IDF2012
INTEL DEVELOPER FORUM

Intel and McAfee: Hardening and Harnessing the Secure Platform

Vincent Zimmer, Principal Engineer, Intel
Roy Hopkins, Senior Software Engineer, McAfee Inc.

EFIS003

Sponsors of Tomorrow.

Please Fill Out The Online Session Evaluation Form

Enter to win fabulous prizes including Ultrabooks™, SSDs and more!

You will receive an email with a link to the online session evaluation prior to the end of this session. Please submit the evaluation by 10am tomorrow to be entered to win.

Winners will be announced by email

Sweepstakes rules are available at the Help Desk on Level 2
All sessions evaluations must be submitted by Friday Sept 14 at 5pm

IDF2012
INTEL DEVELOPER FORUM

Agenda

- UEFI & PI Security Overview
- Hardening the Platform & Development Assurance Practices
- Introducing McAfee* Endpoint Encryption
- Value Proposition of a Secured Preboot
- Maintaining the Chain of Trust

Agenda

- UEFI & PI Security Overview
- Hardening the Platform & Development Assurance Practices
- Introducing McAfee* Endpoint Encryption
- Value Proposition of a Secured Preboot
- Maintaining the Chain of Trust

Pressure on BIOS

Industry requirements (ex. UEFI 2.3.1+ Ch 27, TCG)

Government requirements (ex: US NIST SP800-147)

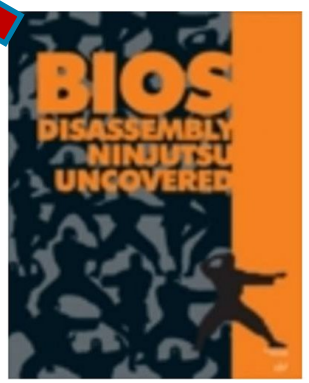
Product dvlp requirements (ex. SDL)

Customers requiring security (ex. US DoD, Corporate IT)

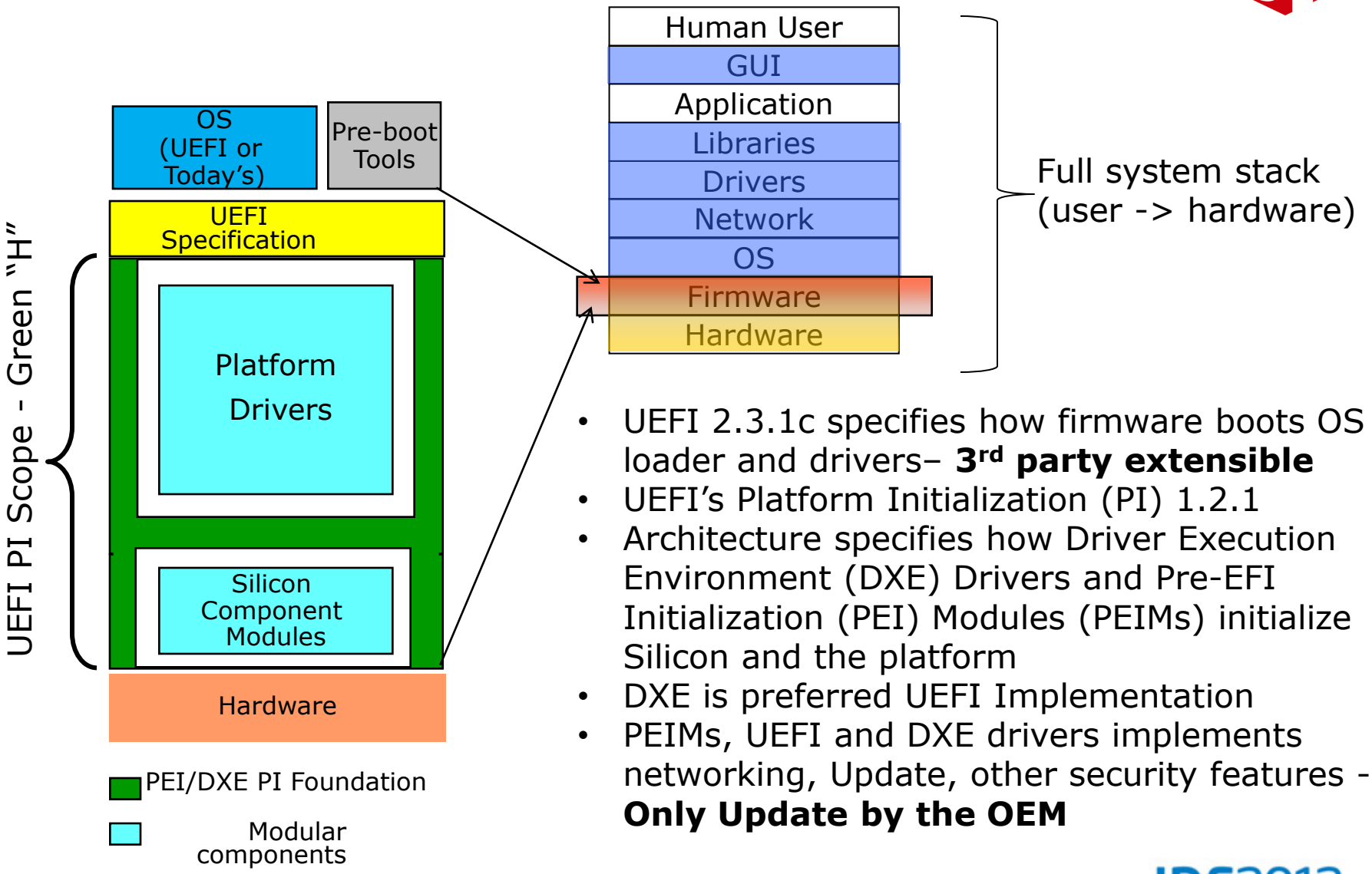
BIOS

Malware (ex. Chernobyl, 2000 Bootkits, 2011 etc)

Researchers (ex. Invisible Things Lab BMP attacks 2004)

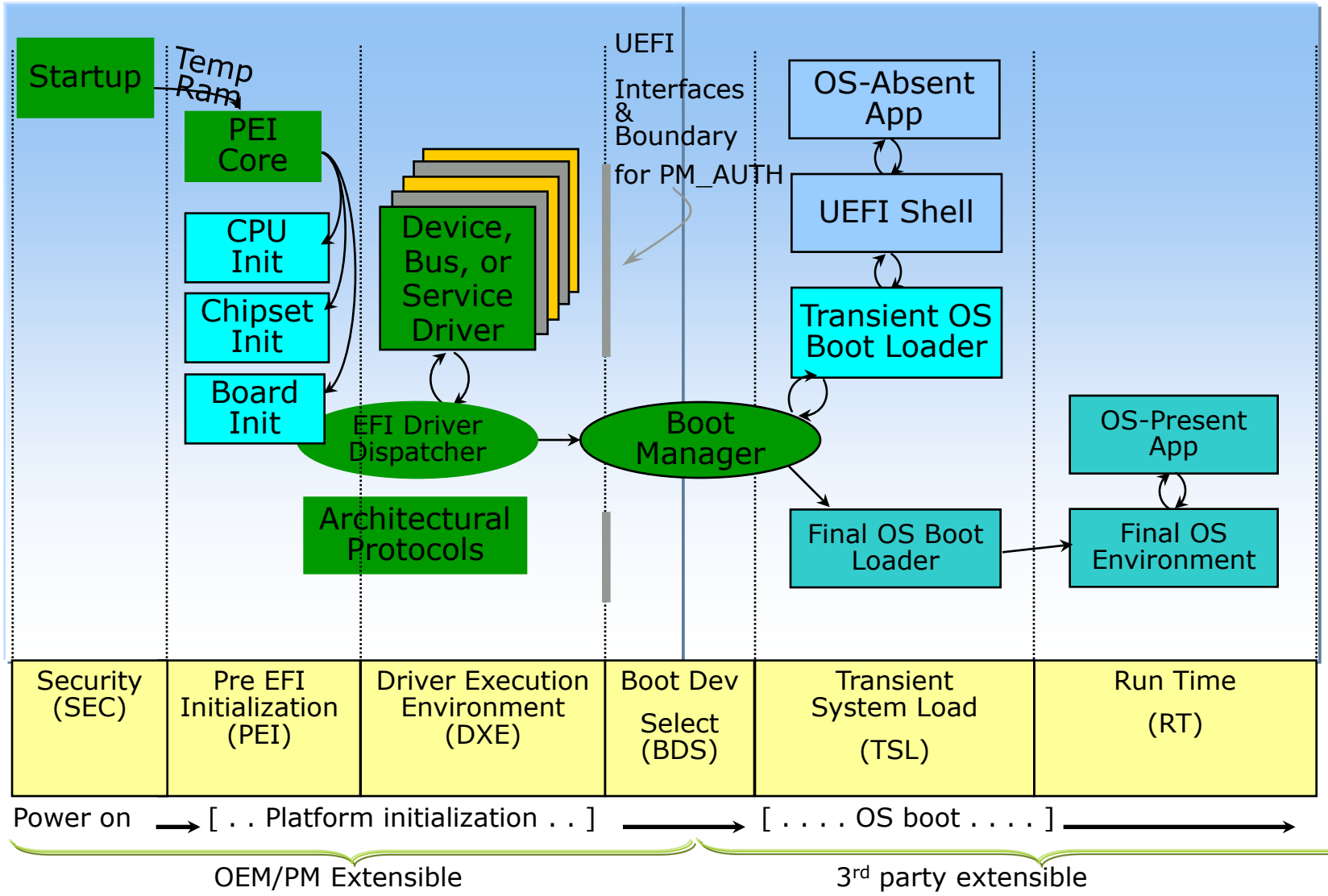


What is UEFI? UEFI Platform Initialization Overview



- UEFI 2.3.1c specifies how firmware boots OS loader and drivers– **3rd party extensible**
- UEFI's Platform Initialization (PI) 1.2.1
- Architecture specifies how Driver Execution Environment (DXE) Drivers and Pre-EFI Initialization (PEI) Modules (PEIMs) initialize Silicon and the platform
- DXE is preferred UEFI Implementation
- PEIMs, UEFI and DXE drivers implements networking, Update, other security features - **Only Update by the OEM**

Boot flow and Integrity



UEFI Protects through the Boot Flow

Agenda

- UEFI & PI Security Overview
- Hardening the Platform & Development Assurance Practices
- Introducing McAfee* Endpoint Encryption
- Value Proposition of a Secured Preboot
- Maintaining the Chain of Trust

Development Practices - Themes

- Practice defense in depth
  - Use several protection layers when designing and implementing security mechanisms
- Do not rely on security by obscurity
- Fail intelligently, Fail Safe
  - Fail secure – fail closed
    - Robust crisis recovery, signed updates/signed recovery FV, etc.
  - Don’t provide hints to hackers (e.g., by disclosing information on failure).
  - Log errors and failures for auditing
    - Trusted Computing Group (TCG) measured boot
- Check all return values
- Keep security critical code short and simple

Design in security from the start

Development Practices – Code Review

- Avoid unsafe calls (e.g., gets() equivalent)
- ASSERTs that should be error checking
- Check for valid input and reject everything else
- Perform sanity checks and bound checks – Check Type, Length, Range, Format
- Validate as much and as deep as possible to prevent unintended errors if code is changed; balance against coding time/performance
- Be careful of boundary conditions (e.g., off-by-one errors, array indices) and conditionals (e.g., reverse logic)
- Don't implement your own crypto algorithms or protocols Intel® UEFI Development Kit 2010 (Intel® UDK2010) uses OpenSSL* to meet the spirit of this

It's not implementing the feature, but also how you write the code

Defensive Coding – Adding Robustness

- Validate input before using
  - Network packet
  - On-disk data structures/GPT
  - UEFI Variables
  - Device paths
- Storing secrets
  - Avoid if possible
  - Clear buffers to zero when done
- Key management
  - Access control storage to PI elements. SMM based authenticated variable driver in Intel® UDK2010.
- Fuzz testing
  - SCTS – positive testing "Does it work with expected input"?
  - Fuzzing is negative testing "What happens with unexpected input?"

It's not just functional verification

Example of Safe Versus Unsafe Code

Example: Validate all input

```
PartEntry = AllocatePool (PrimaryHeader->NumberOfPartitionEntries
                          * sizeof (EFI_PARTITION_ENTRY));                    A
Status = DiskIo->ReadDisk (
    DiskIo,
    MediaId,
    MultU64x32(PrimaryHeader->PartitionEntryLBA, BlockSize),
    PrimaryHeader->NumberOfPartitionEntries * (PrimaryHeader->SizeOfPartitionEntry),   B
    PartEntry                                                                 C
    );
```

Problem:

- The memory is allocated with A
- However, ReadDisk block is with B
- Buffer overflow occurs when the code reads a GPT with C

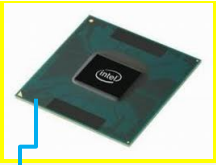
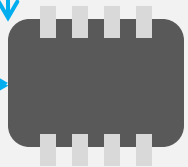
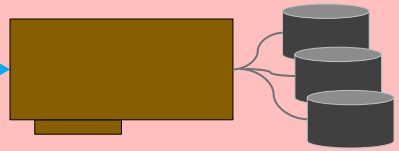
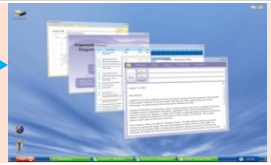
Fix:

```
PartEntry = AllocatePool (PrimaryHeader->NumberOfPartitionEntries
                          * PrimaryHeader->SizeOfPartitionEntry);
```

Rationale for Input Validation

UDK2010 example:
http://edk2.svn.sourceforge.net/svnroot/edk2/trunk/edk2/MdeModulePkg/Universal/Disk/PartitionDxe/Gpt.c

Technologies – Putting it Together

Reset	Assets	Threats
	**BIOS Flash** Hardware protection	ROM Swap Bit rot
	**System BIOS** -PEI recovery. -SMM,UEFI Core. -PK,KEK. CRTM	Erase flash part Overwrite flash part
	**Option ROMs** UEFI drivers	Erase op ROM Overwrite op ROM
	**Network Boot** IPv6 for the cloud	Network attacks
	**Pre-OS UEFI application** OS Boot loader, ***McAfee** ***Endpoint Encryption***	Spoof UEFI application

TCG Measurements into PCRs 0..7

Intel® Silicon

SP800-147 Capsules

UEFI 2.3.1c

Different colors for different vendors

Agenda

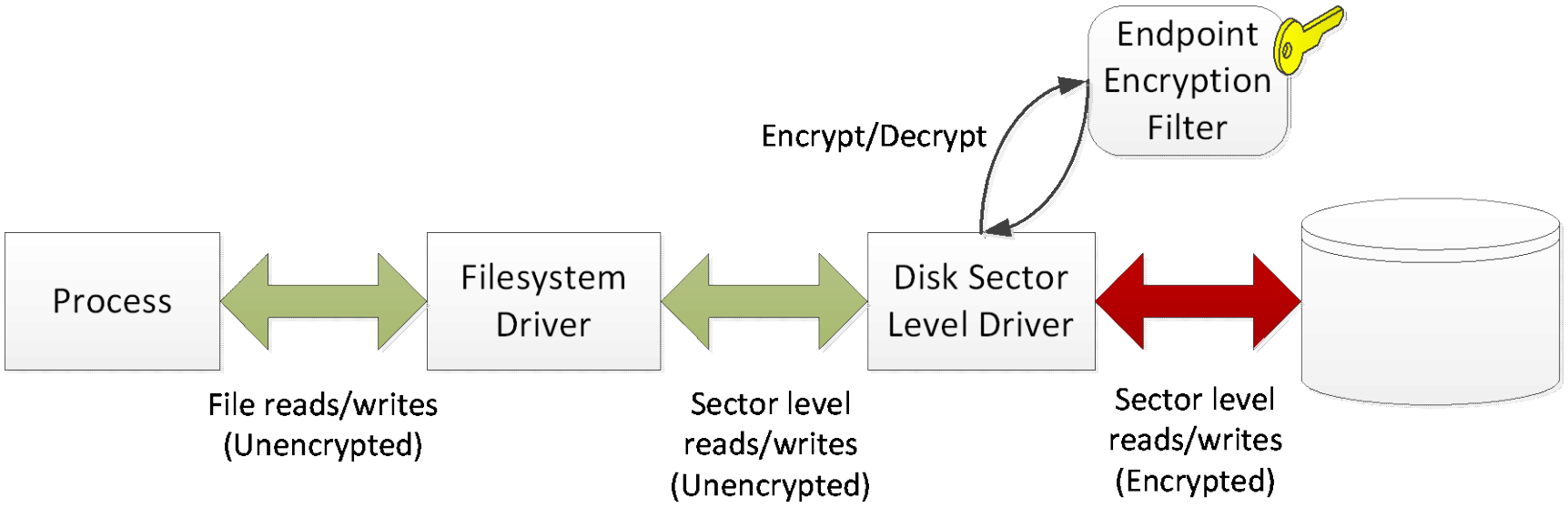
- UEFI & PI Security Overview
- Hardening the Platform & Development Assurance Practices
- Introducing McAfee* Endpoint Encryption
  - Product Overview
  - What is Full Disk Encryption?
  - UEFI Preboot Application
  - GPT Disks
  - Endpoint Encryption and the Boot process
- Value proposition of a Secured Preboot
- Maintaining the Chain of Trust

Product Overview

- McAfee endpoint encryption is a Full Disk Encryption product
  - Provides "data at rest" protection
  - Operating system data and user data is encrypted at the sector level
- Strong encryption algorithms protect data
  - Various methods of encrypting data are available
    - Software based AES256 CBC
    - Hardware accelerated AES256 CBC using AES-NI instructions
    - Self encrypting disks

What is Full Disk Encryption?

- Full Disk Encryption encrypts data at the sector level
  - The product has no knowledge of directories or files
  - The encryption is completely transparent to the file system
  - A disk can be partially encrypted and still operate normally; this allows the system to be encrypted online



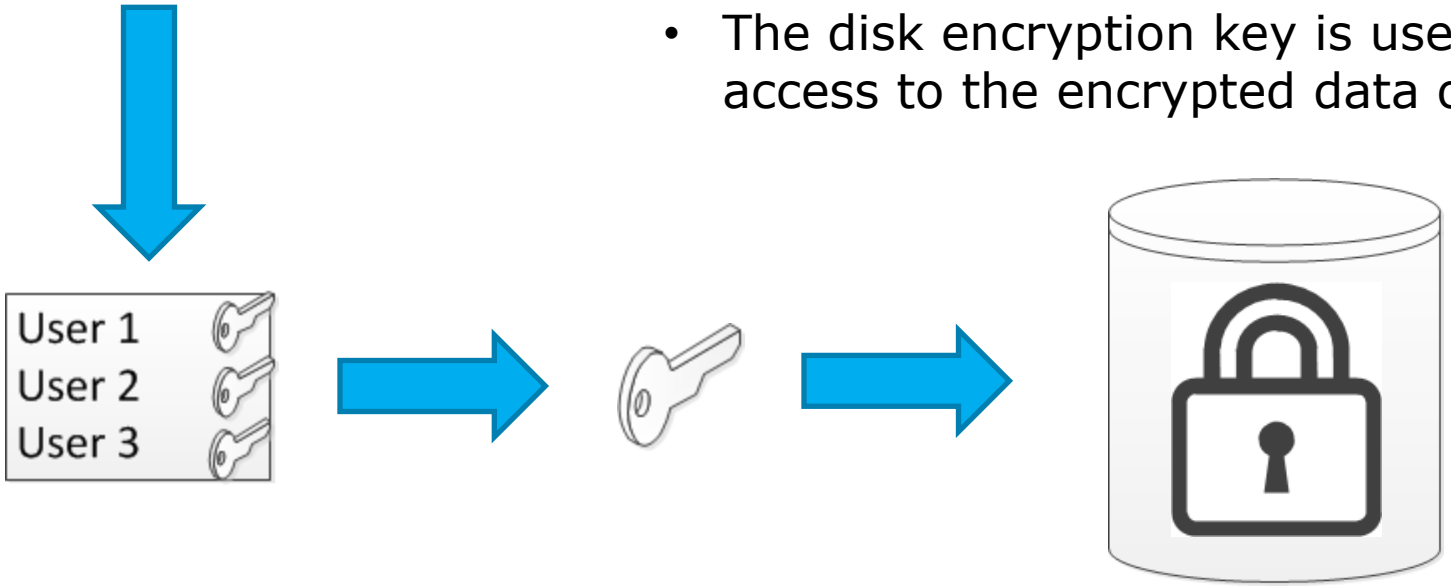
Endpoint Encryption Pre-Boot Application

- Encrypted disk data cannot be accessed until a user authenticates and the encryption key is obtained
- Operating system kernel and critical files lie within the encrypted data on disk
- A "Pre-Boot Application" (PBA) is required to authenticate and unlock the disk

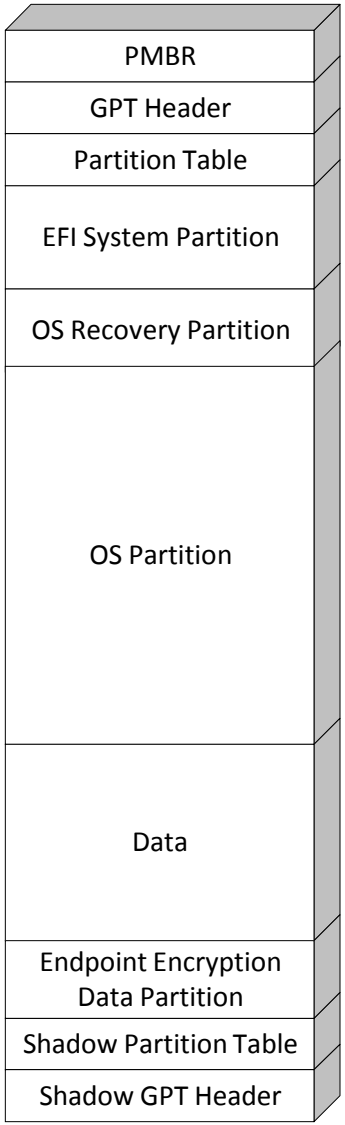
- The McAfee* Endpoint Encryption PBA is a UEFI application
  - Started by the UEFI Boot Manager before the Windows* bootloader
  - Uses standard UEFI protocols for GUI implementation (Graphics Output Protocol, Simple Pointer Protocol, etc.)
  - Supports USB smartcard readers and tokens using standard USB protocol

McAfee* PBA: Unlocking Your Data

- Disk is unlocked by authenticating using McAfee* Endpoint Encryption Pre-Boot Application (PBA)
- User authenticates using token; password, smartcard, recovery process, etc.
- Once authenticated, the token releases the disk encryption key
- The disk encryption key is used to gain access to the encrypted data on disk

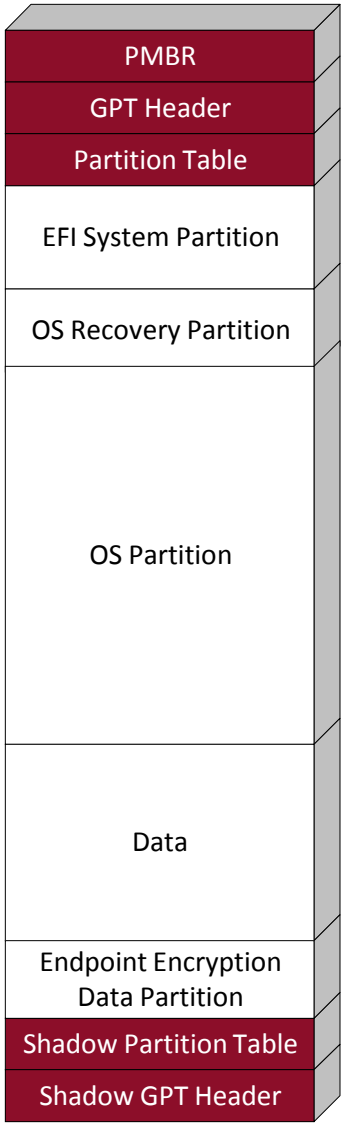


GPT Disks: What's Encrypted?



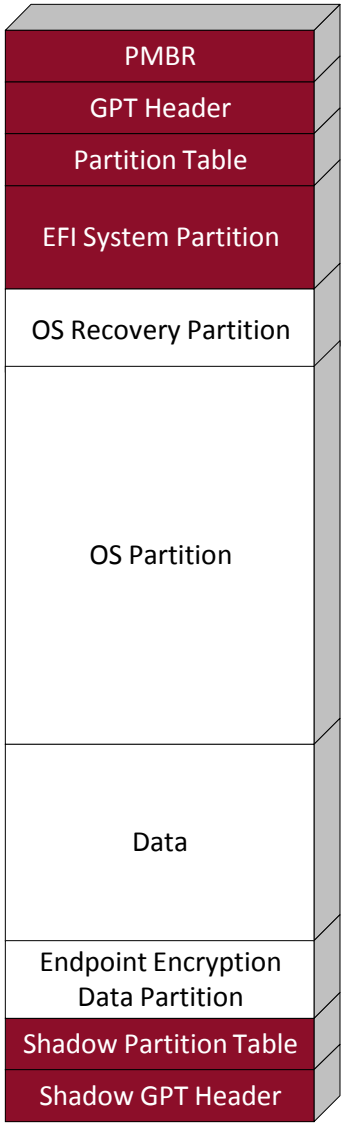
- Some parts of the disk need to remain unencrypted
  - Endpoint Encryption PBA is not implemented in firmware
  - PBA needs to be loaded from disk by UEFI boot manager
  - Disk must be recognisable by UEFI partition and file system drivers in order to load PBA

GPT Disks: What's Encrypted?



- Protective MBR, GPT Headers and Partition Tables cannot be encrypted
  - The data in these regions is required before the disk is unlocked
  - The disk would not be recognised as a valid GPT disk and the system would be unable to boot

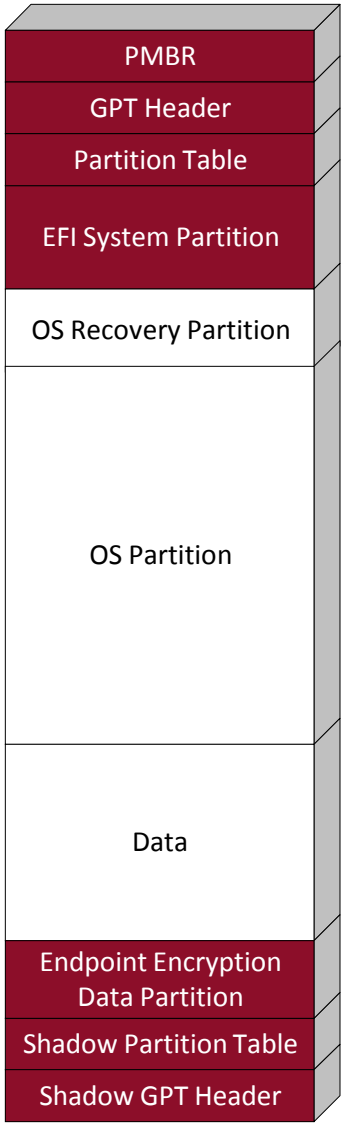
GPT Disks: What's Encrypted?



- Protective MBR, GPT Headers and Partition Tables cannot be encrypted
  - The data in these regions is required before the disk is unlocked
  - The disk would not be recognised as a valid GPT disk and the system would be unable to boot

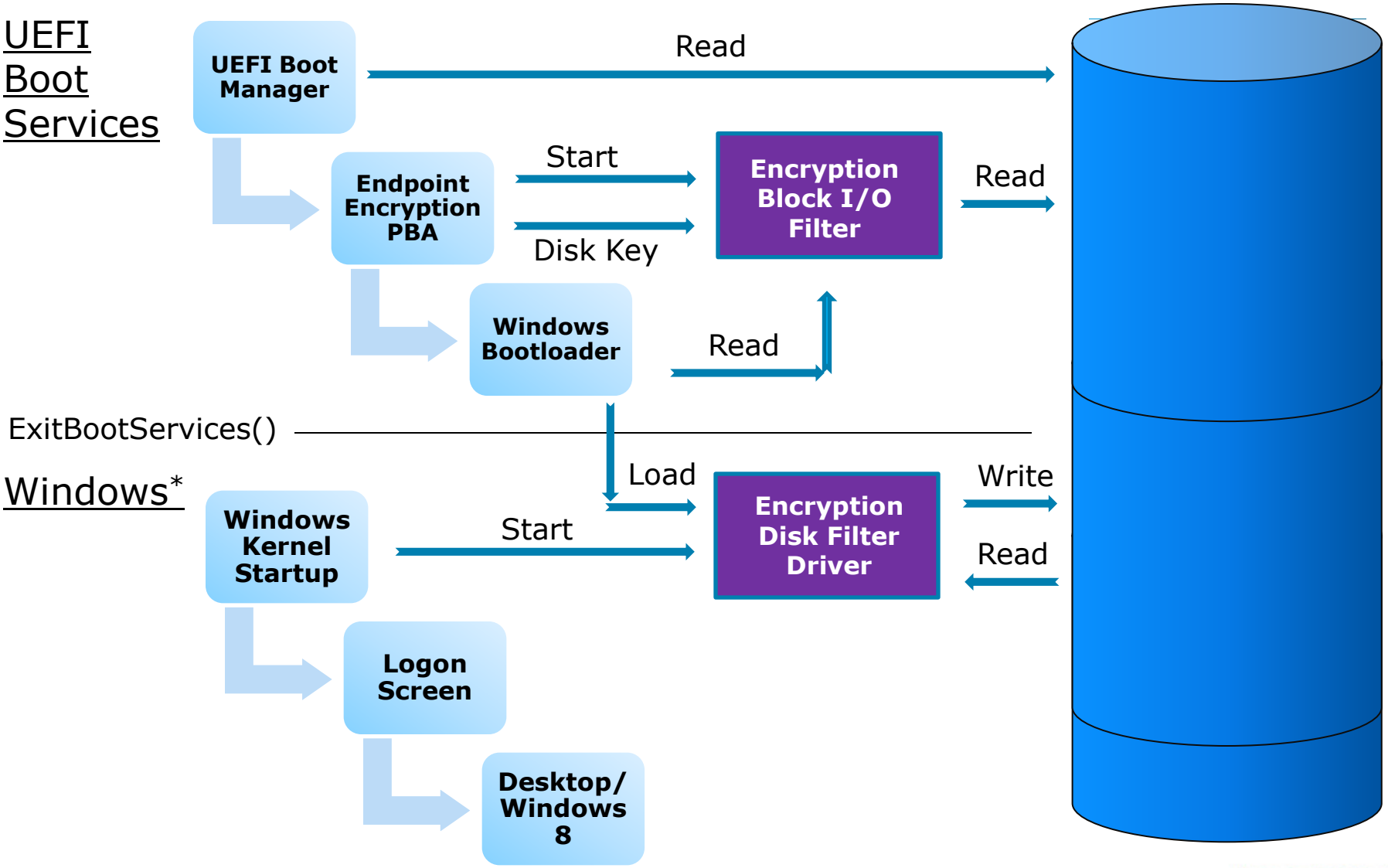
- EFI System Partition cannot be encrypted
  - Contains the executable McAfee* Endpoint Encryption preboot application image that is run by the UEFI Boot Manager
  - Also contains the Block I/O driver that performs the sector level encryption/decryption when authenticated

GPT Disks: What's Encrypted?



- Protective MBR, GPT Headers and Partition Tables cannot be encrypted
  - The data in these regions is required before the disk is unlocked
  - The disk would not be recognised as a valid GPT disk and the system would be unable to boot
- EFI System Partition cannot be encrypted
  - Contains the executable McAfee* Endpoint Encryption preboot application image that is run by the UEFI Boot Manager
  - Also contains the Block I/O driver that performs the sector level encryption/decryption when authenticated
- Endpoint Encryption Data Partition cannot be encrypted
  - Contains themes and localisation data for PBA
  - Contains database of users and token data
  - All data is required by the PBA prior to the disk being unlocked

The Boot Process



Agenda

- UEFI & PI Security Overview
- Hardening the Platform & Development Assurance Practices
- Introducing McAfee* Endpoint Encryption
- Value proposition of a Secured Preboot
  - What does "Secure Platform" Mean for Endpoint Encryption?
  - Malware Threat for Endpoint Encryption preboot
- Maintaining the Chain of Trust

What does "Secure Platform" Mean?

- There are some considerations for deploying UEFI applications and drivers on a secure platform
  - All UEFI applications and drivers must be signed
  - The image hashes or signing certificate must be trusted by the platform
  - UEFI applications and drivers need to be careful not to execute untrusted code
- Secure Boot provides benefits to Endpoint Encryption
  - Without Secure Boot, the PBA is vulnerable to malware attacks; keyloggers, denial of service
  - Tamper-resistant PBA provides platform for checking integrity of configuration files – signed policies

Maintain the Chain of Trust!

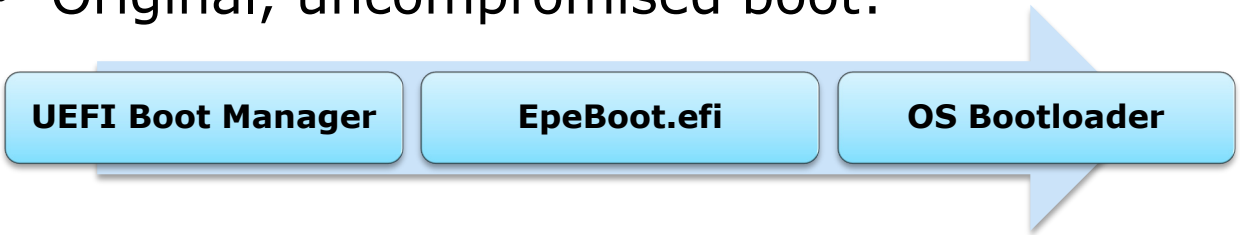
Malware Threat: Keylogger

```
A  BS->LocateHandleBuffer(ByProtocol, &simple_text_input_ex_protocol_guid, NULL, &num_handles,
                          &handles);
   for (i = 0; i < num_handles; ++i) {
B    BS->OpenProtocol(handles[i], &simple_text_input_ex_protocol_guid, &st, ImageHandle,
                      NULL, EFI_OPEN_PROTOCOL_GET_PROTOCOL);
C    hooked_protocols[i].st = st;
     hooked_protocols[i].orig_read_key_ex = st->ReadKeyStrokeEx;
     st->ReadKeyStrokeEx = keylogger_read_keystroke_ex;
   }
D    // Now chain load the original bootcode "EpeBoot.efi"
```

- All devices supporting EFI_SIMPLE_TEXT_INPUT_EX_PROTOCOL are enumerated representing keyboards and input devices at A
- A pointer to each protocol is obtained at B
- The function pointer that is used to obtain keystrokes is replaced with a function that logs the keystrokes and chains to the original at C
- The keylogger application loads and executes the original subverted UEFI application at D

Malware Threat: Keylogger Installation

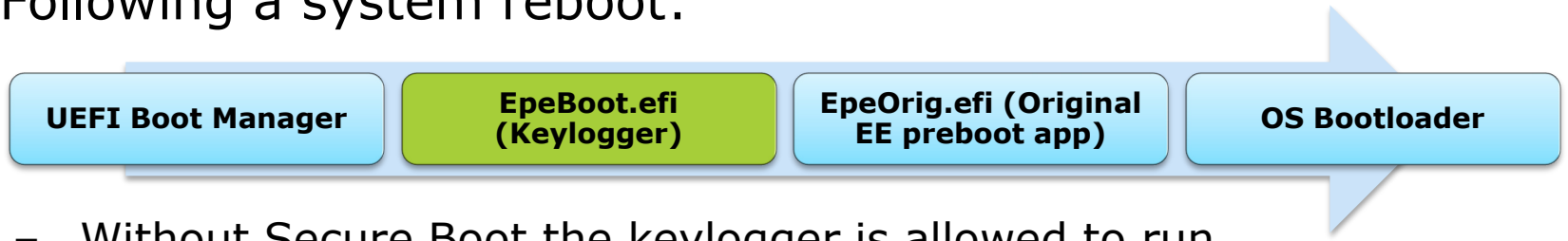
- Original, uncompromised boot:



- Without Secure Boot, installation of the keylogger is simple:

```
C:\> mountvol /s z:
C:\> copy z:\EFI\McAfee\EpeBoot.efi z:\EFI\McAfee\EpeOrig.efi
C:\> copy f:\keylogger.efi z:\EFI\McAfee\Epe\EpeBoot.efi
```

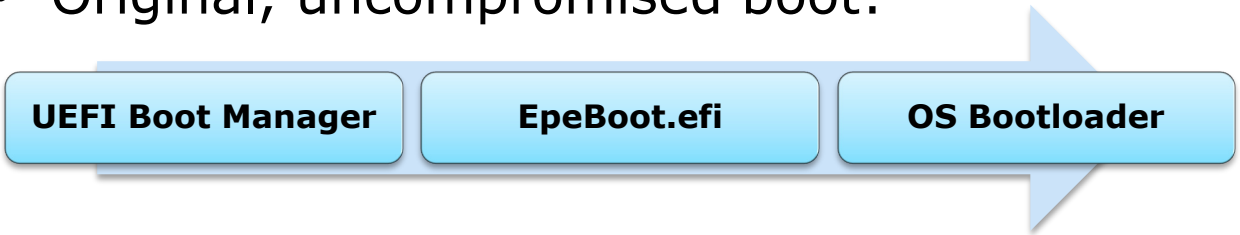
- Following a system reboot:



  - Without Secure Boot the keylogger is allowed to run
  - Endpoint Encryption PBA will execute but all keystrokes will be logged to disk

Malware Threat: Keylogger Installation

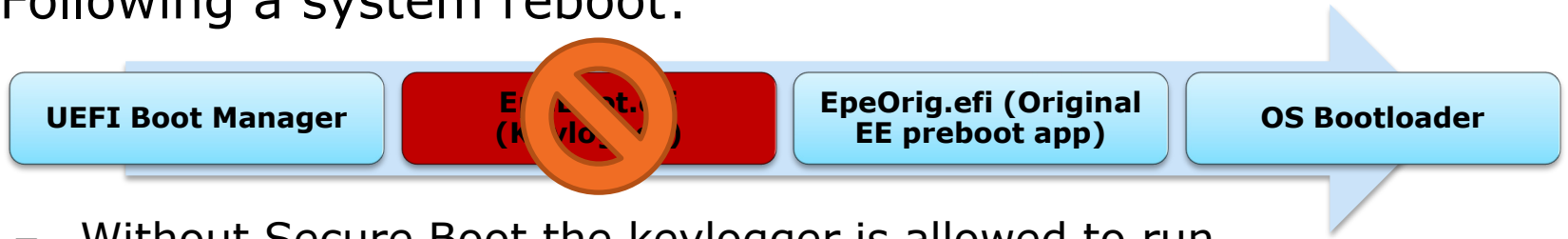
- Original, uncompromised boot:



- Without Secure Boot, installation of the keylogger is simple:

```
C:\> mountvol /s z:
C:\> copy z:\EFI\McAfee\EpeBoot.efi z:\EFI\McAfee\EpeOrig.efi
C:\> copy f:\keylogger.efi z:\EFI\McAfee\Epe\EpeBoot.efi
```

- Following a system reboot:



  - Without Secure Boot the keylogger is allowed to run
  - Endpoint Encryption PBA will execute but all keystrokes will be logged to disk

With Secure Boot, execution of the keylogger is prevented

Agenda

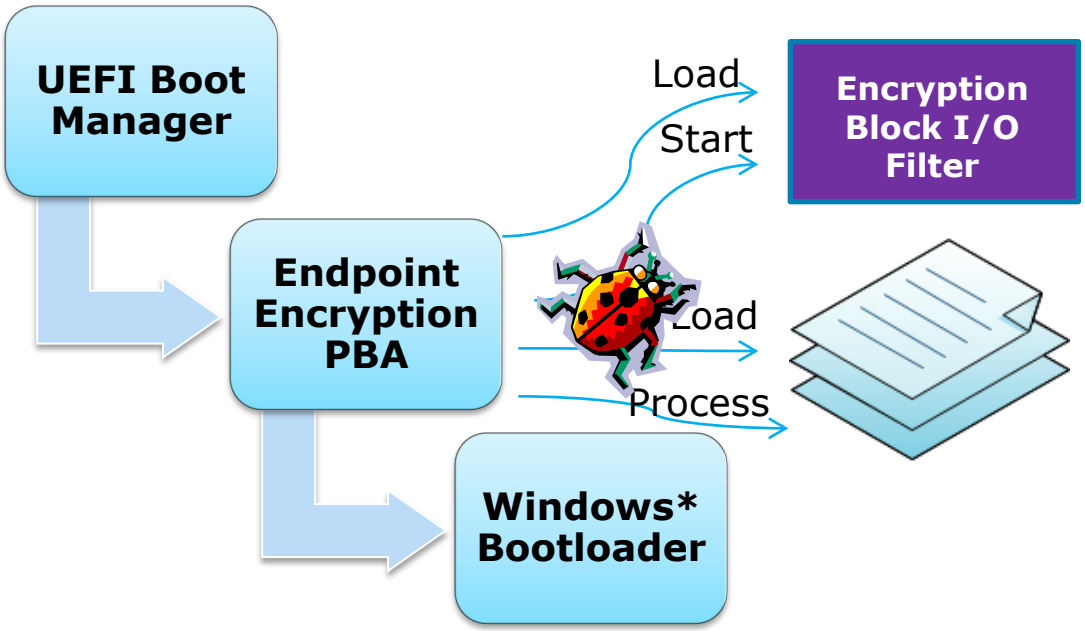
- UEFI & PI Security Overview
- Hardening the Platform & Development Assurance Practices
- Introducing McAfee* Endpoint Encryption
- Value proposition of a Secured Preboot
- Maintaining the Chain of Trust
  - Chain of Trust Considerations
  - What can go wrong?
  - Handling Loadable Modules/Data Files
  - Example Data File Breach

Chain of Trust: Considerations

- Why is the chain of trust important to Endpoint Encryption?
  - The chain of trust prevents malware from performing malicious actions such as keylogging or preventing the system from booting
  - Hardened boot process enables Endpoint Encryption PBA to validate configuration files – *Trusted Data*
    - Policy files and other important configuration files can be signed using a certificate
    - Certificate can be embedded in trusted UEFI application
- What needs to be considered?
  - Care must be taken to ensure the Chain of Trust can not be broken by unauthorised loadable modules or invalid data

What Can go Wrong?

- Even with Secure Boot the chain of trust can be broken if care is not taken



- Secure Boot ensures the Endpoint Encryption PBA and Windows* Bootloader are authentic
- PBA loads and executes Block I/O filter driver
- PBA loads and processes configuration and data files
- Careless coding may provide an exploitable bug to malware

Chain of Trust: Loadable Modules

- The Endpoint Encryption UEFI application allows for plugin modules
  - Used for adding support for USB smartcard readers
- **This poses a risk to the chain of trust**
  - It is the responsibility of the Endpoint Encryption UEFI application to ensure untrusted code cannot be executed
- The problem is easily solved:
  - Loadable modules are built as UEFI drivers
  - The modules are loaded using the Boot Services "LoadImage()" function
  - If the loadable module is not trusted by the platform, "LoadImage()" returns EFI_SECURITY_VIOLATION
  - **The chain of trust is maintained!**

Chain of Trust: Data Files

- Why are data files a threat to the Chain of Trust?
  - The McAfee* Endpoint Encryption PBA uses many configuration files
  - Malware may maliciously modify configuration files to attempt to crash the PBA
  - Modified configuration files can be engineered to execute malicious code
    - Common exploits overflow stack variables to modify function return address to jump to unauthorised code
    - ***The chain of trust is broken!***
- How can this be prevented?
  - ***All*** buffers that are populated from disk are carefully checked to prevent overflow
  - Data file signing can be used to verify authenticity of files

Data File Threat

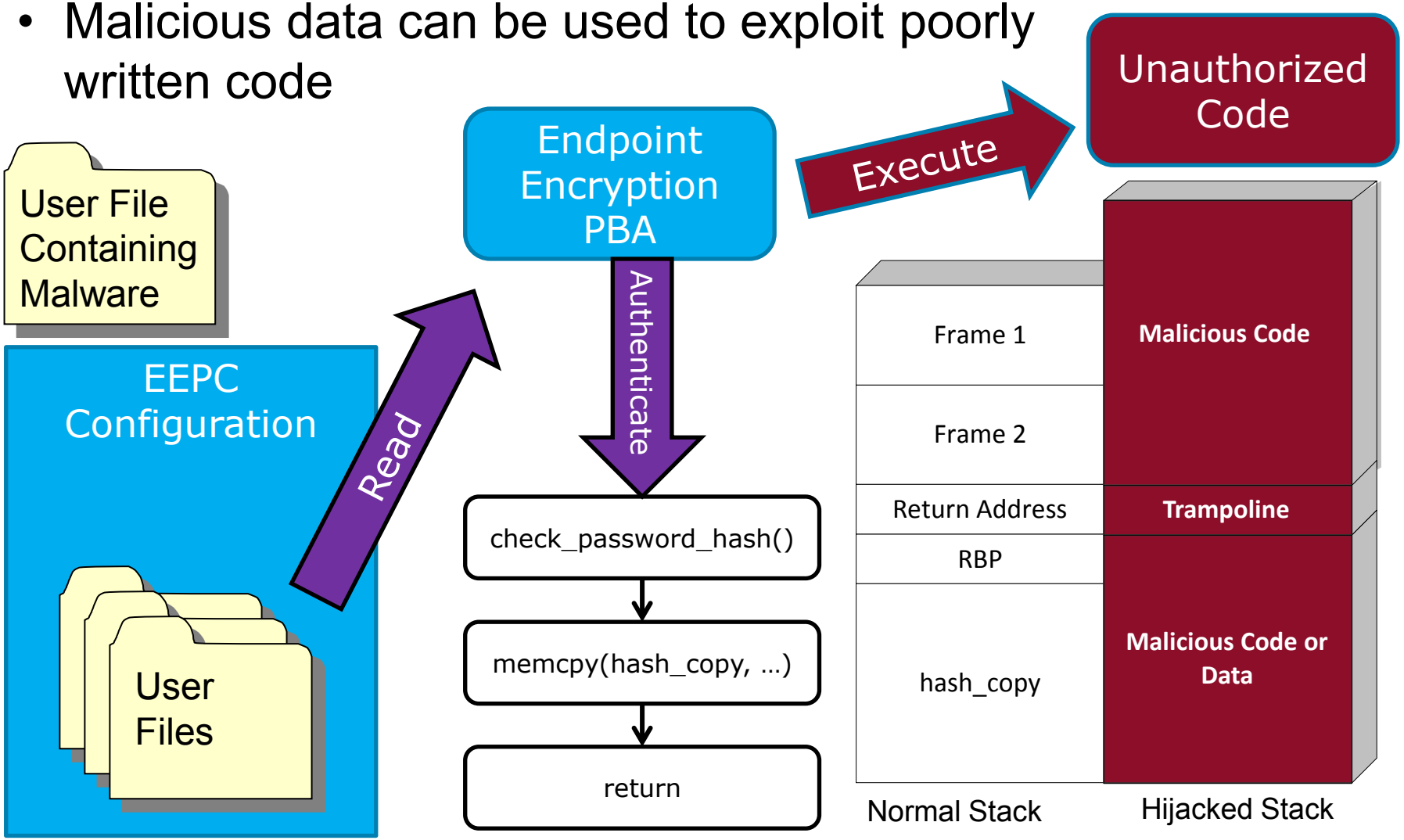
```
A struct USER_DATA {
    char    username[MAX_USERNAME_LENGTH + 1];
    long    hash_length;
    char    password_hash[MAX_PASSWORD_HASH_LENGTH];
  }
  int check_password_hash(USER_DATA* user_data, char* hash) {
B   char hash_copy[MAX_PASSWORD_HASH];
    // Take a copy of the hash so we can modify the buffer
    // !! No check to ensure the hash length is valid !!
C   memcpy(hash_copy, user_data->password_hash, user_data->hash_length);
    // Perform some calculation on the copied buffer

D   return success;
  }
```

- Structure that mimics user file on disk is defined at A
- Fixed length buffer assigned on stack at B
- Memory copied from disk buffer to stack without validating input at C. Stack has been compromised
- Return address D from function jumps to malicious code

Example: Malicious Data

- Malicious data can be used to exploit poorly written code



Validate all configuration and input!

Summary

- Platform security is maintained by a combination of hardware and software using many technologies and specifications
- UEFI Secure Boot is a vital part of the chain that keeps the platform protected
- Malware infiltration during the boot process is prevented by the Chain of Trust
- McAfee* Endpoint Encryption adds data security to the hardened security provided by the Secure Boot process
- Precautions need to be taken when writing software to prevent the Chain of Trust from being breached

Get More Information

- Intel UEFI Community - http://intel.com/udk
- UEFI Forum Learning Center
  - http://www.uefi.org/learning_center/
- Use the TianoCore edk2-devel mailing list for support from other UEFI developers
- Read the "A Tour Beyond BIOS into UEFI Secure Boot" whitepaper at tianocore.org

- Technical Showcase Booth #946

Other UEFI Sessions @ IDF

Session	Title	RM	Day	Date	Time
✓ EFIS001	Developing UEFI Support for Linux*	2008	Tue	11-Sep	10:30
✓ EFIS002	Using Wind River Simics* Virtual Platforms to Accelerate Firmware Development	2008	Tue	11-Sep	12:45
✓ EFIS003	Intel and McAfee: Hardening and Harnessing the Secure Platform	2008	Tue	11-Sep	3:30
EFIS004	Microsoft* Windows* 8 Firmware Developments and Intel® Platforms	2008	Wed	12-Sep	10:30
SECS004	Security Innovations in Intel® Platforms and Microsoft* Windows* 8	2008	Wed	12-Sep	2:00
EFIC001	Poster: Intel® UEFI Development Kit Debugger Tool	Poster	Thur	13-Sep	11:15
EFIC002	Poster: UEFI Driver Development Tools	Poster	Thur	13-Sep	11:15

✓ = DONE

Please Fill Out The Online Session Evaluation Form

Enter to win fabulous prizes including Ultrabooks™, SSDs and more!

You will receive an email with a link to the online session evaluation prior to the end of this session. Please submit the evaluation by 10am tomorrow to be entered to win.

Winners will be announced by email

Sweepstakes rules are available at the Help Desk on Level 2
All sessions evaluations must be submitted by Friday Sept 14 at 5pm

IDF2012
INTEL DEVELOPER FORUM

Q&A

Legal Disclaimer

INFORMATION IN THIS DOCUMENT IS PROVIDED IN CONNECTION WITH INTEL PRODUCTS. NO LICENSE, EXPRESS OR IMPLIED, BY ESTOPPEL OR OTHERWISE, TO ANY INTELLECTUAL PROPERTY RIGHTS IS GRANTED BY THIS DOCUMENT. EXCEPT AS PROVIDED IN INTEL'S TERMS AND CONDITIONS OF SALE FOR SUCH PRODUCTS, INTEL ASSUMES NO LIABILITY WHATSOEVER AND INTEL DISCLAIMS ANY EXPRESS OR IMPLIED WARRANTY, RELATING TO SALE AND/OR USE OF INTEL PRODUCTS INCLUDING LIABILITY OR WARRANTIES RELATING TO FITNESS FOR A PARTICULAR PURPOSE, MERCHANTABILITY, OR INFRINGEMENT OF ANY PATENT, COPYRIGHT OR OTHER INTELLECTUAL PROPERTY RIGHT.

- A "Mission Critical Application" is any application in which failure of the Intel Product could result, directly or indirectly, in personal injury or death. SHOULD YOU PURCHASE OR USE INTEL'S PRODUCTS FOR ANY SUCH MISSION CRITICAL APPLICATION, YOU SHALL INDEMNIFY AND HOLD INTEL AND ITS SUBSIDIARIES, SUBCONTRACTORS AND AFFILIATES, AND THE DIRECTORS, OFFICERS, AND EMPLOYEES OF EACH, HARMLESS AGAINST ALL CLAIMS COSTS, DAMAGES, AND EXPENSES AND REASONABLE ATTORNEYS' FEES ARISING OUT OF, DIRECTLY OR INDIRECTLY, ANY CLAIM OF PRODUCT LIABILITY, PERSONAL INJURY, OR DEATH ARISING IN ANY WAY OUT OF SUCH MISSION CRITICAL APPLICATION, WHETHER OR NOT INTEL OR ITS SUBCONTRACTOR WAS NEGLIGENT IN THE DESIGN, MANUFACTURE, OR WARNING OF THE INTEL PRODUCT OR ANY OF ITS PARTS.
- Intel may make changes to specifications and product descriptions at any time, without notice. Designers must not rely on the absence or characteristics of any features or instructions marked "reserved" or "undefined". Intel reserves these for future definition and shall have no responsibility whatsoever for conflicts or incompatibilities arising from future changes to them. The information here is subject to change without notice. Do not finalize a design with this information.
- The products described in this document may contain design defects or errors known as errata which may cause the product to deviate from published specifications. Current characterized errata are available on request.
- Intel product plans in this presentation do not constitute Intel plan of record product roadmaps. Please contact your Intel representative to obtain Intel's current plan of record product roadmaps.
- Intel processor numbers are not a measure of performance. Processor numbers differentiate features within each processor family, not across different processor families. Go to: http://www.intel.com/products/processor_number.
- Contact your local Intel sales office or your distributor to obtain the latest specifications and before placing your product order.
- Copies of documents which have an order number and are referenced in this document, or other Intel literature, may be obtained by calling 1-800-548-4725, or go to: http://www.intel.com/design/literature.htm
- Intel, Ultrabook, Sponsors of Tomorrow and the Intel logo are trademarks of Intel Corporation in the United States and other countries.

- *Other names and brands may be claimed as the property of others.
-

Legal Disclaimer

Software Code Disclaimer

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice (including the next paragraph) shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

Risk Factors

The above statements and any others in this document that refer to plans and expectations for the second quarter, the year and the future are forward-looking statements that involve a number of risks and uncertainties. Words such as "anticipates," "expects," "intends," "plans," "believes," "seeks," "estimates," "may," "will," "should" and their variations identify forward-looking statements. Statements that refer to or are based on projections, uncertain events or assumptions also identify forward-looking statements. Many factors could affect Intel's actual results, and variances from Intel's current expectations regarding such factors could cause actual results to differ materially from those expressed in these forward-looking statements. Intel presently considers the following to be the important factors that could cause actual results to differ materially from the company's expectations. Demand could be different from Intel's expectations due to factors including changes in business and economic conditions, including supply constraints and other disruptions affecting customers; customer acceptance of Intel's and competitors' products; changes in customer order patterns including order cancellations; and changes in the level of inventory at customers. Uncertainty in global economic and financial conditions poses a risk that consumers and businesses may defer purchases in response to negative financial events, which could negatively affect product demand and other related matters. Intel operates in intensely competitive industries that are characterized by a high percentage of costs that are fixed or difficult to reduce in the short term and product demand that is highly variable and difficult to forecast. Revenue and the gross margin percentage are affected by the timing of Intel product introductions and the demand for and market acceptance of Intel's products; actions taken by Intel's competitors, including product offerings and introductions, marketing programs and pricing pressures and Intel's response to such actions; and Intel's ability to respond quickly to technological developments and to incorporate new features into its products. Intel is in the process of transitioning to its next generation of products on 22nm process technology, and there could be execution and timing issues associated with these changes, including products defects and errata and lower than anticipated manufacturing yields. The gross margin percentage could vary significantly from expectations based on capacity utilization; variations in inventory valuation, including variations related to the timing of qualifying products for sale; changes in revenue levels; segment product mix; the timing and execution of the manufacturing ramp and associated costs; start-up costs; excess or obsolete inventory; changes in unit costs; defects or disruptions in the supply of materials or resources; product manufacturing quality/yields; and impairments of long-lived assets, including manufacturing, assembly/test and intangible assets. The majority of Intel's non-marketable equity investment portfolio balance is concentrated in companies in the flash memory market segment, and declines in this market segment or changes in management's plans with respect to Intel's investments in this market segment could result in significant impairment charges, impacting restructuring charges as well as gains/losses on equity investments and interest and other. Intel's results could be affected by adverse economic, social, political and physical/infrastructure conditions in countries where Intel, its customers or its suppliers operate, including military conflict and other security risks, natural disasters, infrastructure disruptions, health concerns and fluctuations in currency exchange rates. Expenses, particularly certain marketing and compensation expenses, as well as restructuring and asset impairment charges, vary depending on the level of demand for Intel's products and the level of revenue and profits. Intel's results could be affected by the timing of closing of acquisitions and divestitures. Intel's results could be affected by adverse effects associated with product defects and errata (deviations from published specifications), and by litigation or regulatory matters involving intellectual property, stockholder, consumer, antitrust, disclosure and other issues, such as the litigation and regulatory matters described in Intel's SEC reports. An unfavorable ruling could include monetary damages or an injunction prohibiting Intel from manufacturing or selling one or more products, precluding particular business practices, impacting Intel's ability to design its products, or requiring other remedies such as compulsory licensing of intellectual property. A detailed discussion of these and other factors that could affect Intel's results is included in Intel's SEC filings, including the company's most recent Form 10-Q, Form 10-K and earnings release.

Rev. 5/4/12