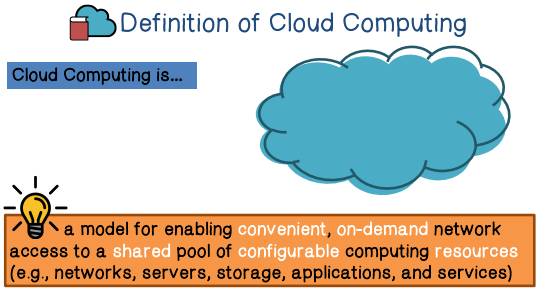
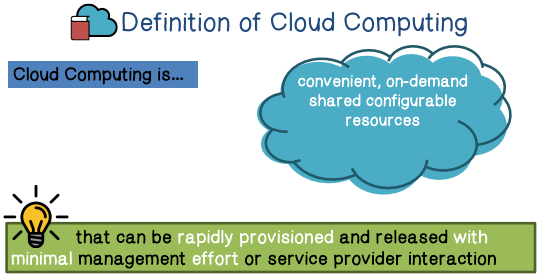
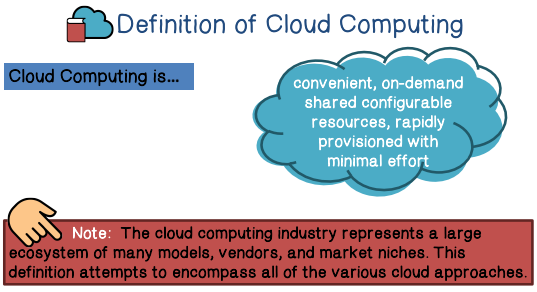
In this lesson, we will discuss cloud security. First, we define what the cloud is. Then, we will discuss the unique security issues associated with it. We will then discuss virtual machine monitoring, because virtualization is the most important core technology of cloud computing.



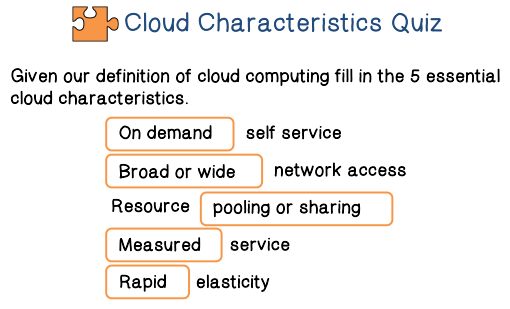
First, let us go over the definition of cloud computing. Cloud computing is a model for enabling convenient, on-demand network access to a shared pool of configurable computing resources. For example, networks, servers, storage, applications, and services.



What this definition says is that, cloud computing uses resources that can be rapidly provisioned and require low management overhead.

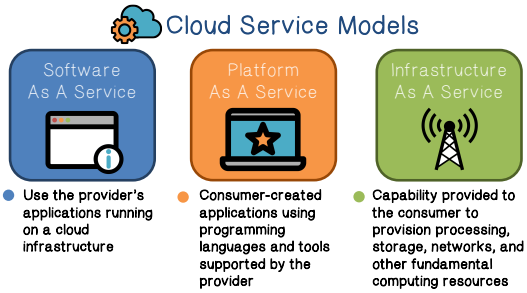


Please note that the cloud computing industry represents a large ecosystem of many models, vendors, and markets. And therefore, our definition here is very general.

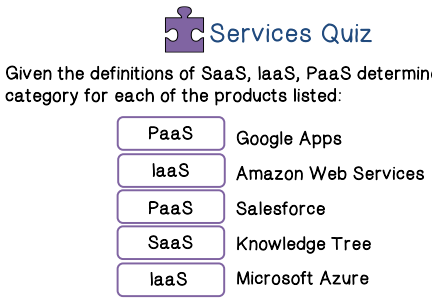


Now that we have a general definition of cloud computing, let us do a quiz on the characteristics of cloud computing. Given out definition of cloud computing, write the five essential cloud computing characteristics.

The first is on demand self-service. The second is broad or wide network access. The third is resource pooling or sharing. The fourth is measured service. The fifth is rapid elasticity.

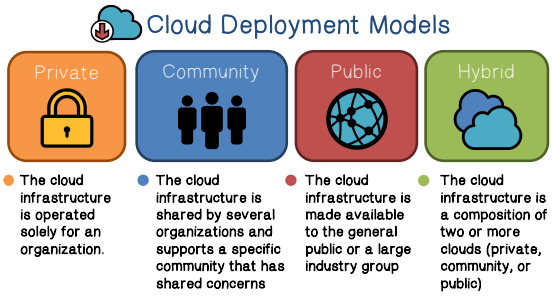


There are three cloud service models. The first one is Software as a Service. It enables a customer to use a provider's applications running on a cloud infrastructure. This applications on a cloud can be accessed from various kind devices through a thin client interface such as the web browser. The second model is Platform as a Service. It enables a consumer to deploy on to the cloud infrastructure consumer created applications using programming languages and tools supported by the cloud provider. The third model is Infrastructure as a Service. It enables a consumer to provision processing storage networks and other fundamental computing resources on the cloud where the consumer is able to deploy and run arbitrary software, which can include operating systems and applications.



Now let us do a quiz on the cloud computing service models. Given the definition of these models, determine the service category for each of the products listed below.

Google Apps is platform as a service. Amazon Web Services is infrastructure as a service. Salesforce is platform as a service. Knowledge Tree is software as a service. Microsoft Azure is infrastructure as a service.

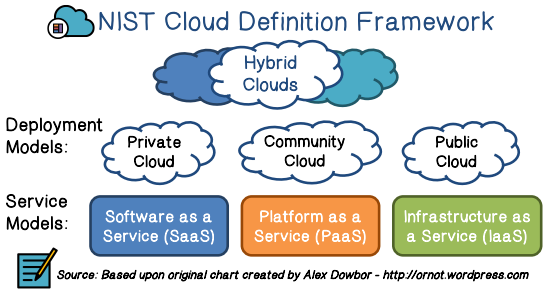


There are several cloud deployment models. The first one is private cloud. Here, the cloud infrastructure is operated solely for an organization. It may be managed by the organization or a third party and may exist on-premise or off-premise. Here, the cloud infrastructure is shared by several organizations and supports a specific community that has shared concerns. It may be managed by the organizations or a third party and may exist on-premise or off-premise. The third model is public cloud. Here, the cloud infrastructure is made

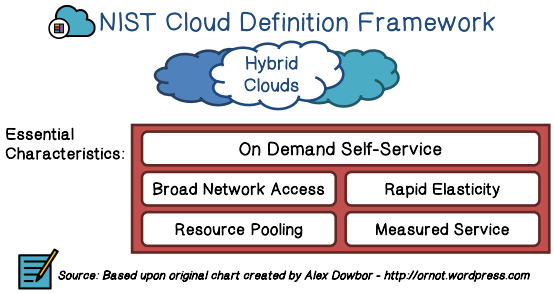
available to the general public or a large industry group and is owned by an organization selling the cloud services. The fourth one is hybrid, is a composition of two or more of these models.

Now let us do a quiz on the common characteristics of cloud computing. Please list some of the characteristics that are shared by all four cloud models.

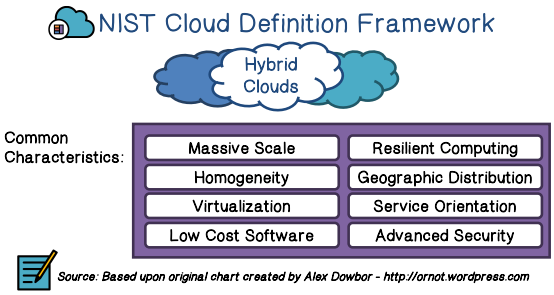
These characteristics are massive scale, homogeneity, virtualization, resilient computing, low cost software, geographic distribution, service orientation, and advanced security technologies.



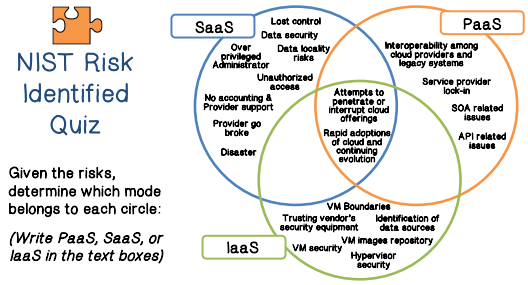
To summarize what we have discussed thus far. Let us go over the NIST cloud definition framework, there are several deployment models and each can use one of these service models. Again, the deployment models include Hybrid, Private, Community and Public and the service models include software as a service, platform as a service and infrastructure as a service.



All Cloud environments share some essential characteristics. These include On Demand Self-Service, Broad Network Access, Resource Pooling, Rapid Elasticity, and Measured Service.

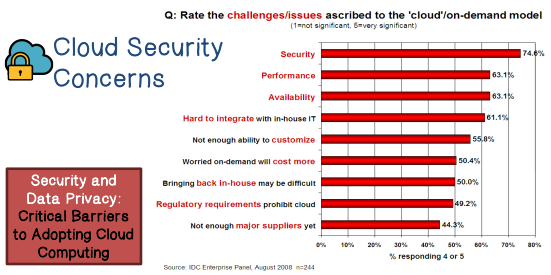


And they also share some common characteristics. This includes Massive Scale, Homogeneity, Virtualization, Low Cost Software, Resilient Computing, Geographic Distribution, Service Orientation, and Advanced Security.

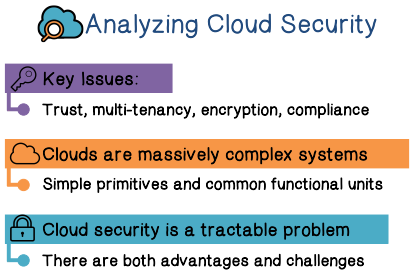


Now let us do a quiz on the risks associated with cloud computing. Given the risks listed in each circle, please identify the service model associated with each circle.

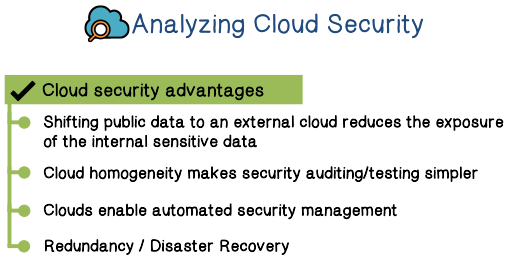
The blue circle includes risks such as data security, data locality risks and unauthorized access. And this circle is associated with the model software as a service. In the orange circle, we include SOA related issues, API related issues and so on. And this circle is associated with platform as a service model. In the green circle, we include risks such as virtual machine security, hypervisor security, and so on. And this circle is associated with infrastructure as a service model.



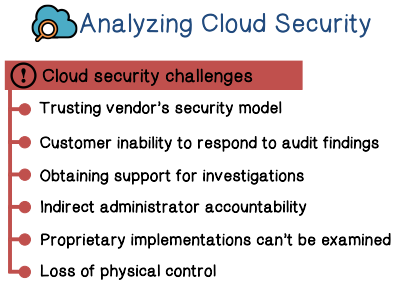
Now let us discuss cloud security. The first question is, is security important to cloud computing? According to this survey, security is the main concern when people consider moving to the cloud. So, it is obvious that security is one of the most important concerns with cloud computing.



There are several key security issues. For example, how do we trust the cloud computing environment because we do not manage it. Furthermore, multiple different organizations may be using the same cloud computing environment. And in terms of data protection, obviously, we should use encryption to protect confidentiality. But we also need to consider various compliance issues. These security issues become very challenging because the clouds are massively complex systems. Although the primitives and common functions are simple, the complexity comes from the fact that these primitives and function units are replicated thousands of times. On the other hand, as we will discuss shortly, cloud security is a tractable problem because there are both advantages and challenges associated with cloud computing.

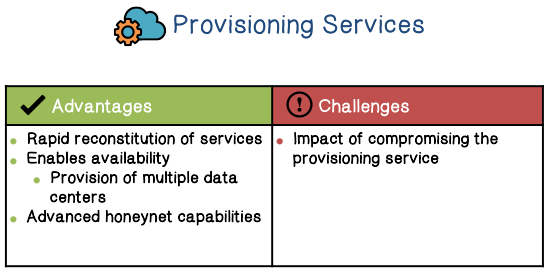


Let us first discuss the advantages. First, you can put public data away from the internal network to the cloud. And it is easy to manage testing and security patches when most systems in a cloud computing environment use the same software. And recovery is easy and fast because it is easy to set up the same system.

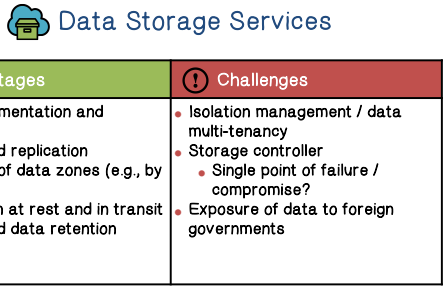


But there are also challenges. First, you have to trust the cloud providers. And you cannot respond to all the findings, or investigate directly, because you have to rely on the cloud providers who are the direct admins, and some software running in the cloud is proprietary. And finally, there is no physical control of the computing resources that you use.

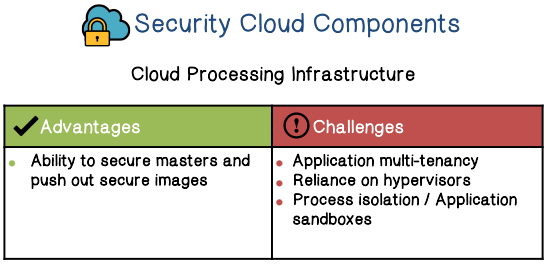
All the core components of a cloud computing environment are relevant to security. This include cloud provisioning services, cloud data storage services, cloud processing infrastructure, cloud support services, cloud network and perimeter security and elastic elements: Storage, processing and virtual networks.



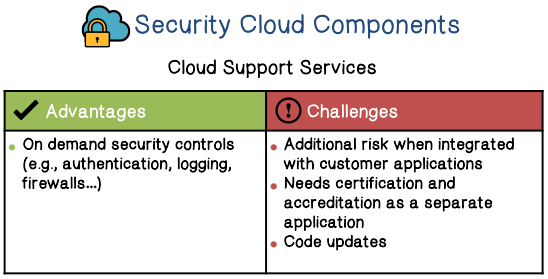
Now, let us discuss the security advantages and challenges for each of these core components. The first is provisioning services. There are several security advantages. This include, rapid reconstitution of services. Greater data availability because we can provision multiple data centers and advanced honeynet capabilities. The main security challenge is that, if the provisioning service is compromised, the impact is disruptive.



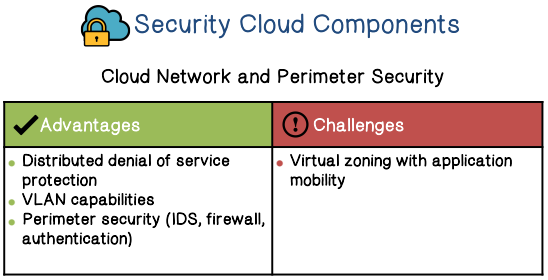
For data storage services, there are several security advantages. This includes data can be fragmented and dispersed, therefore, data can be more resilient to attacks. In addition, data can be encrypted at rest and in transit. There is also automatic backup and replication of data. But there are also several security challenges. For example, data from multiple organizations may be put in the same storage server. And the storage server may be in a different country.



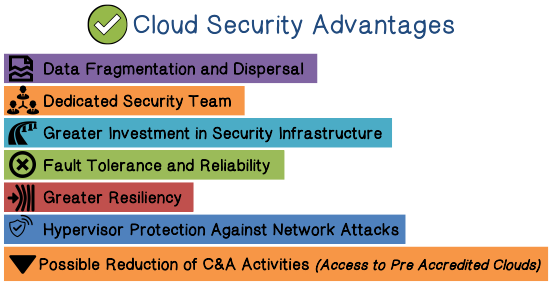
For the cloud processing infrastructure, the main security advantage is that we can secure the master copy of a program and then replicate that copy throughout the cloud computing environment. But then, there is also several security challenges, for example, multiple applications can be running on the same physical machine. And therefore, achieving a real isolation is not easy. For example, cash and memory access are side channels that can leak information. And of course, we should assume that the hypervisors or the by virtual monitors are secure.



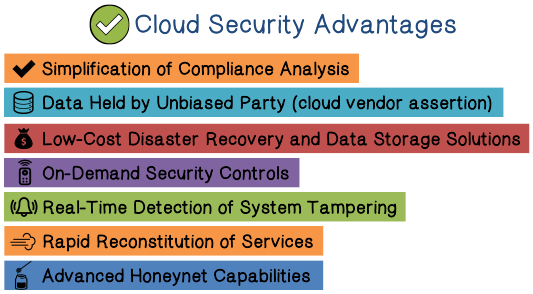
Cloud Support Services can provide on demand security controls for customer applications. But now, the cloud providers need to make sure that customer applications will not cause security problems in the cloud environment.



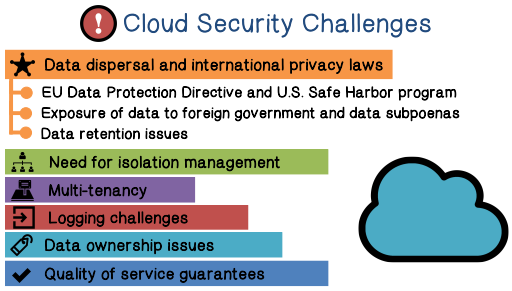
In terms of networks security, since the computing systems are now distributed, denial of service attacks are now harder to succeed. Furthermore, standard perimeter security measures such as IDS and firewalls are deployed by default. But since different applications share the same cloud, it is challenging to create security zones where the resources share a common security exposure or security risk.



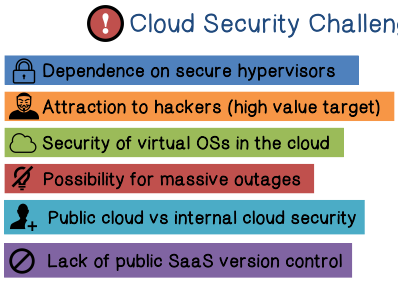
Let us summarize the security advantages a cloud computing environment can provide. These include Data Fragmentation and Dispersal, Dedicated Security Team, Greater Investment in Security Infrastructure, Fault Tolerance and Reliability, Greater Resiliency, Hypervisor Protection Against Network Attacks and the Possible Reduction of C&A Activities.



Furthermore, the compliance analysis is simplified, and data can be held by unbiased party. There is Low-Cost associated with Disaster Recovery and Data Storage. There are on demand security controls and real time detection of system tampering. Reconstitution of services is easy and vast. And an advanced Honeynet can be deployed.



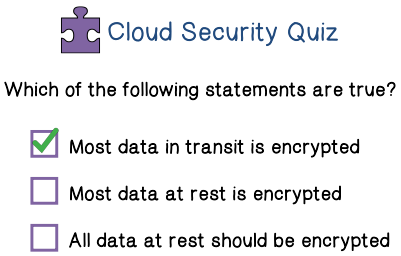
Let us also summarize the security challenges. First, we have to consider various international privacy laws, for example, the EU Data Protection Directive and the U.S. Safe harbor program. And data can be subject to subpoenas from governments, including a foreign government. And since multiple organizations may be using the same cloud computing environment, we need very strong isolation protection. And since computation and data can be distributed, logging becomes a challenge. Similarly, since the cloud is a distributed computing environment it is challenging to provide consistent quality of service. And finally, since the customer does not own the computing environment there could be potential data ownership issues. For example, who owns the one-time data produced by a customer application?



There are additional challenges. These include we have to depend on a secure hypervisor, the cloud is an attractive target to attackers, we still need the security of the operating systems running in the virtual machines, and the potential impact of a successful attack on the cloud can be massive.

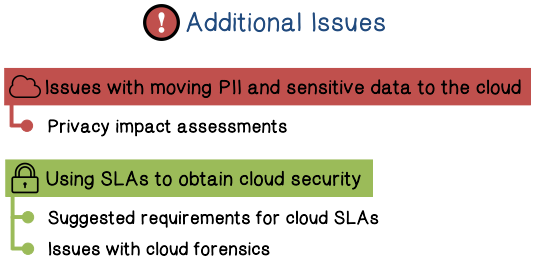
And since some organizations now start to use public cloud, it is challenging to reconcile the security policies of the internal network versus the public cloud. And for public cloud that uses software as a service model, a customer cannot control what software or what version to use.

It is obvious that we should protect data security using encryption. In particular, we need to encrypt not only the data at rest, but also access to resources because otherwise every service can learn about the computing tasks being performed.

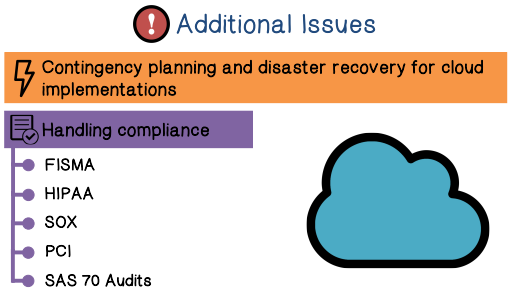


Now let us do a quiz on cloud security, which of the following statements are true?

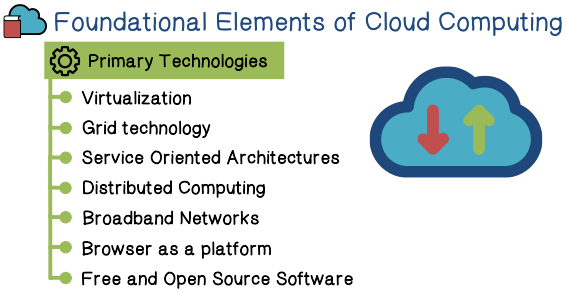
The first statement, most data in transit is encrypted, this is true. Close to 90% of the service providers encrypt data in transit. The second statement, most data at rest is encrypted, this is false. Only 10% of the providers encrypt data at rest. The third statement, all data at rest should be encrypted, this is false. In reality there is plenty of data that does not require security protection, and therefore such data does not need to be encrypted at rest.



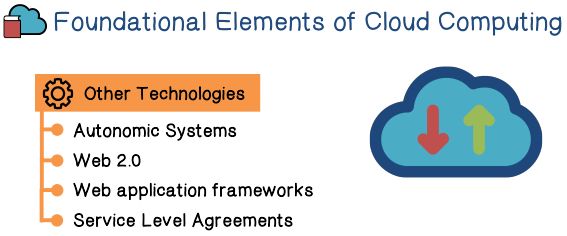
There are some additional issues associated with cloud security. In particular, we need to consider the privacy impact when PII data is moved to the cloud. PII stands for personal identifiable information, and it is any data that could potentially identify a specific individual. Customers should be aware that they need to negotiate with the cloud providers to obtain a security service level agreement. But then we need to make sure that we have proof that the service level agreements have been satisfied.



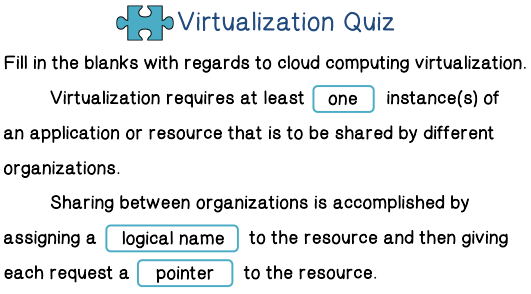
Obviously, a cloud provider needs to plan for disaster recovery. In addition, a cloud provider needs to consider a number of compliance issues. For example, HIPAA for healthcare data and PCI for payment data.



Let us discuss the foundational elements of cloud computing. Cloud computing is built on a number of technologies. These include virtualization, grid technology, service-oriented architectures, distributed computing, broadband networks, browser as a platform, and free and open source software.

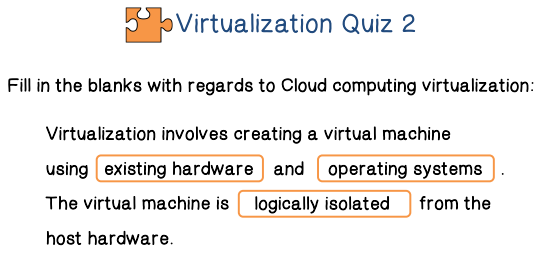


Other technologies include autonomic systems, Web 2.0, web application frameworks, and service level agreements.



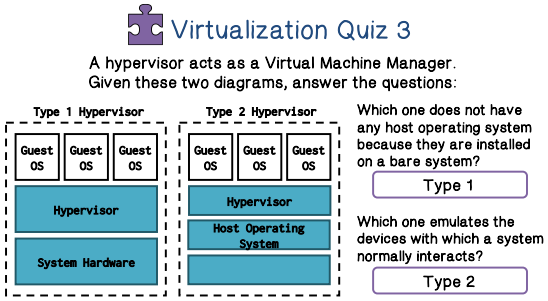
Now let us do a quiz on virtualization. Fill in the blanks with regards to cloud computing virtualization.

Virtualization requires at least one instance of an application or resource that is to be shared by different organizations. Sharing between organizations is accomplished by assigning a logical name to the resource and then giving each request a pointer to the resource.



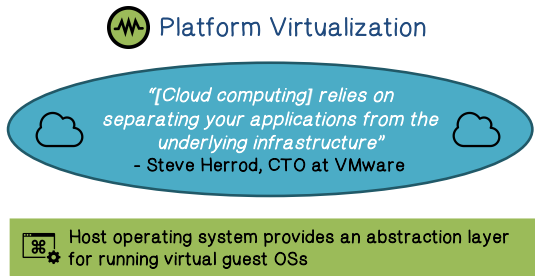
Fill in the blanks with regards to cloud computing virtualization.

Virtualization involves creating a virtual machine using existing hardware and operating systems. The virtual machine is logically isolated from the host hardware.

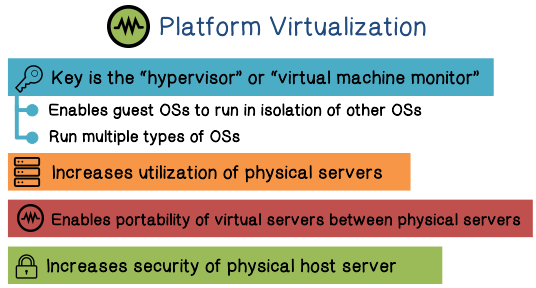


Let us do another quiz on virtualization. A hypervisor acts as a Virtual Machine Manager. Given these two diagrams, answer the question.

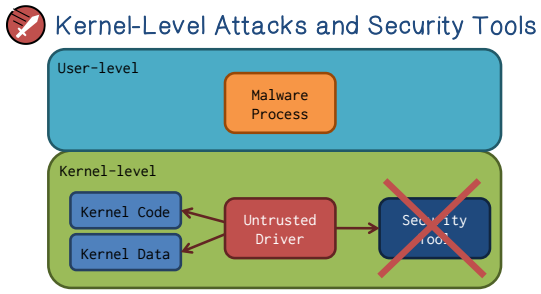
Which one does not have any host operating system because they are installed on a bare system? That is a Type 1 hypervisor. Here, notice that the operating system runs in a virtual machine on top of the hypervisor. Second question. Which one emulates the devices with which a system normally interacts? That is the Type 2 hypervisor. Here, the hypervisor emulates the hardware.



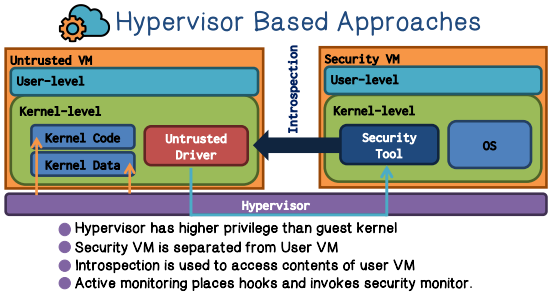
The most important technology for cloud computing is virtualization. This is because cloud computing relies on separating applications from the underlying infrastructure.



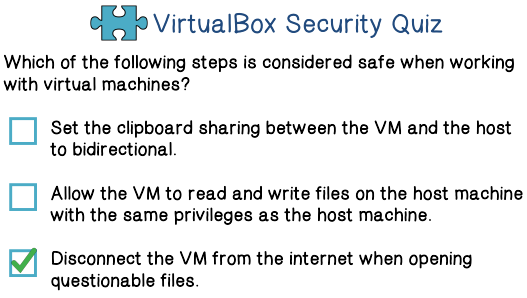
The key to virtualization is the hypervisor or the virtual machine monitor in addition to increased utilization. For example, running many virtual machines on a single hardware, virtualization can also improve security and we will discuss this next.



Let us first review what happens when we placed a security tool in the opening system kernel, which has higher privilege than user-level applications. A kernel-level security tool has a higher privilege, and therefore, it can detect and remove a malware process at user-level. But since the security tool has the same privilege as the attacker in the kernel, such as a rootkit, it can be compromised. In other words, the security tool cannot be isolated or protected from a kernel-level attacker.

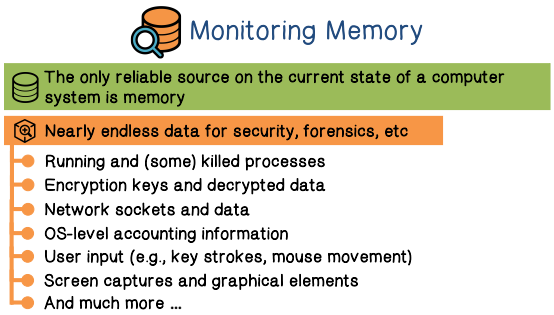


With virtualization, we can put the security tool in a separate virtual machine. For example, we call this the security virtual machine dedicated for security analysis. Now the security tool is isolated from the virtual machine that has the malware. But then the question is, how can a security tool detect or stop a malware in a different virtual machine? As we will discuss next, the security tool, with the help of the hypervisor, can perform introspection of the other virtual machine.

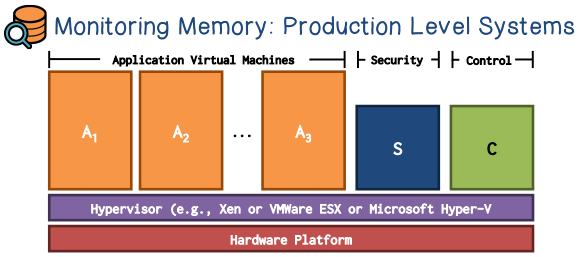


First, let us take a moment to think about how virtual machines can be exploited. Here is a quiz on VirtualBox security. Which of the following steps is considered safe when working with virtual machines?

First, set the clipboard sharing between the virtual machine and the host to be bidirectional, this is not safe. Second, allow the VM to read and write files on the host machine with the same privilege as the host machine, again, this is not safe. This is similar to allowing an attacker to write files on your machine. The final statement, disconnect the virtual machine from the internet when opening questionable files, this is safe because this will prevent potential malware from contacting its command control server.

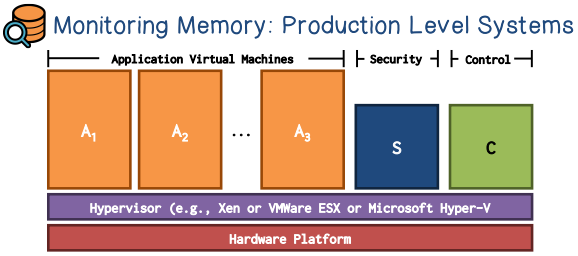


One of the most essential tasks of virtual machine security is memory analysis. This is because memory is the only reliable source of the current state of a computer. For example, using memory analysis we can find out a list of the running processes. We can also find out the encryption keys being used and the decrypted data. We can look at network socket and data. We can also find OS-level accounting information, as well as user inputs. We can also read the screen captures and find the graphical elements of an application and so on and so forth. And these are just a few examples.



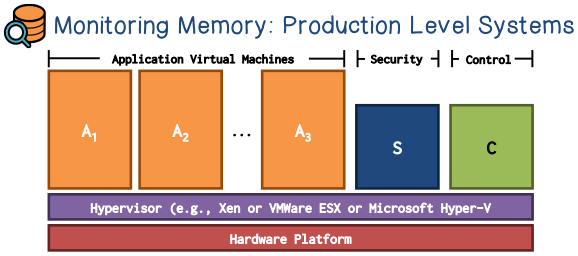
Passive Monitoring: (1) Viewing memory in A_n from S without any timing synchronization between the two virtual machines

Let us discuss memory monitoring and analysis in a virtualization environment. The security and control virtual machines are smaller because they can run a stripped-down OS. The security machine gets a raw memory view, meaning that it just sees the physical memory from the other virtual machines. If you want anything useful, you need to rebuild the abstraction levels on your own. And this is challenging, and we will discuss this shortly. There are several types of monitoring. The first is passive monitoring. This means that the security virtual machine takes a snapshot view of the raw memory of a virtual machine.



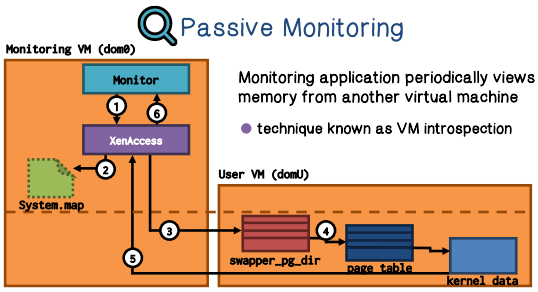
Active Monitoring: Viewing memory in A_n from S with event notification being sent from A_n to S to permit monitoring at relevant times

The second type is active monitoring. This means that the security virtual machine takes a view of the memory of another virtual machine when there is an event being triggered at that virtual machine.

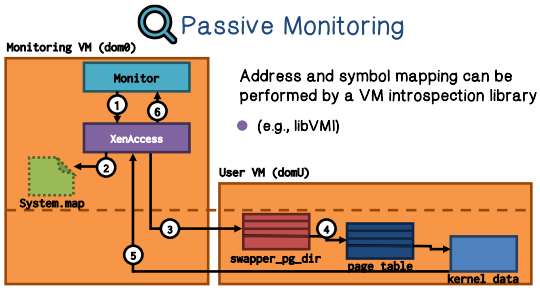


Locating Valuable Data: find critical data structures within the raw memory view for the monitoring task

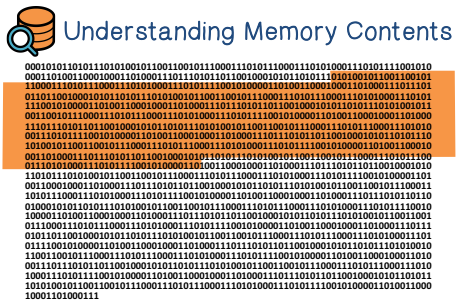
An important goal of memory monitoring is to locate the important data structures in memory for security analysis, for example, examining the process list.



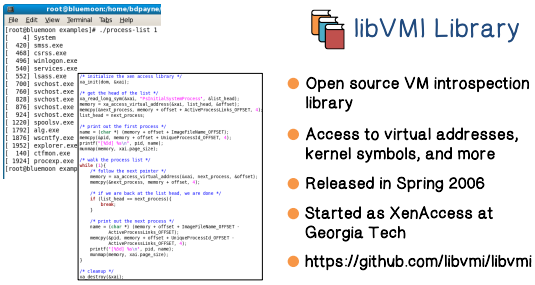
Here is an example of passive monitoring. Again, this means that the security virtual machine monitors that application periodically by getting the view of the memory from another virtual machine. This is known as performing virtual machine introspection. And here are the main steps in passive monitoring. In the security virtual machine, the security tool performs an API call to access a kernel symbol. The address of the kernel symbol has been looked up and then page tables for the memory of the user virtual machines are being traversed to locate the kernel data. And then, the pointer to the memory is returned to the security monitor.



We use a virtual machine introspection library called libVMI to convert the raw memory view into something meaningful, such as, virtual addresses, kernel symbols, etc., and we will discuss libVMI shortly.

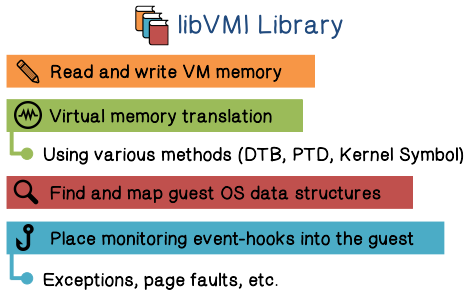


Now we need to address one very important detail. When you read memory, all you see are 1s and 0s. But how do you know what the 1s and 0s mean, unless you know the memory layout, according to data structure definitions? For example, this blob may represent a data structure that you are looking for. Therefore, as we have discussed in passive monitoring, we need to convert the raw memory view into something meaningful such as virtual addresses kernel symbols, etc.

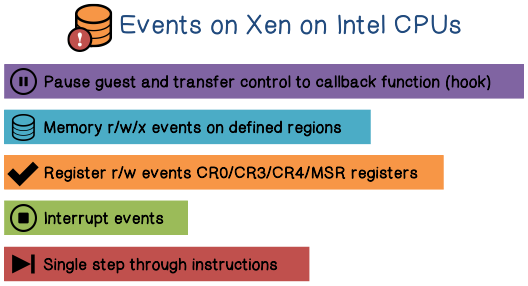


We use libVMI, or previously called XenAccess, to analyze memory contents. This is an example of the code that used libVMI APIs to obtain the list of learning processes. And this is the output. The point here is that with libVMI or XenAccess, you do not even write a whole lot of code in order to

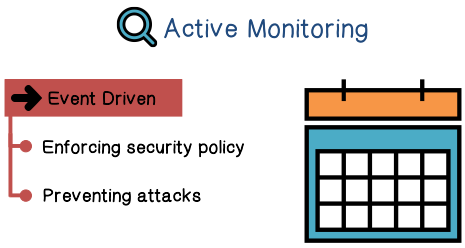
obtain such useful runtime information of the other virtual machine. libVMI is open source, it provides access and analysis of virtual addresses, kernel symbols. It was first released as XenAcess in Spring 2006, and here is the GitHub repository.



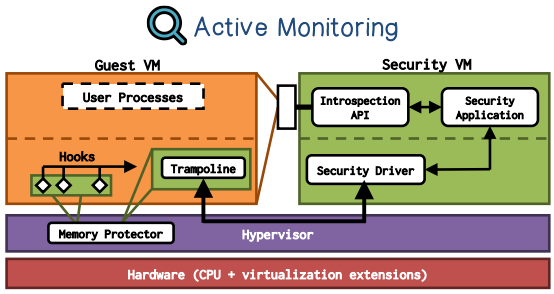
Here's a quick summary of the libVMI features, you can perform virtual memory translation, for example so that you can know which are the kernel symbols. You can also place monitoring hooks into a guest virtual machine, for example, to trap exceptions and page faults.



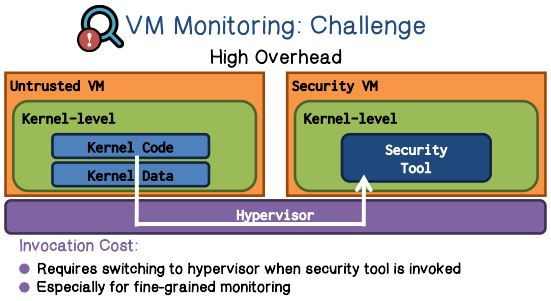
For example, hooks can be placed in a virtual machine to trap the following events. For example, memory read-write-execute events, register read-write events, interrupts and single stepping of instructions. This is useful to trace a log of program execution.



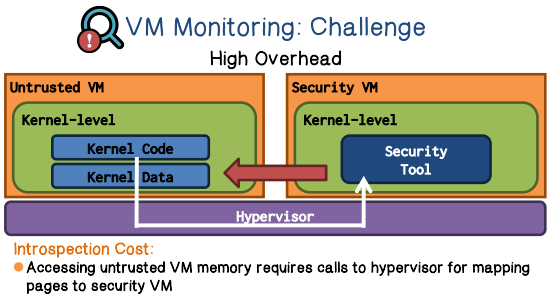
Now, let's discuss active monitoring. Active monitoring is event driven which allows for enforcing security policy and stopping attacks before they happen.



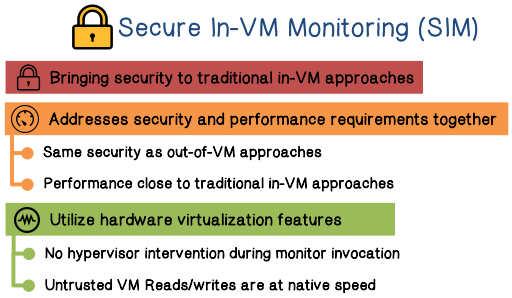
Here's an example of active monitoring. The security application receives event notification from the guest virtual machine when the code execution, which is one of the hooks, hooks invoke trampoline which transfer the control to the security application. The hooks in the associated code in the guest virtual machine are protected using the memory protection provided by the hyperviser. Of course, when a security application receives an event notification, it's going to perform virtual machine inspection, for example, using the lip VMI.



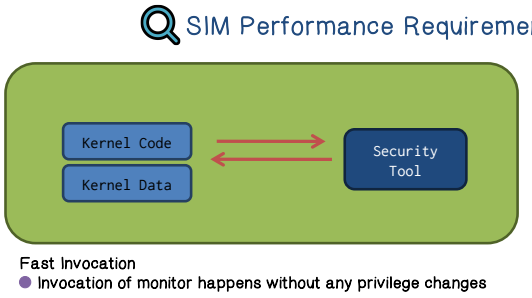
There are several challenges emerge in machine monitoring. The first is high overhead. The overhead comes from several sources. The first is invocation. Switching from a virtual machine to the hypervisor into a virtual machine is very expensive.



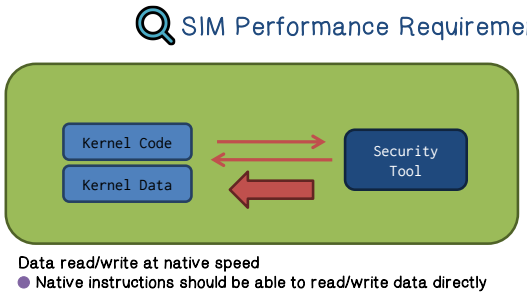
The second source overhead is introspection. Again, accessing the memory of another virtual machine requires calls to the hypervisor.



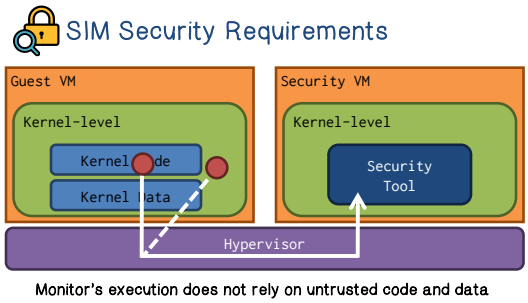
The question is can we have both the security benefit provided by virtual machine monitoring which means out of VM monitoring and the efficiency of traditional monitoring which means In-VM monitoring. We have developed a secure In-VM monitoring approach. We call this approach SIM for secure In-VM monitoring. It provides the same security as the out of VM approach. And its performance is similar to the traditional In-VM approach, the main idea is to use hardware which virtualization features to minimize the need to switch to hypervisor. In other words, we can read or write the memory of an untrusted virtual machine at native speed.



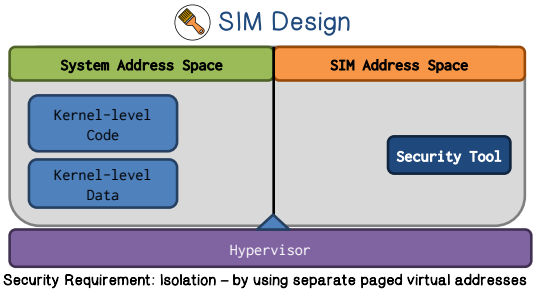
Let's analyze the requirements of SIM. We want the invocation to be really fast. This means that there's no need to switch to hypervisor.



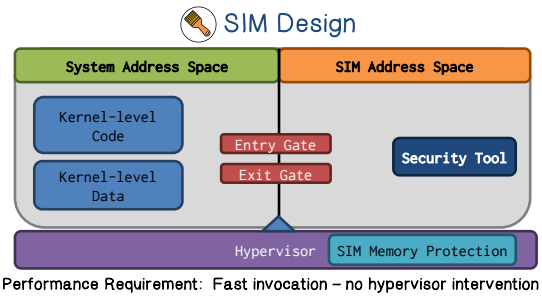
We also want data read and write at native speed. This means that we can rewrite data directly, without going through hypervisor.



But there are also security requirements. The code and data of the security tool need to be isolated from the untrusted machine. The handling of the security events has to be secure. Furthermore, the security tool should not rely on untrusted code and data.

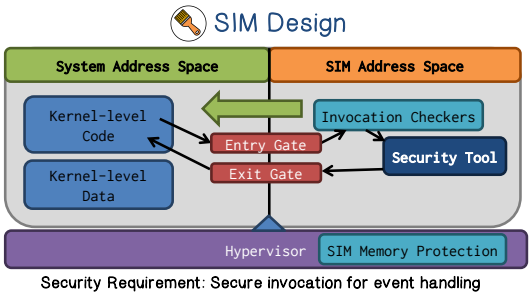


Now let's discuss how SIM can satisfy both the performance and security requirements. Recall that in operating systems, paging-based virtual memory is generated by creating page tables that map virtual addresses to physical addresses. An operating system creates a separate page table for each process so that it can have its own virtual memory address space. And the necessary isolation can be achieved. The SIM address space here contains all kernel code data and also the SIM data in its own address space. Therefore, the instruction as part of the security monitor can access in native speed. Notice that the guest operating system has it's own virtual address space. That is, although we put SIM into the same virtual machine as the guest operating system, they have their own separate and different module address space.



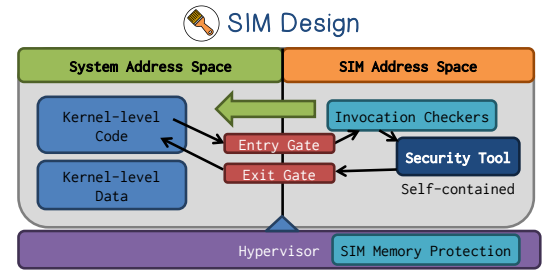
In order to perform security monitoring, SIM needs to look at the address space of the guest operating system. And this can only be done through the Entry Gate and the Exit Gate since this requires the switching of address spaces. We need to modify the CR3 register contents directly. Intel VT, contains a feature that

doesn't check out a VM exit to the hypervisor, if the CR3 is switched to an address space that's predefined and maintained by hypervisor. Therefore, by predefining the SIM address space and system address space we can achieve fast switching without exiting to hypervisor. We also use the Hypervisor memory protection to protect the security of the SIM address space and protect the entry and exit gates.



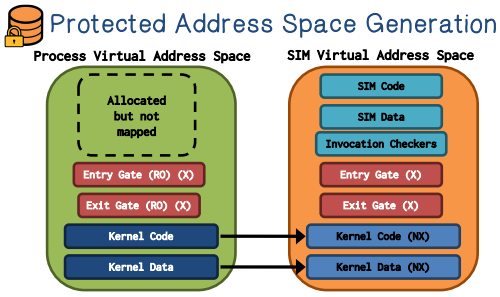
Security Requirement: Secure invocation for event handling

The entry gates and exit gates are the only ways to switch between the SIM address space and the system address space. We use inter-hardware feature called the last branch recording in the invocation checker. We last branch recording. We know the last few basic blocks leading to the entry gate.

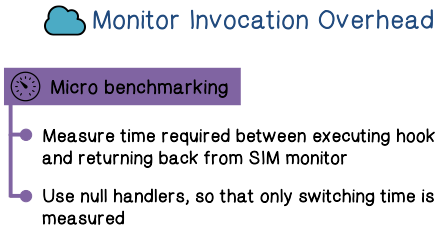


Security Requirement: Monitor's code does not rely on anything untrusted

The SIM security tool is self contained, meaning that it does not caught any quote or using any data from the kernel of the untrusted region. The SIM security tool can read my memory in native speeds.



Let's compare the memory layout between the SIM virtual address space and a system virtual address space. As you can see the SIM code and data cannot be viewed by the untrusted system. The entry gate and exit gate are executable in both spaces. Their security is provided by memory protection in the hypervisor and the invocation checker in SIM. The kernel code and data from the untrusted system can be read directly by SIMbut y SIM does not execute such untrusted code and data.



We perform experiments to measure the overhead of SIM and compare it with the out-of-VM monitoring approach.

Monitor Invocation Overhead

Monitor Type	Avg. time (u sec)	Std. dev (u sec)
SIM approach	0.469	0.051
Out-of-VM approach	5.055	0.132

And here are the results, we can see that SIM is much faster than Out-of-VM approach.

Secure and Flexible Monitoring of Virtual Machines

The monitoring of virtual machines has many applications in areas such as security and systems management. A monitoring technique known as introspection has received significant discussion in the research literature, but these prior works have focused on the applications of introspection rather than how to properly build a monitoring architecture. In this paper we propose a set of requirements that should guide the development of virtual machine monitoring solutions. To illustrate the viability of these requirements, we describe the design of XenAccess, a monitoring library for operating systems running on Xen. XenAccess incorporates virtual memory introspection and virtual disk monitoring capabilities, allowing monitor applications to safely and efficiently access the memory state and disk activity of a target operating system. XenAccess' efficiency and functionality are illustrated through a series of performance tests and practical examples.

This paper described XenAccess, a monitoring library for Xen virtual machines. XenAccess' development was guided by a set of design principles aimed at providing a solid foundation for secure and flexible virtual machine monitoring. XenAccess implements virtual memory introspection and virtual disk monitoring capabilities by leveraging Xen's existing infrastructure. By using it to access the target VM's raw memory pages and disk I/O, XenAccess is able to infer OS data structures and filesystem operations at a useful abstraction level. Our evaluation revealed that XenAccess imposes a minimal performance overhead to the target OS memory and disk operation. We also showed practical examples of the type of information that memory introspection and disk monitoring can gather, illustrating the potential of each technique.

Secure In-VM Monitoring Using Hardware Virtualization

The paper proposes SIM, a general-purpose Secure In-VM monitoring framework that provides the same guarantees of out-of-VM monitoring, along with the low performance overhead of in-VM monitoring. It enables security monitoring applications to be placed in an untrusted guest VM for efficiency, without sacrificing any security guarantees.

Requirements:

The performance requirements of SIM are fast invocation, and data read/write at native speed. The security requirements of the monitor are isolation of code and data, designated point for switching into code, handler-hook execution restrictions, and secure behavior that is not

maliciously alterable. Refer Section 2, 'Background and Requirements' for a detailed discussion on the requirements.

Design:

Figure 2 shows the overall design of SIM. Section 3 describes the design in detail and must be read carefully to understand the paper. The approach utilizes hardware memory protection and hardware virtualization features to create a protected address space. This space can only be accessed through specially constructed protected gates. The gates are write-protected by the hypervisor and cannot be modified by in-guest code. Normal operation of the monitor continues without hypervisor invention. But, any attempts to breach SIM are trapped by the hypervisor and prevented from occurring. Invocation of a gate needs to be checked to ensure it was called only from the hook that has the permission to call a gate. This makes use of a hardware debugging feature called last branch recording (LBR).

Implementation:

A prototype of the SIM framework based on KVM and Windows guest OS was implemented and described in Section 4. It consists of two phases, initialization and execution.

The task of initialization is to allocate VM address space for the guest to place entry and exit gates based on required hooks, initiate creation of SIM virtual address space, initiate the loading of the security monitor into the address space, and finally create entry and exit gates and invocation checking routines. The initialization driver communicates with the hypervisor of SIM using hypercalls. Read Section 4.1 for a detailed explanation.

At runtime, the security of SIM must be protected. Memory protection is verified whenever a guest page table entry is propagated to the shadow table maintained by the hypervisor. Figure 6 provides a visual summary of runtime protection.

Two security monitoring applications were created, and their performance evaluated in Section 5, but the main takeaway is that there is a far lower overhead in SIM as compared to out-of-VM approaches. A systematic security analysis was conducted (Section 3.3) against a number of possible threats. It is shown that SIM provides at least the same security guarantees as those achieved by out-of-VM monitors.