IDF2013
INTEL DEVELOPER FORUM

Build Safety from Bare Metal
- Practices on Hardening and Harnessing the Secure Platform

Dong Wei, Fellow, HP
Qin Long, Software Architect, Intel
Jie Shen, Senior Security Consultant, McAfee Inc.

PTAS002

Sponsors of Tomorrow. intel

Agenda

- Overview of UEFI and its Security Handling
- Platform Hardening Practices
- McAfee* Endpoint Encryption and Secure Boot

The PDF for this Session presentation is available from our Technical Session Catalog at the end of the day at: intel.com / go / idfsessionsBJ

URL is on top of Session Agenda Pages in Pocket Guide

Overview of UEFI and Its Security Handling

Dong Wei
Fellow, Hewlett Packard
VP, UEFI Forum

Latest Updates from UEFI Forum

- Linux Foundation has signed the agreement to become a UEFI Forum Contributor
- UEFI 2.3.1d errata available soon
- UEFI 2.3.1c SCT Final Draft soon
- UEFI 2.4 reaches content complete
- PI 1.3 reaches content complete
- Future of UEFI with system configu[illegible]n and management considerations

Real World!

Researchers find attack on Millions of printers

Can a hacker take control of you printer? Using it to sniff informat from the network, steal confident information, or even attack other machines. Researchers have foun attack impacting millions of printe around the world.

Link Discovered Between TDSS Rootkit and DNSchanger Trojan

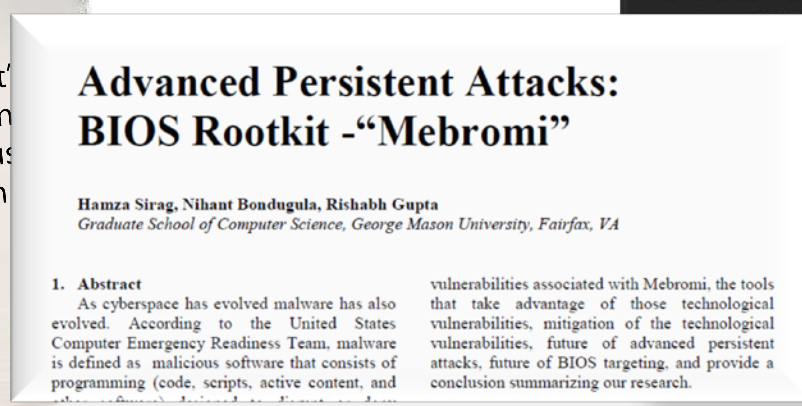
TDSS rootkit, the sophisticated and difficult to remove malware behind many advanced attacks also appears to have helped spread the DNSchanger Trojan.

Researcher finds attack on Apple battery firmware. [Blackhat 2011]

The firmware used to control the charging of Apple's laptop batteries could be attacked by malware. Allowing the attacker to potentially override safety mechanism which could lead to an attack.

Is Mebroot the stealthiest Rootkit in the world?

Federal agents raided unnamed operators of the Rustock "botnet" vast network of computers aroun the globe infected with maliciou software that allows distribution huge volumes of spam.



Advanced Persistent Attacks: BIOS Rootkit -"Mebromi"

Hamza Sirag, Nihant Bondugula, Rishabh Gupta
Graduate School of Computer Science, George Mason University, Fairfax, VA

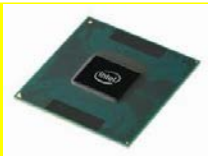
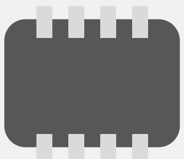
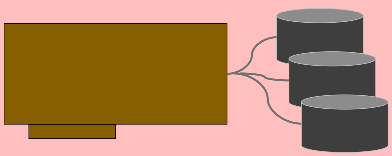
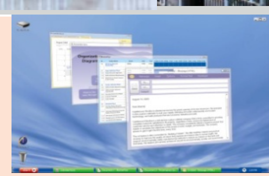
1. Abstract

As cyberspace has evolved malware has also evolved. According to the United States Computer Emergency Readiness Team, malware is defined as malicious software that consists of programming (code, scripts, active content, and

vulnerabilities associated with Mebromi, the tools that take advantage of those technological vulnerabilities, mitigation of the technological vulnerabilities, future of advanced persistent attacks, future of BIOS targeting, and provide a conclusion summarizing our research.

IDF2013 INTEL DEVELOPER FORUM

Assets & Threats

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	Spoof UEFI application

Different colors for different vendors

IDF2013
INTEL DEVELOPER FORUM

UEFI Security – Motivation & History

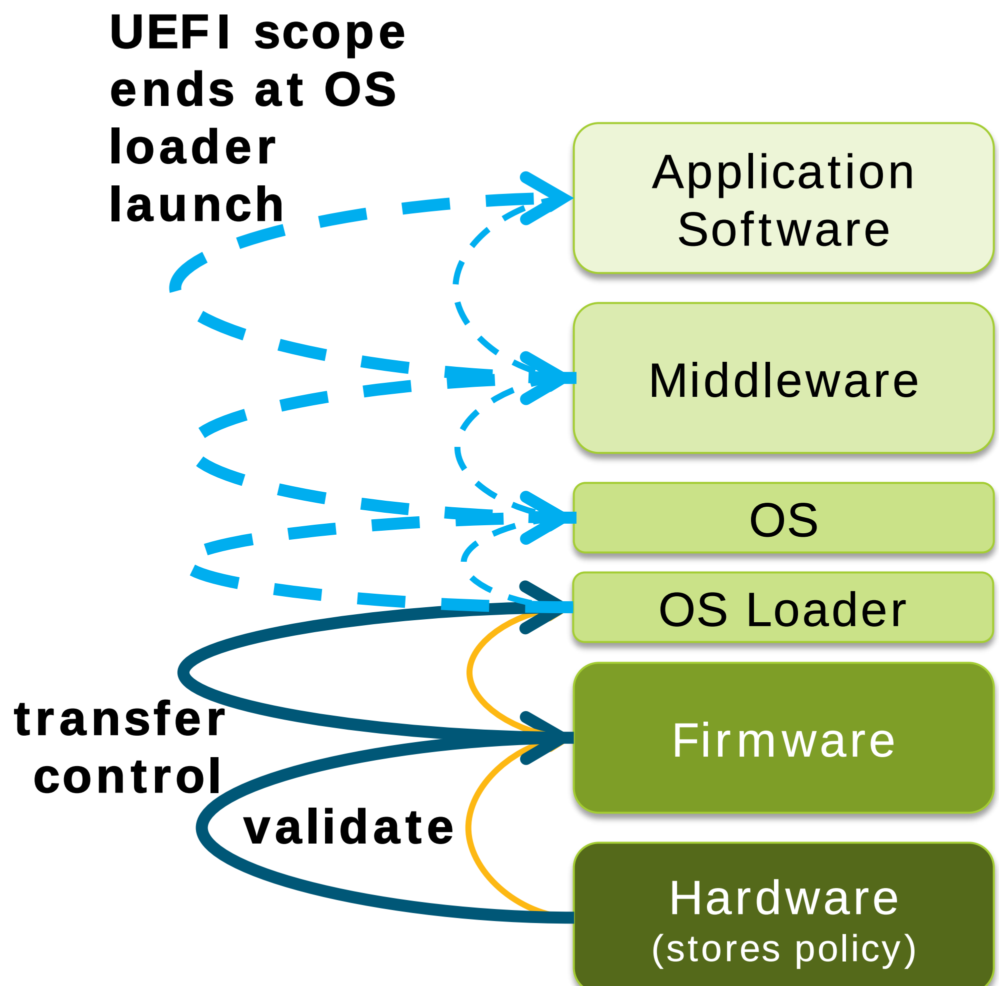
- As OS becomes more resistant to attack, the threat targets the weakest element in the chain
- History
  - Phoenix* initiated the discussion on the need for secure boot
  - USST (UEFI Security Sub-team) formed to address the topic
  - The secure boot architecture was defined in the UEFI 2.3 Specification
  - Microsoft* contributed additional capabilities for UEFI 2.3.1 Specification
    - Append support for the authenticated variables
    - Timestamp-based authenticated variable for roll-back protection
    - Authenticode specification for use in UEFI
    - UEFI Secure Boot support in Windows* 8

UEFI Security Enabling is an Industry Effort

IDF2013
INTEL DEVELOPER FORUM

UEFI Secure Boot: Enforcing Boot Policy

- The concept of UEFI secure boot is to have each component in the chain be ***validated and authorized*** against a given policy before allowing it to execute
- UEFI secure boot policy implementations can range from digital signatures to preloaded hash values…



IDF2013
INTEL DEVELOPER FORUM

Securing the Stack from Bare Metal

- UEFI 2.3.1 security enhancements specifically address the “secure boot” issue
- Securing the firmware itself further strengthens the UEFI Secure Boot concept
  - *How is the firmware update protected?*
  - *How is the firmware put into “admin mode”?*
- NIST has created *BIOS Protection Guidelines*
  - Secure Flash* update requirements
  - Maintain firmware core root of trust
- UEFI 2.3.1 contains the framework to develop secure Flash update

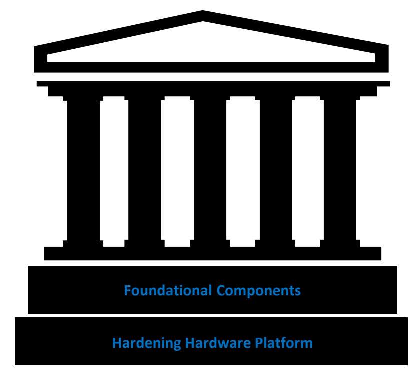
Platform Hardening Practices

Qin Long
Software Architect, Intel Corporation

IDF2013
INTEL DEVELOPER FORUM

Design in Security From the Start

- 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 and Fail Secure
  - Don't provide hints to hackers (e.g., by disclosing information on failure)
  - Log errors and failures for auditing
- Check all return values
- Keep security critical code short and simple



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

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 (Self-Certification Tests) – positive testing “Does it work with expected input”?
  - Fuzzing is negative testing “What happens with unexpected input?”

It’s not just functional verification

IDF2013 INTEL DEVELOPER FORUM

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:

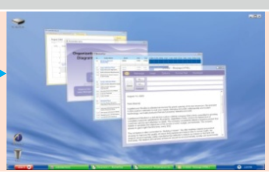
```
PartEntry = AllocatePool (PrimaryHeader->NumberOfPartitionEntries
                          * PrimaryHeader->SizeOfPartitionEntry);
```

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

TCG Measurements into PCRs 0..7

Reset	Assets	Threats	
	**BIOS Flash** Hardware protection	ROM Swap Bit rot	**Intel® Silicon**
	System BIOS -PEI recovery -SMM,UEFI Core -PK, KEK, CRTM	Erase flash part Overwrite flash part	**SP800 -147 Capsules**
	Option ROMs UEFI drivers	Erase op ROM Overwrite op ROM	**UEFI 2.3.1c**
	**Network Boot** IPv6 for the cloud	Network attacks	
	**Pre-OS UEFI application** OS Boot loader, ***McAfee* Endpoint Encryption***	Spoof UEFI application	

Different colors for different vendors

IDF2013
INTEL DEVELOPER FORUM

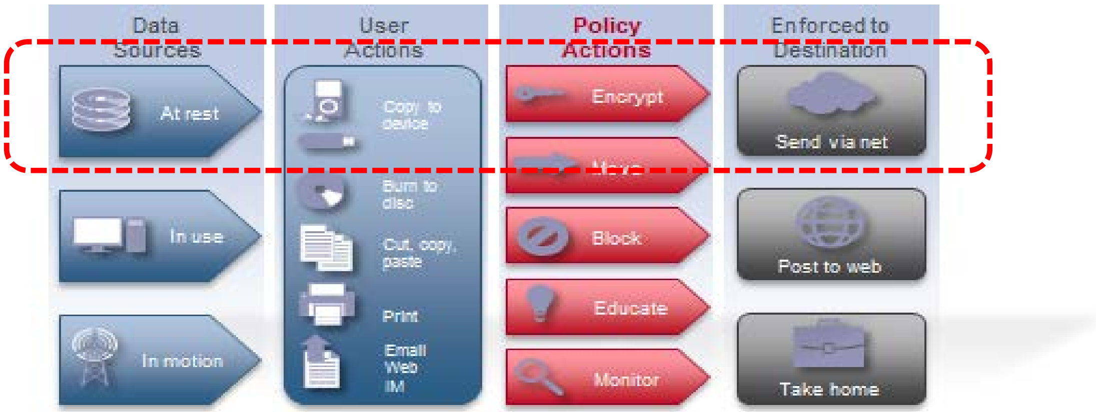
McAfee* Endpoint Encryption & Secure Boot

Jie Shen
Senior Security Consultant, McAfee Inc.

IDF2013
INTEL DEVELOPER FORUM

Product Overview

- McAfee* endpoint encryption is a Full Disk Encryption product
  - Provides "data at rest" protection



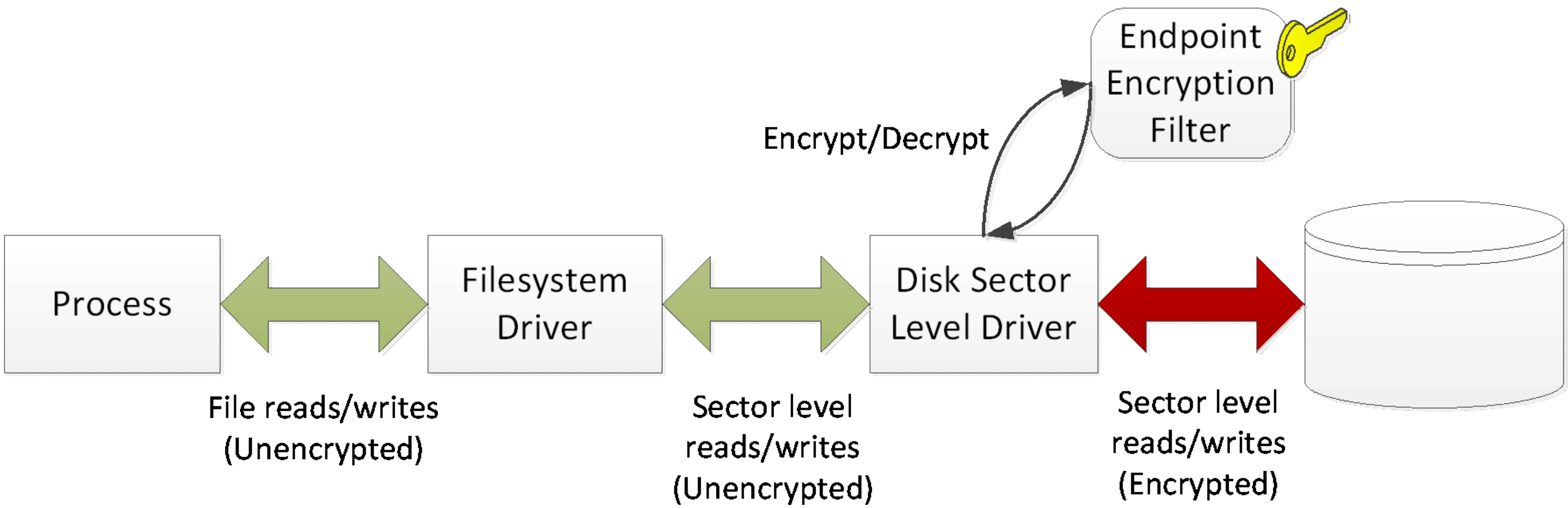
  - Operating system data and user data is encrypted at the sector level
- Strong encryption algorithms protect data
  - Various methods of encrypting data are available
    - Software based AES256 CBC (Cipher Block Chaining)
    - Hardware accelerated AES256 CBC using AES-NI instructions
    - Self encrypting disks

McAfee An Intel Company

IDF2013 INTEL DEVELOPER FORUM

What is 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



IDF2013 INTEL DEVELOPER FORUM

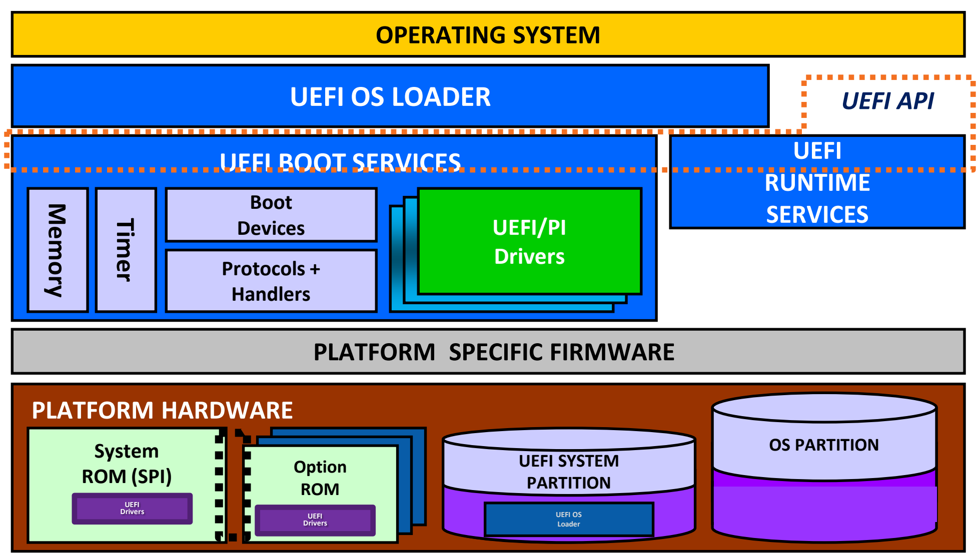
Encrypted Disk Unlock

- 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

- 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

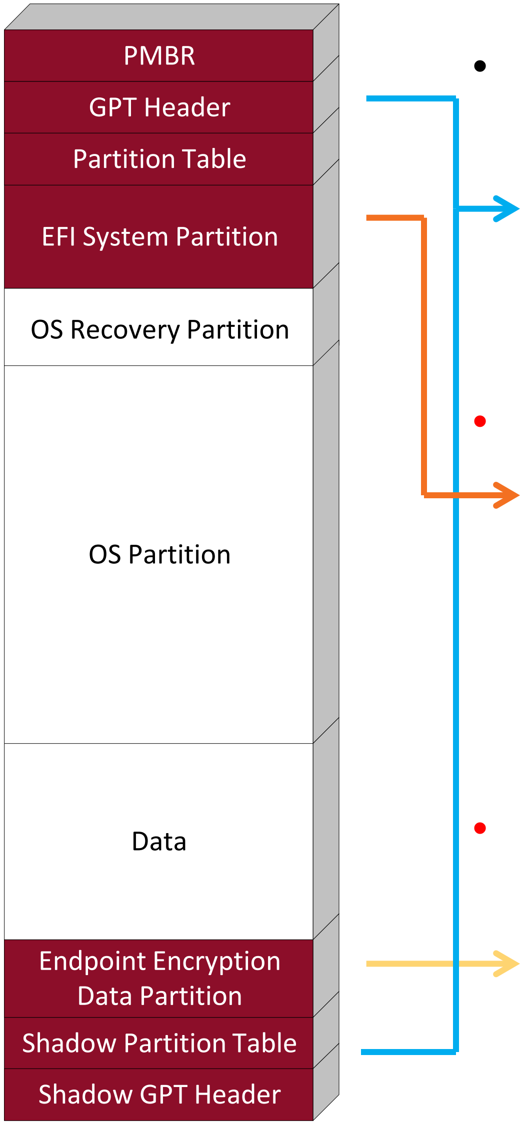
The McAfee* Endpoint Encryption PBA

- 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



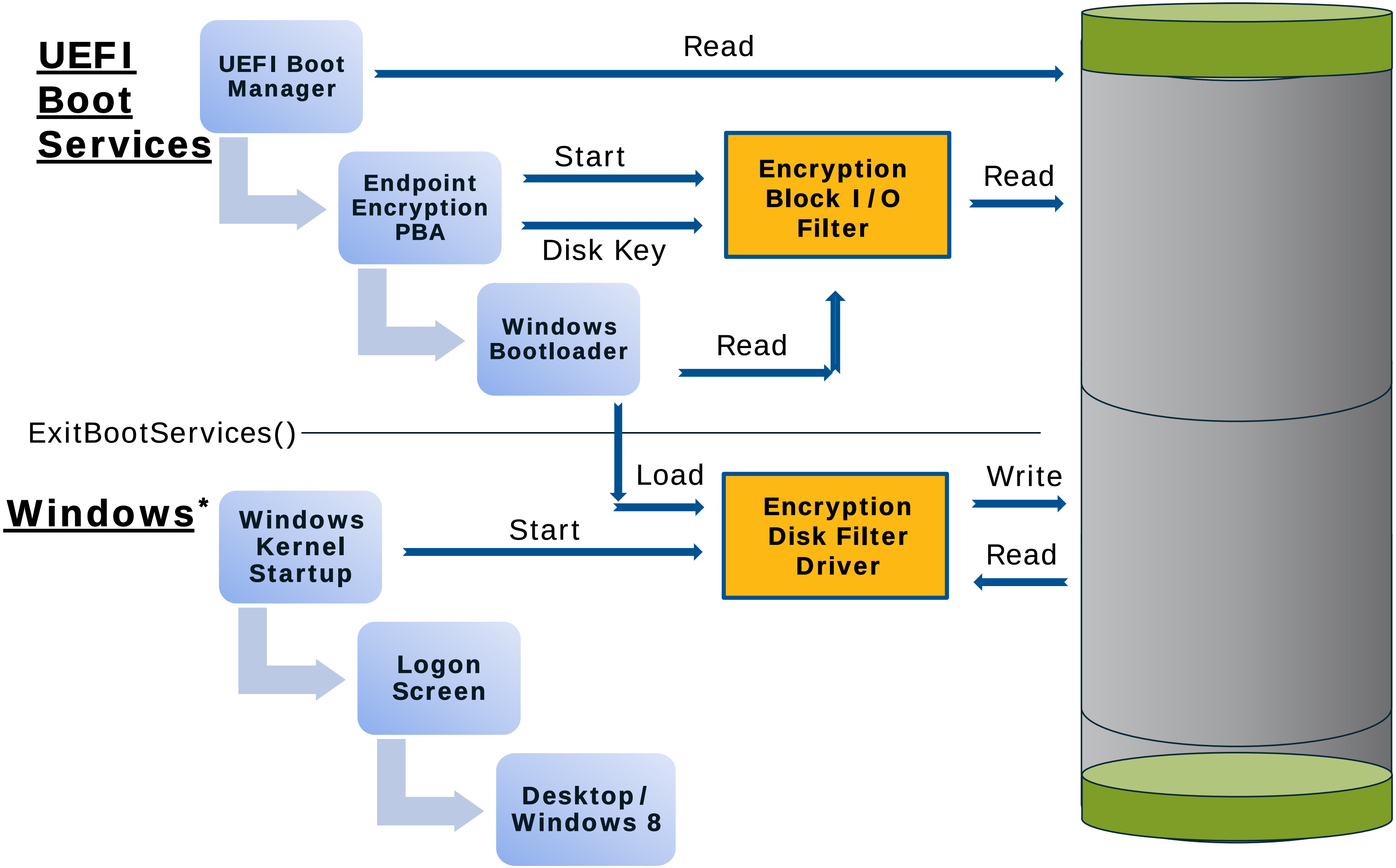
IDF2013 INTEL DEVELOPER FORUM

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



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!

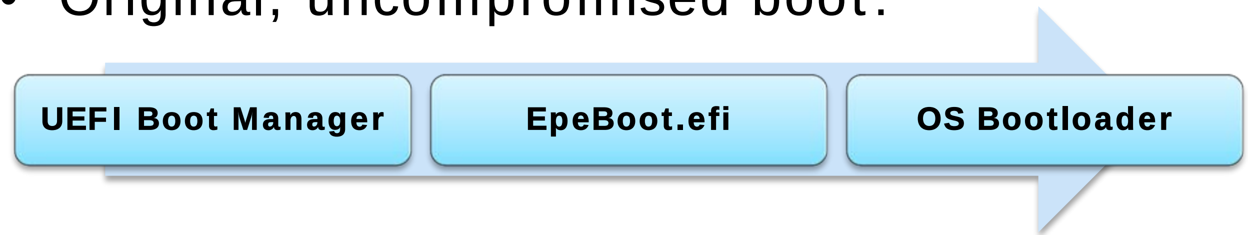
Malware Threat: Keylogger

```
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);
    hooked_protocols[i].st = st;
C   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

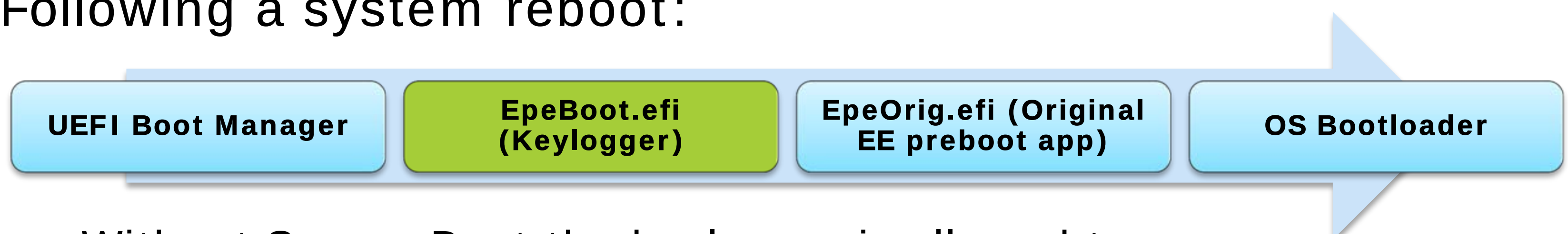
- Original, uncompromised boot:



- 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
```

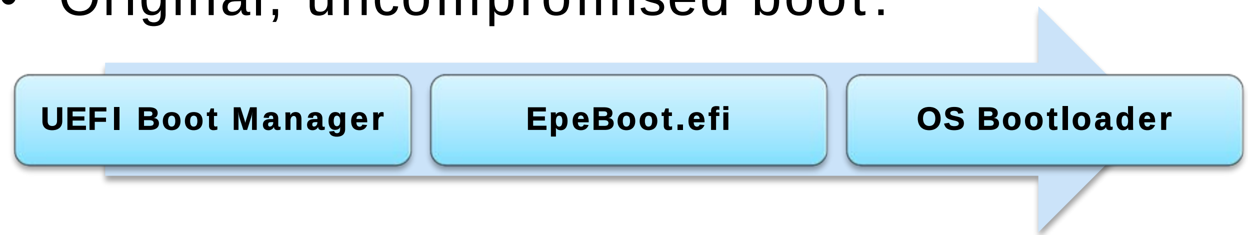
- Following a system reboot:



  - 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
```

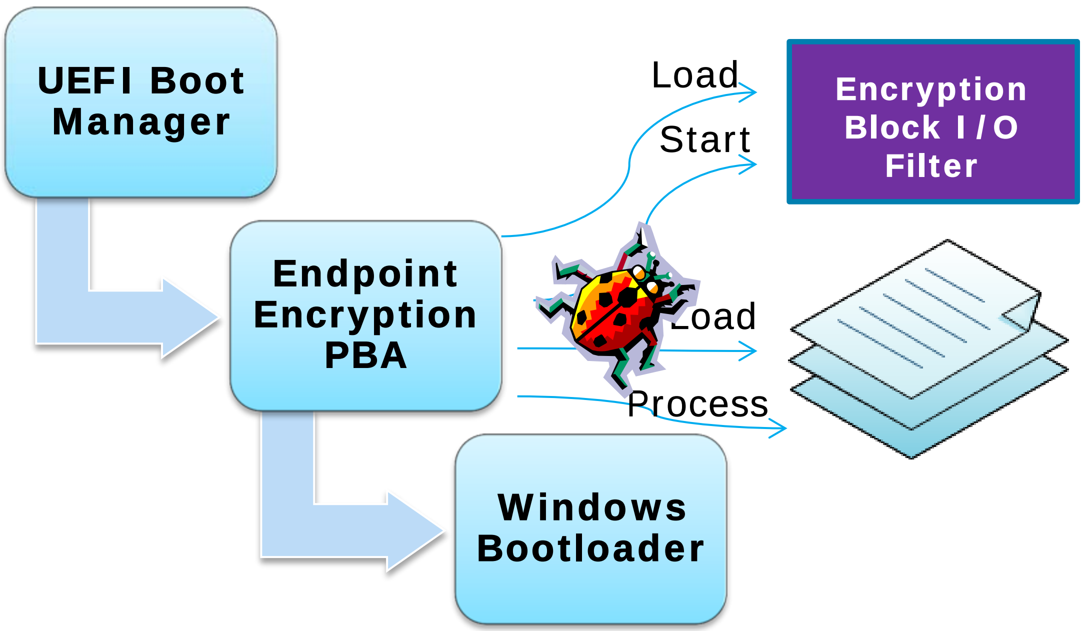
- Following a system reboot:

  - 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

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

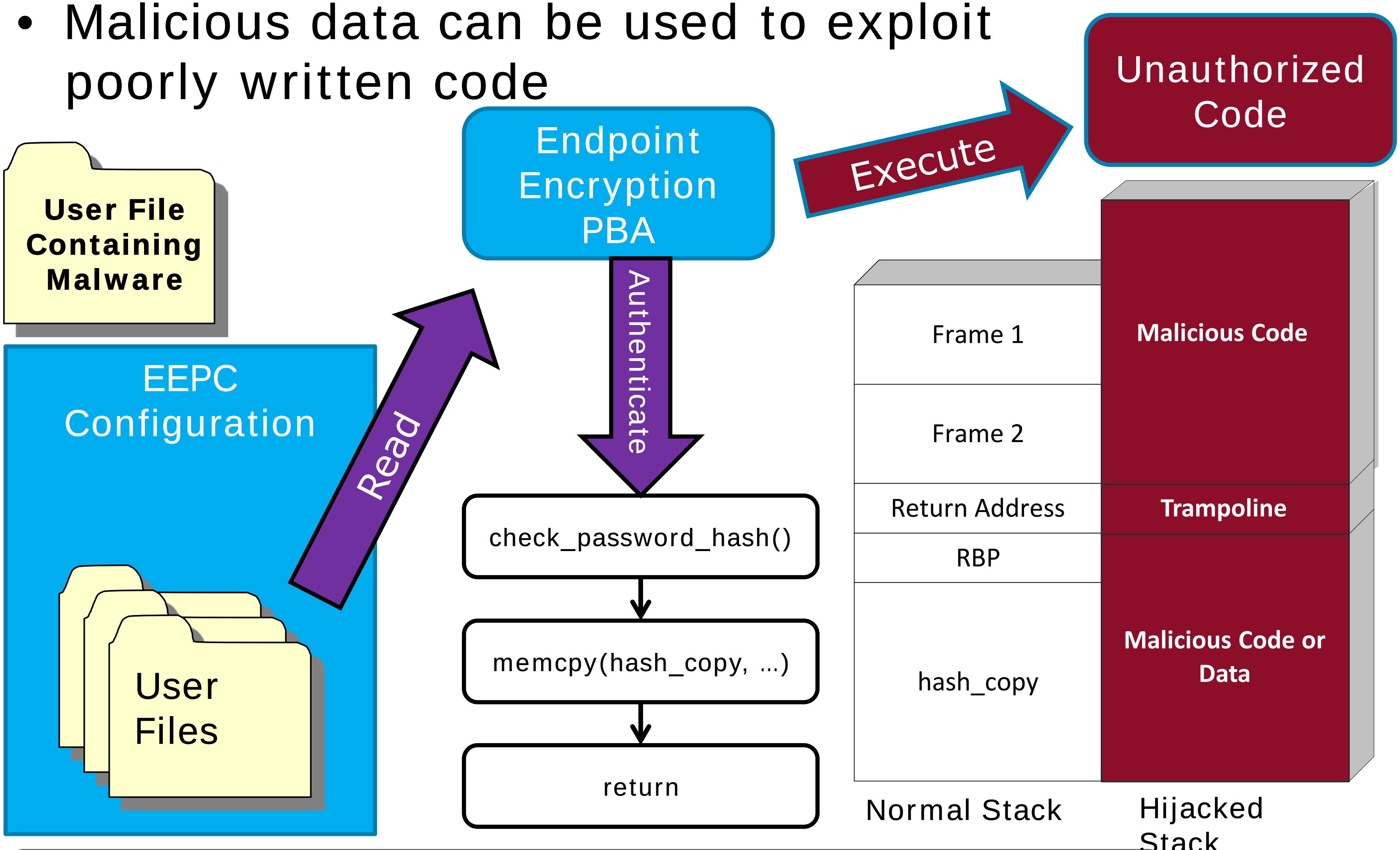
Data File Threat

```
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!

IDF2013 INTEL DEVELOPER FORUM

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

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, Sponsors of Tomorrow and the Intel logo are trademarks of Intel Corporation in the United States and other countries.

Legal Disclaimer

- Software Source Code Disclaimer: Any software source code reprinted in this document is furnished under a software license and may only be used or copied in accordance with the terms of that license.
  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 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 first 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; customer acceptance of Intel’s and competitors’ products; supply constraints and other disruptions affecting customers; 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. 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. 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 current chief executive officer plans to retire in May 2013 and the Board of Directors is working to choose a successor. The succession and transition process may have a direct and/or indirect effect on the business and operations of the company. In connection with the appointment of the new CEO, the company will seek to retain our executive management team (some of whom are being considered for the CEO position), and keep employees focused on achieving the company’s strategic goals and objectives. 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, report on Form 10-K and earnings release.

Rev. 1/17/13