Mitigating Exploits, Rootkits and Advanced Persistent Threats

David Durham, Senior Principal Engineer
Intel Corporation

Hot Chips Tutorial
Problem
Better Protection
Solid Foundations
Usages
Minimizing TCB
Summary
Increasingly Sophisticated Attacks

Operation Aurora: "Google announced an attack targeting it and what is believed to be more than 30 other companies....."

CNET January 12, 2010

Stuxnet: "...a novel way to use computers to sabotage an enemy's lifeline infrastructure suggests a powerful new kind of weapon is moving within reach of weak states, militant groups and criminals..."

Reuters Nov 30, 2010

The Heartbleed Vulnerability: What It Is and How It Affects You: "...Heartbleed is not a virus, but rather a mistake written into OpenSSL" April 2014

Banking Malware (SpyEye) Monitors Victims by Hijacking Webcams and Microphones
May 2012, PCWorld

Meet ‘Flame,’ The Massive Spy Malware
"...designed primarily to spy on the users of infected computers and steal data from them, including documents, recorded conversations and keystrokes." May 2012, Wired
Malware Signatures More & More...

- Malware samples continue sharp rise
- Polymorphic viruses
- Methods of packing, redistributing existing malware
- Looking for known malware misses 0-days and targeted attacks
0-Day: Vulnerability, Armed, Exploited
Sample: APSA13-02 Exploit Analysis

Emerging ‘Stack Pivoting’ Exploits Bypass Common Security - APSA13-02 exploit:
Generalized Attack Vectors

Circumvent

Shared Address Space

FuncA  FuncB  Malware

Input Validation

Bypass

Internal Functions

Return Oriented Programming

Disable

Shared Address Space

Damage Resource

Attack External Dependencies

Valid kernel components

Corrupt driver / malicious code

Inject/Modify

Shared Address Space

Data Pages

write write write write

write write write write

write write write write

write write write write

Comp A  Comp B

Buffer Overflow

Eavesdrop

Shared Address Space

Write Secret

Read

Secret Key

Malware
Problem
Better Protection
Solid Foundations
Usages
Minimizing TCB
Summary
Paradigm Shift

Malicious App -> App -> Anti-Virus

Signature -> Behavior

Monitor behavior
Detect onset of the attack
Prevent suspicious access

“Behavior-based” detection to stop zero-day attacks
Putting the Defenses Together

- Trusted Launch
- Measurement
- Software Isolation
- Device Isolation
- Crypto Acceleration
- Introspection Acceleration
- Supervisory Execution Prevention

Forward Looking...
- Buffer Overflow Protections
- Minimal TCB

...
Evolution of Memory Protections

Privilege levels based on protection rings

Multiple virtual machine isolation

Granular isolation within an address space

Page Level Protections

Reduce the attack surface while minimizing overhead
Overlaying Granular Protections

Accommodates existing OS methodology and legacy code
Accelerating Granular Isolation

- Intel® Virtualization Technology (Intel® VT) enables protections beyond the OS
- Overlays additional protections and monitoring policies by enabling memory views
- Provides continuous detection of illicit behaviors
- Accelerated using VM Functions and Virtualization Exceptions...

**Additional Monitoring:** Privileged software monitors OS activity
Extended Page Tables for Isolation within VM

- VM Function (VMFUNC) to switch EPTs under guest
- Virtualization Exceptions (VE) directly notify guest of EPT access violations

Use VMFUNC to cross EPT domains

Memory View 1

VMFUNC

Memory View 2

Extended Page Tables (EPT)

Extended Page Table Pointer List indexed by VMFUNC

Report EPT Violations Via VE

Hypervisor

Intel® VT-x with EPT

EPT Walker

TLBs

CPU0

Physical Pages

#VE

VMFUNC

Virtualization Exceptions (VE)
Granular Isolation (Before)

Execution

View1

VMX Root Monitor

View2

Additional Overhead

R-- (read-only)

R-X (read-execute)

R--

R-X

EPT fault

EPT fault
Granular Isolation with VM Functions

Execution

EPT 1

R-X

VMFUNC 0

R--

R-X

R-- (read-only)

R-X (read-execute)

EPT 2

R--

R-X

VMFUNC 0

R--

Enforce Control Flow Integrity with Intel® VT
Monitoring with Virtualization Exceptions

Execution EPT 1

Guest code Page using VMFUNC0

VMFUNC (0)

R-X

R--

R-X

Guest #VE Handler

R-- (read-only)

R-X (read-execute)

EPT Exceptions directed to the guest
VM Function for Switching

Relative Performance Comparison

Context Switching
- Privilege Level
- State Save

Virtual Machine Switching
- Monitor Overhead
- New Paging Hierarchy

VMFUNC Switching

Varies

Varies

Varies
Layering Virtualization and Introspection

- Root VMM supports VT nesting (with EPT shadowing)
- Root VMM and guest opt-in to enable features
- VMFUNC switches authorized EPTs without engaging VMM(s)
- EPT violations reported via #VE to guest directly
  - No VM Exits for guest policies
  - No additional overhead for VMMs
- Root VMM decides which pages #VE and which will VM Exit
  - Disambiguates copy-on-write and other VMM notification needs

Enables efficient introspection across multiple VMs
Revisiting Attack Vectors

**Circumvent**

Shared Address Space

- FuncA
- FuncB
- Malware

Input Validation

Internal Functions

Bypass

Return Oriented Programming

**Disable**

Shared Address Space

- Damage Resource
- Attack External Dependencies
- Corrupt driver / malicious code

Valid kernel components

**Inject/Modify**

Shared Address Space

- Comp A
- Comp B
- Buffer Overflow

write

write

write

write

write

write

write

write

**Eavesdrop**

Shared Address Space

- Hidden Page
- Write Secret
- Read

Secret Key

Malware

Overlay memory views to monitor software behavior
Overlaying Protections: Things to Consider...

Registers

Memory Mappings

Async Events

May also address external devices depending on usage
Monitoring Processor Registers

• VMM can be configured to intercept changes to:
  – Model Specific Registers
  – Control Registers
  – Debug Registers
  – Descriptor Tables (IDTR...)
  – Mode dependent...

• VMCS determines what registers to monitor

• GPRs including the stack pointer can be checked at boundaries and on events

Monitor processor state to prevent attacks
Page Table Edit Control

CR3 Register

CR3 Target List

Virtual Address

Guest Physical Memory

EPTP

Host Physical Memory

Write Protect

Prevent attacks that remap virtual memory
Interrupts & Asynchronous Events

- Protect Interrupt Descriptor Table & Register
- Trust Interrupt Service Routines or own ISR stub
- Stub code protects state
  - Stack
  - General Purpose Registers
Devices

- Device space configuration
  - Programmed IO in/out
  - Relocation of device registers in memory/BAR change
  - Trigger VM Exit

- Memory Mapped IO
  - Device registers
  - Covered by EPT policy

- DMA
  - Buffers in memory
  - Covered by EPT policy

Intel® VT-d protects against compromised devices
McAfee® Deep Defender Overview

- McAfee® DeepSAFE™ technology in the McAfee® Deep Defender product can safely monitor writes to critical memory assets
- The Deep Defender component within the operating system understands the O/S layout and rootkit techniques
- The DeepSAFE component uses CPU primitives to monitor CPU and memory so that pages containing sensitive code and data are access-controlled
Providing Better Protections

Input

Audio

Storage

Video
Problem
Better Protection
Solid Foundations
Usages
Minimizing TCB
Summary
Democratizing Security

- Software relies on a Trusted Execution Environment (TEE) in case other defenses fail
- Isolated hardware and restricted modes limit use
- A future where there are enough TEEs for all?

Scaling Trusted Execution Environments for the many
Trust How Many?

- Trusted Computing Base (TCB):
  - Set of all hardware, firmware & software part of a trusted environment
- Minimizing the TCB:
  - Remove Software Stacks
  - Remove Drivers
  - Remove Devices
  - Remove Firmware...
Trust The Processor
Scaling Trust with a Minimal TCB

- Define precise trust boundaries
- Remove extraneous components from the trust boundary
- Reduce TCB to the processor

Threat surface reduction

**Scalable security within mainstream environment**

Utilize Existing Instruction Set Security Primitives

E.g. Intel® Advanced Encryption Standard New Instructions:

- **AESKEYGENASSIST**
  - ShiftRows()
  - SubBytes()
  - MixColumns()
  - AddRoundKey()

- **AESIMC**
  - InvShiftRows()
  - InvSubBytes()
  - AddRoundKey()

- **AESENC**
  - InvShiftRows()
  - InvSubBytes()
  - InvMixColumns()
  - AddRoundKey()

- **AESENCLAST**
  - ShiftRows()
  - SubBytes()
  - AddRoundKey()

- **AESDEC**
  - InvShiftRows()
  - InvSubBytes()
  - InvMixColumns()
  - AddRoundKey()

- **AESDECLAST**
  - InvShiftRows()
  - InvSubBytes()
  - AddRoundKey()

- **... And many more**

- **Enclave**
- **Enclave**

- **OS**
- **VMM**

Summary

• Increasingly sophisticated attacks require better defenses
• Moving from signatures to behavioral models
• Next generation processors deliver new capabilities for advanced software monitoring and protection
• Ability to layer protections over legacy software
• Minimizing the Trusted Computing Base is the next step...
Biography

David Durham is a Senior Principal Engineer and Director in Intel Labs. His research team developed anti-malware and cryptographic security features currently found in hundreds of millions of Intel processors. David also developed policy-based network management technologies, created security solutions shipping in Intel® vPro™ platforms and worked with McAfee to deliver virtualization-based anti-malware products. Collaborating with industry leaders, his team developed IEEE 802.1 security protocols and advanced network access control capabilities now embedded in tens of millions of Intel platforms. He is a prolific author on computer communications, having written a book, multiple publications and several Internet protocol standards deployed in millions of connected devices. David received two Intel Achievement Awards, was granted over 100 US and international patents and earned his B.S. and M.S. degrees in Computer Engineering from Rensselaer Polytechnic Institute.
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. 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® Virtualization Technology (Intel® VT) requires a computer system with an enabled Intel® processor, BIOS, and virtual machine monitor (VMM). Functionality, performance or other benefits will vary depending on hardware and software configurations. Software applications may not be compatible with all operating systems. Consult your PC manufacturer. For more information, visit http://www.intel.com/go/virtualization.

Intel, Look Inside 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.
Copyright ©2013-2014 Intel Corporation. All rights reserved.