Dynamic Malware Analysis Using IntroVirt: a Modified Hypervisor-Based System

Joshua White
whitej@ainfosec.com

Senior Computer Engineer
Assured Information Security
http://ainfosec.com

Date: April 30th, 2013
Release: Unclassified // Public

Copyright 2012 Assured Information Security, Inc.
Outline:

Traditional static analysis
Dynamic Analysis
Virtualization
Virtual Machine Introspection (VMI)
IntroVirt
Future Work
Conclusion
Static Analysis:

- Involves powering the target system down
  - Creating a copy of the OS
  - Data in memory typically lost
    - Unpacked malware disappears
- State of the machine lost
  - process list
  - open network ports
  - installed kernel modules
Dynamic Analysis:

- Examination of a system while running
  - Can't examine a live system without making changes
    - Investigator logs on to the system
    - Logs are recorded
    - Temporary files are created and deleted
    - Network connections can be opened/closed
    - History files are updated
    - Registry entries are queried/added/modified

- System may have malware that hides:
  - open network ports, user accounts, presence of files and folders in a file system
  - A system could be configured to detect a live analysis attempt
Specialized Hardware:

- Some hardware based devices available
  - Address the danger inherent in live analysis
  - Use the DMA controller to acquire system memory
    • no operating system or CPU interaction
- Risk:
  • Memory spoofing: results are different than what the actual CPU sees
Virtualization:

- Hypervisor, VMM – Virtual Machine Monitor
  - software layer that allows several virtual machines to run on a physical machine
  - The physical OS and hardware are called the Host
  - The virtual machine OS and applications are called the Guest

### Type 1 (bare-metal)
- Hypervisor
- Hardware
- VM1
- VM2

### Type 2 (hosted)
- Hypervisor
- OS
- Hardware
- VM1
- VM2
- Process

VMware ESX, Microsoft Hyper-V, Xen

VMware Workstation, Microsoft Virtual PC, Sun VirtualBox, QEMU, KVM
- Not 100% bare metal
- Para-Virtualization

• guest operating system is modified
• guest runs in parallel with other guests
• 2 x Domains
  – Dom-0 = Privileged Domain
  – Dom-U = User Domain
• Dom-0 supports analysis applications
**XEN Structure:**

* XEN 3.0 Logical Structure, http://www.xen.org
VM Introspection:

- Observing the state of a VM
  - Occurs at either the VMM or another VM run by the same Hypervisor
  - Allows for live system analysis
    • target system unchanged
  - By definition: analysis in such a way that the target system is unable to detect monitoring
Typical VMI System:

- Dozens of systems exist, typically they:
  - Pause operation of the targeted Guest
  - Map some of its memory into the Dom0
  - Acquire and decode memory pages
  - Resume operation of the target VM
  - Traverse the list of task_structs
  - Reference task_struct data structures
    - process ID
    - process name
    - memory map
    - execution time
VM Introspection Problems:

- Problems Exist:
  - Timing Analysis
  - Page Fault Analysis
  - Detection of OS running on VM
    • Various clues:
      – CPU Info
      – Network Controller
      – Disk Controller
      – Many more...
IntroVirt: an Introspective Hypervisor
- Builds on XEN
  - Potentially many others
- Different: it does not require pausing the VM
- Can “lie” to the Guest OS

* IntroVirt is Trademark AIS Inc.
** IntroVirt was developed under AIS IR&D
IntroVirt:

- Hypervisor Hooks
  - IntroVirt uses XEN
  - The changes to the XEN have been designed to be small and unobtrusive, and are essentially just hooks

- libIntroVirt
  - The libIntroVirt library provides the communication mechanisms for Dom0 to interact with the hypervisor

- libWintroVirt
  - Relies on libIntroVirt, provides introspection features specifically tailored for the Microsoft Windows operating system
IntroVirt Continued:

- Multiple API Language support inc/Python
- Application Interference
- Problems still exist:
  - Potentially can be detected using timing attacks
  - However, solves all other VMI problems researched
- Still being developed (Future Work):
  - Full support for Linux is pending
  - Support for other Hypervisor's is under investigation
- Other efforts are utilizing it now:
  - MAAGI (Malware Analysis and Attribution through Genetic Information)
    - DARPA/AFOSR
Thank You!

Joshua White
whitej@ainfosec.com

Senior Computer Engineer
Assured Information Security
http://ainfosec.com

Date: April 30th, 2013
Release: Unclassified // Public