Under the Hood: How Actaeon Unveils Your Hypervisor

Mariano Graziano and Andrea Lanzi

Eurecom, Sophia Antipolis, France, EU

HITB Kuala Lumpur, October 2013

Table of Contents

Eurecom S3 Security Lab

Actaeon Motivations

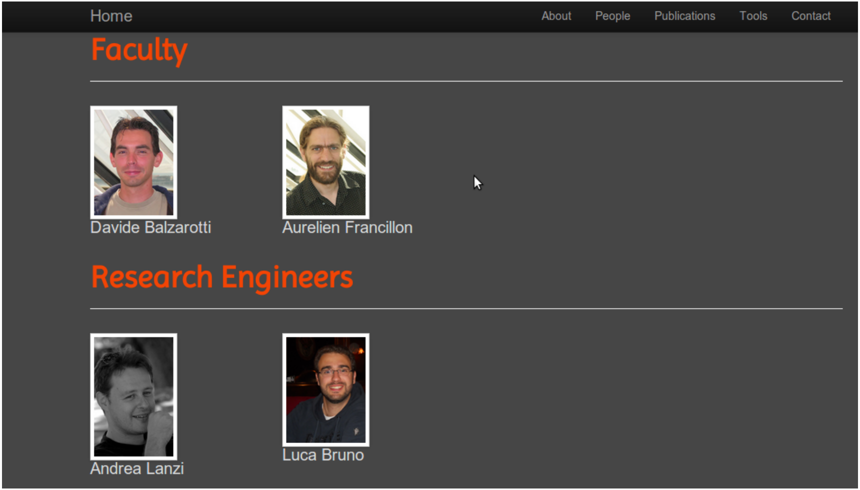
- What's the problem we are facing
- Why this tool should be useful
- What Actaeon is not
- What is Actaeon?

Actaeon Internals

- VMCS Reverse Engineering Layout
- Locate Hypervisor in Memory
- Nested Virtualization: Turtle Technology
- Extended Page Table: (EPT)

Conclusion

Eurecom S3 Security Lab



Faculty

Davide Balzarotti

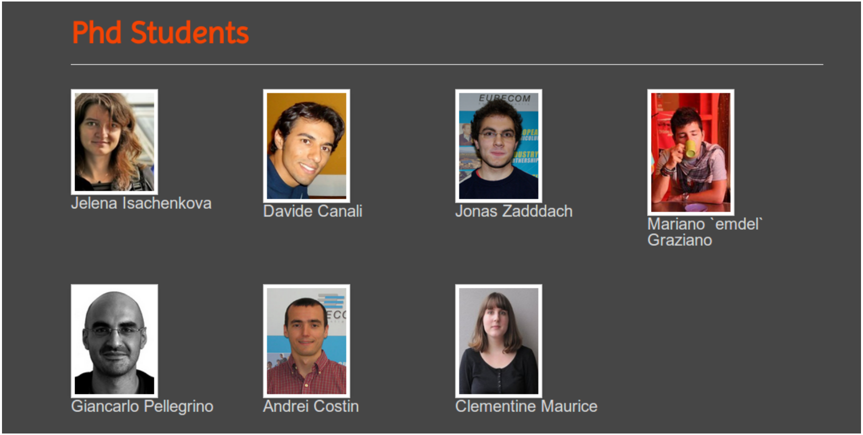
Aurelien Francillon

Research Engineers

Andrea Lanzi

Luca Bruno

Eurecom S3 Security Lab



Phd Students

Jelena Isachenkova

Davide Canali

Jonas Zadddach

Mariano `emdel` Graziano

Giancarlo Pellegrino

Andrei Costin

Clementine Maurice

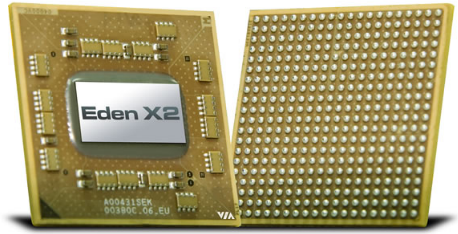
Eurecom S3 Security Lab

Main Topics of our research lab:

- Exploitation Techniques.
- Reverse Engineering.
- Hardware Security: Firmware Analysis, Scada Systems.
- Web Security.
- Virtualization: Malware Detection.
- Forensics Analysis.

Check out our publications: `http://s3.eurecom.fr/publications.html` Looking for motivated and Skilled PhD students!!!

What's the problem we are facing?



Starting from a physical memory dump we design a system for **recognize/analyze Hypervisors in memory** and provide an interface for **extract/analyze the virtual machines memory** (HW-assisted Intel Virtualization).

Why this tool should be useful?

- Virtual machines are everywhere nowadays (Clouds technologies, Multiple running OSs, Security Solutions etc.)
- There's no forensic tool available for an automatic analysis of Virtual Machines.
- Even Volatiliy, the facto standard for Forensics memory analysis does not provide any tools/plugins for analyzing the virtual machines.

What Actaeon is not

Actaeon is not:

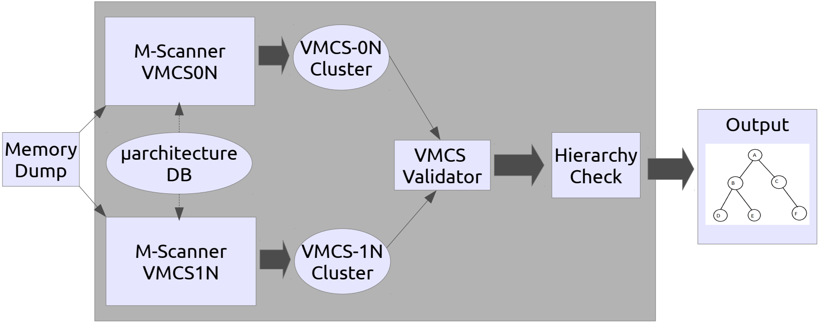
- A real-time detector for malicious Hypervisors.
- A forensic tool for Physical Memory Dump.
- A malware detector.

What is Actaeon?

Actaeon is a forensic Analyzer based on Volatiliy core that is able to spot and analyze hypervisors from a physical memory dump. Actaeon provides three main functionalities:

- **VMCS Layout Extractor**: This component is designed to extract the exact layout of a VMCS, by implementing a reverse engineering algorithm.
- **Hyper-ls**: This component is implemented for the Volatility framework, Its goal is to scan the memory image to extract the VMCSs memory structures.
- **Virtual Machine Introspection Patch**: Acteon provides a transparent interface for analyzing physical memory of virtual machines.

Actaeon Architecture Analyzer



VMCS Reverse Engineering Layout

- The core of the Actaeon detection **is based on finding out VMCS** memory structures.
- VMCS is the memory structures that contains information for **keeping the state of the system** when it switches from Hypervisor to virtual machines and viceversa (not-root-mode/root-mode).
- **VMCS bitmap** fields set up which operations hypervisor will intercept during the system execution.
- **VMCS fields are defined by the Intel manual** but the Layout is micro-architectures dependent and it is not known a prior.

VMCS reverse engineering Layout

By using the initialization hypervisor code (e.g. HyperDbg) we devise a reverse algorithm :

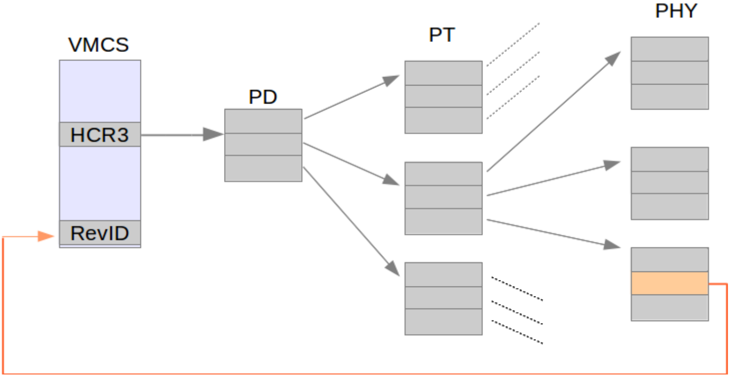
- Every field in the VMCS is associated with a **32 bits value, (encoding)**, that needs to be provided to the VMREAD/VMWRITE instructions to specify how the values has to be stored.
- Since **the position (encoding) of the field is encapsulated into the instructions processor microcode** in order to discover that, we fill the VMCS with some strings related to that field (typically numbers).
- Then we simply associate the encoding values utilized for filling a particular VMCS fields with the chosen numbers values and **we can re-build the position of every fields in the VMCS** memory structures.

DEMO:Reverse Engineering VMCS Layout

Locate Hypervisor in Memory (VMCS Clustering)

- We base our search on some particular fields values defined inside the VMCS memory structure:
  - **Revision ID**: It is the identifier that determines the layout of the rest of the structure. For the VMCS of the top hypervisor, this field has to match the value of the IA32 VMX_BASIC_ MSR.
  - **VMX ABORT INDICATOR**: This is the VMX abort indicator and its value has to be zero. The field is the second entry of the VMCS area.
  - **VmcsLinkPointerCheck**: The values of this field consists of two consecutive words that, according to the Intel manual, should be set to 0xffffffff.
  - **Host CR4**: This field contains the host CR4 register. Its 13th bit indicates if the VMX is enabled or not.

Locate Hypervisor in Memory (VMCS Validation)

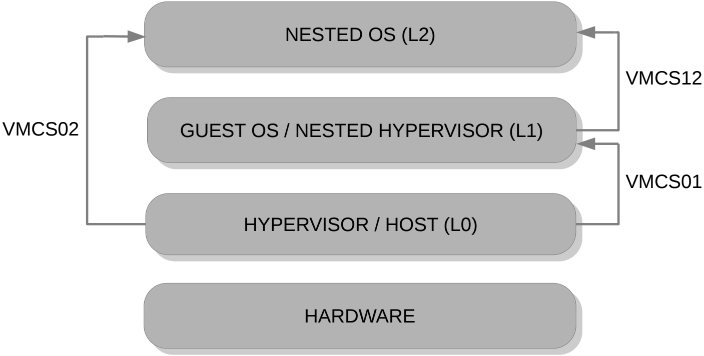


DEMO: Locate Hypervisor in Memory

Nested Virtualization: Turtle Technology

- Most of the commodity hypervisors support nested virtualization, **extracting the hierarchy of nested hypervisors could help an analyst** to see what is running inside the system.
- **KVM and Xen** implement it using the **Turtle approach**, and a similar technique to multiplex the inner hypervisors VT-x/EPT is **also used by VMware**.
- By looking for the nested VMCS structure or by recognizing the VMCS of a Turtle-like environment, **Actaeon can provide an extensible support to reconstruct the hierarchy of nested virtualization.**

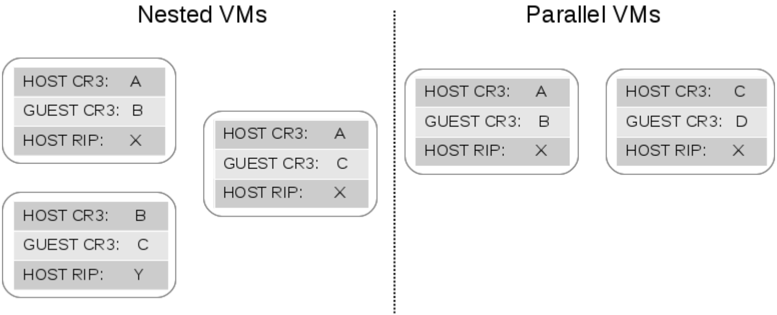
Nested Virtualization: Turtle Technology



Virtualization Hierarchy Analysis

- **We can infer the hierarchy between the hypervisors and distinguish between parallel and nested VMs** by comparing the values of three fields: the GUEST CR3, the HOST CR3, and the HOST RIP.
- In a nested setup we can have **two different hypervisors** (represented by the **two different HOST RIP** addresses) while for **parallel virtual machine** the hypervisor is the same (**same value of HOST RIP**).
- **By comparing the GUEST CR3 and HOST CR3 values we can distinguish among VMCS01, VMCS02, and VMCS12** in a nested virtualization setup.

Virtualization Hierarchy Analysis



Nested VMs

HOST CR3: A
GUEST CR3: B
HOST RIP: X

HOST CR3: B
GUEST CR3: C
HOST RIP: Y

HOST CR3: A
GUEST CR3: C
HOST RIP: X

Parallel VMs

HOST CR3: A
GUEST CR3: B
HOST RIP: X

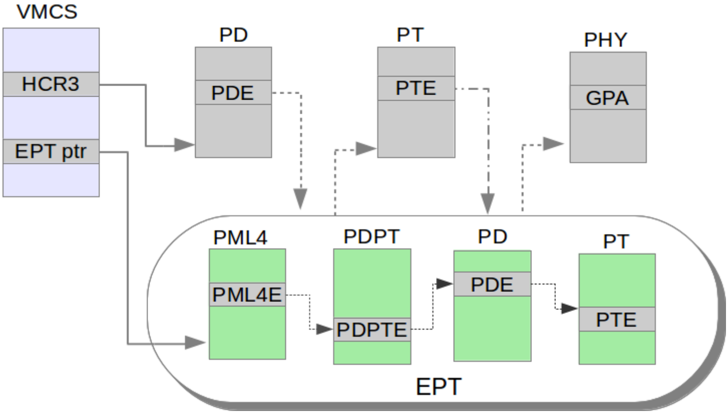
HOST CR3: C
GUEST CR3: D
HOST RIP: X

DEMO: Nested Virtualization

Extended Page Table: (EPT)

- **EPT technology provides memory isolation among different virtual machines.** It is marked with a dedicated flag in the Secondary Based Execution Control Field in the VMCS structure.
- EPT **is a transparent hardware mechanism that basically provide another level of translation** from virtual address of guest OS and the real physical memory page.
- The translation happens through different stages involving **four EPT paging structures** (namely PML4, PDPT, PD, and PT).

Extended Page Table: (EPT)



Extended Page Table: (EPT)

- **First of all we recognize and extract the EPT pointer** inside the VMCS memory structure.
- Then **we simulate the EPT translation** by programmatic walking through the PML4, PDPT, PD, and PT tables for each address that need to be translated.
- **We patch the core of Volatility, all the address translation functions**, in order to have the chance to execute all the plugins without modifying the code.

DEMO:Extended Page Table (EPT)

Conclusion

- We present the **basic background concepts about HW-assisted virtualization**.
- **We describe the internals of Actaeon** as a forensic tool analyzer for virtual machine environment.
- **We show a practical forensic analysis session** on how to use Actaeon for analyzing virtual machine.

References & Contacts

Actaeon **won the 1st Volatility Plugin Contest - 2013/08**

- **Prof. Balzarotti and Prof. Francillon** e-mails: (davide.balzarotti@eurecom.fr,aurelien.francillon@eurecom.fr)
- **Actaeon web page**: http://s3.eurecom.fr/tools/actaeon/
- **Our lab web page**: http://s3.eurecom.fr/
- **Authors/Speakers**:
  Mariano Graziano (graziano@eurecom.fr) (@emd3l),
  Andrea Lanzi (lanzi@eurecom.fr)

Q&A

Thank You!
Q&A?