Flying Spaghetti Hypervisor

Fionnbharr Davies / thoth

About me

- Finishing up a comp sci degree @ UNSW in Sydney, Australia.
- Starting work at Security Assessment next year.
- Worked previously at Sensory Networks on breaking their IPS software.

Introduction

- New hiding technique for malicious malware such as rootkits
- How the work and some implementation details
- Rootkits
- Hypervisor shims
- Detection techniques

Virtualisation

- What is a hypervisor?
- Full Virtualisation

Rootkits

- Used to hide than an attacker has root
- Generally for malicious purposes
- 4 Different types presented in this talk, based from Rutkowska’s malware taxonomy.

Type 0

- Userland based
- Old style but still in use
- Can be detected fairly easily

Type I

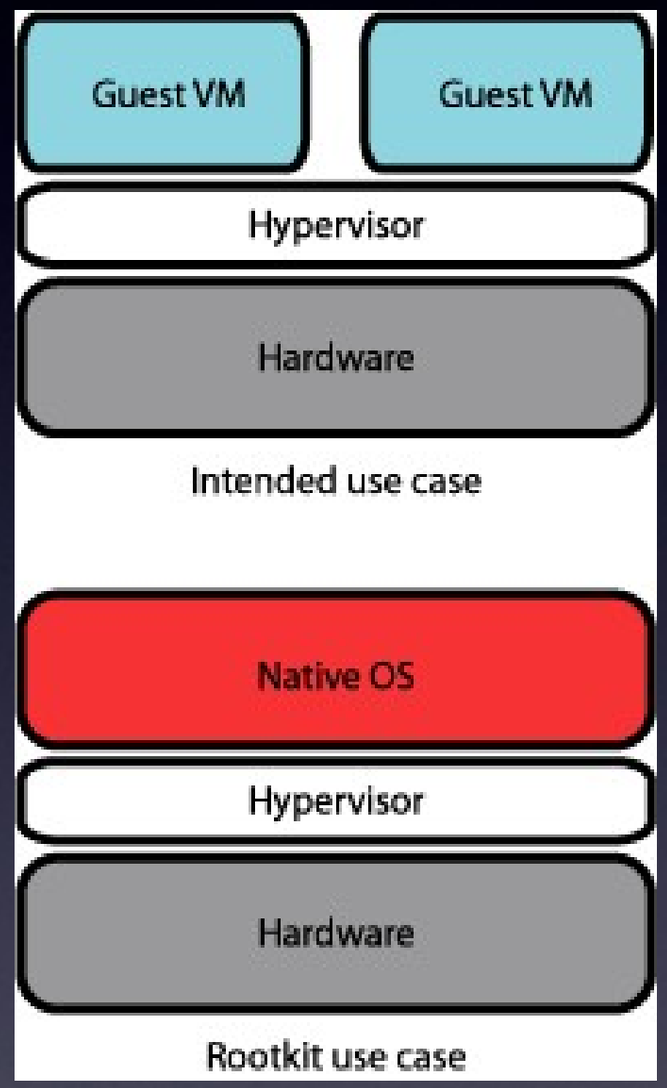
- Lives in the kernel
- Infection techniques
  - LKM
  - /dev/kmem (or /dev/mem)
- Changes parts of the kernel that shouldn’t be changed.
- Don’t _really_ have any really good ways to detect them

Type II

- Also in the Kernel
- Lives in dynamic parts instead of static
- Prrf example

Type III

- Very cool.
- Lives outside the OS
- Uses new hardware extensions to assist virtualisation
- Turns current OS into a VM guest on the fly (‘hyperjacking’)
- VMX root (ring –1)
- Non-intrusive
- No real change in the guest OS



* image based from Lawson, Goldsmith and Ptacek at BH07

‘Hyperjacking’

+

*actual photo of hntr’s coc

From the Intel Spec...

“There is no software-visible bit whose setting indicates whether a logical processor is in VMX non-root operation. This fact may allow a VMM to prevent guest software from determining that it is running in a virtual machine.”

Flying Spaghetti Hypervisor

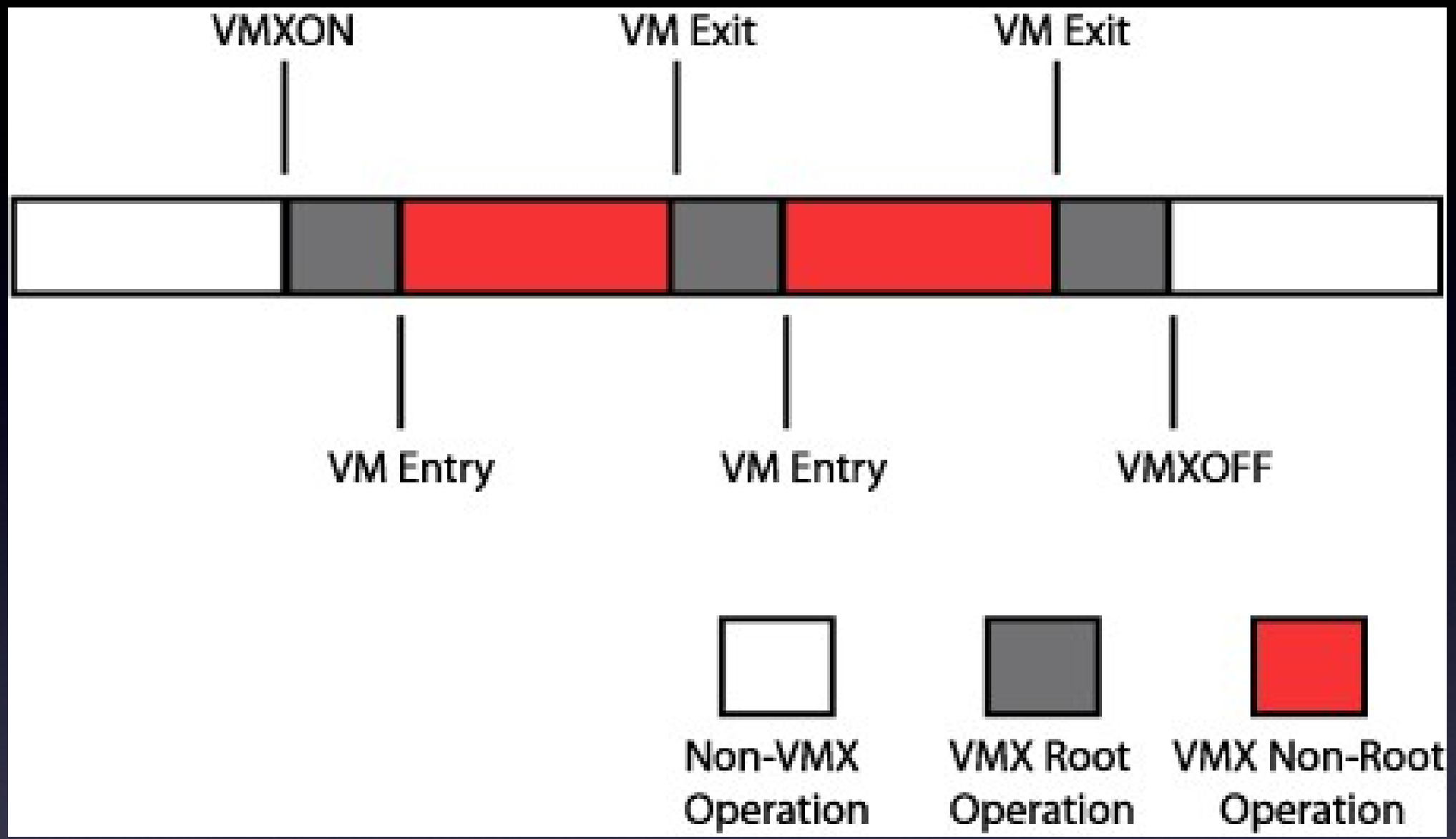
- My incarnation a type III rootkit
- Based heavily off mobydefrags VMM framework & a touch of Xen.
- LKM
- memory resident
- 2 Main sections
  - VM Setup
  - VMM handling code

VM Setup

- Fill the Virtual Machine Control Structure (VMCS)
  - Control Registers (CR0, CR3 and CR4)
  - Selector fields for the segment registers (CS, SS, DS, ES, FS GS and TR)
  - Base address fields (for FS, GS, TR, GDTR and IDTR; RSP, RIP and some MSRs)
  - VM-exit & entry control fields
  - VM-execution control fields
- Hit go.

VMM Handler

- The part of the program that handles VM-Exits
- What it handles
  - VM* Instructions
    - VMCall - Uses this as the backdoor
    - INVD (Invalidate Internal Caches)
    - Read/WriteMSR
    - CPUID
    - MOV to/from CR3


VMXON
VM Exit
VM Exit
VM Entry
VM Entry
VMXOFF
Non-VMX Operation
VMX Root Operation
VMX Non-Root Operation

Rootkit/Supervisor

- 2 Different modes of operation
- Rootkit
  - has a backdoor using vmcall (could use any of the previous list but vmcall is easy)
- Supervisor
  - doesn’t let any other hypervisors to run while it is running.

Hypervisor Shim / Supervisor

- Lowest and first wins
- Smallish impact on the performance
- Only allow verified VM’s or hypervisors to be loaded up.
- Operating system’s already shipping with their own hypervisors (Solaris has Logical domains since 2001, Linux kernel now has KVM as of 2.6.20, Microsoft has Viridian in Windows Server 2008 RC0, OSX doesn’t need functionality since its pretty.)

‘You are absolutely deluded, if not stupid, if you think that a worldwide collection of software engineers who can’t write operating systems or applications without security holes, can then turn around and suddenly write virtualization layers without security holes.’

Theo de Raadt - Oct 24th 2007 to the OpenBDS misc mailing list

Detection

- Nested VM’s
- Latency
  - CPUID & RDTSC
  - other counters (performance, HPET, ACPI and some MSRs).
  - abusing multiple cores
- TLB & CPUID
- Direct Memory Access (DMA)
- CPU Errata
- External Hardware Timing
- and more...

Detection Issues

- Detecting virtualisation != detecting a type III rootkit <-- Rutkowska’s point
- Detection techniques are all hacks and errata.
  - Even the ‘simple’ detection techniques are tricky when you try and do them.
  - That kinda sucks.

Should we be scared?

- Not at the moment.
- They’re cool but at the moment impractical
  - Have to have VMX turned on and be on some of the newer hardware.
- Kernel is too sweet a place to hide.
- OS’s will probably have shims in them soon. Maybe.

What virtualisation hax should we be scared of?

- Attacking running VMMs
  - Xbox360 hack
    - injected code into a closed source hypervisor in a non-friendly environment.
  - A bunch of new bugs in VMware lately
  - A VMM exploit could potentially be better than a root exploit in the future.

Conclusion?

- Don’t panic, we’re going to be allllllright.
- These rootkits are cool but I don’t think they’re ever going to be a problem.
- Source code to mine will be out next week, check www.cse.unsw.edu.au/~fdavies or www.loweruppermiddleclass.com

Acknowledgements

Richard Buckland (My thesis supervisor)

Bob Krouse, Joanna Rutkowska, Dino Dai Zovi and Thomas Ptacek

Over The Wire / Pull The Plug

#ruxcon

Lower Upper Middle Class Crew (LUMC eastern suburbz unite)

Questions?