScript engine. This is not the JScript.DLL that has been an ongoing problem. An attacker could exploit these two bugs by tricking the user into visiting a specially crafted website in IE11 or by marking an ActiveX control "safe for initialization" in an application or Microsoft Office document that hosts the IE11 rendering engine. These bugs specifically require user interaction and would rely on some form of social engineering on the attacker's part. Although these two are rated critical, they at least rely upon some user interaction.... Not that that's such a high bar these days.

We wrap up the top 20 of this month's 25 critical vulnerabilities with CVE-2020-0881 and CVE-2020-0883. Also remote execution vulnerabilities in GDI+. These last two are a little trickier to exploit because two requirements must be met: An attacker would need to lure their vulnerable victim to a hostile website and secondarily induce them to open a malicious document which has been designed to exploit this vulnerability.

List of free services during Corona

Last week we took note of four companies, Microsoft, Google, Cisco and LogMeIn, who were all making their various telecommuting resources available for 90 or 180 days to help with the lifestyle changes being driven by the need for isolation to control the spread of this novel coronavirus.

BleepingComputer's Lawrence Abrams has expanded upon this. He has assembled and is maintaining a list of free services being offered to aid people whose lives have been displaced by the Coronavirus pandemic.

https://www.bleepingcomputer.com/news/software/list-of-free-software-and-services-during-coronavirus-outbreak/

Lawrence wrote:

> In response to the Coronavirus (COVID-19) outbreak, many organizations are asking their employees to work remotely. This, though, brings new challenges to the workplace as users adapt to video meetings, screen sharing, and the use of remote collaboration tools.
>
> To assist a new wave of remote works and get some publicity at the same time, many software developers and service providers have started to offer free licenses or enhanced versions of their software and services.

Below is a roundup of all the free upgrades to services and software licenses being offered during the Coronavirus outbreak.

If you are a software developer or technology service provider and would like to add any free offers to this list, please contact us and let us know.

In response to the Coronavirus (COVID-19) outbreak, many organizations are asking their employees to work remotely. This, though, brings new challenges to the workplace as users adapt to video meetings, screen sharing, and the use of remote collaboration tools.

To assist a new wave of remote works and get some publicity at the same time, many software developers and service providers have started to offer free licenses or enhanced versions of their software and services.

Below is a roundup of all the free upgrades to services and software licenses being offered during the Coronavirus outbreak.

If you are a software developer or technology service provider and would like to add any free offers to this list, please contact us and let us know.

The good news is there are so many that exploring each individual detail would require two more podcasts. So, I'm just going to whet everyone's appetite with the current list of organizations on Lawrence's page:

Adobe, AT&T, Avid, Cisco, Cloudflare, Discord, Drastic Technologies Ltd., Google, Instant Housecall, LinkedIn, Logmein, Loom, Microsoft, OneClick, Splashtop, TechSmith, Zoho, Zoom

If you or your organization might be in need of any telepresence or other services offered by those above, allow me to encourage you to check out Lawrence's page for the detailed descriptions of each organization's offerings.

Oh!... and one other item here. I noted that Lawrence did not include PornHub on his list of special services being offered during this stay-at-home response to the global pandemic.

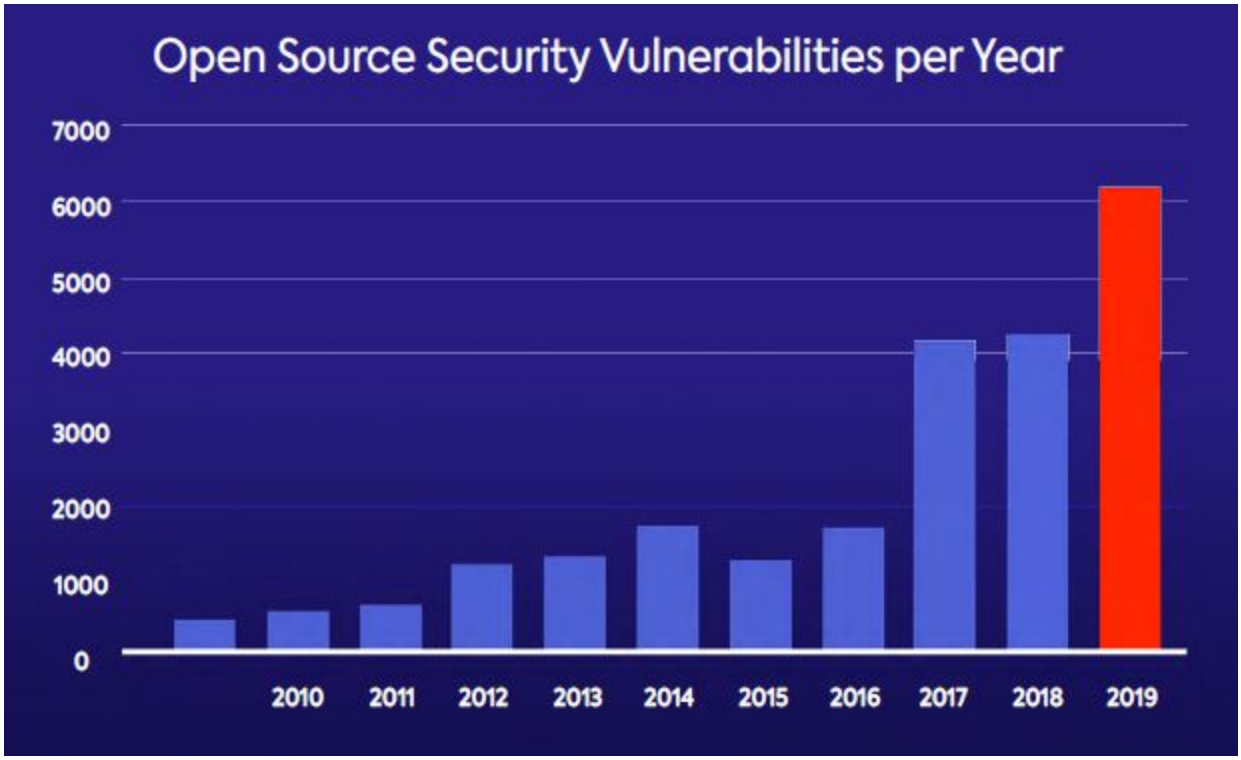
But that was in the new. So I went looking for the details and figured I'd go to the horse's mouth. A Google search returned the headline "Coronavirus-Free Video for Quarantined Italians" at PornHub. But what I was confronted with was definitely not the horse's mouth.

The State of Open Source Vulnerabilities
"WhiteSource" has surveyed over 650 developers, collected data from the national vulnerabilities database, security advisories, peer-reviewed vulnerability databases, issue trackers and more, to formulate a snapshot of the state of software vulnerabilities among open source projects.

https://www.whitesourcesoftware.com/open-source-vulnerability-management-report/

Sophos was brutal in their summary of the report. They wrote:

> Open source bugs have skyrocketed in the last year, according to a report from open source licence management and security software vendor WhiteSource.
>
> The number of open source bugs sat steady at just over 4,000 in 2017 and 2018, the report said, having more than doubled the number of bugs from pre-2017 figures that had never before broken the 2,000 mark.
>
> Then, 2019's numbers soared again, topping 6,000 for the first time, for a rise of 50%.



Okay. So, CWE is the "common weakness enumeration" system which broadly classifies bug types and the report states that by far the most common CWE encountered in the open source world is cross-site scripting (XSS). This class of flaw accounted for nearly 25% of all bugs and was the top for all languages except C. Cross-site scripting was followed by improper input validation, buffer errors, out-of-bound reads, and information disclosure. "Use after free", another memory flaw, came in last at under 5% of errors.

The bug distribution is no surprise. It's common. So there's been no explosion in one particular class of bug. What we want to know is whether the increased numbers arise from there actually being more bugs per line of code, that is, reduced QUALITY of code, is it increased scrutiny of

same quality code which, we know, will reveal more previously undiscovered bugs?, or is it that there's been a rise in the quantity of similar-quality code, thus a naturally higher bug count?

Under the category of "The Good News," the report notes that over 85% of open source security vulnerabilities are disclosed with a fix already available. Tech Giants have invested heavily in better securing and managing open source projects over the past few years, and the community is working hard at security research to publish newly discovered open source security vulnerabilities along with a fix. The fix will usually be an updated version or a patch for the vulnerable code.

The report states:

"Given the continued increase of both open source usage and security research, the number of reported open source vulnerabilities will surely keep rising. In addition, we're starting to see the open source community looking for new initiatives in order to address the chaos in the open source security process. One good example is the GitHub Security Lab, which aims to help researchers, open source project maintainers, and users to easily report suspected vulnerabilities in a secure manner without exposing a zero-day vulnerability into the world.

That's a REALLY good point that I had never really focused upon before: It's easy to keep a critical 0-day secret in proprietary closed-source software secret, since its discoverer only needs to privately contact the software's publisher. But in an inherently open world, where all regular business is conducted publicly and in full view of the world, some mechanism needs to be able to operate behind the scenes. So it's very cool that we have the GitHub Security Lab.

GitHub's embedded disclosure process will encourage open source project maintainers to properly report vulnerabilities, rather than just push a fix. Having the maintainers themselves report vulnerabilities should also lead to higher-quality metadata, like affected versions and fixed-in versions, as opposed to a third party reporting the problem.

Our concern is that, while these tools will help to report vulnerabilities in a proper manner, they will probably aggravate the current problem as software developers are already struggling to keep up with the increased rate of reported open source vulnerabilities.

They also noted:

Another aspect we wanted to look at was the types of vulnerabilities that were most common in 2019. The top five CWE's in 2019 have been consistent over the past several years, and are all related to information disclosure. What's concerning is that the most common CWE's are due to simple code errors and imprecise coding, that all developers could avoid by sticking to fairly basic coding standards.

So, there are a growing number of vulnerability reports facing insufficient developer resources required to examine, evaluate and repair everything. This suggests that assigning useful priorities becomes increasingly critical. It turns out that some recent changes have made this a bit more tricky. The report explains it:

The rising number of reported vulnerabilities demands that development teams quickly prioritize their security alerts. The CVSS (Common Vulnerability Scoring System) is usually the go-to parameter for remediation prioritization, but should it be?

CVSS was updated several times over the past few years (v2 to v3, and most recently v3.1), in the hopes of achieving a measurable, objective standard that helps support all organizations and industries. However, it has also changed the definition of what a high severity vulnerability is.

We looked at over ten thousand vulnerabilities from 2016 to 2019 and checked their CVSS v2, v3.0, and v3.1 to compare the severity breakdown of vulnerabilities in each scoring version over the past four year.

The most noticeable change that we saw in the update from v2 to v3 is that scores rose substantially, since a vulnerability that would have been rated as a 7.6 under CVSS v2 could quickly find itself with a 9.8 under CVSS v30. With CVSS v3.0, teams faced a higher number of high and critical severity vulnerabilities.

Still missing are the tools to prioritize and address them, or even fully understand the vulnerabilities' impact, on their project.

So, yeah… if all vulnerabilities are "hair on fire", then we lose all of the value of ranking and prioritizing.

A despicable attack on encryption

It surely does appear that our government, embodied by crypto-naive politicians, is, one way or another, going to figure out how to break into the encryption-protected assets of American citizens.

The most recent effort, dubbed the "EARN IT" act is almost despicable. First of all "EARN IT" is the most tortured abbreviation we've encountered in some time. It stands for: "Eliminating Abusive and Rampant Neglect of Interactive Technologies."

So, get a load of this. What is it that strong data encrypting companies would be "earning"? The legislation proposes to strip the protection provided by section 230 of the Communications Decency Act from certain apps and companies which would then hold them responsible for user-uploaded content... **unless** they provide a means for "lawful access" to their encryption-protected content.

In other words, the legal protections that currently serve to hold all of our online social media companies harmless for whatever their users post, would now need to be "earned" by allowing law enforcement to have access.

Sadly, EARN IT is a bipartisan effort, having been introduced by (no surprise) anti-encryption crusader Lindsey Graham, Richard Blumenthal and other legislators who continually use the specter of online child exploitation to argue for the weakening of encryption.

Remember that we discussed this back in December 2019: While grilling Facebook and Apple, Lindsey threatened to regulate encryption unless the companies give law enforcement access to encrypted user data while pointing to child abuse.

Graham said to the assembled tech-company heads:

> "You're going to find a way to do this or we're going to go do it for you. We're not going to live in a world where a bunch of child abusers have a safe haven to practice their craft. Period. End of discussion."

The EFF notes that one of the problems with the EARN IT bill, among many, is that the proposed legislation "offers no meaningful solutions" to the problem of child exploitation. They wrote:

"It doesn't help organizations that support victims. It doesn't equip law enforcement agencies with resources to investigate claims of child exploitation or training in how to use online platforms to catch perpetrators. Rather, the bill's authors have shrewdly used defending children as the pretense for an attack on our free speech and security online."

If passed, the legislation will create a "National Commission on Online Child Sexual Exploitation Prevention" tasked with developing "best practices" for owners of Internet platforms to "prevent, reduce, and respond" to child exploitation online. But, as the EFF maintains, "Best practices" would essentially translate into legal requirements:

"If a platform failed to adhere to them, it would lose essential legal protections for free speech."

It turns out that the "best practices" approach arose from pushback over the bill's predicted effects on privacy and free speech – pushback that caused its authors to roll out the new structure. The best practices would be subject to approval or veto by the Attorney General (currently William Barr, who has himself already issued a public call for backdoors), the Secretary of Homeland Security (ditto), and the Chair of the Federal Trade Commission (FTC).

CNET talked to Lindsey Barrett, a staff attorney at Georgetown Law's Institute for Public Representation Communications and Technology Clinic who said that the way that the bill is structured is a clear indication that it's meant to target encryption:

"When you're talking about a bill that is structured for the attorney general to give his opinion and have decisive influence over what the best practices are, it does not take a rocket scientist to concur that this is designed to target encryption."

If the bill passes, the choice for tech companies comes down to either weakening their own encryption and endangering the privacy and security of all their users, or foregoing Section 230 protections and potentially facing liability in a wave of lawsuits.

A senior legislative counsel for the American Civil Liberties Union, said:

"The removal of Section 230 liability essentially makes the 'best practices' a requirement. The cost of doing business without those immunities is too high."

Miscellany

The COVID-19 Backgrounder from Ars Technica

Ars Technica has assembled an excellent backgrounder on COVID-19, from which I learned a lot. For example, I hadn't seen an explanation of "why COVID-19?" Was it preceded by COVID-16, -17 and -18? Nope. The "19"" is from "2019".

Ars excellent piece was created by Beth Mole whose bio begins:

> Beth is Ars Technica's health reporter. She's interested in everything from biomedical research to infectious disease, health policy and law. And she loves all things microbial. Beth has a bachelor's degree in biology and world music from the College of William and Mary, and a Ph.D. in microbiology from the University of North Carolina at Chapel Hill.

When I found that it came as no surprise because it was clear from her phraseology that she was entirely comfortable with the subject. She used every term correctly.

Here's a short link to the piece: https://grc.sc/covid

The SMBGhost Fiasco

Although Microsoft has not commented, what appears to have happened is that Microsoft had become aware of an extremely critical flaw residing in its implementation of the latest version v3.1.1 of its SMB file and resource sharing protocol. And it had prepared a fix for this problem. But, for whatever reason, Microsoft apparently decided not to include it in this last Tuesday's March Madness patchfest. Perhaps they wanted to wait until April? Who knows. They're not saying… but we'll see why that was likely the right call at the time.

What we do know is that news of the apparently planned but canned vulnerability update became public knowledge last Tuesday:

Yes, you heard that right. Shades of the WannaCry and NotPetya worms. The flaw in SMBv3 could not be more critical. Those two infamous SMB worms exploited flaws in SMBv1. But this was a flaw in Microsoft's most recent release of SMB. In fact, it affected only 32 and 64-bit versions of Windows 10 and Server versions 1903 and 1909 because earlier versions don't support the SMBv3.1.1.

No technical details have been published, but short and official summaries describing the bug were posted on the websites of two cyber-security firms, Cisco Talos and Fortinet. Cisco's entry for it was quickly taken down once it became clear that this maximally-critical vulnerability was not actually part of the Patch Tuesday rollout.

Fortinet's entry was definitive: https://fortiguard.com/encyclopedia/ips/48773

They wrote: "The vulnerability is due to an error when the vulnerable software handles a maliciously crafted compressed data packet. A remote, unauthenticated attacker can exploit this to execute arbitrary code within the context of the application." They didn't add "within the context of the server" because both ends of an SMBv3.1.1 connection turn out to be vulnerable. And anytime we hear that "compression" is part of the problem, we nod wisely and say "Ah, yes… another interpreter."

CVE-2020-0796

Next, the guys over at the cybersecurity firm Kryptos Logic performed an Internet-wide scan for all systems or networks exposing the default port 445 to the public. Rather than just scanning for port 443, they checked out the version of SMB answering at each discovered port and determined that approximately 48,000 Windows 10 hosts are, or were, vulnerable to attacks targeting the SMBv3.1.1 vulnerability.

And anyone who's curious can do the same: There are several vulnerability scanners designed to detect Windows devices exposed to attacks on GitHub, including one created by Danish security researcher ollypwn which is designed to determine whether SMBv3 is enabled on the device and whether the compression capability that triggers the bug is enabled.

So this leaves us with the question of how the information leaked and the two theories which are currently circulating:

The first one looks at the Common Vulnerability Reporting Framework (CVRF) and the Microsoft Active Protections Program (MAPP). These systems provide Microsoft with a mechanism for sharing details about upcoming patches with trusted industry partners, such as antivirus makers and hardware vendors. The theory here is that Microsoft may have shared a list of upcoming vulnerabilities, but then removed the bug from the list with little time for some vendors, namely Cisco Talos and Fortinet, to update their own security advisory pages.

Someone with the handle "regnil" tweeted:

> Microsoft releases early versions of their CVRF to MAPP, and then pulls CVEs at the last minute without telling anyone. When they then fail to publish the public CVRF in a timely manner, it's understandable that mistakes like this will happen.
>
> — regnil (@regnil) March 10, 2020

The second theory is that the info about CVE-2020-0796 was accidentally shared via the Microsoft API, which some antivirus vendors, sysadmins, and reporters scrape for information about Patch Tuesday patches, as soon as they come out. The working theory there is that the bug may have been initially scheduled to be patched last week, but was later pulled... without also being removed from the API's backend database, thus eventually making its way into the Talos and Fortinet advisories.

Microsoft has been mum on all of this, reportedly not returning anyone requests for comment. And at the time they would not even say when a patch for this glaring disaster would be delivered.

Knowing enough about the nature of the problem -- which was a convenient side-benefit of the inadvertent disclosure -- many sites quickly posted their recommendations for immediate remediation: "Users are encouraged to disable SMBv3 compression and block TCP port 445 on firewalls and client computers."

Most individuals were never in any danger from this, just as they were not from Wannacry. Remember that the NAT routers we are all behind simply drop anything inbound that's not expected. If it's not a reply to something we originally initiated outbound, it has nowhere to go and is simply ignored.

What happened next??

Two days later, on Thursday, March 12th, Microsoft release a security advisory just as though this had been its plan all along:

https://portal.msrc.microsoft.com/en-US/security-guidance/advisory/CVE-2020-0796

In their advisory, they wrote:

> A remote code execution vulnerability exists in the way that the Microsoft Server Message Block 3.1.1 (SMBv3) protocol handles certain requests. An attacker who successfully exploited the vulnerability could gain the ability to execute code on the target server or client.
>
> To exploit the vulnerability against a server, an unauthenticated attacker could send a specially crafted packet to a targeted SMBv3 server. To exploit the vulnerability against a client, an unauthenticated attacker would need to configure a malicious SMBv3 server and convince a user to connect to it.
>
> The security update addresses the vulnerability by correcting how the SMBv3 protocol handles these specially crafted requests.

Microsoft offers no mitigations but they do have a workaround:

> The following workaround may be helpful in your situation. In all cases, Microsoft strongly recommends that you install the updates for this vulnerability as soon as they become available even if you plan to leave this workaround in place:
>
> Disable SMBv3 compression
>
> You can disable compression to block unauthenticated attackers from exploiting the vulnerability against an SMBv3 Server with the PowerShell command below.
>
> Set-ItemProperty -Path "HKLM:\SYSTEM\CurrentControlSet\Services\LanmanServer\Parameters" DisableCompression -Type DWORD -Value 1 -Force
>
> - No reboot is needed after making the change.
> - This workaround does not prevent exploitation of SMB clients; please see item 2 in the FAQ to protect clients.
> - SMB Compression is not yet used by Windows or Windows Server, and disabling SMB Compression has no negative performance impact.
>
> You can disable the workaround with the PowerShell command below.
>
> Set-ItemProperty -Path "HKLM:\SYSTEM\CurrentControlSet\Services\LanmanServer\Parameters" DisableCompression -Type DWORD -Value 0 -Force
>
> Note: No reboot is needed after disabling the workaround.

So, an examination of all available evidence strongly suggests that this super-critical bug was at first going to be patched on Tuesday. Then, for some reason Microsoft pulled it at the last minute, but not before their original intentions had been broadcast to their security partners. Reading the description of the problem makes clear that this would be really bad. So some curious George went looking for Tuesday's patch for CVE-2020-0796... and found it missing. That made news since, OMG!... and then Microsoft was pretty much forced to release the patch two days later.

But wait… there's more!! Then the other shoe dropped. Can you guess what it was?

If you guessed "A multitude of botched installations and other problems?" you get the brass ring. Having attempted to put this out into the world after the news of the problem had been mistakenly published, it now appears that Microsoft's original decision to hold this one back until it was fully ready was correct. Because at the moment it appears to be anything but ready.

Microsoft's Thursday patch to fix the CVE-2020-0796 SMBv3.1.1 vulnerability went by the knowledge base number KB4551762. According to user reports it is failing to install and worse. It is throwing 0x800f081f, 0x80004005, 0x80073701, 0x800f0988, 0x80071160 & 0x80240016 errors during the installation process.

Win 10 Updates giving me grief

So I've had this issue since KB4497165, but the latest KB4551762 is also giving me the same problem. Basically after it installs, it gets to 7% on the "working on updates" part, then tells me that it failed, and it's undoing changes. Since KB4497165 had people complaining about bugginess and the like, I figured maybe that specific update was the issue, and used the tool that lets you hide updates. Sadly, the current update is giving me the same issues, and I've done all the things like using DISM, deleting the software distribution folder, using the update troubleshooter, CHKDSK, etc. Everything is supposedly fine, even though there's clearly something wrong. Kind of at my wit's end. I know I can just hide/pause updates, but it's not exactly a good idea to do so for extended periods. What can I do?

(Right. And we know that Windows 10 has taken effective control from its users. We can temporarily "defer" updates, but not for long.)

Stuck when update:
My laptop current in version 1909 KB4532693. But when I update to KB4551762 my device show we couldn't complete the updates. Undoing changes. Don't turn off your computer and stuck there. Thank you

Users are posting on Microsoft's official Feedback Hub, on the Microsoft Community website, and on Reddit that none of the usual workarounds for the many various errors helped.

0x800f0988 and 0x800f0900 installation errors were also spotted and reported by Günter Born, one day after KB4551762 was released by Microsoft.

One user reported through Microsoft's Feedback Hub: "Manual Windows Update on the local client works ONCE. It finds the patch, then does nothing! One can attempt to download and install from that page, but it doesn't work! Next, go to the Catalog. Select the correct configuration. Download the patch. Attempt to install it. Doesn't install!"

Another user reported: "When downloading this update my PC started becoming slow and sluggish, the update got stuck at 100%. I restarted the PC then windows updates broke and started looping for a while when checking updates, its now back to normal but now I have a failed cumulative update."

Someone on Reddit wrote: "So I've had this issue since KB4497165, but the latest KB4551762 is also giving me the same problem. After it installs, it gets to 7% on the "working on updates" part, then tells me that it failed, and it's undoing changes."

And those were just the installation problems! They may have been a saving grace for those users, since once the update **is** installed all manner of things start going sideways.

Though less numerous, there are mentions of CPU usage spikes, high disk system usage, system slowdowns, and freezes.

One user wrote: “These issues began yesterday 3/13/20. The update, '2020-03 Cumulative Update for Windows 10 Version 1909 for x64-based Systems (KB4551762)' has failed every time I try to install it with the error code, '0x80071160. When this issue began my disk drive also went up to 100% with little change. I restarted my pc multiple times but both issues persisted.”

Or... “After installing KB4551762 and KB4540673, my system has gone to thrash. Extremely slow and takes ages to get past the Welcome screen. After spending hours trying to login, I somehow managed to uninstall both the updates, rebooted, disabled and re-enabled HyperV but my system won't go back to being normal.”

A report on Microsoft's Feedback Hub says... “Simply downloading the update caused my computer to overheat and freeze multiple times. Finally, with no programs open in the background, the download was able to go through. When I attempted to restart so the update could take effect, it would get stuck at 93% installing the update. Always stuck at 93%.”

And, to top it all off, there are also reports of random restarts or failures to boot, as well as users who are having gaming issues after installing KB4551762 with the monitor starting to flicker after a game starts and the issue going away after closing the game.

Although the KB4551762 installation issues are widespread according to users, some users have managed to successfully deploy the security update using an approach known as a Windows 10 in-place upgrade. The procedure is detailed in a video produced by Microsoft Support which I’ve linked to at the end of the show notes. It is a final Hail Mary for when your system is borked and nothing else works:

https://youtu.be/9BWh0YRmnT8

So, it appears that Microsoft must have had some idea that the KB4551762 update might not be ready for prime time and therefore pulled it back at the last minute. And in retrospect this was probably the right decision. This problem is as critical as it gets, but we’ve seen that the total public Internet exposure is around 48,000, not 48 million. And it wasn’t a 0-day. At the time it was still unknown and not known to be exploited in the wild. So while Microsoft was right to want this fixed fast, if the fix wasn’t really ready, it could have withstood another month of development… except that due to a fluke in the industry’s patch reporting system, the fact of the unpatched vulnerability leaked out and forced Microsoft’s hand.