# Virtualisation security: Virtual machine monitoring and introspection

Increasingly, critical systems are being virtualised in the name of cost savings. At the same time, there has been an increase in standards and legal/regulatory obligations. **Fotios Tsifountidis** and **Dr. Geraint Price** examine how security will have to adapt to monitor virtualised environments.

## INTRODUCTION

Organisations are gradually gaining confidence in virtualisation, having realised the benefits it can bring over traditional computing infrastructures and practices. Today, virtualisation is most commonly used to describe the partitioning of a physical system into multiple virtual instances (OS), each working individually and separately from the others.

Increasingly, critical systems are being virtualised in the name of, amongst other things, cost savings. At the same time there has been an increase in standards and legal/regulatory obligations, which force businesses to comply with them. It is well known that information security can have a significant impact on the requirements for compliance. The road to success in ensuring the confidentiality, integrity and availability of information in virtualised environments requires a greater understanding of virtualisation's security pitfalls and the tools available for tackling them.

Until recently, the security mechanisms built for protecting traditional environments have also been used for protecting virtualised environments. The existence of mechanisms like intrusion detection and prevention systems can be traced back to long before their recent adoption by virtualisation environments. These systems have powerful monitoring capabilities and are extensively and successfully used in traditional computing environments today.

The fact that these technologies were not originally designed with virtualisation in mind means that adapting them for use in virtual environments often makes them less effective. The major trade-offs faced when using these protection technologies within a virtualised environment are mostly due to either unacceptable resource utilization or an inadequate ability to inspect the internal workings of the OS. These issues gradually led to the diminished use of protection mechanisms of this kind — at least in virtualised environments.

New protection technologies, such as virtual machine introspection (VMI), have emerged to protect virtualised systems. VMI technology is explicitly built and tailored for monitoring virtualised environments. Although it successfully increases the level of protection offered, it does have its own quirks and limitations.

This article gives an insight into virtualisation technology, emerging threats, and available monitoring mechanisms. It also discusses certain critical limitations of VMI's ability to provide a secure platform and briefly describes how, in a clever

twist, we may turn these vulnerabilities back on the attacker and take advantage of these vulnerabilities to provide an additional layer of defence.

## VIRTUALISATION BACKGROUND

Virtualisation is not a new concept. Its virtualisation begins approximately four decades ago, when the first prototype of a system that incorporated virtualisation was built by IBM with its System VM/360. The main driver behind the creation of this technology was the need to take full advantage of the massive, expensive and powerful mainframe computers which marked that era.

As in the past, virtualisation's main function today is to offer a virtual copy of the underlying hardware to the software running on the system. By creating many "copies" of the underlying hardware in this way, it is possible to install several software environments (guest virtual machines (VM) — e.g. operating systems) on one physical machine (host system) and have each guest virtual machine operate individually in isolation from the others.

The users that interact with these environments have the illusion that they are interacting directly with the OS and the hardware. However, the requests that the software (in both the application and OS layers) make to the virtual environment in which it is running aren't directly communicated to the hardware, but are processed by the many computing layers that exist between the virtual machine and the actual hardware.

*With virtualisation, all these requests have to go through the software component that lies at the heart of every virtualisation solution in use today — the virtual machine monitor (VMM).*

With virtualisation, all these requests have to go through the software component that lies at the heart of every virtualisation solution in use today — the virtual machine monitor (VMM) or hypervisor. The VMM is responsible for managing the hardware resources, making these resources available to each virtual machine, and enforcing isolation between the different virtual environments. It is critical that the VMM correctly handles requests made by the VM to the hard-

ware and it is this role that makes the VMM such a sensitive component to the security of the system as a whole.

Due to their sensitive role, VMMs are perceived to be components that have strong security controls and robust designs. However, being software implementations, VMMs inevitably inherit the associated drawbacks of software design and coding. Good old buffer overflows and similar problems caused by unsafe coding practices can still find their way into a VMM.

Despite the security concerns, virtualisation enjoys significant acceptance today, mainly because of the cost savings it can deliver. These cost savings mainly come in the form of system consolidation, which allows several distinct systems to run on one machine and which gives rise to lower energy and space costs.

In addition, virtualisation is the foundation of the upcoming trend of cloud computing. This new trend allows services and operations to be outsourced to third-party vendors and delivered via the Internet. This allows a third-party vendor to maximise the use of its resources (processing power, memory, storage, etc.) by splitting them between several clients. Ultimately, users only pay for the resources they consume.

## SECURITY RISKS

Given the recent interest in virtualisation, it has the potential to change the way people approach their computing requirements at a fundamental level. Unfortunately, virtualisation inherits many of the security risks traditional computing faces and additionally introduces new security problems that do not exist in traditional computing environments.

**Malware.** Today, malware represents a major problem that can threaten every computerised environment. Businesses face huge financial or reputational losses due to malware infections. Virtualisation cannot protect against such attacks, and, in fact, may make matters worse as it provides additional paths for infecting a system. Modern malware can detect a virtual environment and respond by disabling or infecting critical components such as the VMM. There are instances of sophisticated and persistent malware, specifically rootkits, which use virtualisation themselves. After infecting a system they host it in a virtual machine and use virtualisation techniques to intercept the owner's actions.

**Network.** Owing to the complexities of the virtual environment, network configuration becomes even harder when virtual machines are introduced to the network. A physical machine

may host several virtual machines and the task of configuring the connection between the virtual machines and the rest of the network is not a trivial task. The traditional hardware-based network switches, responsible for connecting network segments, are now replaced by software implementations (vSwitches) in virtualised environments. vSwitches are prone to traditional layer-2 networking attacks and the limited visibility of these components only serves to lower the network's operational and configuration assurance.

**Software Flaws.** As mentioned previously, virtualisation inherently comes with the same security problems that affect all types of software. Numerous advisories have been issued for bugs in all major virtualisation platforms in use today. Certain flaws can lead to VM escape issues — attacks that allow a malicious piece of software to escape the confines of a virtual machine and compromise the host machine. Users usually find themselves with a false sense of security when they just patch the host and guest operating systems. These vulnerabilities target the virtualisation layer and can be successfully exploited even if the patching regime of the operating systems and their applications is strict. Thus, managers need to treat the VMM as a critical element in the security system and keep it up to date and in line with the vendor's security patches and updates.

**Administration.** Businesses tend to create virtual machines for testing/development purposes to isolate them from the production environment. For instance, a developer might have a virtual machine for testing his code/application to avoid potential crashes in his own operating system. The reality is that both administration and security controls for these testing environments are usually loose, making them an easy way for an attacker to infiltrate the production environment.

Robust administration and least privilege enforcement are needed to identify and prevent random people from introducing unauthorised VMs to the infrastructure. Certain advantages of virtualisation technology can also become weak points. For example, a VM's configuration is stored as a single file, which makes it easier for an attacker to copy or delete these files and potentially steal a whole VM (and its stored information).

Similarly, the ability to rollback VMs and restore them to previous states can assist administrators in resolving several problems (e.g. malware infections). However, there have been instances where the rollback functionality has caused problems with cryptographic protocols. Many authentication protocols use the system's state to produce unique

one-time keys. After being rolled back, a system could potentially recreate identical keys in future communications and diminish the security of these protocols.

*Good practices and information security standards today mandate the existence of robust monitoring functions.*

## MONITORING MECHANISMS

The security challenges, along with the complexity and manageability issues mentioned in the above sections, need to be efficiently solved if virtualisation is to be managed and monitored effectively. Good practice and information security standards mandate the existence of robust monitoring functions to detect and prevent security incidents. Monitoring capabilities will not only help increase confidence in an environment's security posture but also contributes to the administration of today's increasingly complex virtual environments. The administrative burden and infrastructure complexities can be reduced by using virtualisation management software to monitor the VMs' operation.

## MANAGEMENT SOFTWARE

High-level management of virtual machines usually takes place through centralised software. Such software can issue alerts about the state of each VM within the virtualised infrastructure using indicators like performance characteristics, network traffic, patch information, and so on. Modern virtualisation management software can be seen as a first level of security monitoring. Unfortunately, the VMM vendor controls the amount of access that a piece of third-party management software is allowed, and this is usually limited. Effectively, customers are left with few options to choose from when it comes to management suites.

## INTRUSION DETECTION AND PREVENTION

Management software is adequate for administering and overseeing the virtual infrastructure but does very little when it comes to security. This role is filled by intrusion protection systems (IPS). These systems are based on either hardware or software implementations which strategically place software components (sensors) in the protected systems. These sensors offer security-oriented inspection capabilities and are triggered in case of system violations.

There exist two different approaches to the

development and use of an IPS. The first approach places the IPS inside the VM — this is known as a host-based intrusion protection system (HIPS). A different approach installs the IPS in the network perimeter and allows the IPS to monitor a complete network segment (domain) with multiple VMs — this is known as a network-based intrusion protection system (NIPS). Figure 1 depicts how the two approaches described above are implemented in a virtualised environment. Each of these approaches has its own advantages and disadvantages (which we will briefly discuss). The way an IPS is placed within the infrastructure is a major factor in determining the security guarantees it offers.

*The major advantage of HIPSs is their defence-in-depth capabilities.*

The major advantage of HIPSs is their defence-in-depth capabilities. Since they are installed inside the VM they have complete visibility of the running processes and are able to monitor internal VM operation for potential dangers. This means that previously unknown threats (e.g. zero-day threats) could potentially be detected and neutralised due to their "abnormal" behavior.

On the other hand, the main advantage of NIPSs is the capability to monitor all communications between the VMs and the VMM (and ultimately the world). Packets sent between the world and the VMs can be inspected for malicious traffic and any attacks contained in these packets can be prevented before the attacks materialise. A centralised point of security enforcement is usually beneficial, as it offers easier configuration opportunities and imposes less of an administrative burden. Lastly, the fact that the NIPS resides outside of the VMs gives a measure of independence from the VMs' operating systems and deployments.

However, this discussion clearly shows the limitations of host-based and network-based intrusion prevention systems. By their nature, host-based mechanisms are vulnerable to sophisticated malware attacks that disable the HIPS from inside the VM. Even without attacks, the need to configure each HIPS separately is likely to be an overwhelming administrative burden and it is not uncommon for HIPSs to suffer operational difficulties when the core components of the VM are updated. Lastly, dedicated protection for each VM would consume large amounts of the host's processing and memory resources.

All of these limitations could be avoided by

using the NIPS approach, but unfortunately this comes at a price too. Network-based protection means that the IPS cannot monitor actions taking place inside the VMs. Perimeter protection also means that attacks originating from one VM and which target another VM in the same domain cannot be prevented. In this context a NIPS can only view a domain as a single entity and inspects traffic between that domain and the hypervisor.

*Given the advantages and disadvantage of HIPSs and NIPSs, we can clearly see that what is offered by one approach cannot be offered by the other, and vice versa.*

Intercommunication between VMs in the same domain flows through a vSwitch without being inspected (see Figure 1). When NIPSs are used, more attention is needed to ensure that the VMs residing within the same domain share similar risk profiles. It would not be wise to have VMs with different security requirements and risk appetites being protected by a single NIPS under the same security policy.

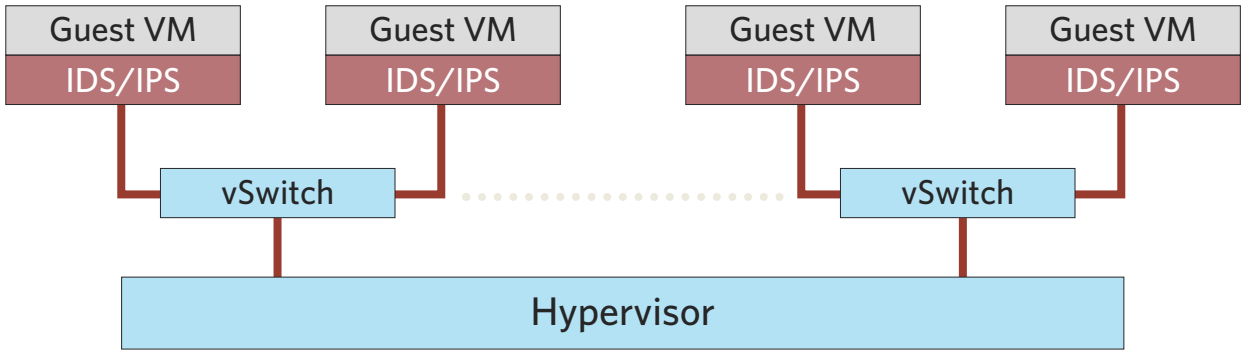
Given the advantages and disadvantage of HIPSs and NIPSs, we can clearly see that what is offered by one approach cannot be offered by the other, and vice versa. These technologies complement each other and in an ideal world they would be used in conjunction. However, for virtual environments, where one physical system hosts a number of VMs, the hybrid approach would introduce substantial performance overhead to the host's limited hardware resources.

## VIRTUAL MACHINE INTROSPECTION

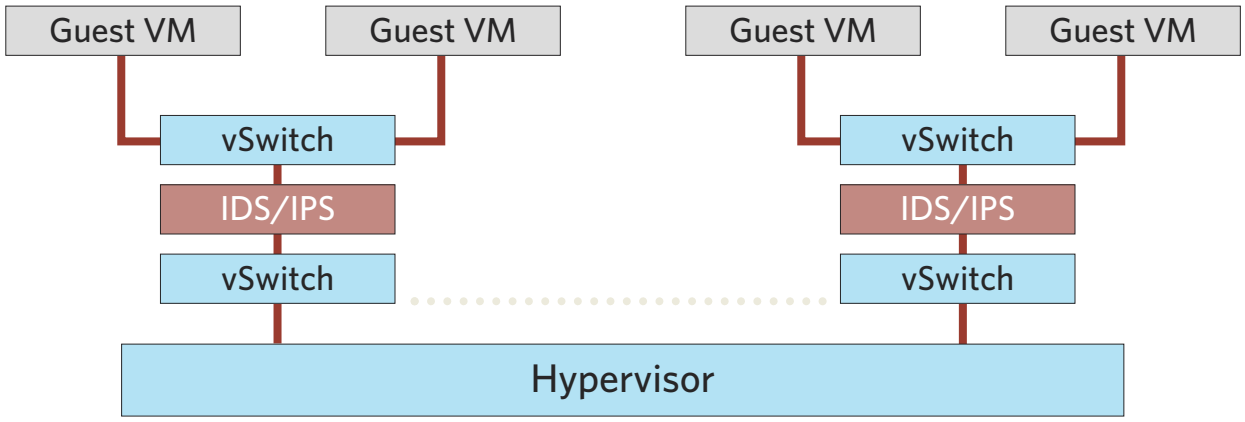
Virtual machine introspection (VMI) is a hybrid approach to virtualisation monitoring and protection placement. It aims to combine the functionality of the two protection systems (HIPS and NIPS) discussed previously in an effort to achieve the best result overall.

VMI architecture is based on a dedicated VM responsible for managing and protecting all other VMs. The protection VM incorporates the security policies and specific rules that are to be enforced on the guest VMs. This special VM interfaces with the VMM and intercepts and inspects all low-level information (interrupts, memory accesses, etc.) that flows between the VMM and

FIGURE 1

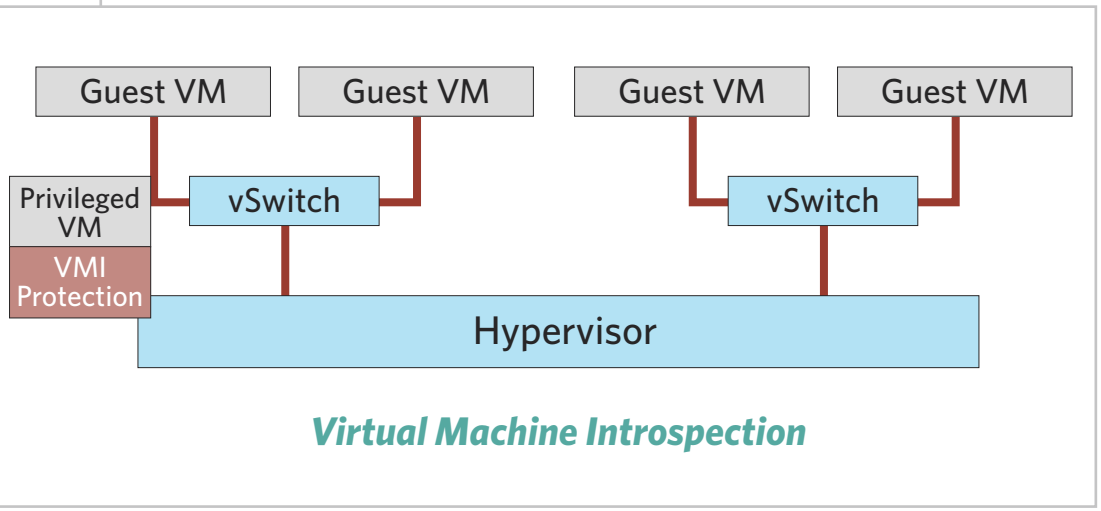



***Host-based protection***




***Network-based protection***

FIGURE 2




*Virtual Machine Introspection*

the guest VMs. As all security operations are performed with the VMM's assistance, the monitoring process incurs little performance overhead. The security VM is isolated from the other guest VMs and runs with higher privileges than the guest VMs. This adds an extra layer of protection against malware attacks that originate in the unprivileged VMs. Figure 2 depicts the VMI's topology within the infrastructure.

The outcome is a protection mechanism with all the advantages of traditional monitoring technologies: a robust, efficient and centralised security point that is able to inspect all VM communications and obtain some information about their internal operation. The majority of today's malware threats and attacks can be mitigated by VMI technology. The numerous advantages of VMI over traditional intrusion prevention systems are well understood by virtualisation vendors and they have incorporated VMI technology into their commercial product lines. The majority of today's virtualised infrastructures are protected by VMI-based security implementations.

## VMI LIMITATIONS

No security technology is perfect and VMI technology is no exception. Its limitations are closely related to the fact that the security VM is isolated from the VMs that it protects. The security VM

has a large amount of low-level information about the internal processes of the guest VMs (e.g. all VM memory access requests). However, this information by itself is useless, due to the difficulty in reconstructing the actual operation of the guest VM from the low-level information obtained by the security VM. Effectively, even though the security VM can collect a lot of information about the guest VM's actions, it still cannot understand the meaning behind those actions. This issue is known as the semantic gap. Without the necessary semantic awareness it's impossible to determine the actual purpose of the contents of the VM's memory.

*Without the necessary semantic awareness it's impossible to determine the actual purpose of the contents of the VM's memory.*

There exist several different approaches to gathering low-level VM information and constructing a level of semantic awareness about the protected system. However, to date, none of these approaches has successfully and completely bridged the semantic gap.

All existing information gathering methods are based on the assumption that a VMI application will be presented with correct information about the internal processes and operation of a guest VM. This is not always the case, especially for a compromised VM which may present the VMI with flawed information. Effectively, VMI applications run the risk of gathering information that has been tampered with.

## ATTACKS

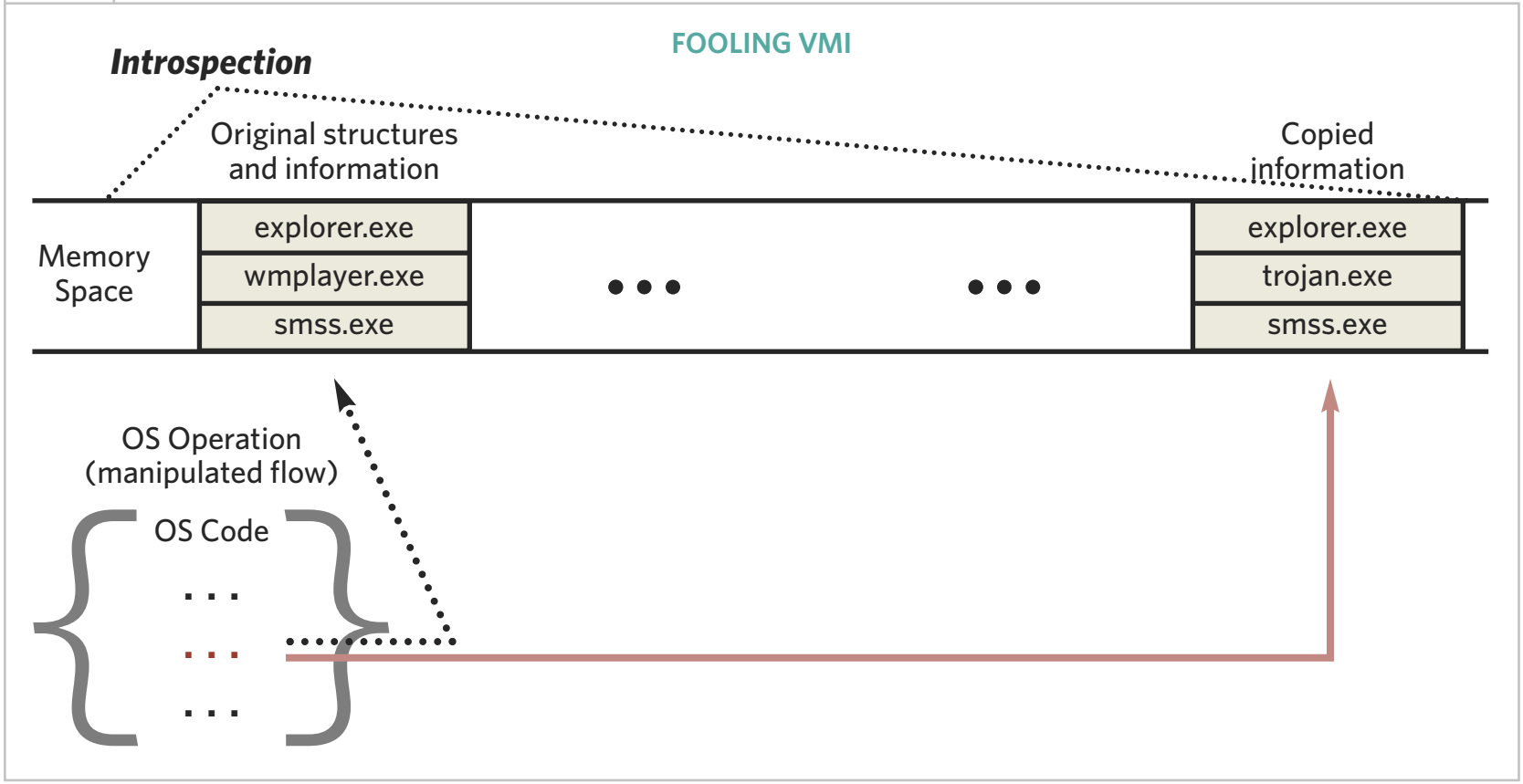
New low-level attacks, called direct kernel structure manipulation (DKSM) attacks, have made use of this limitation. These attacks manipulate the VM data that VMI applications use to construct semantic awareness. VM operating information (e.g. the process list) is copied to a new memory location by malicious code and the infected VM is instructed to continue operating using the information stored in the new locations. If a VMI application monitors the original memory locations to gather security information, then their contents will appear benign, as these memory locations are no longer used by the infected VM.

Ultimately, the attacker has the ability to deter-

mine what the VMI application sees. The diagram in Figure 3 shows an example where a VM has been infected by a Trojan application which is invisible to a VMI application. The Trojan application is presented to the VMI as a legitimate application (i.e. Windows Media Player).

Although a VMI application can inspect the locations with the copied information, their contents are just seen as memory bits without specific semantics. These attacks even make it possible to hide data from the VM itself. Using the example in Figure 3, all it takes for the attacker is to change

FIGURE 3



trojan.exe to something that looks legitimate, e.g. svchost.exe. Effectively, the VMI application sees wmplayer.exe and the protection mechanism inside the VM sees svchost.exe, which allows the actual contents of the memory (the malicious application) to run unhindered.

*The potential result is worrying: an elusive attacker with the ability to covertly take over the virtual machines.*

These attacks are not limited to just manipulating process names, but can be used to alter any kind of system information. The potential result is worrying: an elusive attacker with the ability to covertly take over the virtual machines. Furthermore, none of the current VMI products have the capability to detect or prevent these attacks.

## FINAL NOTES

These attacks seem devastating, but it is possible that some good could from them. We might be able to take advantage of them for defensive purposes. The objective would be to instruct a VM to automatically conceal its own sensitive information in the case of an attack.

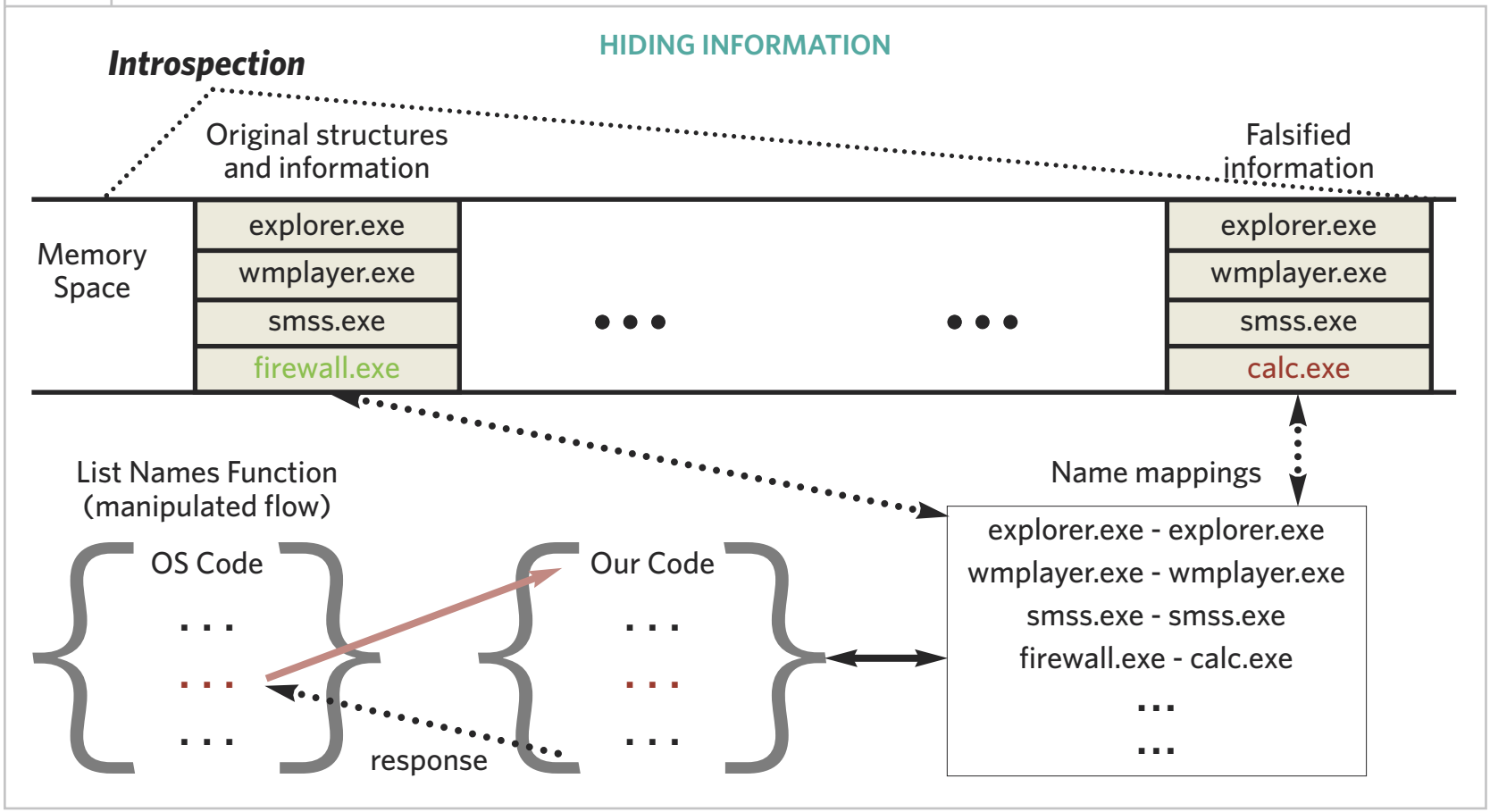
We briefly outline how such a defensive mechanism might work — the VM's operating information has to be copied to a location in memory and partially changed. A new piece of code becomes the intermediary between all OS applications that query VM data (e.g. for listing processes) and the actual data itself. Triggers/hooks are placed within the VM for detecting potential attack attempts. As long as the system is perceived to be safe, the new protection code returns the genuine value to a request. If a hook is been triggered (i.e. a potential attack is detected), then the intermediary returns a falsified representation of the requested value. Figure 4 depicts the notion of this concept when we want to hide firewall.exe from an attacker and present it as calc.exe. As with the attack, this allows us to misrepresent system information to a VM application, except in this case it is the malware that is being deceived.

This approach should only be seen as a last resort attempt to prevent information disclosure in the case where everything else has failed. The operational lifetime of this protection mechanism should be short — it only needs to operate in the period between attack detection and the VM owner's remedial efforts. Ideally, the falsified

information would be used to entice or entrap the attacker into performing certain actions in an effort to identify their purpose and attack strategy. The VM's behaviour is automatically transformed into that of a honeypot.

Obviously, this protection mechanism is not meant to replace any of the existing security controls within the VM. Potential implementations, applications, and limitations of this concept are out of this article's scope.

FIGURE 4



## CONCLUSION

There is likely to be an increase in the adoption of virtualisation due to the promising opportunities for system consolidation and cost-savings. It is well understood that traditional monitoring and protection systems are inadequate for meeting virtualisation's security needs. There are major limitations of these protection systems ranging from limited security guarantees to their performance impact.

Virtual machine introspection (VMI) technology is an important advance for monitoring virtualised environments. However, VMI technology is not as secure as it has been widely preached. VMI offers great protection against today's most commonly-faced attacks but its monitoring reliability is inadequate for protecting against emerging attacks. Although the attacks mentioned in this article are not widely deployed at the moment, it does not mean that virtualisation users can rest assured in their security.

The semantic gap is still a major topic of investigation. Research on system and memory analysis is being undertaken in an attempt to overcome this limitation. For the time being, widely deployed VMI technology cannot be considered entirely secure and reliable until the semantic gap has been completely bridged. ■

## ABOUT THE AUTHORS

*Fotios Tsifountidis* *After completing his M.Sc at Royal Holloway in 2010, Fotios Tsifountidis joined Hewlett-Packard Labs in Bristol and is currently involved in the Security Analytics team. His main focus is on the scientific analysis of security risks and associated trade-offs for providing sound, evidence-based decisions and cost-effective security investments.*

*Dr. Geraint Price* *is a Lecturer in Information Security at Royal Holloway. His research interests include secure protocols, public key infrastructures, denial-of-service attacks and resilient security.*