

Security Now! #771 - 06-16-20

Lamphone

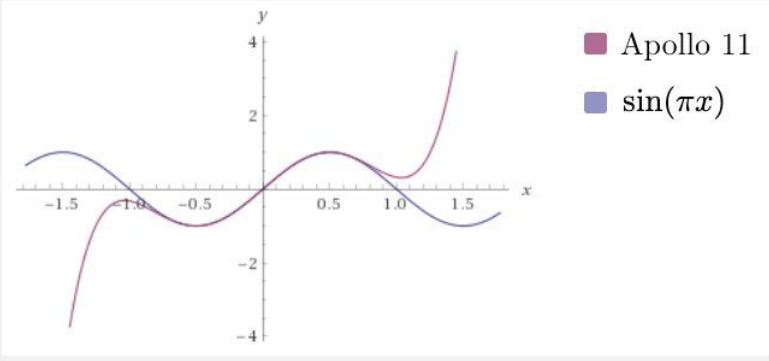
This week on Security Now!

This week we address an accident that the Brave browser guys regret. We take a look at last week's Patch Tuesday and its several ramifications and consequences, we note a few odd new and unwelcome behaviors from this year's 2004 Win10 feature update and dip into yet another side-channel attack on Intel chips. But we also note that a long-awaited powerful anti-malware technology is also about to ship from Intel. We look at the latest new SMB vulnerability named SMBleed, and conclude with an examination of the latest and more-practical-than-most techniques for covertly eavesdropping on a remote location... via a hanging lightbulb.

In 1969 Apollo 11, the spaceflight that first landed humans on the Moon, used just 30 simple instructions to calculate the transcendental functions like sine and cosine essential for navigation:


$\cos(\pi x) = \sin(\pi(x + 1/2))$ $\sin(\pi x)$

Polynomial approximation

$$\sin(\pi x/2) = 2((C_{5/2}x^2 + C_{3/2})x^2 + C_{1/2})x$$
$$C_{5/2} \approx \left(\frac{\pi}{2}\right)^5 \cdot \frac{1}{2 \cdot 5!} \quad C_{3/2} \approx -\left(\frac{\pi}{2}\right)^3 \cdot \frac{1}{2 \cdot 3!} \quad C_{1/2} \approx \frac{\pi}{2} \cdot \frac{1}{2}$$


Legend:
■ Apollo 11
■ $\sin(\pi x)$

Line #	Instruction	Count	Opz/Inter	Notes
32	# SINGLE PRECISION SINE AND COSINE			
33				
34				
35	SPCOS	AD	HALF	# ARGUMENTS SCALED AT PI
36	SPSIN	TS	TEMK	
37		TCF	SPT	
38		CS	TEMK	
39	SPT		DOUBLE	
40		TS	TEMK	
41		TCF	POLLEY	
42		XCH	TEMK	
43		INDEX	TEMK	
44		AD	LIMITS	
45		COM		
46		AD	TEMK	
47		TS	TEMK	
48		TCF	POLLEY	
49		TCF	ARG90	
50	POLLEY		EXTEND	
51		MP	TEMK	
52		TS	SQ	
53			EXTEND	
54		MP	C5/2	
55		AD	C3/2	
56			EXTEND	
57		MP	SQ	
58		AD	C1/2	
59			EXTEND	
60		MP	TEMK	
61		DDOUBL		
62		TS	TEMK	
63		TC	Q	
64	ARG90		INDEX	A
65		CS	LIMITS	
66		TC	Q	# RESULT SCALED AT 1



Browser News

Brave Browser caught and chastised...

... for tweaking user-entered URLs for its benefit

<https://twitter.com/BrendanEich/status/1269313200127795201>

"Cryptonator 1337", was the first to note this behavior which he or she found objectionable, tweeting: "So when you are using the @brave browser and type in 'binance.us' you end up getting redirected to 'binance.us/en?ref=35089877' – I see what you did there mates."

BrendanEich (@BrendanEich), Brave's CEO and co-founder, whose Twitter bio also notes that he co-founded Mozilla & Firefox and created JavaScript, tweeted:

1/ We made a mistake, we're correcting: Brave default autocompletes verbatim "http://binance.us" in address bar to add an affiliate code. We are a Binance affiliate, we refer users via the opt-in trading widget on the new tab page, but autocomplete should not add any code.

("Binance" is a cryptocurrency exchange which pays Brave for referrals from Brave's widget on the browser's "New tab" page.)

Brendan continues for another 6 tweets explaining, after which someone replied: "It's not a mistake, you did it on purpose"

To which Brandan tweeted: "I think you used "mistake" where you meant "accident." I never said it was accidental. We were treating it like a search query (which all big browsers do tag with an affiliate IDentifier to get paid from by the search provider). But a valid domain name is not a search query. Fixing."

So... What happened was that users discovered that manually entering a URL into Brave's URL field would invoke the browser's autocomplete of the URL. Which is a nice convenience. But Brave went a step further to proactively tag that user-entered URL with their own affiliate tag, which the user never entered into the browser's URL field. Brave's users felt that that was taking "affiliation" a step or two too far. And, when Brave was publicly confronted with this behavior they owned up to it, apologized and have removed affiliate links from manually entered URLs.

I'm sure this wasn't nefarious, and it would not have altered the browser's cookie exchange in any way. So it would have no effect upon user tracking. The browser's User-Agent header could have identified the visitor as a Brave-sourced user, but I presume that Binance.us is only equipped to accept affiliate tags through the query URL. Brave is now a full Chromium-based browser, which makes sense since maintaining a secure browser and keeping it state-of-the-art is a massive burden. We would prefer not to have a monoculture in browsers, but a modern web browser has become like an operating system: It's just no longer something that can be built up from scratch. And if someone said "yeah, but I can do it" the proper response is: "Why bother? We already have browsers and they're fine."

Security News

Patch Tuesday

Microsoft continues its record-break streak. Or as Sophos put it:

"Whoosh. You hear that? It's the sound of Microsoft's security fire hose spraying out a river of CVE fixes. That's right – Patch Tuesday was last [this] week and the software giant released patches to fix 129 CVEs."

In other words, it has once again broken its all-time record for the most patches released in one month.

While most of those are regarded as and rated "important", 11 of those 129 CVE are CRITICAL remote code execution vulnerabilities which Windows 10, since last Tuesday, no longer has.

There's CVE-2020-1286, a Windows shell RCE triggered by improper file path validation.

And 1299, an RCE bug that an attacker could exploit using a malicious .LNK file and associated binary. Note that either we still haven't got .LNK link files working right, or we keep breaking them, since Windows has been having security problems with link files from the start. In this case Microsoft warns us that if a malicious link file was placed onto a removable drive or network share, clicking on the .LNK file would run the attacker's malicious code in the file.

There's 1281, a vulnerability in the Windows Object Linking and Embedding (OLE) code stemming from poor input validation and it's exploitable via a malicious website, file, or email message.

1248 is a memory object handling bug in GDI, Windows Graphics Device Interface, which is deliverable by a website, instant message, or document file.

Those all affected Win10, of course, since Windows 7 is no longer being maintained, and many of these also affected the latest 2004 build of Windows 10 since, of course, most of the code never changes.

Not to be forgotten, IE had its own batch of critical vulnerability bumbles. Both IE 9 and 11 were susceptible to RCE via bug CVE-2020-1213, 1216 and 1260, all memory handling errors affecting VBScript.

The original Edge browser (isn't that history, yet?) had a critical vulnerability, 1073, a memory handling bug in its ChakraCore JavaScript engine. And CVE-2020-1219 affects both IE and EdgeHTML with more memory-handling issues.

1181 is a bug in the SharePoint Server. It can be exploited by unsafe ASP.Net controls that don't filter properly. Attackers able to upload a malicious page to the server (not clear how they would do that, but perhaps through remote website authoring) could achieve pwnage. As a consequence, admins of SharePoint Enterprise Server 2016, Foundation 2010 SP2 and 2013 SP1, or SharePoint Server 2019 should all patch now.

There's also 1300, a long standing bug in Windows' handling of cabinet files. It affects most versions of Windows, Win7 through Win10 2004, and also Windows Server.

And, believe it or not, those were just the 11 **critical** bugs. If I were to attempt to detail the other one hundred and eighteen "important" flaws, this entire podcast would have to be retitled: "Patch Tuesday." I'll spare us that, since we have plenty more to talk about. In the meantime, Microsoft, **BIG** congrats on achieving another lifetime milestone.

And speaking of milestones, we also have...

The case of the disappearing printer port

Microsoft's disclosure of this oddball Win10 delight is titled: "USB printer port missing after disconnecting printer while Windows 10 (version 1903 or later) is shut down" and it is stated as applying to: Windows 10, version 1903, all editions. Windows 10, version 1909, all editions. And Windows 10, version 2004, all editions.

<https://support.microsoft.com/en-us/help/4566779/usb-printer-port-missing-after-disconnecting-printer-while-windows-10>

What happens? Microsoft explains:

"If you connect a USB printer to Windows 10 version 1903 or later, then shut down Windows and disconnect or shut off the printer, when you start Windows again the USB printer port will not be available in the list of printer ports. Windows will not be able to complete any task that requires that port."

Resolution:

You can avoid the issue by connecting a powered-on USB printer before starting Windows.

"Microsoft has confirmed that this is a problem in the Microsoft products that are listed in the "Applies to" section. We are working to fix the issue in a future version of the operating system."

According to reporting of this in the tech press, if you need to print something to your USB-connected printer and you didn't have it on **before** you started Windows, no problem. Just shut down your computer, turn the printer on and wait for it to finish initializing and settle down, then you can fire up Windows and the printer port should reappear and you'll be able to print. Because this is a state-of-the art modern operating system.

And, believe it or not, in a related but separate matter...

Last week;s Patch Tuesday broke ALL PRINTING (even to PDFs) for many users:

Windows 10 users are reporting that they are unable to print to printers from several vendors after installing last week's updates for Windows 10 versions 1903, 1909, and 2004 OS's.

The two specific patches causing the trouble have been determined to be cumulative updates KB4560960 and KB4557957. Although Microsoft hasn't yet gone official, a Microsoft Answers Independent Community Advisor has stated that Microsoft engineers are "already aware of this

issue and working a patch to be deployed in the next update." Oh, joy. No printing for a month.

So after updating their machines last Tuesday, users started flooding both Microsoft Answers forums and Reddit with reports of printing issues affecting various models of HP, Canon, Panasonic, Brother, and Ricoh devices. Typical posted included:

- "Unable to print after installing update KB4560960 and/or KB4561608. Uninstalling updates fixes problem. This is happening to every Windows 10 computer in our organization as updates install."
- Another says that right after installing the KB4560960 on multiple systems, users started reporting "Windows cannot print due to a problem with the current printer setup" errors that went away after uninstalling the update.
- Someone wrote: "Found this problem today where all clients at a customer site had the same problem," others complained. "They have Ricoh, but a few other brands too. Even the virtual PDF printers do not work anymore. Explorer.exe crashes completely when doing a test-print..."
- A network technician posted: "HPs seem to be hit or miss with this issue. Ricoh / Canon / Brother / KM / Kyocera all seem to be experiencing problems. As everyone else is saying, backing out update KB4560960 and postponing updates seems to be our only salvation at this point."
- "Hopefully Microsoft will produce a patch for this quickly, call volume is picking up with everybody returning to work, this is going to make things awfully hectic!"

Affected users have found that the printer's native driver **can** be replaced with PCL6 drivers which reportedly work, or by uninstalling last week's cumulative updates to restore printing, and also to restore those 11 critical remote code execution bugs. You'll be fine. It's been determined that attempting to uninstall and reinstall the printer, or updating its drivers, does not help. PCL6 printer drivers do work... either vendor-specific PCL6 drivers or the universal Windows 10 PCL6 drivers for Canon, HP, Ricoh, Kyocera, and Brother.

Windows 10 2004 is messing up SSDs and non-SSDs.

Just a quick note for those running Windows 10 who have moved to 2004 with SSDs: The 2004 feature update has broken Windows awareness that it has ever previously defragmented the system's drive. As a result, rather than only defragging occasionally, like once a month by design to improve the performance of Windows "volume shadow copy on write" performance, Win10 is defragging every time the system is started.

This isn't a huge problem since SSDs should have strong write endurance, but it's still not what we want. Microsoft has acknowledged the problem but hasn't indicated when it will be resolved. The release notes for the Insider Preview build 19551 states: "Thank you for reporting that the Optimize Drives Control Panel was incorrectly showing that optimization hadn't run on some devices. We've fixed it in this build."

And in another oddity, Win10 2004 is also attempting to use the TRIM command on non SSD drives. That fails and logs an error into the Windows error log. But it should not be trying.

Our longtime listeners will recall that SSDs have a TRIM command to allow the operating system to inform the drive of the drive regions that are not in use by the OS. Normally, drives treat all sectors alike and only the OS has any awareness of which regions are in use by its file system, and which are free. Hard drives write data by simply overwriting what was there before. But SSDs are only able to set bits that have been previously reset by an erase cycle. And erase cycles erase large blocks of the SSD all at once. This means that to write a small region of a larger block, the previous contents of the larger block must first be read and held in RAM while the underlying block is all reset. Then the cached data must be rewritten into the block. But IF the SSD has an awareness of which sectors are not in use, it can leave them reset rather than needing to rewrite them with unneeded data. AND those reset and unwritten blocks can later be written to directly without needing any pre-erase since they were left erased.

But although doing this clearly makes no sense for hard drives, some new bug introduced into 2004 is causing Windows to issue these superfluous TRIM commands to spinning hard drives nonetheless.

There were also reports that many programs would no longer run at all after last Tuesday's updates, but it turned out that the problem was caused by an interaction with a recent update for Avast and AVG anti-malware software. They hook into a feature that allows them to intercept the running of other programs and that didn't go as expected.

Overall, much as Win10 2004 is promising some new features, it does feel as though perhaps holding back a bit and waiting for things to settle down might be prudent.

SMBleed

I should mention that a new information leakage vulnerability was introduced in Windows 10 1903 and it's present in all releases since. With Win10 1903, Microsoft's very troubled SMB protocol was upgraded to support optional compression. It's present in SMBv3.1.1.

The trouble is, this allows new information probing attacks to succeed thanks to the foothold this provides an attacker. This is loosely related to the SMB Ghost vulnerability that so alarmed Microsoft earlier this year when they warned everyone that it could be turned into a wormable exploit. As we know, that never happened, but it was successfully used against selected targets.

Microsoft's security advisory published last week stated that "an attacker who successfully exploited the vulnerability could obtain information to further compromise the user's system."

"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."

So the problem will probably mostly affect Win10 servers (and Win10 core servers are affected)

having their SMB services publicly exposed. And at this point, anyone who has an SMB protocol service open to the public has few excuses when something uses that port to crawl into their network. It's not only an open port, it's an open invitation.

The ZecOps guys who disclosed and detailed their findings have been taking some heat for also releasing a pair of proof-of-concept demos:

- An uninitialized kernel memory read PoC that creates a local file containing target computer kernel memory
- A pre-Auth RCE PoC combining SMBleed with SMBGhost that opens a reverse shell with system access

When they were asked why they didn't wait for Windows users to patch their systems and chose to publish their PoCs when SMBleed was disclosed, the ZecOps guys said that the security vulnerability was not critical on its own.

They argued that "After the patch was made available, the vulnerability is easy to spot and reproduce." and that "Only [when used] in combination with another primitive, such as SMBGhost [would] SMBleed be critical." And since SMBGhost was patched three months ago, people should be safe.

The takeaway for our listeners? if your company still has SMB exposed to the public, really, really, invest in a VPN solution with multifactor authentication. Since this attack works against unauthenticated attackers, not even multifactor SMB authentication would protect your servers.

I'm sure that no one listening would fall for this, but just for the record:

This fraudulent website extortion scam has been in the news recently because those behind it are if nothing else, persistent... and because the eMail is apparently written by someone with at least somewhat passable English, which is somewhat unusual. The eMail reads:

Subject: Your Site Has Been Hacked

PLEASE FORWARD THIS EMAIL TO SOMEONE IN YOUR COMPANY WHO IS ALLOWED TO MAKE IMPORTANT DECISIONS!

We have hacked your website [URL REDACTED] and extracted your databases.

How did this happen?

Our team has found a vulnerability within your site that we were able to exploit. After finding the vulnerability we were able to get your database credentials and extract your entire database and move the information to an offshore server.

What does this mean?

We will systematically go through a series of steps of totally damaging your reputation. First your database will be leaked or sold to the highest bidder which they will use with whatever their intentions are. Next if there are e-mails found they will be e-mailed that their information

has been sold or leaked and your site [website URL] was at fault thusly damaging your reputation and having angry customers/associates with whatever angry customers/associates do. Lastly any links that you have indexed in the search engines will be de-indexed based off of blackhat techniques that we used in the past to de-index our targets.

How do you stop this?

We are willing to refrain from destroying your site's reputation for a small fee. The current fee is [ransom amount] USD in bitcoins (BTC).

If you decide not to pay, we will start the attack at the indicated date and uphold it until you do, there's no counter measure to this, you will only end up wasting more money trying to find a solution. We will completely destroy your reputation amongst google and your customers.

The ransom demands generally range between \$1,500 and \$3,000 in Bitcoin.

This is, of course, nonsense. I've received exactly this eMail several times myself at the address registered for my one remaining domain at Network Solutions. Unlike Hover, where I have moved everything that's moveable, Network Solutions does not offer registered eMail blinding with their service, and I refuse to pay extra for it.

But security experts are nothing if not curious. And since the "Breachstortion" campaign, as one group has taken to calling it, lists the extortionist's pay-to bitcoin wallet addresses, and since it's possible to monitor the Bitcoin blockchain for payments made to wallets, the web security firm WebARX decided to see how successful this campaign might be. By scouring social media postings for references to this campaign, WebARX identified the multiple Bitcoin wallets being used to collect the ransom payments. WebARX wrote:

Unfortunately, looking at the different bitcoin wallets linked to these attacks there have been at least 5 people who have fallen to the scam and paid ransom. One of the wallets linked to the attack has received close to \$2000 worth of payments in bitcoins. Another wallet used in this scam which has not yet received payments, but has been reported for abuse for 81 times.

Those paying are apparently not the brightest bulbs in the box since the threat and payment demand contains no eMail or website contact details. The scammers tell the recipient not to bother replying to the email at all, and that there's no website where you can trace your payment and see whether they've received the money. They explain to their intended victim that "Bitcoin is anonymous and no one will find out that you have complied."

Presumably that's meant to set the target's mind at rest by convincing them that the act of paying will not itself draw attention to the (fake) "breach", even though it means you're relying entirely on the crooks to keep track of which payments were made to "protect" which website's data. And unless they are sending a separate Bitcoin address to each eMail address, and tracking them all, or generating custom per-demand amounts, neither of which seem likely, there's no way for the cretins to know who paid up and who didn't.

Clearly, this is just a blind spam gamble, hoping that it will pay off. But, as one of P.T. Barnum's more well-known quotes observes: "There's a sucker born every minute."

And remember that SophosLabs similarly reported on the apparent success of those porn-scramming solicitations. You know, the one that claims to have activated your webcam and recorded what you were doing while watching online porn. It's a bit unnerving the first time one of those arrives, and I've had a few friends who have also received that spam ask me whether that's possible. So Sophos did some similar digging into the Blockchain and discovered that the porn scamming crooks are generating enough money that we can expect those particular eMail to keep coming in. Believe it or not, that campaign is generating as much as \$100,000 a month.

What IS real...

That said, there **are**, unfortunately, some authentic database ransom attacks. Insecure SQL servers exist for online merchants and there have been multiple accounts of their databases being copied and ransom notes left demanding payment... or else.

Being authentic, those attacks have been far more successful in collecting ransoms. Researchers have tracked a total of 5.8 BTC (currently worth around \$54,500) having been sent to the attacker's address by over 100 victims into just two of the attackers' wallets. In the past, similar database attack ransom attacks have targeted MongoDB databases and MySQL servers.

Please please please never leave SQL servers exposed to the public Internet. Not even on non-standard ports. Changing the port is no longer sufficient protection.

Another side-channel attack on Intel chips

As we know all too well, the past two years have been a watershed period for security researchers poking into the seemingly unlimited supply of new vulnerabilities which have arisen as a consequence of our processor architectures -- mostly Intel's -- attempts to cleverly squeeze every last microcycle of possible computational power from our chips.

Unfortunately, even something as seemingly benign as a cache, which is used to decouple the demands of the fast processor from the comparatively lethargic reluctance of DRAM to give up its data, can, as we have learned, have its contents probed by other processes sharing the same hardware.

Throughout 2018, Intel's engineers learned to just what degree their clever engineering, meant to optimize their chip's performance, could be used by malicious code to exfiltrate data across virtually every security boundary that had been erected.

So today, no one would be surprised to learn that yet another fault has been discovered and leveraged by researchers to penetrate Intel's SGX -- software guard extension -- secure enclave. Last week, two separate academic teams disclosed two new distinctive exploits that do exactly that: pierce Intel's Software Guard eXtensions which is designed to be, by far, the most secure region created by Intel's processors.

And, of course, these new attacks aren't the first to soften the SGX security wall. Back in 2018, a different team of researchers used the attack known as "Meltdown", and still another team broke SGX earlier this year. Things are not looking good for Intel's security at the moment.

As we know, Intel mitigated the earlier SGX vulnerabilities by introducing microcode updates. But they were insufficient. So Intel has again released the new updates which should be available to end users before long. As we know, Intel's microcode can be patched at boot time and either the motherboard's startup code or our operating system's boot can apply on-the-fly fixes. So that's what will eventually happen.

The most recent attacks are known as SGXaxe and CrossTalk. Both use new and different side-channel attacks to infer what's going on within the walled-off region.

SGXaxe is able to steal large chunks of SGX-protected data of an attacker's choice such as wallet addresses or other secrets used in financial transactions involving blockchains. And SGXaxe can steal the cryptographic keys that SGX uses for "attestation," -- the process of proving to a remote server that the hardware is a genuine Intel processor and not a malicious simulation of one. A remote server can require connecting devices to provide these attestation keys before it will carry out financial transactions, play protected videos, or perform other restricted functions. In their paper titled "SGXaxe: How SGX Fails in Practice", the researchers from the University of Michigan and the University of Adelaide in Australia wrote:

With the machine's production attestation keys compromised, any secrets provided by [the] server are immediately readable by the client's untrusted host application while all outputs allegedly produced by enclaves running on the client cannot be trusted for correctness. This effectively renders SGX-based DRM applications useless, as any provisioned secret can be trivially recovered. Finally, our ability to fully pass remote attestation also precludes the ability to trust any SGX-based secure remote computation protocols.

Anyway, everyone gets the idea. For anyone wanting more information, Dan Goodin assembled some comprehensive coverage of this for ArsTechnica. The link is here in the show notes:

<https://arstechnica.com/information-technology/2020/06/new-exploits-plunder-crypto-keys-and-more-from-intels-ultrasecure-sgx/>

But the bottom line for most of us is that this is far far from as important as never clicking on any form of solicitation in eMail, nor updating your browser's FLASH player when a website tells you to. Ultimately, this research is of crucial importance because, as an industry, all of our security is ultimately built upon the assumption that the underlying hardware is secure. It's impossible to build anything secure on top of hardware that isn't. So the fact that these skirmishes are going on in the background between Intel and academic researchers with microcode patches pouring forth is all for the good.

So that's the bad news. Here's some truly good news:

Control-flow Enforcement Technology gets real

Meanwhile, Intel has just announced that it's long-awaited hardware Control-flow Enforcement Technology (CET) is ready and will be included in their next "Tiger Lake" mobile CPUs.

I talked about CET a while back. Recall that it's the technology that maintains a completely separate shadow hardware stack which, unlike the processor's traditional hybrid stack that mixes together subroutine return addresses with subroutine calling parameters and dynamically allocated data and buffers, the CET stack only contains the control-flow data, namely subroutine returns. Since this data is maintained by the hardware and is not visible in any way to software, unlike the software-visible stack, it's not subject to malicious manipulation. So when any return-from-subroutine instruction is encountered, this new Intel hardware will compare the return address it previously stored to the return address on the software stack and if they are not in agreement the executing thread will be terminated.

This is a massive win for Intel processor security since in one fell swoop it eliminates a broad class of attacks, all of those surprisingly effective "return oriented programming" (ROP) attacks that the previous randomization of operating system, kernel and program address space (ASLR and KASLR) were attempting to mitigate in software. Since ASLR can only place software modules in a finite and small number of places, we've seen that some attackers are fine with just guessing and failing... because some percentage of the time, like around 4% (1/256th) they will succeed. And when it's a numbers game, that might be good enough.

But the presence of a hardware shadow stack ends those games finally, once and for all.

Thanks to the fact that Intel has been working on this for the past 4 years, since 2016 when they first published their CET specification, software publishers have had time to get ready and add support to their code, waiting for the day then this next-generation active anti-malware technology would finally ship.

CET support is already present in the GNU standard C library "Glibc", and Microsoft has also added CET support to Windows Insiders, calling their support of this feature "Hardware-enforced Stack Protection."

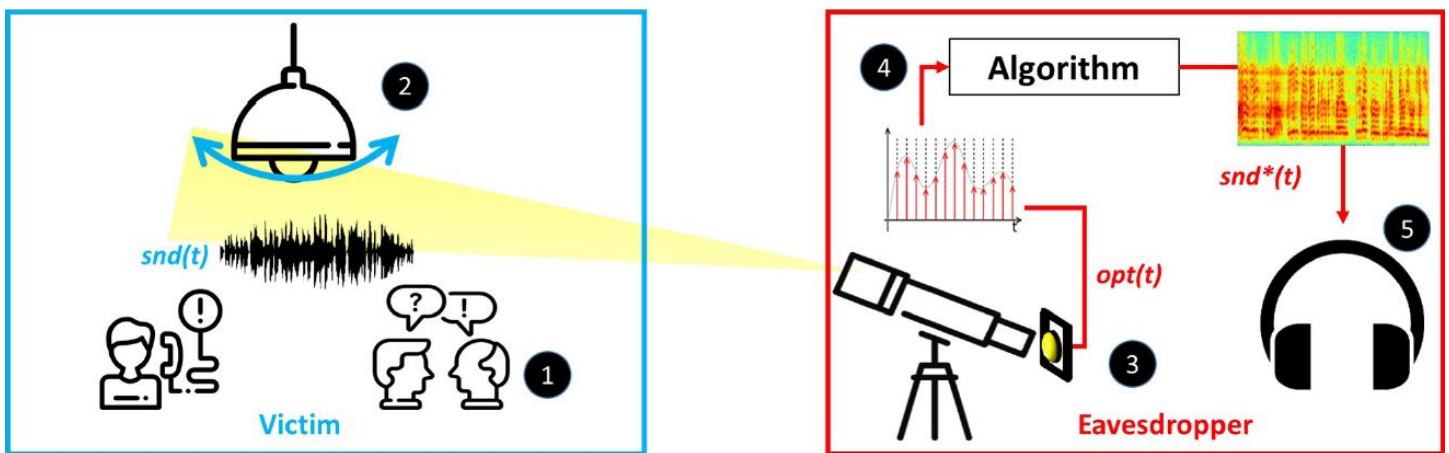
Once we actually have the chips apps and operating systems will be able to enable their support and opt-in to the truly significant protection CET provides.

As I noted, CET will first appear in the Tiger Lake architecture mobile chips, but the technology will also be available in desktop and server platforms.

Lamphone

I was originally going to title this week's podcast "Newly Vulnerable" and group the various new vulnerabilities here at the end. And I was going to place the "Lamphone" under "This Week in Wacky Surveillance Technology." But the researchers have a quite-serious 15-page research paper (which, you know, always hooks me) and their work will be presented at the Black Hat security conference in August. So I decided that it won the spot for this week's title and topic.

No one will be surprised to learn that this research hails from the Ben-Gurion University of the Negev and the Weizmann Institute of Science, which produces a steady stream of new and imaginative data exfiltration schemes. What makes their work stand out is that they wrestle every whacko scheme to the ground, applying solid science and physics to the challenge.



https://ad447342-c927-414a-bbae-d287bde39ced.filesusr.com/ugd/a53494_443addc922e048d89a664c2423bf43fd.pdf

This time I'm going to skip the abstract of their paper and share their introduction, since this really does represent an advance in the state of the eavesdropping art thanks to the back-end computational transform that's needed to convert a lightbulb's highly nonlinear frequency response into reconstructed speech.

Imagine a tuning fork. As we know, you thwack it and it vibrates at its natural resonance frequency. In terms of speech, it's an extremely sharp bandpass filter. And a lightbulb is not much better. So rather than directly obtaining the speech waveform, what you get is more of a speech frequency-modulated amplitude modulation.

So here's what they explained...

I. INTRODUCTION

Eavesdropping, the act of secretly or stealthily listening to a target/victim without his/her consent, by analyzing the side effects of sound waves on nearby objects (e.g., a bag of chips) and devices (e.g., motion sensors) is considered a great threat to privacy. In the past five years, various studies have demonstrated novel side-channel attacks that can be applied to eavesdrop

via compromised devices placed in physical proximity of a target/victim. In these studies, data from devices that are not intended to serve as microphones (e.g., motion sensors, speakers, vibration devices, and magnetic hard disk drives) are used by attackers to recover sound.

Sound eavesdropping based on the methods suggested in the above mentioned studies is very hard to detect, because applications/programs that implement such methods do not require any risky permissions (such as obtaining data from a video camera or microphone). As a result, such applications do not raise any suspicion from the user/operating system regarding their real use (i.e., eavesdropping). However, such methods require the eavesdropper to compromise a device located in proximity of a target/victim in order to: (1) obtain data that can be used to recover sound, and (2) exfiltrate the raw/processed data.

To prevent eavesdroppers from implementing the above mentioned methods which rely on compromised devices, organizations deploy various mechanisms to secure their networks (e.g., air-gapping the networks, prohibiting the use of vulnerable devices, using firewalls and intrusion detection systems). As a result, eavesdroppers typically utilize three well-known methods that don't rely on a compromised device.

The first method exploits radio signals sent from a victim's room to recover sound. This is done using a network interface card that captures Wi-Fi packets sent from a Wi-Fi router placed in physical proximity of a target/victim. While routers exist in most organizations today, the primary disadvantages of these methods is that they cannot be used to recover speech or they rely on a previously collected dictionary to achieve their goal (i.e., only words from the pre-collected dictionary can be classified).

[We talked previously about how reflected Wi-Fi signals can be used to detect a person's motion and movements within a room. A group of researchers managed to detect the micro-movements of the mouths of people speaking and then mapped them back into what they must be saying. Needless to say, this was strictly a proof of concept and would hardly be practical.]

The second method, the laser microphone, relies on a laser transceiver that is used to direct a laser beam into the victim's room through a window; the beam is reflected off of an object and returned to the laser transceiver which converts the [audio modulated] beam into an audio signal. In contrast to the previous limited radio-based methods, laser microphones can be used in real time to recover speech. However, the laser beam can be detected using a dedicated optical sensor. The third method, the Visual Microphone, exploits vibrations caused by sound from various materials (e.g., a bag of chips, glass of water, etc.) in order to recover speech by using a video camera that supports a very high frame per second (FPS) rate (over 2200 Hz). In contrast to the laser microphone, the Visual Microphone is totally passive, so its implementation is much more difficult for organizations/victims to detect. However, the main disadvantage of this method, according to the authors, is that the Visual Microphone cannot be applied in real time, because it takes a few hours to recover a few seconds of speech, since processing high resolution and high frequency (2200 frames per second) video requires significant computational resources. In addition, the hardware required (a high FPS rate video camera) is expensive.

So...

In this paper, we introduce "Lamphone," a novel side-channel attack that can be applied by eavesdroppers to recover sound from a room that contains [an exposed] hanging lightbulb. Lamphone recovers sound optically via an electro-optical sensor which is directed at a hanging bulb; such bulbs vibrate due to air pressure fluctuations which occur naturally when soundwaves hit the hanging bulb's surface. We explain how a bulb's response to sound (a millidegree vibration) can be exploited to recover sound, and we establish a criterion for the sensitivity specifications of a system capable of recovering sound from such small vibrations.

Then, we evaluate a bulb's response to sound, identify factors that influence the recovered signal, and characterize the recovered signal's behavior. We then present an algorithm we developed in order to isolate the audio signal from an optical signal obtained by directing an electro-optical sensor at a hanging bulb.

We evaluate Lamphone's performance on the tasks of recovering speech and songs in a realistic setup. We show that Lamphone can be used by eavesdroppers to recover human speech (which can be accurately identified by the Google Cloud Speech API) [which I suppose means that it would be possible to setup an entirely automated listening station which triggers human involvement only when specific keywords were voiced] and [they say] singing (which can be accurately identified by Shazam and SoundHound) from a bridge located 25 meters (27 years) away from the target office containing the hanging bulb.

We also discuss potential improvements that can be made to Lamphone to optimize the results and extend Lamphone's effective sound recovery range. Finally, we discuss countermeasures that can be employed by organizations to make it more difficult for eavesdroppers to successfully use this attack vector.

A. Contributions

We make the following contributions: We show that any hanging light bulb can be exploited by eavesdroppers as a means of recovering sound from a victim's room. Lamphone does not rely on the presence of a compromised device in proximity of the victim (addressing the limitation of Gyrophone, Hard Drive of Hearing, and other methods. Lamphone can be used to recover speech without the use of a previously collected dictionary (addressing the limitations of other external and internal methods). Lamphone is totally passive, so it cannot be detected using an optical sensor that analyzes the directed laser beams reflected off the objects (addressing the limitation of a laser microphone). Lamphone relies on an electro-optical sensor and can be applied in real-time scenarios (addressing the limitations of the Visual Microphone).

In other words, it's arguably a true and potentially useful advance in remote passive conversational audio eavesdropping.

I was unsure how much time we would have, so I wasn't sure I'd be able to share some of the techie details to give everyone a sense for what these guys did. But here's their discussion of the challenges of "Lightbulbs as Microphones"...

IV. BULBS AS MICROPHONES

In this section, we perform a series of experiments aimed at explaining why light bulb vibrations can be used to recover sound and evaluate a bulb's response to sound empirically.

We measure the vibration of a hanging bulb and we establish a criterion for the sensitivity specifications of a system capable of recovering sound from these vibrations

1) Measuring a Hanging Bulb's Vibration:

First, we measure the response of a hanging bulb to sound. This is done by examining how sound produced in proximity to the hanging bulb affects a bulb's three-dimensional vibration.

Experimental Setup: We attached a gyroscope (MPU-6050GY-5216) to the bottom of a hanging E27 LED light bulb (12watts); that bulb was not illuminated during this experiment. A Raspberry Pi 3 was used to sample the gyroscope at 800 Hz. We placed Logitech Z533 speakers very close to the hanging bulb (one centimeter away) and played various sine waves (100, 150, 200, 250, 300, 350, 400 Hz) from the speakers at three volume levels (70, 95, 115 dB). We obtained measurements from the gyroscope while the sine waves were played.

Results: Based on the measurements obtained from the gyroscope, we calculated the average peak-to-peak difference (in degrees) for θ (theta) and ϕ (phi). The average peak-to-peak difference was computed by calculating the peak-to-peak difference between every 800 consecutive measurements (that were collected from one second of sampling) and averaging the results. The frequency response as a function of the average peak-to-peak difference reveal three interesting insights: the average peak-to-peak difference for the angle of the bulb is: (1) very small (0.005-0.06 degrees), (2) increases as the volume increases, and (3) changes as a function of the frequency. Based on the known formula of the spherical coordinate system, we calculated the 3D vector (x,y,z) that represents the peak-to-peak vibration on each of the axes (by taking the distance between the ceiling and the bottom of the hanging bulb into account). We calculated the Euclidean distance between this vector and the vector of the initial position. The results show that sound affected the hanging bulb, causing it to vibrate in 300-950 microns between the range of 100-400 Hz.

2) Capturing the Optical Changes:

We now explain how attackers can determine sensitivity of the equipment (an electro-optical sensor, a telescope, and an ADC) needed to recover sound based on a bulb's vibration. We've established the criterion for recovering sound: the attacker's system (consisting of an electro-optical sensor, a telescope, and an ADC) must be sensitive enough to capture the small optical differences that are the result of a hanging bulb that moves in 300-950 microns. In order to demonstrate how eavesdroppers can determine the sensitivity of the equipment they will need to satisfy the above mentioned criterion, we conduct another experiment.

Experimental Setup: We directed a telescope at a hanging 12 watt E27 LED bulb. [I placed a diagram at the top of this coverage in the show notes.] We mounted an electro-optical sensor (the Thorlabs PDA100A2, which is an amplified switchable gain light sensor that consists of a photodiode, used to convert light to electrical voltage) to the telescope.

The voltage was obtained from the electro-optical sensor using a 16-bit ADC NI-9223 card and was processed in a LabVIEW script that we wrote. The internal gain of the electro-optical sensor was set at 50 dB.

We placed the telescope at various distances (100, 200, 300, 420, 670, 830, 950 cm) from the hanging bulb and measured the voltage that was obtained from the electro-optical sensor at each distance.

Results: The results of this experiment were used to compute the linear equation between each two consecutive points. Based on the linear equations, we calculated the expected voltage at 300 microns and 950 microns. From this data, we can determine which frequencies can be recovered from the obtained optical measurements. A 16-bit ADC with an input range of $[-10,10]$ volts (e.g., like the NI-9223 card used in our experiments) provides a sensitivity of 300 microvolts, provided by a 16-bit ADC, which is sufficient for recovering the entire spectrum (100-400Hz) in the 200-300 cm range, because the smallest vibration of the bulb (300 microns) from this range is expected to yield a difference of 300 microvolts. However, this setup cannot be used to recover the entire spectrum in the 670-830 cm range, so an ADC that provides a higher sensitivity is required. A 24-bit ADC with an input range of $[-10,10]$ voltage provides a sensitivity of 1 microvolt. This is sufficient for recovering the entire spectrum (100-400Hz) in the range of 670-830 cm, because the smallest vibration of the bulb (300 microns) from this range is expected to yield a difference of 54 microvolts. In order to optimize the setup so it can be used to detect frequencies that cannot be recovered, attackers can: (1) increase the internal gain of the electro-optical sensor, (2) use a telescope with a lens capable of capturing more light (we demonstrate this later in the paper), or (3) use an ADC that provides a greater resolution and sensitivity (e.g., a 24/32-bit ADC).

Anyway... everyone should now understand that listening in on remote conversations is no longer difficult or particularly expensive. And if the interior of that area is remotely visible to an eavesdropper it is no longer necessary to pre-install any sort of listening technology within the area of interest

Congrats to the guys at Ben-Gurion University and the Weizmann Institute of Science. I wonder who's reading their paper with interest?

