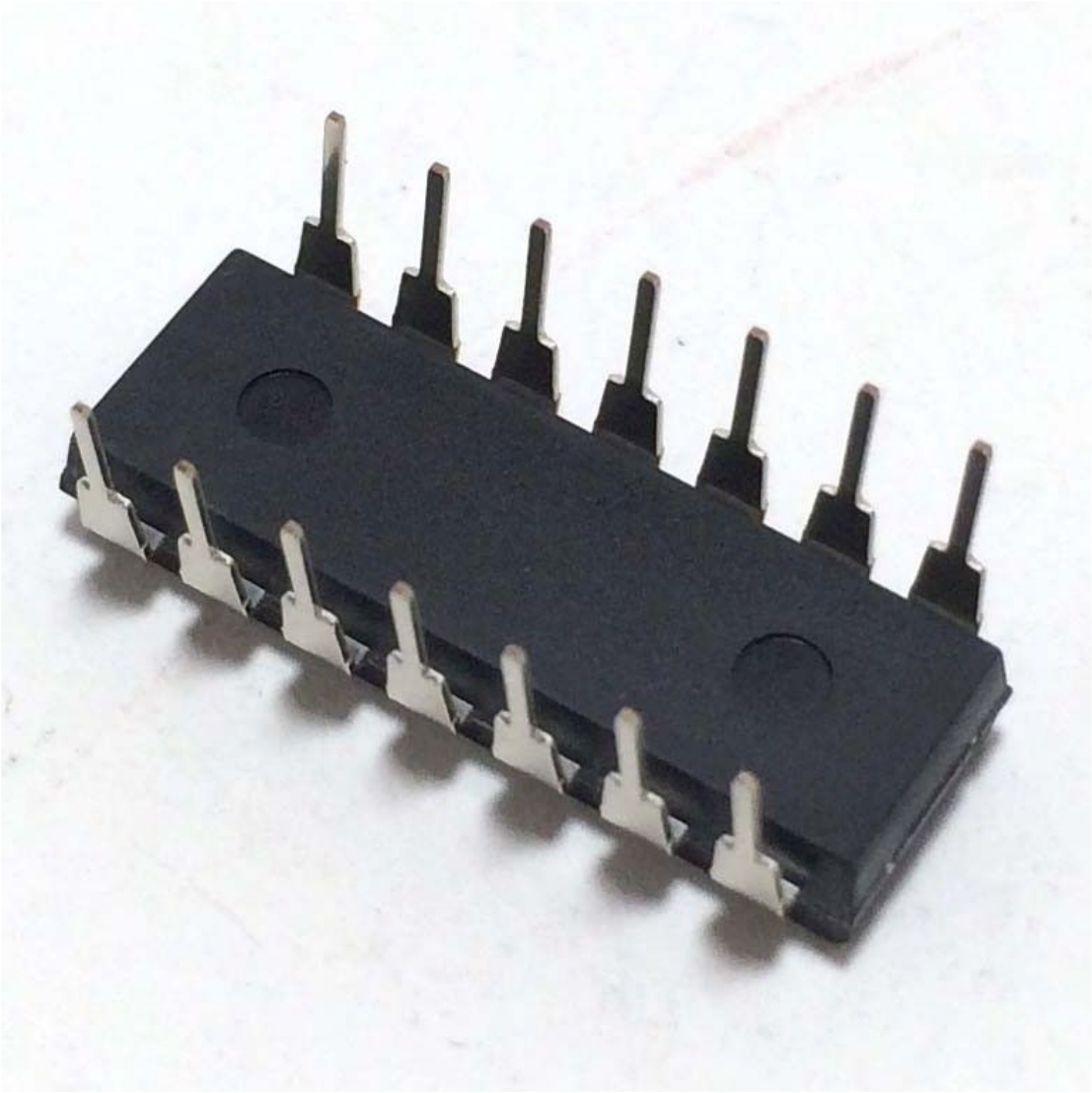
Security Now! #668 - 06-19-18

Lazy FPU State Restore

This week on Security Now!

This week we examine a rather "mega" patch Tuesday, a nifty hack of Win10's Cortana, Microsoft's official "when do we patch" guidelines, the continuing tweaking of web browser behavior for our sanity, a widespread Windows 10 rootkit, the resurgence of the Satori IoT botnet, clipboard monitoring malware, a forthcoming change in Chrome's extensions policy, hacking apparent download counts on the Android store, some miscellany, an update on the status of Spectre & Meltdown... and, yes, yet another brand new speculative execution vulnerability our OSes will be needing to patch against.

"It's Dead, Jim..."



Security News

Microsoft's June Mega-Patch Tuesday:
Last Tuesday Microsoft patched more than 50 vulnerabilities, affecting Windows, Internet Explorer, Edge, MS Office, MS Office Exchange Server, ChakraCore (Edge's JS engine), and Adobe Flash Player.

Among these, 11 were rated "critical" and 39 were "important".

However, there were no Windows 0-days fixed this month.

As hoped, the bad JScript flaw (CVE-2018-8267) =was= fixed.

- Dmitri Kaslov (Telspace Systems) responsibly reported to Trend Micro's Zero-Day Initiative (ZDI).
- Trend Micro forwarded and gave Microsoft four months. / PoC / Went public.

Microsoft also patched that annoyance that keeps on giving -- the Adobe Flash 0-day that was being actively exploited in targeted attack through Microsoft Office (CVE-2018-5002).

But there WERE some VERY worrisome problems fixed last week:

CVE-2018-8225 was a flaw in Windows DNSAPI.dll which affected all versions of Windows starting from 7 through 10, including the server editions. Believe it or not, there was a flaw in the way Windows was parsing DNS responses which could be exploited by sending corrupted DNS responses to a targeted system from an attacker-controlled malicious DNS server. Successful exploitation of this vulnerability would allow an attacker to run arbitrary code in the context of the Local System Account.

Think about this for a second. DNS over UDP without encryption.
DNS is resolved in the kernel where the DNSAPI.dll resides.
This is a remote attack on any Windows machine -- including Windows servers -- that allows a DNS reply to take over the machine and execute attacker-controlled code with kernel privileges.

The big problem is... we KNOW that not all Windows systems -- and especially servers -- are updated immediately.

But wait... there's more!

CVE-2018-8231 was another horrifying remote code execution (RCE) flaw in another publicly-exposed network component, the HTTP protocol stack (HTTP.sys). Fortunately, only Windows 10 and Windows Server 2016 are affected... but it allows remote attackers to execute arbitrary code and take control of the affected systems. Whoopsie! This occurs because HTTP.sys was found to mishandle memory which allows attackers to send a specially crafted packet to an affected Windows system to trigger arbitrary code execution.

Windows systems which are not patched are likely to become targets once the patched changes are reverse engineered and proof of concept code is made widely available.

Microsoft also fixed 7 other critical memory corruption bugs, one in the Chakra scripting engine, another in the ChakraCore, three in Edge, and another in Windows Media Foundation—all which could lead to remote code execution.

Also fixed: Cortana's "Elevation of Privilege" hack.
Until last week (and also until it's patched) Cortana can be leveraged to give an attacker who has physical access to a locked Windows 10 machine full access... including changing the system's password:

The elevation of privilege vulnerability was discovered and responsibly reported by Cedric Cochin of McAfee's Advanced Threat Research (ATR) team.

Microsoft has classified the flaw as "important" because exploitation of this vulnerability requires an attacker to have physical or console access to the targeted system and the targeted system also needs to have Cortana enabled.

Cedric posted their discovery once it had been patched with the title: "Want to Break Into a Locked Windows 10 Device? Ask Cortana (CVE-2018-8140)"

https://securingtomorrow.mcafee.com/mcafee-labs/want-to-break-into-a-locked-windows-10-device-ask-cortana-cve-2018-8140/

<quote> June’s “Patch Tuesday” (June 12) is here, but it is likely many Windows 10 users have not yet applied these updates. If you have not, just be sure not to leave your laptop lying around! The patches in this cycle fix a code execution vulnerability using the default settings for Windows 10 and the “Cortana” voice assistant. We’ll detail how this vulnerability can be used to execute code from the locked screen of a fully patched Windows 10 machine (RS3 at the time of our original submission, and confirmed on RS4 prior to this patch cycle). The vulnerability was submitted to Microsoft as part of the McAfee Labs Advanced Threat Research team’s responsible disclosure policy, on April 23. Attribution for this vulnerability submission goes to Cedric Cochin, Cyber Security Architect and Senior Principle Engineer. </quote>

<quote> This will come as a surprise and lies at the core of all the issues we found, but simply typing while Cortana starts to listen to a query on a locked device will bring up a Windows contextual menu. </quote>

After awakening the machine with a "Hey Cortana" they begin to type and the contextual menus show partial matching. So all files beginning with "pas" -- and also the contents of many files are indexed.

<quote> Next, we asked the question: Could we go a step further and get code execution in the context of the authenticated user? Remember we are using only a combination of voice commands and mouse/touchpad/touchscreen to gain access to the contextual menu at this point. We observed that just by hovering over a file, the full path or content of the file would be displayed. What happens if we were to click on it? That depends on the target. If the file being opened is an application or an executable (such as notepad or calc.exe), the file will run and be accessible only after the user properly logs in. If it is a document, script, or text file, it will be

opened by an editor instead of being executed. At this point we can execute various preloaded Windows utilities such as calculator, but we cannot pass any parameters to the command line. We can open scripts including PowerShell, but instead of being executed, they will be opened in a text editor (notepad). The lack of parameters is a limitation for a “live off the land” attack, which uses current tools and content to achieve a malicious purpose; however, there are plenty of malicious activities that could be performed even with these restrictions. For example, many uninstallers will happily remove software without any need for parameters.

Let’s return to our goal: code execution from the lock screen. The only requirement for something to show up in the contextual menu is for it to be indexed.

They go on to demonstrate how a USB drive can be presented and how, in a clever three-stage attack -- which is now publicly known and documented -- it's possible to get PowerShell to run without UAC protection to give an attacker full access to the user's locked session.

In the absence of patching Windows, the fastest and easiest mitigation technique is to simply turn off Cortana on the lock screen -- which really seems like a sound piece of advice in general.

What's Microsoft's policy on when things are patched versus fixed with a next release?
https://msdnshared.blob.core.windows.net/media/2018/06/Microsoft-Security-Servicing-Commitments_SRD.pdf
Microsoft has released a draft "Microsoft Security Servicing Commitments"

<quote> Our commitment to protecting customers from vulnerabilities in our products, services, and devices includes providing security updates that address these vulnerabilities when they are discovered. We also want to ensure we are transparent with our customers in our approach. This document helps to describe the criteria the Microsoft Security Response Center (MSRC) uses to determine whether a reported vulnerability will be addressed through servicing, or in the next version of a product. For vulnerabilities in products, this servicing takes the form of a security update, most commonly released as security updates on Update Tuesday. The purpose of this document is to clarify the commitments as they pertain to Windows.

Security Servicing Criteria
The criteria used by Microsoft when evaluating whether or not to provide a security update for a reported vulnerability involves answering two key questions

1. Does the vulnerability violate a promise made by a security boundary or a security feature that Microsoft has committed to defending?
2. Does the severity of the vulnerability meet the bar for servicing?

Security boundaries and features with servicing commitments:
Microsoft’s products, services, and devices rely on promises made by a number of security boundaries and security features in order to achieve our security goals.

Security boundaries
A security boundary provides a logical separation between the code and data of security domains with different levels of trust. For example, the separation between kernel mode and user mode is a classic and straightforward security boundary. Microsoft software depends on multiple security boundaries in order to isolate devices on the network, virtual machines, and applications on a device.

- Network Boundary
- Kernel Boundary
- Process Boundary
- AppContainer sandbox boundary
- Logon Session Boundary
- Web browser boundary (sandbox and Same-Origin-Policy)
- Virtual Secure Mode (VSM trustlet or enclave)

All of those receive Microsoft's "servicing commitment" and warrant a Bug Bounty.

Security features (as distinct from "boundaries")
A security feature provides protection against one or more threats. In some cases, a security feature may make a promise related to the threat they are protecting against and there are not expected to be any by design limitations that prohibit delivering on that promise. The following table summarizes the security features that Microsoft has defined that make a promise that has a servicing commitment.

- Device Security: BitLocker & Secure Boot
- Platform security: Windows Defender System Guard
- Application Security: Windows Defender Application Guard
- Identity & access control:Windows Hello / Biometrics / Windows Resource Access Control
- Cryptography API
- Authentication Protocols.

All of those also also receive Microsoft's servicing commitment and Bug Bounty.

However... Microsoft hasa this to say about "Defense-in-Depth" features:

In some cases, a security feature may provide protection against a threat without making a promise. These security features are typically referred to as defense-in-depth features or mitigations because they provide additional security but may have by design limitations that prevent them from making a promise.

A bypass for a defense-in-depth security feature does not pose a direct risk because an attacker must also have found a vulnerability that affects a security boundary, or they must rely on social engineering to achieve the initial stage of a device compromise.

Any vulnerability or bypass that affects these security features will not be serviced by default, but it may be addressed in a future version or release. Many of these features are being continuously improved across each product release and are also covered by active bug bounty programs.

User Safety:
UAC, AppLocker, Controlled Folder Access, Data Execution Prevention

Exploit Mitigations:
ASLR and KASLR, Arbitrary Code Guard (memory protection), DLL signing protection, restriction in child process creation, Safe Structured Exception Handling, Heap randomization, Windows Defender exploit guard. Virtual machine shielding.

Severity of vulnerabilities
The second dimension that Microsoft uses to evaluate whether or not a reported vulnerability should be serviced is based on the severity of the vulnerability. The severity of a vulnerability is determined by mappinh the properties of the vulnerability (impact, scenario, etc.) to its severity. As we know, Microsoft defines five severity levels: Critical, Important, Moderate, Low, and None.

If a vulnerability is rated as Critical or Important, and the vulnerability applies to a security boundary or security feature that has a servicing commitment, then the vulnerability will be addressed through a security update.

Soon... no browsers will be auto-playing obnoxious website audio and video.
Chrome has been blocking auto-play since 66.

Mozilla has committed to adding this to Firefox sometime later this year.

And now Microsoft has stated that Edge will also have this, though NOT enabled by default.

Edge users who seek peace and quiet will need to go into Edge -> Advanced Settings -> "Allow sites to automatically play media" and disable that option.

It should be appearing in the next Insider build. (It just missed making it into the just-released Build 17692, so it should be in the one that follows.)

BitDefender LABS reports on a Win10-based Adware Rootkit
https://labs.bitdefender.com/2018/06/six-years-and-counting-inside-the-complex-zacinlo-ad-fraud-operation/

"Zacinlo" 104 page PDF Whitepaper:
https://labs.bitdefender.com/wp-content/uploads/downloads/six-years-and-counting-inside-the-complex-zacinlo-ad-fraud-operation/

89% Win10, 5.3% Win7, and the rest Win8 and 8.1.

Not that a rootkit in Win10 represents a defeat of Secure Boot, since the entire purpose of Secure Boot is to establish an authoritative chain of execution from the motherboard's first power up to system operation.

Unlike many of the threats we see which are often concentrated in various Asia, European or

Middle Eastern countrties... THIS one is virtually ALL United States targeted.

<quote> Last year we came across a digitally signed rootkit capable of installing itself on most Windows operating systems, including the newest releases of Windows 10. Since rootkits these days account for under 1 percent of the malware output we see worldwide, this immediately drew our attention and prompted us to carry out an extensive analysis of the payload, its origins and the spread. We discovered an ample operation whose central component is a very sophisticated piece of adware with multiple functionalities.

Our information indicates that the adware has been active since 2012-2013. We have identified at least 25 different components found in almost 2,500 distinct samples. While tracking the adware, we noticed some of the components were continuously updated with new functionalities, dropped altogether or integrated entirely in other components. This once again reinforces our inital assumption that the adware is still being developed as of the writing of this paper.

While looking at the communication mechanism of the adware, we identified that a multitude of domains bought from Enom were acting as command-and-control centers. These domains were all registered to two email addresses, included in the IoC chapter at the end of this paper.

The main features of this adware that drew our attention are:

· The presence of a rootkit driver that protects itself as well as its other components. It can stop processes deemed dangerous to the functionality of the adware while also protecting the adware from being stopped or deleted.The presence of man-in-the-browser capabilities that intercepts and decrypts SSL communications. This allows the adware to inject custom JavaScript code into webpages visited by the user.

- It features an adware cleanup routine used to remove potential "competition" in the adware space. This routine is rather generic and does not target a particular family or type of adware.

- The adware can uninstall and delete services based on the instruction it receives from the command and control infrastructure.

- It reports some information about the environment it is running in to the C&C. This information includes whether an antimalware solution is installed (and if so, which one), which applications are running at start-up and so on.

- It takes screen captures of the desktop and sends them to the command and control center for analysis. This functionality has a massive impact on privacy as these screen captures may contain sensitive information such as e-mail, instant messaging or e-banking sessions.

- It can accommodate the installation of virtually any piece of software on the fly and thus extend its functionality.

- It features an automatic update mechanism.

- It redirects pages in browsers
- It adds or replaces advertisments while browsing by searching DOM objects by size, style, class or specific regular expressions
- Uses many platforms to pull advertising from advertising, including Google AdSense.
- Obsolete or expired ads can be easily replaced by new ones
- Silently renders webpages in the background in hidden windows and interracts with them as a normal user would: scrolling, clicking, keyboard input. This is typical behavior for advertising fraud that inflicts significant financial damage on online advertising platforms.
- Its extensive use of open-source projects and libraries (e.g: chromium, cryptopop, jsoncpp, libcef, libcurl, zlib, etc.)
- It uses Lua scripts to download several components (most likely as a way to fly under the radar of some antimalware solutions that detect suspicious downloads and block them as such)
- Extremely configurable and highly modular design that can expand functionality via scripts and configuration files made available via the command and control infrastructure

How does this nightmare get into a system?
The adware components are silently installed by a downloader that is presented as a free and anonymous VPN service (s5Mark), distributed in an installer. s5Mark has a simple graphical interface used as a decoy for the intrusive unwanted behavior taking place behind the scenes. Note that a non-technical user is led to believe that a VPN connection is established even though no such thing is even attempted.

Port 8000 scanning traffic jumped during the past week...
....as our old friend the Satori IoT Botnet went into overdrive.

Working backward from the evidence, the guys at China's 360 Netlab realized that PoC code to exploit a buffer overflow vulnerability (CVE-2018-10088) in XionMai uc-httpd 1.0.0 -- which is a lightweight web server package often found embedded inside the firmware of routers and IoT equipment sold by Chinese vendors.

These HTTPd server are listening on port 8000 (rather than 80 or 8080) and a malformed packet can takeover the devices.

The Internet-wide search for devices with port 8000 exposed began a day after the PoC's publication and it caught the attention of security groups specializing in botnet tracking who began tweeting their "WTF?" questions as this scanning exploded onto the Internet at a huge rate.
According to honeypot data from the 360 Netlab folks and SANS, port 8000 scans have since subsided. But this was only because Satori had changed its focus to D-Link DSL-2750B devices

for which a PoC was published earlier. It turns out that the D-Link devices can be exploited via ports 80 and 8080.

It appears that the Satori gang are working hard to commandeer as many routers as they can before other botnets get into the act.

As we know, it had previously targeted the optical network GPON routers. And with the addition of these additional exploits Satori continues to grow. It's authors appear to be intent upon maintaining and growing their network. Satori hijacks crytocurrency miners, steals funds and launched DDoS attacks.

And speaking of stealing Cryptocurrency...
Another bit of nasty clipboard monitoring malware reportedly has more than 300,000 victims.

360 Total Security Blog:

https://blog.360totalsecurity.com/en/new-cryptominer-hijacks-your-bitcoin-transaction-over-300000-computers-have-been-attacked/

New CryptoMiner hijacks your Bitcoin transaction. Over 300,000 computers have been attacked.

360 Security Center has discovered a new type of actively spreading CryptoMiner: ClipboardWalletHijacker.

The Trojan monitors clipboard activity to detect when it contains the account address of Bitcoin and Ethereum. It then replaces the receiving address with its own address to redirect the cryptocurrency to its own wallet. This Trojan has been detected on more than 300 thousand computers within a week.

The replacement address is “0x004D3416DA40338fAf9E772388A93fAF5059bFd5”. There have been 46 successful transactions in total.

If the address is not Ethereum, the Trojan checks if it is Bitcoin address, and the address number begins with 1 or 3. If the current date is earlier than 8th of the month, replace the address to “19gdjoWaE8i9XPbWoDbixev99MvvXUSNZL”. Otherwise, use “1FoSfmjZJFqFSsD2cGXuccM9QMMa28Wrn1” instead.

The Trojan has successfully hijacked five Bitcoin transaction already. The amount of the latest transaction is 0.069 BTC (approximately equivalent to 500 US dollars).

Google Blocks Chrome Extension Installations From 3rd-Party Sites
https://thehackernews.com/2018/06/chrome-extension-intallation.html
https://blog.chromium.org/2018/06/improving-extension-transparency-for.html

3rd-party websites often offer visitors using Chrome the option of installing a Chrome extension to enhance their experience.

Well... Google's Chromium blog from last week is titled: "Improving extension transparency for users" and, in short, aims to wind down that practice.

We strive to ensure choice and transparency for all Chrome users as they browse the web. Part of this choice is the ability to use the hundreds of thousands of extensions available in the Chrome Web Store to customize the browsing experience in useful and productivity-boosting ways. However, we continue to receive large volumes of complaints from users about unwanted extensions causing their Chrome experience to change unexpectedly — and the majority of these complaints are attributed to confusing or deceptive uses of inline installation on websites. As we've attempted to address this problem over the past few years, we've learned that the information displayed alongside extensions in the Chrome Web Store plays a critical role in ensuring that users can make informed decisions about whether to install an extension. When installed through the Chrome Web Store, extensions are significantly less likely to be uninstalled or cause user complaints, compared to extensions installed through inline installation. Later this summer, inline installation will be retired on all platforms. Going forward, users will only be able to install extensions from within the Chrome Web Store, where they can view all information about an extension's functionality prior to installing. This change will roll out in three phases:

Starting today, inline installation will be unavailable to all newly published extensions. Extensions first published on June 12, 2018 or later that attempt to call the chrome.webstore.install() function will automatically redirect the user to the Chrome Web Store in a new tab to complete the installation.

Starting September 12, 2018, inline installation will be disabled for existing extensions, and users will be automatically redirected to the Chrome Web Store to complete the installation.

In early December 2018, the inline install API method will be removed from Chrome 71.

If you distribute an extension using inline installation, you will need to update install buttons on your website to link to your extension's Chrome Web Store page prior to the stable release of Chrome 71. And if you haven't already, be sure to read up on how to create a high quality store listing, and consider using our install badge on your site.

And speaking of misleading and deceptive...
Cheesy developers -- or at least developers of cheesy Android applications -- have figured out how to mislead people about the popularity of their applications.

It's possible for an Android author to set their name to "more than 1,000,000 downloads" to have that deception show underneath the application's icon. And others are even embedding the fraudulent download count =IN= their application's icon. The result is that hurried and

unsuspecting users will assume that an application with absolutely no "reputation" is super-popular.

Miscellany:

SQRL on the SQRL forums.

SpinRite

Brian in Albany, NY
Subject: Spinrite "reboot after scan completed"
Date: 05 Jun 2018 03:19:50
:
Leo: Long time listener, sometime feed-back provider.

Steve: I am not sure if this will make the June 5th episode or not.

Is there a reboot option in Spinrite 6? I have some machines that I would love to run this on for maintenance. If only, there is an option in settings to reboot after scan.

This would allow the Spinrite operator to start a scan and walk away. So long as there are no issues. If the machine would reboot after 100% of scan, the Spinrite operator would not have to intercept the machine prior to the user coming into work at 8am. I am using the ISO made from the exe.

Is this a possibility in a future release? I searched the SpinRite manual and there is no mention of this that I could find.

All the Best.

Brian in Capitaland, NY

Meltdown, Spectre & Lazy Restores

As of last Tuesday's patches...

- Windows 7, 8.1 & 10 have Meltdown mitigations ENABLED by default.
- They also have the first two Spectre variation mitigations ENABLED by default.
- Windows 7 and 10 have the most recent SpectreNG mitigation available but DISABLED by default.
- Windows 8.1 does not have SpectreNG mitigation available at all.
- ALL of the server variants (Server 2008 Rs, Server 2012 R2 and Server 2016) have ALL of the mitigations disabled by default.

And... there are some weird side effects being induced by the Spectre and Meltdown patches:

1) Some non-English platforms may display the following string in English instead of the localized language: ”Reading scheduled jobs from file is not supported in this language mode.” This error appears when you try to read the scheduled jobs you've created and Device Guard is enabled.

2) When Device Guard is enabled, some non-English platforms may display the following strings in English instead of the localized language:

- "Cannot use '&' or '.' operators to invoke a module scope command across language boundaries."
- "'Script' resource from 'PSDesiredStateConfiguration' module is not supported when Device Guard is enabled. Please use 'Script' resource published by PSDscResources module from PowerShell Gallery."

Some users running Windows 10 version 1803 may receive an error "An invalid argument was supplied" when accessing files or running programs from a shared folder using the SMBv1 protocol.

1) A stop error occurs on computers that don't support Streaming Single Instructions Multiple Data (SIMD) Extensions 2 (SSE2).

2) There is an issue with Windows and third-party software that is related to a missing file (oem< number >.inf). Because of this issue, after you apply this update, the network interface controller will stop working.

Lazy FP state restore
https://www.intel.com/content/www/us/en/security-center/advisory/intel-sa-00145.html
Some German security researchers came up with yet another clever way of leveraging an Intel Core optimization:

Product family: Intel® Core-based microprocessors.

AMD processors are not affected by this.
Recent versions of Linux—from kernel version 4.9, released in 2016, and later are not affected by this flaw.
Only older versions of the Linux kernel might be at risk.

And recent versions of Windows, including Server 2016, and latest editions of OpenBSD and DragonflyBSD are not affected by this flaw.

Microsoft has published a security advisory, offering guidance for the Lazy FP State Restore vulnerability and explaining that the company is already working on security updates to be released on July's Patch Tuesday.

Microsoft says that Lazy restore is enabled by default in Windows and cannot be disabled, adding that virtual machines, kernel, and processes are affected by this vulnerability. However, customers running virtual machines in Azure are not at risk.

System software may utilize the Lazy FP state restore technique to delay the restoring of state until an instruction operating on that state is actually executed by the new process.

Systems using Intel® Core-based microprocessors may potentially allow a local process to infer data utilizing Lazy FP state restore from another process through a speculative execution side channel.

System software may opt to utilize Lazy FP state restore instead of eager save and restore of the state upon a context switch.

Lazy restored states are potentially vulnerable to exploits where one process may infer register values of other processes through a speculative execution side channel that infers their value.

Unlike Spectre and Meltdown, the latest vulnerability does not reside in the hardware. So, the flaw can be fixed by pushing patches for various operating systems without requiring new CPU microcodes from Intel.

~30~