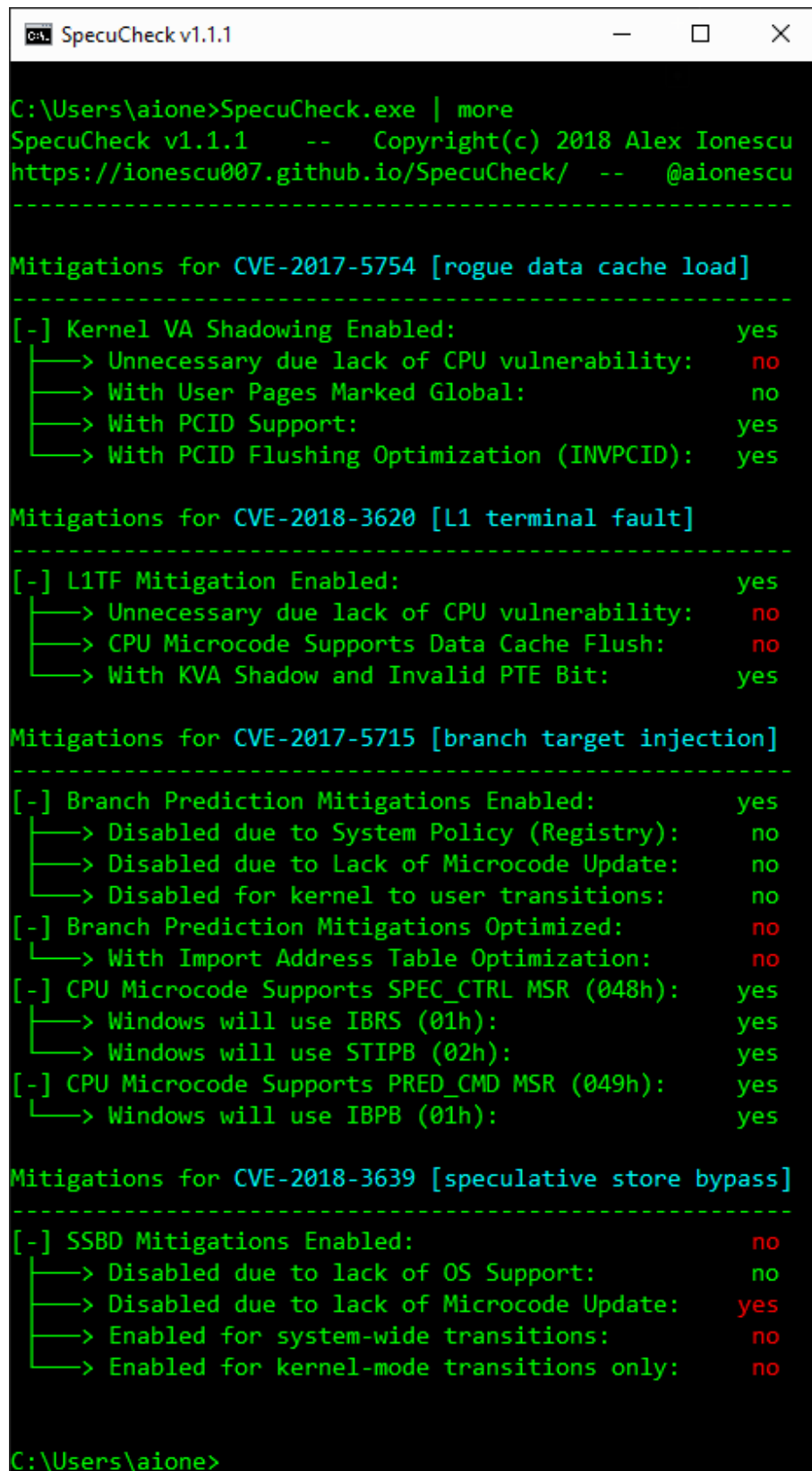

SpecuCheck

SpecuCheck is a Windows utility for checking the state of the software and hardware mitigations against CVE-2017-5754 (Meltdown), CVE-2017-5715 (Spectre v2), CVE-2018-3260 (Foreshadow), and CVE-2018-3639 (Spectre v4). It uses two new information classes that were added to the NtQuerySystemInformation API call as part of the recent patches introduced in January 2018 and reports the data as seen by the Windows Kernel.

An official Microsoft Powershell Cmdlet Module now exists as well, which is the recommended and supported way to get this information.

Screenshots



```
C:\Users\ai>SpecuCheck.exe | more
SpecuCheck v1.1.1    --    Copyright(c) 2018 Alex Ionescu
https://ionescu007.github.io/SpecuCheck/  --    @aionescu
-----

Mitigations for CVE-2017-5754 [rogue data cache load]
-----
[-] Kernel VA Shadowing Enabled:                yes
  └─> Unnecessary due lack of CPU vulnerability:  no
  └─> With User Pages Marked Global:              no
  └─> With PCID Support:                          yes
  └─> With PCID Flushing Optimization (INVPCID): yes

Mitigations for CVE-2018-3620 [L1 terminal fault]
-----
[-] L1TF Mitigation Enabled:                    yes
  └─> Unnecessary due lack of CPU vulnerability:  no
  └─> CPU Microcode Supports Data Cache Flush:    no
  └─> With KVA Shadow and Invalid PTE Bit:         yes

Mitigations for CVE-2017-5715 [branch target injection]
-----
[-] Branch Prediction Mitigations Enabled:       yes
  └─> Disabled due to System Policy (Registry):    no
  └─> Disabled due to Lack of Microcode Update:    no
  └─> Disabled for kernel to user transitions:      no
[-] Branch Prediction Mitigations Optimized:     no
  └─> With Import Address Table Optimization:      no
[-] CPU Microcode Supports SPEC_CTRL MSR (048h): yes
  └─> Windows will use IBRS (01h):                 yes
  └─> Windows will use STIPB (02h):                 yes
[-] CPU Microcode Supports PRED_CMD MSR (049h):  yes
  └─> Windows will use IBPB (01h):                 yes

Mitigations for CVE-2018-3639 [speculative store bypass]
-----
[-] SSBD Mitigations Enabled:                    no
  └─> Disabled due to lack of OS Support:           no
  └─> Disabled due to lack of Microcode Update:     yes
  └─> Enabled for system-wide transitions:          no
  └─> Enabled for kernel-mode transitions only:     no

C:\Users\ai>
```

Introduction

On January 3rd 2018, Intel, AMD and ARM Holdings, as well as a number of OS Vendors reported a series of vulnerabilities that were discovered by Google Project Zero:

- Variant 1: bounds check bypass (CVE-2017-5753)
- Variant 2: branch target injection (CVE-2017-5715)
- Variant 3: rogue data cache load (CVE-2017-5754)

Microsoft released patches for Windows 7 SP1 and higher later that same day. These patches, depending on architecture, OS version, boot settings and a number of hardware-related properties, apply a number of software and hardware mitigations against these issues. The enablement state of these mitigations, their availability, and configuration is stored by the Windows kernel in a number of global variables, and exposed to user-mode callers through an undocumented system call.

Additionally, new side channel attacks were reported, such as Spectre Variant 4: speculative store bypass (CVE-2018-3639) and Foreshadow: L1 terminal fault (CVE-2018-3620) which were fixed in Windows 7 SP1 and higher with patches in August's Patch Tuesday.

SpecuCheck takes advantage of this system call in order to confirm if a system has indeed been patched (non-patched systems will fail the call) and what the status of the mitigations are, which can be used to determine potential performance pitfalls.

Motivation

There was originally a lot of noise, hype, and marketing around the issue, and not a lot of documentation on how to see if you were affected, and at what performance overhead. SpecuCheck aimed to make that data easily accessible by users and IT departments, to avoid having to use a Windows debugger or reverse engineer the API themselves.

Since then, Microsoft has done great work to expose that data from the kernel-mode in a concise matter, which succinctly indicates the kernel's support and usage of the various mitigating technologies and hardware features, and released a PowerShell CmdLet Module to retrieve that data. SpecuCheck, therefore, remains only as a research tool and is not recommended – please use the Microsoft-approved PowerShell Module instead.

Installation on Windows

To run SpecuCheck, simply execute it on the command-line:

```
c:\SpecuCheck.exe
```

Which will result in an informational screen indicating which features/mitigations are enabled. If you see the text:

Your system either does not have the appropriate patch, or it may not support the information **class** required

This indicates that your system is not currently patched to mitigate against these vulnerabilities.

References

If you would like to know more about my research or work, I invite you to check out my blog at <http://www.alex-ionscu.com> as well as my training & consulting company, Winsider Seminars & Solutions Inc., at <http://www.windows-internals.com>.

You should also definitely read the incredibly informative Project Zero Post.

For additional information on the appropriate and required Windows patches, please read the Microsoft Advisory and additional Microsoft Guidance.

Caveats

SpecuCheck relies on undocumented system calls and information classes which are subject to change. Additionally, SpecuCheck only returns the information that the Windows Kernel is storing about the state of the mitigations and hardware features – based on policy settings (registry, boot parameters) or other compatibility flags, the Windows Kernel's state may not match the true hardware state. The goal of this tool is to give you a Windows-specific assessment, not a hardware assessment that is OS-agnostic.

SpecuCheck is only a research tool and is not recommended for general or commercial use – please use the Microsoft-approved PowerShell Module instead.

License

```
1 Copyright 2018 Alex Ionescu. All rights reserved.
2
3 Redistribution and use in source and binary forms, with or without
  modification, are permitted provided
4 that the following conditions are met:
5 1. Redistributions of source code must retain the above copyright
  notice, this list of conditions and
6 the following disclaimer.
7 2. Redistributions in binary form must reproduce the above copyright
  notice, this list of conditions
```

8 and the following disclaimer in the documentation and/or other
9 materials provided with the
10 distribution.

11 THIS SOFTWARE IS PROVIDED BY ALEX IONESCU ``AS IS'' AND ANY EXPRESS OR
12 IMPLIED
13 WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF
14 MERCHANTABILITY AND
15 FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL ALEX
16 IONESCU
17 OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL
18 , EXEMPLARY, OR
19 CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF
20 SUBSTITUTE GOODS
21 OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION)
22 HOWEVER CAUSED
23 AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY,
24 OR TORT (INCLUDING
25 NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS
26 SOFTWARE, EVEN IF
27 ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

28 The views and conclusions contained in the software and documentation
29 are those of the authors and
30 should not be interpreted as representing official policies, either
31 expressed or implied, of Alex Ionescu.