CHIPSEC ON NON-UEFI PLATFORMS

Erik Bjorge, Maggie Jauregui & Brian Richardson
Platform Armoring & Resiliency team

OSFC - 2018

CHIPSEC History

- CHIPSEC is a framework for analyzing the security of PC platforms including hardware, system firmware (BIOS/UEFI), and platform components.
- Originally developed by Yuriy Bulygin (@c7zero)
- First version of CHIPSEC was released in March 2014 at CanSecWest
- Currently used by firmware developers, system validation and system integrators

https://github.com/chipsec/chipsec.git

Current CHIPSEC Assumptions

- Runs on an Intel based platform
- Firmware has a threat model that is compatible with the Unified Extensible Firmware Interface (UEFI)
- All platforms require the same level of security

Threat Model Assumptions

- Configuration Settings
  - Registers that can be locked should be programmed and locked
  - Non-Volatile data that is updatable from OS should be minimized
- Flash Access
  - Serial Peripheral Interface (SPI) flash is runtime updateable
  - System Management Mode (SMM) or Protected Range (PRx) register are used to protect SPI flash
  - Flash programming matches guidance from Intel
- Others…

So What’s the Problem?

- Different methods to secure the platform exist
  - Read Only (RO) backup firmware with forced recovery
  - Physical presence for update
  - RO firmware
- Platforms have different security requirements
  - Open Development System
  - High Assurance Critical System
- CHIPSEC modules do not comprehend different platforms requirements

Processing Results

- Know your platform
  - Understand the security assumptions of your platform
- Know your security requirements
  - Physical attack in scope
  - Develop and deploy custom firmware
- Understand why modules may be skipped

CHIPSEC Example

- CHIPSEC run on Chromebook in developer mode
  - Legacy Boot to Linux
  - Skylake Y processor
- Results Summary
  - Failure in bios_wp
  - Warnings in expected locations
  - UEFI tests skipped as expected
  - All others passed

Thanks to John Loucaides from Eclypsium for the log file.

False Positive Example in bios_wp

```
[x][ =======================================================================
[x][ Module: BIOS Region Write Protection
[x][ =======================================================================
[*] BC = 0x0000008D << BIOS Control (b:d.f 00:31.5 + 0xDC)
    [00] BIOSWE           = 1 << BIOS Write Enable
    [01] BLE              = 0 << BIOS Lock Enable
    [02] SRC              = 3 << SPI Read Configuration
    [04] TSS              = 0 << Top Swap Status
    [05] SMM_BWP          = 0 << SMM BIOS Write Protection
    [06] BBS              = 0 << Boot BIOS Strap
    [07] BILD             = 1 << BIOS Interface Lock Down
[-] BIOS region write protection is disabled!
```

- Failure due to different security model being used
  - SMM based protections disabled
  - Configuration locked (good)
- User needs to understand that this is not a real failure

What Can the Community Do?

- Discuss updates to CHIPSEC to support different threat models
  - Open an issue on GitHub for this
  - Looking for the community to provide guidance on implementation
- Create new modules or update existing modules to support multiple threat models
- Submit issues and pull requests on GitHub

https://github.com/chipsec/chipsec

Get Involved Today

Learn to write CHIPSEC modules and utility commands at my next talk.

- **Writing CHIPSEC Modules & Tools**

Participate:

https://github.com/chipsec/chipsec

Contact the Intel CHIPSEC Team:

chipsec@intel.com

Legal Notice

No computer system can be absolutely secure.

No license (express or implied, by estoppel or otherwise) to any intellectual property rights is granted by this document.

Intel disclaims all express and implied warranties, including without limitation, the implied warranties of merchantability, fitness for a particular purpose, and non-infringement, as well as any warranty arising from course of performance, course of dealing, or usage in trade.

This document contains information on products, services and/or processes in development. All information provided here is subject to change without notice.

The products and services described may contain defects or errors known as errata which may cause deviations from published specifications. Current characterized errata are available on request.

Intel, the Intel logo are trademarks of Intel Corporation or its subsidiaries in the U.S. and/or other countries.

*Other names and brands may be claimed as the property of others intel®

Software