Hypervisor vs. Host Based Security

Chris Brenton

Blog: www.cloudpassage.com

1

Full Disclosure

- I am the principal security architect for a cloud security product
- The product is a host based solution
- My opinions drove my choice of employment, not the other way around

What We'll Cover

- Virtualization basics
- Hypervisor security
- Host based security
- Which to choose?

Leveraging Virtualization In The Cloud

- Cloud deployments are all about pooling resources to increase efficiency
  - Reduce cost
  - Decrease administration overhead
- This makes virtualization a natural platform for building clouds
  - Hardware abstraction provides foundation
  - Tear down and setup of software far quicker and efficient than hardware

An overwhelming majority of IaaS clouds leverage virtualization for their foundation.

What is Introspection?

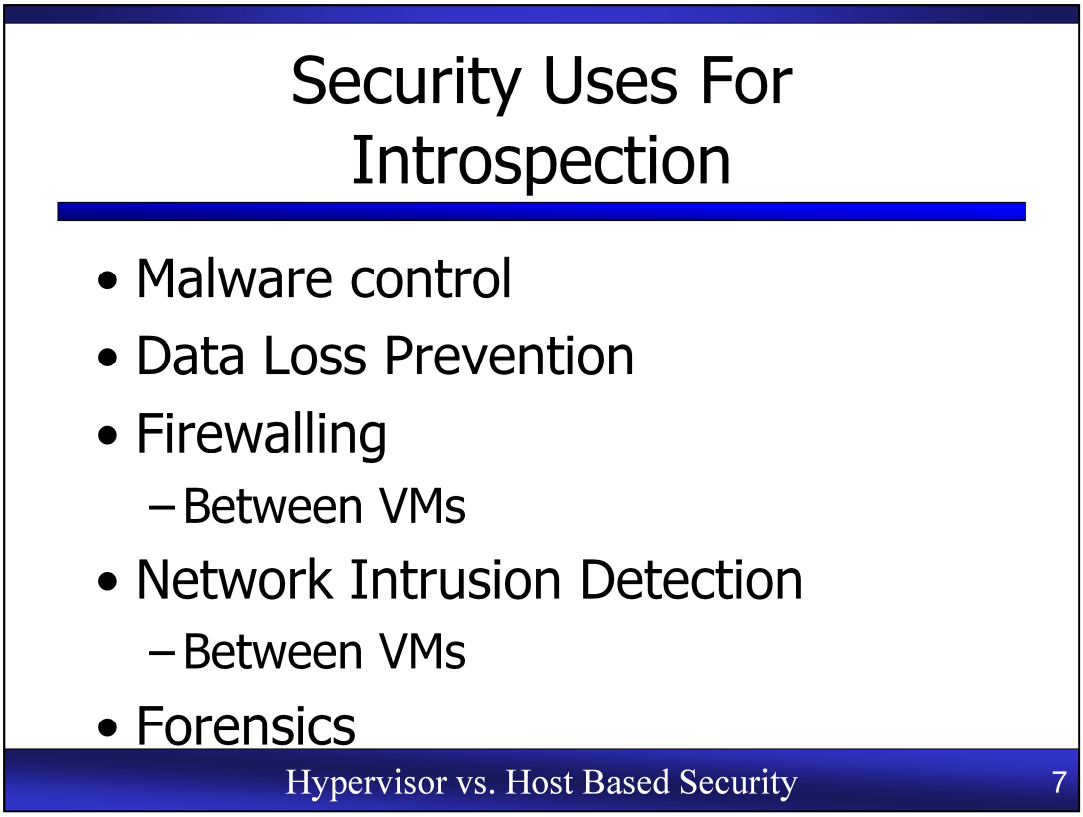
- Expansion of the hypervisor's capabilities
  - Beyond monitoring compute and store calls
- Introspection permits you to monitor:
  - Memory and program execution
  - Access to data files within storage
  - Network traffic
- Conduit for deeper VM analysis
  - Can be leveraged to implement security

With the use of virtualization comes the use of a hypervisor. Normally, the hypervisor simply provisions resources as needed by each VM. Introspection expands that capability permitting you to monitor program execution, file storage and network traffic. This permits the hypervisor to be leverage as a security tool for monitoring VM activity.

Why Introspection is Such a Powerful Tool

- Hypervisor runs with higher perms than VM
- This means within the VM, you cannot hide from the hypervisor
- Kernel Level Rootkit
  - Normally can stealth from security tools
  - KLR can't stealth files from hypervisor
  - Since introspection can validate program execution, no place to hide

Let's look at a practical use for introspection. One of the biggest security issues we have in the industry today is kernel level rootkits. This is because a kernel level rootkit effectively turns the core operating system into malware. The result is the rootkit has the highest level of system permissions, and can leverage these to hide itself from detection. With introspection, the VM is run at a lower level of permissions than the hypervisor. This means that if a rootkit infects a VM's kernel, it will still have lower permissions than the hypervisor, and thus will be unable to hide from it. This dramatically improves our ability to detect the presence of the rootkit.



Introspection's capability can be leveraged for a wide range of security applications.

Introspection Performance Benefits

- Permits consolidation of resources
- Anti-virus good example
  - Running multiple host agents can cause "AV Storms"
  - Hypervisor based AV can reduce overall processing
  - Single appliance for multiple VMs

Let's look at a usage case for introspection. One of the most problematic pieces of software in an IaaS environment is anti-virus software. This is because AV software can be extremely CPU and disk intensive when checking file storage for malware. If multiple VMs initiate a full disk scan at the same time, an IaaS cloud can become extremely unresponsive. In fact this situation is frequently referred to as "an AV storm", and it's impact can increase costs as well as reduce the number of VMs which can be supported on a given hardware platform.

This problem has caused many AV vendors to retool their security solution to better fit cloud deployments. For example cloud friendly products setup a single AV instance for an entire IaaS cloud, and check storage on all VMs via introspection. Further, results from one VM scan can be leveraged to expedite the checking of others. If "myfile.com" has been found to be virus free on one VM, and the file exists on multiple other VMs as well, full malware scanning is skipped provided the additional copies of myfile.com have the same hash signature. This retooling results in the same level of malware protection but with a dramatic decrease in CPU utilization and disk activity.

Who Supports Introspection?

- Vmware
  - Originally via vSafe APIs
  - Now via vShield
    - Want API access, buy our firewall!
- Xen Introspection project
  - Working to provide similar to Xen

http://wiki.xensource.com/xenwiki/Project_Information

What's The Downside?

- Secure code = small & efficient code
- Introspection causes hypervisor bloat
  - Introduces more line of code
  - Which can contain bugs or insecurities
- Let's third parties write device drivers
  - These become part of the hypervisor code!
  - Introducing more insecurity possibilities
  - SDL process becomes more difficult

It can be argued that along with security benefits, introspection also introduces a number of security issues that can elevate risk against an environment. The first is hypervisor bloat. Much of hypervisor security is predicated on keeping the code as small as possible. The fewer the lines of code, the less likely an attacker will find a problem which can be leverage for malicious gain. By adding introspection capability to the hypervisor we increase the amount of code being processed. If we are supporting third party software, we are also increasing the number of programmers submitting code, which can also increase the likelihood of insecure code being introduced.

Increased Attack Surface

- Introspection increases interaction with VM
- This increases interaction with a potentially suspect source
  - Thus increasing the opportunities for compromise
- Removes much of the "quarantine" between VM and hypervisor
  - Creates new potential conduits for compromising the hypervisor
- If the hypervisor becomes 0wn3d, all is lost

Introspection also increases the attack surface of the hypervisor. Any time we permit our code to interact with an untrusted source, we elevate the risk of being attacked. The more interaction being permitted, the greater the likelihood that an attacker will find a problem that can be exploited.

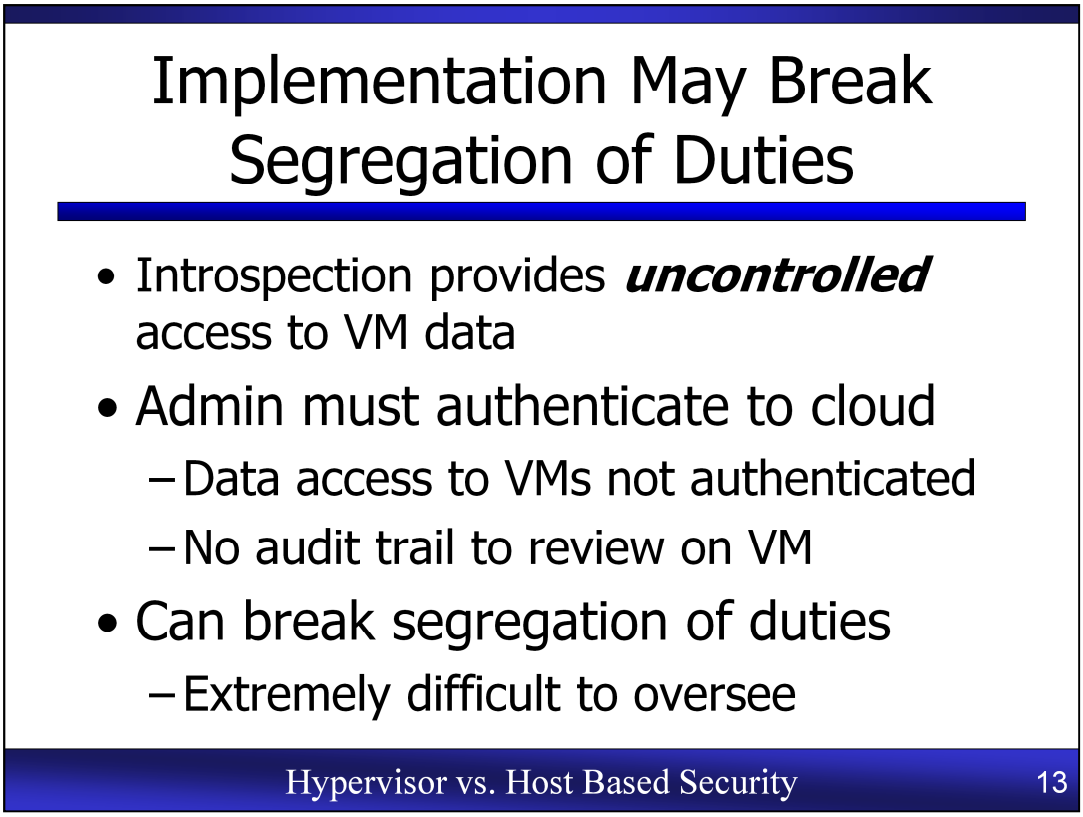
For example, network based intrusion systems are leveraged to check passing packets for potential attack patterns. They don't actually run the code, they simply process packets and pattern match against them in order to look for pre-defined malicious patterns. Over the years we have seen attackers create a number of exploits that specifically target network based intrusion detection systems. These attacks have ranged from simple denial of service (crashing the intrusion system) to providing the attacker with high level access.

These attacks are relevant here in that the intrusion system, like introspection within the hypervisor, is not processing the code but simply checking it for know security issues. The difference is that an attack against a network based intrusion system impacts that single host. A similar attack against a hypervisor could crash the entire IaaS infrastructure or provide a high level of access to all VMs.

Secures The Infrastructure Not The Data

- Only Secures VMs on that hypervisor
  - If VM migrates, security is gone
  - Problematic is you wish to burst to public
  - Insufficient for hybrid environments
- Can result in vendor lock-in
  - Swap hypervisors and scrap all current introspection security solutions

Many have become concerned with the loss of flexibility experienced when an introspection based solution is deployed. If you wish to migrate the VM to another virtualization infrastructure, any risk mitigation provided by introspection may be lost. For example let's say I run an internal IaaS cloud using software provided by "vendor A", and wish to move that VM to another virtualization infrastructure created by "vendor B". If all of my security is introspection based, all of that risk mitigation is lost during the migration. This can lead to vendor lock-in, limitations in infrastructure options., or possibly the need to run multiple security systems in parallel.



Another concern is that introspection can potentially break segregation of duties. In many environments, the amount of access granted to IT staff members is limited. For example the network administrator may have high level access to the routers and switches, but have limited to no access to the servers. System administrators may have high level access to production servers, but no access to the logging servers which record their activity. Segregation of duties ensures that all activity can be monitored and recorded and that no specific job title gets too powerful.

Remember that introspection has full visibility into each VM. It can see all network activity, files stored on disk and applications executing in memory. This means that any administrator with access to introspection effectively has full access to each server running beneath it. So you may not have intended to grant the virtualization administrator full access to the accounting server, but effectively that is the end result. What's worse is that the VM is unable to log this activity. So if introspection is leveraged to access a critical financial file, the VM hosting the file will not record any authentication or file access activity. This could possibly be logged via the introspection tool, but that's up to the plug-in vendor to include as a feature.

Not Fit For Public Consumption?

- Do you really want your provider to have uncontrolled access to all of your data?
- Do you really want other tenants having device driver access to your hypervisor?
- Would make public cloud providers a prime attack target
  - Without introspection, providers are not a conduit to your data
  - With introspection, they have access

For some environments, the segregation of duties issues covered in the last slide may be considered manageable or a non-issue within a private environment. However what if your VM is being hosted on a public provider?

Host Based Security

- Each VM to be protected runs a software agent
- Agent responsible for implementing security
- Needs a central point of management

With a host based security solution, each VM a piece of software that is responsible for protecting the host itself. The classic example is anti-virus software. You run the AV software in the background and it attempts to ensure that malware is not permitted to execute on the system.

Security Uses For Host Based

- Host firewall management
- Patch management
- Vulnerability assessment
- Account management
- Audit and integrity testing
- Log collection

Wait, Don't We Already Have Host Based Security?

- Gen2 host based hogs the CPU
  - Assumes free CPU within VM is "free"
  - AV Storm example
- This can increase costs in a cloud environment
  - Increases private cloud hardware costs
  - Increase public cloud billing charges

Host based security is not a new solution, we have leveraged it for years to protect our systems. However legacy host based solutions do not scale well into the cloud. This is because in the legacy server model we ran one operating system per hardware platform. The host based security solution would monitor CPU utilization and assume that any free CPU cycles could be consumed in the interest of implementing security. The problem with this model in cloud computing is that there may be other VMs that need to use that CPU time. So as we've shifted towards an IaaS cloud model, our legacy host based security solutions have driven up costs.

Gen3 Host Based Security

- Agent needs to be:
  - Small footprint
  - CPU friendly
  - Self verifying
- A SaaS backend can meet these requirements
  - Perform a majority of processing on the backend
  - Force agent to regularly validate its integrity
- A proper implementation should minimize CPU utilization on the protected host

The answer of course is to retool our host based security solutions to be more cloud aware and resource friendly. For example rather than just assuming all free CPU time can be used as needed, processing needs to be throttled. Since we are talking cloud, it may also be possible to offload much of the work to another cloud for processing. For example legacy anti-virus does all the processing required on the host which is being protected. In a cloud model, we could hash the file to be checked and then process that hash on a different cloud to see if it is known malware. The result is that the processing and storage required to check the hash can be done by an outside system, thus reducing the costs associated with running the host being protected. There are already many security companies that are supporting cloud by adopting similar models.

Does An Agent Increase The Attack Surface?

- Maybe…it depends…if:
- Requires an inbound listening socket
  - Creates an additional attack point
  - Attackers could leverage for evil
- Calls home like a rootkit
  - Leverages a reverse connection
  - Can be run without requiring inbound session establishment

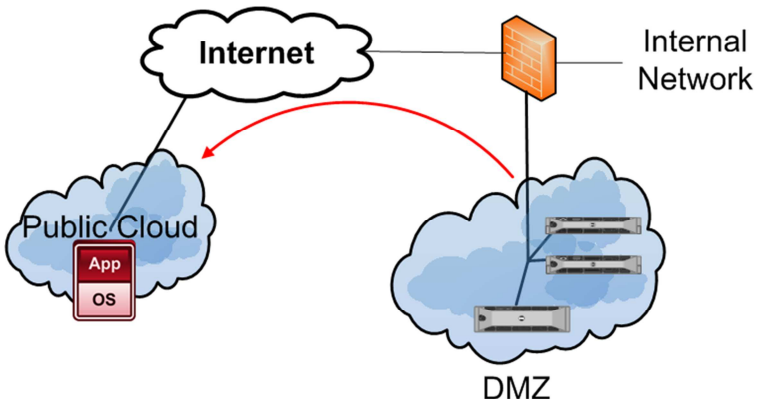
We've already discussed that implementing introspection can increase the attack surface against the hypervisor. Can the same be said for host based security agents? Unfortunately here the answer is a bit more vague as it depends on how the security agent is designed and deployed. For example, if the agent is designed to hold open a listen port in order to accept command and controls from a central management system, then yes we have increased risk to the system by opening up another port of entry into the system, as well as exposing additional code to access via the wire. If however the agent leverages a reverse socket connection to call home at specified intervals without the need to open a dedicated listening port, then obviously the agent is not exposing additional code to access from the wire. We then need to validate the integrity of the communication channel (Is it vulnerable to DNS spoofing attacks? Are all communications encrypted? Etc.).

Focuses On The VM, Not The Infrastructure

- Agent based tied to each individual VM
- Can provide security regardless of location
  - Public, private, mobile between the two
- Migrating VM does not change the level of risk mitigation
  - Unlike introspection

One of the nice things about agent based security is that it focuses on securing the VM, not the infrastructure. This means that if the VM is migrated from one IaaS could to another, security moves with the VM. So moving a VM from a private IaaS cloud to a public provider does not mean that you have lost all of your risk mitigation. A well designed agent should be able to protect the VM regardless of where it is located. Further, this opens the possibility of managing VMs located in both public and private space with the same security tool.

What about Security?



When a VM is migrated, host based security travels with the VM, while hypervisor based security gets left behind

Segregation of Duties

- Dependant on implementation
  - Will the agent generate an audit trail?
  - Does the agent have access to data?
- Can provide security without granting full system access
  - This can augment segregation of duties
  - Again, depends on implementation

Because agent based security executes within the VM itself, classic permissions and audit trail processes can be leveraged to identify what the agent does on the system. This is in contrast to the introspection model where the VM has no ability to log or control what actions are performed.

Support In Public Space

- No provider software interaction needed
  - Like the hypervisor or management API
  - Touches your VM only
- This let's you support any provider
  - Immune to vendor lock in
  - Migrate VMs as required

Another benefit of agent based security is that it can have the ability to work seamlessly with any virtualization environment or cloud infrastructure. This becomes important as VMs become more mobile. You may wish to change where a VM gets executed based on cost or proximity to clients. If you are leveraging introspection, then your choices will be limited to the virtualization deployments that are supported by your introspection solution. If the security is agent based, the virtualization platform becomes ambiguous, so you are free to use other metrics besides virtualization platform vendor when deciding on the best location to execute a particular VM.

Hybrid support

- Host based can typically secure all systems, regardless of location or model
  - Public IaaS
  - Private IaaS and PaaS
  - Stand alone servers
- This can provide a single pane of glass for security management
  - Hypervisor security requires different tools

One of the benefits of a host based security model is that it can usually be extended back into legacy servers. Most environments still run a mixture of VMs and servers on dedicated hardware. If I've implemented a hypervisor based security solution, it obviously will not work on my dedicated servers. Agent based solutions however can usually support both deployment models, thus reducing the number of security tools required to secure an environment.

Which To Choose?

- Perhaps a bit of both…
- Remember that Gen3 decouples the network from the workload
  - Implement security accordingly

When formulating a security plan, remember that cloud decouples the network from the server. In other words, you can no longer limited to running a server within a specific rack on a specified network segment. Servers are free to move to the most ideal location for execution. If you think about it, this is similar to what we went through with workstations. Originally people worked with desktops that were limited to being used on the employee's desk. Then laptops came along and the employee could now work from what ever physical location made the most sense (home, client site, hotel, etc.). Servers are now going through a similar transition.

Implement Introspection

- When focus is protecting the network
- In private IaaS environments
- When risks to the hypervisor itself can be minimized
- Were segregation of duties is not an issue
  - Or the risks can be mitigate

Implement Host Based

- When focus is protecting the data
- When working in public space
- When VMs may be migrated
- To avoid vendor lock
- When a single pane of glass is needed for public, private and stand alone