The relationship between virtualization and security is a complex one and has two distinct aspects: virtualization

for security and the security of virtualization [215]. In Section 5.1 we praised the virtues of

virtualization.We also discussed two of the problems associated with virtual environments: (a) the negative

effect on performance due to the additional overhead; and (b) the need for more powerful systems

to run multiple virtual machines. In this section we take a closer look at the security of virtualization.

One of the most important virtues of virtualization is that the complete state of an operating system

running under a virtual machine is captured by the VM. This state can be saved in a file and then the

file can be copied and shared. There are several useful implications regarding this fact:

1. Ability to support the IaaS delivery model. In this model a user selects an image matching the local

environment used by the application and then uploads and runs the application on the cloud using

this image.

2. Increased reliability. An operating system with all the applications running under it can be replicated

and switched to a hot standby7 in case of a system failure.

3. Straightforward mechanisms to implement resource management policies:

• To balance the load of a system, an OS and the applications running under it can be moved to

another server when the load on the current server exceeds a high-water mark.

• To reduce power consumption, the load of lightly loaded servers can be moved to other servers

and then these servers can be turned off or set on standby mode.

4. Improved intrusion prevention and detection. In a virtual environment a clone can look for known

patterns in system activity and detect intrusion. The operator can switch to a hot standby when

suspicious events are detected.

5. Secure logging and intrusion protection. Intrusion detection can be disabled and logging can be

modified by an intruder when implemented at the OS level. When these services are implemented

at the VMM/hypervisor layer, the services cannot be disabled or modified. In addition, the VMM

may be able to log only events of interest for a post-attack analysis.

6. More efficient and flexible software testing. Instead of a very large number of dedicated systems

running under different operating systems, different versions of each operating system, and different

patches for each version, virtualization allows the multitude of OS instances to share a small number

of physical systems.

Is there a price to pay for the benefits of virtualization outlined here? There is always the other side

of a coin, so we should not be surprised that the answer to this question is a resounding “yes.” In a

2005 paper [132] Garfinkel and Rosenblum argued that the serious implications of virtualization on

system security cannot be ignored. This theme was revisited in 2008 by Price [297], who reaches similar

conclusions.

A first type of undesirable effects of virtualization leads to the diminished ability of an organization

to manage its systems and track their status:

• The number of physical systems in the inventory of an organization is limited by cost, space, energy

consumption, and human support. Creating a VM reduces ultimately to copying a file; therefore the

explosion in the number of VMs is a fact of life. The only limitation for the number of VMs is the

amount of storage space available.

• In addition to quantity, there is also a qualitative aspect to the explosion in the number of VMs.

Traditionally, organizations install and maintain the same version of system software. In a virtual

environment such homogeneity cannot be enforced; thus, the number of different operating systems,

their versions, and the patch status of each version will be very diverse, and this heterogeneity will

tax the support team.

• Probably one of themost critical problems posed by virtualization is related to the software life cycle.

The traditional assumption is that the software life cycle is a straight line, so patch management is based on a monotonic forward progress. However, the virtual execution model maps to a tree

structure rather than a line; indeed, at any point in time multiple instances of the VM can be created

and then each one of them can be updated, different patches installed, and so on. This problem has

serious implications for security, as we shall see shortly.

Let us now concentrate our discussion on direct implications of virtualization on security. A first

question is: How can the support team deal with the consequences of an attack in a virtual environment?

Do we expect the infection from a computer virus or a worm to be less manageable in a virtual

environment? The surprising answer to this question is that an infection may last indefinitely.

Some of the infected VMs may be dormant at the time when the measures to clean up the systems

are taken and then, at a later time, they could wake up and infect other systems. This scenario can repeat

itself and guarantee that infection will last indefinitely. This is in stark contrast to the manner in which

an infection is treated in nonvirtual environments; once an infection is detected, the infected systems

are quarantined and then cleaned up. The systems will then behave normally until the next episode of

infection occurs.

The more general observation is that in a traditional computing environment a steady state can be

reached. In this steady state all systems are brought up to a “desirable” state, whereas “undesirable”

states – states in which some of the systems are either infected by a virus or display an undesirable pattern

of behavior – are only transient. This desirable state is reached by installing the latest version of the

system software and then applying the latest patches to all systems. Due to the lack of control, a virtual

environment may never reach such a steady state. In a nonvirtual environment the security can be compromised

when an infected laptop is connected to the network protected by a firewall or when a virus is

brought in on removable media. But unlike a virtual environment, the system can still reach a steady state.

A side effect of the ability to record in a file the complete state of a VMis the possibility to roll back a

VM. This opens wide the door for a new type of vulnerability caused by events recorded in the memory

of an attacker. Two such situations are discussed in [132]. The first is that one-time passwords are

transmitted in the clear and the protection is guaranteed only if the attacker does not have the possibility

to access passwords used in previous sessions. If a system runs the S/KEY password system8 an attacker

can replay rolled-back versions and access past sniffed passwords.

The second situation is related to the requirement of some cryptographic protocols, and even noncryptographic

protocols, regarding the “freshness” of the random-number source used for session keys

and nonces.9 This situation occurs when a VM is rolled back to a state in which a random number has

been generated but not yet used.

Even noncryptographic use of random numbers may be affected by the rollback scenario. For example,

the initial sequence number for a new TCP connection must be “fresh”; when it is not, the door to

TCP hijacking is left open.

Another undesirable effect of the virtual environment affects the trust. Recall from Section 9.4 that

trust is conditioned by the ability to guarantee the identity of entities involved. Each computer system

in a network has a unique physical, or MAC, address; the uniqueness of this address guarantees that an

infected or malicious system can be identified and then cleaned, shut down, or denied network access.

This process breaks down for virtual systems when VMs are created dynamically. Often, to avoid name

collision, a random MAC address is assigned to a new VM. The other effect discussed at length in

Section 9.8 is that popular VM images are shared by many users.

The ability to guarantee confidentiality of sensitive data is yet another pillar of security affected by

virtualization. Virtualization undermines the basic principle that the time-sensitive data stored on any

system should be reduced to a minimum. First, the owner has very limited control over where sensitive

data is stored; it could be spread across many servers and may be left on some of them indefinitely.

To be able to roll it back, a VMM records the state of a VM. This process allows an attacker to access

sensitive data the owner attempted to destroy.