Memory Forensics using Virtual Machine Introspection for Cloud Computing

Tobias Zillner, BSc MSc MSc
ABOUT ME

Tobias Zillner, BSc MSc MSc

- Vienna, Austria
- Founder of Zillner IT-Security
- Independent Security Consultant & Researcher
  - Consulting, Audit, Advisory, Training
- Security Research
  - Internet of Things, Smart Homes
  - Wireless Security
- www.zillner.tech

SDR Enthusiast
WHAT IS IT ABOUT?
AND WHY DO WE NEED IT?
OUTLINE

- Introduction & Background
- Virtual Machine Introspection (VMI)
- Use cases
- Prototype
- Summary
MOTIVATION

- Relocation of systems and services into cloud environments is on the rise
- Users lose direct access / control over their systems
- Forensic methods are limited in the cloud
- Enable the user to perform their own forensic investigations
- Forensic as a Service
MEMORY FORENSICS & VIRTUAL MACHINE INTROSPECTION
FORENSIC PROCESS

Collection → Examination → Analysis → Reporting

Media → Data → Information → Evidence
HARDWARE VIRTUALIZATION

- One / Multiple guest OS on virtualized hardware
- Managed by Virtual Machine Monitor (VMM) – Hypervisor
- Provides interfaces and controls interactions with hardware
  - CPU, memory, network, storage,…
- Hypervisor on own OS – Host OS
**Native vs. Hosted Virtualization**

**Native virtualization**:
- Hardware
- Hypervisor
- Guest OS
- Applications

**Hosted virtualization**:
- Hardware
- Hypervisor
- Host OS
- Applications

**Key Components**:
- Hardware
- Hypervisor
- Guest OS
- Host OS
- Applications
Virtual Machine Introspection

“Virtual Introspection (VI) is the process by which the state of a virtual machine (VM) is observed from either the Virtual Machine Monitor (VMM), or from some virtual machine other than the one being examined.”

SEMANTIC GAP

- Difference between the presentation of data from volatile memory by the OS and the raw data format

- Requires VMI to perform the same translation of the raw memory data as the OS

- At least some knowledge about the guest OS is necessary
HOW DOES IT WORK?

http://libvmi.com/docs/gcode-intro.html
ADVANTAGES

- No altering of the target system
- Very hard to detect the monitoring
- Live analysis of memory content
- Data size for analysis (storage much larger than memory)
- Detection of advanced memory only malware
- More reliable data
  - No data corruption through malware
COUNTERMEASURES

Detection
- **Timing analysis** - unusual patterns in the frequency at which it is scheduled for execution
- **Page fault analysis** - the target VM may be able to detect unusual patterns in the distribution of page faults

Direct Kernel Structure Manipulation (DKSM)
- VMI assumes that OS implement certain kernel- and data structures
- DKSM modifies this structures and prevents monitoring
- **Syntanx based**: targeted deletion/addition/manipulation of data structures
- **Sematic**: semantics of the data structures are changed
- **Combined**: mix of syntax and semantics manipulation
FIELDS OF APPLICATION

EXAMPLES

_rootkit detection_
  - Manipulation of memory access
  - Interception of system calls

_Cryptographic key extraction_
  - On the fly encrypted container
  - Network forensics

_IDS / IPS_
Prototype
SOLUTION APPROACH

- Combining existing tools for a novel approach
- Open Source
- Minimal overhead
- Transparent for the user
ARCHITECTURE

- Cloud Solution
  - Open Nebula
- Cloud Management Server
- Cloud Node
- Host OS - Ubuntu
- Guest VM
- Memory Forensic Services
- VMI Library – LibVMI
- Forensic Tool – Volatility
- Hypervisor - Xen
OPEN NEBULA EXTENSIONS
MEMORY FORENSIC SERVICES

- Self developed management and control services
- Client – Server model
- Platform independent
- PKI for secure communication
- Command whitelisting

1. onevm memfor <VMID> <Volatile Profil> <Volatile Kommando>
FORENSIC PROCESS

Collection  Examination  Analysis  Reporting

Media  Data  Information  Evidence
Forensic Process

- OS on Cloud Node
- Data provided by LibVMI
- Collected by Volatility

Collection → Examination → Analysis → Reporting

Media → Data → Information → Evidence
FORENSIC PROCESS

- OS on Cloud Node
- Collected data checked by Volatility
- Data extraction for forensic purpose

Collection → Examination → Analysis → Reporting

Media → Data → Information → Evidence
FORENSIC PROCESS

- Collection
- Examination
- Analysis
- Reporting

Media → Data → Information → Evidence

- Partially OS on Cloud Node
- Collected data checked by Volatility
- Partially on user system
- Analysis with additional tools by user
FORENSIC PROCESS

- Completely on user system

Collection → Examination → Analysis → Reporting

Media → Data → Information → Evidence
ADVANTAGES

- User gets easy access to the data
- No changes on the target VM necessary
- Memory analysis not on the possibly compromised system
- No stop/pausing of the analyzed machine required
- Operation of the VM does not get influenced
- Analysis can be done either local or over the network
  - Reduction of local load / network load
- Usage of existing authentication and authorization system
DISADVANTAGES

- Configuration necessary
- Knowledge about the guest OS required
- Installation overhead for cloud provider
- Additional attack surface
- Security is crucial for the added services
- User segregation is very important
LibVMI Config Example

```c
UbuntuLucid {
    sysmap = "/usr/local/libvmi-0.8/Systemmaps/vm1lucid/
            System.map-2.6.32-45-server";
    ostype = "Linux";
    linux_name = 0x490;
    linux_tasks = 0x258;
    linux_mm = 0x290;
    linux_pid = 0x2b8;
    linux_pgd = 0x50;
    linux_addr = 0x100;
}
```
Volatility / libVMI usage

1. python vol.py -l vmi://win7 pslist  # win7 is the target
USE CASE

KERNEL LEVEL ROOT KIT DETECTION

- Modifying of data structures, which display the processes currently running on the system
- System call interception
- Interrupt hooking
- Modifying the kernel memory image
- Intercepting calls handled by the VFS
- Virtual memory subversion
USE CASE

ENDUSER VM IN IAAS CLOUD
DEMO
SUMMARY

- Investigations in cloud environments get more and more common
- Hypervisor forensics VMI is a very interesting solution approach
- Fully Open Source based working prototype
- Enables fast responses to security incidents
- Lot of room for enhancements
- Different use cases for VMI in clouds possible
Black Hat Sound Bytes

- Hypervisor forensics / VMI are very powerful and interesting technologies
- FaaS gives power to the end user
- Memory analysis is a huge benefit for forensic investigations
Q & A

Please fill out the Black Hat Feedback Form
Contact

Tobias Zillner
tobias@zillner.tech
www.zillner.tech
+43 664 8829 8290