MTR160066

MITRE TECHNICAL REPORT

# Resiliency Mitigations in Virtualized and Cloud Environments

Ellen Laderman
Ken Cox

March 2016

Sponsor: NSA
Dept. No.: J83C
Contract No.: WF56KGU-15-C-0010
Project No.: 0716N832-RA

The views, opinions and/or findings contained in this report are those of The MITRE Corporation and should not be construed as an official government position, policy, or decision, unless designated by other documentation.

**Bedford, MA**

.

## Approved By

Rosalie M. McQuaid
Department Head

Date

# Executive Summary

This paper presents an analytic approach to addressing risks associated with moving mission systems and applications to a virtualized or cloud environment, and applying cyber resiliency techniques as potential risk mitigation measures. That approach takes, as a starting point, the general risk model presented in National Institute of Standards and Technology (NIST) Special Publication (SP) 800-30R1 [1], focusing on adversarial threat sources. It uses an extensible set of threat event types [1] (called attack mechanisms in this document), a set of predisposing conditions inherent in virtualized or cloud environments, and a set of potential system effects which, if achieved by an adversary, will enable the adversary to create adverse impacts on missions. It provides a way to compare alternative design and technology options that might reduce risks and improve mission assurance. Because the likelihood and level of impact will depend on the specific mission(s), environments, and implementation, the analytic approach does not seek to evaluate potential risk reduction. Instead, the analytic approach looks at whether an alternative addresses predisposing conditions, reduces adverse effects, and is mature enough to consider for use in the target environment (in this case, a general virtualized or cloud environment).

This paper focuses on the additional risk due to the system being virtualized and/or migrated to the cloud. The mitigations mentioned in this document reduce the likelihood of an architectural factor being exploited and minimize the impact of exploitation; however, they do not eliminate all risks in the environment.

It is not feasible to apply all cyber resiliency techniques to an architecture. Based on the number of risks addressed, the relative maturity and readiness for cyber resiliency application, and the potential interactions between the techniques, seven cyber resiliency techniques with eleven associated approaches were selected to mitigate the exploitation of architectural factors in virtual machine (VM) and cloud environments as well as mitigate the effects if they are exploited.

# Acknowledgments

The authors would like to acknowledge the expert input from Lisa Mitchell, Daryl Baker, Don Faatz, and Deb Bodeau. The authors would also like to acknowledge the valuable review given by Don Faatz, Deb Bodeau, Richard Graubart, Jeff Picciotto, Beth Abramowitz, Lisa Ray and Dan Woycke.

# Table of Contents

## List of Figures

## List of Tables

This page intentionally left blank.

# 1 Introduction

Transitioning from a physical platform environment to a virtual cyber environment poses new challenges as well as opportunities in risk management. Moving a locally-hosted cyber environment to a cloud-hosted environment poses similar challenges and opportunities. While these types of transitions provide capabilities for reducing costs, incorporating redundancy, improving continuity of operations, and increasing cyber resiliency,[1] they also reduce the separation between systems,[2] the separation between environments, the capabilities for trusted insight into the systems and their environments, the hands-on management, and the control of systems and their environment. This document discusses the challenges posed by virtual and cloud environments, and how cyber resiliency techniques can increase mission assurance in environments whose architectures are based on virtual infrastructure and cloud services. Virtual environments are not the same as cloud environments, yet they frequently support cloud environments; therefore, the risks associated with virtual environments often must be considered in addressing the risks associated with cloud environments.

A virtual machine (VM) environment is one in which multiple guest operating systems (OSs) are hosted on a single physical platform. Each guest OS, together with a set of applications and data stores, is run on a separate VM that acts as a dedicated computer with its own network configuration and full suite of software. When the sensitivity levels of VMs on a single physical machine platform differ from each other, the risks posed by one VM to another and to the platform as a whole must be taken into account to adequately assess and mitigate risk. Broadly speaking, a VM may be at risk of losing its availability, integrity, and/or confidentiality caused by an attack on, or by, another VM that has been designed to support an environment with different risks.

Cloud computing and storage services provide the capability to store and process data in data centers. These data centers may be private or public – the key concept with regard to risk is that the data owner has outsourced control of, and therefore some degree of responsibility for, the data storage and processing platforms. In addition, cloud data centers are generally based on virtualized environments. The priorities of the cloud resource managers and contractual agreements regarding management of, use of, and access to the cloud environments are critical to consider when assessing and mitigating risk for these environments.

This document focuses on using cyber resiliency techniques to remediate the increase in risk due to transitioning to cloud and VM environments. While the risks due to traditional vulnerabilities are addressed here, it is only with respect to how the VM and cloud environments enable or expand the risks associated with these vulnerabilities. In general, both operational and technical mitigations may be employed to address risks in VM environments. In contrast, risks in cloud environments are mainly addressed by operational mitigations, such as agreements between cloud service providers and cloud service consumers. No detailed environment specifications are provided here. This document is intended as a guide to creating resiliency requirements for virtualized environments and cloud-based environments, to be applied in conjunction with knowledge of specific environment's objectives and conditions.

[1] VM and Cloud environments are useful in implementing Adaptive Response, Deception, Diversity, Dynamic Positioning, Non Persistence, and Unpredictability cyber resiliency techniques.

[2] In this document the term "system" refers both to an individual system as well as a system-of-systems.

The rest of this paper is organized as follows. Section 2 describes typical virtual and cloud environments. Section 3 describes the architectural factors specific to cloud and virtualized environments and potential adverse effects that can occur when these factors are leveraged in an attack. Section 4 discusses cyber resiliency techniques and approaches, and how they can be applied to the architectural factors found in cloud and virtual environments and the adverse effects that are specific to these environments.

# 2 Virtualized Environments and Resilience-related Features

In order to analyze the impact of virtualization on the cyber resilience of an environment, one must understand the primary components of a virtualized environment and how they are used in a typical application.

The first three parts of this section discuss the virtualized and cloud environments, specifically, virtualized hosting platforms, software defined networks, and cloud services. The final part of this section provides a use case to show how these three concepts can be assembled in an integrated solution, and highlights some of the design points that become resilience-related architectural factors discussed in Section 3.

## 2.1 Hosting Platforms

In a virtualized hosting platform, multiple VMs often share a single physical host computer, including its hardware and network connection(s). The sharing of physical resources is arbitrated by the hypervisor. This includes managing the processor, memory (e.g., random access memory [RAM]), and storage (e.g., disk) resources amongst the VMs.

Figure 1 shows a virtualized environment[3] modeled as six layers: guest applications, guest OSs, virtual machines, the hypervisor, physical hardware, and network. The hypervisor is an operating kernel implemented in software, firmware, or hardware that hosts multiple VMs and enables them to share the physical resources of the host system.

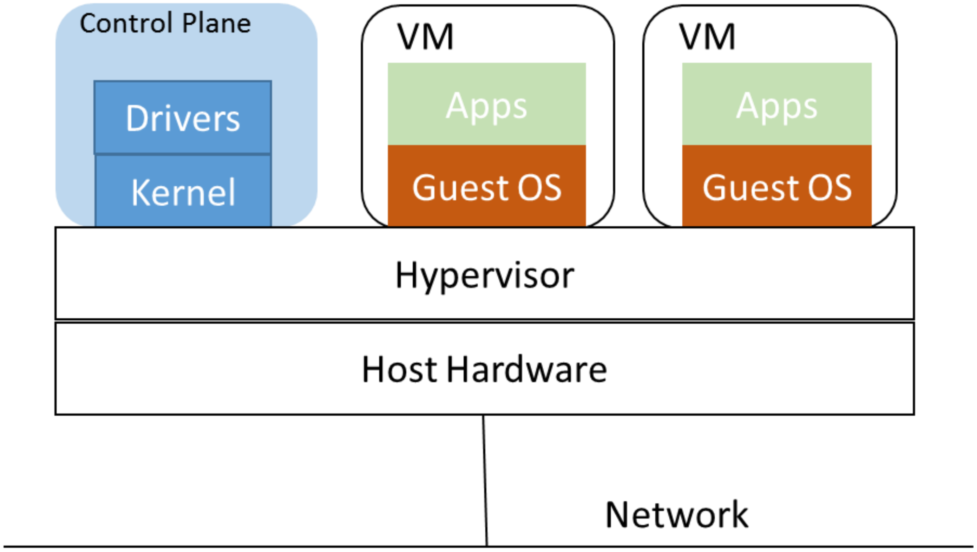


**Figure 1. Notional Virtual Environment Representation**

A virtualization solution will have a control domain that provides a management interface to the hypervisor and manages interfaces between the hypervisor, the device drivers, and other host resources.  In some vendor solutions, the control domain can be separate from the hypervisor

[3] This paper focuses on native, “bare-metal” (Type 1) hypervisors, which have direct access to the host hardware. Hosted (Type 2) hypervisors are those that run on top of the host operating system, so hypervisor access to the hardware goes through the host OS. Type 2 hypervisors are typically deployed only in non-production environments and are rarely subjected to resilience requirements.

(e.g., Citrix Xen's dom0, or Microsoft's Hyper-V control partition), in others, the control domain is integrated with the hypervisor (e.g., VMware ESXi). Guest VM access to the device drivers are typically managed directly by the hypervisor, but in the case of Xen, device drivers can be integrated with the hypervisor or managed in the control domain (e.g., to support Xen-aware guests).

The control domain is configured by the administrator to provide access to shared resources for guest VMs, and to inform the hypervisor in order to enforce sharing policies in accordance with defined service agreements that reflect mission priorities. The configuration of the control domain is trusted to ensure that VMs share resources fairly according to policy, and do not monopolize key resources, such as disk or memory channels, thereby starving other VMs from access to the shared resources.

The default implementation of most virtualization solutions use the assumption of "no direct sharing between VMs" of storage resources such as disk areas. For instance, if a virtual disk area is accessible by VM ***a***, only VM ***a*** can access that data. However, most solutions also support certain features that can be configured to allow for certain direct sharing of storage, arbitrated through the hypervisor. In that way, both VM ***a*** and VM ***b*** can access the same virtual disk storage area under certain conditions. While this can be a useful feature for certain implementations, it is also a potential covert channel available in a virtualized environment.

In summary, the hypervisor is responsible for enforcing the assigned policies for VM separation and fair use of the physical resources of the platform. Therefore, evaluating the trusted operation of the hypervisor and its control domain are essential components of a resiliency analysis.

## 2.2 Software-Defined Network Solutions

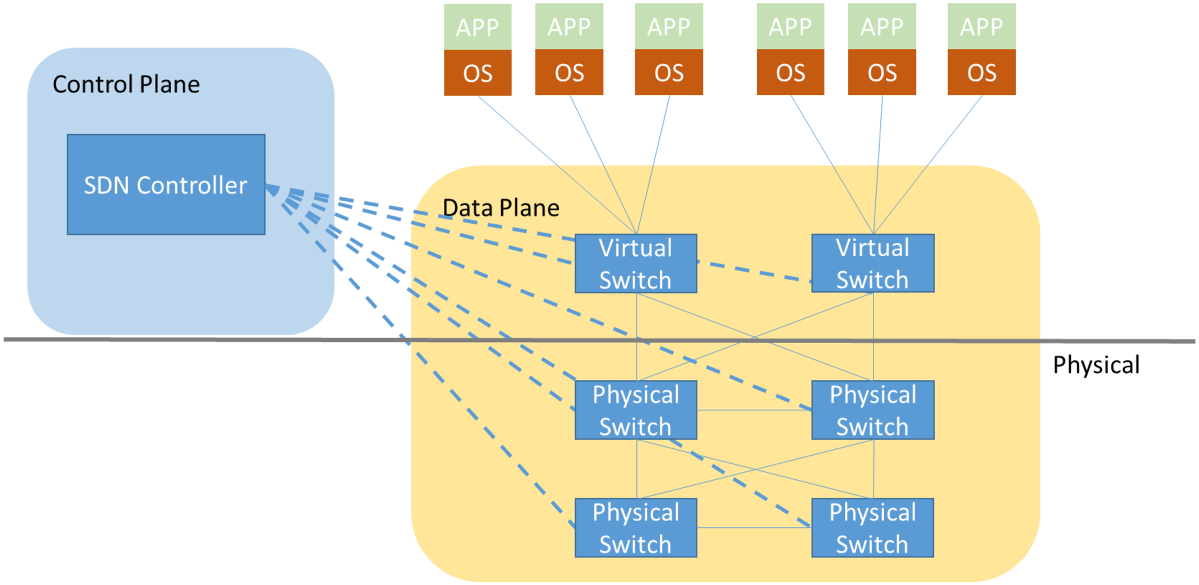
Software-defined networking (SDN) adds another layer of physical-to-virtual abstraction in the modern data center. The term SDN includes several types of network technology aimed at making networks agile and flexible.

The description here focuses on the general concepts of SDN rather than specific implementations or technology. The processing within network devices can be divided into two aspects:

- A network control plane that determines where and how to forward data traffic across the network.
- A data plane that carries the traffic by executing the switching, prioritization, and filtering rules defined by the control plane.

In the traditional network device model, both network control and data planes reside on an embedded network operating system (NOS) in each network device. Data packets go *through* the router or switch that contains decision logic local to that point in the network topology. The routers and switches use the configuration in their independent network control planes to determine how to process the incoming and outgoing packets on their data planes. As the number of routers and switches scale up in a multi-tenant data center environment, the burden of managing the individual configurations must also scale.

In contrast, SDN separates the network control plane from the data plane on the network device. Therefore, the control plane, as a separate software entity, can reside outside the platform hosting the data plane [2]. This allows for centralized management of the control logic and network policies, enabling flexibility in scaling the network configuration as well as some flexibility in the selection of platforms hosting the network control and data planes. A notional SDN representation is shown in Figure 2.



APP = Application

**Figure 2. Notional SDN Representation**

As in the case of virtualizing the hosting platform, the SDN model has centralized authority and broad scope of control granted to the control plane. Therefore, the control plane takes on remarkable importance in the resilience discussion.

## 2.3 Cloud Computing Considerations

An additional layer of abstraction and virtualization is introduced when cloud applications or storage services are incorporated in an information system architecture. The use of clouds is characterized by outsourcing an application or service layer to an external entity. Rather than acquiring and running the entire infrastructure in-house, certain operations (e.g., email, mass storage, or geospatial data services) are delegated to a cloud service provider through a service level agreement (SLA) or memorandum of agreement (MOA). This is depicted in Figure 3.

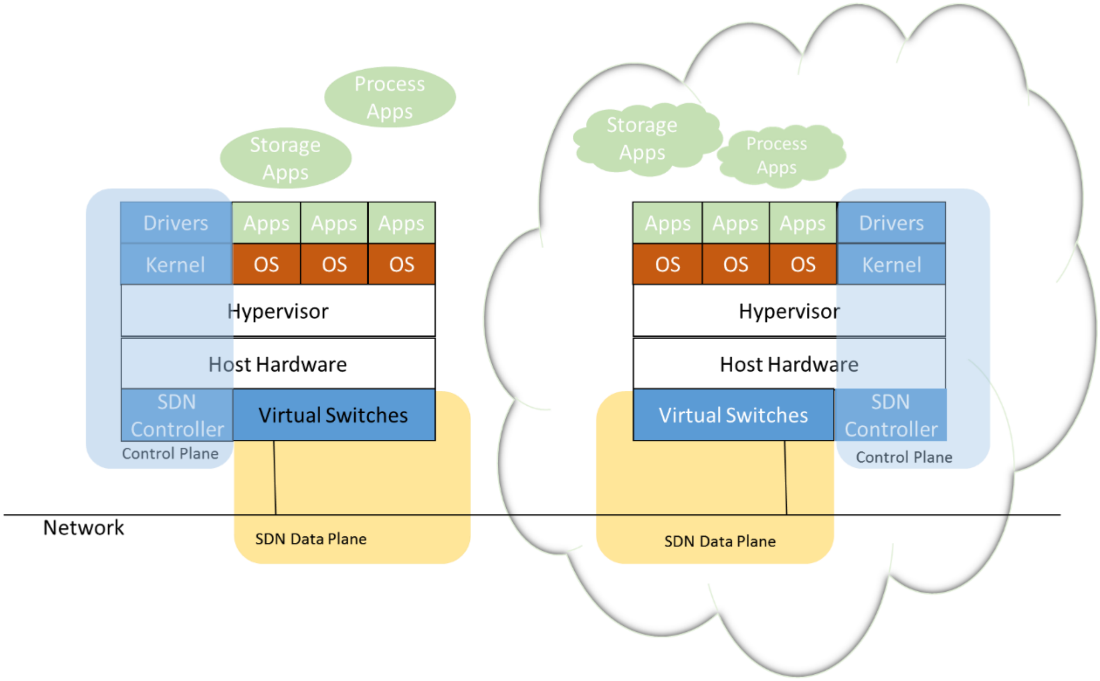


**Figure 3. Notional Cloud Environment Representation**

In many cases, the service provider will deliver the services via platforms based on VMs and SDN, so typically the outsourced services are delivered via resources shared with other customers using the services. Any system that shares resources, whether they are hardware platforms, applications, or services, may be impacted by other systems using those resources. This is true whether those resources are outsourced to another part of the organization or an outside provider. Data and control information flow across all layers, thus all layers need to be considered when mitigating risk in cloud environments. In the cloud model, the terms of the SLA or MOA are often the only control the customer has over the operation of the outsourced service.

## 2.4 Use Case: Web-based Database Application in VM/Cloud Environment

To better understand the resilience-related architectural factors discussed in Section 3, it may be helpful to follow the data flow in a hypothetical use case. In this section, we illustrate the use of a virtualized environment by describing the operation of a notional web-based database application that is a front-end to database repository hosted by a cloud service provider.

In Figure 4, the user workstation on the left accesses the application via web (hypertext transport protocol [HTTP]) services hosted in a virtualized environment in the data center. The application provides access control and user interface logic as a front-end service to the data store. The application will in turn access the database services via an application programming interface (API) such as Structured Query Language (SQL). This use case assumes that the notional application leverages an identity and authentication (I&A) service, such as Microsoft (MS) Active Directory for user access control, and all processes will rely on infrastructure services such as traffic routing, service name/address mapping (domain name system [DNS]), and time services (Network Time Protocol [NTP]).

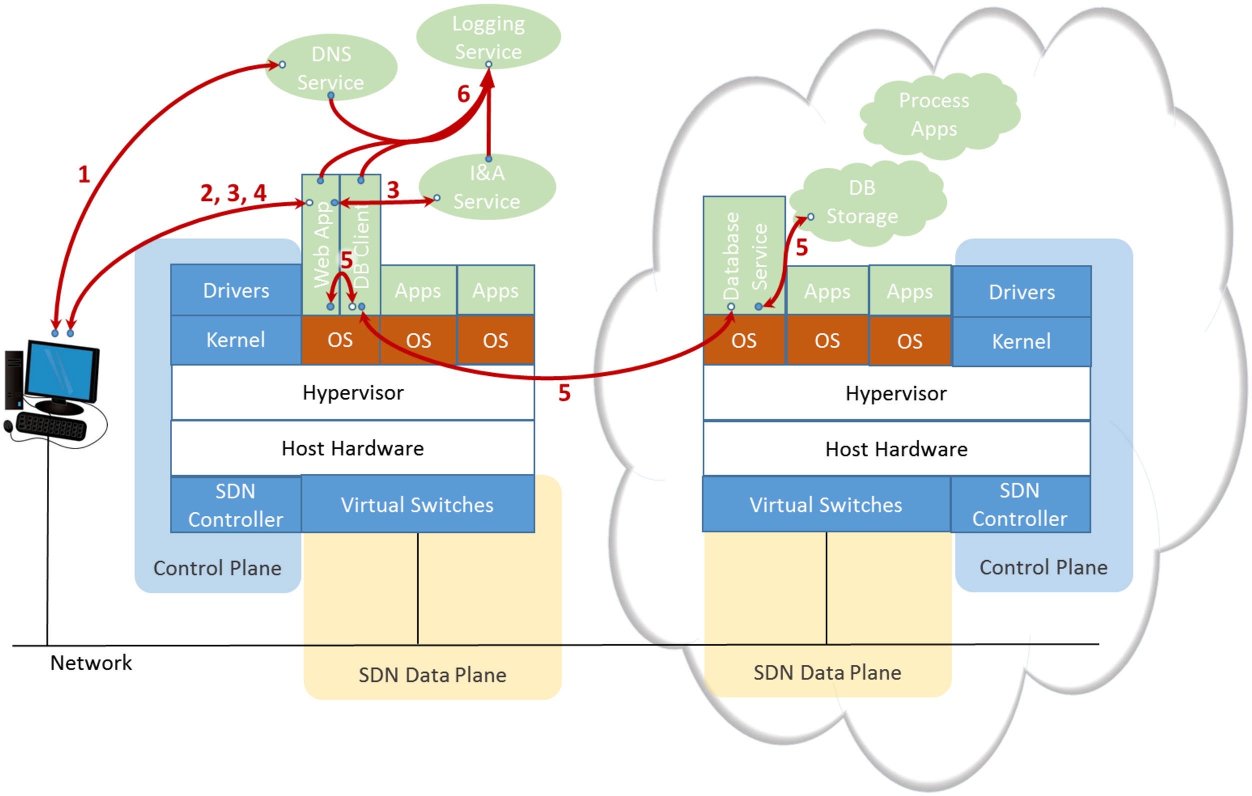


**Figure 4. Diagram for a Use Case with a Networked Environment with Virtual Machines**

The following is a path summary for a notional user session:

1. The user issues a request from a workstation browser to start and run an application session. This generates a DNS request for the location of the application website and the DNS server responds.
2. The workstation passes the session initiation request (via HTTP Secure [HTTPS]) to the application server process. The application responds with a Web page that requests the user to supply identification credentials.
3. The browser supplies the credentials and the application verifies them by issuing a request and getting a response from the I&A service.
4. The HTTPS session continues between the browser and the application as additional user interface requests are issued and responses received.
5. As the session proceeds, the application will need to read, create, or update records in the database repository via the following data flow:
   a. The application will establish and maintain a database connection to be shared amongst application users. This occurs at application initialization and is periodically refreshed. This connection is usually accomplished through a database client process that connects across the network to the defined database service provider. Depending on scale and load, this connection may in fact be a managed pool of connections.

   b. As the individual user’s session requires data services, the application process issues requests to its local database client process, which in turn passes the request to the defined database (DB) services via the established connection. The DB service executes the request and passes the response back through that connection.

6. Throughout this session, most processes and services are generating time-stamped event log records and perhaps forwarding them to logging servers.

The traces of the multiple requests and responses in the summary above show the many layers of infrastructure and software through which each request/response must pass. The attack points discussed in this paper that are present in physical machines are still present in the physical hardware supporting virtualized environments. In addition, there are attacks that are specific to the virtualized and cloud environments enabled by the sharing and outsourcing of resources. Section 3 will discuss the architectural factors and how they may be leveraged in attacks.

# 3 Architectural Factors and Adverse System Effects

The easiest, least expensive way to compromise a system, from an adversary's point of view, is where architectural factors create weak points in the environment. These include weaknesses caused by the architecture as well as traditional weaknesses that have greater impact in a cloud or virtual environment. Figure 5 provides a picture of how the adversary uses tools to create adverse effects in the environment and how cyber resiliency is used to mitigate both the adverse effects and the architectural factors that create weak points in the environment. The term "architectural factors" are specific predisposing conditions[4] found in VM and cloud environments that increase the likelihood that threat events result in adverse impact. This section has three parts. The first describes the types of architectural factors found in clouds and virtualized environments that cause these weak points. The second describes the adverse effects the adversary can create by leveraging these factors. The third section provides a summary of this information, setting the stage for Section 4 in which the application of cyber resiliency techniques to mitigate risks is discussed.

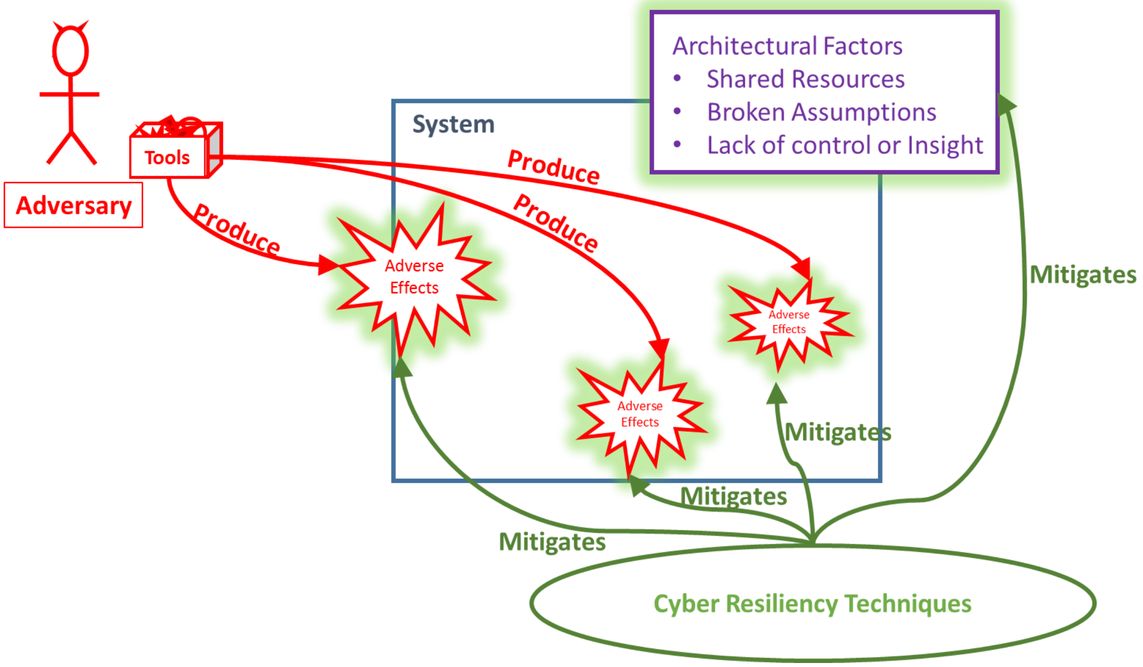


**Figure 5. The Adversary, the System and Cyber Resiliency**

## 3.1 Architectural Factors

When VMs are collocated on the same hardware, they are sharing resources and hence risk. This shared risk is due to the possibility of an external attack on one of the VMs, as well as the possibility of one VM being used to attack another VM or the host. It is important to take into account each layer and how much separation is needed. The VMs that are collocated should have

[4] A predisposing condition is defined as "a condition that exists within an organization, a mission or business process, enterprise architecture, information system, or environment of operation, which affects (i.e., increases or decreases) the likelihood that threat events, once initiated, result in adverse impacts to organizational operations and assets, individuals, other organizations, or the Nation." [17]

compatible risk tolerance (i.e., how much risk can be tolerated) and risk exposure (i.e., how much risk is generated by the VMs' activities). Each VM should have similar requirements for defense-in-depth and resiliency. One VM should not heighten the risk of attack (e.g., by being a honey pot) for another VM.

Applications and services migrated to a cloud may share risk with all systems using that cloud. As with VMs sharing a physical platform, when systems share applications and services in a cloud, it is important to take into account the amount of separation needed and the risk tolerance and exposure. Addressing the risks is made more complicated by the fact that the control of these resources has been outsourced to another part of the organization or an outside provider.

Since resources are shared in both virtualized environments and cloud environments, attacks that work in a traditional environment (i.e., on locally managed physical systems) may have greater impact in these environments affecting all systems that share the attacked resource. In addition, the hardware and software specific to virtualization (e.g., hypervisors (7) and SDN (9) in Figure 6) may also have vulnerabilities.

A model of a generic environment is shown in Figure 6. Each number in the figure identifies a component that can be compromised. When determining how to separate applications, functions, and guest systems, the risk tolerance and exposure should determine what mitigations to implement (e.g., end-to-end encryption) and at what layer they should be implemented (e.g., separation based on risk profiles). The control plane in this figure refers to both the network control plane as well as the control domain referred to in Section 2.1.

The architectural factors that cause or increase weak points in cloud and virtual environments can be grouped into three overlapping categories: resource sharing, broken assumptions, and lack of trusted insight. In the following descriptions, the numbers for the relevant components shown in Figure 6 will be referenced in parenthesis when relevant.

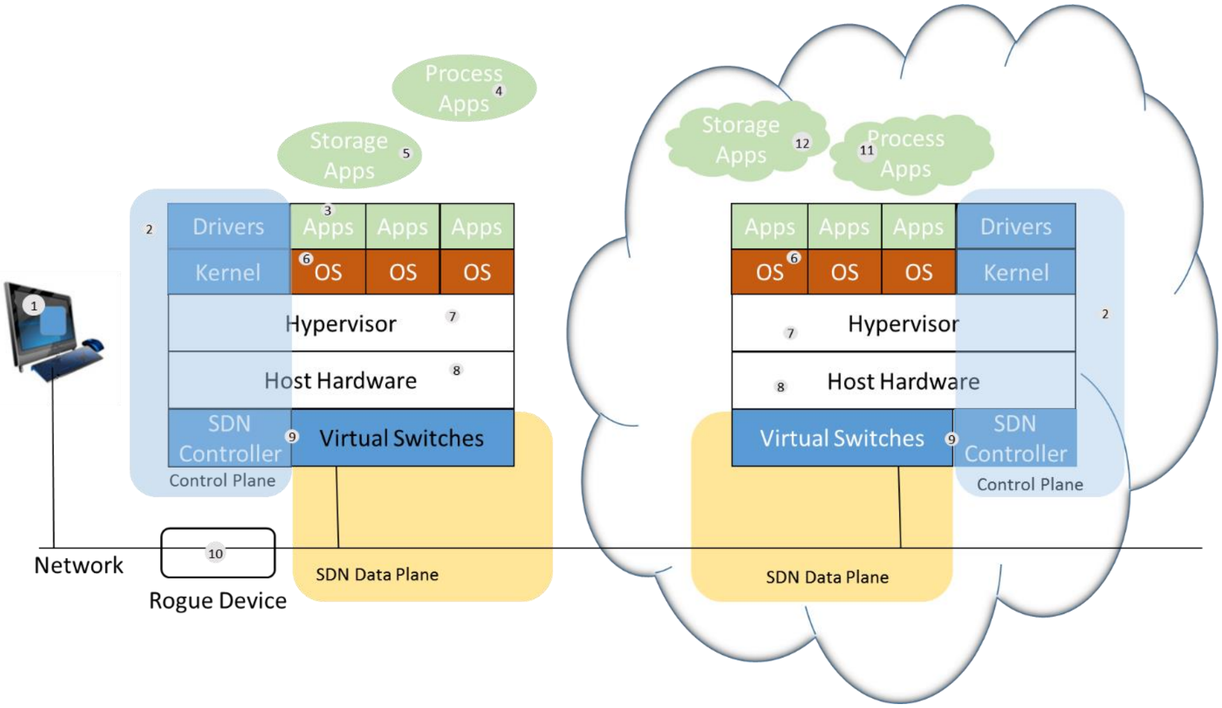


**Figure 6. Diagram of a Networked Environment with Virtual Machines**

### 3.1.1 Shared Resources

Attacks due to shared resources (e.g., covert channels consisting of storage channels and signaling channels between compromised VMs on a single hardware platform) can happen when processing applications (4), or storage systems (5), are shared among VMs on a single hardware platform, or when VMs from multiple hardware platforms share processing applications (11) and storage systems (12). Shared resources can also include device drivers and network connections (8) and SDN (9). These shared resources can be the attack vector by which malicious actors gain access to protected information, or deny or corrupt resources used by the targeted VM. In some cases, the impact of the attack is enhanced by the sharing of resources. Instead of only impacting one system, it might impact all VMs that share the attacked resource.

### 3.1.2 Broken Assumptions

When changes to the underlying architecture take place, implicit assumptions about isolation and control are frequently broken. In the case of virtual machines and cloud computing, several guest VMs are hosted on a single hardware platform and resources are shared not only across that platform (4 and 5), but also across multiple platforms (11 and 12). Assumptions about the control plane (2), permissions granted to access applications and storage, where the applications are hosted, where application data is stored, and how configurations are maintained (6), as well as who controls the network (9 and 10) and where it is physically, may all be broken. For example, those using cloud services might assume the service provider provides only locked down images and maintains compliance to specific standards while the service provider really only provides partially locked down images and does not provide updates and assumes that the service consumer's information technology team will validate and adjust the configuration for full compliance. In addition, implicit assumptions about data access and resource availability and separation may be broken when migrating a physical system to a virtual implementation.

### 3.1.3 Lack of Trusted Control or Insight

When the control plane and the network are rendered in software and support multiple VMs, there is the potential for a lack of trusted insight into how controls and networks are configured and administered. The system administrator maintaining the system may not be knowledgeable in network management and the tools to gain insight may not be available. While this combining of roles is another example of breaking assumptions as discussed in the previous paragraph, the problem is aggravated by a loss of visibility of some aspects of the computing environment due to changes such as virtualizations, shared resources, and lack of tools. Virtualized and cloud environments should have administrative roles defined to ensure separation of duties but nothing forces this. Due to the differences between local physical environments and virtualized cloud environments, traditional roles and skills may not directly translate into new environments. Administrative roles designed for a physical local environment may not ensure separation or duties in a virtualized cloud environment.

## 3.2 Adverse System Effects

A variety of attack mechanisms are unique to, or are made worse in, virtualized environments, whether on a single platform or in a cloud computing setting. The effects of those attack mechanisms can be roughly categorized as Full Control, Privilege Escalation, Unauthorized Information Sharing, and Configuration-Dependent. For each of these categories of effects, representative mechanisms that could produce those effects are described. These mechanisms will be used, along with the architectural factors that increase the likelihood of the mechanisms producing an impact, to organize the discussion on how to mitigate the risks.

The VM attack mechanisms discussed in this section, exploit, or are magnified by, the architectural factors described in Section 3.1. This subsection categorizes, by adversary effects, a representative set of 10 attack mechanisms drawn from [3] [4], provides examples of specific attacks, when applicable, and identifies the architectural factors leveraged. In some of the cases described below, traditional security techniques, such as patching and configuration management, will mitigate some risk. As stated in the introduction to this paper, the focus is the additional risk presented by the environment being virtualized and/or migrated to the cloud. While the mitigations mentioned below reduce the likelihood that the attack mechanism will be exploited or minimize the impact of exploitation, they do not eliminate the risk associated with the attack mechanism.

### 3.2.1 Full Control

There are three attack mechanisms that can provide the attacker with full control of the hypervisor and the VMs that reside on it. These are VM Escape, Device Drivers in Privileged Domains, and Hyperjacking.

The VM Escape mechanism is one in which an operating system residing in a VM encapsulation breaks out of the VM to interact directly with the hypervisor [3]. Using this mechanism, an attacker can abuse communication side channels, such as Chat and File Transfer Protocol that are shared among VMs or exploit vulnerabilities in device driver emulations that allow data to be written to higher privileged areas. This attack mechanism can result in the attacker having access to all VMs hosted on the hypervisor, and potentially the host machine, depending on privileges settings. Examples include: VMcat, VMChat, VM Drag-n-Sploit and VMftp [5], as well as VENOM [6], SNAFU [7], and one that is listed as CVE 2015-5154 and is yet to be named [8]. The VM Escape attack mechanism exploits the shared resources to expand the scope of the attack. Assumptions about separation and security standards may also be violated.

The Device Drivers in Privilege Domain attack mechanism is based on flaws in device drivers for hardware that can be used to gain malicious access to the control domain (e.g., “Dom 0”, the initial startup domain in a Xen Hypervisor [9]). By exploiting this mechanism, an attacker may gain access to the hypervisor control stack and be able to execute code of the attacker’s choice including changing device settings. The attacker may use bugs in the device drivers to exploit this attack mechanism. The architecture of the hypervisor itself is critical as this attack mechanism depends on an environment of shared resources.

Hyperjacking is an attack mechanism where a user can take malicious control over the hypervisor. This mechanism is due to a lack of separation between control flows and data flows, guest OS access to the hypervisor (e.g., via a management tool on the guest OS), or an unpatched system [10]. The exploitation of this mechanism can result in the attacker gaining unlimited access to the entire virtualization server and the guest VMs. This attack mechanism can result from poorly managed control and data flows as well as poorly managed shared access to resources.

### 3.2.2 Elevated Privilege

There are two attack mechanisms that an attacker can exploit to escalate privileges: Control and Management vulnerabilities, and Excessive Administrative Privilege vulnerabilities.

The Control and Management attack mechanism is one in which the hypervisor management – either the capabilities to manage the VMs on a single physical host or those management products intended to manage virtualization across multiple physical resources – are used as the attack vector [11]. The result is that the hypervisor is subverted to gain higher access privileges. The seriousness of this attack mechanism is dependent on the specific vulnerability leveraged. This attack mechanism uses the administrative environment and control channels between the guest VM and the administrative domain. It exploits a lack of control or insight into the control and data flows.

The Excessive Administrative Privilege attack mechanism can arise in a virtual infrastructure because the server administration, network administration, and security configuration and monitoring are all controlled through the single administrative interface that controls both VMs and the virtual network [12]. This single point of control may result in a virtualization administrator with excessive privileges and capability, who may not have the knowledge to appropriately administer all of the layers in the VM environment. The seriousness of exploiting this attack mechanism depends on the configuration, management policy, procedural controls on the administrator, and the administrator's knowledge level – both how well the administrator can manage the system and the level of damage the administrator can cause. This attack mechanism is based on broken assumptions (i.e., that the administration of these layers will be performed by different teams or individuals) as well as lack of insight into, or separation of, the data and control flows.

### 3.2.3 Unauthorized Sharing

Resource Pooling and Guest-to-Guest VM attack mechanisms are two mechanisms that an attacker can exploit to gain unauthorized access to information.

Using the Resource Pooling attack mechanism, one guest VM can impact another guest VM through shared resources [3]. This mechanism can be used to create a denial-of-service (DOS) if, for example, one guest VM uses too much memory thereby depriving other guest VMs of the memory they need. This attack mechanism can also result in unauthorized information sharing if information is not wiped from memory before releasing it. This attack mechanism is due to a failure to adequately separate resources used by individual guest VMs, both in setting limits on memory allocation and wiping shared memory space after it is used. This attack mechanism might be mitigated, to some extent, by controlling how much resources on a physical entity are oversubscribed; however, this assumes the underlying hypervisor resource management capabilities are not compromised. Examples of exploits of this attack mechanism are Flush and Reload attacks on Cache [13] and Transparent Page sharing in VMware [14]. This mechanism can succeed because the Guest VMs are on the same physical platform and are using shared resources.

The Guest-to-Guest VM attack mechanism arises when the virtual network switch, which the host implemented to share the network interface card, is used by one guest VM to attack another guest VM [3]. The result is that the malicious VM can intercept packets and cause them to be redirected, or can implement a man-in-the-middle attack. This attack mechanism can be exploited because packet capture or intrusion detection systems, which are external to the hardware host, cannot see what is going on at the hypervisor layer. This attack mechanism is based on shared resources and the lack of insight into, and management of, the data and control flows.

### 3.2.4 Configuration-Dependent Effects

Three additional attack mechanisms result in exploits whose impact depends on the specific environment and configuration. These are VM Sprawl, Host and Guest OS Vulnerabilities, and Unsecured VM Migration. These mechanisms are made more serious by the existence of VM aware malware – that is malware that can identify whether the system it is attacking is a hosted virtual machine. All three attack mechanisms can be mitigated to some extent by good configuration control and management practices.

The VM Sprawl attack mechanism arises when virtual systems are propagated in an uncontrolled manner [5]. This creates an environment where rogue machines can consume resources and bandwidth. These rogue systems can remain unpatched and unmonitored, and hence present new vulnerabilities. This situation occurs when forgotten systems are not cleaned up, or monitoring procedures and tools are inadequate and allow intentionally hidden rogue systems to remain hidden. The impact of this attack mechanism is dependent on the environment in which it is found, and is based on the shared resources in both VMs and cloud environments. The attack mechanism is based on a lack of insight into, or management of, data and control flows.

The vulnerabilities referred to in the Host and Guest OS Vulnerabilities attack mechanism are the same vulnerabilities that occur in operating systems on physical machines and may arise either in the administrative environment or the guest OS [10]. The results are similar to those found in a physical system; however, in a virtualized environment, these vulnerabilities may be leveraged to attack not only the OS in which it is found but the other OSs on the hardware. Both shared resources as well as a lack of insight or control exacerbate the impacts of this attack mechanism.

The Unsecured VM Migration attack mechanism arises when a VM is migrated to a new host, and security policies and configurations are not updated to reflect the change [10]. The result is that the VM host and guest OS running on that host, including any that were migrated, could become vulnerable to attack. A Xensploit proof-of-concept was demonstrated in 2008 [15]. This mechanism can be exploited because assumptions about the configuration and security policies are broken and the monitoring that would provide the control and insight into this is lacking. Unfortunately, this can be defined as a “traditional attack point” as security configurations and policies are not always monitored, enforced, or updated as the environment changes.

## 3.3 Summary of Adverse Effects

The architectural factors that enable the attack mechanisms discussed above, and the resulting adverse system effects, are summarized in Table 1. Shared resources and lack of control or insight have the most impact as measured by the number of mechanisms they enable. While broken assumptions do not enable as many mechanisms, they should not be ignored as the mechanisms they do enable may have serious adverse system effects (e.g., enabling an adversary to gain full control of a critical mission system or gain critical information and thereby degrade or derail the mission).

**Table 1. Adverse System Effects, Attack Mechanism and Architectural Factors**

| Adverse System Effects | Mechanism | Architectural Factors: | | |
|---|---|---|---|---|
| | | Shared Resources | Broken Assumptions[5] | Lack of Trusted Control or Insight[6] |
| **Full Control** | VM Escape: abuse of communication Side Channels | Yes | Yes | |
| | Device Drivers in Privileged Domain: use flaws in device drivers to gain malicious access to Dom0/privileged domain | Yes | | |
| | Hyperjacking: lack of separation between control and data flows; guest OS access to hypervisor; unpatched systems | Yes | | Yes |
| **Privilege Escalation** | Control and Management of Administrative VMs: Control Channel between guest VMs and the administrative domain | | | Yes |
| | Excessive Administrative Privilege: a single administrative interface may control both VMs and virtual network | | Yes | Yes |
| **Unauthorized Information Sharing** | Resource Pooling: failure to set limits on memory allocation; shared memory space not wiped after use | Yes | | |
| | Guest to Guest VM Vulnerabilities: No Packet capture or Intrusion Detection System (IDS) inspection; malicious actor can perform simple Address Resolution Protocol (ARP) poisoning to enable Man-in-the-Middle (M-I-M) attacks or packet redirection | Yes | | Yes |
| **Configuration-Dependent Effects** | VM Sprawl: forgotten machines that are not cleaned up; intentionally hidden rogue machines | Yes | | Yes |
| | Host and Guest OS Vulnerabilities: traditional vulnerabilities found in Guest OS, administrative environment, Dom0 or parent partition | Yes | | Yes [7] |
| | Unsecured VM Migration: security policies and configurations are not updated to reflect current conditions | | Yes | Yes |

[5] The details of broken assumptions are discussed in Section 3.1.2.

[6] Lack of Control or Insight refers to lack of trust due to a reliance on software to gain the control and insight and the lack of trust in this software. Section 3.1.3 discusses this further.

[7] This is categorized as lack of control and insight because the traditional vulnerability may exist since and the system can't be or isn't scanned due to its virtual nature.

# 4 Mitigating Risks Using Cyber Resiliency Techniques

The cyber resiliency techniques described in *Cyber Resiliency Engineering Aid* [16] are used in this section to mitigate the attack mechanisms described in Section 3. Cyber resiliency (also referred to as cyber resilience) can be defined as "the ability to anticipate, withstand, recover from, and adapt to adverse conditions, stresses, attacks, or compromises on cyber resources."[8] Subsection 4.1 summarizes the resiliency techniques and associated resiliency approaches presented in [16] that are most salient to the virtualization and cloud discussion.

It is not feasible to apply all cyber resiliency techniques to an architecture, so the system architect is compelled to select the most effective subset of those techniques while considering the impact on the overall system. Some considerations when selecting cyber resiliency techniques are: how the technique addresses the types of risks in the architecture under consideration, the relative maturity and readiness for cyber resiliency application, and the potential interactions between the techniques – both conflicting and synergistic. Further considerations include the effects on the adversary[9] and additional political, operational, economic and technical (POET) factors. It is not possible to adequately incorporate these further considerations without assessing the specific architecture and environment. Cyber resiliency techniques are focused on achieving one or more cyber resiliency objectives.[10] In addition, some techniques work better in certain types of architectures than others. For this reason, the discussion here is focused on applicability and will also discuss the maturity (i.e., usability) of, and the interactions between, the resiliency techniques but will not discuss the effects on the adversary or the POET factors.

Subsections 4.2 and 4.3 identify which cyber resiliency techniques address the architectural factors and adversary system effects discussed in Section 3. The fourth subsection looks at the relative maturity and readiness for cyber resiliency application of the techniques identified in 4.2 and 4.3. Recommendations are made in Section 4.5.

## 4.1 Cyber Resiliency Techniques and Approaches

Table 2 summarizes cyber resiliency techniques and the rationale for applying them (e.g., the objective an organization using it expects to achieve).

**Table 2. Cyber Resiliency Techniques**

| Cyber Resiliency Technique | Rationale |
|---|---|
| **Adaptive Response**: Implement nimble cyber courses of action to manage risks | Optimize the organization's ability to respond in a timely and appropriate manner to adverse conditions, stresses, or attacks, thus maximizing the ability to maintain mission operations, limit consequences, and avoid destabilization. |

[8] Cyber resources are "Separately manageable resources in cyberspace, including information in electronic form, as well as information systems, systems-of-systems, network infrastructures, shared services, and devices." - derived from NIST SP 800-39 [NIST 80039].

[9] Appendix A provides a brief discussion of the potential effects on the adversary using the cyber resiliency approaches described in this paper.

[10] Cyber resiliency objectives are described in [16].

| Cyber Resiliency Technique | Rationale |
|---|---|
| **Analytic Monitoring**: Gather, fuse, and analyze data on an ongoing basis and in a coordinated way to identify potential vulnerabilities, adverse conditions, stresses, or attacks, and damage | Maximize the organization's ability to detect potential adverse conditions, reveal the extent of adverse conditions, stresses, or attacks, and identify potential or actual damage. Provide data needed for cyber situational awareness. |
| **Coordinated Defense**: Manage multiple, distinct mechanisms in a non-disruptive or complementary way | Ensure that failure of a single defensive barrier does not expose critical assets to threat exposure. Require threat events to overcome multiple safeguards; in the case of adversarial events, this makes it more difficult for the adversary to successfully attack critical resources, increasing the cost to the adversary, and raising the likelihood of adversary detection. Ensure that uses of any given defensive mechanism do not create adverse unintended consequences by interfering with other defensive mechanisms. |
| **Deception**: Mislead, confuse, or hide critical assets from the adversary | Mislead or confuse the adversary, or hide critical assets from the adversary, making them uncertain how to proceed, delaying the effect of their attack, increasing the risk to them of being discovered, causing them to misdirect or waste their attack and expose their tradecraft prematurely. |
| **Diversity**: Use heterogeneity to minimize common mode failures, particularly attacks exploiting common vulnerabilities | Limit the possibility of a collapse of critical functions due to failure of replicated common components. In the case of adversarial threats, cause the adversary to work harder by developing malware or other Tactics, Techniques, and Procedures (TTPs) appropriate for multiple targets, increase the chance that the adversary will waste or expose TTPs by applying them to targets for which they are inappropriate, and maximize the chance that some of the defending organization's system's will survive the adversary's attack. |
| **Dynamic Positioning**: Distribute and dynamically relocate functionality or assets | Increase the ability of an organization to rapidly recover from non-adversarial events (e.g., fires). Impede an adversary's ability to locate, eliminate or corrupt mission/business assets, and cause the adversary to spend more time and effort to find the organization's critical assets, thereby increasing the chance of the adversary revealing their actions and tradecraft prematurely. |
| **Dynamic Representation**: Construct and maintain current representations of mission posture in light of cyber events and cyber courses of action | Support situational awareness, enhance understanding dependencies among cyber and non-cyber resources, reveal patterns/trends in adversary behavior; and validate the realism of courses of action. |
| **Non-Persistence**: Generate and retain resources as needed or for a limited time | Reduce exposure to corruption, modification or compromise. Provide a means of curtailing an adversary's advance and potentially expunging an adversary's foothold from in the system. |
| **Privilege Restriction**: Restrict privileges required to use cyber resources, and privileges assigned to users and cyber entities, based on the type(s) and degree(s) of criticality | Limit the impact and probability that unintended actions by authorized individuals will compromise information or services. Impede the adversary by requiring them to invest more time and effort in obtaining credentials; curtail the adversary's ability to take full advantage of credentials that they have obtained. |
| **Realignment**: Align cyber resources with core aspects of mission/business functions | Minimize the connections between mission critical and non-critical services, thus reducing likelihood that a failure of non-critical services will impact mission critical services. Reduce the attack surface of the defending organization by minimizing the chance that non-mission/business functions could be used as an attack vector. |

| Cyber Resiliency Technique | Rationale |
|---|---|
| **Redundancy**: Provide multiple protected instances of critical resources | Reduce the consequences of loss of information or services; facilitate recovery from the effects of an adverse cyber event; limit the time during which critical services are denied or limited. |
| **Segmentation/Isolation**: Define and separate (logically or physically) components on the basis of criticality and trustworthiness | Contain adversary activities and non-adversarial stresses (e.g., fires) to the enclave/segment in which they have established a presence; for adversarial cyber activities this limits the number of possible targets to which malware can easily be propagated. |
| **Substantiated Integrity**: Ascertain whether critical services, information stores, information streams, and components have been corrupted | Facilitate determination of correct results in case of conflicts between diverse services or inputs. Detect attempts by an adversary to deliver compromised data, software, or hardware, as well as successful modification or fabrication; provide limited capabilities for repair. |
| **Unpredictability**: Make changes randomly or unpredictable | Increase the adversary's uncertainty regarding the cyber defenses that they may encounter, thus making it more difficult for them to ascertain the appropriate course of action. |

Cyber resiliency “approaches” are specific ways to implement cyber resiliency techniques. For the above resiliency techniques, the Cyber Resiliency Engineering Framework (CREF) [16] defines 44 representative approaches to implementing those techniques. Table 3 provides the CREF definitions for selected resiliency approaches that address the concerns relevant to cloud and VM environments raised in Section 3.

**Table 3. Definitions of Cyber Resiliency Approaches Relevant to Virtualized Environments**

| Cyber Resiliency Technique | Cyber Resiliency Approach | Definition |
|---|---|---|
| **Analytic Monitoring** | Monitoring and Damage Assessment | Monitor and analyze behavior and characteristics of components and resources to look for indicators of adversary activity, detect and assess damage, and watch for adversary activities during recovery and evolution. |
| | Sensor Fusion and Analysis | Fuse and analyze monitoring data and preliminary analysis results from different components, together with externally provided threat intelligence. |
| **Coordinated Defense** | Technical Defense-in-Depth | Use multiple protective mechanisms at different architectural layers or locations. |
| | Coordination and Consistency Analysis | Apply processes, supported by analytic tools, to ensure that defenses are applied and cyber courses of action are defined and executed in a coordinated, consistent way that minimizes interference. |
| **Diversity** | Architectural Diversity | Use multiple sets of technical standards, different technologies, and different architectural patterns. |
| | Design Diversity | Use different designs to meet the same requirements or provide equivalent functionality. |
| | Synthetic Diversity | Transform implementations to produce a variety of instances. |

| Cyber Resiliency Technique | Cyber Resiliency Approach | Definition |
|---|---|---|
| **Dynamic Representation** | Dynamic Mapping & Profiling | Maintain current information about resources, their status, and their connectivity. |
| | Mission Dependency & Status Visualization | Maintain current information about mission dependencies on resources, and the status of those resources with respect to threats. |
| **Non Persistence** | Non Persistent Information | Refresh information periodically, or generate information on demand, and delete the information when no longer needed. |
| | Non Persistent Services | Refresh services periodically, or generate services on demand and terminate services after completion of a request. |
| **Privilege Restriction** | Privilege Management | Define, assign, and maintain privileges associated with end users and cyber entities, based on established trust criteria, consistent with principles of least privilege. |
| | Privilege-Based Usage Restrictions | Define, assign, maintain and apply usage restrictions on cyber resources based on mission criticality and other attributes. |
| **Realignment** | Purposing | Ensure cyber resources are used consistent with critical mission purposes. |
| | Restriction | Remove or disable unneeded risky functionality or connectivity, or add mechanisms to reduce the risk. |
| **Redundancy** | Surplus Capacity | Maintain extra capacity for information storage, processing, and/or communications. |
| **Segmentation** | Predefined Segmentation | Define and separate components on the basis of criticality and trustworthiness. |
| **Substantiated Integrity** | Behavior Validation | Validate the behavior of a system, service, or device against defined or emergent criteria. |

## 4.2 Cyber Resiliency Techniques and Architectural Factors

Table 4 lists the cyber resiliency techniques that reduce the likelihood the architectural factors described in Section 3.1 will be exploited or reduce the impact if a factor is exploited. The table also provides a description of how specific approaches to cyber resiliency techniques mitigate the factors.

**Table 4. Cyber Resiliency Techniques Applicability Based on Architectural Factors**

| Architectural Factors | Cyber Resiliency Technique | Approach Applicability |
|---|---|---|
| **Shared Resources** | Non Persistence | *Non Persistent Information* flushes information from memory and storage before it is reused by another resource, and thus prevents one resource from accessing information left in memory or storage by another resource. |
| | Privilege Restriction | *Privilege-Based Usage Restrictions* restricts access and use of critical resources based on privilege mitigating the possibility of an entity (e.g., guest OS) taking unauthorized control of a critical asset. |

| Architectural Factors | Cyber Resiliency Technique | Approach Applicability |
|---|---|---|
| | Redundancy | *Surplus Capacity* maintains extra capacity for information storage, processing, and/or communications prevents one guest OS from causing a DOS for another guest OS. |
| | Segmentation | *Predefined Segmentation* allows defenders to protect and isolate resources, as necessary (e.g., defining and enforcing which VMs can share a physical platform and associated resources). |
| **Broken Assumptions** | Privilege Restriction | *Privilege Management* helps ensure established trust criteria are maintained in the face of changes in the physical environment and how systems are managed.<br>*Privilege-Based Usage Restrictions* strengthens the basis for logical isolation and resource usage that were originally implemented physically. |
| | Realignment | Both *Purposing* and *Restriction* approaches ensure that the risk profiles (both what risk levels can be tolerated and what types of threats are attracted to the environment) of all systems sharing the resource are compatible with each other. |
| | Substantiated Integrity | *Behavior Validation* increases the likelihood that if the behavior of a system deviates from assumed norms the defenders will be made aware of the issue. |
| **Lack of Trusted Control or Insight** | Analytic Monitoring | *Monitoring and Damage Assessment* increases the likelihood of detecting adversarial behaviors by ensuring that data such as IDS, sensors, and event logs are appropriately managed in the increased complexity and layers associated with cloud and VM environments.<br>*Sensor Fusion and Analysis* provides tools that can be tailored to the layers and complexity in the cloud and VM environments, increasing accuracy of, and trust in, alerts. |
| | Coordinated Defense | *Technical Defense-in-Depth* provides support for the *Monitoring and Damage Assessment*.<br>*Coordination and Consistency Analysis* provides increased trust and coordination in control guest and host configuration and management. |
| | Dynamic Representation | *Dynamic Mapping and Profiling* can detect software and components that do not conform to policy or that are behaving in unexpected ways.<br>*Mission Dependency and Status Visualization* can increase insight by identifying consequences of adversarial actions as they occur. |
| | Substantiated Integrity | *Behavior Validation* can help identify rogue virtual systems, services, or devices. |

## 4.3 Cyber Resiliency Techniques and Adverse System Effects

Table 5 lists the cyber resiliency techniques that provide mitigation against each of the adverse system effects described in Section 3.2. The table also provides a description of how each technique mitigates the effect.

**Table 5. Cyber Resiliency Techniques Applicability Based on Adversarial Effect**

| Adversarial Effect | Cyber Resiliency Technique | Applicability |
|---|---|---|
| **Full Control** | Analytic Monitoring | *Monitoring and Damage Assessment* can increase the probability that attacks will be identified earlier in the cyber attack lifecycle[11] before the adversary attains full control.<br>*Sensor Fusion and Analysis* provides tools that can be tailored to the layers and complexity in the cloud and VM environments, increasing accuracy of, and trust in, alerts. This increases the probability that attacks will be identified earlier in the cyber attack lifecycle. |
| | Coordinated Defense | *Coordination and Consistency Analysis* ensures that defenses are applied and cyber courses of action are defined and executed in a coordinated, consistent way that makes it more likely defenders will prevent the adversary from gaining full control of a system. |
| | Diversity | *Architectural Diversity/ Heterogeneity*, *Design Diversity/ Heterogeneity* and *Synthetic Diversity* increases the difficulty of targeting a specific system (the attacker does not know the specifics of the system). This reduces the likelihood that the adversary's tactics will be completely successful thereby reducing the probability the adversary will gain full control of the system. |
| | Non Persistence | *Non Persistent Information* flushes information both preventing the adversary from using information from an OS the adversary does not control as well as flushing any information that the adversary placed in memory.<br>*Non Persistent Services* refreshes or resets services so that any attempts to use malware the adversary has inserted into those services to gain full control of a system is degraded or derailed. |
| | Segmentation | *Predefined Segmentation* reduces the ability of an attack to spread between platforms and throughout cloud environments. |
| **Privilege Escalation** | Analytic Monitoring | Both *Monitoring and Damage Assessment* and *Sensor Fusion and Analysis* approaches increase the likelihood of detecting indications of adversary activities earlier in the lifecycle thereby reducing the likelihood of a privilege escalation or minimizing the effects of exploiting this tactic. |
| | Diversity | *Architectural Diversity/Heterogeneity*, *Design Diversity/Heterogeneity* and *Synthetic Diver*sity increases the difficulty of targeting a specific system (the attacker does not know the specifics of the system). This reduces the likelihood that the adversary's tactics will be completely successful thereby reducing the probability the adversary will be able to execute a privilege execution on a targeted system. |
| | Privilege Restriction | *Privilege Management* helps ensure established trust criteria are maintained thereby minimizing privilege escalation.<br>*Privilege-Based Usage Restrictions* restricts access and use of critical resources based on privilege mitigating the possibility of an entity (e.g., guest OS) successfully escalating privilege. |
| | Non Persistence | *Non Persistent Information* flushes any information that the adversary placed in memory that might be used in a privilege escalation attack.<br>*Non Persistent Services* refreshes or resets services so that any attempts to use malware the adversary has inserted into those services to escalate privileges is degraded or derailed. |

[11] More information on the Cyber Attack Lifecycle is provided in Appendix A.

| Adversarial Effect | Cyber Resiliency Technique | Applicability |
|---|---|---|
| | Segmentation | *Predefined Segmentation* reduces the ability of a privilege escalation attack to spread between segmented areas. |
| **Unauthorized Sharing** | Non Persistence | *Non Persistent Information* flushes information from memory before it can be reused by another resource.<br>*Non Persistent Services* refreshes or resets services so that data associated with those services is not shared. |
| | Privilege Restriction | *Privilege Management* helps ensure established trust criteria are maintained thereby minimizing unauthorized sharing.<br>*Privilege-Based Usage Restrictions* reduces unauthorized sharing by restricting access and use of critical resources based on privilege mitigating the possibility of an entity (e.g., guest OS) obtaining unauthorized access. |
| | Realignment | *Purposing* ensures that cyber resources are used in a manner consistent with mission purposes thereby reducing the incidence of unauthorized sharing.<br>*Restriction* removes or disables unneeded risky functionality or connectivity thereby reducing the incidence of unauthorized sharing. |
| | Segmentation | *Predefined Segmentation* reduces the probability that unauthorized sharing will occur by limiting the connectivity between segmented areas. |
| **Configuration-Dependent Effects** | Non Persistence | *Non Persistent Information* flushes information from memory before it is reused by another resource thereby reducing the probability that the information can be misused by an attacker.<br>*Non Persistent Services* refreshes or resets services so that any attempts to use malware the adversary has inserted into those is degraded or derailed. |
| | Privilege Restriction | *Privilege Management* minimizes access to resources based on established trust criteria thereby reducing access by attackers.<br>*Privilege-Based Usage Restrictions* minimizes access to critical resources thereby reducing the risk of unauthorized usage of those resources. |
| | Segmentation | *Predefined Segmentation* reduces the probability that unauthorized sharing will occur by segmenting what areas can share resources thereby reducing the probability of the information being misused by an attacker. |

## 4.4 Maturity and Readiness for Adoption of Cyber Resiliency Techniques[12]

The maturity level of the cyber resiliency techniques and approaches are variable. The relative maturity is directly related to how easily a technique or approach can be integrated into a system or mission architecture. A related, but somewhat distinct consideration is how readily the techniques or approaches can be adopted for cyber resiliency. The relative maturity or readiness for adoption, of a technique or approach, is independent of its relative effectiveness. The relative maturity and readiness for adoption do, however, directly impact the usability of a technique. One could have a fairly immature technique or approach that could also be highly effective against the Advanced Persistent Threat (APT). Incorporating and maintaining such a technique/approach into a system would likely require considerable time, resources, and staff expertise, and for some organizations that may not be feasible. In these situations, selecting less effective, but more mature and adoption-ready techniques/approaches would be a better approach.

[12] The content in Subsection 4.4 is derived from [16].

Figure 7 depicts the various cyber resiliency approaches listed in Tables 4 and 5 relative to their maturity and readiness for adaption for support of cyber resiliency. As the figure shows, not all approaches that support a given resiliency technique are of the same maturity or readiness to adopt. This information will be one of the factors used in Section 4.5 when choosing which approaches should be considered in deciding how to best mitigate the enabling weaknesses and adverse system effects.

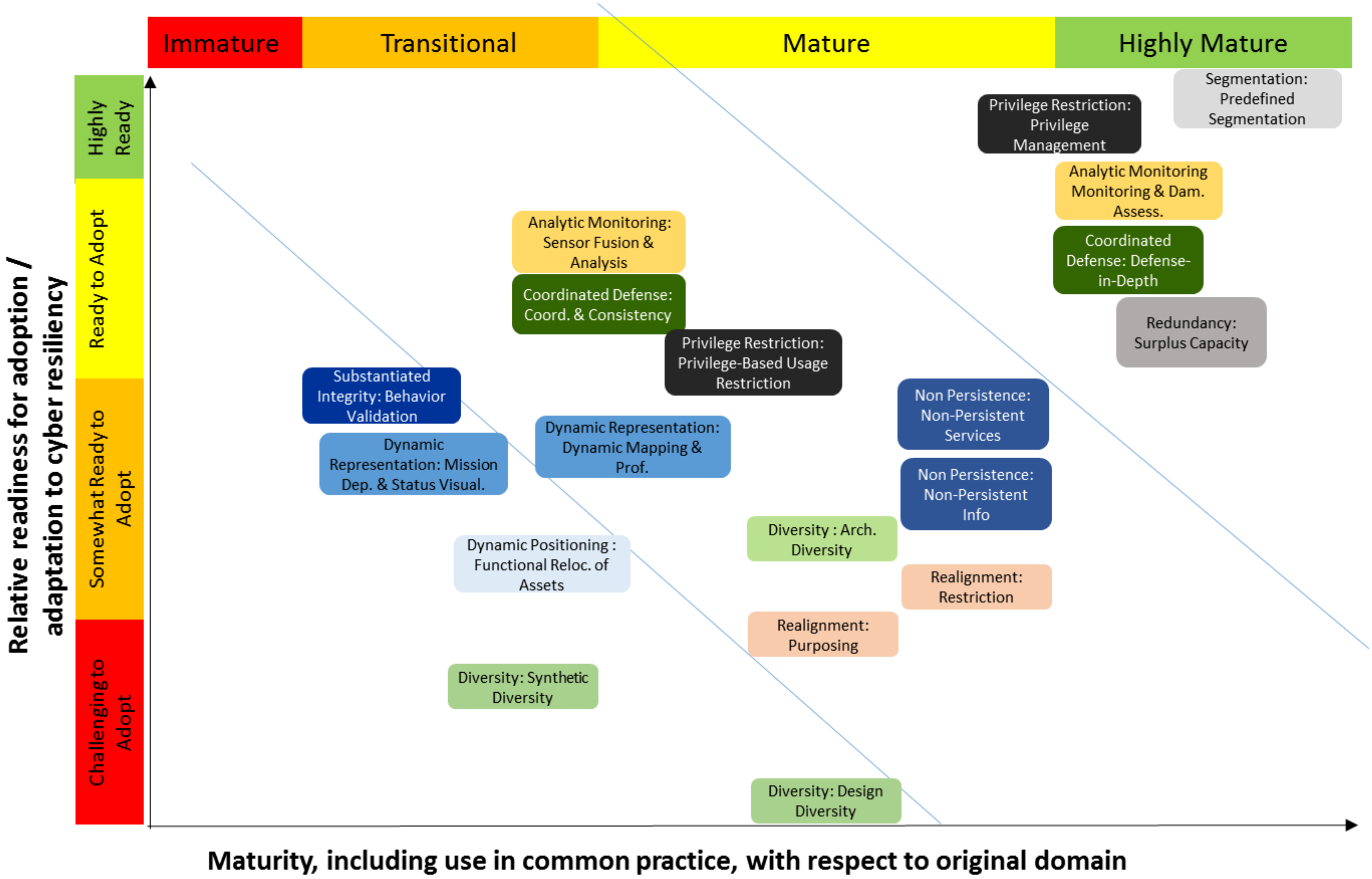

Immature
Transitional
Mature
Highly Mature
Highly Ready
Ready to Adopt
Somewhat Ready to Adopt
Challenging to Adopt
Relative readiness for adoption / adaptation to cyber resiliency
Segmentation: Predefined Segmentation
Privilege Restriction: Privilege Management
Analytic Monitoring Monitoring & Dam. Assess.
Coordinated Defense: Defense-in-Depth
Redundancy: Surplus Capacity
Analytic Monitoring: Sensor Fusion & Analysis
Coordinated Defense: Coord. & Consistency
Privilege Restriction: Privilege-Based Usage Restriction
Substantiated Integrity: Behavior Validation
Dynamic Representation: Mission Dep. & Status Visual.
Dynamic Representation: Dynamic Mapping & Prof.
Non Persistence: Non-Persistent Services
Non Persistence: Non-Persistent Info
Diversity : Arch. Diversity
Dynamic Positioning : Functional Reloc. of Assets
Realignment: Restriction
Realignment: Purposing
Diversity: Synthetic Diversity
Diversity: Design Diversity
Maturity, including use in common practice, with respect to original domain

**Figure 7. Relative Maturity and Ease of Adoption for Approaches to Implementing Cyber Resiliency**

## 4.5 Recommended Cyber Resiliency Techniques for the Virtualized and Cloud Environments

This section provides a notional set of recommendations for cyber resiliency approaches to implement in cloud and virtual environments. As discussed earlier in this paper, selection of these recommended approaches can be guided by:

- The number of architectural factors the approach addresses.
- The number of the adverse effects the approach mitigates.
- The relative maturity and adaptability of the approach.

The items that appear most frequently in the set of resiliency techniques and approaches that are applicable to architectural factors and adverse effects (Tables 4 and 5) are:

- Analytic Monitoring: Monitoring & Damage Assessment
- Analytic Monitoring: Sensor Fusion & Analysis
- Non Persistence: Non Persistent Information
- Non Persistence: Non Persistent Services
- Privilege Restriction: Privilege-Based Usage Restrictions
- Privilege Restriction: Privilege Management
- Segmentation: Predefined Segmentation

Each of these approaches address at least three architectural factors or adverse effects (i.e., the number of architectural factors plus the number of adverse effects they address is at least three).

The set of approaches that appears next most frequently in Tables 4 and 5 (i.e., the cases in which the number of architectural factors plus the number of adverse effects they address is equal to two) are:

- Coordinated Defense: Coordination and Consistency Analysis
- Diversity: Architectural Diversity
- Diversity: Design Diversity
- Diversity: Synthetic Diversity
- Realignment: Purposing
- Realignment: Restriction
- Substantiated Integrity: Behavior Validation

Cyber resiliency approaches do not exist in isolation from each other. Sometimes these approaches can support one another and sometimes they can complicate each other's implementation.[13] For example, the approaches that address concerns within cloud and VM environments, both Coordinated Defense: Technical Defense-in-Depth and Substantiated Integrity: Behavior Validation support Analytic Monitoring: Monitoring & Damage Assessment.

When the approaches' relative maturity and ease of adaptation (shown in Figure 3) are taken into account, along with the synergistic relationship of the approaches discussed above, a second tier of approaches to consider emerges. The approaches in this second tier are:

- Coordinated Defense: Technical Defense-in-Depth

[13] Reference [16] provides a description of which techniques and approaches support, use, complicate and conflict with other techniques and approaches.

- Coordinated Defense: Coordination and Consistency Analysis
- Realignment: Restriction
- Substantiated Integrity: Behavior Validation

A full cyber resiliency assessment of a specific mission system and its environment would evaluate how the recommended cyber resiliency approaches affect the adversary across the cyber attack lifecycle. Because this paper focuses on a notional environment without a specific mission, that discussion would not lead to practical results. However, we encourage the reader to perform that analysis when considering a specific mission system and threat environment. To assist in that evaluation, a discussion of the cyber attack lifecycle and a prototype chart of effects on the adversary's attack are included in Appendix A.

Given the discussion above, a summary list of recommended approaches for virtualized environments includes:

- Analytic Monitoring: Monitoring & Damage Assessment
- Analytic Monitoring: Sensor Fusion & Analysis
- Non Persistence: Non Persistent Information
- Non Persistence: Non Persistent Services
- Privilege Restriction: Privilege-Based Usage Restrictions
- Privilege Restriction: Privilege Management
- Segmentation: Predefined Segmentation
- Coordinated Defense: Technical Defense-in-Depth
- Coordinated Defense: Coordination and Consistency Analysis
- Realignment: Restriction
- Substantiated Integrity: Behavior Validation

The first seven approaches are roughly equivalent in their ranking as are the last four. It is important to remember that these are just initial recommendations. This list is an initial set of approaches—not the definitive set—that may be applicable and effective in specific VM and cloud environments. For specific situations, some approaches may be better than others.

Tables 6 and 7 provide a summary of how those approaches address the architectural factors and adverse system effects, respectively.

**Table 6. Cyber Resiliency Approaches for Mitigating Architectural Factors**

| Architectural Factor | Selected Cyber Resiliency Approaches for Mitigation |
|---|---|
| **Shared Resources occurs when: processing applications or storage systems are shared among VMs on a single hardware platform, or VMs from multiple hardware platforms share processing applications and storage systems.** | *Non Persistence: Non Persistent Information* prevents one resource from accessing information left in memory by another resource. |
| | *Privilege Restriction: Privilege-Based Usage Restrictions* mitigates the possibility of an entity (e.g., guest OS) taking unauthorized control of a critical asset. |
| | *Segmentation: Predefined Segmentation* allows defenders to protect and isolate resources, as necessary. |
| **Broken Assumptions: caused when changes to the** | *Privilege Restriction: Privilege Management* helps ensure established trust criteria are maintained. |

| Architectural Factor | Selected Cyber Resiliency Approaches for Mitigation |
|---|---|
| **underlying architecture take place causing implicit assumptions about isolation and control to be broken.** | *Privilege Restriction: Privilege-Based Usage Restrictions* strengthens the basis for logical isolation and resource usage that were originally implemented physically. |
| | *Realignment: Restriction* ensures the risk profiles (both what risk levels can be tolerated and what types of threats are attracted to the environment) of all systems sharing a resource are compatible with each other. |
| | *Substantiated Integrity: Behavior Validation* increases the likelihood that if the behavior of a system deviates from assumed norms the defenders will be made aware of the issue. |
| **Lack of Trusted Control or Insight: When systems and networks are rendered in software and control is migrated to the cloud, the system administration may not be knowledgeable regarding new roles, assumptions about separation of duties may be violated, and insight into some aspects of the computing environment may be lost.** | *Analytic Monitoring: Monitoring and Damage Assessment* increases the likelihood of detecting adversarial behaviors.<br>*Analytic Monitoring: Sensor Fusion and Analysis* increases accuracy of, and trust in, alerts. |
| | *Coordinated Defense: Technical Defense-in-Depth* provides support for the *Monitoring and Damage Assessment*.<br>*Coordinated Defense: Coordination and Consistency Analysis* increases trust and coordination in controlling configuration and management. |
| | *Substantiated Integrity: Behavior Validation* can help identify rogue virtual systems, services, or devices. |

**Table 7. Cyber Resiliency Approaches for Mitigating Adverse System Effects**

| Adverse System Effect | Selected Cyber Resiliency Approaches for Mitigation |
|---|---|
| **Full Control: Attacks that provide the attacker with full control of the hypervisor and the VMs that reside on it.** | *Analytic Monitoring: Monitoring and Damage Assessment* helps identify attacks earlier in the cyber attack lifecycle before the adversary attains full control.<br>*Analytic Monitoring: Sensor Fusion and Analysis* also increases the probability that attacks will be identified earlier in the cyber attack lifecycle. |
| | *Coordinated Defense: Coordination and Consistency Analysis* increases coordination in planning and executing defensive actions making it more likely defenders will prevent the adversary from gaining full control of a system. |
| | *Non Persistence: Non Persistent Information* flushes information preventing the adversary from using information from OS the adversary does not control, as well as flushing any information that the adversary placed in memory.<br>*Non Persistent Services* refreshes or resets services so that any attempts to use malware the adversary has inserted into those services to gain full control of a system is degraded or derailed. |
| | *Segmentation: Predefined Segmentation* reduces the ability of an attack to spread. |
| **Privilege Escalation: Attacks that allow attackers to escalate privileges.** | Both *Analytic Monitoring: Monitoring and Damage Assessment*, and *Analytic Monitoring: Sensor Fusion and Analysis* approaches increase the likelihood of detecting indications of adversary activities earlier in the lifecycle reducing the likelihood of a privilege escalation. |
| | *Non Persistent Information* flushes any information that the adversary placed in memory that might be used in a privilege escalation attack.<br>*Non Persistent Services* refreshes or resets services so that any attempts to use malware the adversary has inserted into those services to escalate privileges is degraded or derailed. |
| | *Segmentation: Predefined Segmentation* reduces the ability of an attack to spread. |

| Adverse System Effect | Selected Cyber Resiliency Approaches for Mitigation |
|---|---|
| **Unauthorized Sharing: Attacks that allow attackers to gain unauthorized access to information.** | *Privilege Restriction: Privilege Management* helps ensure established trust criteria are maintained (e.g., de-privileging certain pieces of the control domain, such as the device drivers in the control domain).<br>*Privilege Restriction: Privilege-Based Usage Restrictions* mitigates the possibility of an entity (e.g., guest OS) obtaining unauthorized access. |
| | *Non Persistence: Non Persistent Information* flushes information from memory before it can be reused by another resource.<br>*Non Persistent Services* refreshes or resets services so that data associated with those services is not shared. |
| | *Realignment: Restriction* removes or disables unneeded risky functionality or connectivity thereby reducing the incidence of unauthorized sharing. |
| | *Segmentation: Predefined Segmentation* reduces the probability of unauthorized sharing. |
| **Configuration Dependent: Attacks whose effects are dependent on configuration.** | *Non Persistence: Non Persistent Information* flushes information from memory before it is reused by another resource, reducing the probability of the information being misused. |
| | *Privilege Restriction: Privilege Management* minimizes access to resources, reducing access by attackers.<br>*Privilege Restriction: Privilege-Based Usage Restrictions* minimizes access to critical resources, reducing the risk of unauthorized usage of those resources. |
| | *Segmentation: Predefined Segmentation* reduces the probability that unauthorized sharing will occur thereby reducing the probability of the information being misused by an attacker. |

# 5 Conclusions

This document discussed the challenges posed by virtual and cloud environments, and cyber resiliency techniques that may be able to increase mission assurance in systems whose architectures are based on virtual infrastructure and cloud services. The discussion was framed by architectural factors and adverse system effects that are specific to the VM and cloud environments.

This paper focused on the additional risk due to the system being virtualized and/or migrated to the cloud. While the mitigations mentioned in this document reduce the likelihood of an architectural factor being exploited and minimizes the impact of exploitations, they do not eliminate all risk associated with the vulnerabilities.

It is not feasible to apply all cyber resiliency techniques to an architecture. Identifying the techniques that most frequently addressed the architectural factors and adverse effects discussed in this document, and taking into account the relative maturity and readiness for cyber resiliency application and the potential interactions between the techniques (both conflicting and synergistic), resulted in a list of recommended cyber resiliency approaches for each applicable technique.

The final list of recommended approaches is:

- Analytic Monitoring: Monitoring & Damage Assessment
- Analytic Monitoring: Sensor Fusion & Analysis
- Non Persistence: Non Persistent Information
- Non Persistence: Non Persistent Services
- Privilege Restriction: Privilege-Based Usage Restrictions
- Privilege Restriction: Privilege Management
- Segmentation: Predefined Segmentation
- Coordinated Defense: Technical Defense-in-Depth
- Coordinated Defense: Coordination and Consistency Analysis
- Realignment: Restriction
- Substantiated Integrity: Behavior Validation

This list addressed all of the architectural factors and adverse system effects. Each architectural factor and each adversarial effect is mitigated by at least three different approaches, insuring a robust set of mitigations. Both Substantiated Integrity: Behavior Validation and Realignment: Restriction also support other approaches as well as address the factors and effects directly.

## 5.1 Next Steps

The analysis described in this document is based on generic concerns related to virtualized and cloud environments. The next step is to assess specific environments in the presence of specific threats (e.g., public exploits). This would allow for a more realistic analysis of the recommended mitigations and enable the development of metrics that quantify a range of effectiveness and efficiency measures (i.e., effectiveness of mitigation, performance impact, difficulty of implementation, etc.).

# 6 References

[1] NIST Joint Task Force Transformation Initiative, "Guide for Conducting Risk Assessments," September 2012. [Online]. Available: http://csrc.nist.gov/publications/nistpubs/800-30-rev1/sp800_30_r1.pdf. [Accessed 10 February 2016].

[2] "Define The Cloud," [Online]. Available: www.definethecloud.net. [Accessed 24 November 2015].

[3] T. McNevin, R. W. Schmeichel and D. B. Faatz, "Mitigating Hypervisor Vulnerabilities," The MITRE Corporation, Bedford, 2010.

[4] K. M. Bitting , L. D. Mitchell and R. P. Starski, "Certificate Management Infrastructure (CMI) Virtualization Study," The MITRE Corporation, McLean, VA, 2013.

[5] D. Shackleford, *Virtualization Security: Protecting Virtualized Environments*, John Wiley & Sons, 2013.

[6] "CVE," [Online]. Available: http://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2015-3456. [Accessed 21 September 2015].

[7] "Computer World," [Online]. Available: http://www.computerworld.com/article/2886952/lenovo-admits-to-superfish-snafu-plans-to-release-clean-up-tool.html. [Accessed 8 September 2015].

[8] S. Sharewood, "The Register," 28 July 2015. [Online]. Available: http://www.theregister.co.uk/2015/07/28/xen_reports_new_guesthost_escape_this_time_through_cdroms/. [Accessed 25 November 2015].

[9] R. Wojtczuk and J. Rutkowska, "Subverting the Xen Hypervisor," August 2008. [Online]. Available: Wojtczuk, R., and J. Rutkowska. http://invisiblethingslab.com/resources/bh08/part1.pdf. [Accessed 26 August 2015].

[10] V. Vaidya, "Virtualization Vulnerabilities and Threats:A Solution White Paper," RedCannon Security, Inc, 2009.

[11] Cloud Security Alliance - Top Threats Working Group, "Cloud Security Alliance," February 2013. [Online]. Available: https://downloads.cloudsecurityalliance.org/initiatives/top_threats/The_Notorious_Nine_Cloud_Computing_Top_Threats_in_2013.pdf. [Accessed 1 December 2015].

[12] J. Hietala, "SANS," [Online]. Available: https://www.sans.org/reading-room/whitepapers/analyst/top-virtualization-security-mistakes-and-avoid-them-34800. [Accessed 1 December 2015].

[13] G. Irazoqui, M. S. Inic, T. Eisenbarth and B. Sunar, "Wait a minute! A fast, Cross-VM attack on AES," Worcester Polytechnic Institute, Worcester, MA, 2014.

[14] G. I. Apecechea, M. S. Inci, T. Eisenbarth and B. Sunar, "Fine grain Cross-VM Attacks on Xen and VMware are possible!," Worcester Polytechnic Institute, Worcester, MA, 2014.

[15] J. Oberheid, E. Cooke and F. Jahanian, "Empirical Exploitation of Live Virtual Machine Migration," Ann Arbor, MI.

[16] D. Bodeau, R. Graubart, W. Heinbockel and E. Laderman, "Cyber Resiliency Engineering Aid – The Updated Cyber Resiliency Engineering Framework and Guidance on Applying Cyber Resiliency Technique," The MITRE Corporation, Bedford, MA, 2015.

[17] Joint Task Force Tranformation Initiative, "Computer Security Division: Computer Security Resource Center," September 2012. [Online]. Available: http://csrc.nist.gov/publications/nistpubs/800-30-rev1/sp800_30_r1.pdf.

This page intentionally left blank.

# Appendix A Cyber Resiliency Approaches and the Cyber Attack Lifecycle[14]

One consideration in determining which cyber resiliency approaches to apply to a specific environment is how the cyber resiliency approaches effect the adversary throughout the lifecycle (CAL). The CAL[15] provides a framework for understanding and analyzing how distinct adversary activities contribute to an attack. Understanding the CAL gives insight into the steps the adversary needs to complete to be successful. This understanding enables the defender to identify actions and opportunities for countering adversary activities. Rather than focusing on a single stage of the lifecycle (e.g., trying to prevent delivery of malware), the defender can attempt to counter the adversary at various stages, as the adversary needs to satisfy all the stages to achieve its goals.

Figure 8 depicts and Table 8 describes the CAL stages of a malware-based cyber attack. The pre-exploit stages represent a defensive opportunity to proactively deter, detect, and mitigate threats before the adversary establishes a foothold. The structure of the adversary cyber attack campaign is recursive. In the post-exploit stages, the adversary attempts lateral movement to extend the foothold in the organization and the cycle repeats. Post-exploit, organizations can perform incident detection/response together with resilient operations to ensure that mission-critical assets continue to support mission operations.

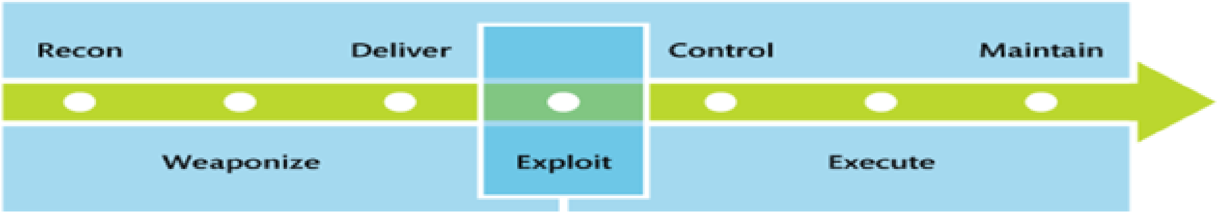


**Figure 8. Cyber Attack Lifecycle**

**Table 8. Stages of the Cyber Attack Lifecycle**

| Stage | Description |
|---|---|
| **Recon** | The adversary identifies a target and develops intelligence to inform attack activities. The adversary develops a plan to achieve desired objectives. |
| **Weaponize** | The adversary develops or acquires an exploit (e.g., a "0-day"), places it in a form that can be delivered to and executed on the target device, computer, or network. |
| **Deliver** | The exploit is delivered to the target system. (e.g., tailored malware is included in a spear phishing email attachment or compromised components inserted in the supply chain are integrated into a target network). |
| **Exploit** | The initial attack on the target is executed. (e.g., a vulnerability is exploited and malware is installed on an initial target system). |

[14] Information in this appendix is based on [16].

[15] There are multiple versions of the Cyber Attack Lifecycle, also referred to as the Cyber Kill Chain. The one depicted here is consistent with what is described as a cyber campaign in National Institute of Standards and Technology (NIST) Special Publication (SP) 800-30 R1 [17].

| Stage | Description |
|---|---|
| **Control** | The adversary employs mechanisms to manage the initial targets, perform internal reconnaissance, and compromise additional targets. |
| **Execute** | The adversary executes the plan and achieves desired objectives (e.g., exfiltration of sensitive information, corruption of mission-critical data, fabrication of mission or business data, degradation or denial of mission-critical services). |
| **Maintain** | The adversary ensures a sustained, covert presence on compromised devices, systems, or networks. To do so, the adversary may erase indications of prior presence or activities. |

Table 9 depicts the potential effects the cyber resiliency approaches, discussed in this paper, may have on an adversary throughout the cyber attack lifecycle. The effects listed in this table would need to be considered in light of the specific environment before determining whether the approaches would actually have these effects.

**Table 9. Potential Effects of Cyber Resiliency Approaches on Adversary Activities in Virtual Machine and Cloud Environments[16]**

| Approach | Reconnaissance | Weaponize | Deliver | Exploit | Control | Execute | Maintain |
|---|---|---|---|---|---|---|---|
| **Segmentation: Predefined Segmentation** | Constrain | | Degrade | | Contain<br>Delay<br>Degrade<br>Detect | Contain<br>Delay<br>Degrade<br>Detect | Contain<br>Delay<br>Degrade |
| **Analytic Monitoring: Monitoring & Damage Assessment** | Detect | | Detect | | Detect | Scrutinize | Detect |
| **Non Persistence: Non Persistent Information** | | | | | | Shorten | |
| **Privilege Restriction: Privilege-Based Usage  Restrictions** | | | | Prevent<br>Contain<br>Degrade | Prevent<br>Contain<br>Degrade | Prevent<br>Contain<br>Degrade | Prevent<br>Contain<br>Degrade |
| **Analytic Monitoring: Sensor Fusion & Analysis** | Detect<br>Scrutinize | | | | Detect<br>Scrutinize | | Detect<br>Scrutinize |
| **Privilege Restriction: Privilege Management** | Degrade<br>Delay | | | Contain<br>Delay<br>Prevent | Contain<br>Delay<br>Prevent | Contain<br>Delay<br>Prevent | Contain<br>Delay<br>Prevent |
| **Coordinated Defense :Technical Defense in Depth** | | Delay | | Degrade<br>Delay | | | |
| **Redundancy: Surplus Capacity** | | | | | | Degrade<br>Recover | |
| **Coordinated Defense:** | | | | | Detect<br>Degrade | Degrade<br>Delay | Detect<br>Degrade |

[16] While the information about which cyber resiliency approach has which effect on the adversary is taken from [14], the specific nomenclature has been updated based on [13].

| Approach | Reconnaissance | Weaponize | Deliver | Exploit | Control | Execute | Maintain |
|---|---|---|---|---|---|---|---|
| **Coordination and Consistency Analysis** | | | | | Delay | | Delay |
| **Dynamic Representation: Dynamic Mapping & Profiling** | | | | | Detect | | Detect |
| **Substantiated Integrity: Behavior Validation** | | | | | Detect<br>Shorten | Detect<br>Shorten | Detect<br>Shorten |
| **Realignment: Purposing** | Degrade<br>Delay | | Prevent<br>Degrade | | Prevent<br>Degrade | Prevent<br>Degrade | Prevent<br>Degrade |
| **Diversity: Architectural Diversity** | | Degrade<br>Delay | | Prevent<br>Degrade | Degrade<br>Contain | Degrade<br>Contain | Recover |
| **Dynamic Representation: Mission Dependency & Status Visualization** | | | | | | | Detect<br>Recover |
| **Diversity: Design Diversity** | | Degrade<br>Delay | | Prevent<br>Degrade | Degrade<br>Contain | Degrade<br>Contain | Recover |
| **Diversity: Synthetic Diversity** | | Degrade<br>Delay | | Prevent<br>Degrade | Degrade<br>Contain | Degrade<br>Contain | Recover |

# Appendix B Abbreviations

| | |
|---|---|
| API | Application Programming Interface |
| APP | Application |
| APT | Advanced Persistent Threat |
| ARP | Address Resolution Protocol |
| CAL | Cyber Attack Lifecycle |
| CREF | Cyber Resiliency Engineering Framework |
| DB | Database |
| DNS | Domain Name System |
| DOS | Denial-of-Service |
| HTTP | Hypertext Transport Protocol |
| HTTPS | HTTP Secure |
| I&A | Identity and Authentication |
| IDS | Intrusion Detection System |
| IT | Information Technology |
| M-I-M | Man-in-the-Middle |
| MOA | Memorandum of Agreement |
| MS | Microsoft |
| NIST | National Institute of Standards and Technology |
| NOS | Network Operating System |
| NTP | Network Time Protocol |
| OS | Operating System |
| POET | Political, Operational, Economic and Technical |
| RAM | Random Access Memory |
| SDN | Software-defined Networking |
| SLA | Service Level Agreement |
| SP | Special Publication |
| SQL | Structured Query Language |
| TTP | Tactics, Techniques, and Procedures |
| VM | Virtual Machine |