
Virtualization and Security

Rachel Greenstadt

Center for Research on Computation and Society
Harvard School of Engineering and Applied Sciences

Manizales, Colombia
October 2007



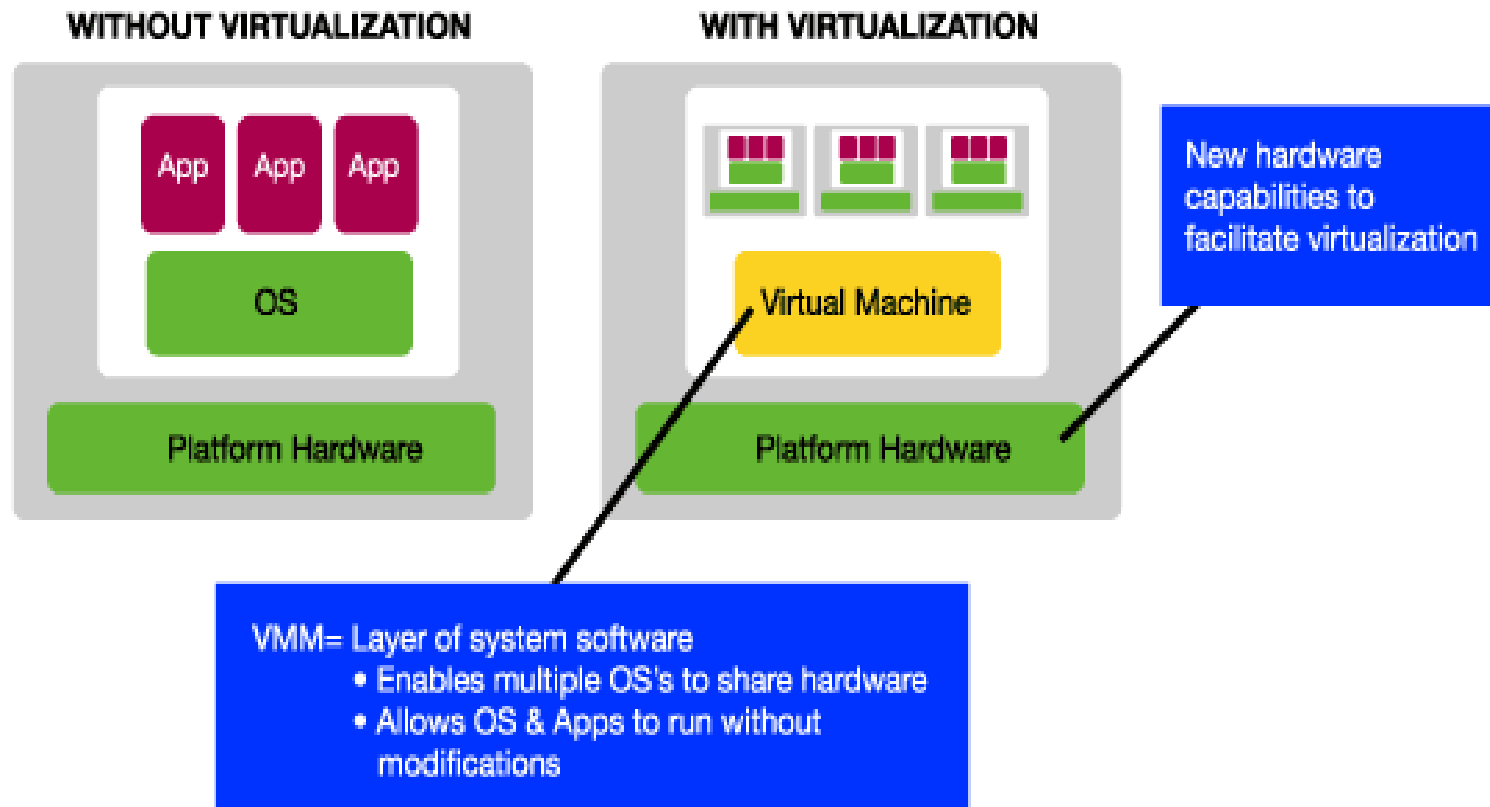
Overview

- ◆ Background on virtualization
- ◆ Virtualization to protect software/platforms from malware
- ◆ Virtualization to detect malware
- ◆ Virtualized malware

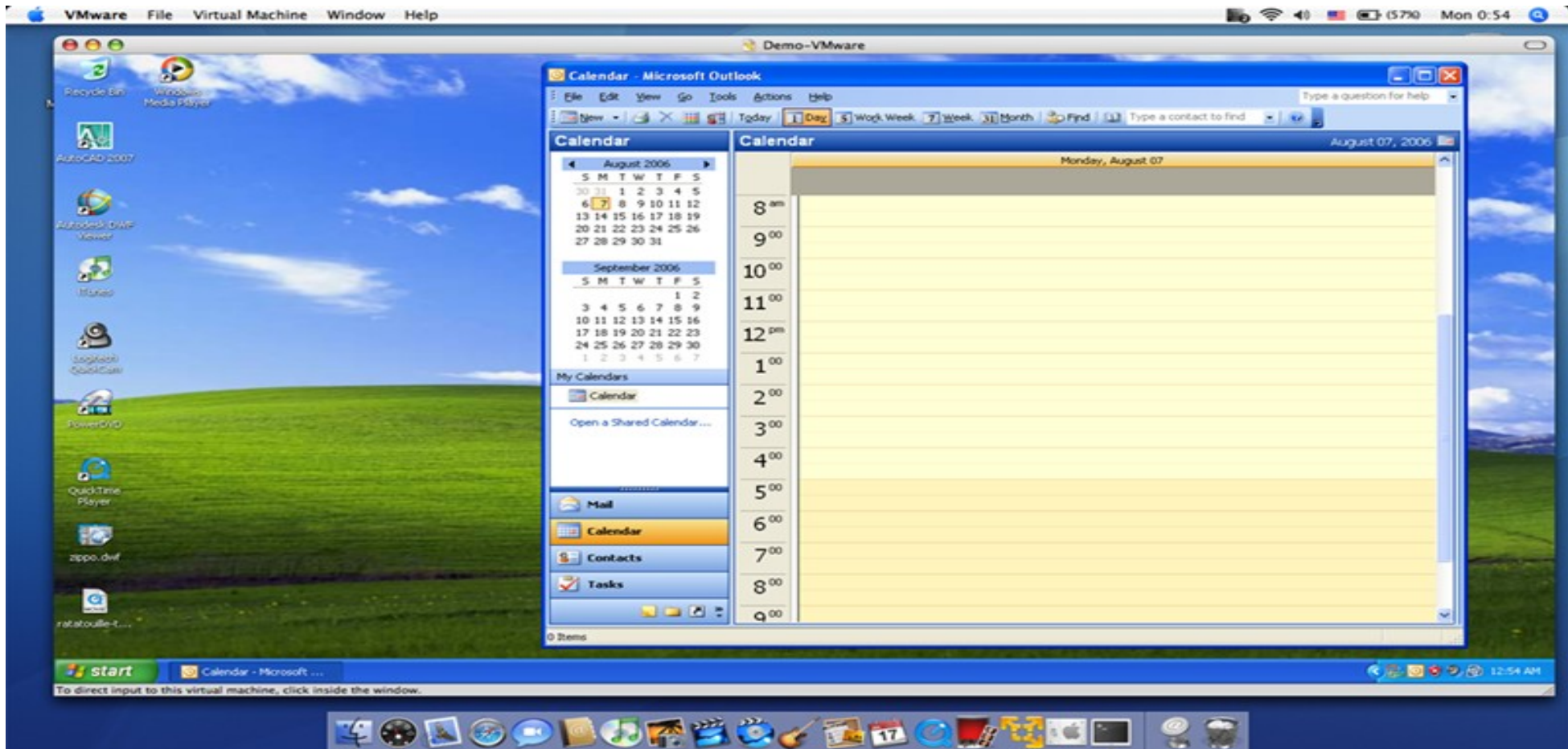


What is virtualization?

- ◆ Abstraction of computational resources



VMWare on OS X



Examples of virtualization

- ◆ Emulation – translate all instructions
 - ◆ QEMU, Bochs
- ◆ Paravirtualization – requires changes to OS
 - ◆ UML, Xen
- ◆ Native virtualization – require same architecture, most instructions executed natively
 - ◆ VMWare
- ◆ OS-Level virtualization – same OS, isolate resources
 - ◆ VServer



Recent Advances in virtualization

- ◆ Faster hardware
- ◆ New hardware support: Intel -VT and AMD-V
- ◆ Uses
 - ◆ Compatibility
 - ◆ Innovation
 - ◆ Hardware cost control
 - ◆ Security



Virtualization and Security

- ◆ Isolate and protect software from malicious programs (malware)
 - ◆ Sandbox programs in own address space
 - ◆ Mediate with lean, secure VMM
- ◆ Detect malware
 - ◆ Use emulation to run malware in a safe environment
 - ◆ Stop, move, replay events in virtual machines
- ◆ Virtualized malware
 - ◆ Rootkits that hide underneath OS



Virtualization for Isolation

- ◆ Presenting virtual interfaces to resources can limit and control the access of malicious programs to resources
- ◆ A Virtual Machine Monitor (VMM) can do more rigorous access control



One Laptop Per Child



- ◆ OLPC project aims to give all children laptops
- ◆ Cannot demand 6 year old children secure their machines
- ◆ Need way to allow them to get hacked and still be ok
- ◆ Answer: virtualization



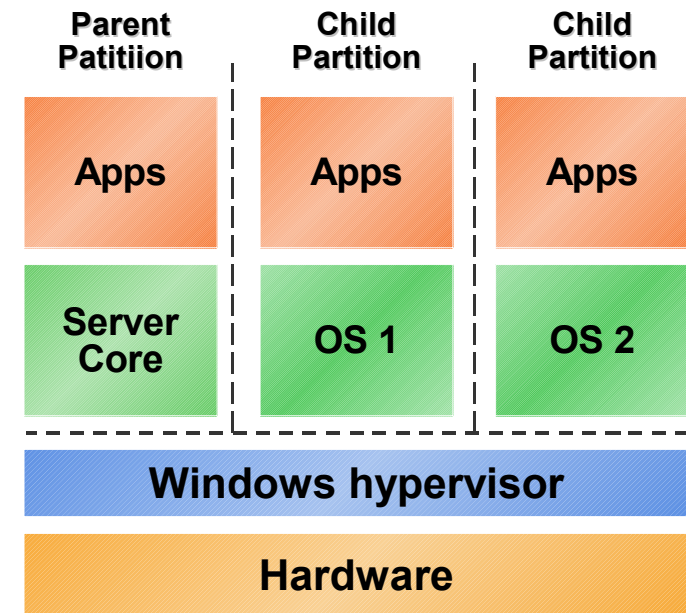
OLPC, VServer and Bitfrost

- ◆ Current applications have no isolation
- ◆ Solitaire can access network, camera, user documents, passwords etc
- ◆ Programs execute in virtual machines (or sandboxes or jails) that can only see its own files/processes
- ◆ Programs must request additional permissions from the VMM



Thin Hypervisors

- Existing OS kernels too big and complex to avoid vulnerability
- Small hypervisor does scheduling, isolation
- Parent partition manages permissions
- Keep hypervisor secure
- Push dangerous apps into virtual machines



Virtualization to Detect Malware

- ◆ Many properties of virtualization useful for detecting malicious software
 - ◆ Tamper resistance
 - ◆ Safety/Isolation
 - ◆ Instrumentation
 - ◆ Rewind/replay properties



Limitations of Using Virtualization for Malware Detection

- ◆ Rice's theorem – nontrivial properties of computer programs cannot be determined automatically
- ◆ Semantic gap – more context inside of a virtual machine than outside of it
- ◆ An operating system sees processes and files versus registers, disk blocks and memory pages outside
- ◆ Instrumentation can help (but possibly corruptible)



Emulation in Anti-virus tools

- ◆ Modern viruses use polymorphism to evade detection by signature
- ◆ Only way to detect this malware is through emulation in an isolated virtual machine
- ◆ Specific emulation techniques can be countered by virus authors (due to limitations on previous slide)
- ◆ Result: security through obscurity works here, want to use a different anti-virus tool than the masses



Running Out-of-the-Box Detection in the Virtual Machine Layer

- ◆ Xuxian Jiang et al to appear ACM CCS 2007
- ◆ Recreate internal state externally in the VMM (bridge the semantic gap)
 - ◆ Reconstruct files/directories from raw virtual disk
 - ◆ Reconstruct process information from virtual memory
- ◆ Compare internal view with reconstructed view to find root kits
- ◆ View reconstruction good enough to run out-of-the-box anti-virus software (Symantec, Kaspersky, McAfee, F-Secure, etc)



Virtual Honeynets

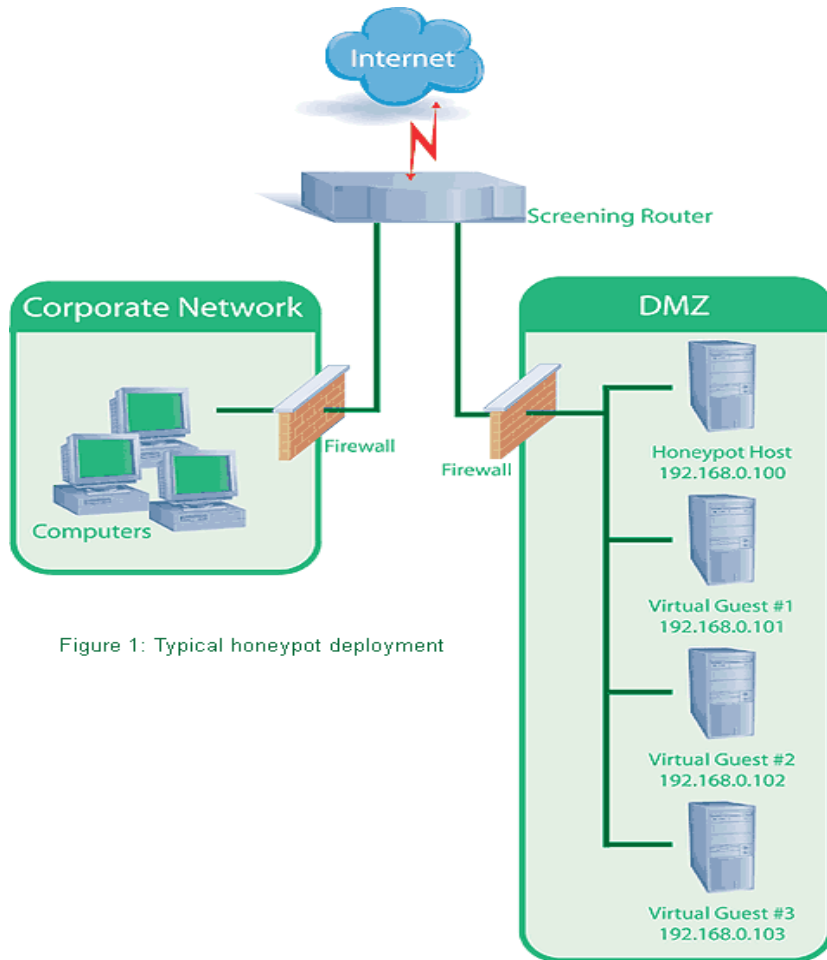


Figure 1: Typical honeypot deployment

- ◆ Honeynets are useful for network security
- ◆ Virtualization allows a whole network to be run on a single computer
- ◆ Cost savings



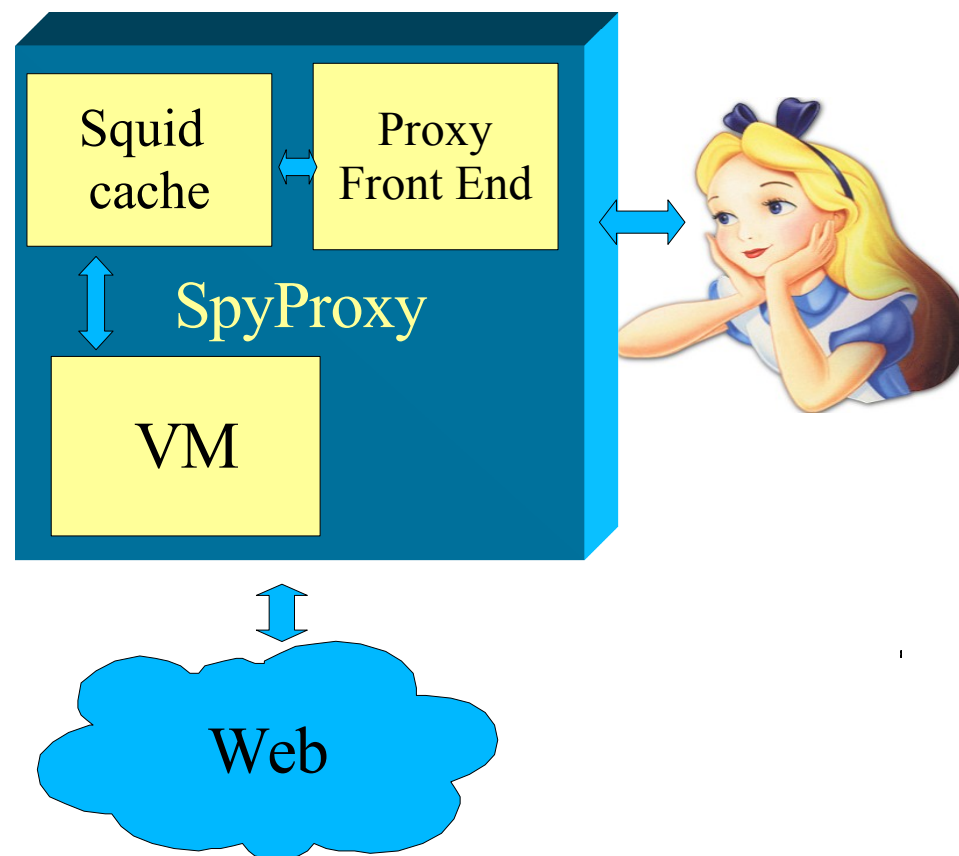
Finding Malicious Websites

- ◆ web 2.0 means websites can infect machines using browser vulnerabilities
- ◆ Compromised hosting facilities mean not just sketchy websites
- ◆ Google used VMs to analyze web-based malware
 - ◆ Run IE in a virtual machine
 - ◆ Record all HTTP fetches, new processes, changes to registry and file systems



SpyProxy (Moshchuk et al)

- ◆ Usenix Security 2007
- ◆ Tool to detect malicious websites
- ◆ Static analysis
- ◆ VM downloads the webpage
- ◆ If safe, send to user
- ◆ Results put in the cache
- ◆ Flaw: nondeterminism



Virtualized Malware

- ◆ Not all added security, new opportunities for malware
- ◆ Virtualized rootkits
- ◆ Side Channel attacks
- ◆ Bugs in VMMs



Take the “Blue Pill”

- ◆ Developed by Joanna Rutkowska
- ◆ VMM as root kit
- ◆ Instead of hiding your control of a compromised system in the kernel, insert thin hypervisor below it
- ◆ Runs at a higher level of privilege, so can still control the machine
- ◆ Hard to detect within the OS



Techniques to Detect Virtualized Malware

- ◆ Very difficult to hide that virtualization is happening
- ◆ Timing issues
- ◆ TLB profiling
- ◆ Defeating these requires a big code base
- ◆ Hard to distinguish “good” VMM from “bad” VMM



Side Channel Attacks

- ◆ How isolated are virtual machines?
- ◆ Can they spy on each other?
- ◆ VMs share the resources of a single host
- ◆ If multiple VMs try to use the same resource they may notice
- ◆ Mitigated by only allowing each machine a specified share of the resource
- ◆ But not completely, and at great efficiency loss



Future Directions

- ◆ Ubiquitous virtualization changes security landscape
- ◆ Effects both good and ill
- ◆ Secure Launch?



Conclusions

- ◆ Virtualization has important security implications
- ◆ “Arms race” between hackers and defenders
- ◆ Lot of interesting work in this space

