|  |
| --- |
| DOCUMENT NUMBER MP130351 |
| MITRE PRODUCT |



The views, opinions and/or findings contained in this report are those of The MITRE Corporation and should not be construed as an official government position, policy, or decision, unless designated by other documentation.

MITRE document number MP130351

©2013 The MITRE Corporation.   
All rights reserved.

Approved for public release. Distribution Unlimited 13-2687

Copernicus User/Installation Guide

Sam Cornwell, John Butterworth, Corey Kallenberg, Xeno Kovah

June 2013

Executive Summary

Copernicus is a tool to evaluate BIOS security on Intel PCs. A writable BIOS opens a system up to a very stealthy, powerful, and persistent backdoor. It allows the possibility that the attacker can brick the system (make it unbootable) on demand. Unlike attacks on software, this firmware attack cannot be recovered from without physical removal of the BIOS firmware chip.

The Copernicus agent takes two actions to evaluate BIOS vulnerability. First it dumps the contents of the flash chip. Second it checks access control registers that determine whether the BIOS is writable or not. This data can then be analyzed. The BIOS dump can be compared against a known clean copy to check for the presence of implants. The control registers can be be analyzed to determine of the BIOS is writable and therefore vulnerable.

This document includes information about installation, analysis, and the data collected.

Table of Contents

[1 System Evaluation 1-1](#_Toc360638177)

[1.1 Obtaining BIOS dump & BIOS configuration files 1-1](#_Toc360638178)

[1.2 Analysis 1-1](#_Toc360638179)

[1.2.1 Protections 1-1](#_Toc360638180)

[Appendix A CSV Output Field Descriptions A-1](#_Toc360638181)

[A.1 General Information A-1](#_Toc360638182)

[A.2 BIOS Information A-1](#_Toc360638183)

[A.3 SMM Information A-2](#_Toc360638184)

[Appendix B Abbreviations B-1](#_Toc360638185)

[B.1 List of Abbreviations B-1](#_Toc360638186)

List of Tables

Table A-1. General System Identification & Compatibility Information A-1

Table A-2. BIOS Access Control Information A-1

Table A-3. SMRAM Access Control Information A-2

This page intentionally left blank.

# Disclaimer

This software is the copyrighted work of MITRE.  No ownership or other proprietary interest in this software is granted to you other than what is granted in this license.

MITRE IS PROVIDING THE SOFTWARE "AS IS" AND ACCORDINGLY MAKES NO WARRANTY, EXPRESS OR IMPLIED, AS TO THE ACCURACY, CAPABILITY, EFFICIENCY, MERCHANTABILITY, OR FUNCTIONING OF THE SOFTWARE AND DOCUMENTATION.  IN NO EVENT WILL MITRE BE LIABLE FOR ANY GENERAL, CONSEQUENTIAL, INDIRECT, INCIDENTAL, EXEMPLARY, OR SPECIAL DAMAGES RELATED TO THE SOFTWARE, EVEN IF MITRE HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

You accept this software on the condition that you indemnify and hold harmless MITRE, its Board of Trustees, officers, agents, and employees, from any and all liability damages to third parties, including attorneys' fees, court costs, and other related costs and expenses, arising out of your use of this software irrespective of the cause of said liability.

# System Evaluation

This section describes how to obtain a sample from a system and then run some post-processing scripts across it (or several samples) for analysis.

## Obtaining BIOS dump & BIOS configuration files

1. Ensure that copernicus.sys, run.bat, and standalone.bat are in the same directory.
2. To run Copernicus, right-click standalone.bat and run as Administrator.
3. The tool will typically take up to 30 seconds to run.
4. Once complete, the output files Copernicus\_Log.txt, Copernicus\_CSV\_Out.csv, and Copernicus\_BIOS.bin will be found in C:\.

It is possible that a Copernicus\_BIOS.bin file will not be generated on some machines, because it will not run on hardware (chipset) configurations we have not explicitly tested. In the event that a no .bin is generated, you should send the .log and .csv files to the authors of this document.

## Analysis

We have also included some post-processing scripts used to analyze and aggregate the data received from system deployments.

### Protections

In order to evaluate the vulnerability of a system, we have created a script to determine the vulnerability of the system to BIOS and SMRAM writes. When a system has a writable BIOS or SMRAM, we described this as “unlocked”, as seen in the script’s output. The script is given a directory containing CSV files from Copernicus, attempts to parse them all, and outputs its result. The script will only try to process files with the .csv extension and will ignore files which do not contain Copernicus data. The output is in the form of a table printed to the command line, as well as optionally written to a CSV file. It can display the results for each individual CSV file, or it can display results aggregated by version (consisting of manufacturer, model, and BIOS revision). There are also several command line options for displaying more fine-grain results in order to determine why a BIOS or SMRAM is writable. An example execution of this script would be as follows:

python protections.py --per-file -s -p -i -o output.csv C:\copernicus\_data

To see the meaning of the individual command line options, you can execute:

python protections.py -h

The only primary dependency of the python script is the python module "docopt". You can install the newest version by following the instructions on the following page:

https://github.com/docopt/docopt

This page intentionally left blank.

###### CSV Output Field Descriptions

The following are descriptions of fields inspected by Copernicus.

General Information

For stability reasons, we do not allow Copernicus to run on chipsets that we have not tested on. Therefore this information is used to identify whether a system has been tested on, as well as to give a human-readable sense of what family a chipset belongs to.

Table A-. General System Identification & Compatibility Information

|  |  |
| --- | --- |
| Field Name | Field Description |
| CHIPSET\_DEV\_ID | This is the PCI Device ID used to identify the chipset used on the system. |
| CHIPSET\_FAMILY | The chipset families supported by Copernicus are ICH7, ICH8, ICH9, ICH10, and PCH-based chipsets. It is determined from the CHIPSET\_DEV\_ID. It is included for ease of quickly determining which other Copernicus fields are relevant to a particular machine. |

BIOS Information

The BIOS\_CNTL bits provide the broadest and highest level of protection of the flash chip. When properly set, the entire flash chip is write-protected. The PR (protected region) registers can offer the same functionality, but also provide read-protection and finer granularity while still being able to protect the entire flash chip. The FLOCKDN must be set in order to properly enforce PR protections.

Table A-. BIOS Access Control Information

|  |  |
| --- | --- |
| Field Name | Field Description |
| FREGx | These registers are only applicable in descriptor mode. They contain the base and limit addresses for the 5 possible regions in the flash chip. A chip may not implement all regions, but the supported regions (0 through 4) are: Flash Descriptor, BIOS, Intel ME, Gigabit Ethernet, and Platform Data. We are primarily concerned with region 1, the BIOS. |
| FLOCKDN | This is a bit is the HSFS register. Once set, certain configuration registers become "locked down" (read only). This is primarily used for locking down security configurations. Once set, this bit cannot be cleared without a system reset. |
| FLASH\_DESCRIPTOR\_MODE | This parameter indicates whether the flash chip is running in descriptor mode. In descriptor mode, the flash chip is segmented into regions described by the FREG registers. In non-descriptor mode, the chip is considered to be one monolithic region. It is determined from a bit in the HSFS register. |
| BIOSWE\_LOCK | When this is True, it indicates that the BIOSWE bit is disabled and locked down by SMM mode. This means that the BLE bit is set and SMM is blocking any attempts to turn on BIOSWE. BIOSWE and BLE are found in the BIOS\_CNTL register, which is described below. |
| FLASH\_DESCRIPTOR\_OVERRIDE | This is true whenever the flash descriptor override is enabled. While this is enabled, the FLMSTR1 protections are overridden with the FRAP protections. Note that this override is a physical pin which overrides the FLMSTR1 protections and it cannot be set by software. |
| PRx | These registers allow for hardware-enforced read or write protection of arbitrary regions of the flash chip. On ICH7-based chipsets, PR3 and PR4 are not applicable. These registers are read-only when FLOCKDN is set. |
| PREOP, OPMENU, OPTYPE, UVSCC\_ERASE, LVSCC\_ERASE | Collectively, these registers specify the operations which may be performed on the flash chip using the SPI software sequencing functionality. These operations are various types of read and write operations. Flash chips in non-descriptor mode only support software sequencing, whereas flash chips in descriptor mode support both hardware sequencing and software sequencing functionality. These registers are only interesting to us on (usually older) chips that are running in non-descriptor mode. This is because the chip cannot be written if these registers do declare any write operations, therefore the register CAN provide write-protection functionality. A chip operating in descriptor mode, however, will always have write operations available to use (barring any other write-protection mechanisms). PREOP, OPMENU, and OPTYPE become read-only when FLOCKDN is set. |
| FRAP | This register is only applicable in descriptor mode. When FLASH\_DESCRIPTOR\_OVERRIDE is enabled, the permissions in FLMSTR1 are overridden by the permissions in this register. |
| FLMSTRx | These registers are only applicable in descriptor mode. There are three of these registers, one each for flash regions 1, 2, and 3. The bits in these registers control the ability of code in these regions to read or write to any of the other regions. Each of these registers provide enough bits to individually control readability and writability for each of the 5 regions. FLMSTR1 specifies permissions for the Host CPU as well as the BIOS. |
| BIOS\_CNTL | This contains the BIOSWE and BLE bits which are used to determine and protect writability of the entire flash chip. BIOSWE enables writability to the flash chip. When BLE is set, then SMM receives an interrupt whenever there is an attempt to enable BIOSWE, and SMM then has the ability to block the change (this is up to the vendor which provides the SMM code). |

SMM Information

There are three potential physical memory regions that can to be used as SMM RAM (SMRAM). These are the compatibility segment (CSEG), high memory segment (HSEG), and top of memory segment (TSEG). The enabled regions are determined by reading the SMRAMC and ESMRAMC registers. SMM runs at a higher privilege level than even ring 0 and so these regions should be protected from reads and writes when the CPU is not running in SMM mode. These protections are provided by setting the D\_LCK bit and using the SMRR registers. Due to caching features on modern CPUs, the SMRAM may be vulnerable even if D\_LCK protections are enabled (through cache poisoning). Therefore the SMRR registers should be used, if they are available, to lock down the entire SMRAM region. The implementation of the SMRR registers in the CPU, as opposed to the chipset, allows caching vulnerabilities to be mitigated.

Table A-. SMRAM Access Control Information

|  |  |
| --- | --- |
| Field Name | Field Description |
| SMRAMC | This register allows SMRAM to be enabled. It also features the D\_LCK bit. When set, reads and writes to the SMRAM regions are blocked by the chipset. In addition, most of SMRAMC and ESMRAMC become read-only when D\_LCK is set, and D\_LCK can only be cleared again by a system reset. |
| ESMRAMC | This register allows HSEG, TSEG, or both to be enabled. When HSEG is enabled, CSEG is effectively disabled by the chipset because HSEG memory addresses map to the same DRAM region that CSEG would have mapped to. This register also features the bits which can be set to enable caching of HSEG and TSEG. Cacheability of HSEG is the advantage of using it rather than CSEG. |
| SMRR\_PHYSMASK, SMRR\_PHYSBASE | These registers combined specify a single contiguous region of physical memory which is only readable and writable in SMM mode. They also specify cacheability options which override any other system caching settings. These registers themselves are only writable in SMM mode. |

###### Abbreviations

List of Abbreviations

|  |  |
| --- | --- |
| BIOS | Basic Input/Output System |
| SMM  SMRAM | System Management Mode  System Management RAM |
|  |  |