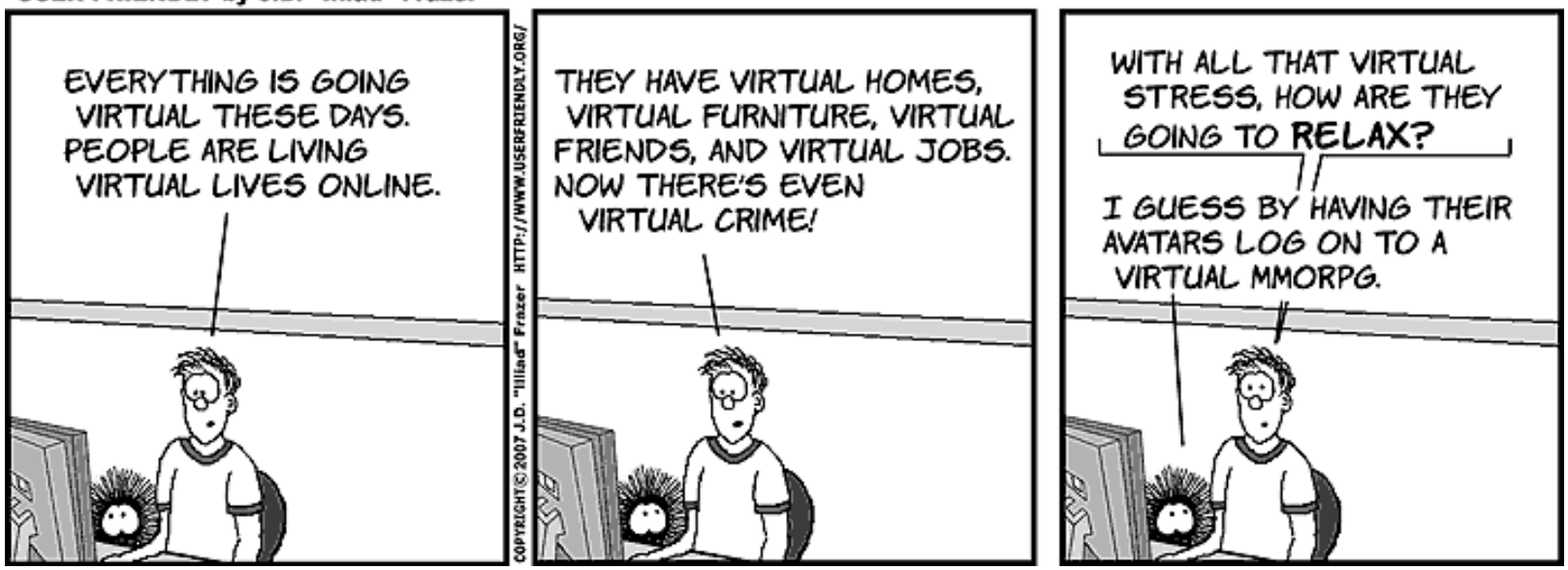

USER FRIENDLY by J.D. "Illiad" Frazer
EVERYTHING IS GOING VIRTUAL THESE DAYS. PEOPLE ARE LIVING VIRTUAL LIVES ONLINE.
THEY HAVE VIRTUAL HOMES, VIRTUAL FURNITURE, VIRTUAL FRIENDS, AND VIRTUAL JOBS. NOW THERE'S EVEN VIRTUAL CRIME!
WITH ALL THAT VIRTUAL STRESS, HOW ARE THEY GOING TO RELAX?
I GUESS BY HAVING THEIR AVATARS LOG ON TO A VIRTUAL MMORPG.
COPYRIGHT© 2007 J.D. "Illiad" Frazer HTTP://WWW.USERFRIENDLY.ORG/

Internet Security Systems

Virtualization Technology
A Manifold Arms Race

Michael H. Warfield
Senior Researcher and Analyst
mhw@linux.vnet.ibm.com

Food for Thought

Is “Virtual Reality” an oxymoron or is it the department of redundancy department and how could you tell?

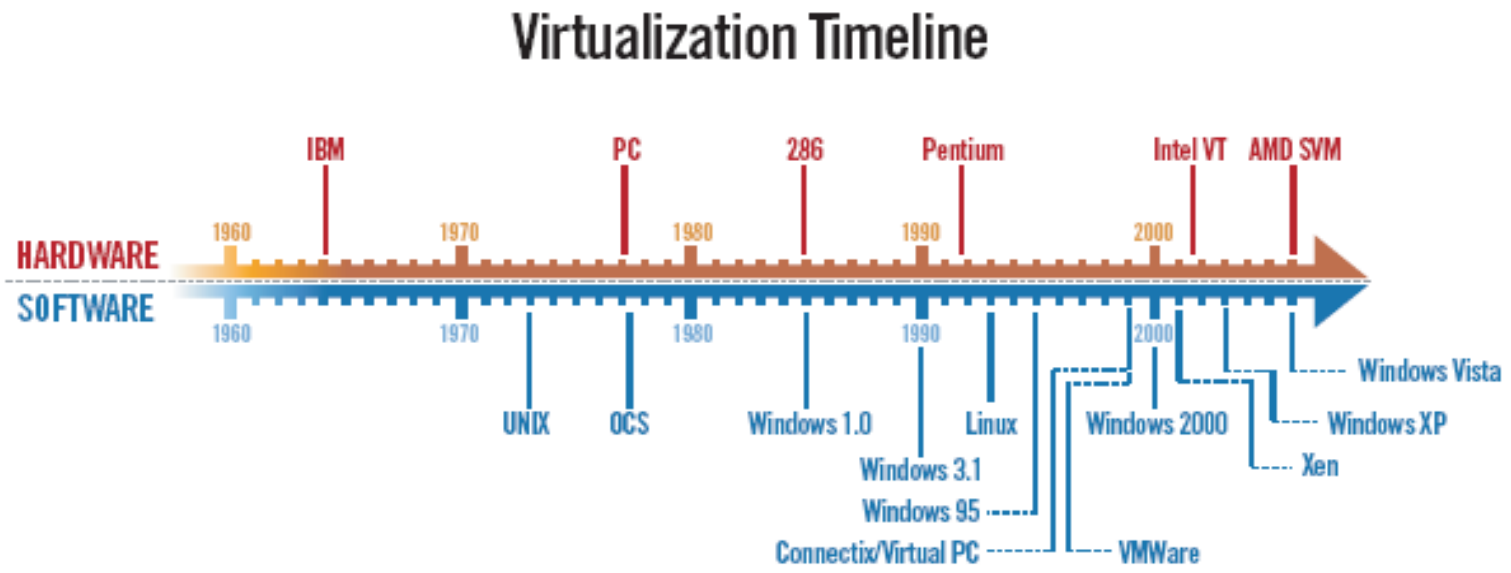
Introduction

- Virtualization has been a hot topic in IT news.
- Frequent topic of discussion in user's groups.
- Products like VMware and Xen are making great advances.
- Hardware advances are improving virtualization capabilities.
- Malware like “blue pill” are worrying researchers.
- Virtualization is much older than all this.
- Virtualization is much broader than all this.
- Virtualization is much more pervasive than all this.

Virtualization

- Virtualization, in computer technology, is the process of grouping and dividing the resources of a computer system into multiple execution environments by applying one or more methodologies or technologies such as hardware or software partitioning, partial or complete machine simulation, and/or emulation.
  - Corollary 1: Virtualization should have no operational impact on applications running under virtualization.
  - Corollary 2: A normally operating application should experience no change to its operation when running under virtualization.
- A virtual machine environment is a virtualization in which an entire system is virtualized.
  - Corollary 1: A Virtual Machine Environment should have no operational impact on the system or applications running under virtualization.
  - Corollary 2: A normally operating system and its applications should experience no change to its operation when running under virtualization.

Virtualization Timeline



Some events in virtualization in virtualization history and their relationship to other computer related events and timeframes.

Classical Features and Attractions

- Hardware consolidation and reduction
  - Demand for electricity is reduced.
  - Fewer systems require less floor space.
  - Fewer systems require less air conditioning and lighting.
  - Even with redundancy factored in, the savings are impressive.
- Hardware task provisioning
- Software compartmentalization
- Creation of “software appliances”
- Snapshots of entire servers
  - Live backups of running servers
  - Diagnostic and analytical rollbacks
- Hot migration of live servers across hardware
- Maintenance of legacy systems as virtual images

Advanced Features and Attractions

- Greatest advantage is in expanded flexibility.
- Number of systems != Number of hardware machines.
- Multiple machines, multiple cores, provide horsepower.
- Dynamically allocate clustering resources.
- Multiple virtual machines provisioned across hardware cluster.
- Fast redundancy and failover.
- Dynamically add virtual systems.
- Dynamically add hardware.
- Remote management systems without dedicated hardware.

Virtualization Techniques

Simple Forms of Virtualization

- Application
  - Configuration of resources so that one application may appear to be more than one instance.
  - Multiple personalities and configurations
- Emulation
  - Performed through emulation of a machine's instructions.
  - Tends to be quite slow and inefficient.
  - Can be extremely versatile.
  - Can virtualize foreign hardware and architectures.
- Virtual Machine Interpreters
  - These interpret their own specific language.
  - Generally have a virtual environment “sandbox”.
- API Virtualization
  - Replaces the API layer with a virtual subsystem library.
  - Applications execute native machine instructions.
  - Replaces foreign API libraries with libraries and interfaces to the host facilities and libraries.

Operating System Virtualization

- Operating System Virtualization does to systems what Application Virtualization does to applications.
- Core system provisions virtual machines and resources through partitioning and/or system configurations.
- Applications run on a common core system or kernel.
- This tends to be highly efficient.
- There is little CPU overhead.
- It does result in some memory overhead.
- Dedicated disk images are not required.
- It cannot run multiple disparate operating systems.
- Examples:
  - Virtuoso, OpenVZ, Linux Vservers
  - Solaris Zones, FreeBSD Jails/Gaols

Paravirtualization

- Paravirtualization is machine virtualization which uses hooks into the virtualized environment to aid in the virtualization.
- Guest systems are modified to be virtualization aware.
- This can run adapted foreign operating systems.
- Virtualized guests cooperate with the virtualizing hypervisor.
- Reasonably efficient use and load optimization.
- Examples:
  - XEN (with or without hardware virtualization)
  - Virtual Iron
  - Virtual Box
  - UML (User Mode Linux)

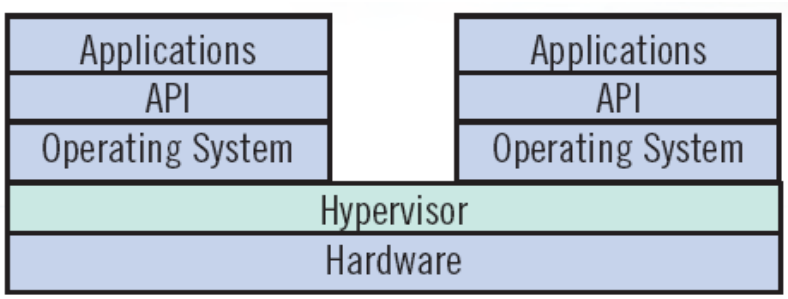
Native Virtualization

- Native virtualization is a step closer to the hardware.
- Guest systems do not need to be modified.
- Specific devices are emulated which guest system recognize.
- Emulated devices need not correspond to host devices.
- Device drivers specific for the virtualization are not required.
- Virtualization tools may be used to facility guest operation.
- Examples:
  - VMware Workstation
  - Virtual PC
  - Parallels (Mac)

Hardware Virtualization

- Hardware Virtualization takes the closest step to the hardware.
- The hypervisor manages the supervisory state and hardware access and contention between the virtual systems.
- The hypervisor can be made much thinner.
  - Guests are managing most of the work.
  - No need for emulated device drivers.
- This can virtualize very difficult foreign systems.
- Examples:
  - VMware ESX Server with HVM / Intel VT / AMD SVM
  - XEN 3 with HVM / Intel VT / AMD SVM

Hardware Virtualized System



A system with hardware virtualization supports running multiple different operating systems sharing the same hardware and managing contention between the guest systems. Each guest should be unaware that they are sharing resources with other guests.

Choosing the Right Virtualization

- Operating System Virtualization is highly efficient.
  - Applications must run on the same kernel.
- Paravirualization can run other operating systems efficiently.
  - Operating systems must be modified or limited.
- Native virtualization can run a wider variety of guests.
  - Native virtualization is more complex and less efficient.
- Hardware virtualization is the most versatile.
  - Hardware virtualization is less efficient and unable to take advantage of device driver and scheduler accelerations.
- Composite virtualization can take advantage of combinations.
  - Each individual method has it's own advantages and disadvantages.

The Arms Races
(None of this has been lost on the bad guys)

The Arms Races

- Virtualization is a powerful, versatile, featureful computing facility.
- This has not been lost on the malware community.
- Malware writers see virtualization as a tool to improve root-kits.
  - Root-kits are tools designed to actively evade detection.
  - There is interest in hypervisor based root-kits.
- They want to evade virtualization used in analyzing malware.
- They want to tune and design attacks against virtualization.
- This has led to a manifold, or multiple, arms race.

Detecting Virtualization

- The first arms race is in detecting virtualization.
- A lot of malware now refuses to run under virtualization.
  - This can generally be evaded by researchers.
- Attackers want to break out of virtualization environments.
  - They want to attack other guest systems.
  - They especially want to attack the host system.
  - Anything on the host system is “dios ex machina” to processes on the guests.
- Many common techniques for detecting VMware and VirtualPC.
  - Device drivers
  - I/O ports
  - Communications channels
- XEN, Virtual Box, and Virtual Iron are not commonly checked for.
- Good guys need to detect malicious hypervisors.
  - Timing irregularities
  - Resource consumption

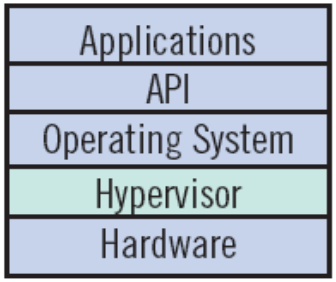
Undetectable Virtualization

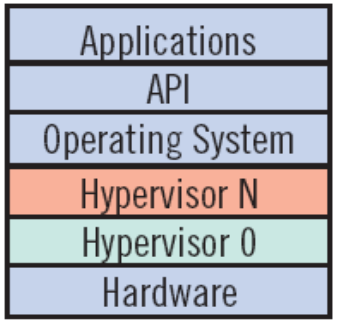
- Flip side of detection is in creating “undetectable” virtualization.
- Legitimate hypervisors strive to have little impact on guests.
  - Legitimate hypervisors have no need to “hide”, per se.
  - Legitimate hypervisors have to be compatible with guest systems.
  - Legitimate hypervisors may strive to be difficult to detect by malware.
- Malicious hypervisors also strive to have little detectable impact.
  - Malicious hypervisors do need to hide.
  - Malicious hypervisors need to be transparent.
- Transparency is not the same as compatibility.
- Transparency should not possible in theory.
  - We're already having trouble detecting what's already there.
- Compatibility is a much easier goal to reach.

The Race for Ring 0

- The first two races together create a pressure for a third race.
- Ring 0 is the supervisory state of the processor, the highest level of privilege on the hardware.
- Ideally, hypervisors can virtualize other hypervisors in a stack.
  - This can be a test for detecting incomplete virtualization.
- Superior hypervisors can easily detect other inferior hypervisors.
  - This can be a detection for virtualization attacks.
- Inferior hypervisors should have a hard time detecting parents.
- The first hypervisor to control ring 0 wins.
- Subsequent hypervisors are virtualized under (inferior to) the hypervisor sitting in ring 0.
- Future systems may have thin hypervisors in boot firmware.

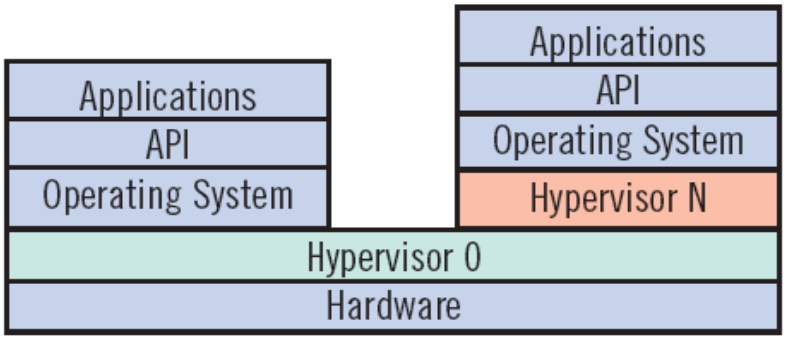
Nested Hypervisors





- *A simple system with a hypervisor and a single guest*
- *A system with nested hypervisors (which one is the bad guy?)*

Complex System



A system with a hypervisor in ring 0 infected with a malicious hypervisor in one virtual machine of many

Defense and Counter Measures

- If ring 0 is the holy grail of the previous race, defense of ring 0 is paramount in the next race.
- Without a preexisting hypervisor, ring 0 must be protected.
- Virtualization provides some prophylactic protection.
- It's much harder to tell a good hypervisor from a bad hypervisor.
- Hypervisors may still be subject to hypervisor specific attacks.
- Hardware virtualization should be disabled where possible when not needed.
- Not all processors and BIOS permit disabling HVM features.

Virtualization and Security

Virtualization Improves Security

- Virtualization helps compartmentalize functionality.
- Functions which use to share a common machine can be isolated.
- Web server compromise doesn't lead to DNS compromise, etc.
- Security services can be isolated from public services.
- Security monitoring can be “out of band” to what's monitored.
- Independent security subsystems can be “bolted on”.

Virtualization Degrades Security

- Merely consolidating systems does nothing to improve security.
- Adding a hypervisor layer and addition OS complicates things.
- Complexity is the enemy of security.
- Hypervisors may have their own vulnerabilities.
- Attacks between virtual machines are possible.
- Networking becomes much more complicated.

Virtualization Complicates Security

- Complex virtual systems have their own internal networking.
- Networking may be bridged or routed or NAT or any combination.
- Internal networks have their own virtual etherswitches or routers.
  - Control and isolation of promiscuous bridge interfaces is very complicated.
- Virtualized networks complicate firewall logic.
- External IDS/IPS can not monitor inter-VM traffic.
- Virtual IDS/IPS may have limited scope.
- Inter-VM traffic and switch logic varies with hot migrations.
- Tracking down phantom virtual systems can be challenging.
- Virtualization complicates issues with TPM and DRM.

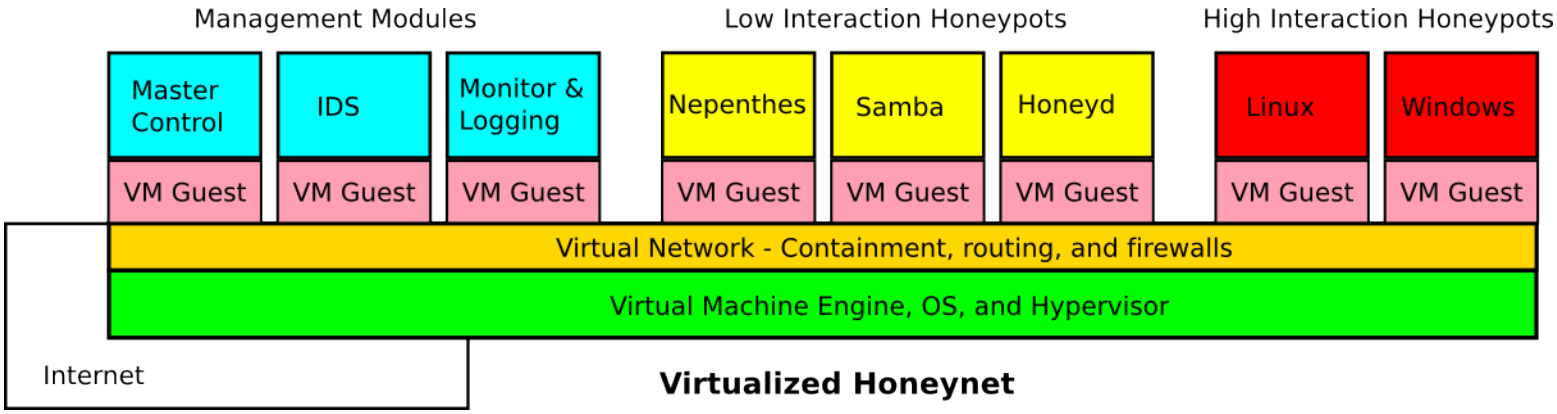
Attacks Against Virtual Environments

- Communications channels
- Shared resources
  - Shared folders
- Device drivers
  - Devices with direct memory access
  - USB and Firewire devices
- Emulated device firmware
  - Network cards
  - Direct bus communications
- Incomplete emulation or hypervisor bugs
- Most require some knowledge of the type of virtualization

Defense Against Attackers

- Disable communications channels when not needed.
  - Disables VMware Tools.
- Disable shared resources.
  - Shared folders may not available.
- Minimize services in the host system.
- Monitor host systems for virtualization probing.
- Change virtualization system defaults.
  - Doesn't protect against attacks.
  - Makes probes easier to detect.

A Defensive Virtualized Honeynet



- Compound system incorporating all elements of a honeynet
- Virtualized low and high interaction honeypots
- Containment firewall and independent routing
- IDS, Master Control, and Logging (and dead man switches)

Joanna's Little Blue Pill

- Blue Pill was written by Joanna Rutkowska.
- It's just one example of a malicious “thin hypervisor”.
- Proof of concept was presented at Black Hat.
- It's memory resident, extremely difficult to detect.
- Exploited a swapped-out device driver on Windows Vista to load.
- It contained no real malicious code in any payloads.
- Is detectable by “timing” tests.
- Newest instantiation claims to be able to virtualize a hypervisor.
- Newer detection tests look for TLB changes and other resources.
- There is an arms race here, between researchers, in and of itself.

Closing

The Future

- Virtualization will continue to become cheaper and cheaper.
- Virtualization has been introduced into the Linux Kernel (lkvm).
- Virtualization API's in the kernel have been released.
- More vendors are offering more products.
- More projects are maturing rapidly.
- Hardware vendors are promising better virtualization features.
- System vendors are working to deliver thin hypervisors in firmware.
- The future is very bright for virtualization.
- That light could be the headlight of an on-coming train.

Virtualization Technology
A Manifold Arms Race

Thank You!

Michael H. Warfield
Senior Researcher and Analyst
mhw@linux.vnet.ibm.com