Microsoft

Windows SMM Security Mitigations Table

Version 1.0 - April 18, 2016

Abstract

This specification contains details of an Advanced Configuration and Power Interface (ACPI) table that was created for use with Windows operating systems that support Windows virtualization-based security (VBS) features.

This information applies for the following operating systems:

Windows Server Technical Preview 2016
Windows 10, version 1607

For the latest information, see:

http://www.microsoft.com/whdc/system/platform/virtual

This document is published under the Microsoft Community Promise.

WINDOWS SMM SECURITY MITIGATIONS TABLE (WSMT) ACPI TABLE SPECIFICATION LICENSE

IMPORTANT—READ CAREFULLY: This Microsoft License Agreement ("Agreement") is a legal agreement between you (either an individual or a single entity) and Microsoft Corporation for the version of the Microsoft specification identified above which you are about to download ("Specification"). BY DOWNLOADING, COPYING OR OTHERWISE USING THE SPECIFICATION, YOU AGREE TO BE BOUND BY THE TERMS OF THIS AGREEMENT. IF YOU DO NOT AGREE TO THE TERMS OF THIS AGREEMENT, DO NOT DOWNLOAD, COPY, OR USE THE SPECIFICATION.

The Specification is owned by Microsoft or its suppliers and is protected by copyright laws and international copyright treaties, as well as other intellectual property laws and treaties. **THE SPECIFICATION IS LICENSED, NOT SOLD.**

1. **GRANT OF LICENSE**.

   (a) Provided that you comply with all terms and conditions of this Agreement, Microsoft grants to you the following non-exclusive, worldwide, royalty-free, perpetual, non-transferable, non-sublicensable, limited license under any copyrights or patents that cover the table values described in the Specification and that are owned or licensable by Microsoft without payment of consideration to third parties,

      (i) to reproduce copies of the Specification for your and your contractor's internal use for the sole purpose of (1) modifying your firmware and/or BIOS for computing devices (**"Firmware"**) so that it writes to memory the appropriate table values in the Specification or (2) modifying your software so that it may read from memory the appropriate table values (the **"Purpose"**),

      (ii) to implement the table values in your firmware and/or BIOS,

      (iii) to license to third parties directly and indirectly the table values as part of your Firmware (and any related documentation),

      The foregoing license is granted only to the extent necessary to accomplish the Purpose and to license and/or distribute your Firmware containing the table values to third parties. The foregoing license shall not extend to any features of your Firmware that (i) are not required to comply with the Specification or (ii) to which there was a practicable alternative to infringing a patent.

   (b) Microsoft reserves all other rights it may have in the Specification, its implementation and any intellectual property therein. The furnishing of this document does not give you or any other entity any license to any other Microsoft patents, trademarks, copyrights or other intellectual property rights. Microsoft does not grant to you or any other entity any implied licenses or rights whatsoever under this Agreement. Specifically, this Agreement does not grant any express or implied licenses or rights to any enabling technologies that may be necessary to fully utilize the tables or the table values described in the Specification.

2. **ADDITIONAL LIMITATIONS AND OBLIGATIONS**.

   (a) You may use all or some of the tables provided, however for each table that you use, You must implement such table in its entirety (i.e. all fields) and without modification (e.g., byte length, offset, and permissible values as described in the Specification).

   (b) Your license rights to the Specification are conditioned upon you not creating, modify, or distributing your Firmware in a way that such creation, modification, or distribution may (a) create, or purport to create, obligations for Microsoft with respect to the Specification (or intellectual property therein) or (b) grant, or purport to grant, to any third party any rights or immunities to Microsoft's intellectual property or proprietary rights in the Specification.

   (c) The foregoing license is applicable only to the version of the Specification which you are about to download. It does not apply to any additional versions of or extensions to the Specification.

   (d) Without prejudice to any other rights, Microsoft may terminate this Agreement if you fail to comply with the terms and conditions of this Agreement. In such event you must destroy all copies of the Specification and must not further distribute the table values.

3. **INTELLECTUAL PROPERTY RIGHTS**. All ownership, title and intellectual property rights in and to the Specification, and any copies you are permitted to make herein, are owned by Microsoft or its suppliers

4. **DISCLAIMER OF WARRANTIES. To the maximum extent permitted by applicable law, Microsoft and its suppliers provide the Specification (and all intellectual property therein) and any (if any) support services related to the Specification ("Support Services") *AS IS AND WITH ALL FAULTS*, and hereby disclaim all warranties and conditions, either express, implied or statutory, including, but not limited to, any (if any) implied warranties or conditions of merchantability, of fitness for a particular purpose, of lack of viruses, of accuracy or completeness of responses, of results, and of lack of negligence or lack of workmanlike effort, all with regard to the Specification, any intellectual**

property therein and the provision of or failure to provide Support Services. ALSO, THERE IS NO WARRANTY OR CONDITION OF TITLE, QUIET ENJOYMENT, QUIET POSSESSION, CORRESPONDENCE TO DESCRIPTION OR NON-INFRINGEMENT, WITH REGARD TO THE SPECIFICATION AND ANY INTELLECTUAL PROPERTY THEREIN. THE ENTIRE RISK AS TO THE QUALITY OF OR ARISING OUT OF USE OR PERFORMANCE OF THE SPECIFICATION, ANY INTELLECTUAL PROPERTY THEREIN, AND SUPPORT SERVICES, IF ANY, REMAINS WITH YOU.

5. **EXCLUSION OF INCIDENTAL, CONSEQUENTIAL AND CERTAIN OTHER DAMAGES. To the maximum extent permitted by applicable law, in no event shall Microsoft or its suppliers be liable for any special, incidental, indirect, or consequential damages whatsoever (including, but not limited to, damages for loss of profits or confidential or other information, for business interruption, for personal injury, for loss of privacy, for failure to meet any duty including of good faith or of reasonable care, for negligence, and for any other pecuniary or other loss whatsoever) arising out of or in any way related to the use of or inability to use the SPECIFICATION, ANY INTELLECTUAL PROPERTY THEREIN, the provision of or failure to provide Support Services, or otherwise under or in connection with any provision of this AGREEMENT, even in the event of the fault, tort (including negligence), strict liability, breach of contract or breach of warranty of Microsoft or any supplier, and even if Microsoft or any supplier has been advised of the possibility of such damages.**

6. **LIMITATION OF LIABILITY AND REMEDIES. Notwithstanding any damages that you might incur for any reason whatsoever (including, without limitation, all damages referenced above and all direct or general damages), the entire liability of Microsoft and any of its suppliers under any provision of this Agreement and your exclusive remedy for all of the foregoing shall be limited to the greater of the amount actually paid by you for the Specification or U.S.$5.00. The foregoing limitations, exclusions and disclaimers shall apply to the maximum extent permitted by applicable law, even if any remedy fails its essential purpose.**

7. **APPLICABLE LAW**. If you acquired this Specification in the United States, this Agreement is governed by the laws of the State of Washington. In respect of any dispute which may arise hereunder, you consent to the jurisdiction of the state and federal courts sitting in King County, Washington.

8. **ENTIRE AGREEMENT. This Agreement is the entire agreement between you and Microsoft relating to the Specification supersedes all prior or contemporaneous oral or written communications, proposals and representations with respect to the Specification or any other subject matter covered by this Agreement. To the extent the terms of any Microsoft policies or programs for Support Services conflict with the terms of this Agreement, the terms of this Agreement shall control.**

Copyright

This document is provided "as-is." Information and views expressed in this document, including URL and other Internet Web site references, may change without notice.

Some examples depicted herein are provided for illustration only and are fictitious. No real association or connection is intended or should be inferred.

This document does not provide you with any legal rights to any intellectual property in any Microsoft product. You may copy and use this document for your internal, reference purposes.

Please refer to Microsoft Trademarks for a list of trademarked products.

Portions of this software may be based on NCSA Mosaic. NCSA Mosaic was developed by the National Center for Supercomputing Applications at the University of Illinois at Urbana-Champaign. Distributed under a licensing agreement with Spyglass, Inc.

All other trademarks are property of their respective owners.

Document History

Date	Change
January 21, 2016	First draft, Version 0.9
April 4, 2016	Revised to incorporate feedback, Version 0.9b
April 12, 2016	Revised for additional clarity, Version 0.9c
April 18, 2016	Released

Contents

Introduction

Microsoft introduced virtualization-based security (VBS) in Windows 10. Virtualization-based security utilizes the Microsoft hypervisor and other system level software in conjunction with the platform's hardware virtualization extensions to create a secure environment in which various security-related applications and functions can run.

For example, VBS allows for a hypervisor based Secure Kernel to interpose on the operation of kernel mode (CPL0) components in order to uphold overall system integrity after boot. This is achieved by creating a separate operating system context, controlled by the hypervisor and enforced in hardware virtualization features, and using this separate "synthetic" privilege level for applying security policies and protecting critical data.

The advent of VBS changes the traditional trust boundaries and security model on which System Management Mode (SMM[1]) may have made assumptions. Historically, in many implementations of system firmware, SMM has treated all kernel OS components running at CPL0 as trusted code. Specifically, the traditional operating system is considered to be in the same trust boundary as SMM.

With the introduction of VBS, this no longer holds true. In the security model expected with VBS, the platform firmware executing in SMM must be deemed trusted by VBS, and the traditional operating system components executing in the kernel in CPL0 are not. In fact, SMM must now treat kernel OS components in CPL0 as potentially hostile.

To properly protect the secure environment created by VBS from exploits mounted via SMM, a number of changes to SMM programming practices and assumptions must be introduced. Without these changes, the Windows OS may choose to degrade or disable specific security features or capabilities in order to ensure the integrity of platform security assets, and provide a robust and reliable platform on which to develop new end user scenarios. However, the OS must be able to determine what SMM security mitigations have been implemented on a specific platform. Since SMM is opaque to the OS, and since the OS must assume SMM is within the same trust domain as the OS itself, the OS must rely on SMM firmware to accurately self-report which of the Microsoft recommended security best practices it has implemented. To accomplish this, Microsoft has defined the Windows SMM Security Mitigations Table (WSMT).

This paper serves to describe the format of the WSMT table, wherein firmware communicates specific patterns of protections that it has enabled and enforcements that it has put in place, in acknowledgement of this new trust boundary.

Support for the WSMT is included in the following versions of Windows:

- Windows Server Technical Preview 2016
- Windows 10, version 1607

[1] For the purposes of this document, SMM and ARM TrustZone are synonymous.

This document describes the layout of the WSMT table, together with a more detailed description of the layout and contents of the WSMT Protections Flags field.

Windows SMM Security Mitigations Table

To use the WSMT, system firmware must create a static WSMT table in the ACPI namespace of the platform. Supported versions of Windows operating systems read the WSMT early during initialization, prior to start of the ACPI interpreter and subsequent evaluation of the _OSI method. The Protection Flags field indicates the presence of specific BIOS security mitigations in system firmware. Firmware setting any of the Protections Flags represents an attestation by the platform to OSPM that the corresponding firmware feature or coding practice has been implemented. OSPM may not be able to functionally validate that this is indeed the case, and must rely on system firmware to accurately represent its capabilities. Windows operating systems may elect to enable, disable, or de-feature certain security features based on the presence of these SMM Protections Flags.

Windows SMM Security Mitigations Table Format

Table 1 details the layout of the WSMT.

Table 1. Windows SMM Security Mitigations Table

Field	Byte length	Byte offset	Description
ACPI Standard Header			
Signature	4	0	Signature for the WSMT
Length	4	4	Length, in bytes, of the WSMT. Must be 40 for Revision 1.
Revision	1	8	1
Checksum	1	9	Entire table, which must sum to zero
OEMID	6	10	Original equipment manufacturer (OEM) identifier (ID)
OEM Table ID	8	16	Manufacturer model ID
OEM Revision	4	24	OEM revision for supplied OEM table ID
Creator ID	4	28	Vendor ID of the ASL compiler utility that created the table
Creator Revision	4	32	Revision of the ASL compiler utility that created the table
Protection Flags	4	36	Container of a bitmask of the system implemented WSMT protections. Bits in this field represent that certain protections/enforcements are enabled and active for firmware executing in SMM context after ExitBootServices(). See Table 2 for a detailed description of this field.

WSMT Protections Flags

Table 2 describes the bit definitions for the Protections Flags field of the WSMT.

Table 2. Protection Flags Field

Length	Bit offset	Description
1	0	FIXED_COMM_BUFFERS If set, expresses that for all synchronous SMM entries, SMM will validate that input and output buffers lie entirely within the expected fixed memory regions.
1	1	COMM_BUFFER_NESTED_PTR_PROTECTION If set, expresses that for all synchronous SMM entries, SMM will validate that input and output pointers embedded within the fixed communication buffer only refer to address ranges that lie entirely within the expected fixed memory regions.
1	2	SYSTEM_RESOURCE_PROTECTION Firmware setting this bit is an indication that it will not allow reconfiguration of system resources via non-architectural mechanisms.
	31:3	Reserved; must return 0 when read.

WSMT Protections Flags Details

FIXED_COMM_BUFFERS – Firmware setting this bit should refer to the SMM Communication ACPI Table defined in the UEFI 2.6 specification. Firmware should also consider all other possible data exchanges between SMM and non-SMM, including but not limited to EFI_SMM_COMMUNICATION_PROTOCOL, ACPINVS in ASL code, general purpose registers as buffer pointers, etc.

COMM_BUFFER_NESTED_PTR_PROTECTION – Firmware setting this bit must also set the FIXED_COMM_BUFFERS bit.

SYSTEM_RESOURCE_PROTECTION – After ExitBootServices(), firmware setting this bit shall not allow any software to make changes to the locations of: IOMMU's, interrupt controllers, PCI Configuration Space, the Firmware ACPI Control Structure (FACS), or any registers reported through ACPI fixed tables (e.g. PMx Control registers, reset register, etc.). This also includes disallowing changes to RAM layout and ensuring that decodes to RAM and any system resources as described above take priority over software configurable registers. For example, if software configures a PCI Express BAR to overlay RAM, accesses by the CPU to the affected system physical addresses must decode to RAM.