VULNERABILITY PREDICTION AND RISK ASSESSMENT OF THE XEN HYPERVISOR

A THESIS SUBMITTED TO AUCKLAND UNIVERSITY OF TECHNOLOGY
IN PARTIAL FULFILMENT OF THE REQUIREMENTS FOR THE DEGREE OF
DOCTOR OF PHILOSOPHY

Supervisors

Dr Alan T. Litchfield

Dr Brian Cusack

April 2019

By

Abid Shahzad

School of Engineering, Computer and Mathematical Sciences

Abstract

This thesis presents a vulnerability prediction and risk assessment process of the Xen hypervisor. The process predicts the number of unknown Xen vulnerabilities that may appear in the future. It also determines the risk severity levels a specific Xen version provides. The hypervisor is a key component of virtualisation to offer an Infrastructure-as-a-Service delivery model. Thus, the hypervisor is an attractive target of attackers to compromise critical assets that usually belong to different tenants. When such a critical component is compromised, the assets of service customers are consequently at risk.

Cloud computing has matured with time, but many organisations have security concerns due to new risks compared to conventional IT environment. The types of risk also vary from one service delivery model to another. Much research has been conducted to assess the risk of cloud computing, but it has viewed and assessed risk from a broader perspective instead of focusing on the hypervisor which provides the base for Infrastructure-as-a-Service. Moreover, cloud service providers are responsible for managing the security of the hypervisors which makes service customers utterly unaware of the security of their data if they move to cloud virtualised infrastructure. Therefore, to encourage customers to adopt Infrastructure-as-a-Service free of security concerns, a new assessment platform specific to the hypervisor is required. However, the following questions arise: *How can the unknown vulnerabilities be predicted in large software applications such as the Xen hypervisor to mitigate exploitation scenarios? How can the determination of the risk of unknown Xen vulnerabilities be presented such*

that it aids cloud infrastructure service consumers?

This research targets the Infrastructure-as-a-Service delivery model and presents a Xen vulnerability prediction and risk assessment process. Different analysis and research methods are used in this research. The Time Series Holt-Winters method is used to predict unknown vulnerabilities. The regression analysis method is used to predict unknown vulnerabilities with regard to the impact levels (High, Medium, and Low). ENISA risk framework is considered to adopt XEN vulnerability impact ratings. A structured analysis approach using attack trees is used to determine threat likelihood levels. A risk estimation matrix is used to map the vulnerability impact ratings and threat likelihood levels to determine qualitative risk severity levels.

The vulnerability prediction and risk assessment process allows customers to use results of vulnerabilities and risk of Xen to make informed security decisions. The process is very effective for the small organisations that do not have security professionals or experts to assess the security risks they could face after moving their critical services to cloud virtualised infrastructure. Nine technical risks to the Xen hypervisor are identified and security recommendations are made for customers regarding each of the risk categories. However, customers are encouraged to identify and add new risks in the assessment process that may be specific to their services, data, and information. The customers can then consider the security recommendations made in this research to select a cloud service provider after analysing the security controls which are in place to mitigate these risks.

The vulnerability prediction and risk assessment process is developed on the Xen hypervisor and tested on the two other popular open source, infrastructure level software packages. Vulnerability prediction and risk assessment of Apache HTTP and Squid Proxy servers is performed to evaluate the process to ensure its generalisability and applicability. In each case, the results of vulnerability prediction and risk assessment are good to fair.

Contents

List of Tables

List of Figures

Attestation of Authorship

I hereby declare that this submission is my own work and that, to the best of my knowledge and belief, it contains no material previously published or written by another person nor material which to a substantial extent has been accepted for the qualification of any other degree or diploma of a university or other institution of higher learning.

Signature of student

Publications

Vulnerability and Risk Assessment of Xen Hypervisor	In Proceedings of the 24^{th} Americas Conference on Information Systems (AMCIS), New Orleans, USA, August 2018.
Xen Hypervisor: Vulnerability and Risk Assessment Process	9^{th} New Zealand Information Systems Doctoral Consortium (NZISDC), Auckland, New Zealand, July 2018.
A Systematic Review of Vulnerabilities in Hypervisors and Their Detection	In Proceedings of the 23^{rd} Americas Conference on Information Systems (AMCIS), Boston, USA, August 2017.
Virtualization Technology: Cross-VM Cache Side Channel Attacks Make It Vulnerable	In Proceedings of the Australasian Conference on Information Systems (ACIS), Adelaide, AUS, December 2015.

Acknowledgements

First of all, I would like to say sincere thanks and express utmost respect to primary supervisor, Dr Alan T. Litchfield for his kind support, suggestions, guidance and encouragement throughout this research thesis. I do not have enough words to say thanks to Dr Alan for his help to achieve this milestone. I am very grateful to him for taking as his PhD student. I have learned a lot from him and admired greatly as a mentor.
I also want to extend sincere gratitude and thanks to secondary supervisor Dr Brian Cusack for his support and guidance to meet the requirements of this thesis. His valuable feedback was very critical throughout this research.
I would also like to extend thanks to all colleagues in Service and Cloud Computing Research Lab (SCCRL) for their support and help. I am also very thankful to AUT and especially the School of Engineering, Computer and Mathematical Sciences for giving this opportunity to make a dream come true to be the first PhD in the family.

Dedication

I am dedicating this thesis to loving parents for their extraordinary love and prayers. Their unconditional support throughout this journey helped to achieve targets and finish this thesis.
It is also dedicated to caring sisters and brothers, and especially to loving and charming wife who has encouraged a lot. She always motivated me to get through the difficult phases to complete this research.

Chapter 1

Introduction

1.1 Introduction

This research provides a vulnerability prediction and risk assessment process for the Xen hypervisor. Potential cloud customers can use this process to predict unknown vulnerabilities to the Xen hypervisor. The results of the risk assessment process can be used by customers to understand the risk of moving their data and information to a virtualised infrastructure. This process is quite useful for the organisations without enough knowledge and security experts to assess vulnerabilities and risk before selecting an appropriate cloud service provider (CSP) and moving their services and data to a cloud virtualised infrastructure.

Cloud computing (CC) offers organisations the use of outsourced services through the Internet and typically through a pay-per-use model, as opposed to buying and setting up resources in-house (Srinivasan, Sarukesi, Rodrigues, Manoj & Revathy, 2012; Vaquero, Rodero-Merino & Morán, 2011). Besides providing a cost-effective solution, CC has added benefits such as elasticity, scalability, and multi-tenancy (Khorshed, Ali & Wasimi, 2012; Tianfield, 2012; Hashizume, Rosado, Fernández-Medina & Fernandez, 2013). CC offers three different service delivery models: Software-as-a-Service (SaaS),

Platform-as-a-Service (PaaS), and Infrastructure-as- a-Service (IaaS).

This chapter is organised as follows: an overview of the hypervisor and common types of hypervisors is provided in Section 1.2. Section 1.3 details the cloud service delivery models. Security responsibilities for each of these models are provided in Section 1.3.1. Section 1.3.2 highlights virtualisation security concerns. The hypervisor security is covered in Section 1.3.3. Section 1.4 provides the problem statement and Section 1.5 provides research objectives. In Section 1.6, the structure of the thesis is provided. The conclusion of this chapter is provided in Section 1.7.

1.2 Hypervisor or Virtual Machine Monitor

This section provides details of the hypervisor and its types. The hypervisor is a core component which provides the platform for virtualization. Multiple Virtual Machines (VMs) can be created on a physical server using the hypervisor. The hypervisor provides a software layer between hardware and OS of the host server (You, Peng, Liu & Xue, 2012). The hypervisor controls the flow of instructions between the guest VM OS and the hardware, involving elements such as Processor core, CPU Cache, RAM, hard disk drives, and Network Interface Cards (NICs) (Ayala, Vega & Vargas-Lombardo, 2013). The hypervisor provides the isolation between multiple VMs running on the shared hardware and ensures these VMs are separate entities (You et al., 2012).

1.2.1 Categories of Common Hypervisors

Native or bare-metal virtualization hypervisors run directly on the underlying hardware without the need of a Host OS. The XenServer, VMware ESXi, and Microsoft Hyper-V are common bare-metal hypervisors. These hypervisors have direct access to the hardware of the physical server, resulting in efficient and high levels of performance.

VMware Workstation/Player, Oracle VirtualBox, and Kernel based VM (KVM) (Pék, Buttyán & Bencsáth, 2013) are the common hosted hypervisors. These hypervisors require a Host OS to be installed and managed. These hypervisors provide hardware resources to the VMs through the Host OS. Such hypervisors are not as efficient as bare-metal hypervisors.

The virtualization market is dominated by VMware, Hyper-V, XEN, and KVM hypervisors. According to Perez-Botero, Szefer and Lee (2013), 93 % of the hypervisor market is covered by these four hypervisors. In 2012, Nexenta conducted a survey and highlighted that the current market is highly dominated by these hypervisors. The survey involved around 4,000 users and took over two months to complete. The survey results show that 16 % of users use Hyper-V, 13 % use either KVM or XEN Server, and 58% use VMware as their primary hypervisors. Moreover, 56 % of users selected VMware, 17 % selected Hyper-V, 14 % selected KVM, and 13 % selected XEN Server as their preferred hypervisor for the next 12 months.

1.3 Cloud Service Delivery Models

This section provides an overview of cloud service delivery models. Different CSPs offer any of these services individually or all three services together (Modi, Patel, Borisaniya, Patel & Rajarajan, 2013; Hashizume et al., 2013; Chhabra & Dixit, 2015).

IaaS is the delivery of virtualised infrastructure that mainly includes servers, storage, and networking. Customers can use the virtualised infrastructure through the internet service. Amazon Web Services (AWS) is the popular CSP that offers IaaS cloud (Amazon EC2). Customers use their Application Program Interfaces (APIs) to configure and manage virtual servers, storage, and networks.

Through PaaS, customers access different online tools required for development. Through PaaS, CSPs provide development tools which customers can access even in

collaboration with others. CSPs use APIs, gateway software, or portals that are installed on the customer's premises. The most common example of PaaS is GoogleApps.

SaaS is a type of cloud offering where CSPs deliver centrally hosted applications through the internet. SaaS offerings are often referred to as web-based or hosted software, and all the software runs on the CSPs' servers. Customers can access software via the internet instead of installing and managing it locally. All the SaaS applications are usually accessed by customers through web browsers. Figure 1.1 provides examples of popular CSPs which offer these service delivery models.

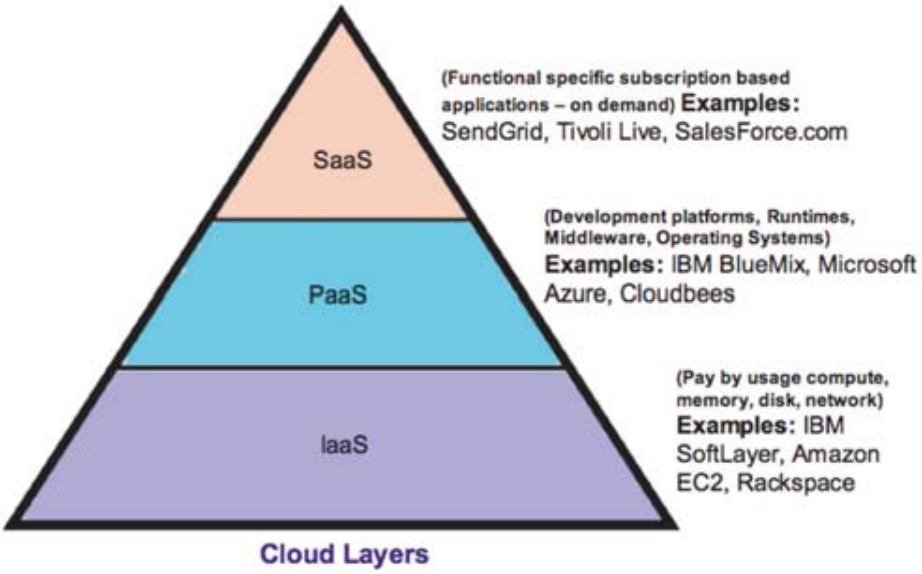


Figure 1.1: Cloud Service Delivery Models (Barrowclough & Asif, 2018)

1.3.1 Security Responsibilities

In a traditional data centre environment, organisations manage infrastructure and security by themselves. However, security management processes in cloud service delivery models are different. Figure 1.2 provides a summary of the security responsibilities of customers and CSPs in each of the delivery models.

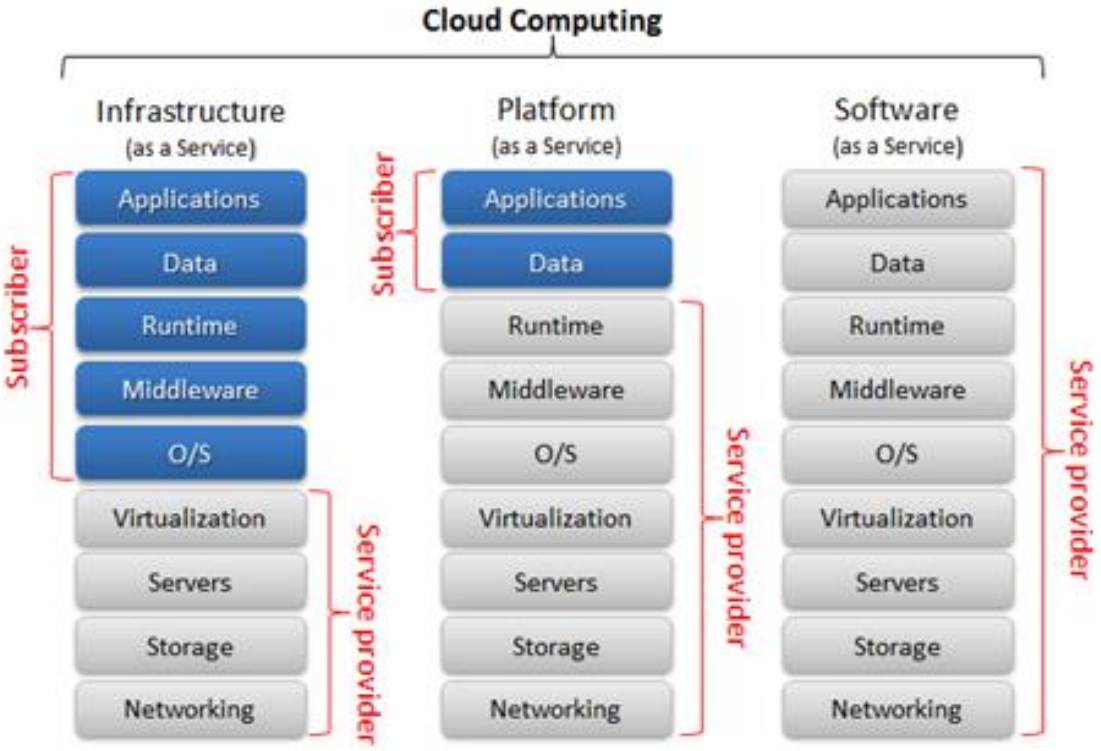


Figure 1.2: Security Responsibilities (Chou, 2010)

In PaaS, customers do not manage the security of physical servers, network, data storage equipment, operating systems (OS) and platform software packages. Customers only ensure the security of their applications by applying service packs and patches. However, CSPs are responsible for managing the security of almost everything else (Albakri, Shanmugam, Samy, Idris & Ahmed, 2014).

In SaaS, managing security is mostly the responsibility of CSPs. CSPs manage infrastructure, the platform, and installed software packages (Albakri et al., 2014). Customers are only responsible for ensuring the security of their applications. For example, SaaS customers must implement a policy to setup strong credentials and the protection of these credentials. The username and password should not be shared with others. Weak, shared, and lost credentials can result in unauthorised access by attackers.

IaaS, which is the focus of this research, provides infrastructure services such as VMs, storage equipment, network, and database services to customers. Customers'

involvement in managing the security in IaaS is more as compared to PaaS and SaaS (Albakri et al., 2014). Usually, the customers manage OS, software packages, applications and data access permissions. However, managing the security of a core IaaS component, the hypervisor and its virtual assets, physical servers, storage, and networking is the sole responsibility of the CSPs.

1.3.2 Virtualisation Security Concerns

Virtualisation is a core aspect of the IaaS delivery model (Shoaib & Das, 2014). Virtualisation allows the creation of VMs on a single physical server. Different software and applications can run concurrently on these VMs by sharing the same hardware resources. In other words, virtualisation allows the sharing of hardware resources between different VMs that emulate the physical server (Chhabra & Dixit, 2015). Despite the many benefits of using virtualisation, the biggest concern to virtualisation is the security of the cloud infrastructure. Schulze (2015) presented a survey where approximately 1000 security professionals were consulted for their views and concerns about the security of the cloud. The survey highlights that around 90 % of the respondents are concerned about the security of public cloud virtual infrastructures. Around 45 % of the respondents mentioned that the security concerns are the main barriers to customers' adopting cloud-based services. 41 % respondents mentioned that data loss and leakage is their bigger concern. Many respondents also mentioned that unauthorised access to resources, account hijacking, and malicious insiders are the biggest risk to public cloud services. However, despite these security concerns, cloud adoption rate is on the rise. Cost saving is the biggest attraction for the organisations which motivates them to adopt cloud services and neglect the fact of understanding the risk to the cloud.

Along with existing threats to the traditional IT environment, virtualisation leverages new types of risk. In a typical IaaS scenario, customers are responsible for managing

the security of all the applications running on VMs which are provided by the CSP. However, customers are unaware of the security of VMs on which their applications are running. The customers manage the OS, programs and applications in the IaaS delivery model which gives them more control and satisfaction as compared to PaaS and SaaS models (Section 1.3.1). However, customers are concerned about the risk of storing their sensitive data and information on cloud virtualised infrastructure if they adopt the IaaS model.

1.3.3 Hypervisor Security

Moving from a conventional IT environment to the cloud virtual infrastructure raises security concerns. However, these security concerns are not completely different from the IT environment. For example, a vulnerability that exists in the physical server can also exist in a virtual cloud server. A common understanding is that the hypervisor provides a virtualisation layer between the hardware and software of the server that might reduce the impact of vulnerability exploitation. However, vulnerabilities in hypervisors may provide new Attack Vectors (AVs) that may result in more sophisticated attacks.

Resource Isolation

The hypervisor provides VM isolation and manages guest VM's access to shared hardware. It ensures isolation by partitioning hardware resources for each VM individually. It does not allow a VM to access the hardware resources used by other VMs. This guest VM isolation prevents unauthorised access to hardware resources and increases the protection against malware injection by malicious VMs such as injecting malicious code into other VM's memory. This isolation helps in minimising the chances of Denial of Service (DoS) by preventing malicious VMs exceeding resource consumption in

other VMs OSs running on the same hypervisor.

Resource isolation provided by the hypervisor can be physical or logical. In the case of physical isolation, the hypervisor allows a separate portion of hardware resources to the VMs. Through logical isolation, the hypervisor divides resources of a shared server between different VMs as in a pool of resources with the same security impact level categorisation. In this case, the hypervisor allows multiple guest VMs to share the same CPU, RAM, and disk drive. Physical resource isolation provides better security and performance as compared to the case where logical isolation is considered.

Resource isolation offers security benefits and enhances the hardware server's reliability by preventing the actions of one VM affecting other VMs. For example, if one VM is attacked or infected, all other VMs will not be affected by this VM. Isolating VMs from one another and allowing each VM to access dedicated hardware resources, is called sandboxing.

Despite the fact that the hypervisor ensures the isolation of resources between VMs, the hypervisors are however vulnerable to side-channel attacks (Lawson, 2009). Through a side-channel attack, an attacker exploits the physical hardware to extract useful information such as cryptographic keys by monitoring the access patterns or behaviour of guest VMs for Cache memory and CPU (Y. Zhang, Juels, Reiter & Ristenpart, 2012).

Introspection

The hypervisor monitors each guest VM running on top of it using introspection. Introspection provides complete auditing capabilities that may otherwise be unavailable. Monitoring through hypervisor introspection capabilities covers memory access, processor access, network access, and different other features of a VM. The hypervisor can provide information to the additional security controls that are learned from introspection. These security controls can be used by different virtualisation products to allow

the security policies to be enforced and also to be moved when a VM is moved from one physical server to another.

However, the virtual network traffic between guest VMs or between guest VMs and the host OS should be monitored. The optimised host-based security controls are required to monitor the virtual network traffic because of standard network security controls and are ineffective due to the fact that traffic does not pass through these physical network controls.

Security of VM Images

Vulnerabilities in guest OS, applications, and services are not affected by creating VM images and snapshots; however, security of images and snapshots is critical because these images and snapshots contain sensitive customer data that can affect their reputation. Though VM images and snapshots are easy to manage, the security of the data in these images and snapshots is very critical. The data stored in snapshots is even more sensitive because this data also contains the contents of memory. Moreover, there are chances that snapshots might include information that was accessed from another network server or storage and not from the local virtual disk drive.

1.4 Problem Statement

An overview of cloud service delivery models and the security responsibilities of customers and CSPs was provided in Section 1.1. It also highlighted virtualisation and hypervisor security concerns. This section provides the problem area and how it is addressed through this research. The hypervisor is a critical asset for CSPs and is often a target of attackers. If the hypervisor is compromised, it can damage critical virtual assets that belong to different customers. It provides a point of possible attack for hackers to gain access to virtual assets. An attacker can take control of all guest VMs and the

data stored on those VMs after successfully exploiting the hypervisor. Hypervisors are considered to be secure and robust. However, like other software packages, they contain vulnerabilities. Exploitation of hypervisor vulnerabilities (Kortchinsky, 2009; Wojtczuk, 2008; Elhage, 2011; Rutkowska & Wojtczuk, 2008) provides opportunities for attackers to launch further attacks to compromise virtual assets. Also, the vulnerabilities in the hypervisors can lead to the destruction of virtual infrastructure which is running on top of it. Though CSPs are improving the security of the cloud and ensure resource isolation, a few cases have been reported in the near past where hypervisor vulnerabilities are patched by the CSPs. Some possible attack scenarios to the hypervisors are as follows:

1.4.1 Breach of Isolation

Hypervisor escape is an attack type where an attacker uses a malicious VM to acquire root level access by breaking or escaping the barrier provided by the hypervisor. The hypervisor escape attack can be realised by exploiting a hypervisor design vulnerability or vulnerability in the device driver modules of the Dom0. Through a successful attack, a malicious VM can access the portion of the Cache memory which belongs to other VMs or the hypervisor itself during a particular time slot. Moreover, a malicious VM can access the storage devices where virtual machine images or snapshots are stored. Moreover, a compromised hypervisor provides a single point of failure for all the guest VMs where single vulnerability exploitation can put all the guest VMs and their data at risk (Modi et al., 2013; Pék et al., 2013; Shoaib & Das, 2014; Pearce, Zeadally & Hunt, 2013; Luo, Lin, Chen, Yang & Chen, 2011).

1.4.2 Denial of Service

An attacker can execute a DoS attack by unnecessarily consuming hardware resources of the hypervisor host server. Such an action by a malicious VM would result in DoS to

all the legitimate VMs and the host server. In another scenario, a DoS attack is possible when an attacker can control a host OS or the hypervisor and create a large number of unnecessary VMs to consume the hardware resources of the server (Kazim, Masood, Shibli & Abbasi, 2013).

1.4.3 Breach of Network Isolation

An attacker can use a malicious VM to break network isolation by spoofing the Internet Protocol (IP) or Media Access Control (MAC) address. It would allow an attacker to intercept the network traffic of legitimate VMs running on the same hypervisor and using the same virtual network. Such a breach would result in the loss of confidentiality because a malicious VM can access the information which belongs to other VMs.

1.5 Research Objectives

In Section 1.4, the problem area was highlighted. This section details the objectives of this research to address the problem. In general, the hypervisor appears to be secure and robust. However, hypervisor functionalities have broad security complications. Therefore, a hypervisor vulnerability and risk assessment platform is presented to ensure the security of infrastructure services provided by the target hypervisor. However, performing vulnerabilities, threat, and risk assessment of a large software such as the hypervisor is a complex task.

1.5.1 Xen Hypervisor

Xen is an open source hypervisor developed by The University of Cambridge Computer Laboratory in 2003. It is a bare-metal hypervisor and runs directly on hardware without the need for a host OS to manage hardware resources for all the VMs (Figure 1.3).

It creates a privileged Dom0 which is aware of the Xen. Dom0 performs all the VM management functions for Xen. All the VMs usually called DomUs, are not aware of the virtualisation layer when runs in full virtualisation mode. In addition to basic administrative tasks, Dom0 connects an instance of a device emulator, QEMU, to each DomU resulting in exposing the emulated devices (Perez-Botero et al., 2013).

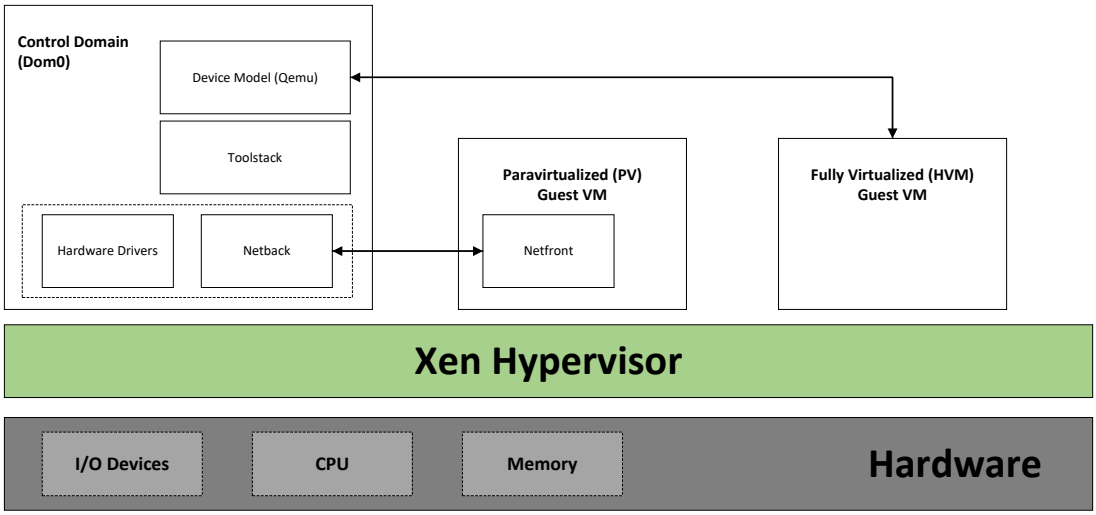


Figure 1.3: Xen Hypervisor

According to NVD search, Xen is more vulnerable than other conventional hypervisors as 62 vulnerabilities were reported in Xen during 2017. However, this is not a definitive explanation; it is probable that Hyper-V's owner (Microsoft) would prefer not to have vulnerabilities known publicly, to protect its reputation and reduce the chances of mass attacks against its OSs that use Hyper-V internally. This view is supported by Arora, Nandkumar and Telang (2006) that not publishing known vulnerabilities and publishing unpatched vulnerabilities (those that have not yet exploited) attracts fewer attacks than the publication of known and already patched vulnerabilities. On the other hand, all Xen vulnerabilities are reported, and the information is available because it is an open source hypervisor. With the number of exploits reported to NVD there is a good reason for concern. Therefore, an optimised risk assessment process for Xen is desirable. Such a process can guide customers to assess the risk to their virtual assets

provided by the Xen hypervisor. This research presents a qualitative risk assessment process for the Xen hypervisor. The process used for Xen vulnerability prediction and risk assessment is as follows:

1.5.2 Vulnerability Prediction

Vulnerability exploitations in common hypervisors result in compromise of CIA of the critical cloud assets. One of the most common reported vulnerabilities in the hypervisors is a DoS vulnerability. However, it is not evident what factors lead to the exploitation of these DoS vulnerabilities. It appears that there is a link between the number of vulnerabilities found, and knowledge of the type of vulnerability in the population. It is hypothesised that, as the knowledge of a type of vulnerability grows (the triggers, what code, software behaviours and so on), then more of that type of vulnerability are found in software.

The prediction of vulnerabilities can help minimise the damage which can be caused by the exploitation of unknown vulnerabilities when found. Therefore, it is desirable to identify the vulnerabilities earlier in software systems to help reduce the cost of damage and also the loss of reputation which can be caused by a successful exploitation. The existing research presents some techniques and tools for the identification of vulnerabilities using component characteristics such as code complexity and code churn (Shin, Meneely, Williams & Osborne, 2011). However, the existing techniques seem ineffective and lack applicability. This research presents a Xen vulnerability prediction model which is based on a Holt-Winters time series prediction method. The Holt-Winters method is a very simple prediction model and leverages the Xen reported vulnerability data set to make the prediction. The reported vulnerability data set is collected from the National Vulnerability Database (NVD). The number of reported vulnerabilities in Xen version 4.x is extracted from the database and used as input to

the prediction model. The vulnerability dataset contains Xen vulnerabilities which have been reported during the past five years. The prediction results seem to be very promising as the prediction model achieves an average accuracy on the whole of 4.x series of the Xen hypervisor.

1.5.3 Risk Assessment

Risk assessment is a process of risk management (Cayirci, 2015; Albakri et al., 2014). Risk assessment allows organisations or individuals to determine the impact of vulnerabilities, the likelihood of threats, and the levels of overall risk to the assets to make informed security decisions. Risk assessment can be quantitative (numeric value such as probability or proportion) or qualitative (non-numeric or descriptive). It can also be inductive or deductive (Cayirci, 2015). Inductive risk assessment induces all the possible AVs and consequences of an exploitation scenario. An attack tree is an example of inductive risk assessment. On the other hand, deductive risk assessment starts with reasons and deduces from reasons until the attacker's goal is determined.

The hypervisor vulnerability exploitations by sophisticated threats pose different risks to cloud virtualised infrastructure. The hypervisor is a large and complex software. It provides a very dynamic environment which makes risk assessment more difficult. However, the attackers often want to exploit the hypervisors to get unauthorised access to the virtual assets provided by them. Therefore, undertaking a risk assessment of the hypervisor and its virtualised infrastructure is necessary. Protecting the hypervisors from attacks is challenging as the number of unknown vulnerability exploits increases. Furthermore, as compared to a conventional IT environment, the deployment of security applications and solutions for the cloud environment is complicated. Therefore, risk assessment of the cloud and effective management is a challenging research problem. To minimise the security concerns of customers and to convince them to adopt IaaS,

risks and their severity levels must be known to make informed security decisions.

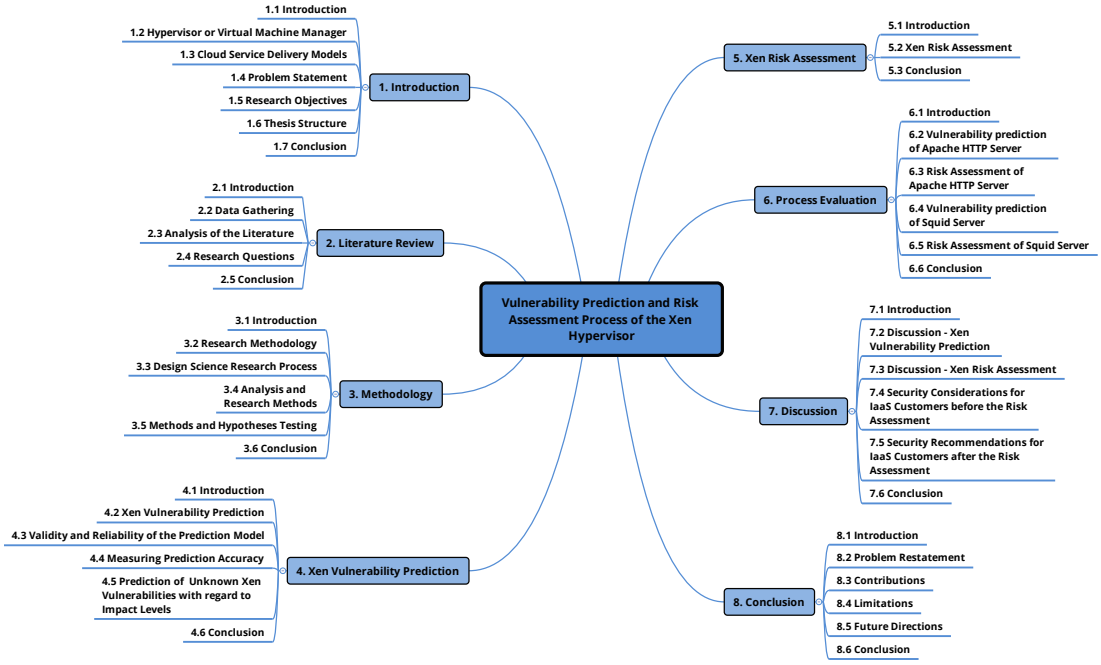
This research targets IaaS and proposes a Xen risk assessment process for customers. The Xen risk assessment process provides the risks and their severity levels to customers' data and information when they move to cloud virtualised infrastructure. In other words, it offers a platform for customers to use the outcomes of risk assessment to make informed security decisions to select an appropriate CSP which is in compliance with security controls and reduces the concerns of customers. The risk assessment process consists of the following key steps:

- Identification of hypervisor vulnerabilities and determination of impact ratings. The process compliments ENISA's risk framework (Catteddu & Hogben, 2009) to adopt impact ratings of the hypervisor vulnerabilities.
- Identification of threats and determination of likelihood levels using a structured analysis approach by developing attack trees. Threat likelihood assessment is performed using two different threat actors (TAs).
- Nine main risk categories to the Xen hypervisor are identified. The mapping of vulnerability impact ratings and threat likelihood levels is performed using a risk estimation matrix to determine risk severity levels.

Different organisations who are planning to move to cloud infrastructure can adopt the Xen risk assessment process. This can enable customers to make informed security decisions to select an appropriate CSP. Nine different technical and generic hypervisor risks are identified, and their severity levels are determined. However, organisations are encouraged to identify and add a new type of risk in the assessment process that may be specific to their services and data type.

1.6 Thesis Structure

In this section, the structure of the thesis is presented.



Chapter 2 presents a systematic literature review (SLR). The SLR directs this research to identify, evaluate, and integrate the findings of existing research to address the initial research questions. The problem is investigated by establishing the context of existing research, relationships, possible areas of research, and inconsistencies in the literature. It further leads to evaluating, developing and extending a theory to address the research gaps.

Chapter 3 provides the methodology to conduct this research. It details the research questions and hypotheses. It also provides the research methods selected to test these hypotheses. A vulnerability prediction and risk assessment process for Xen is in Chapter 3.

For Chapter 4, Xen vulnerability prediction is covered. Unknown Xen vulnerabilities

are predicted for 2018. Unknown Xen vulnerabilities are also predicted with regard to the impact levels to extend the scope of the vulnerability prediction process.

Chapter 5 provides the Xen risk assessment process. Risk assessment is performed to determine vulnerability impact ratings, threat likelihood levels, and severity levels of risk to Xen. The research methods are applied, and all the hypotheses are tested to address the research questions.

For Chapter 6, vulnerability prediction and risk assessment of the Apache HTTP and Squid Proxy servers is performed to evaluate the process to ensure its generalisability and applicability. The evaluation process shows accurate results by successfully predicting unknown vulnerabilities and determining the risk severity levels to Apache and Squid.

Chapter 7 provides a discussion of analysis of the results. It also details how results support the answers to the research questions and address the research gaps.

Chapter 8 provides the conclusion of the thesis. It restates the problem and highlights the research contribution made to the body of knowledge through this research. It also highlights the limitations and future research directions.

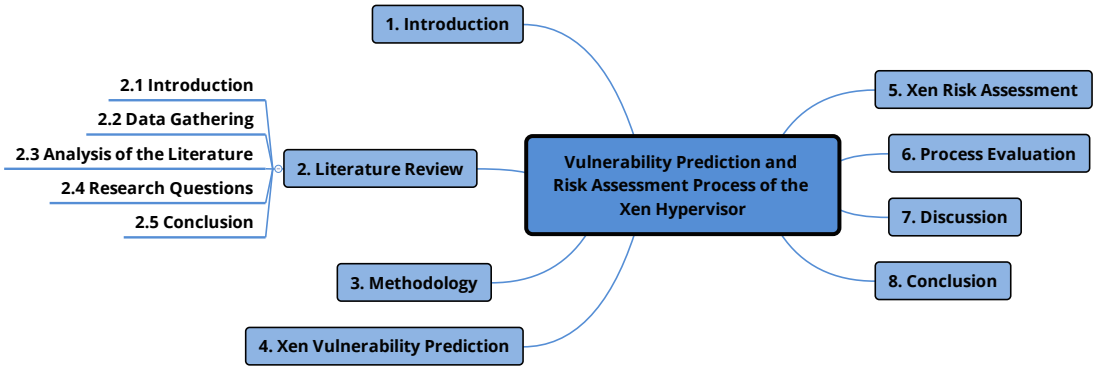
1.7 Conclusion

The hypervisors are key components for the CSPs to offer IaaS. However, the new types of risk to hypervisors raises security concerns and results in lack of IaaS adoption by many organisations. The existing research focuses risk assessment from a broader perspective. However, risk varies between cloud service delivery models thus making a generic risk assessment framework ineffective. Moreover, in IaaS, customers are unaware of the new risks to their data and information once they move to cloud virtualised infrastructure. Therefore, this research presents a Xen vulnerability prediction and risk assessment process. Through this process, customers can predict unknown

vulnerabilities, understand threats and their likelihood, and determine risk and their severity levels to the Xen hypervisor. It enables organisations to select an appropriate CSP by comparing the risk assessment results with the security controls and procedures implemented by a proposed CSP. The intended audience for this research is the organisations that are planning to move to cloud virtualised infrastructure but do not have enough security professionals and resources to assess the risk to their data and information after they move to IaaS. The next chapter provides an SLR which highlights the problem area by identifying the possible research areas from the literature. It also provides a theory to address these research gaps.

Chapter 2

Literature Review



2.1 Introduction

In Chapter 1, an introduction, problem area, research objectives, and structure of the thesis are presented. This chapter presents an SLR conducted to review the existing research and identify possible areas of research. According to Wooley (2011), a "literature review provides a meaningful context of a project within the universe of already existing research" (p. 27).

An SLR is a process or method to review the literature by utilising the standard and

pre-specified techniques. For the SLR, the research bias is minimised before conducting the review by preparing the rationale, research hypotheses and research methods. The hypotheses and research methods guide one to perform the literature review process. The outcome of the SLR is to identify, analyse, and summarise the existing research and evidence concerning a research problem. Furthermore, an SLR allows one to identify themes that require further exploration. Moreover, an SLR is considered as one of the best methods to synthesise an evidence specific research question.

2.1.1 Why SLR?

An SLR is slightly different from traditional literature review techniques. It allows researchers to search and select all the articles related to a specific research question (RQ). It uses a methodology that is developed to minimise the effect of selection, publication, and extracting information bias (Nightingale, 2009). It aims to identify the relevant material which addresses a specific RQ to give a fair and unbiased summary of the literature. The SLR is conducted by following the method by Kitchenham et al. (2009). The method describes a process to analyse the literature search results, identify research gaps, formulate RQs, and propose a solution to address the apparent gaps. Initial search questions are developed along with the inclusion/exclusion criteria to collect data.

This chapter is organised as follows: Section 2.2 covers the data gathering process along with the initial search questions to perform a literature search. It also provides literature search criteria to select the research articles. A literature review analysis in Section 2.3 includes the details of threats, vulnerabilities, and risks to hypervisors. It also covers details of the vulnerability and the risk assessment process. Section 2.4 provides the RQs. The conclusion of this chapter is provided in Section 2.5.

2.2 Data Gathering

This section provides the literature review search questions, literature search criteria, and inclusion and exclusion criteria. The process of data collection and extraction is also discussed in this section.

2.2.1 Initial Search Questions

The literature search is based on two initial questions.

SLRQ1 Do hypervisors make cloud computing secure or vulnerable?

SLRQ2 Is using hypervisor software putting cloud assets at risk?

Peer-reviewed journals, conference papers, book chapters and published theses since 2009 define the scope of the literature search. The literature search criteria in Section 2.2.2 is used to select research articles for SLRQ1. For SLRQ2, articles are considered from 2010 onwards, as articles published prior to 2010 lack the focus of CC risk assessment.

2.2.2 Literature Search Criteria

In this section, a literature search criterion to select the research articles is provided. An extensive manual search process is performed using the keywords extracted from the initial RQs. Academic and peer-reviewed articles are selected which cover hypervisor security, threats and vulnerabilities to hypervisors, and risk assessment and risk management of CC. The articles which cover risk assessment in general instead of CC are not included in the review. The selected research articles, book chapters, and theses are in English and published after 2009.

Some popular databases such as Science Direct, Elsevier, Springer Link, IEEE, and ACM were searched to select the research articles. These databases were selected

because the information systems focus encompasses risk assessment and management in CC. The literature search produced a good number of articles; however, only 47 research articles were found relevant after applying the search criteria. The rejection decisions were made due to their lack of relevance to SLR questions after reviewing the abstract and introduction sections of these articles.

Out of these 47 articles, nine more articles were rejected due to low methodological quality. The distribution of these 38 articles which were selected for SLR is: 19 articles were from conferences and book chapters, 18 from journals, and one dissertation. Furthermore, 21 articles were relevant to SLRQ1 and 17 articles were relevant to SLRQ2. Figure 2.1 shows the number of research articles relevant to each literature search question.

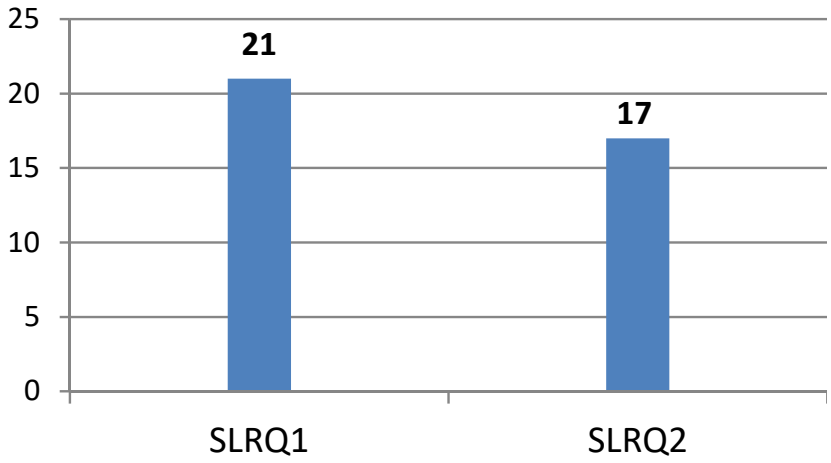


Figure 2.1: Articles in Relation to SLRQs

Figure 2.2 depicts the distribution of selected articles per year. It is notable that a fair number of research articles were published from 2010 onwards, which makes the research problem worthy of interest.

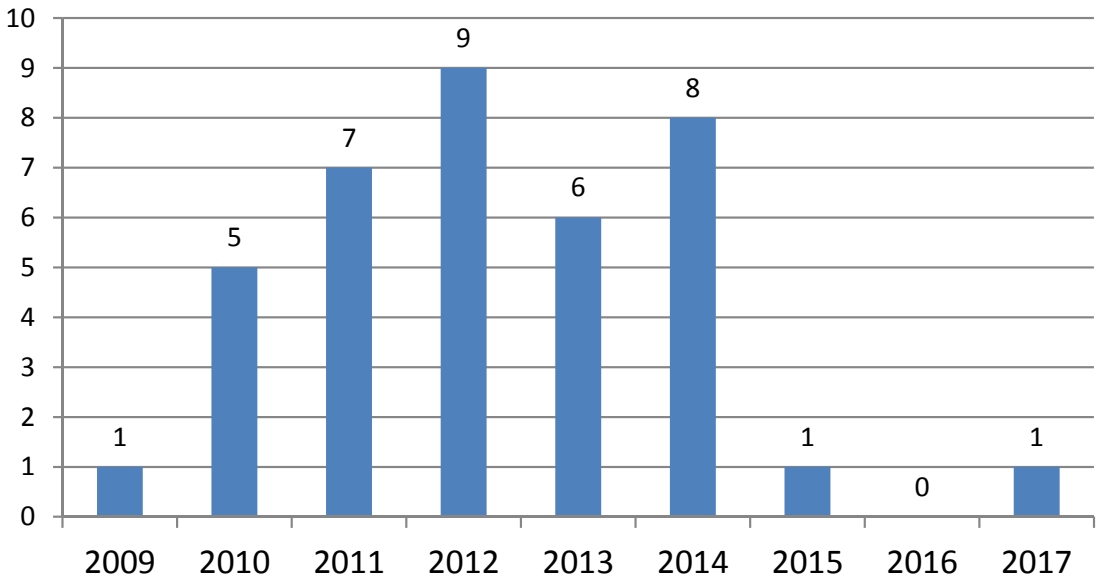


Figure 2.2: Selected Articles Per Year

Nine articles were not selected for the review. The main reason for rejection was low methodological quality and relevance to the SLR questions. Table 2.1 provides a summary of the articles and also the reasons for not including these articles for the review.

Table 2.1: Summary of the Articles Not Selected for Review

Author	Year	Article Title	Reason for rejection
Joh et al.	2008	Vulnerability Discovery Modelling using Weibull Distribution	Uses Weibull distribution which has not been used for modelling vulnerability discovery so far. Thus, the accuracy of the results is questionable
Novak et al.	2010	Taxonomy of Static Code Analysis Tools	Provides taxonomy of static code analysis tools but does not provide the uses of static tools to discover vulnerabilities
Sommestad et al.	2012	Efforts estimate for vulnerability discovery projects	Estimates only the efforts required for a tester to find zero-day vulnerabilities, instead of discovering vulnerabilities themselves

continued . . .

Summary of the Articles Not Selected for Review ...continued

Author	Year	Article Title	Reason for rejection
Johnson et al.	2016	Time between vulnerability disclosures: A measure of software product vulnerability	Only provides a measure to capture the time between vulnerability disclosure instead of the number of vulnerabilities
Sgandurra and Lupu	2016	Evolution of Attacks, Threat Models, and Solutions for Virtualized Systems	Categorises threat models and attacks for virtualised systems but not the risk
Chhabra and Taneja	2011	Cloud Computing: Towards Risk Assessment	Provides only an overview of different risk that exists at different layers of CC instead of a risk assessment model
Kholidy et al.	2016	A Risk Mitigation Approach for Autonomous Cloud Intrusion Response System	Focuses on detection of network-based attacks to provide risk assessment and mitigation capabilities. The paper lacks CC focus.
Chatzipoulidis et al.	2015	Information Infrastructure Risk Prediction through Platform Vulnerability Analysis	Measures zero-day risk using risk prediction methodology for information infrastructure. This paper lacks focus on cloud-based infrastructure
Anand et al.	2016	Threat Assessment in the Cloud Environment – A Quantitative Approach for Security Pattern Selection	Provides only threat assessment and does not cover risk assessment of CC

2.3 Analysis of the Literature

This section covers a detailed review of hypervisor security to determine whether or not hypervisors make CC secure (SLRQ1). It also provides an assessment to find out how the hypervisors put cloud virtual assets at risk (SLRQ2).

Guest OS isolation and resource sharing between multiple VMs are the main functionalities which a hypervisor provides. Hypervisors offer these functionalities like

a standard OS which provides isolation between different programs and applications running on the server. Similarly, the hypervisor provides isolation between multiple VMs running on the same host server. The hypervisor also leverages access to hardware for different VMs similar to an OS to ensure the process isolation. It also manages the mediation of access to devices by calling modules either from the host OS kernel or running in dedicated VMs such as Dom0 (in the case of Xen).

2.3.1 Hypervisor Threat Sources

The hypervisor allows VMs to access hardware resources of the server and ensures the isolation between all VMs. VMs access hardware resources such as CPU and memory controlled by the hypervisor. However, VMs access to network and storage devices are managed through driver modules that are available in the host OS kernel module or in Dom0 (a privileged VM). The network isolation between different VMs is provided by assigning a unique MAC or IP address to each VM. The VLANs can be defined which enables each VM to have an appropriate network ID.

A Xen hypervisor runs on top of hardware of the host server which is connected to an enterprise network. It also allows the execution of multiple VMs that are generally connected through a virtual network inside a hardware server. However, in some cases, VMs can be part of an isolated network or share the host server's network. Three different threat sources are identified for a Xen based virtualised infrastructure:

Threat Source 1 Threats from or through the enterprise network where the hypervisor host server is located.

Threat Source 2 Threats from malicious VMs running on a hypervisor through channels such as shared Cache memory and virtual network inside the host server.

Threat Source 3 Threats from hypervisor management interfaces and VM management daemon.

Threat source 1 is common to all types of hypervisors; however, threat sources 2 and 3 are unique to a virtualised environment defined by a respective hypervisor. Therefore, threats from sources 2 and 3 are considered in this SLR.

2.3.2 Threats and Vulnerabilities to the Hypervisors

In this section, an overview of vulnerabilities and threats to hypervisors is presented. Hypervisors seem vulnerable to sophisticated threats (Brohi, Bamiah, Brohi & Kamran, 2012; Bazargan, Yeun & Zemerly, 2012; Sabahi, 2011) as the vulnerability reporting and exploitation rate is increasing (Litchfield & Shahzad, 2017). By exploiting vulnerabilities, an attacker can gain unauthorised access to a hypervisor to control and exploit other customers' VMs (Khorshed et al., 2012; Modi et al., 2013). Infrastructure sharing is a core cloud service but because of hardware sharing as its limitation, lacks basic mechanisms to protect customer network traffic, data and other applications. This limitation provides an attacker with a chance to hijack user credentials and eavesdrop on information to control other users' VMs. So, a hypervisor can leverage different threats that result in compromise of CIA of the information (Dawoud, Takouna & Meinel, 2010). The threats and vulnerabilities to hypervisors are as follows.

Authentication, Authorisation and Accounting Vulnerabilities

An attacker can gain unauthorised access to a shared hardware resource if weak authentication, authorization, and accounting (AAA) policies are in place (Modi et al., 2013). An attacker can get access to resources by using customers' weak or insecure credentials, and credentials stored on a guest machine. Furthermore, password-based authentication attacks are not new. However, their impact on a cloud environment is

higher due to the fact that organisations are running cloud-based corporate applications which are exposed to the internet. Therefore, stronger or multi-factor authentication to assess cloud services is needed.

Lack of Resource Isolation

The hypervisor allows sharing of physical hardware re- sources among different VMs that usually belong to different IaaS customers. However, vulnerabilities in hypervisors can lead to unauthorised access to the shared hardware resources used by other VMs. A hypervisor in IaaS allows CSPs to develop and share a proprietary VM management interface among customers. However, vulnerabilities in the management interfaces can result in unauthorised access to information of other customers. Furthermore, it may allow an attacker to manipulate hypervisor functionality and cause a DoS. DoS vulnerability allows an attacker to make the cloud services unavailable to customers by occupying the computational resources of the host server (Djenna & Batouche, 2014; Luo et al., 2011; Khorshed et al., 2012; Ayala et al., 2013). For example, through DoS, an attacker can create a large number of malicious VMs to occupy server resources required by legitimate VMs (Kazim et al., 2013). It may also lead to a data breach or theft through a hypervisor escape vulnerability where an attacker can use a malicious VM to acquire root level access by breaking or escaping the barrier provided by the hypervisor. The VM escapes the isolation layer when an attacker injects an undetectable malware code by using a malicious VM which seems like a legitimate VM to the hypervisor (Modi et al., 2013; Pék et al., 2013; Shoaib & Das, 2014; Pearce et al., 2013; Luo et al., 2011). Once the attacker has control over the hypervisor, all other VMs and their data can be accessed. Lack of security controls in the cloud to check co-residence makes it is possible to map the cloud infrastructure to find out where a particular target VM resides (Ristenpart, Tromer, Shacham & Savage, 2009). Afterwards, two VMs are co-located on the shared hardware, so the extraction of keystroke timing information

(since the cache is shared) is possible using an access-driven side channel attack.

Network Probing

IaaS customers that are part of the same subnet can perform net- work mapping and port scanning to locate a target VM. The primary function of the hypervisor is to provide isolation and prevent one VM monitoring other VMs' access to shared hardware resources (Dawoud et al., 2010). However, a vulnerability in the hypervisor allows a malicious user to compromise the virtual switch configuration, VLANs configuration, and ARP tables to monitor the network traffic of other VMs (Ibrahim, Hamlyn-harris & Grundy, 2010). Once the target VM is compromised, all other VMs running on the same hypervisor are also vulnerable to the same attack due to sharing the same virtual switch (Wooley, 2011).

Side-channel Vulnerabilities

Side channel vulnerabilities allow an attacker to exploit the information obtained through the use of shared hardware resources (Vaquero et al., 2011). This may include the CPU cores and high-level cache memory. In a typical Cross-VM side channel attack scenario, an attacker can monitor the cache access behaviour of the victim's VM. For example, cache timing information is obtained by measuring the execution of different operations of the victim's VM (T. Zhang & Lee, 2014). Once the actions of the victim's VM are identified, useful information such as cryptographic keys can be extracted (Y. Zhang et al., 2012).

Eavesdropping Communication between VMs and Host

A hypervisor acts as a middleware between the guest VM and hardware resources of the system, for example, disk I/O and NIC. All the communication from a VM to another

VM and from a VM to the Host (Dom0) passes through the hypervisor (Dawoud et al., 2010). All the traffic commuting through the hypervisor can be monitored by an attacker by injecting malicious code to eavesdrop the communication and gain control of the VMs (Wooley, 2011).

Physical Attacks

Physical attacks on the server systems are difficult to execute as compared to the software attacks. However, the consequence of a successful physical attack is greater because it can allow an attacker to gain full control of the secure server. Therefore, physical attacks should be given consideration like other sophisticated attacks such as DoS. The CSPs implement state-of-the-art physical access controls and surveillance mechanisms to mitigate physical attacks. However, these security controls and mechanisms may only prevent unauthorised access to the servers and seems ineffective to mitigate physical attacks launched by a malicious insider (Szefer, Jamkhedkar, Perez-Botero & Lee, 2014).

Malicious insider attack on the cloud enviornment is not given a careful attention. In CC, physical attacks by the malicious insiders to compromise the CSPs' infrastructure is a prime target by many criminal organisations. Moreover, most of the physical attacks are carried out by the malicious insiders who are authorised to access the servers, thus makes these attacks very difficult to detect and prevent. The CSPs can use different techniques such as, migrate the data away from the local storage of the servers, encrypt the data that is stored on the servers, and backup the data from the servers to avoid unauthorised access to mitigate a physical attack (Szefer et al., 2014). However, more efforts are required to mitigate malicious insider attacks instead of just implementing physical security solutions. These technological solutions seem ineffective to detect malicious insiders. Therefore, there is need of proactive actions by the CSPs to mitigate malicious insider attacks. For example, the CSPs needs to perform the background

checks before hiring an IT employee, educate IT staff, use proper termination practices, and use of monitoring solutions to mitigate the risk of physical attack by a malicious insider.

Inadequate Physical Security Procedures at CSPs End

Security in the cloud is not virtual (Rackspace, 2017). There are a host of physical controls that must be in place to ensure the security of the cloud. Though the cloud is virtual, data and information are still placed on hardware servers in different locations. The data centres must be highly secured by placing some security measures to ensure the security of the data. Also, multi-factor authentication should be used to ensure that only authorised personnel have access to the data centres. Furthermore, sensitive equipment such as cloud servers and storage equipment should be secured in a sub-area within a secure perimeter with additional security controls. Physical break-ins to data centres are not new. Matis (2017) highlights some of the break-in cases. These break-ins could have been avoided by implementing better security controls. For example, hard disk drives were stolen from an insurance agency's data centre due to weak physical security which resulted in a breach of around two million member's personal information.

Remote Access to Management Interfaces

The hypervisor management interfaces present danger. Through management interfaces, the administrators can start, stop, suspend, and migrate VMs running on a hypervisor. If an attacker can gain unauthorised access to the management interface, the VMs can be stopped, deleted or the data stored on these VM can be stolen. This type of vulnerability results in a high impact of the virtual asset (Catteddu & Hogben, 2009).

In a nutshell, the above threats and vulnerabilities raise concerns about the security of hypervisors. The literature review results show that vulnerable hypervisors put cloud virtual assets at risk.

2.3.3 Vulnerability Assessment

This section provides an overview of the vulnerability assessment process. Protecting large software packages from exploitation is an ongoing process, where software vendors protect their software by patching them. On the other hand, an attacker tries to exploit loopholes in these software packages. Unknown software vulnerabilities are the main reason for security concerns. Ignoring these unknown vulnerabilities results in more sophisticated threats to the software packages. By considering the increased number of exploitation scenarios and the ever-growing threat of breaches, the importance of software security has dramatically increased over time (Roumani, Nwankpa & Roumani, 2015). Like other software packages, hypervisors are also vulnerable and contain vulnerabilities (Catteddu & Hogben, 2009).

Xen is a standard open source hypervisor with thousands of lines of code (Perez-Botero et al., 2013). This makes it very difficult to analyse the Xen code to discover unknown vulnerabilities using different techniques such as static code analysis, and Vulnerability Discovery Models (VDMs). Static code analysis, also called static code inspection, is a manual process to evaluate the software code to find out vulnerabilities. It is a traditional approach to go through the code to find coding errors which can result from vulnerability exploitation (Liu, Shi, Cai & Li, 2012). It analyses the code without executing the software code which takes less time but requires expertise from the tester (Khan, 2014). Different static code analysis tools are now available to help testers to find the data and control flow. However, static code analysis techniques are not feasible for large software applications due to false positives and false negatives. Moreover, static code analysis looks for patterns which can lead to vulnerability. If there are no patterns, then static code analysis will not be able to find the vulnerability (Shah & Mehtre, 2015). Furthermore, static analysis mostly depends on the source code and does not identify design bugs.

VDMs are based on Software Reliability Models (SRMs). VDMs are quantitative methods to predict vulnerabilities that may exist in a software system (Alhazmi & Malaiya, 2005b). VDMs are also used to determine the resources that are required to assess a particular software. According to Alhazmi and Malaiya (2005a), VDMs can describe the rate of vulnerability discovery or the cumulative number of vulnerabilities discovered in time. The output of VDMs is an estimate of the total number of vulnerabilities and the mean time to the next vulnerability (Liu et al., 2012). Some VDMs are based on assumptions and need to be validated. Furthermore, one VDM can specifically be used only for one software. Also, some VDMs lack acceptance from the testing community.

Vulnerability Prediction

Finding vulnerabilities using static code analysis and VDMs for a large software such as a hypervisor is difficult and time-consuming. This raises the question that *How can the unknown vulnerabilities be predicted in large software applications such as the Xen hypervisor to mitigate exploitation scenarios?* [1]

This research presents a vulnerability prediction process to address this RQ. Unknown vulnerability prediction is a better approach compared to discovering vulnerabilities (Last, 2015). The prediction models reduce the time and effort required to mitigate vulnerabilities (Walden, Stuckman & Scandariato, 2014). Much research has been produced in the last decade, and different prediction models are proposed. However, the existing vulnerability prediction models are not mature and need improvements. Overall, the field of vulnerability prediction needs to evolve along a similar trajectory (Walden et al., 2014). There are few studies that also compare different prediction models using a different data type for the prediction; however, there are no standard data sets to be used for prediction by the existing prediction models.

[1] The first RQ which is addressed through this research.

2.3.4 Risk to Hypervisors

In this section, risk to hypervisors are presented. The risks to hypervisors vary between the cloud service delivery models. The risks are different due to the type of hypervisor, security controls and procedures, and risk management methods implemented to protect these hypervisors. The new and unknown risk to hypervisors discourages customers from adopting cloud virtualised infrastructure. Some of the technical risks that vulnerable hypervisors pose to virtualised infrastructure are discussed as follows.

Loss of Business Reputation Due to Co-Tenant Activities

Resource sharing provides a platform for malicious tenants to carry out unauthorised activities to affect the reputation of other tenants (Wang, Liu & Liu, 2012; Cayirci, 2015). Examples are port scanning by exploiting a virtual switch of the hypervisor, spamming, and serving of malicious content from cloud infrastructure. It can lead an attacker to block IP addresses range that affects all other tenants, and also seizure of resources due to neighbour activities. These malicious actions can impact the delivery of cloud service, data loss, and loss of an organisation's reputation.

Isolation Failure

Hypervisors allow CSPs to create a multi-tenanted architecture to generate more revenue by using the infrastructure effectively (Shoaib & Das, 2014). However, there is a risk of resource isolation failure (Saripalli & Walters, 2010; Wang et al., 2012) when a malicious tenant manipulates the hypervisor (using VM escape, SQL Injection and Cross VM Side Channel) to break the isolation layer. Moreover, the likelihood of resource isolation failure depends on the cloud model and its level increases from a private to a public cloud type.

Malicious Insider

A malicious insider can be a rogue employee (an IT administrator in this case), an IT contractor, or an IT business partner of the CSP who can misuse physical access to the servers to compromise the CIA security objectives (Samani, Reavis & Honan, 2014). In some cases, a malicious insider attack can be realised by a former employee who is still able to maintain access to the servers. A case has been reported by Schwartz (2011) where a Virtual Private Network (VPN) token is maintained by an ex-employee who was accused of using a VPN to access the former employer's network and deleting virtual servers, taking a storage area network offline, and deleting mailboxes from the corporate email server (Samani et al., 2014).

A malicious insider poses a high risk to CSP's infrastructure (Alva et al., 2013). The severity level of a malicious insider risk is high for the cloud infrastructure because the data, information, and services that can be compromised by the malicious insider normally belongs to cloud service consumers. Furthermore, a physical attack executed by a malicious insider can affect the CSP's reputation that could result in loss of customers' trust.

Intercepting Data in Transit

As compared to a traditional data centre environment, CC implies more data in transit due to synchronising the images of multiple distributed machines between the cloud and web clients. Data transit scenarios are vulnerable to side channel, sniffing, spoofing, and man-in-the-middle attacks in a cloud environment. Also, the lack of clarity of security controls and procedures implemented by CSPs raises concerns about the circulation of confidential information within the cloud (Cayirci, 2015; Saripalli & Walters, 2010; Wang et al., 2012).

Data Leakage within the Cloud

Data communication between a CSP and a cloud customer is continuously at risk. Overall, the characteristics of this risk are the same as the risk of intercepting data in transit (previous risk) (Sinanc & Sagiroglu, 2013; Khan, Oriol, Kiran, Jiang & Djemame, 2012; Wang et al., 2012).

Undertaking Malicious Probes or Scans

This risk is related to malicious probes, and network mapping (Khan et al., 2012; Wang et al., 2012). However, it is not a direct threat to cloud assets, but these threats can be used to gather information to realise further attacks such as hacking. This type of risk could result in loss of CIA of the cloud data.

Compromise Hypervisor

A hypervisor sits on top of hardware and provides various levels of abstraction. It manages the customers' resources provided by the CSP through IaaS. A hypervisor is an application with a large code base and has vulnerabilities like other software packages. An attacker can use a malicious VM to exploit a hypervisor vulnerability and obtain high-level privileges to access data and information that belongs to other tenants (Kazim et al., 2013; Pék et al., 2013).

Privilege Escalation

This risk refers to gaining high-level privileges by evading an authentication process (Pearce et al., 2013; Wang et al., 2012). An attacker can gain unauthorised access to the system by exploiting a hypervisor's vulnerability to execute operations like an authenticated user. Also, an attacker can use these privileges to monitor and modify data in a transparent way. An attacker can also allocate more hardware resources to a

malicious VM to cause a DoS for all other VMs running on top of the compromised hypervisor.

Management Interface Compromise

CSPs provide customer management interfaces access to their administrators through the internet. Usually, administrators are able to access a large set of resources and thus pose an increased risk. The severity level of risk even increases when combined with remote access and browser vulnerabilities. Therefore, the management interfaces should be secured from network and web attacks.

2.3.5 Risk Assessment

This section details the risk assessment process in CC. Risk assessment can be performed to prevent and mitigate threats, adverse actions, and attacks to quantify the risk that poses severity levels above the acceptable threshold. Customers as owners of data and information are responsible for security once they move to cloud-based services. However, the level of customer control varies in each cloud service delivery model. For example, in IaaS, customers only have control over the OS, programs, and applications, whereas, the management of virtualised infrastructure is the responsibility of the CSP (Albakri et al., 2014). Usually, CSPs ensure the security of the cloud virtualised infrastructure to meet the needs of all the customers and require fewer changes or customisation. A CSP's selection and acceptance of its security controls and procedures considers their efficiency, effectiveness, and limitations based on the policies, standards, laws, and rules and regulations, with which a CSP must comply. However, CSPs do not consider the security requirements of individual customers and project security requirements as a generic core set for a large number of customers. Customers are happier to accept the risk when they have more control over the data and infrastructure.

A higher level of control allows customers to weigh alternatives, set priorities and make informed security decisions when dealing with risk. For the successful adoption of cloud infrastructure services, the most important thing for customers is to understand the characteristics of these services. Customers should also have knowledge of architectural components, type, and actors of each service to have a secure cloud environment. Therefore, customers should have the ability to identify the risk and all the required security controls. Customers should also have the knowledge to request CSPs about the implementation of security controls and procedures (Cayirci, 2015).

The existing research focuses on risk assessment from a broad perspective instead of risk assessment of virtualised infrastructure or the hypervisor. Moreover, the existing research assesses the risk from a CSP's perspective and lacks a customer's perspective. Saripalli and Walters (2010) present a quantitative impact and risk assessment framework for CC based on six key categories. The categories are confidentiality, integrity, availability, trust, mutual audit ability, and usability. Similar work is presented by Fitó and Guitart (2014) on the basis of impact on business objectives. However, both cases focus on the business objectives of CSPs and not customers. Leitold and Hadarics (2012) present a mathematical assessment model (a directed graph and a matrix to discover risk) for threats. It considers the communication of risk for separate entities and calculates risk for the target infrastructure. However, the model does not specify the cloud TAs. Also, justification is not provided for the adaptability of the assessment model and how the model can collaborate in a real environment. This raises the second question: *How can the determination of the risk of unknown Xen vulnerabilities be presented such that it aids cloud infrastructure service consumers?*[2]

[2]The second RQ which is addressed through this research.

Risk Assessment of Cloud Infrastructure from a Customer's Perspective

In IaaS, security of virtualised infrastructure falls outside of the control of customers. Customers are unaware of the security of their data and information if they move to cloud infrastructure. Customers' understanding of different cloud service delivery models and the risks associated with each service model is essential. Risk type varies between service delivery models which make it challenging for customers to assess the risk. Also, a generic risk assessment framework cannot fit all service delivery models. Therefore, customers need to perform a thorough risk assessment for the service delivery model they have adopted to accurately identify the risk and required security controls and procedures as part of the risk mitigation strategy.

It is vital for customers to identify the risk and security controls required to mitigate these risks. A customer's decision to move to cloud virtualised infrastructure depends on their accurate identification of security requirements, risk, analysis of each perspective of CSPs' security controls and procedures, and clarity about the Service Level Agreement (SLA) to build trust with the CSP. An accurate and thorough risk assessment platform would help customers to understand the security of cloud infrastructure services along with adequate guidance on SLAs to make informed security decisions to adopt these infrastructure services.

This research addresses these questions and presents a vulnerability prediction and risk assessment process for customers to help them make IaaS adoption decisions. This research targets the Xen hypervisor, a standard open source hypervisor which is in use by many large CSPs such as Amazon Web Services (AWS) and Rackspace.

Almost all the existing work views the risk from a broader perspective instead of targeting hypervisors which provides the basis for IaaS. We fail to see a concise framework to perform the risk assessment IaaS service delivery model. A generic risk assessment is unsuitable due to different risk and security controls in each of the service

delivery models. Therefore, an adequate risk assessment platform is necessary to assist customers to review the risks and their severity levels.

2.4 Research Questions

In this chapter, SLR develops an understanding of the vulnerability prediction and risk assessment. Almost all the existing work views the risk from a broader perspective instead of targeting the hypervisor which provides the basis for IaaS. Therefore, a need for an effective vulnerability prediction and risk assessment platform is necessary to assist IaaS customers to make informed security decisions. So, to realise a vulnerability prediction and risk assessment process for IaaS, the questions below must be answered.

RQ1 How can the unknown vulnerabilities be predicted in large software applications such as the Xen hypervisor to mitigate the exploitation scenarios?

RQ2 How can the determination of the risk of unknown Xen vulnerabilities be presented such that it aids cloud infrastructure service consumers?

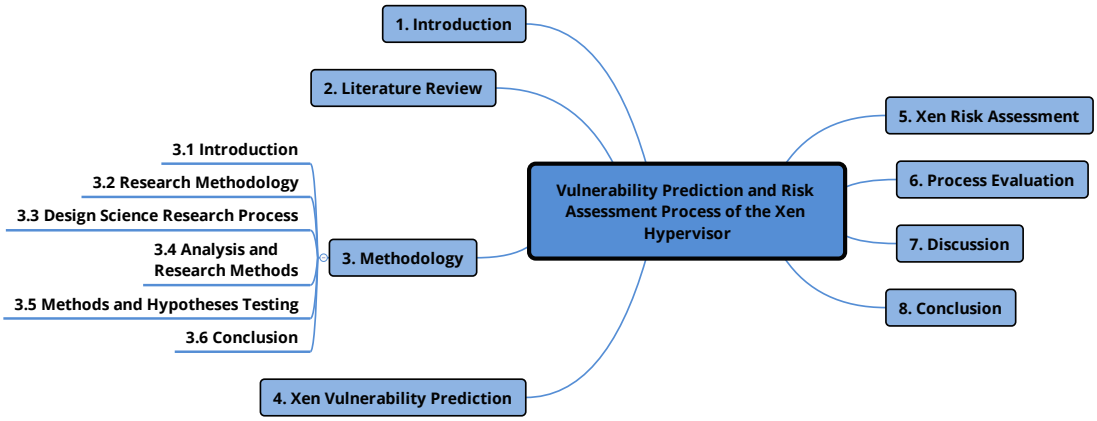
2.5 Conclusion

Despite a lot of encouraging existing research and different risk methodologies and frameworks, we fail to see a concise method or framework for analysing and assessing security risk to IaaS and specifically the hypervisors. The scope of the existing research is broad and also lacks focus on IaaS. Therefore, to realise a risk assessment process for IaaS, the review led to RQs. It is evident that risk to virtualised infrastructure raises more concerns from customers which affect the cloud adoption rate. Therefore, to allow customers to make informed security decisions, a need for a hypervisor risk assessment process arises.

Chapter 3 covers the research methodology adopted for this thesis. It also details the different research methods used to test the hypotheses to answer the RQs. Time Series Holt-Winters and Regression Analysis methods are used to predict the unknown Xen vulnerabilities. For risk assessment, the Common Vulnerability Scoring System (CVSS) is used to score the vulnerabilities to determine the impact ratings. ENISA's risk assessment framework is also considered to adopt the impact ratings of hypervisor related vulnerabilities to realise a complete risk assessment process. A structured analysis approach is used to identify threats and their likelihood levels. Risks and their severity levels are determined using a risk estimation matrix.

Chapter 3

Methodology



3.1 Introduction

In Chapter 2, an SLR was presented which highlights that common hypervisors are vulnerable to sophisticated threats and pose a risk to assets. The literature review also shows that hypervisors bring new risk to CC compared to the traditional IT environment. Therefore, it is necessary to assess hypervisors for vulnerabilities and risk to assist customers to make informed security decisions.

This chapter presents DSR methodology as the main methodology to conduct this

research. DSR as a methodology is concerned with the design, development, and improvement of the solution artefact. DSR provides the best fit where a model is supported by research methods that guide the implementation of the model. This research provides a solution artefact (a process) which is analysed, justified, and evaluated. Therefore, DSR is considered as the main methodology because the focus of this research is the construction of the solution artefact and not an intervention.

This chapter is organised as follows: Section 3.2 provides the rationale of the DSR methodology. Section 3.3 presents the DSR process. Section 3.4 covers the research methods. Section 3.5 provides the testing of hypotheses and a conclusion of this chapter is provided in Section 3.6.

3.2 Research Methodology

This section provides the detail of the research methodology used for this research. DSR gained momentum as a research methodology to be used for IS research in 2004 (Hevner, March, Park & Ram, 2004). It addresses a research problem in more efficient ways and focuses on constructing and evaluating solution artefacts to that research problem. In DSR, a design is referred to as a process (a set of activities) and a product (artefact).

DSR guides to construct artefacts to offer utility. DSR creates and evaluates IT artefacts to solve identified problems (Hevner et al., 2004). It provides effective processes to identify a research problem, and design an artefact as a solution. It also allows researchers to contribute and evaluate the solution designs, summarise, and present the results. It is a solution-oriented methodology which contains a rational decision cycle applied to answer knowledge questions. Initially, there was an argument about whether DSR is an effective methodology to research an IS discipline. However, recent research suggests that DSR is an accepted methodology and an IS research paradigm by

integrating solution design as a key process of the overall research (Offermann, Levina, Schönherr & Bub, 2009).

The DSR process makes sure that the developed artefact should address a research problem and must be developed through a search process based on existing research. It should also highlight the contribution of the artefact developer. Later, artefacts should be evaluated to ensure the utility, quality, and efficiency of the designed solution. Also, the research contributions should be highlighted and communicated to the target audience (Hevner et al., 2004). This research also constructs an artefact which is evaluated, refined, and justified to address the research problem. This research is rigorous as it provides a vulnerability prediction and risk assessment process for organisations who are concerned about the security of the cloud virtualised infrastructure. DSR in IS disciplines is now an accepted research paradigm and approach. Henver and Chatterjee (2010) provides the following guidelines (Table 3.1) for the DSR.

Table 3.1: Guidelines for DSR

Guideline	**Description**
Problem Identification	A problem area must be identified to provide a solution artefact.
Artefact Design	A viable artefact must be produced by the research conducted.
The relevance of the Artefact to the Problem	The objective of the research is to develop an artefact as a solution to address the identified problem.
Evaluation of the Artefact	The solution artefact must be assessed, tested, and justified. The solution should provide the utility.
Research Contributions	The research must provide a clear contribution to the body of knowledge.
Research Rigour	Rigorous methods should be applied for the development and evaluation of the solution artefact.
Communication of the Research	DSR should be presented effectively to both technology and business-oriented audiences.

3.2.1 Motivation for considering DSR as a Methodology

This research involves the development of a vulnerability prediction and risk assessment process to address the research problem. The DSR methodology is adopted because the outcome of this research is aligned with what DSR offers. So, firstly literature is searched, and the problem area is identified. Hypotheses are then developed to be tested to address the problem. The research approach leads to data collection and analysis to test hypotheses and eventually address the research questions. This research is more closely suited to the DSR process because it focuses on solving a research problem by developing a solution artefact as an output. Three phases of DSR have helped this research and development of the vulnerability prediction and risk assessment process. The iterations between phases have guided this research to evolve the vulnerability prediction and risk assessment process until all the elements are clearly defined. Moreover, DSR has provided a good structure for this research and to complete a thesis write-up. Table 3.2 provides how the DSR is applied to this research by executing the DSR activities.

Table 3.2: Application of DSR in this Research

DSR Guideline	**Activity Performed**	**Knowledge Base**
Problem Identification and motivation	The existing research focuses on risk assessment of CC from a broader perspective and provides generic risk assessment frameworks. These frameworks can not be applied to each of the cloud service delivery models as risk varies between these delivery models.	An SLR is conducted that provides vulnerabilities and threats to data and information that belongs to customers once they move to cloud virtualised infrastructure.

continued ...

Application of DSR in this Research ...continued

Guideline	Activity Performed	Knowledge Base
Objectives of the solution	Design of a vulnerability prediction and risk assessment process that is specific to IaaS.	Knowledge of unknown vulnerability prediction and risk assessment which can be used by the customers as a platform to make informed security decisions.
Solution Design	Design of the vulnerability prediction and risk assessment which can be used to predict unknown vulnerabilities that may appear in large software applications such as the Xen hypervisor. The determination of risks and their severity levels to analyse the security controls and procedures required to mitigate these risks to the customers' assets.	Knowledge of unknown vulnerability prediction, determination of vulnerability impact ratings, threat likelihood levels, and risk severity levels.
Evaluation of the process	Analysis of the process and how well the problem is addressed by comparing the research objectives with the obtained results.	Knowledge of two other open source packages to test the process. Research methods are applied and process is evaluated to determine the generalisability. The research methods did not show any significant limitation when applied to Apache HTTP and Squid Proxy servers.
Communication	The research articles are published in three different conferences (A Rank) and a Doctoral Consortium.	The last article is published in the 24^{th} Americas Conference on Information Systems (AMCIS) 2018 that provides a vulnerability and risk assessment process of Xen hypervisor.

Types of Artefacts

Artefacts are defined as end-goals of the DSR projects (Kotzé, van der Merwe & Gerber, 2015). Artefacts are actually the output of the IS research and can be broadly categorised as:

Instantiations A prototype system which is developed as an artefact in IS research.

Methods Methods refers to the type of artefact where research produces algorithms and practices to address the problem.

Models Artefact type when statements or propositions are considered explaining a set of constructs to address the research problem.

Constructs Concepts, syntax, and symbols are used in a specified context to present a problem and produce an artefact to address the problem.

Framework This type of artefact represents both a model and an interrelated method to the relevant model.

Contributions of DSR to this Research

The contributions which DSR provides to this research are as follows:

- DSR guides this research to identify a problem. It also provides a clear description of a problem area.
- It demonstrates that existing research does not provide a clear solution to the problem identified.
- It provides guidelines to design and develop a solution artefact by developing a vulnerability prediction and risk assessment process.
- Once the solution is developed, it enables us to perform a rigorous evaluation of the process and assess its utility.
- It also enables us to express the practical and theoretical values which the process adds to the body of knowledge.

3.3 Design Science Research Process

The DSR process by Offermann et al. (2009) in Figure 3.1 is used as the general basis for this research to construct a vulnerability prediction and risk assessment process. However, this research combines the basic research approach with the research guidelines provided by Hevner et al. (2004). The DSR process used in this research (Offermann et al., 2009) combines qualitative and quantitative research methods. The process consists of seven guidelines to conduct research. These guidelines are categorised into three phases: problem identification, solution artefact, and evaluation of the solution designed. The most critical phase of DSR is designing an artefact as a solution to address a research problem (Peffers, Tuunanen, Rothenberger & Chatterjee, 2007). All three phases can interact with each other within the research process. The steps are not always sequential; they can be referred back to each other. The execution of all three phases of the research produces DSR results.

3.3.1 Problem Identification Phase

The first phase of the DSR is the identification of a problem and definition of what is required to be achieved. The problem area should have practical relevance or might be relevant after the problem is addressed. The problem identification phase is comprised of three steps: identify the problem, formulate RQs, and develop hypotheses.

Two SLRQs are specified, and the practical relevance is determined during this problem identification phase. Two RQs are sought through SLR to address the problem area. Five hypotheses are also developed to answer these RQs. Thus, this phase provides an important foundation for the rest of the research process.

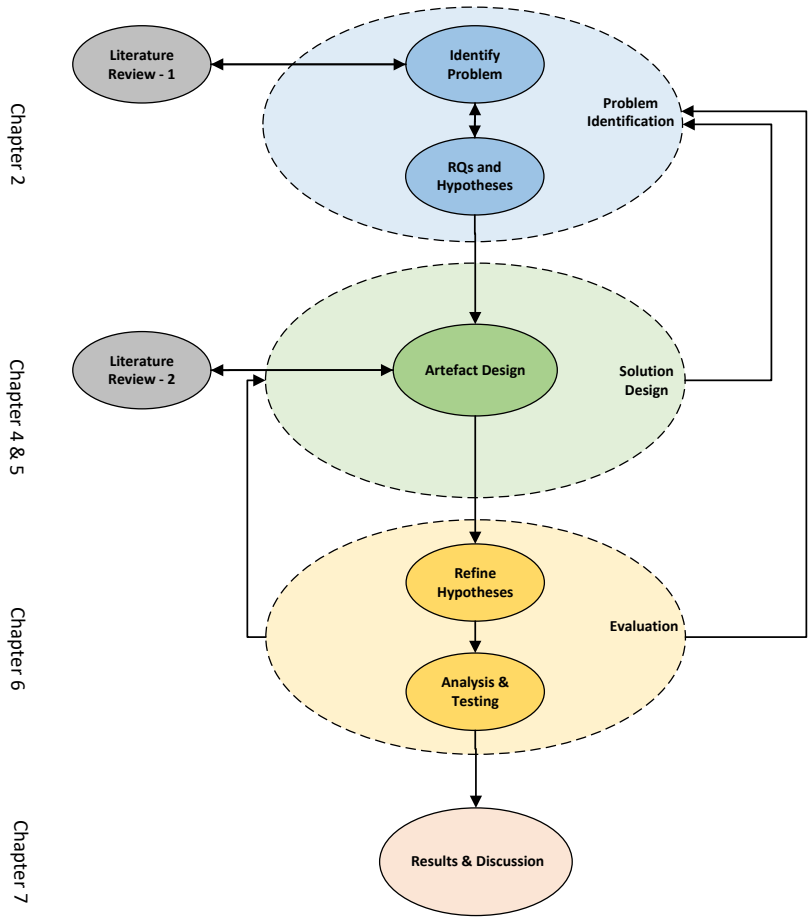


Figure 3.1: DSR Process adapted from Offermann et al. (2009)

Problem Area

A research problem is identified and refined during this phase to ensure the relevance and understanding to construct a solution artefact as the outcome of this research. The vulnerability prediction and risk assessment process (solution artefact) is not specific to one type of audience or organisation. The solution is generalisable as it can be used by different organisations to predict unknown vulnerabilities and understand the risk to the Xen hypervisor. The solution can also be applied to an open source software package to predict unknown vulnerabilities and determine the risk severity levels.

An SLR was conducted in Chapter 2 to identify the problem and possible research directions. Much research has been conducted which presents vulnerability prediction

models. However, these models lack accurate results and require improvements. Furthermore, these models do not use standard data sets for the prediction and they produce inaccurate results. Much research is also conducted that presents risk assessments of CC, but it does not focus on the risk assessment of IaaS and more specifically hypervisors. Also, the existing research assesses the risk from a CSP perspective, and cloud customers are not involved effectively in the risk assessment process. Peer-reviewed articles are reviewed to understand whether the possible solution artefact as the existing framework is effective to be fit for each of the cloud service delivery models. The risk varies between these service delivery models. Thus, it is difficult to consider a generic risk assessment process for all the service delivery models.

Research Questions

The RQs were formulated through the SLR conducted in Chapter 2. The research problem raised two RQs which are addressed through this research.

RQ1 How can the unknown vulnerabilities be assessed in large software applications such as the Xen hypervisor to mitigate exploitation scenarios?

RQ2 How can the determination of the risk of unknown Xen vulnerabilities be presented such that it aids cloud infrastructure service consumers?

Hypotheses

Five hypotheses were constructed to address the RQs and set the base for the solution. Five research methods are selected to test these hypotheses:

H1 Unknown vulnerabilities in the software applications can be predicted to mitigate the exploitation scenarios.

H2 Unknown vulnerabilities can also be predicted with regard to the impact levels to prioritise the mitigation of exploitation scenarios.

H3 By identifying and scoring the Xen vulnerabilities, the impact ratings can be determined to facilitate the risk assessment process.

H4 Threats to the Xen can be modelled to determine the likelihood of threats exploiting a Xen vulnerability.

H5 The results of vulnerability impact ratings and threat likelihood levels can be mapped to determine the severity levels of risks to the Xen.

Relationship between RQs, Hypotheses, and Methods

An outline of the relationship between RQs, hypotheses and research methods is provided in this section to develop an understanding. Figure 3.2 provides the relationship between RQs, hypotheses and research methods.

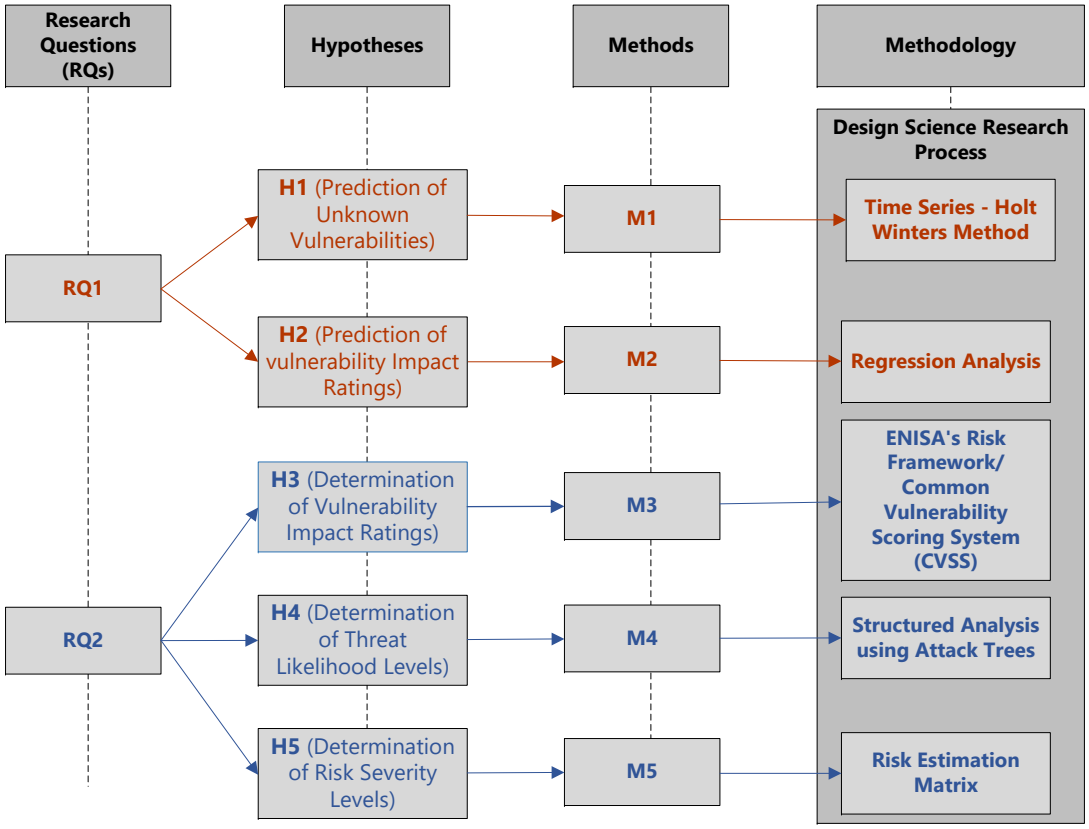


Figure 3.2: Relationship Between RQs, Hypotheses, and Methods

H1 and H2 are developed to address RQ1. H1 refers to the prediction of unknown vulnerabilities using the Time Series Holt-Winters Method (Kalekar, 2004; Tirkes, Guray & Celebi, 2017). H2 refers to the prediction of unknown vulnerabilities with regard to the impact levels using Regression Analysis (Palmer & O'connell, 2009).

RQ2 is addressed by testing H3, H4, and H5. Three Xen vulnerability exploitation

scenarios are developed and scored using CVSS to determine the impact ratings (Mell, Scarfone & Romanosky, 2006; Mell, Kent & Romanosky, 2007). However, it is not practicable to cover all the Xen vulnerability scenarios to determine the impact ratings. Therefore, ENISA's risk framework (Catteddu & Hogben, 2009) is used to test H3. The Xen vulnerability details and their impact ratings are adopted from ENISA to realise a complete risk assessment process. However, the CVSS is used only to determine impact ratings of Apache HTTP and Squid Proxy servers' vulnerabilities in Chapter 6. H4 is tested through Structured Analysis approach using Attack Trees (Saini, Duan & Paruchuri, 2008; K. Edge et al., 2007; Haque, Keffeler & Atkison, 2017). Threats are identified, and their likelihood levels are determined by developing attack trees. Initial threat levels are calculated using capabilities, and motivation characteristics of two different TAs. These threat levels are assigned to the source nodes of the attack tree. The threat levels are propagated through the attack tree to determine likelihood levels to exploit Physical, Local, Adjacent Network, and Network AVs. The overall threat likelihood levels to Xen hypervisor are also determined by propagating threat levels from child nodes to the root node of the attack tree. H5 is tested using a Risk Estimation Matrix (Catteddu & Hogben, 2009). Vulnerability impact ratings and threat likelihood levels are mapped through this risk estimation matrix to determine the severity levels of risk.

A detailed explanation and rationale of using these research methods are provided in Section 3.4.

3.3.2 Solution Design Phase

A solution artefact is constructed in this phase. After the problem is identified and its relevance is determined in phase one, a vulnerability prediction and risk assessment process is constructed as a solution artefact. Existing literature is reviewed extensively

to ensure the research rigour. The existing frameworks and state of the art are considered to come up with a solution to address the research problem. During the development and evaluation of the solution artefact, the problem is restated multiple times to maintain the relevance between problem and solution artefact. The Xen vulnerability prediction and risk assessment process is provided in Figure 3.3.

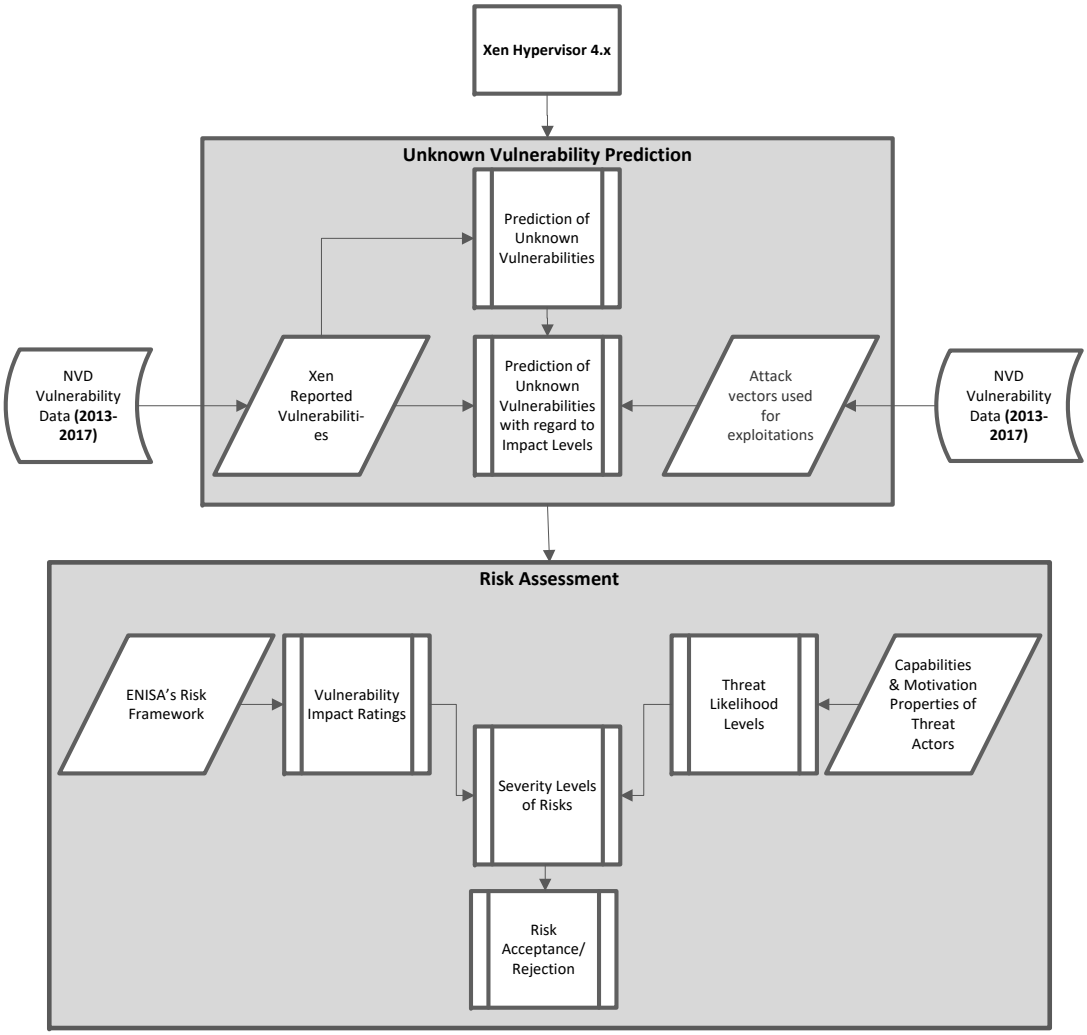


Figure 3.3: Xen Vulnerability Prediction and Risk Assessment Process

In this research, five analysis methods are used to test all hypotheses and realise the solution artefact. The details of these research methods are provided in Section 3.4.

3.3.3 Evaluation Phase

Hevner et al. (2004) presents five different evaluation methods deployed in DSR: observational, experimental, analytical, testing, and descriptive methods. This research uses an observational method (evaluating the process by applying it to two other open source software packages).

Once the solution artefact is developed and reaches an acceptable level, the evaluation process is started. The evaluation phase of the DSR process analyses and evaluates the solution. This phase is also iterated back to the solution design and to the problem identification phase (Offermann et al., 2009). Evaluation is achieved by applying the solution to another two open source packages: Apache HTTP and Squid Proxy servers. Similar to Xen, vulnerability prediction and risk assessment of both of these software packages is performed and covered in Chapter 6.

Refining Hypotheses

It is always difficult to evaluate a general hypothesis as a whole. Therefore, DSR recommends developing smaller hypotheses to have a more precise scope. In this research, the hypothesis is divided into smaller hypotheses which are evaluated using five different research methods. The results of this research are analysed, summarised, and discussed.

3.4 Analysis and Research Methods

In this section, the detail of data analysis and research methods is provided. The data analysis methods include literature review, vulnerability database search, statistical analysis, structured analysis, and estimation matrix.

3.4.1 Research Method 1: Time Series Holt-Winters Method

The Time Series Holt-Winters method is used to test H1, that is to predict unknown vulnerabilities. Roumani et al. (2015) uses time series analysis to predict the number of vulnerabilities for five common browsers. Different time series models such as a simple seasonal, Holt-Winters additive, and Autoregressive Integrated Moving Average (ARIMA) are considered to make the prediction. The Holt-Winters' additive model is used to predict Chrome website browser vulnerabilities. Symmetric Mean Absolute Percent Error (SMAPE) is used to measure the accuracy of the predicted results.

This research also uses the Holt-Winters method to predict unknown vulnerabilities (Kalekar, 2004; Tirkes et al., 2017). The Holt-Winters method was proposed in the early 1960s and extends exponential smoothing. It is prevalent, simple to use, and fits well in practical applications. It depends on the patterns of data variables in past movements, with the most weight given to the most recent values. It uses all previous data patterns to predict future movements. It is applicable to a dataset when it has trend and seasonal components. In other words, the Holt-Winters method is an expansion of an exponential smoothing method where exponential smoothing revises a prediction by assigning more weight to the recent data values and less weight to the older data values from the distant past. An intuitive set of weights is the set that decreases each time by a constant ratio, and all the weights lie on an exponential curve. However, exponential smoothing is not appropriate for the data that includes seasonal and trend components.

Justification for Using the Holt-Winters Method

The Holt-Winters method is a simple model which gives precise prediction results compared to other complex techniques such as Trend Analysis and Decomposition (Tirkes et al., 2017). Trend analysis is a simple prediction method, but it is not always applicable for lengthy time series. In case of lengthy time series, there have been several trends, and

the dataset should not be applicable to cyclic and seasonal data patterns. On the other hand, the decomposition prediction model can examine the dataset in trend, seasonal, cyclic, and random components. However, the decomposition model faces difficulties when it decomposes trend and seasonal components where a few seasonal cycles exist in the dataset. Therefore, this research uses the Holt-Winters method because the size of the reported vulnerability data is very small. On the other hand, both Trend Analysis, and the Decomposition model provide good prediction when the size of the dataset is large.

Application of the Method

In this research, the time series is developed by collecting Xen reported vulnerability data (2013 to 2017). The consecutive points of the data are linked. The evolution of the time series is observed, and conclusions are drawn from past patterns of Xen reported vulnerability data to predict the unknown Xen vulnerabilities. This provides information about future patterns and predicts the unknown vulnerabilities for 2018. To the best of our knowledge, only one other research focuses on predicting the impact levels of unknown vulnerabilities (S. Zhang, Caragea & Ou, 2011). Mostly, researchers have focused on discovering the unknown vulnerabilities.

The Holt-Winters multiplicative method is used in this research because the Seasonal variations are changing proportionally to the Level of the time series. The additive method can be considered when the time series data contains constant seasonal fluctuations, regardless of the overall level of the time series; therefore, the multiplicative method is used because the number of seasonal fluctuations varies, depending on the level of the time series. In the multiplicative method, the time series is represented by:

$$\textit{Systematic Component} = (Level + Trend) \times Seasonal\ Factor \tag{3.1}$$

The Holt-Winters method uses a modified form of exponential smoothing and applies three exponential smoothing formulae to the time series (Kalekar, 2004). The Level component is smoothed to give a local average value of the time series. Trend component, and Seasonal component values (from S_5 onwards) are smoothed to give a seasonal estimate of each period. The exponential smoothing formula is applied to a time series as it contains Trend and Seasonal components.

Smoothing is done using three smoothing equations along with the initial values which are used for the parameters. The current deseasonalised Level component at the end of the Time (*t*) is represented by L_t. T_t is the estimate of the Trend component and S_t is the estimate of the seasonal component.

Level (L_t) component is the random variation of the data in Time (*t*). L_t is calculated using Equation 3.2.

$$L_t = \alpha \times \frac{D_t}{S_t} + (1 - \alpha) \times (L_{t-1} + T_{t-1}) \tag{3.2}$$

where $0 < \alpha < 1$ is a smoothing constant. The number of reported vulnerability data value D_t is divided by S_t to deseasonalise the data to enter the Trend component and the prior value of the permanent component into the updating process for L_t.

The Trend (T_t) component exists in the data and is calculated using Equation 3.3.

$$T_t = \beta \times (L_t - L_{t-1}) + (1 - \beta) \times T_{t-1} \tag{3.3}$$

where $0 < \beta < 1$ is the second smoothing constant. The Trend component is calculated by estimating the smoothed difference between the two successive estimates of the deseasonalised level.

Equation 3.4 is used to smooth Seasonal component, where $0 < \gamma < 1$ is the third smoothing constant

$$S_{t+p} = \gamma \times \frac{D_t}{L_t} + (1 - \gamma) \times (S_t) \tag{3.4}$$

The seasonal component of the data is calculated as the behaviour of the time series data that repeats itself at Time (*t*) periods. The Seasonal component is calculated by combining the most recently observed Seasonal component given by the reported vulnerability data value D_t divided by the deseasonalised Level L_t and the previous best Seasonal component estimate for this Time (*t*). Since Seasonal components represent deviations above and below the average, the average of any consecutive seasonal components should always be 1.

After calculating the values for L_t, T_t, and S_t the prediction for the first Time (*t*) period can be calculated using Equation 3.5.

$$F_t = (L_{t-1} + T_{t-1}) \times S_t \tag{3.5}$$

Accuracy and Validity of the Prediction

It is important to measure the accuracy of the results to evaluate the validity of the time series prediction. There are different measures of prediction errors where the scale depends on the scale of the actual data. The commonly used scale dependent measures are based on the absolute error, or squared errors (Hyndman & Koehler, 2006; Adhikari & Agrawal, 2013). Mean Square Error (MSE) and Root Mean Square Error (RMSE) can be considered to determine the accuracy of the prediction, but they are more sensitive to outliers (Hyndman & Koehler, 2006). MSE is a measure of how close a prediction is to the actual data points. For all the data points, one gets the distance vertically from the data point to the relevant Y value on the data curve fit (error) and squares the value. Afterwards, all the values are added that correspond to all data points. Squaring ensures that negative data values do not cancel positive data values. On the

other hand, RMSE is the square root of the MSE. RMSE is, therefore, the distance on average, of a data point from the fitted line measured along a vertical line.

However, in this research Mean Absolute Deviation (MAD) and Mean Absolute Percentage Error (MAPE) are used to measure the accuracy and validity of the prediction respectively. MAD is the average of the absolute deviations (Equation 3.6). It is suitable to use when analysing the error of a single item. It is also used because the accuracy is expressed in the same units as the actual data. Moreover, MAD concludes whether or not the time series has generated an accurate prediction through Tracking Signal (TS) using a control chart.

$$MAD_n = \frac{1}{n}\sum_{t=1}^{n} A_t \tag{3.6}$$

The validity of the Holt-Winters prediction is determined by using MAPE and comparing the MAPE value of each calibration of the prediction model. MAPE is the average of absolute errors divided by actual observation values. It is also known as the Mean Absolute Percentage Deviation (MAPD). It allows one to measure the accuracy of the prediction model for constructing fitted time series data (Kim & Kim, 2016).

MAPE is one of the most common measures of prediction accuracy as it provides scale-in-dependency and interpret-ability (Kim & Kim, 2016). However, it gives problems when calculating the average MAPE. For example, the time series with a high MAPE might distort a comparison between the average MAPE with one method compared to the average MAPE when using another method. Another measure such as Symmetric MAPE (SMAPE) can be considered to avoid problems of infinite or undefined values for zero actual values. The MAPE is defined as follows:

$$MAPE = \frac{100}{N} \times \sum_{i=1}^{N} \left| \frac{ActualData - Prediction}{ActualData} \right| \tag{3.7}$$

Geng, Ye and Luo (2015) applied the Grey Model (GM)(1,1) to construct a forecasting model. However, the authors predict the vulnerability Impact Score for Xpdf and Lynx software applications. RMSE (Equation 3.8) is used to measure the accuracy of the forecasting results. RMSE is the standard deviation of the prediction results from the actual data where prediction errors are a measure of the data points far from the regression line. Equation 3.8 is used to measure the prediction accuracy vulnerability impact scores for Xpdf and Lynx.

$$RMSE = \sqrt{\frac{1}{N}\sum_{i=1}^{n}(x_i - \hat{x_i})} \tag{3.8}$$

Where x_i is the actual data value, $\hat{x_i}$ is the predicted value, and *N* is the number of the predicted data. So, Geng et al. (2015) uses RMSE and calculates RMSE = 2.51 for Xpdf and RMSE = 2.70 for Lynx. Therefore, Xpdf and Lynx prediction errors are 2.51 and 2.70 points away from the actual data, respectively. However, in this research, the regression analysis method is considered to predict unknown vulnerabilities with regard to impact levels. Regression analysis is a commonly used analysis method to determine relationships, and the strength of relationships between independent variables. The accuracy of the predicted results is calculated as RMSE = 1.81, that is less than the RMSE values determined by Geng et al. (2015). Therefore, regression analysis provides a more accurate prediction than the GM(1,1) model.

Usually, prediction models do not predict accurate results due to less smooth and inconsistent historical data. NVD data is also incomplete which results in less accurate predictions (S. Zhang et al., 2011). The reported Xen vulnerability dataset is also less smooth and does not have a constant trend. For example, five vulnerabilities are reported in the second quarter, and 22 vulnerabilities are reported in the third quarter of 2017. Another limitation of the vulnerability dataset is that it relies on the reporting time of vulnerabilities to NVD instead of the time of discovery (Last, 2015).

3.4.2 Research Method 2: Regression Analysis

Regression Analysis is used to predict unknown Xen vulnerabilities with regard to the impact levels. S. Zhang et al. (2011) uses regression models to predict Time to Next Vulnerability (TTNV) in Linux and Microsoft Windows OSs. The TTNV is referred to as the number which represents the days between the occurrences of vulnerabilities. The results were obtained using the epoch time scheme, and the month and day scheme, in terms of correlation coefficient, for regression models. Bibi, Tsoumakas, Stamelos and Vlahavas (2006) present a software defect prediction approach using regression via classification (RvC). RvC is applied as a machine learning approach to the problem of predicting the number of defects in a software system.

Regression analysis is a statistical method to determine the relationship between one dependent and one or more independent variables. The analysis results in a prediction for the dependent variable from a linear combination of the independent variables. Regression analysis can be used to predict, that includes classification and explanation (Palmer & O'connell, 2009). It selects an appropriate analysis model, realised by the method of test squares, with a view to exploiting the relationship between variables. This helps to determine the expected outcome for a given value of the independent variable.

Application of the Method

Regression analysis is used in this research to test Hypothesis (H2), that is to predict unknown vulnerabilities with regard to the impact levels. The number of vulnerabilities exploited through Local, Adjacent Network, and Network AVs are considered as independant vulnerabilities. The reason for this is that NVD (NIST, 2017) categorises the reported vulnerabilities as either Physical, Local, Adjacent Network, or Network vulnerabilities and then scores these vulnerabilities to determine High, Medium, and

Low impact vulnerabilities. Therefore, the Local, Adjacent Network, and Network vulnerabilities are considered as independent variables and their impact ratings (High, Medium, and Low) are considered as dependent variables for the prediction process. The physical vulnerabilities are not considered due to lower numbers of vulnerabilities reported to NVD for this category.

For the regression analysis, the first step is to calculate the correlation coefficients between the independent variables X_1, X_2, X_3, and dependent variables Y_n. Where, Y_n can be Y_1, Y_2, and Y_3. correlation coefficients are calculated initially to see the strength of the relationship between independent variables used to predict the future values. Correlation coefficient *R* determines the strength of the relationship between the variables. *R* value ranges from -1.0 to 1.0. If it is greater than 0, then it shows a positive linear relationship. The linear relationship is negative if it is less than 0. The *R* value 1.0 shows a strong relationship, and a value of 0 means there is no relationship between the independent variables.

- Correlation between Local AV and Adjacent Network AV (RX_1,X_2)
- Correlation between Local AV and Network AV (RX_1,X_3)
- Correlation between Local AV and impact ratings of Vulnerabilities (RX_1,Y_n)
- Correlation between Adjacent Network AV and Network AV (RX_2,X_3)
- Correlation between Adjacent Network AV and impact ratings of Vulnerabilities (RX_2,Y_n)
- Correlation between Network AV and impact ratings of Vulnerabilities (RX_3,Y_n)

Equation 3.9 is used to predict unknown Xen vulnerabilities with regard to the impact levels.

$$\bar{Y_n} = a + b_1(X_1) + b_2(X_2) + b_3(X_3) \tag{3.9}$$

where,

$\bar{Y_n}$ The predicted value of dependent variable Y_n.

a Is the 'Y' Intercept.

b_1 The change in the value of 'Y' for each one increment change in the value of X_1 that is, Local AV in this case.

b_2 The change in the value of 'Y' for each one increment change in the value of X_2 that is, Adjacent Network AV in this case.

b_3 The change in the value of 'Y' for each one increment change in the value of X_3 that is, Network AV in this case.

X A value of 'X' that is, the independent variable for which the value of Y_n can be predicted.

3.4.3 Research Method 3: Common Vulnerability Scoring System

The CVSS (Mell et al., 2006, 2007) provides a framework for researchers to perform statistical analysis on vulnerabilities and vulnerability properties (Mell et al., 2007). The CVSS uses a numeric scoring system to determine the impact rating of a vulnerability exploitation scenario (Gallon & Bascou, 2011). Elahi, Yu and Zannone (2010) presents a framework for security requirements elicitation and analysis centred on vulnerabilities. The authors argued that CVSS is very useful to evaluate the impact of vulnerabilities. Another work presented by Gallon and Bascou (2011) uses CVSS to create attack graphs to assess the impact of successful exploitation scenarios, taking into account correlation between successive atomic exploitation scenarios.

The CVSS labels vulnerability impact ratings as, Critical, High, Medium, and Low. It uses three different metric groups such as Base, Temporal, and Environmental. Each metric group further consists of a set of metrics. Figure 3.4 provides an overall CVSS metrics' view where the Base metric group provides the fundamental properties of vulnerability (Mell et al., 2007). The Base metric group further consists of two metrics: Exploitability and Impact Metrics. The Base metric group produces a CVSS Base score from 0 to 10 using a Base equation derived from the Exploitability, and the Impact

sub score equation. Later, the Base score can be modified by scoring the Temporal and Environmental metrics to reflect the risk to user's assets. However, scoring the Temporal and Environmental metrics is not considered by CVSS.

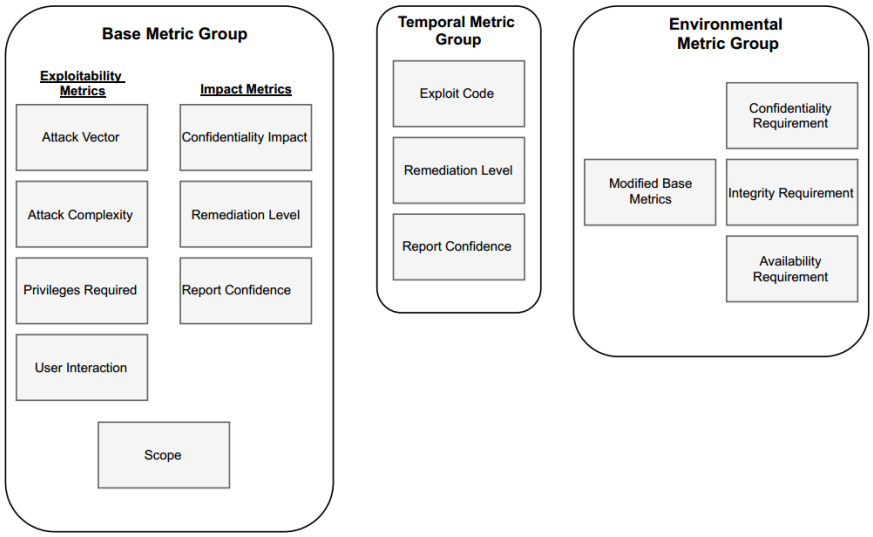


Figure 3.4: CVSS Metric Groups (Mell et al., 2007)

Justification of Using Common Vulnerability Scoring System

There are other vulnerability scoring systems available such as the SANS vulnerability analysis system. The SANS system considers whether the vulnerability exists in the default configurations or systems. Another scoring system offered by Microsoft tries to reflect the difficulty of exploitation and the impact of the vulnerability exploitation scenario. Both these vulnerability scoring systems are useful but provide a one-size-fits-all approach by considering that the impact vulnerability exploitation is the same for individuals or organisations. However, CVSS is designed to allow users to verify a vendor's calculations when desired (Mell et al., 2007). The CVSS uses Base score equations, scoring rubrics, and metrics to score and determine the impact ratings of vulnerabilities[1]. The CVSS calculator version 3.0 is used to score vulnerabilities using

[1]Appendix B and Section B.1 for details about CVSS Base score equations, scoring rubrics, and metrics.

a scale of 0 to 10 and determines qualitative impact ratings as Critical, High, Medium, and Low.

Critical impact level is assigned to a vulnerability if it has a CVSS base score of 9.0-10.0.

High impact level is assigned to a vulnerability if it has a CVSS base score of 7.0-8.9.

Medium impact level is assigned to a vulnerability if it has a CVSS base score of 4.0-6.9.

Low impact level is assigned to a vulnerability if it has a CVSS base score of 0.0-3.9.

Application of the Method

The CVSS is used to test Hypothesis (H3), that is to determine the vulnerability impact ratings. Three Xen vulnerability exploitation scenarios are scored to show CVSS application to determine impact ratings. However, it is not practicable to cover all the vulnerability scenarios to determine impact ratings of large software such as the Xen hypervisor. Therefore, vulnerability impact ratings are adopted for Xen risk assessment. However, in the case of Apache HTTP and Squid server, the CVSS is used completely to determine the vulnerability impact ratings in Chapter 6 (Process Evaluation).

ENISA's Risk Framework

It is not practicable to develop and score all the Xen vulnerability exploitation scenarios to determine impact ratings. Three example scenarios are presented to highlight the application of the CVSS to determine the impact ratings. Therefore, to realise a complete risk assessment process for the Xen, the impact ratings of the hypervisor related vulnerabilities are adopted from ENISA's framework (Catteddu & Hogben, 2009). However, CVSS is used to determine the impact ratings of Apache and Squid servers (Process Evaluation, Chapter 6).

ENISA's risk framework provides a list of 53 vulnerabilities that are combined with assets in a subset. Also, 35 risk scenarios and assets are mapped to the relevant vulnerabilities. ENISA determines vulnerability impact ratings by consulting a group of industry experts and professionals. Therefore, the hypervisor related vulnerabilities and their impact ratings are adopted from ENISA. ENISA labels and determines qualitative vulnerability impact ratings as Very High, High, Medium, Low, and Very Low.

3.4.4 Research Method 4: Structured Analysis Approach

A structured analysis approach using attack trees is used to test Hypothesis (H4), that is to identify threats and determine their likelihood levels. Saini et al. (2008) presents how attack trees could be used to analyse the security of MyProxy system, which is an important subsystem of the Globus toolkit. The authors argue that attack trees are an effective and convenient approach to evaluate possible security threats and to propose a mitigation strategy against those threats. K. S. Edge, Dalton, Raines and Mills (2006) developed attack and protection trees to identify the possible threats to Homeland Security. The authors categorise the protection tree and attack tree as two separate entities. The protection tree is built using the data as output from the attack tree. In this research, the Xen is also evaluated through a sequential and explorative process by creating attack trees to identify AVs which can be used by TAs to exploit Xen (Hutle, Hansch & Fitzgerald, 2015).

Justification of Using Attack Trees

Attack trees provide a simple and effective way to structure different threats that pose a risk to the target system (Schneier, 1999). Attack trees identify and analyse the consequences of a successful attack. Attack trees also make it possible to determine the most vulnerable path through Physical, Local, and Network AVs. Furthermore,

attack trees provide a top-down approach to how, why, and who (TAs) can exploit the target system. Attack trees use a tree structure to represent threats against the system, with the main goal as the root node and the different ways an attacker can choose to achieve that goal. Source nodes of the tree represent sub-subgoals of the attacker. Child nodes connected to source nodes are the ways to achieve the subgoals. It is a top-down approach where the child nodes are actually considered as a specification of the higher level nodes and can be either conjunctive (aggregation or *AND* nodes) or disjunctive (choice or *OR* nodes) (Haque et al., 2017). For *AND* nodes, all the immediate nodes will need to be in effect to achieve the goal. However, for *OR* nodes, any node will be sufficient to fulfil the goal. In other words, an *OR* logical path between the nodes represents alternatives whereas an *AND* logical path represents different steps toward achieving the same goal (Saini et al., 2008).

Attack trees are useful for threat modelling by analysing threats and identifying weak entry points which an attacker can use to exploit the system (Haque et al., 2017; Ingoldsby, 2010). The threat model is required to determine threat likelihood levels to the target system. To determine the threat likelihood and levels of Xen, HMG Information Assurance Standard No.1 (HMG IS1) (CESG, 2009) is used. First the Xen attack tree is developed, and initial threat values are assigned to the source nodes using the capability and motivation properties of the attacker (Hutle et al., 2015). Threat levels are then propagated to determine the threat levels to the root node which is the attacker's main goal. In this case, the child nodes refer to the exploitation of vulnerability through Physical, Local, and Network AVs, whereas the root node refers to manipulation of the Xen. Xen threat likelihood assessment is performed by considering two TAs; A Privileged User (PU) and a Normal User (NU).

Threat Actor Types

A brief description of PU and NU TAs is provided in this section. The description also includes Service Consumer (SC) Indirectly Connected (IC) as TAs. SC and IC TAs are used for threat likelihood determination for Apache and Squid server, respectively. The description of all the TAs is as follows:

Privileged User (PU) A PU is an authorised and registered user who is responsible to *manage* the software applications and services these servers are providing to users.

Normal User (NU) An NU is also an authorised and registered user who *uses* the servers, software applications, and the services offered by these servers. An NU is normally provided with standard access and system privileges as per the organisation's policies.

Service Consumer (SC) An SC is the one who only *uses* the services provided by the server. The server may also require that an SC should be a registered user to access the services. However, an SC is different from an NU. For example, an SC can only view the website, but cannot access the web server directly.

Indirectly Connected (IC) An IC is not an authorised user of the system. However, an IC may be able to access the server and its services because of onward connections from business partners, or through network connections to which the server has a direct connection such as the Internet. An IC TA refers to all internet users who can attack the system.

Capabilities and Motivation Properties of TAs

Capability and motivation properties of TAs are used to determine the initial threat levels which are assigned to the attack trees. (Table 3.3). The HMG IS1 standard outlines capability as a characteristic of TA and a component of threat. Capability defines a level

which indicates the type and technical sophistication of the threat. However, motivation is a measure of TA's desire to attack and compromise an asset or group of assets.

Table 3.3: Threat Actor Capabilities

Capability	**Description**
Formidable	TA has expert level knowledge about computers and security.
Significant	TA has professional level knowledge about computers and security.
Limited	TA is a trained computer or network user.
Little	TA is an average computer or network user.
Very Little	TA has very little knowledge about computers and security.

See Appendix A, Section A.1.1 for a detailed description of TA capability levels to exploit a vulnerability.

Table 3.4 provides motivation properties of a TA.

Table 3.4: Threat Actor Motivation Levels

Motivation	**Description**
Very High (Focused)	TA with such a motivation level will try to exploit the system by all means necessary.
High (Committed)	TA will try to exploit the target system on a frequent or constant basis.
Medium (Interested)	TA will try to exploit the target system if an opportunity exists.
Low (Curious)	TA will investigate the target system casually and attack if there is any weakness.
Very Low (Indifferent)	A TA with indifferent motivation level does not pose any risk.

See Appendix A, Section A.1.2 for a detailed description of TA motivation level to exploit a vulnerability.

Application of the Method

Structured asset-driven threat analysis allows a holistic analysis of threats (Hutle et al., 2015). Attack trees for different attack vectors are combined to generate a single attack tree for all the attacker's goals. An approach to creating individual attack trees for CIA is not considered due to limitations (Hutle et al., 2015). The Xen attack tree is a directed tree *(V, E)*, where every node (*n*) in (*V*) is labelled with either a logical *AND* or *OR*. The process of developing the Xen attack tree is as follows:

Definition of Attacker's Main Goal Manipulation of the Xen hypervisor and its assets is defined as the main goal of the attacker.

Breakdown of Overall Goal into Subgoals The attacker's main goal is broken down into subgoals: the exploitation of Physical, Local and Network AVs.

Decomposition of Sub-subgoals Attacker's subgoals are further broken down into sub-subgoals. All the source nodes of the Xen attack tree (nodes incoming edges) are the attacker's sub-subgoals.

Assigning Initial Threat Levels Initial threat levels are assigned to all source nodes using the capability and motivation properties of TAs.

Propagation of Threat Levels After assigning initial threat levels to all source nodes, threat levels are propagated to the root node to determine the overall threat likelihood level to the Xen. Threat levels for all the child nodes are adjusted to achieve the min/max condition. Attack trees use the commonly done min/max function. A Logical *AND* relationship means that all the source nodes are required to exploit to achieve the attacker's primary goal. The sub-goal with the lowest threat level would determine the difficulty level of the attack scenario. So, for every logical *AND* node, the *minimum* threat level of the source nodes is

propagated to the respective child node. A logical *OR* relationship means that one sub-goal is required to be exploited to achieve the attacker's goal. Therefore, the sub-goal with the highest threat level determines the difficulty level of the attack scenario. So, for every logical *OR* node, the *maximum* threat level of the source nodes is assigned the respective child node. *OR* nodes (child or source) have the value of their cheapest node. On the other hand, AND nodes (child or source) have the value of the sum of their nodes.

Threat Likelihood Levels Once all the nodes of the attack tree are computed, the threat likelihood to the overall Xen can be determined. Likelihood levels to Physical, Local and Network AVs are also determined.

Qualitative threat likelihood levels are determined and recorded as Severe, Substantial, Moderate, Low, and Negligible. Table 3.5 provides TA group characteristics and threat likelihood levels which these characteristics derive.

Table 3.5: Threat Likelihood Levels

Threat Level	**Threat Actor Group Characteristics**
Severe	This threat level can be observed when a TA behaves very severely and ignores all the security policies.
Substantial	This threat level can be observed when a TA does not behave well and sometimes ignores all the security policies and procedures.
Moderate	TA poses this threat level if one is a reliable and trustworthy person and behaves well.
Low	This threat level will be low if a TA behaves exceptionally well and is a trustworthy person.
Negligible	This threat level can be ignored by the organisations.

See Appendix A, Section A.2 for a description of qualitative threat levels.

3.4.5 Research Method 5: Risk Estimation Matrix

Risk assessment of CC is difficult due to its dynamic nature and different stakeholders. Also, a generic risk assessment process cannot be applied to all the service delivery models (Cayirci, 2015). Governmental and non-governmental organisations in Europe, such as ENISA (Catteddu & Hogben, 2009) and CNIL (Daskala & Le Metayer, 2012) provides risk assessment studies for the cloud. However, ENISA and CNIL provide generic frameworks for CC and do not differentiate between risk assessments of service delivery models. For example, ENISA provides 32 risks to CC and classifies these risks into three categories: organisational, technical, and legal risk. Organisational risk refers to the risk that affects the reputation of organisations and their businesses. Technical risk affects the CIA security objectives of cloud services and supporting systems. Legal risks are related to the compliance of security and privacy mandates by regulatory organisations (Alruwaili & Gulliver, 2014). However, ENISA does not provide risk scenarios for service delivery models. CNIL provides a risk management scheme to assess risk level management. CNIL generically determines threats against the privacy of CC from internal users; external users such as the service provider, competitors; and non-human sources such as malware, natural disasters, and so forth. However, it does not focus on a particular service delivery model as risks are different for each model.

Justification for Risk Assessment

The existing research (Hussain & Abdulsalam, 2011; Saripalli & Walters, 2010; Fitó & Guitart, 2014; Leitold & Hadarics, 2012; Tanimoto, Hiramoto, Iwashita, Sato & Kanai, 2011) has limitations in terms of its scope and applicability in the real cloud environment. The existing research views the risk to CC from a broader perspective, instead of

targeting specifically the hypervisor which provides the base for IaaS. These studies assess the risk from the CSPs point of view and do not include customers in the risk assessment process (Saripalli & Walters, 2010; Fitó & Guitart, 2014; X. Zhang, Wuwong, Li & Zhang, 2010). The existing research also lacks focus on the risk assessment of service delivery models. The generic risk assessment of the CC environment does not fit all the models and provides inaccurate results. Customers should be included in the risk assessment process or should have an independent platform to analyse risks and their severity levels. Therefore, this research presents a qualitative inductive risk assessment process that complements ENISA. The risk assessment process only considers technical risks and targets IaaS.

Application of the Method

A 5×5 risk estimation matrix (Catteddu & Hogben, 2009) is used to test Hypothesis (H5); that is to determine the severity levels of risk. Cayirci, Garaga, De Oliveira and Roudier (2016) presents a Cloud Adoption Risk Assessment Model (CARAM) which also complements ENISA to compute risk levels. The authors consider the qualitative risk estimation matrix used by ENISA as probability and impact values. CARAM maps these values to a quantitative scale to determine risk levels.

In this research, risk severity levels are determined by mapping the vulnerability impact ratings and threat likelihood levels using ENISA's risk estimation matrix. Nine risk types are listed, and their qualitative severity levels are determined as High, Medium, and Low if the risk scale is from 7-9, 4-6, and 1-3, respectively.

3.5 Methods and Hypotheses Testing

This section provides the rationale between RQs, hypotheses, research methods, and how each research method tests the relevant hypothesis.

3.5.1 Testing of Hypothesis 1

Table 3.6 provides that how the Time Series Holt-Winters method is used to test H1.

Table 3.6: Testing of Hypothesis 1

Method 1	**Time Series Holt-Winters Method**
Research Question	How can the unknown vulnerabilities be assessed in large software applications such as the Xen hypervisor to mitigate exploitation scenarios?
Hypothesis	Unknown vulnerabilities in the software applications can be predicted to mitigate the exploitation scenarios.
Analysis	Unknown Vulnerability Prediction. **Constant:** Xen vulnerabilities. **Independent Variable:** The *number of reported* Xen vulnerabilities. **Dependent variable:** The *number of unknown* Xen vulnerabilities. **Experiment:** Unknown Xen vulnerabilities are predicted for 2018 using reported vulnerabilities of the last five years. The vulnerability prediction results show that the hypothesis is correct as 41.85 unknown vulnerabilities are predicted using the Holt-Winters Model (very close to 43.80 reported vulnerabilities in each of the last five years).
Output Data	**Quantitative:** The number of unknown vulnerabilities are predicted for 2018.
Analysis of Results	MAD and MAPE are used to measure the accuracy and validity of the prediction results, respectively. The control charts are also used to track the prediction results against a UCL and LCL.
Evaluation	The Holt-Winters model is also applied to Squid and Apache software packages to determine its applicability and generalisability by predicting unknown vulnerabilities of these software packages. The model resulted in good prediction results. Different prediction modes such as the Holt-Winters Additive, ARIMA, and decomposition can also be used for the prediction. However, the Holt-Winters Multiplicative model produced good results (for three different software packages) using the reported vulnerability datasets retrieved from the NVD .

3.5.2 Testing of Hypothesis 2

Table 3.7 provides that how regression analysis is used to test H2 and address RQ1.

Table 3.7: Testing of Hypothesis 2

Method 2	**Regression Analysis**
Research Question	How can the unknown vulnerabilities be assessed in large software applications such as the Xen hypervisor to mitigate exploitation scenarios?
Hypothesis	Unknown vulnerabilities can also be predicted with regard to the impact levels to prioritise the mitigation of exploitation scenarios.
Analysis	Prediction of unknown vulnerabilities with regard to the impact levels. **Constant:** Xen vulnerabilities. **Independent Variable:** The number of reported Xen vulnerabilities exploited through Local (X_1), Network (X_2), and Adjacent Network (X_3) AVs. **Dependent Variable:** Prediction of High (Y_1), Medium (Y_2), and Low (Y_3) impact unknown vulnerabilities. **Experiment:** High, Medium, and Low impact unknown Xen vulnerabilities are predicted for 2018. The prediction results show that 10.43 unknown vulnerabilities will be of High impact. Similarly, 27.47 Medium and 9.91 Low impact unknown vulnerabilities are predicted.
Output Data	**Quantitative:** The number of High, Medium, and Low Impact unknown vulnerabilities that may appear in the Xen.
Analysis of Results	The predicted results are compared with the average number High, Medium, and Low impact vulnerabilities reported during the last five years. 10.43 High impact unknown vulnerabilities are predicted. This result is an accurate prediction as it is very close to the average 10 High impact vulnerabilities reported each year in the last five years. Both Medium and Low impact unknown vulnerabilities are predicted accurately. The prediction results are close to the average 24.40 Medium and 9.40 Low impact reported vulnerabilities.

continued ...

Testing of Hypothesis 2	...continued
Evaluation	Regression analysis is also used to predict High, Medium, and Low impact unknown vulnerabilities of Squid and Apache software packages. Regression analysis produced accurate results (close to average High, Medium, and Low reported vulnerabilities). This makes it a reasonable model to be used for the prediction when more than one independent variables are used for the predictions (three in this case).

3.5.3 Testing of Hypothesis 3

CVSS is used to test H3 and address RQ2. Table 3.8 provides the summary of testing H3.

Table 3.8: Testing of Hypothesis 3

Method 3	**Common Vulnerability Scoring System**
Research Question	How can the determination of the risk of unknown Xen vulnerabilities be presented such that it aids cloud infrastructure service consumers?
Hypothesis	By identifying and scoring the Xen vulnerabilities, the impact ratings can be determined to facilitate the risk assessment process.
Analysis	Determination of Impact Ratings of Xen vulnerabilities. **Constant:** Impact ratings of Xen vulnerabilities. **Independent Variable:** The Base Score Metrics (Exploitability Metrics and Impact Metrics). **Dependent Variable:** The impact ratings of Xen vulnerabilities. **Experiment:** The vulnerability exploitation scenarios are scored using CVSS to determine the impact ratings of Xen vulnerabilities to facilitate the risk assessment process. CVSS uses a scale of 0 to 10 to determine qualitative impact ratings as Critical, High, Medium, and Low.
Output Data	**Qualitative:** The impact ratings of Xen vulnerability exploitation scenarios.

continued ...

Testing of Hypothesis 3 ...continued	
Analysis of Results	Three Xen vulnerability exploitation scenarios (Physical, Local, and Network) are developed to score vulnerabilities from two different threat actors. A Medium impact rating is determined when a PU exploits a physical vulnerability. High impact ratings are determined for two scenarios when an NU exploits a local and a network vulnerability.
Evaluation	CVSS is also used to score Apache and Squid vulnerabilities to determine impact ratings. The CVSS is a generic framework and was very suitable to perform the qualitative analysis of different vulnerability scenarios to determine the impact ratings using Exploitability and Impact Base metrics. The SANS and Microsoft also offer scoring systems. However, these scoring systems provide a one-size-fits-all approach for individuals or organisations.

3.5.4 Testing of Hypothesis 4

Table 3.9 provides that how a structured analysis approach is used to test H4 and address RQ2.

Table 3.9: Testing of Hypothesis 4

Method 4	**Structured Analysis Approach**
Research Question	How can the determination of the risk of unknown Xen vulnerabilities be presented such that it aids cloud infrastructure service consumers?
Hypothesis	Threats to the Xen can be modelled to determine the likelihood of these threats exploiting a Xen vulnerability.

continued ...

Testing of Hypothesis 4	...continued
Analysis	Determination of threat likelihood levels. **Constant:** Threats to the Xen. **Independent Variable:** Capability and motivation properties of threat actors to exploit a Xen vulnerability. **Dependent Variable:** Threat likelihood levels. **Experiment:** Xen attack trees for a PU and an NU threat actor are developed to determine threat likelihood levels to Xen. The capabilities and motivation properties of these threat actors are used to determine initial threat levels. Afterwards, threat levels are propagated through the attack tree to determine the overall threat likelihood level to Xen from these threat actors.
Output Data	**Qualitative:** Severe, Substantial, Moderate, Low, and Negligible threat likelihood levels to Xen from two different threat actors.
Analysis of Results	A separate attack tree is developed for each of the threat actors. A Moderate threat likelihood level is determined from a PU. On the other hand, a Severe threat likelihood level is determined from an NU. The results show that an NU is most likely going to realise a network attack by exploiting a vulnerability that exists in the network stack of the Xen host (Dom0).
Evaluation	Attack trees are also developed to determine the threat likelihood levels to Apache and Squid software packages. Attack trees provided an easy way to model threats to Apache and Squid to determine threat likelihood levels. A different threat modelling technique such as STRIDE could have been used for threat modelling. However, it cannot be used in isolation. The understanding about other techniques such as Microsoft security life cycle development process, use cases and architectures was required. Therefore, attack trees are used and provided a simple structure to determine threat likelihood levels by presenting the AVs which can be used to exploit vulnerabilities.

3.5.5 Testing of Hypothesis 5

Table 3.10 provides that how risk estimation matrix is used to test H5 and address RQ2.

Table 3.10: Testing of Hypothesis 5

Method 5	**Risk Estimation Matrix**
Research Question	How can the determination of the risk of unknown Xen vulnerabilities be presented such that it aids cloud infrastructure service consumers?
Hypothesis	The results of vulnerability impact ratings and threat likelihood levels can be mapped to determine the severity levels of risks to the Xen.
Analysis	Determination of severity levels of risks to the Xen. **Constant:** Risk to the Xen. **Independent Variable:** Vulnerability impact ratings and threat likelihood levels. **Dependent Variable:** Risk severity levels. **Experiment:** The Xen vulnerability impact ratings and threat likelihood levels are mapped using risk estimation matrix to determine severity levels of nine technical risks to Xen. Nine risk types are listed, and their qualitative severity levels are determined as High, Medium, and Low if the risk scale is from 7-9, 4-6, and 1-3 respectively.
Output Data	**Qualitative:** High, Medium, and Low risk severity levels to Xen from a PU and an NU threat actor.
Analysis of Results	High risk severity levels are determined for two risk types: Malicious Insider and Compromise Hypervisor from a PU threat actor. An NU poses High risk severity levels for four risks: Isolation Failure, Intercepting Data in Transit, Undertaking Malicious Probes or Scans, and Compromise of Hypervisor.
Evaluation	Risk severity levels to Apache HTTP and Squid servers are also determined using risk estimation matrix. The results of Apache and Squid risk assessment supported the hypothesis. The risk estimation matrix provided the risk severity levels which can be used by organisations to make informed security decisions. The risk matrix was effective because vulnerability impact ratings and threat likelihood levels were estimated accurately to determine the severity levels of risks.

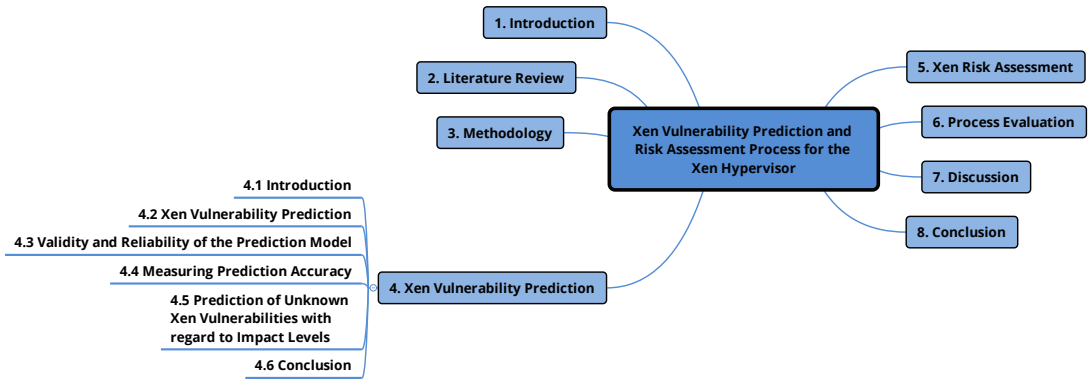
3.6 Conclusion

In this chapter, an overview of DSR methodology and a rationale of the adopted DSR process was covered. Five hypotheses were constructed to address the RQs. Five

research methods, time series Holt-Winters, regression analysis, CVSS, structured analysis approach, and risk estimation matrix, were discussed to test these hypotheses. All research methods were applied within the context of the DSR process. Chapter 4 next provides an analysis of the Xen vulnerability prediction process and how the Holt-Winters and regression analysis methods are used to predict unknown vulnerabilities for the future.

Chapter 4

Xen Vulnerability Prediction



4.1 Introduction

In Chapter 3, DSR methodology was covered including research questions, hypotheses, and research methods. This chapter provides vulnerability prediction of the Xen hypervisor. Through the vulnerability prediction process, unknown Xen vulnerabilities are predicted for 2018. The Time series Holt-Winters method is used for the prediction. The reliability of the prediction model is determined using the Mean Absolute Percentage Error (MAPE). MAD is also used to measure the accuracy of the prediction model.

To extend the scope of the prediction process, unknown vulnerabilities are also

predicted with regard to the impact levels. However, regression analysis is used for predicting the impact ratings as more than one independent data variable is used as input for the prediction. High, Medium, and Low impact unknown vulnerabilities are predicted for 2018.

This chapter is organised as follows: Section 4.2 covers the Xen vulnerability prediction process. Section 4.3 provides details of the validity and reliability of the prediction model. Section 4.4 provides details of measuring the accuracy of the prediction model. Prediction of unknown vulnerabilities with regard to the impact levels is covered in Section 4.5. Section 4.6 provides the conclusion of this chapter.

4.2 Xen Vulnerability Prediction

In this section, the unknown Xen vulnerabilities are predicted using reported vulnerability data from NVD (NIST, 2017). NVD is a reliable vulnerability database and it is publicly available. It helps researchers and organisations research the automation of vulnerability management. It also provides different security and compliance goals. Users can search it for reported vulnerabilities, checklists, impact metrics and different statistics. This research targets to predict the number of unknown Xen vulnerabilities using the reported vulnerability data set.

4.2.1 Data Source

The Xen reported vulnerability data were gathered from NVD from 2013 to 2017. According to the NVD search, 219 vulnerabilities were reported in Xen from 2013 to 2017. The largest number of Xen vulnerabilities were DoS related, indicating a potential weakness in this area. This problem may reflect the type of hypervisor (bare metal), whereas KVM (hosted) would benefit from the underlying OS which provides a

degree of protection through resource and memory management. Table 4.1 provides a summary of Xen reported vulnerabilities per year from 2013 to 2017.

Table 4.1: Xen Vulnerabilities

Year	Vulnerabilities
2013	43
2014	45
2015	41
2016	28
2017	62

Table 4.1 shows that from 2013, an alarming number of vulnerabilities were reported in the Xen. This may be a result of increased awareness of the types of vulnerabilities. 62 vulnerabilities were reported in 2017 which is the highest number compared to other years. It is assumed that the number of vulnerabilities may be more than 62. This assumption leads to a hypothesis that there is a relationship between the size and complexity of a software package, and the number of potential vulnerabilities reported. It may be that there still exist vulnerabilities being open source and given the large size of the Xen hypervisor code. Therefore, the Xen is thought to be suitable for the prediction of unknown vulnerabilities.

4.2.2 Prediction of Unknown Xen Vulnerabilities

The Holt-Winters method (Kalekar, 2004; Tirkes et al., 2017) is used to predict the number of unknown vulnerabilities. It predicts the systematic component of the actual data (reported vulnerabilities) to predict the random component. The systematic component usually contains *Level (L)*, *Trend (T)* and *Seasonal Factors (S)*. A multiplicative Holt-Winters method (Equation 3.1) is used to calculate the systematic component of the actual data. The prediction in time (t) for actual data in time ($t + l$) is calculated using Equation 4.1.

$$F_{t+1} = \left(L_{t-1} + T_{t-1}\right) \times S_{t+1} \tag{4.1}$$

Xen version 4.x is targeted for vulnerability prediction. The first version of 4.x series was released in April 2010. However, it is observed that data before 2013 is not complete, or this series was not efficiently used between 2010 and 2012. Therefore, the number of Xen reported vulnerabilities from 2013 to 2017 is used to make the prediction. Table 4.2 provides Xen reported vulnerabilities which are distributed per quarter for each year.

Table 4.2: Xen Reported Vulnerabilities

Time (t)	**Year**	**Quarter**	**Reported Vulnerabilities**
1	2013	1	7
2		2	7
3		3	13
4		4	16
5	2014	1	9
6		2	20
7		3	4
8		4	12
9	2015	1	9
10		2	12
11		3	5
12		4	15
13	2016	1	5
14		2	16
15		3	6
16		4	1
17	2017	1	17
18		2	5
19		3	22
20		4	18

Deseasonalising Actual Data

The first step is to deseasonalise the actual data (D_t) to start the prediction process. Deseasonalising D_t results in deseasonalised data which would be available without seasonal fluctuations. *P* denotes the number of periods, and the D_t cycle repeats after every four periods (quarters). Therefore, the value of $p = 4$.

To deseasonalise the data, equal weight is given to each season as required. The average of p consecutive periods of the D_t is taken. The average of D_t from (l + 1) to (l + p) provides the deseasonalised data for $[l + (\frac{p+1}{2})]$. As p is even in this case, deseasonalised data at a point between $[l + (\frac{p}{2})]$ and $[l + 1 + (\frac{p}{2})]$ is calculated. Deseasonalised data for $[l + 1 + (\frac{p}{2})]$ is calculated by averaging the D_t values (l + 1) to (l + p) and (l + 2) to (l + p + 1). Equation 4.2 is used to calculate $\bar{D}_t$, for the time *t*, where *p* is even.

$$\bar{D}_t = \frac{D_{t-(\frac{p}{2})} + D_t + (\frac{p}{2}) + \sum_{i=t+1-(\frac{p}{2})}^{i=t-1+(\frac{p}{2})} 2D_i}{2p} \tag{4.2}$$

By using the Equation 4.2, deseasonalised data values from t_3 to t_{18} are calculated. Table 4.3 provides the deseasonalised data for these time periods.

Table 4.3: Deseasonalised Data

Time (t)	Year	Quarter	Reported Vulnerabilities (D_t)	Deseasonalised Data
1	2013	1	7	-
2		2	7	-
3		3	13	11.00
4		4	16	12.88
5	2014	1	9	13.38
6		2	20	11.75
7		3	4	11.25
8		4	12	10.25
9	2015	1	9	9.38

continued ...

Deseasonalised Data ...continued

Time (t)	Year	Quarter	Reported Vulnerabilities (D_t)	Deseasonalised Data
10		2	12	9.88
11		3	5	9.75
12		4	15	9.75
13	2016	1	5	10.38
14		2	16	8.75
15		3	6	8.50
16		4	1	8.63
17	2017	1	17	9.25
18		2	5	13.38
19		3	22	-
20		4	18	-

The next step is to perform a linear regression using the relationship between deseasonalised data and Time (t) based on the change in D_t over time (Equation 4.3).

$$\bar{D}_t = L + T \times t \tag{4.3}$$

In Equation 4.3, $\bar{D}_t$ is the deseasonalised data in time (t). L is the Level value and T is the rate of growth of data at t_0. The L and T values at t_0 are calculated using linear regression. Deseasonalised data values are considered as dependent variables, and time (t) values as independent variables. Using Table 4.3, initial Level L_0 is calculated as 12.18 (*intercept coefficient*), and initial Trend T_0 is calculated as -0.16 (*X variable or slope*). Therefore, for any time (t), the deseasonalised data $\bar{D}_t$ is calculated using Equation 4.4.

$$\bar{D}_t = 12.18 + (-0.16) \times t \tag{4.4}$$

Table 4.4 provides $\bar{D}_t$ values (after linear regression) for all the time periods from t_1 to t_{20}.

Table 4.4: Deseasonalised Data After Regression

Time (t)	Year	Quarter	Reported Vulnerabilities	Deseasonalised Data	$\bar{D}_t$
1	2013	1	7	-	12.03
2		2	7	-	11.87
3		3	13	11.00	11.71
4		4	16	12.88	11.55
5	2014	1	9	13.38	11.39
6		2	20	11.75	11.23
7		3	4	11.25	11.07
8		4	12	10.25	10.91
9	2015	1	9	9.38	10.75
10		2	12	9.88	10.59
11		3	5	9.75	10.43
12		4	15	9.75	10.27
13	2016	1	5	10.38	10.11
14		2	16	8.75	9.95
15		3	6	8.50	9.79
16		4	1	8.63	9.63
17	2017	1	17	9.25	9.47
18		2	5	13.38	9.31
19		3	22	-	9.15
20		4	18	-	8.99

Determination of Initial Seasonal Factors

The next step in the prediction process is to calculate $\bar{S}_t$ values for initialisation. The $\bar{S}_t$ values for time (*t*) is the ratio of D_t to $\bar{D}_t$. Equation 4.5 is used to calculate $\bar{S}_t$.

$$\bar{S}_t = \frac{D_t}{\bar{D}_t} \tag{4.5}$$

Table 4.5 provides $\bar{S}_t$ for the time period S_1 to S_{20}.

Table 4.5: Initial Seasonal Factor Values

Time (t)	**Year**	**Quarter**	D_t	**Deseasonalised Data**	$\bar{D}_t$	$\bar{S}_t$
1	2013	1	7	-	12.03	0.58
2		2	7	-	11.87	0.59
3		3	13	11.00	11.71	1.11
4		4	16	12.88	11.55	1.39
5	2014	1	9	13.38	11.39	0.79
6		2	20	11.75	11.23	1.78
7		3	4	11.25	11.07	0.36
8		4	12	10.25	10.91	1.10
9	2015	1	9	9.38	10.75	0.84
10		2	12	9.88	10.59	1.13
11		3	5	9.75	10.43	0.48
12		4	15	9.75	10.27	1.46
13	2016	1	5	10.38	10.11	0.49
14		2	16	8.75	9.95	1.61
15		3	6	8.50	9.79	0.61
16		4	1	8.63	9.63	0.10
17	2017	1	17	9.25	9.47	1.80
18		2	5	13.38	9.31	0.54
19		3	22	-	9.15	2.41
20		4	18	-	8.99	2.00

After calculating $\bar{S}_t$ values for initialisation, S_t for a given time period is obtained by averaging $\bar{S}_t$ values that correspond to similar time periods. For example, as the value of p = 4, $\bar{S}_t$ are similar at the time periods t_1, t_5, t_9, t_{13}, and t_{17}. Therefore, seasonal factors of these time periods are calculated as the average of five seasonal factors. The S_t is obtained using Equation 4.6 for all the time periods of Table 4.5, $(pt + i), (1 \leq i \leq p)$, for given data cycles r. Recall that L_0 and T_0 are already calculated as 12.18 and -0.16, respectively.

$$\bar{S}_i = \frac{\sum_{j=0}^{r=l} S_j p+i}{r} \tag{4.6}$$

As $p = 4$, $t = 20$, and seasonal cycle $r = 5$, S_t values are calculated using Equation 4.6.

S_1 = 0.90

S_2 = 1.13

S_3 = 0.99

S_4 = 1.21

Table 4.6 provides initial S_1 to S_4 values for the first four time periods.

Table 4.6: Initial S_t Values

Time (t)	**Year**	**Quarter**	D_t	**Deseasonalised Data**	D_t	S_t	Initial S_t Values
1	2013	1	7	-	12.03	0.58	0.90
2		2	7	-	11.87	0.59	1.13
3		3	13	11.00	11.71	1.11	0.99
4		4	16	12.88	11.55	1.39	1.21
5	2014	1	9	13.38	11.39	0.79	-
6		2	20	11.75	11.23	1.78	-
7		3	4	11.25	11.07	0.36	-
8		4	12	10.25	10.91	1.10	-
9	2015	1	9	9.38	10.75	0.84	-
10		2	12	9.88	10.59	1.13	-
11		3	5	9.75	10.43	0.48	-
12		4	15	9.75	10.27	1.46	-
13	2016	1	5	10.38	10.11	0.49	-
14		2	16	8.75	9.95	1.61	-
15		3	6	8.50	9.79	0.61	-
16		4	1	8.63	9.63	0.10	-
17	2017	1	17	9.25	9.47	1.80	-
18		2	5	13.38	9.31	0.54	-
19		3	22	-	9.15	2.41	-
20		4	18	-	8.99	2.00	-

Prediction Using Smoothing Parameter 0.10

Level L_t, *Trend* T_t, and S_5 to S_{20} are calculated to make the prediction for 2018. Therefore, to perform these calculations, three smoothing parameters α, β, and γ are considered. The predicted values are compared with the values of the data set. Table 4.7 provides Holt-Winters smoothing parameters.

Table 4.7: Smoothing Parameters

Smoothing Parameters	**Value**
Level (α), Trend (β), Seasonal (γ)	0.10

α is used to calculate L_t values (Equation 4.7), and T_t values (Equation 4.8) are calculated using β. Smoothing parameter γ is used to calculate remaining S_t values (Equation 4.9) from (S_5 to S_{20}). Equation 4.7, Equation 4.8, and Equation 4.9 are used to calculate these values.

$$L_t = \alpha \times \frac{D_t}{S_t} + (1 - \alpha) \times (L_{t-1} + T_{t-1}) \tag{4.7}$$

$$T_t = \beta \times (L_t - L_{t-1}) + (1 - \beta) \times T_{t-1} \tag{4.8}$$

$$S_{t+p} = \gamma \times \frac{D_t}{L_t} + (1 - \gamma) \times (S_t) \tag{4.9}$$

Earlier, initial *Level* (L_0) and *Trend* (T_0) are calculated as 12.18 and -0.16 respectively. S_1, S_2, S_3, and S_4 are also calculated. Thus, unknown vulnerabilities for the first quarter of 2013 are predicted using Equation 4.10.

$$F_{t+1} = (L_{t-1} + T_{t-1}) \times S_{t+1} \tag{4.10}$$

$$\begin{aligned} F_1 &= (L_0 + T_0) \times S_1 \\ &= (12.18 + (-0.16)) \times 0.90 \\ &= 10.83 \end{aligned} \tag{4.11}$$

After the first prediction is made, it is assumed that time moves forward as 10.83 vulnerabilities were predicted for the first quarter of 2013. As a high value is predicted

for the first quarter, L_t, T_t, and S_t can be updated using Equations 4.12, 4.13, and 4.14. Therefore, *Level* (L_1) is updated with an assumption that α = 0.10.

$$\begin{aligned} L_1 &= \alpha \times \frac{D_1}{S_1} + (1-\alpha) \times (L_0 + T_0) \\ &= 0.10 \times \frac{7}{0.90} + (1 - 0.10) \times (12.18 + (-0.16)) \\ &= 11.60 \end{aligned} \tag{4.12}$$

Also, the *Trend* (T_1) is updated with an assumption that β = 0.10.

$$\begin{aligned} T_1 &= \beta \times (L_1 - L_0) + (1-\beta) \times T_0 \\ &= 0.10 \times (11.60 - 12.18) + (1 - 0.10) \times (-0.16) \\ &= -0.20 \end{aligned} \tag{4.13}$$

After updating Level and Trend components using α and β, S_5 is also updated with an assumption that γ = 0.10.

$$\begin{aligned} S_5 &= \gamma \times \frac{D_1}{L_1} + (1-\gamma) \times (S_1) \\ &= 0.10 \times \frac{7}{11.60} + (1 - 0.10) \times (0.90) \\ &= 0.87 \end{aligned} \tag{4.14}$$

After calculating L_t, T_t, and S_t values for all 20 time periods, the predictions for the four quarters of 2018 are made.

$$\begin{aligned} F_{21} &= [L_{20} + (T_{20} \times 1)] \times S_{17} \\ &= [10.13 + (0.03 \times 1)] \times 0.84 \\ &= 8.54 \end{aligned} \tag{4.15}$$

$$
\begin{aligned}
F_{22} &= [L_{20} + (T_{20} \times 2)] \times S_{18} \\
&= [10.13 + (0.03 \times 2)] \times 1.22 \\
&= 12.41
\end{aligned}
\tag{4.16}
$$

$$
\begin{aligned}
F_{23} &= [L_{20} + (T_{20} \times 3)] \times S_{19} \\
&= [10.13 + (0.03 \times 3)] \times 0.89 \\
&= 9.07
\end{aligned}
\tag{4.17}
$$

$$
\begin{aligned}
F_{24} &= [L_{20} + (T_{20} \times 4)] \times S_{20} \\
&= [10.13 + (0.03 \times 4)] \times 1.16 \\
&= 11.83
\end{aligned}
\tag{4.18}
$$

Table 4.8 provides all the L_t, T_t, S_t and predicted values. The model predicts 41.85 Xen vulnerabilities for 2018 when all three smoothing parameters are assigned a 0.10 value.

Table 4.8: Prediction of Unknown Xen Vulnerabilities using 20 Periods

Time (t)	D_t	**Seasonal Factors for Initialisation**	S_t Values	L_t Values	T_t Values	**Prediction**
0	-	-	-	12.18	-0.16	-
1	7	0.58	0.90	11.60	-0.20	10.83
2	7	0.59	1.13	10.88	-0.25	12.88
3	13	1.11	0.99	10.87	-0.23	10.56
4	16	1.39	1.21	10.90	-0.20	12.88
5	9	0.79	0.87	10.66	-0.21	9.31
6	20	1.78	1.08	11.26	-0.13	11.30
7	4	0.36	1.01	10.41	-0.20	11.28
8	12	1.10	1.24	10.16	-0.20	12.62
9	9	0.84	0.87	10.00	-0.20	8.64
10	12	1.13	1.15	9.86	-0.19	12.28
11	5	0.48	0.95	9.23	-0.24	9.19
12	15	1.46	1.23	9.31	-0.21	11.06
13	5	0.49	0.87	8.77	-0.24	7.93

continued ...

Prediction of Unknown Xen Vulnerabilities using 20 Periods ...continued

Time (t)	D_t	**Seasonal Factors for Initialisation**	S_t Values	L_t Values	T_t Values	**Prediction**
14	16	1.61	1.16	9.06	-0.19	9.87
15	6	0.61	0.91	8.64	-0.21	8.07
16	1	0.10	1.27	7.67	-0.29	10.70
17	17	1.80	0.84	8.66	-0.16	6.21
18	5	0.54	1.22	8.07	-0.20	10.37
19	22	2.41	0.89	9.55	-0.03	6.99
20	18	2.00	1.66	10.13	0.03	11.00
21	Prediction	-	-	-	-	**8.54**
22	Prediction	-	-	-	-	**12.41**
23	Prediction	-	-	-	-	**9.07**
24	Prediction	-	-	-	-	**11.83**

Figure 4.1 provides the prediction of Xen unknown vulnerabilities using smoothing parameters = 0.10 and 20 periods.

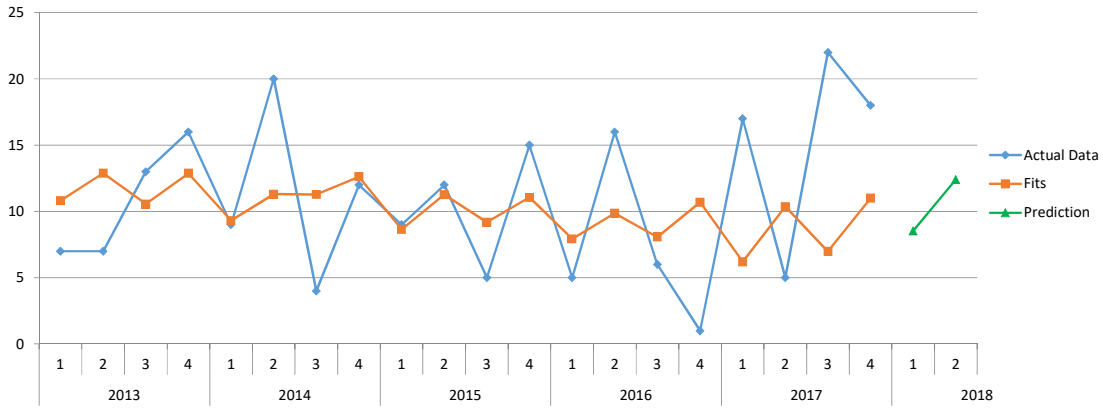


Figure 4.1: Prediction Graph Using 0.10 Smoothing Parameters

Prediction with Smoothing Parameters 0.20 and 0.30

Now the predictions are made using smoothing parameters values, 0.20 and 0.30. Figure 4.2 provides the prediction of Xen unknown vulnerabilities using smoothing parameters = 0.20 and 20 periods.

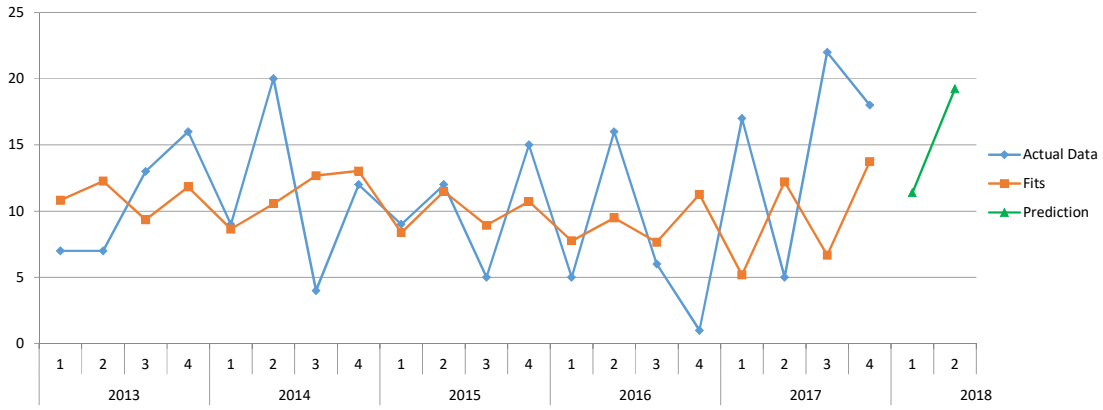


Figure 4.2: Prediction Graph Using 0.20 Smoothing Parameters

Figure 4.3 provides the prediction of Xen unknown vulnerabilities using smoothing parameters = 0.30 and 20 periods.

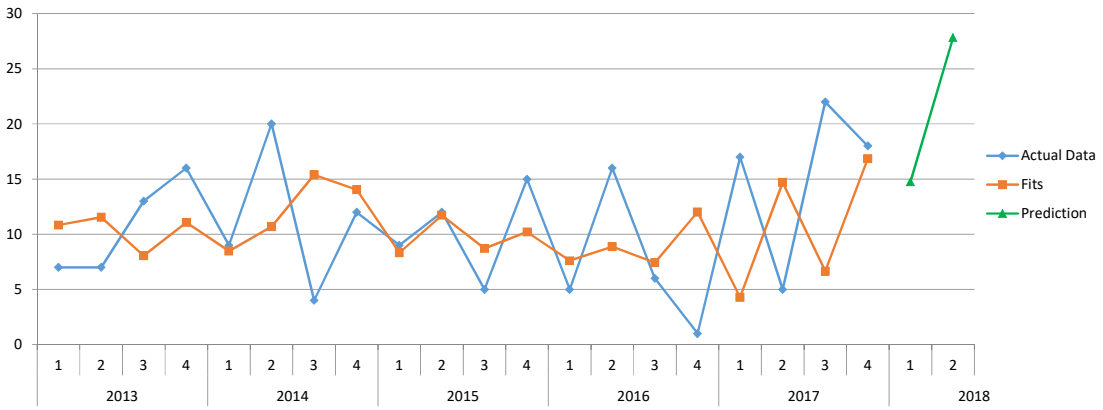


Figure 4.3: Prediction Graph Using 0.30 Smoothing Parameters

It is observed that the prediction model predicted 41.85 unknown vulnerabilities when 0.10 value is assigned to all three smoothing parameters. The result is accurate as it is very close to the average reported vulnerabilities (43.80 vulnerabilities per year). The prediction model predicted 61.63 and 84.64 vulnerabilities when 0.20 and 0.30 values are assigned to smoothing parameters respectively.

4.3 Validity and Reliability of the Prediction Model

In this section, the validity and reliability of the prediction model is performed. The validity of the Holt-Winters prediction is determined by measuring the percentage that the prediction model is a good fit for predicting unknown Xen vulnerabilities. This is done by using MAPE (Equation 4.19).

$$MAPE = \frac{100}{N} \times \sum_{i=1}^{N} |\frac{ActualData - Prediction}{ActualData}| \tag{4.19}$$

Whereas, actual data is the data-set used for prediction. Prediction is the estimated or predicted time series observations. *N* is the number of non-missing data points.

Table 4.9 provide Holt-Winters smoothing parameters and MAPE value.

Table 4.9: Smoothing Parameters and MAPE

Smoothing Parameters	**MAPE Value**
Level (α) 0.10, Trend (β) 0.10, Seasonal (γ) 0.10	95.53
Level (α) 0.20, Trend (β) 0.20, Seasonal (γ) 0.20	101.73
Level (α) 0.30, Trend (β) 0.30, Seasonal (γ) 0.30	111.29

4.3.1 Prediction for periods 23, 24, 25 and 26

The lower MAPE value is observed when the 0.10 smoothing parameter is assigned to Level, Trend, and Seasonal components when evaluated with 20 quarters of actual data. The reported vulnerability data values for quarter 21 and 22 are then added to the sample data set. The 0.10 smoothing parameter calibration is applied to 22 periods. The final prediction is made for periods 23, 24, 25 and 26.

Table 4.10: Prediction of Unknown Xen Vulnerabilities using 22 Periods

Time (t)	D_t	**Seasonal Factors for Initialisation**	S_t Values	L_t Values	T_t Values	**Prediction**
0	-	-	-	11.13	-0.02	-
1	7	0.63	0.87	10.80	-0.06	9.64
2	7	0.63	1.11	10.30	-0.10	11.88
3	13	1.18	0.93	10.57	-0.07	9.48
4	16	1.45	1.15	10.85	-0.03	12.08
5	9	0.82	0.85	10.80	-0.03	9.16
6	20	1.83	1.06	11.57	0.05	11.44
7	4	0.37	0.96	10.87	-0.03	11.15
8	12	1.10	1.18	10.77	-0.03	12.82
9	9	0.83	0.85	10.73	-0.04	9.08
10	12	1.11	1.13	10.69	-0.04	12.08
11	5	0.46	0.90	10.14	-0.09	9.59
12	15	1.39	1.18	10.32	-0.06	11.81
13	5	0.47	0.84	9.83	-0.10	8.67
14	16	1.49	1.13	10.17	-0.06	10.98
15	6	0.56	0.86	9.80	-0.09	8.69
16	1	0.09	1.20	8.82	-0.18	11.68
17	17	1.60	0.81	9.87	-0.06	7.01
18	5	0.47	1.17	9.26	-0.11	11.52
19	22	2.08	0.84	10.87	0.06	7.64
20	18	1.71	1.09	11.48	0.12	11.96
21	4	0.38	0.90	10.88	0.04	10.46
22	5	0.48	1.11	10.28	-0.02	12.13
23	Prediction	-	-	-	-	**8.57**
24	Prediction	-	-	-	-	**11.20**
25	Prediction	-	-	-	-	**9.22**
26	Prediction	-	-	-	-	**11.32**

Figure 4.4 provides Holt-Winters prediction with smoothing parameters = 0.10 and 22 periods.

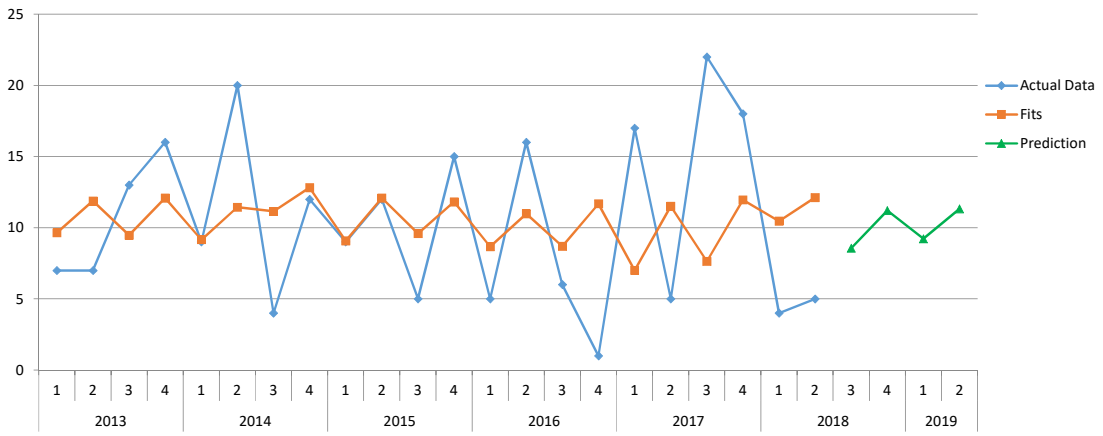


Figure 4.4: Prediction with Smoothing Parameters 0.10 and 22 Periods

The in-sample MAPE is determined as 105.13 with smoothing parameter 0.10. This MAPE value is larger than the MAPE value (95.53) which was observed earlier with original 20 periods and smoothing parameter 0.10. However, the difference between both MAPE values is not significant which indicates the reliability of the prediction model. Therefore, the Holt-Winters model can be used as a simple prediction model to predict software vulnerabilities to address the research problem or conduct short-term planning. However, the utility of the Holt-Winters prediction model is extensive, and it can be used for prediction in different other domains such as healthcare, sales, marketing, and production. It is also observed that the Holt-Winters method provides a good fit when the sample data set has Level, Trend, and Seasonal components.

4.4 Measuring Prediction Accuracy

In Section 4.3, the reliability of the prediction results was measured using MAPE. However, to have more clarity about the accuracy of the prediction model, MAD is also used to measure the accuracy of the prediction model. A good prediction model should include a Mean, an Estimate, and how the predicted values vary from the Mean value. MAD is used to calculate prediction errors because the accuracy is measured in

the same units as the actual data. The first step is to calculate the prediction error to measure the accuracy of the prediction. The prediction error is the difference between the actual data and the predicted values for time (*t*). Prediction errors for all quarters (2013-2017) are measured using Equation 4.20.

$$\begin{aligned} E_t &= Actual - Prediction \\ &= D_t - F_t \end{aligned} \tag{4.20}$$

Prediction errors are then summed to determine Bias (*B*) to check whether the prediction is biased or not. Equation 4.21 is used to determine *B*.

$$B_n = \sum_{t=1}^{n} E_t \tag{4.21}$$

The next step is to calculate Absolute Deviation (A_t) which is the absolute value of the prediction error ($|E_t|$). A_t is calculated using Equation 4.22.

$$A_t = |E_t| \tag{4.22}$$

Once A_t is calculated, MAD is determined by averaging the A_t values of all (*t*) periods.

4.4.1 Tracking Signal (TS)

Tracking Signal (TS) evaluates the model and determines whether or not the prediction needs revision for accurate results. TS is calculated by taking the ratio of Bias and MAD. A common rule of thumb is that TS should be between -3 and 3 where the Mean value is 0. If TS is out of range, the prediction model should be investigated. TS is calculated using Equation 4.23.

$$TS_t = \frac{bias_t}{MAD_t} \tag{4.23}$$

Table 4.11 provides a summary of prediction error and TS for all quarters. Accumulative MAD for all quarters is 5.02 when smoothing parameters value used is 0.10.

Table 4.11: Prediction Error Tracking Signal

Time (t)	**Prediction**	**Error** E_t	**Bias**	$A_t = \|E_t\|$	**MAD**	**TS**
1	10.83	-3.83	-3.83	3.83	3.83	-1.00
2	12.88	-5.88	-9.71	9.71	4.86	-2.00
3	10.56	2.44	-7.27	12.15	4.05	-1.80
4	12.88	3.12	-4.15	15.27	3.82	-1.09
5	9.31	-0.31	-4.46	15.58	3.12	-1.43
6	11.3	8.70	4.24	24.28	4.05	1.05
7	11.28	-7.28	-3.04	31.56	4.51	-0.67
8	12.62	-0.62	-3.66	32.18	4.02	-0.91
9	8.64	0.36	-3.30	32.54	3.62	-0.91
10	11.28	0.72	-2.58	33.26	3.33	-0.78
11	9.19	-4.19	-6.77	37.45	3.40	-1.99
12	11.06	3.94	-2.83	41.39	3.45	-0.82
13	7.93	-2.93	-5.76	44.32	3.41	-1.69
14	9.87	6.13	0.37	50.45	3.60	0.10
15	8.07	-2.07	-1.70	52.52	3.50	-0.49
16	10.7	-9.70	-11.40	62.22	3.89	-2.93
17	6.21	10.79	-0.61	73.01	4.29	-0.14
18	10.37	-5.37	-5.98	78.38	4.35	-1.37
19	6.99	15.01	9.03	93.39	4.92	1.84
20	11	7.00	16.03	100.39	5.02	3.19

4.4.2 Control Chart

A control chart is used to track the prediction errors for all quarters. A control chart consists of three elements: a control chart starts with a period graph, a Mean of prediction error samples as a central line to show shifts or trends, and Upper and Lower

Control Limits (UCL and LCL) placed equidistant from the Mean. UCL and LCL are calculated by estimating the Standard Deviation (σ) of prediction errors sample. σ is multiplied by 3 and added (3σ to Mean) to calculate UCL and subtracted (3σ from the Mean) to calculate LCL. Equation 4.24 is used to determine CLs.

$$CL = \overline{X} \pm 3\sigma \tag{4.24}$$

Usually, Mean ± 3σ should account for 99.7% of the distribution of observations. Only 0.3% observations (3.75 MAD or 3σ) should be out of the range. A control chart measures TS and compares Bias with UCL and LCL to see whether the predicted values are on the positive or the negative side of the threshold. If the prediction model is under-predicting, then the TS will be on the positive side of the Mean. If it is over-predicting, then TS will be on the negative side of the Mean. However, if the prediction model is out of control, or net-cumulative error is beyond 3σ from the Mean then TS is over 3.75 MAD. This shows that the prediction process requires evaluation. The prediction errors are distributed with a Mean of 0.80. A 3σ deviation from the Mean is equal to a control chart with the UCL and LCL set at 20.01 and -18.41, respectively from Mean 0.85. Table 4.12 provides the Mean values, UCL, and LCL for all the quarters.

Table 4.12: Control Limits to Measure Prediction Accuracy

Time (t)	**Prediction**	**Error** E_t	**MAD**	**TS**	**Mean**	**UCL**	**LCL**
1	10.83	-3.83	3.83	-1.00	0.80	20.01	-18.41
2	12.88	-5.88	4.86	-2.00	0.80	20.01	-18.41
3	10.56	2.44	4.05	-1.80	0.80	20.01	-18.41
4	12.88	3.12	3.82	-1.09	0.80	20.01	-18.41
5	9.31	-0.31	3.12	-1.43	0.80	20.01	-18.41
6	11.3	8.70	4.05	1.05	0.80	20.01	-18.41
7	11.28	-7.28	4.51	-0.67	0.80	20.01	-18.41
8	12.62	-0.62	4.02	-0.91	0.80	20.01	-18.41
9	8.64	0.36	3.62	-0.91	0.80	20.01	-18.41

continued ...

Control Limits to Measure Prediction Accuracy ...continued

Time (t)	Prediction	Error E_t	MAD	TS	Mean	UCL	LCL
10	11.28	0.72	3.33	-0.78	0.80	20.01	-18.41
11	9.19	-4.19	3.40	-1.99	0.80	20.01	-18.41
12	11.06	3.94	3.45	-0.82	0.80	20.01	-18.41
13	7.93	-2.93	3.41	-1.69	0.80	20.01	-18.41
14	9.87	6.13	3.60	0.10	0.80	20.01	-18.41
15	8.07	-2.07	3.50	-0.49	0.80	20.01	-18.41
16	10.7	-9.70	3.89	-2.93	0.80	20.01	-18.41
17	6.21	10.79	4.29	-0.14	0.80	20.01	-18.41
18	10.37	-5.37	4.35	-1.37	0.80	20.01	-18.41
19	6.99	15.01	4.92	1.84	0.80	20.01	-18.41
20	11	7.00	5.02	3.19	0.80	20.01	-18.41

Figure 4.5 provides TS of Xen vulnerability prediction. It shows that the model is over-predicting for most of the quarters. However, the prediction model does not exceed threshold levels (UCL or LCL). Thus, the prediction is acceptable.

Figure 4.5: Tracking Prediction Accuracy

4.5 Prediction of Unknown Xen Vulnerabilities with regard to the Impact Levels

In Section 4.2.2, the prediction of Xen unknown vulnerabilities was performed. This section covers the prediction of unknown vulnerabilities with regard to the impact levels. The purpose of predicting vulnerabilities with regard to the impact levels is to expand the scope of the vulnerability prediction process. Regression analysis is used for the prediction. The number of reported vulnerabilities exploited using Local Access, Adjacent Network, and Remote Network AVs are considered as the independent variables (X_1, X_2, and X_3). High, Medium and Low impact ratings of reported vulnerabilities are used as dependent variables (Y_1, Y_2, and Y_3). The predicted values are calculated as $\bar{Y_1}$, $\bar{Y_2}$, and $\bar{Y_3}$ for High, Medium, and Low impact levels of unknown vulnerabilities, respectively.

Table 4.13 provides the data variables used for prediction.

Table 4.13: Independent and Dependent Variables

		X_1	X_2	X_3	Y_1	Y_2	Y_3
Year	**Quarter**	**Local**	**Adjacent Network**	**Network**	**High**	**Medium**	**Low**
2013	1	4	3	0	0	6	1
	2	7	0	0	0	4	3
	3	6	7	0	3	9	1
	4	9	7	0	1	11	4
2014	1	3	6	0	2	6	1
	2	6	14	0	1	14	5
	3	2	2	0	0	3	1
	4	6	4	2	2	3	4
2015	1	8	0	1	2	8	2
	2	7	2	3	4	6	2
	3	4	0	1	2	2	1
	4	14	0	1	3	8	4
2016	1	1	4	0	0	3	2

continued ...

Independent and Dependent Variables ...continued

		X_1	X_2	X_3	Y_1	Y_2	Y_3
	2	15	0	1	3	6	7
	3	6	0	0	3	2	1
	4	1	0	0	0	0	1
2017	1	17	0	0	1	11	5
	2	5	0	0	2	2	1
	3	10	0	12	10	11	1
	4	17	0	1	11	7	0

Table 4.13 provides the Mean values in independent and dependent variables used for the prediction.

Table 4.14: Mean Values of Data Variables

	Local	**Adjacent Network**	**Network**	**High**	**Medium**	**Low**
Mean value per year	29.60	9.80	4.40	10.00	24.40	9.40

4.5.1 Prediction of High Impact Unknown Vulnerabilities

Table 4.15 provides X_1, X_2, and X_3 as independent variables and Y_1 (High Impact) as dependent variable. $\bar{Y_1}$ is the predicted value of Y_1.

Table 4.15: Data Variables to Predict High Impact Vulnerabilities

		X_1	X_2	X_3	Y_1
Year	**Quarter**	**Local**	**Adjacent Network**	**Network**	**High Impact Vulnerabilities**
2013	1	4	3	0	0
	2	7	0	0	0
	3	6	7	0	3
	4	9	7	0	1
2014	1	3	6	0	2
	2	6	14	0	1
	3	2	2	0	0
	4	6	4	2	2

continued ...

Data Variables to Predict High Impact Vulnerabilities ...continued

		X_1	X_2	X_3	Y_1
2015	1	8	0	1	2
	2	7	2	3	4
	3	4	0	1	2
	4	14	0	1	3
2016	1	1	4	0	0
	2	15	0	1	3
	3	6	0	0	3
	4	1	0	0	0
2017	1	17	0	0	1
	2	5	0	0	2
	3	10	0	12	10
	4	17	0	1	11

The first step of the prediction process is to calculate correlation coefficients. Table 4.16 provides the correlation coefficients.

Table 4.16: Correlation Coefficients to Predict High Impact Vulnerabilities

	Local	**Adjacent Network**	**Network**	**High Impact**
Local AV	1			
Adj Network AV	-0.28	1		
Network AV	0.19	-0.21	1	
High Impact	0.54	-0.24	0.66	1

R value is calculated which is the combined correlation between Local, Adjacent Network, and Network AVs with High Impact vulnerabilities. The computed value of *R* is 0.78 which is relatively close to 1.00. This shows that the independent and dependent variables have a strong relationship. Multiple regression analysis (Equation 4.25) is used to make the prediction.

$$\bar{Y_1} = a + b_1(X_1) + b_2(X_2) + b_3(X_3) \tag{4.25}$$

where,

- $\bar{Y}$ = The predicted value of dependent variable Y.
- a = Is the Y-Intercept.
- b_1 = The change in the value of Y for each one increment change in the value of X_1 that is, Local AV.
- b_2 = The change in the value of Y for each one increment change in the value of X_2 that is, Adjacent Network AV.
- b_3 = The change in the value of Y for each one increment change in the value of X_3 that is, Network AV.
- X = A value of X that is, the independent variable for which the value of Y is predicted.

Through regression analysis, the below regression coefficients are calculated.

- Y Intercept (a) = -0.12
- Local AV (b_1) = 0.26
- Adjacent Network AV (b_2) = 0.00
- Network AV (b_3) = 0.65

The P-value of AV Local is 0.02, Adjacent Network is 0.97, and Network is 0.00. It is observed that the P-value of Adjacent Network is higher than 0.15. Therefore, it is not considered for the prediction.

Equation 4.26 is used to predict High Impact unknown vulnerabilities for 2018. For example, 29.60 vulnerabilities would appear that exploit Local AV and 4.40 Network AV (see Mean values provided in Table 4.14). Prediction result shows that out of those 34 vulnerabilities, at least 10.43 vulnerabilities will be of High impact rating. This is an excellent prediction as it is almost the same as the per year mean value of High impact reported vulnerabilities of 10.00.

$$\begin{aligned} \bar{Y_1} &= a + b_1(X_1) + b_3(X_3) \\ &= -0.12 + 0.26(29.60) + 0.65(4.40) \\ &= 10.43 \end{aligned} \tag{4.26}$$

4.5.2 Prediction of Medium Impact Unknown Vulnerabilities

Table 4.17 provides the X_1, X_2, and X_3 as independent variables and Y_2 (Medium Impact) as dependent variable to predict unknown vulnerabilities of Medium Impact. $\bar{Y_2}$ is the predicted value of Y_2. Table 4.18 provides the correlation coefficients.

Table 4.17: Data Variables to Predict Medium Impact Vulnerabilities

		X_1	X_2	X_3	Y_2
Year	**Quarter**	**Local**	**Adjacent Network**	**Network**	**Medium Impact Vulnerabilities**
2012	1	0	0	0	0
	2	1	0	0	0
	3	0	0	0	0
	4	33	1	0	22
2013	1	4	3	0	6
	2	7	0	0	4
	3	6	7	0	9
	4	9	7	0	11
2014	1	3	6	0	6
	2	6	14	0	14
	3	2	2	0	3
	4	6	4	2	8
2015	1	8	0	1	3
	2	7	2	3	6
	3	4	0	1	2
	4	14	0	1	8
2016	1	1	4	0	3
	2	15	0	1	6
	3	6	0	0	2
	4	1	0	0	0
2017	1	17	0	0	11
	2	5	0	0	2

continued ...

Data Variables to Predict Medium Impact Vulnerabilities ...continued

	X_1	X_2	X_3	Y_2
3	10	0	12	11
4	17	0	1	7

Table 4.18: Correlation Coefficients to Predict Medium Impact Vulnerabilities

	Local	**Adjacent Network**	**Network**	**Medium Impact**
Local AV	1			
Adj Network AV	-0.15	1		
Network AV	0.08	-0.14	1	
Medium Impact	0.80	0.39	0.18	1

R value is calculated as 0.97 using the correlation coefficients and is almost equal to 1.00, which shows the independent and dependent variables have a solid relationship. So, to proceed with the prediction, regression coefficients are calculated.

- Y Intercept (a) = -0.75
- Local AV (b_1) = 0.61
- Adjacent Network AV (b_2) = 0.85
- Network AV (b_3) = 0.42

P-values of all the AVs is 0.00. Therefore, all the coefficients are considered for the prediction. So, to make the prediction using Equation 4.27, 29.60 vulnerabilities would appear that exploit Local AV, 9.80 Adjacent Network AV, and 4.40 Network AV (see Mean values provided in Table 4.14). So, the prediction result shows that out of those 41.85 vulnerabilities, at least 27.47 vulnerabilities will be of Medium Impact rating during 2018. This result is a fair prediction as it is close to the per year mean value of Medium Impact reported vulnerabilities of 24.40.

$$\begin{aligned}\bar{Y_2} &= a + b_1(X_1) + b_2(X_2) + b_3(X_3) \\ &= -0.75 + 0.61(29.60) + 0.85(9.80) + 0.42(4.40) \\ &= 27.47\end{aligned} \tag{4.27}$$

4.5.3 Prediction of Low Impact Unknown Vulnerabilities

Table 4.19 provides the X_1, X_2, and X_3 as independent variables and Y_3 (Low Impact) as dependent variable. $\bar{Y_3}$ is the predicted value of Y_3. Table 4.20 provides the correlation coefficients.

Table 4.19: Data Variables to Predict Low Impact Vulnerabilities

		X_1	X_2	X_3	Y_3
Year	**Quarter**	**Local**	**Adjacent Network**	**Network**	**Low Impact Vulnerabilities**
2012	1	0	0	0	0
	2	1	0	0	0
	3	0	0	0	0
	4	33	1	0	10
2013	1	4	3	0	1
	2	7	0	0	3
	3	6	7	0	1
	4	9	7	0	4
2014	1	3	6	0	1
	2	6	14	0	5
	3	2	2	0	1
	4	6	4	2	2
2015	1	8	0	1	4
	2	7	2	3	2
	3	4	0	1	1
	4	14	0	1	4
2016	1	1	4	0	2
	2	15	0	1	7
	3	6	0	0	1
	4	1	0	0	1
2017	1	17	0	0	5
	2	5	0	0	1

continued ...

Data Variables to Predict Low Impact Vulnerabilities ...continued

	X_1	X_2	X_3	Y_3
3	10	0	12	1
4	17	0	1	0

Table 4.20: Correlation Coefficients to Predict Low Impact Vulnerabilities

	Local	**Adjacent Network**	**Network**	**Low Impact**
Local AV	1			
Adj Network AV	-0.13	1		
Network AV	0.10	-0.16	1	
Medium Impact	0.78	0.13	-0.10	1

R value is calculated as 0.83 using the correlation coefficients and is close to 1.00, that shows the independent and dependent variables have a solid relationship. So, to proceed with the prediction, regression coefficients are calculated.

- Y Intercept (a) = 0.07
- Local AV (b_1) = 0.28
- Adjacent Network AV (b_2) = 0.16
- Network AV (b_3) = -0.15

P-value of the Local AV is 0.00, Adjacent Network AV is 0.09, and Network AV is 0.24. The P-value of Network AV is more than 0.15; it is not considered for the prediction. So, to make the prediction using Equation 4.28, 29.60 vulnerabilities would appear that exploit Local AV and 9.80 Adjacent Network AV (see Mean values provided in Table 4.14). So, the prediction result shows that out of those 39.40 vulnerabilities, at least 9.91 vulnerabilities will be of Low Impact rating during 2018. This result is a sound prediction as it is close to the per year mean value of Low Impact reported vulnerabilities of 9.40.

$$\begin{aligned} \bar{Y_3} &= a + b_1(X_1) + b_2(X_2) \\ &= 0.07 + 0.28(29.60) + 0.16(9.80) \\ &= 9.91 \end{aligned} \tag{4.28}$$

It is imperative to predict unknown vulnerabilities with regard to the impact levels. High, Medium, and Low impact unknown vulnerabilities are predicted for 2018. Prediction results show that out of 41 unknown vulnerabilities predicted, at least 10.43 unknown vulnerabilities will be of High impact. This result is an accurate prediction as it is very close to the average 10 High impact vulnerabilities reported each year in the last five years. Similarly, 27.47 Medium and 9.91 Low impact unknown vulnerabilities are predicted. Both these predictions are also accurate as results are close to the average 24.40 Medium, and 9.40 Low impact vulnerabilities reported each year in the last five years. Table 4.21 provides a summary of prediction results.

Table 4.21: Summary of Xen Vulnerability Prediction

Vulnerability Impact Level	**Average Impact Levels from 2013-2017**	**Prediction for 2018**	**Prediction Error**
High	10.00	10.43	-0.43
Medium	24.40	27.47	-3.07
Low	9.40	9.91	-0.51

4.6 Conclusion

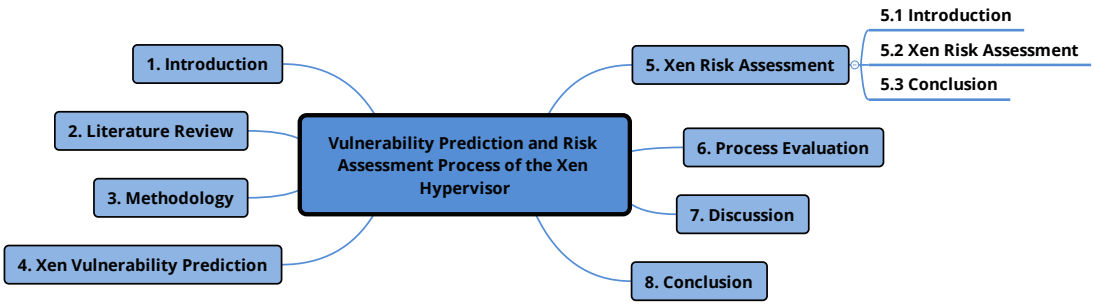
In this chapter, unknown Xen vulnerabilities were predicted for 2018 using the Holt-Winters method. The prediction model predicted 41.85 unknown vulnerabilities when a lower value (0.10) is assigned to the α, β, and γ smoothing parameters. The prediction result (41.85 unknown vulnerabilities) is very close to the average 43.80 reported Xen vulnerabilities during the last five years (2013 to 2017). However, the model

predicted inaccurate results when 0.20 and 0.30 values were assigned to these smoothing parameters. The validity of the prediction model is determined using MAPE. A lower MAPE value (95.53) is observed with 0.10 smoothing parameters. The accuracy of the prediction model is also determined using MAD. The control chart shows that the prediction tracking signal is well between the upper and lower control limits of the chart. To extend the scope of the vulnerability prediction process, unknown vulnerabilities are also predicted with regard to the impact levels using regression analysis. The prediction model predicted accurate results which are very close to the average values of the High, Medium, and Low impact reported vulnerabilities during the last five years.

The prediction shows that 41.85 unknown vulnerabilities may appear in 2018. These unknown vulnerability exploitation scenarios can lead to compromise of the Xen hypervisor based virtualised infrastructure. Therefore, it is desirable to identify the risk these vulnerability exploitations can pose to Xen and determine the risk severity levels. Chapter 5 next covers the risk assessment of Xen. ENISA's risk framework, structured analysis approach, and risk estimation matrix are used to determine the Xen risk and their qualitative severity levels.

Chapter 5

Xen Risk Assessment



5.1 Introduction

In Chapter 4, Xen vulnerability prediction was performed. This chapter provides a risk assessment of the Xen hypervisor. Through risk assessment, the Xen vulnerabilities scenarios are developed and scored to determine the impact ratings. However, the hypervisor related vulnerabilities and their impact ratings are adopted from ENISA to realise a complete risk assessment process as it is not practicable to cover all the Xen exploitation scenarios. Through a structured analysis approach using attack trees, threats are identified and their likelihood levels are determined. Severity levels of risk are then determined by combining the vulnerability impact ratings and threat likelihood

levels. Nine risk categories are listed, and their qualitative severity levels are determined as High, Medium, and Low.

This chapter is organised as follows. Section 5.2 presents a qualitative risk analysis process. Section 5.3 provides the conclusion of this chapter.

5.2 Xen Risk Assessment

This section provides a risk assessment process for the Xen hypervisor which is used as a case in this research. An architecture of Xen and the details of it's selection as a case are covered in Section 1.5.1. A qualitative inductive risk assessment of Xen is performed in this section (Section 1.5.3 provides the description about qualitative vs quantitative, and inductive vs deductive risk assessment). To perform Xen risk assessment, vulnerabilities, assets, and risk types are summarised in Table 5.1. Each risk is correlated with the different types of vulnerabilities and related assets.

Table 5.1: Correlation of Xen Risk, Vulnerabilities, and Assets

Risk No.	**Risk Type**	**Vulnerabilities**	**Assets**
R1	Loss of Business Reputation Due To Co-Tenant Activities	Lack of resource isolation (V1) Hypervisor code vulnerabilities (V2)	Company reputation (A1) Personal sensitive data (A2) Personal data (A3) Service delivery (A4)
R2	Isolation Failure	Lack of resource isolation (V1) Hypervisor code vulnerabilities (V2) Possibility of internal (cloud) network probing (V3) Possibility of co-residence checks (V4)	Company reputation (A1) Personal sensitive data (A2) Personal data (A3) Service delivery (A4) Customer trust (A5)

continued . . .

Correlation of Xen Risk, Vulnerabilities, and Assets ...continued

Risk No.	Risk Type	Vulnerabilities	Assets Affected
R3	Malicious Insider	AAA Vulnerabilities (V5) Inadequate physical security procedures (V6)	Company reputation (A1) Personal sensitive data (A2) Personal data (A3) Service delivery (A4) Customer trust (A5) Employee loyalty and experience (A6) Intellectual property (A7)
R4	Intercepting Data In Transit	Possibility of internal (cloud) network probing (V3) Possibility of co-residence checks (V4) AAA Vulnerabilities (V5)	Company reputation (A1) Personal sensitive data (A2) Personal data (A3) Customer trust (A5) Intellectual property (A7) Backup or archive data (A8)
R5	Data Leakage	Possibility of internal (cloud) network probing (V3) Possibility of co-residence checks (V4) AAA Vulnerabilities (V5)	Company reputation (A1) Personal sensitive data (A2) Personal data (A3) Customer trust (A5) Employee loyalty and experience (A6) Intellectual property (A7) Credentials (A9) Cloud service management interface (A10)
R6	Undertaking Malicious Probes Or Scans	Possibility of internal (cloud) network probing (V3) Possibility of co-residence checks (V4)	Company reputation (A1) Service delivery (A4) Customer trust (A5)
R7	Compromise Hypervisor	Lack of resource isolation (V1) Hypervisor code vulnerabilities (V2)	Personal sensitive data (A2) Personal data (A3) Service delivery (A4)

continued ...

Correlation of Xen Risk, Vulnerabilities, and Assets ...continued

Risk No.	Risk Type	Vulnerabilities	Assets Affected
R8	Privilege Escalation	Hypervisor code vulnerabilities (V2) AAA Vulnerabilities (V5)	Personal sensitive data (A2) Personal data (A3) Access control (A11) User directory - data (A12)
R9	Management Interface Compromise	AAA Vulnerabilities (V5) Remote access to management interface (V7)	Company reputation (A1) Personal sensitive data (A2) Personal data (A3) Service delivery (A4) Cloud service management interface

5.2.1 Examples of Xen Vulnerability Exploitation Scenarios

This section provides the vulnerability exploitation scenarios. Three vulnerability exploitation scenarios are considered from PU and NU TAs. The Xen vulnerabilities are scored using CVSS to determine vulnerability impact ratings from 0 to 10. Vulnerability impact rating is labelled as Low if the Base score is between 0 to 3.9, Medium if the Base score is between 4.0 to 6.9, High if the Base score is between 7.0 to 8.9, and critical if the Base score is between 9.0 to 10.0.

Base Scoring Metric Values

The CVSS uses different metrics and numerical values to determine the vulnerability impact ratings. Table 5.2 provides the metrics, metrics values, and relevant numerical values.

Table 5.2: Metric and Numerical Values of Base Metrics

Metric	**Metric Value**	**Numerical Value**
Attack Vector	Network	.85
	Adjacent Network	.62
	Local	.55
Attack Complexity	Low	.77
	High	.44
Required Privileges	None	.85
	Low	.62 (.68 if the scope is changed)
	High	.27 (.50 if the scope is changed)
User Interaction	None	.85
	Required	.62
CIA Impact	High	.56
	Low	.22
	None	0

Example Scenario 1

Figure 5.1 provides the impact rating of a Physical vulnerability exploited by a PU with physical access to the Xen host server. A PU can exploit a physical vulnerability by misusing physical access, for example, shutting down the server by unplugging the power, or network cable. A PU can also steal the customer's' data by copying it to a removable drive and use it for financial benefits. Such an exploitation scenario is categorised as:

- Attack vector is PHYSICAL
- Attack complexity is low
- Physical Access privileges are required by the attacker to exploit the vulnerability
- End user's interaction is not required to realise the exploitation
- Scope would change (exploitation would affect other components)

- Vulnerability exploitation would impact the confidentiality and availability security objectives

6.8 is the base score calculated using the above scenario, and the vulnerability impact rating is labelled as Medium.

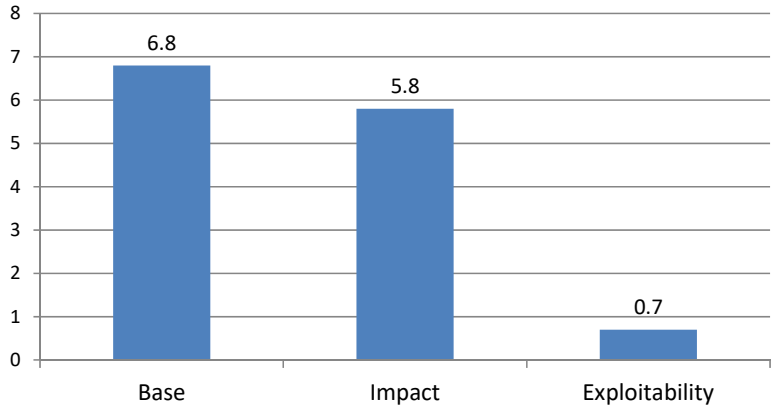


Figure 5.1: Xen Vulnerability Exploited through Physical AV

Example Scenario 2

This section covers the impact rating of a Local vulnerability that can be exploited by an NU. An NU can misuse PV or HVM guest VM user space. So, an NU having a PV guest VM can exploit hypercalls to see another PV guest VMs' requests made to Xen for hardware access. Furthermore, an NU with access to HVM guest VM can exploit a vulnerability that exists in Instruction emulation (MMIO, shadow page tables, and so forth), and emulated platform devices (APIC, HPET, PIT, and so forth). Such an exploitation scenario is categorised as:

- Attack vector is LOCAL
- Attack complexity is low
- Low access privileges are required by the attacker to exploit the vulnerability
- End user's interaction is not required to realise the exploitation
- Scope would not change (exploitation would not affect other components)

- Vulnerability exploitation would impact the confidentiality and integrity security objectives

Figure 5.2 provides the impact rating of a Local vulnerability exploited by a malicious NU. 7.1 is the base score calculated, and the vulnerability impact rating is labelled as High.

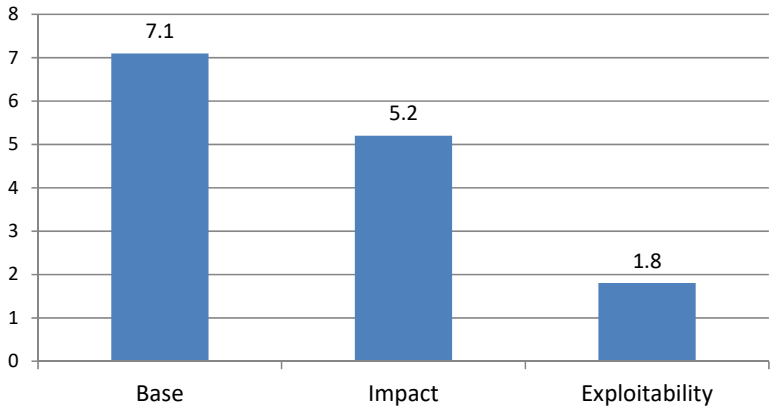


Figure 5.2: Xen Vulnerability Exploited through Local AV

Example Scenario 3

An NU can attack network path by exploiting a Network vulnerability that may exist in hardware driver, bridging and filtering, and netback (netback in case of PV mode). The exploitation allows the attacker to take control of Dom0 kernel, which could lead the attacker to take control of the Xen eventually. An NU with malicious intent can also attack Pygrub that resides within tool-stack of Dom0 by compromising a PV guest VM's disk. This attack is possible due to vulnerabilities that may exist in file system parser, menu parser, domain builder. Furthermore, an NU using HVM guest VM can attack Qemu Device Model by exploiting a vulnerability that may exist in NIC emulator parsing packets, and emulation of virtual devices. Such an exploitation scenario can be categorised as:

- Attack vector is a NETWORK

- Attack complexity is high
- Low access privileges are required by the attacker to exploit the vulnerability
- End user's interaction is not required to realise the exploitation
- Scope would change (exploitation would affect other components)
- Vulnerability exploitation would impact the confidentiality, integrity and availability security objectives

Figure 5.3 provides the impact rating of this type of exploitation scenario. 8.5 is the base score calculated, and the vulnerability impact rating is labelled as High.

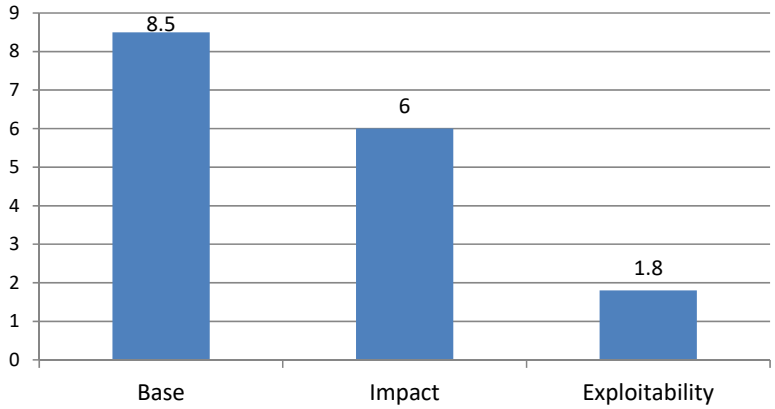


Figure 5.3: Xen Vulnerability Exploited through Network AV

5.2.2 Xen Vulnerability Impact Ratings

This section covers the Xen vulnerability impact ratings. The impact ratings of Xen related vulnerabilities are adopted from ENISA. The vulnerabilities are not scored to determine the impact ratings because it is difficult to determine the impact ratings of all hypervisor vulnerability scenarios due to its complexity and the involvement of more than one stakeholders. Therefore, to realise a complete risk assessment process for Xen, vulnerability impact ratings are adopted from ENISA.

The ENISA determined the impact ratings in terms of loss of confidentiality, integrity, and availability of security objectives. Impact ratings are rated as Very High,

High, Medium, Low, and Very Low. Usually, all vulnerabilities have corresponding threats that may exploit these vulnerabilities (Albakri et al., 2014). Otherwise, these vulnerabilities pose no risk. In CC, it is essential to analyse the vulnerabilities of hypervisors and determine the overall risk to their virtual assets. According to Catteddu and Hogben (2009), vulnerability impact ratings are determined by consulting expert groups. In this Section, Xen vulnerabilities, related assets that can be affected, and risk types are mapped. Table 5.3 provides the mapping of vulnerabilities, related assets, and risk types.

Table 5.3: Xen Vulnerability Impact Ratings

Risk No.	**Risk Type**	**Vuls**	**Impact Rating**	**Reasoning**
R1	Loss of Business Reputation Due To Co-Tenant Activities	V1 V2	High	The impact of V1 and V2 exploitation can affect IaaS delivery and data loss. It can also affect the reputation of cloud customers' businesses.
R2	Isolation Failure	V1 V2 V3 V4	Very High	The impact of these vulnerabilities can result in loss of sensitive data that belongs to customers. Also, can cause reputation and service damage between customers and CSPs.
R3	Malicious Insider	V5 V6	Very High	This type of exploitation scenario would impact the confidentiality, and availability security objectives. It could also impact the Intellectual Property (IP) and damage the reputation of the customers that would result in a business loss.

continued ...

Xen Vulnerability Impact Ratings ...continued

Risk No.	Risk Type	Vuls	Impact Rating	Reasoning
R4	Intercepting Data In Transit	V3 V4 V5	High	The impact of these vulnerabilities results in loss of confidentiality of data in transit. This could also result in loss of IP and customers' trust on CSP because CSPs usually do not offer non-disclosure clauses, or these clauses are not sufficient to ensure the confidentiality of data.
R5	Data Leakage	V3 V4 V5	High	This impact results in loss of confidentiality of data that travels between the CSP and customers. The exploitation impact is high as there are many possible threat sources to exploit these vulnerabilities such as spoofing, sniffing, man-in-the-middle, side channel, and replay attacks.
R6	Undertaking Malicious Probes Or Scans	V3 V4	Medium	The impact of these exploitation scenarios results in loss of confidentiality, availability, and integrity security objectives of data.
R7	Compromise Hypervisor	V1 V2	Very High	These vulnerabilities leverage attacker to compromise hypervisor and result in very high impact. Successful exploitation can lead to the destruction of all the assets managed by the vulnerable hypervisor. The impact could be a severe loss of data and delivery of services.

continued ...

Xen Vulnerability Impact Ratings ...continued

Risk No.	Risk Type	Vuls	Impact Rating	Reasoning
R8	Privilege Escalation	V2 V5	High	These vulnerabilities allow an attacker to gain high privileges. Later, the attacker can use these privileges to compromise the data and assets. The impact could be a loss of credentials and sensitive data.
R9	Management Interface Compromise	V5 V7	Very High	The impact could be very high when an attacker can get access to a large number of cloud resources by compromising the hypervisor management interface.

V1 = Lack of resource isolation, V2 = Hypervisor code vulnerabilities
V3 = Possibility of internal network probing, V4 = Possibility of co-residence checks
V5 = AAA Vulnerabilities, V6 = Inadequate physical security procedures
V7 = Remote access to management interface

5.2.3 Threat Identification and Likelihood Assessment

The next step for Xen risk assessment is to identify threats and determine their likelihood levels. PU and NU TAs are considered for threat likelihood assessment of Xen. A PU is considered as an employee of a CSP, and an NU is a customer who owns a VM (See Appendix A, Section 3.4.4 for a description of these TAs). The potential attacker's goal is identified through threat likelihood assessment. An attacker manipulates the Xen hypervisor and impacts the CIA security objectives of virtual assets.

Capability and motivation properties of PU and NU are assigned a value from 1 to 5 (CESG, 2009). These values are then combined using Table 5.4 to calculate initial threat levels (Negligible considered as level 1, Low as level 2, Moderate as level 3, Substantial as level 4, and Severe as level 5). These threat levels are assigned to the source nodes of the Xen attack tree. Capability and motivation properties are defined on the basis of significant professional judgment as there is no firm rule to define them. However,

assuming the worst-case scenario to define capability and motivation properties can overestimate threat levels. Therefore, judgements are optimistic. Table 5.4 provides the capability and motivation properties and their qualitative values.

Table 5.4: Threat Likelihood Matrix

Motivation	**Capability Level**				
	Very Little	**Little**	**Limited**	**Significant**	**Formidable**
Indifferent	Negligible	Negligible	Low	Low	Moderate
Curious	Negligible	Negligible	Low	Moderate	Substantial
Interested	Negligible	Low	Moderate	Substantial	Severe
Committed	Low	Low	Moderate	Severe	Severe
Focused	Low	Moderate	Substantial	Severe	Severe

Figure 5.4 provides a Xen attack tree that consists of a set of attack trees combined to build a cyclic attack tree.

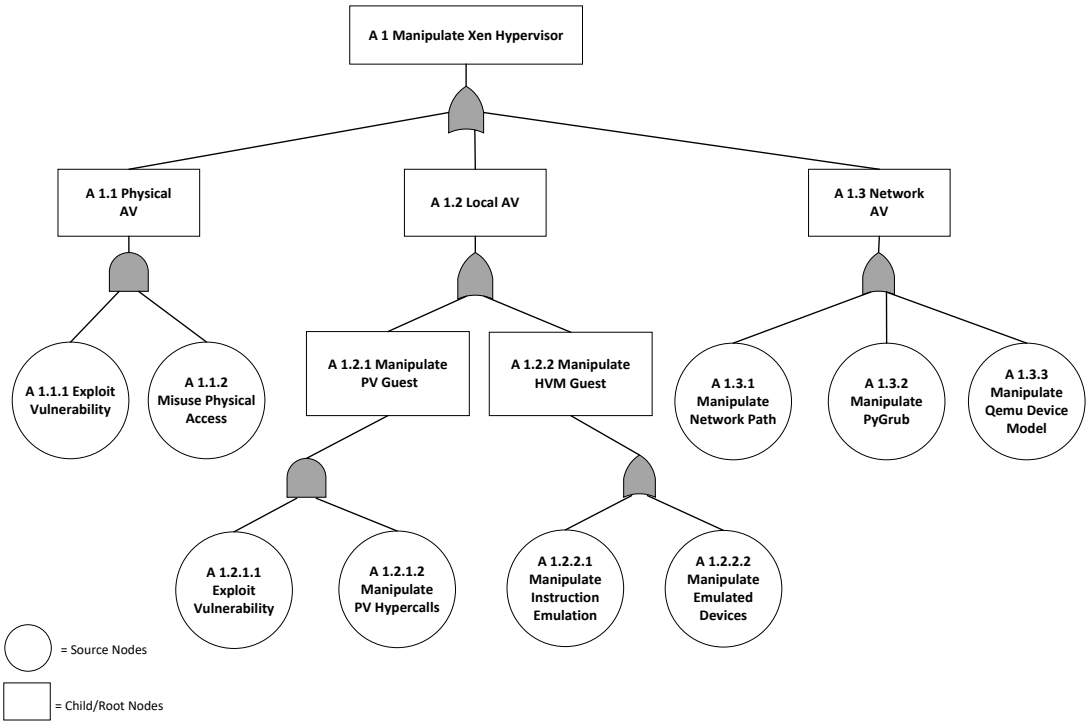


Figure 5.4: Xen Attack Tree

Threat Levels to A 1.1 Attack Step

In this section, threat likelihood levels are assigned to source nodes A 1.1.1 and A 1.1.2. Figure 5.5 provides tree branch connected to the A 1.1 child node.

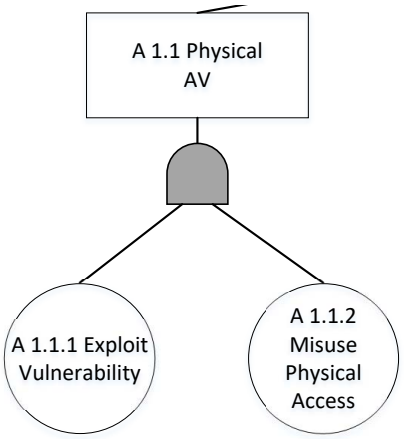


Figure 5.5: Threat Levels to A 1.1

A 1.1.1 Exploit Physical Vulnerability Threat level assignment starts with the source nodes connected to A 1.1 (Physical Attack). So, the threat levels are assigned to A 1.1.1 and A 1.1.2 source nodes. For the attack step A 1.1.1 (Exploit Physical Vulnerability) minimum technical knowledge is required to exploit a physical vulnerability through access by unplugging the power or network cable. Therefore, both the TAs are assumed capable enough to exploit the physical vulnerability. However, the motivation value of NU is less because NUs normally do not have physical access to CSP's location. Table 5.5 provides the threat levels of A 1.1.1 attack step.

Table 5.5: Exploit Physical Xen Vulnerability

A 1.1.1	Capability	Motivation	Likelihood
PU	4	2	3
NU	4	2	3

A 1.1.2 Misuse Physical Access Threat levels are assigned to the second source node of A 1.1 attack step that is, A 1.1.2 (Physical Access). Physical security of cloud infrastructure is the sole responsibility of the CSP. So, usually the Xen host server is provided with adequate physical access protection, such as a locked data centre, video surveillance, and a scanning system. However, the cloud administrator (PU) usually has physical access to manage Xen. If the administrator is influenced by a malicious party to exploit the server, it can lead to physical damage or data theft by using a removable disk drive. The administrator is the only TA who is physically present on the premises where the host server is installed. Therefore, both the capability and motivation (if influenced) values are considered as high for PU. On the other hand, an NU with malicious intent is motivated to exploit a physical vulnerability to steal the data if he gets a chance. However, in most of the cases, NUs do not even know the location of the CSP data centre and thus cannot access it. Table 5.6 provides threat levels assigned to the A 1.1.2 attack step.

Table 5.6: Misuse Physical Xen Access

A 1.1.2	**Capability**	**Motivation**	**Likelihood**
PU	4	3	4
NU	1	2	1

Threat levels of this tree branch are not propagated to A 1.1 child nodes as shown in Figure 5.5.

A 1.1 Physical Attack Threat level from the source nodes A 1.1.1 and A 1.1.2 are propagated to the A 1.1 child node. As shown in Figure 5.5, this tree branch does not contain a loop. Thus, the threat propagation is very simple. Both the source nodes A 1.1.1 and A 1.1.2 connected with A 1.1 by a logical *AND* operator.

Therefore, the minimal threat level of these two source nodes is propagated. Table 5.7 provides the threat levels propagated to A 1.1 attack step.

Table 5.7: Threat Likelihood Level at Physical AV

Type:*AND*	**Likelihood for A 1.1.1**	**Likelihood for A 1.1.2**	**Likelihood for A 1.1**
PU	3	4	3
NU	3	1	1

Threat Levels to A 1.2 Attack Step

In this section, threat likelihood levels are assigned to source nodes A 1.2.1.1, A 1.2.1.1, A 1.2.2.1, and A 1.2.2.2. Figure 5.6 provides the tree branch connected to A 1.2 attack step.

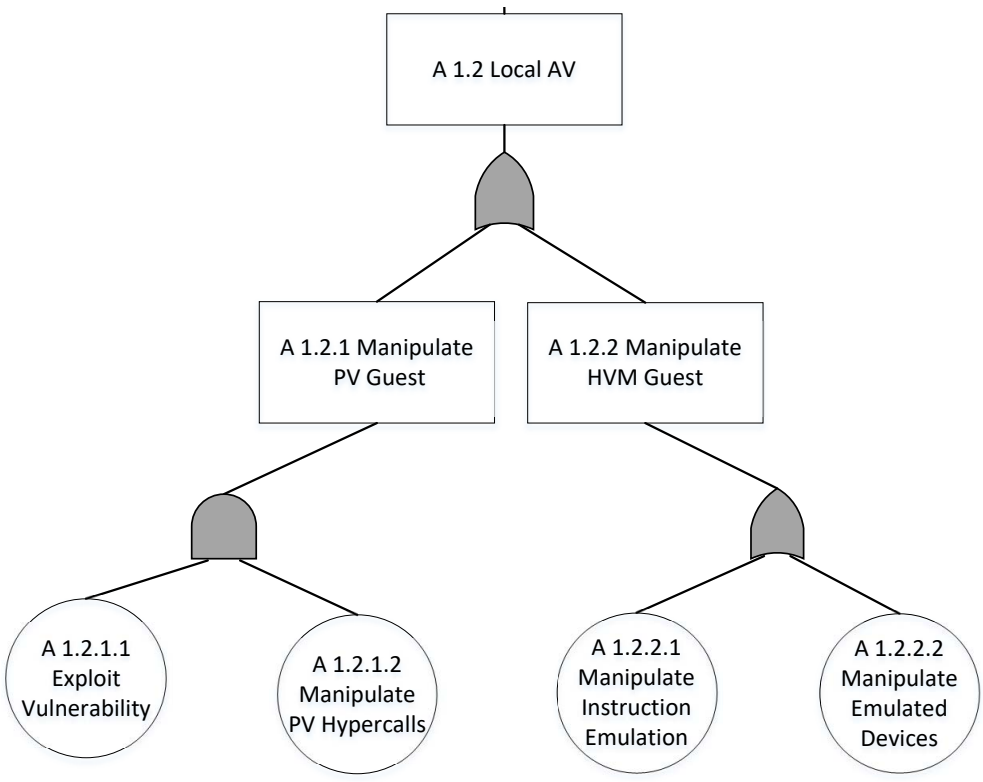


Figure 5.6: Threat Levels to A 1.2

A 1.2.1.1 Exploit Vulnerability An NU is considered as an authorised user of Xen based PV guest VM. So, an NU user with malicious intent can attack hypercall functionality by exploiting a vulnerability, for example, Xen's mishandling of page tables and sub-operations in the FLASK (security module). Therefore, an NU is more capable when compared to a PU seeking to exploit a vulnerability by misusing privileged VM space. Table 5.8 provides threat levels assigned to the A 1.2.1.1 source node of the attack step A 1.2.1.

Table 5.8: Exploit Local Vulnerability

A 1.2.1.1	Capability	Motivation	Likelihood
PU	2	2	1
NU	3	2	2

A 1.2.1.2 Manipulate PV Hypercalls An NU, if motivated by an internal or external malicious party, poses a serious threat to Xen. An NU can misuse privileged PV guest user space to exploit a vulnerability that may exist in a Xen function that handles PV hypercalls. The successful exploitation allows an NU to see another PV guest VMs' requests made to Xen for hardware access. Table 5.9 provides threat levels assigned to the second A 1.2.1.2 source node of the attack step A 1.2.1.

Table 5.9: Manipulate PV Hypercalls

A 1.2.1.2	Capability	Motivation	Likelihood
PU	1	2	1
NU	3	5	4

A 1.2.2.1 Manipulate Instruction Emulation The next attack step A 1.2.2, has two

source nodes: A 1.2.2.1 and A 1.2.2.2. Here, threat levels are assigned to A 1.2.2.1. A motivated NU with access to the HVM guest VM can exploit a vulnerability that may exist in Instruction emulation (MMIO, shadow page tables, and so forth) to cause a DoS (host crash) or obtain sensitive information. Table 5.10 provides threat levels assigned to the A 1.2.2.1 source node of A 1.2.2 attack step.

Table 5.10: Manipulate Instruction Emulation

A 1.2.2.1	Capability	Motivation	Likelihood
PU	1	3	1
NU	3	4	3

A 1.2.2.2 Manipulate Emulated Platform Devices The second source node of A 1.2.2 attack step is assigned with the threat levels. A motivated NU with access to HVM guest VM can exploit a vulnerability that exists in emulated platform devices (APIC, HPET, PIT, and so forth) to cause a DoS (host crash) or gain high-level privileges. Table 5.11 provides threat levels for the A 1.2.2.2 source node.

Table 5.11: Manipulate Emulated Platform Devices

A 1.2.2.2	Capability	Motivation	Likelihood
PU	1	3	1
NU	3	4	3

Threat levels are now propagated to A 1.2.1, A 1.2.2. Threat levels are also propagated to an A 1.2 child node of this tree branch as shown in Figure 5.6.

A 1.2.1 Manipulate Host Server Hardware As shown in Figure 5.6, the child node A 1.2.1 has two source nodes, A 1.2.1.1 and A 1.2.1.2, which are connected by a logical *AND* operator. Therefore, the attacker must complete both A 1.2.1.1 and

A 1.2.1.2 source nodes to realise this attack step. So, the minimal threat level from these two source nodes is propagated to A 1.2.1. Table 5.12 provides the threat levels to A 1.2.1 attack step.

Table 5.12: Manipulate Host Server Hardware

Type:*AND*	**Likelihood for A 1.2.1.1**	**Likelihood for A 1.2.1.2**	**Likelihood for A 1.2.1**
PU	1	1	1
NU	2	4	2

A 1.2.2 Manipulate HVM Guest Threat levels are now propagated to child node A 1.2.2. Both the source nodes A 1.2.2.1 and A 1.2.2.2 are connected to the A 1.2.2 child node with a logical *OR* operator. Therefore, the maximal level of threat from these two source nodes is propagated. Table 5.13 provides the threat levels to A 1.2.2 attack step.

Table 5.13: Manipulate HVM Guest

Type:*OR*	**Likelihood for A 1.2.2.1**	**Likelihood for A 1.2.2.2**	**Likelihood for A 1.2.2**
PU	1	1	1
NU	3	3	3

A 1.2 Local Attack Child nodes A 1.2.1 and A 1.2.2 are connected to A 1.2 by a logical *OR*. The attacker can compromise either A 1.2.1 or A 1.2.2 to achieve the target. Therefore, the maximal threat level from these source nodes is propagated to A 1.2. Table 5.14 provides the threat levels for A 1.2 attack step.

Table 5.14: Threat Likelihood Level at Local AV

Type:*OR*	**Likelihood for A 1.2.1**	**Likelihood for A 1.2.2**	**Likelihood for A 1.2**
PU	1	1	1
NU	2	3	3

Threat Levels to A 1.3 Attack Step

In this section, threat likelihood levels are assigned to source nodes A 1.3.1, A 1.3.2, and A 1.3.3. Figure 5.7 provides a tree branch connected to A 1.3 attack step.

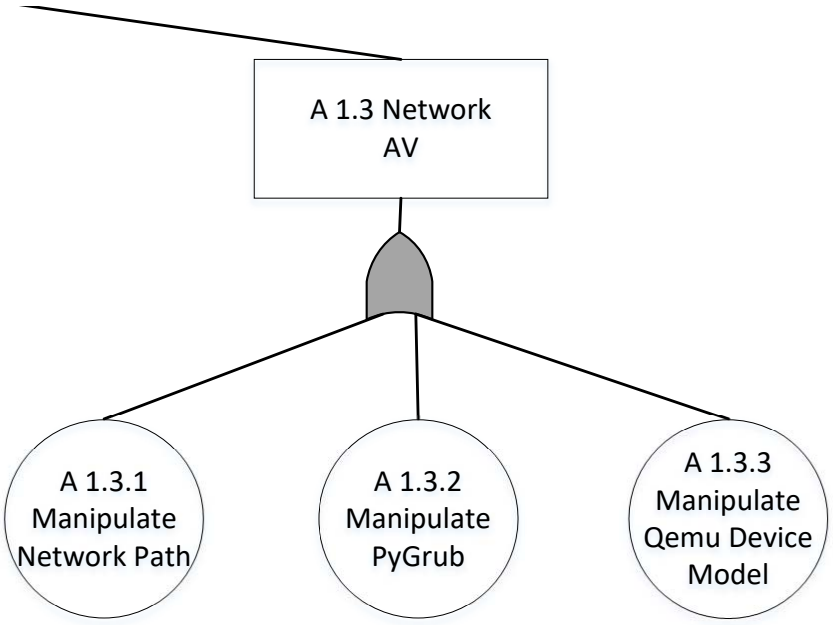


Figure 5.7: Threat Levels to A 1.3

A 1.3.1 Manipulate Network Path Source node A 1.3.1 which is connected to A 1.3 attack step is considered. An NU with malicious intent could attack a network stack and exploit a vulnerability that may exist in the hardware driver, bridging and filtering, and netback (netback in case of PV mode) to get control of a guest VM's NIC. The successful exploitation would allow NU taking control of Dom0

kernel. The exploitation could eventually lead an NU to take control of Xen. Table 5.15 provides threat levels to the A 1.3.1 source node.

Table 5.15: Manipulate Network Path

A 1.3.1	Capability	Motivation	Likelihood
PU	1	2	1
NU	4	5	5

A 1.3.2 Manipulate PyGrub Threat levels are assigned to source node A 1.3.2. An NU with malicious intent can attack Pygrub that resides within the toolstack of Dom0 by compromising a PV guest VM's disk due to vulnerabilities that may exist in the file system parser, menu parser, and domain builder. Table 5.16 provides threat levels to 1.3.2 source node of the attack step A 1.3.

Table 5.16: Manipulate PyGrub

A 1.3.2	Capability	Motivation	Likelihood
PU	1	2	1
NU	3	4	3

A 1.3.3 Manipulate Qemu Device Model An NU using HVM guest VM can attack Qemu Device Model by exploiting a vulnerability that may exist in a NIC emulator that is parsing packets and emulating virtual devices. The exploitation would allow the attacker to get access to Dom0 privileged user space. The exploitation could result in the compromise of Xen. Table 5.17 provides threat levels to 1.3.3 source node.

Table 5.17: Manipulate Qemu Device Model

A 1.3.3	**Capability**	**Motivation**	**Likelihood**
PU	1	2	1
NU	3	4	3

Threat levels are now propagated to the A 1.3 child node of this tree branch, Figure 5.7.

A 1.3 Network Attack The source nodes, A 1.3.1, A 1.3.2, and A 1.3.3 are connected to A 1.3 child node by a logical *OR*. The maximal level of threat from child nodes is propagated to A 1.3. Table 5.18 provides the threat levels to A 1.3 attack step.

Table 5.18: Threat Likelihood Level at Network AV

Type:*OR*	**Likelihood for A 1.3.1**	**Likelihood for A 1.3.2**	**Likelihood for A 1.3.3**	**Likelihood for A 1.3**
PU	1	1	1	1
NU	5	3	3	5

Propagating Threat Levels to A 1 Root Node

Threat levels from Physical, Local, and Network child nodes are now propagated to A 1 (attacker' main goal) as shown in Figure 5.8.

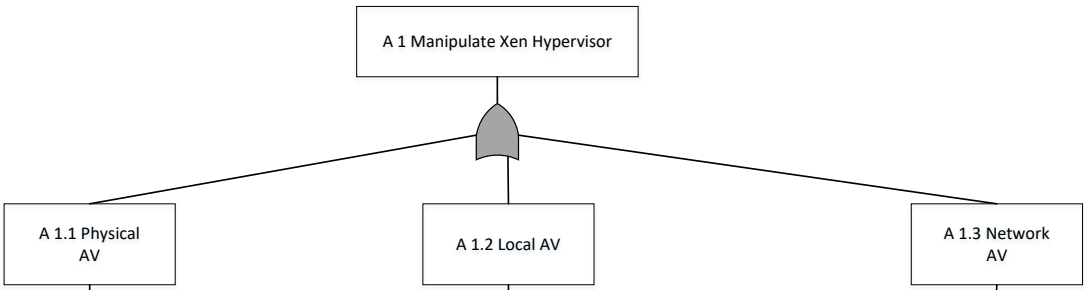


Figure 5.8: Threat Levels to A 1

A 1 Manipulate Xen Hypervisor Finally, the threat levels are propagated to the root node A 1. All three child nodes A 1.1, A 1.2, and A 1.3 are connected to the root node by a logical *OR*. Therefore, the maximal level of threat from these child nodes is propagated. Table 5.19 provides the threat levels to A 1 attack step by considering the threat levels provided in Table 5.7, Table 5.14, and Table 5.18.

Table 5.19: Threat Likelihood Level to Xen

Type:*OR*	**Likelihood for A 1.1 (Physical AV)**	**Likelihood for A 1.2 (Local AV)**	**Likelihood for A 1.3 (Network AV)**	**Likelihood for A 1 (Overall)**
PU	3	1	1	**3**
NU	1	3	5	**5**

Threat Likelihood Levels from PU and NU

Table 5.19 provides the overall threat likelihood levels to Xen from both PU and NU. A Moderate (3) threat likelihood level is determined from a PU. On the other hand, a Severe (5) threat likelihood level is determined from an NU. Figure 5.10 shows that an NU is most likely going to realise a network attack by exploiting a vulnerability that exists in the network stack of Dom0. The threat likelihood levels are coloured red for Severe, orange for Substantial, dark yellow for Moderate, yellow for Low, and green

for Negligible.

Threat Likelihood Levels from PU Figure 5.9 presents a combined threat likelihood level determined at A 1 root node a PU. The overall threat likelihood level to Xen is **Moderate**. It also presents likelihood levels of threats exploiting Physical, Local, and Network AVs determined at child nodes A 1.1, A 1.2, and A 1.3. A **Moderate** threat likelihood level is determined at child node A 1.1. A PU exploiting a Local AV results in **Negligible** threat likelihood level determined at child node A 1.2. At child node A 1.3, also a **Negligible** threat likelihood level determined if a PU exploits Network AV.

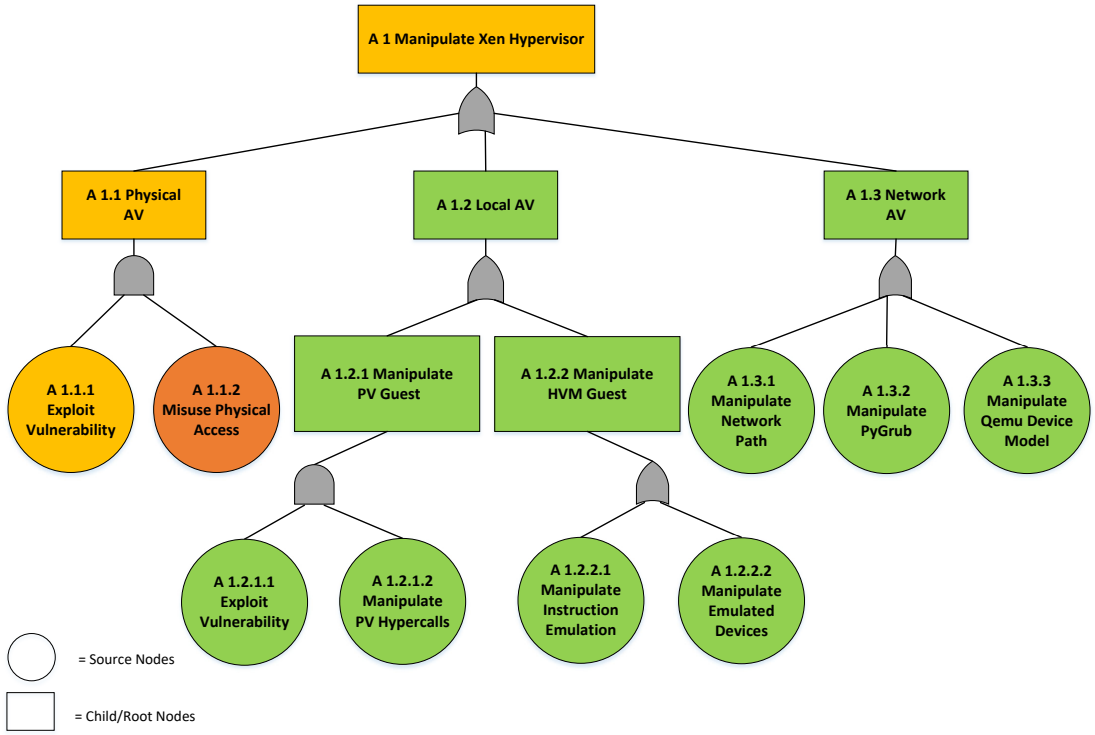


Figure 5.9: Threat Likelihood Levels from PU

Threat Likelihood Levels from an NU Figure 5.10 presents a combined threat likelihood level determined at A 1 root node from an NU. The overall threat likelihood level is determined as **Severe**. **Negligible**, **Moderate**, and **Severe** threat likelihood levels are determined at child nodes A 1.1, A 1.2, and A 1.3 if an NU

exploits a vulnerability through Physical, Local, and Network AVs respectively.

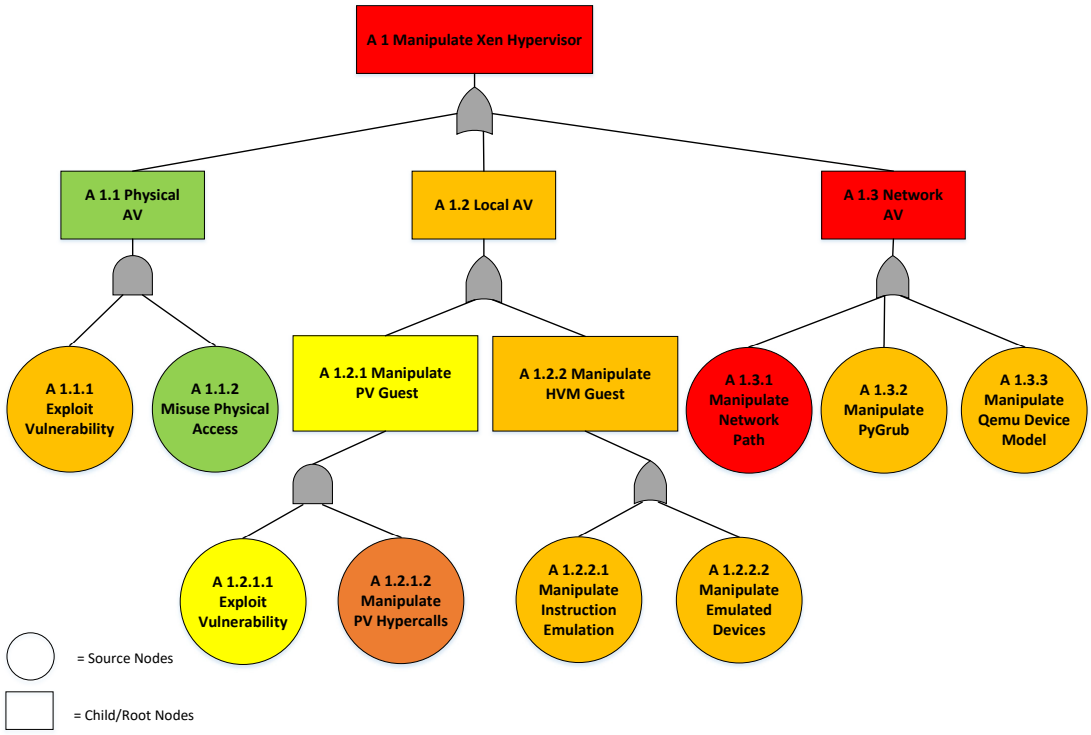


Figure 5.10: Threat Likelihood Levels from NU

5.2.4 Determination of Risk Severity Levels

In this section, risk severity levels are determined using the ENISA's risk estimation matrix provided in Table 5.20. Nine different risks are listed to determine their severity levels. AVs are identified from where these risk can be realised by the attacker. Corresponding vulnerability impact ratings and threat likelihood levels for each of the risk are mapped through the matrix to determine the severity levels of each risk to Xen. Qualitative severity levels are determined as High, Medium, and Low if the risk scale is from 7-9, 4-6, and 1-3, respectively.

Table 5.20: Risk Estimation Matrix

Vulnerability Impact	**Threat Likelihood**				
	Negligible	Low	Moderate	Substantial	Severe
Very Low	1	2	3	4	5
Low	2	3	4	5	6
Medium	3	4	5	6	7
High	4	5	6	7	8
Very High	5	6	7	8	9

Risk Severity Levels from a PU

Table 5.21 provides the severity levels of risk to Xen from a PU. High risk severity levels are determined for R3 (Malicious Insider) and R7 (Compromise Hypervisor). An administrator of the Xen host server, if influenced by a malicious threat source can steal data and information that is stored on VMs running on the installed hypervisor. This would lead to the compromise of CIA security objectives. Therefore, customers must ensure that CSPs follow fair hiring procedures. CSPs must perform proper background and security checks for new hires, and staff profiles should not be public to prevent targeted approaches. Conversely, R6 (Undertaking Malicious Probes or Scans) poses a Low severity level from a PU. All another risk; R1, R2, R4, R5, R8, and R9 pose Medium severity levels. These six risk types should be managed after addressing the risk with High severity levels.

Table 5.21: Risk Severity Levels from PU

Risk Type	**Xen Attack Vector**	**Vul Impact Rating**	**Threat Likelihood Level**	**Level of Risk**
Loss of Business Reputation Due to Malicious Co-Tenant	Local	High	Negligible	Medium
Isolation Failure	Local	Very High	Negligible	Medium
Malicious Insider	Physical	Very High	Moderate	High

continued . . .

Risk Severity Levels from PU ...continued

Risk Type	Xen Attack Vector	Vul Impact Rating	Threat Likelihood Level	Level of Risk
Intercepting Data in Transit	Network	High	Negligible	Medium
Data Leakage	Local	High	Negligible	Medium
Undertaking Malicious Probes or Scans	Network	Medium	Negligible	Low
Compromise Hypervisor	Physical	Very High	Moderate	High
Privilege Escalation	Local	High	Negligible	Medium
Management Interface Compromise	Network	Very High	Negligible	Medium

Risk Levels from an NU

Table 5.22 provides the severity levels of risk to Xen from an NU. Results show that an NU with malicious intent poses a High risk severity levels for R2 (Isolation Failure), R4 (Intercepting Data in Transit), R6 (Undertaking Malicious Probes or Scans), R7 (Compromise of Hypervisor), and R9 (Management Interface Compromise) by misusing privileged guest VM space to exploit the host OS or Xen. The high severity level for R2 is determined when a malicious VM user exploits a local AV through access to shared hardware resources such as cache memory and hard disk. R4, R6, R7, and R9 pose High severity levels where a malicious user can exploit a network vulnerability to manipulate virtual network shared between the VM and the hypervisor. Therefore, CSPs need to ensure that the hypervisor is hardened and configured properly to mitigate R2. To mitigate R4, R6, R7, and R9, CSPs must configure and implement adequate network security controls. Conversely, R1, R3, R5, and R8 poses Medium severity levels from an NU TA and can be managed when the risk with High severity are mitigated.

Table 5.22: Risk Severity Levels from an NU

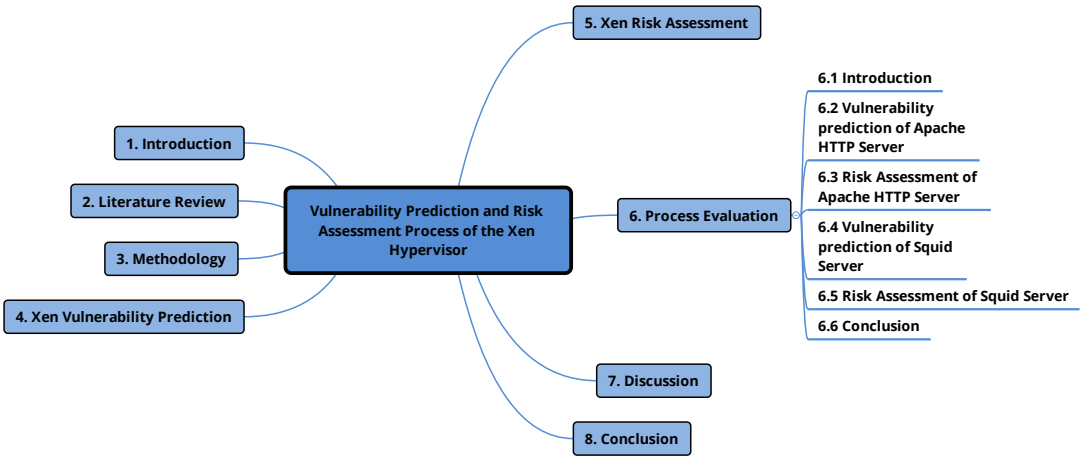
Risk Type	Xen Attack Vector	Vul Impact Rating	Threat Likelihood Level	Level of Risk
Loss of Business Reputation Due to Malicious Co-Tenant	Local	High	Moderate	Medium
Isolation Failure	Local	Very High	Moderate	High
Malicious Insider	Physical	Very High	Negligible	Medium
Intercepting Data in Transit	Network	High	Severe	High
Data Leakage	Local	High	Moderate	Medium
Undertaking Malicious Probes or Scans	Network	Medium	Severe	High
Compromise Hypervisor	Network	Very High	Severe	High
Privilege Escalation	Local	High	Moderate	Medium
Management Interface Compromise	Network	Very High	Severe	High

5.3 Conclusion

In this chapter, risk assessment of the Xen hypervisor was performed. Vulnerability impact ratings were adopted from ENISA's risk framework. Threat likelihood assessment was performed through a structured analysis approach using attack trees. Capability and motivation properties of a PU and NU were used to determine initial threat levels and assigned to all the source nodes of the Xen attack tree. Later, these threat levels were propagated to determine threat likelihood levels for Physical, Local, Network AVs, and Xen. Vulnerability impact ratings and threat likelihood levels were then combined using ENISA's risk estimation matrix to determine the severity levels of the risk to Xen. Chapter 6 next covers the evaluation of the process by applying it to Apache and Squid servers.

Chapter 6

Process Evaluation



6.1 Introduction

Chapter 5 covered vulnerability prediction and risk assessment of the Xen hypervisor. In this chapter, the process is evaluated by applying it to two other open source infrastructure level software packages. Apache HTTP and Squid Proxy servers are selected to demonstrate the generalisability and applicability of the process to open source software packages. Moreover, Apache and Squid are targeted because of their wide use as a web cache and proxy servers respectively. Moreover, these software packages are selected

because their reported vulnerability data is completely available through vulnerability databases.

Apache has 92% of its copies run on Linux platforms. It is a reliable and efficient web server that covers approximately 67% of the market (Space, 2017). It is in use by some large companies such as Cisco, IBM, Salesforce, General Electric, Adobe, VMware, Xerox, Hewlett-Packard, Siemens, eBay, and many more (Kinsta, 2018). Organisations can customise Apache to meet their requirements by adding extensions and modules. However, Apache is often the target of attacks such as DoS, Buffer Overflow, Cross Site Scripting (XSS), Information Leakage, Input Validation, SQL Injection, Session Hijacking, and Phishing. These attacks exploit Apache vulnerabilities and present risk (Acunetix, 2017). A successful attack enables an attacker to gain access to Apache and compromise the data. On the other hand, Squid is in use by many organisations using GNU's General Public License (GPL) of the Free Software Foundation. Squid server is a popular HTTP proxy implementation to provide forward and reverse proxy scenarios (Squid-cache, 2009). Many Internet Service Providers (ISPs) are using Squid as a proxy server since 1990. Therefore, these software packages are worth looking into for their security and also to evaluate the process.

To perfrom the vulnerability prediction and risk assessment of Apache and Squid servers, the Holt-Winters method, Regression Analysis, CVSS, Structure Analysis, and Risk Estimation Matrix methods are used. However, reported vulnerability data from 2008 to 2017 is used for prediction instead of data from 2013 to 2017 (which is the case in Xen). Apache and Squid reported vulnerability data from 2013 to 2017 did not result in accurate prediction. Figure 6.1 provides the Apache and Squid vulnerability prediction process. The only difference is the size of the reported vulnerability data (marked yellow) as compared to the Xen vulnerability prediction process provided in Section 3.3.2 and Figure 3.3.

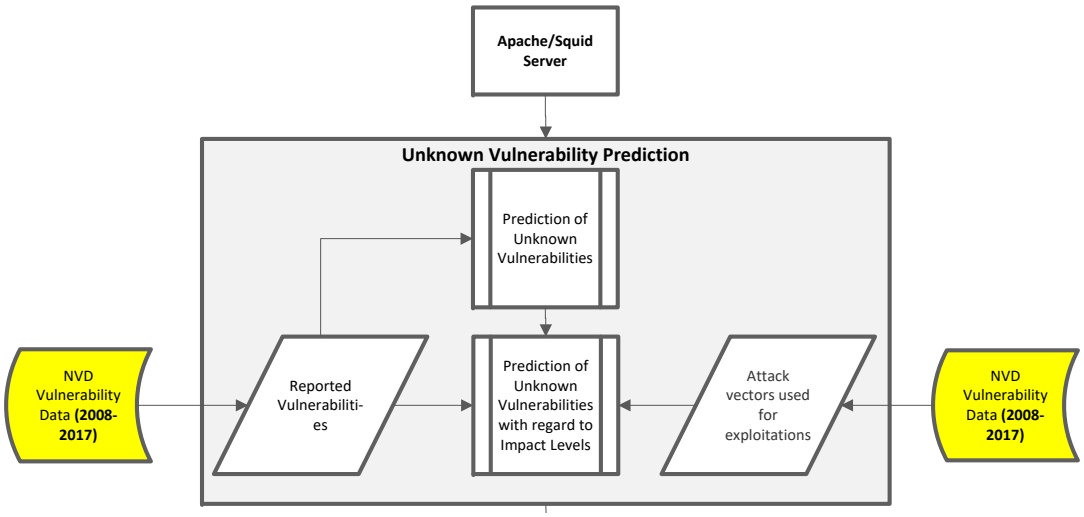


Figure 6.1: Apache and Squid Vulnerability Prediction

For Apache and Squid risk assessment, vulnerability impact ratings are determined using CVSS. Vulnerability impact ratings are not adopted like Xen. However, the same research methods are used to determine threat likelihood and risk severity levels. Figure 6.2 provides the Apache and Squid risk assessment process.

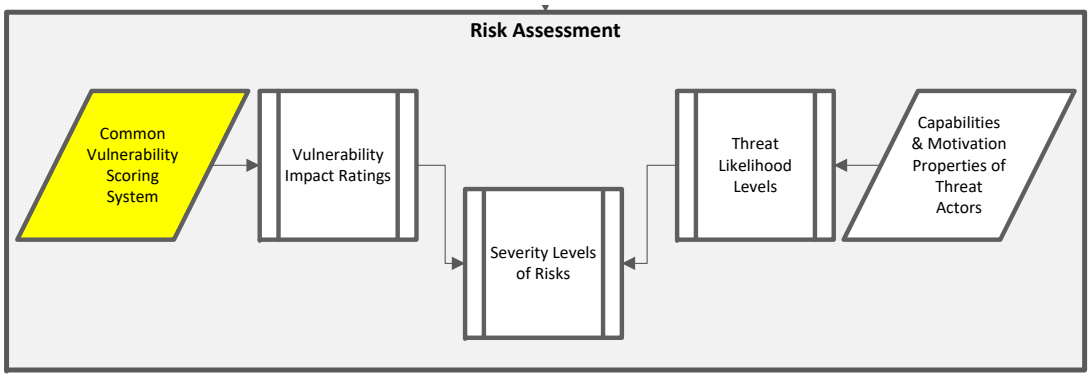


Figure 6.2: Apache and Squid Risk Assessment

This chapter is organised as follows. Sections 6.2 and 6.3 provide the vulnerability prediction and risk assessment process of the Apache HTTP server. The vulnerability prediction process of the Squid proxy server is provided in Section 6.4 followed by the risk assessment process in Section 6.5. The conclusion of this chapter is provided in Section 6.6.

6.2 Vulnerability Prediction of Apache HTTP Server

The process is applied to Apache to perform the vulnerability prediction. The main purpose of the vulnerability prediction of Apache HTTP server is to ensure the generalisability of the process. The vulnerability prediction process would provide organisations a platform to determine the effectiveness of the configuration and security of Apache within the organisation's computing environment. However, the security of Apache HTTP Server depends on the security of the host OS. If the host is not secure and have vulnerabilities, then the Apache running on the same OS is vulnerable. Therefore, security of the host OS should be ensured first to realise the security of the Apache.

6.2.1 Prediction of Unknown Apache Vulnerabilities

In this section, unknown Apache vulnerabilities are predicted for 2018 using the Holt-Winters method. Reported Apache vulnerability data from 2008 to 2017 is used for prediction. According to the NVD search, 100 vulnerabilities were reported in Apache from 2008 to 2017. Table 6.1 provides the reported Apache vulnerabilities from 2008 to 2017.

Table 6.1: Reported Apache Vulnerabilities

Time(t)	Year	Quarter	Vulnerabilities
1	2008	1	8
2		2	2
3		3	2
4		4	0
5	2009	1	1
6		2	3
7		3	5
8		4	2
9	2010	1	6
10		2	3
11		3	2
12		4	0

continued ...

Reported Apache Vulnerabilities ...continued

Time(t)	Year	Quarter	Vulnerabilities
13	2011	1	1
14		2	4
15		3	3
16		4	6
17	2012	1	4
18		2	1
19		3	5
20		4	1
21	2013	1	2
22		2	3
23		3	4
24		4	1
25	2014	1	2
26		2	1
27		3	6
28		4	2
29	2015	1	1
30		2	0
31		3	4
32		4	0
33	2016	1	0
34		2	0
35		3	3
36		4	1
37	2017	1	0
38		2	4
39		3	7
40		4	0

Table 6.2 provides $\bar{S}_t$ for all the time periods, from t_1 to t_{40}.

Table 6.2: Initial Seasonal Factor Values

Time(t)	Year	Quarter	D_t	Deseasonalised Data	$\bar{D}_t$	$\bar{S}_t$
1	2008	1	8	-	3.07	2.61
2		2	2	-	3.03	0.66
3		3	2	2.13	2.99	0.67
4		4	0	1.38	2.96	0.00
5	2009	1	1	1.88	2.92	0.34

continued ...

Initial Seasonal Factor Values ... continued

Time(t)	**Year**	**Quarter**	D_t	**Deseasonalised Data**	$\bar{D}_t$	$\bar{S}_t$
6		2	3	2.50	2.89	1.04
7		3	5	3.38	2.85	1.75
8		4	2	4.00	2.81	0.71
9	2010	1	6	3.63	2.78	2.16
10		2	3	3.00	2.74	1.09
11		3	2	2.13	2.71	0.74
12		4	0	1.63	2.67	0.00
13	2011	1	1	1.88	2.63	0.38
14		2	4	2.75	2.60	1.54
15		3	3	3.88	2.56	1.17
16		4	6	3.88	2.53	2.37
17	2012	1	4	3.75	2.49	1.61
18		2	1	3.38	2.45	0.41
19		3	5	2.50	2.42	2.07
20		4	1	2.50	2.38	0.42
21	2013	1	2	2.63	2.35	0.85
22		2	3	2.50	2.31	1.30
23		3	4	2.50	2.27	1.76
24		4	1	2.25	2.24	0.45
25	2014	1	2	2.25	2.20	0.91
26		2	1	2.63	2.17	0.46
27		3	6	2.63	2.13	2.82
28		4	2	2.38	2.09	0.95
29	2015	1	1	2.00	2.06	0.49
30		2	0	1.50	2.02	0.00
31		3	4	1.13	1.99	2.01
32		4	0	1.00	1.95	0.00
33	2016	1	0	0.88	1.91	0.00
34		2	0	0.88	1.88	0.00
35		3	3	1.00	1.84	1.63
36		4	1	1.50	1.81	0.55
37	2017	1	0	2.50	1.77	0.00
38		2	4	2.88	1.73	2.31
39		3	7	-	1.70	4.12
40		4	0	-	1.66	0.00

After obtaining $\bar{S}_t$, Seasonal Factor for a given *t* is obtained by averaging $\bar{S}_t$ values that correspond to similar *t* periods. As $p = 4$, Time $(t) = 40$ and seasonal cycles (r)

= 10, initial S_1, S_2, S_3 and S_4 are calculated in Equation 6.1. L_0 and T_0 are already calculated as 3.10 and -0.04, respectively.

$$\bar{S}_i = \frac{\sum_{j=0}^{r=l} S_j p+i}{r} \tag{6.1}$$

S_1 = 0.93

S_2 = 0.88

S_3 = 1.87

S_4 = 0.55

Table 6.3: Initial S_t Values of Apache Data

Time(t)	**Year**	**Quarter**	D_t	**Deseasonalised Data**	$\bar{D}_t$	$\bar{S}_t$	Initial S_t Values
1	2008	1	8	-	3.07	2.61	0.93
2		2	2	-	3.03	0.66	0.88
3		3	2	2.13	2.99	0.67	1.87
4		4	0	1.38	2.96	0.00	0.55
5	2009	1	1	1.88	2.92	0.34	-
6		2	3	2.50	2.89	1.04	-
7		3	5	3.38	2.85	1.75	-
8		4	2	4.00	2.81	0.71	-
9	2010	1	6	3.63	2.78	2.16	-
10		2	3	3.00	2.74	1.09	-
11		3	2	2.13	2.71	0.74	-
12		4	0	1.63	2.67	0.00	-
13	2011	1	1	1.88	2.63	0.38	-
14		2	4	2.75	2.60	1.54	-
15		3	3	3.88	2.56	1.17	-
16		4	6	3.88	2.53	2.37	-
17	2012	1	4	3.75	2.49	1.61	-
18		2	1	3.38	2.45	0.41	-
19		3	5	2.50	2.42	2.07	-
20		4	1	2.50	2.38	0.42	-
21	2013	1	2	2.63	2.35	0.85	-
22		2	3	2.50	2.31	1.30	-
23		3	4	2.50	2.27	1.76	-

continued ...

Initial S_t Values of Apache Data ...continued

Time(Y)	**Year**	**Quarter**	D_t	**Deseasonalised Data**	$\bar{D}_t$	$\bar{S}_t$	Initial S_t Values
24		4	1	2.25	2.24	0.45	-
25	2014	1	2	2.25	2.20	0.91	-
26		2	1	2.63	2.17	0.46	-
27		3	6	2.63	2.13	2.82	-
28		4	2	2.38	2.09	0.95	-
29	2015	1	1	2.00	2.06	0.49	-
30		2	0	1.50	2.02	0.00	-
31		3	4	1.13	1.99	2.01	-
32		4	0	1.00	1.95	0.00	-
33	2016	1	0	0.88	1.91	0.00	-
34		2	0	0.88	1.88	0.00	-
35		3	3	1.00	1.84	1.63	-
36		4	1	1.50	1.81	0.55	-
37	2017	1	0	2.50	1.77	0.00	-
38		2	4	2.88	1.73	2.31	-
39		3	7	-	1.70	4.12	-
40		4	0	-	1.66	0.00	-

The Holt-Winters method is used to predict unknown Apache vulnerabilities that may appear in 2018. The prediction for the first quarter of 2008 is calculated using Equation 6.2.

$$\begin{aligned} F_1 &= (L_0 + T_0) \times S_1 \\ &= (3.10 + (-0.04)) \times 0.93 \\ &= 2.87 \end{aligned} \tag{6.2}$$

Now that a value is predicted for the first quarter, L_t, T_t, and S_t are updated to predict vulnerabilities for 2018. Three smoothing parameters α (Equation 6.3), β (Equation 6.4), and γ (Equation 6.5) are used. α value 0.00 is used to calculate L_t values. T_t values are calculated using β value 0.05. γ value 0.00 is used to calculate remaining S_t (S_5 to S_{40}).

$$
\begin{aligned}
L_1 &= \alpha \times \frac{D_1}{S_1} + (1-\alpha) \times (L_0 + T_0) \\
&= 0.00 \times \frac{8}{0.93} + (1-0.00) \times (3.10 - 0.04) \\
&= 3.07
\end{aligned}
\tag{6.3}
$$

$$
\begin{aligned}
T_1 &= \beta \times (L_1 - L_0) + (1-\beta) \times T_0 \\
&= 0.05 \times (3.07 - 3.10) + (1-0.05) \times (-0.04) \\
&= -0.04
\end{aligned}
\tag{6.4}
$$

$$
\begin{aligned}
S_5 &= \gamma \times \frac{D_1}{L_1} + (1-\gamma) \times (S_1) \\
&= 0.00 \times \frac{8}{3.07} + (1-0.00) \times (0.93) \\
&= 0.93
\end{aligned}
\tag{6.5}
$$

After calculating L_t, T_t, and S_t values for all 40 periods, the vulnerabilities are predicted for 2018 (Table 6.4).

$$
\begin{aligned}
F_{41} &= [L_{40} + (T_{40} \times 1)] \times S_{37} \\
&= [1.66 + ((-0.04) \times 1)] \times 0.93 \\
&= 1.52
\end{aligned}
\tag{6.6}
$$

$$
\begin{aligned}
F_{42} &= [L_{40} + (T_{40} \times 2)] \times S_{38} \\
&= [1.66 + ((-0.04) \times 2)] \times 0.88 \\
&= 1.40
\end{aligned}
\tag{6.7}
$$

$$
\begin{aligned}
F_{43} &= [L_{40} + (T_{40} \times 3)] \times S_{39} \\
&= [1.66 + ((-0.04) \times 3)] \times 1.87 \\
&= 2.91
\end{aligned}
\tag{6.8}
$$

$$
\begin{aligned}
F_{44} &= \left[L_{40} + (T_{40} \times 4)\right] \times S_{40} \\
&= \left[1.66 + \left((-0.04) \times 4\right)\right] \times 0.55 \\
&= 0.83
\end{aligned}
\tag{6.9}
$$

Table 6.4: Prediction of Unknown Apache Vulnerabilities

Time(t)	D_t	$\tilde{S}_t$	S_t Values	L_t Values	T_t Values	Prediction
0	-	-	**3.10**	**-0.04**	-	-
1	8	2.61	0.93	3.07	-0.04	2.87
2	2	0.66	0.88	3.03	-0.04	2.67
3	2	0.67	1.87	2.99	-0.04	5.61
4	0	0.00	0.55	2.96	-0.04	1.62
5	1	0.34	0.93	2.92	-0.04	2.73
6	3	1.04	0.88	2.89	-0.04	2.54
7	5	1.75	1.87	2.85	-0.04	5.34
8	2	0.71	0.55	2.81	-0.04	1.54
9	6	2.16	0.93	2.78	-0.04	2.60
10	3	1.09	0.88	2.74	-0.04	2.42
11	2	0.74	1.87	2.71	-0.04	5.07
12	0	0.00	0.55	2.67	-0.04	1.46
13	1	0.38	0.93	2.63	-0.04	2.46
14	4	1.54	0.88	2.60	-0.04	2.29
15	3	1.17	1.87	2.56	-0.04	4.80
16	6	2.37	0.55	2.53	-0.04	1.38
17	4	1.61	0.93	2.49	-0.04	2.33
18	1	0.41	0.88	2.45	-0.04	2.16
19	5	2.07	1.87	2.42	-0.04	4.53
20	1	0.42	0.55	2.38	-0.04	1.30
21	2	0.85	0.93	2.35	-0.04	2.19
22	3	1.30	0.88	2.31	-0.04	2.03
23	4	1.76	1.87	2.27	-0.04	4.26
24	1	0.45	0.55	2.24	-0.04	1.22
25	2	0.91	0.93	2.20	-0.04	2.06
26	1	0.46	0.88	2.17	-0.04	1.91
27	6	2.82	1.87	2.13	-0.04	3.99
28	2	0.95	0.55	2.09	-0.04	1.14
29	1	0.49	0.93	2.06	-0.04	1.92

continued ...

Prediction of Unknown Apache Vulnerabilities ...continued

Time(t)	D_t	$\bar{S}_t$	S_t Values	L_t Values	T_t Values	Prediction
30	0	0.00	0.88	2.02	-0.04	1.78
31	4	2.01	1.87	1.99	-0.04	3.72
32	0	0.00	0.55	1.95	-0.04	1.06
33	0	0.00	0.93	1.91	-0.04	1.79
34	0	0.00	0.88	1.88	-0.04	1.65
35	3	1.63	1.87	1.84	-0.04	3.45
36	1	0.55	0.55	1.81	-0.04	0.99
37	0	0.00	0.93	1.77	-0.04	1.65
38	4	2.31	0.88	1.73	-0.04	1.53
39	7	4.12	1.87	1.70	-0.04	3.18
40	0	0.00	0.55	1.66	-0.04	0.91
41	Prediction	-	-	-	-	**1.52**
42	Prediction	-	-	-	-	**1.40**
43	Prediction	-	-	-	-	**2.91**
44	Prediction	-	-	-	-	**0.83**

Vulnerability Prediction Accuracy

Table 6.5 provides a summary of prediction error and TS calculated for all *t* periods.

Table 6.5: Prediction Error Tracking Signal of Apache

Time(t)	**Vuls**	**Predictions**	**Error** E_t	**Bias**	A_t = |E_t|	**MAD**	**TS**
1	8	2.87	5.13	5.13	5.13	5.13	1.00
2	2	2.67	-0.67	4.46	5.80	2.90	1.54
3	2	5.61	-3.61	0.85	9.41	3.14	0.27
4	0	1.62	-1.62	-0.77	11.03	2.76	-0.28
5	1	2.73	-1.73	-2.50	12.76	2.55	-0.98
6	3	2.54	0.46	-2.04	13.22	2.20	-0.93
7	5	5.34	-0.34	-2.38	13.56	1.94	-1.23
8	2	1.54	0.46	-1.92	14.02	1.75	-1.10
9	6	2.6	3.40	1.48	17.42	1.94	0.76
10	3	2.42	0.58	2.06	18.00	1.80	1.14
11	2	5.07	-3.07	-1.01	21.07	1.92	-0.53
12	0	1.46	-1.46	-2.47	22.53	1.88	-1.32
13	1	2.46	-1.46	-3.93	23.99	1.85	-2.13

continued ...

Prediction Error Tracking Signal of Apache ...continued

Time(t)	**Vuls**	**Predictions**	**Error** E_t	**Bias**	$A_t = \|E_t\|$	**MAD**	**TS**
14	4	2.29	1.71	-2.22	25.70	1.84	-1.21
15	3	4.8	-1.80	-4.02	27.50	1.83	-2.19
16	6	1.38	4.62	0.60	32.12	2.01	0.30
17	4	2.33	1.67	2.27	33.79	1.99	1.14
18	1	2.16	-1.16	1.11	34.95	1.94	0.57
19	5	4.53	0.47	1.58	35.42	1.86	0.85
20	1	1.3	-0.30	1.28	35.72	1.79	0.72
21	2	2.19	-0.19	1.09	35.91	1.71	0.64
22	3	2.03	0.97	2.06	36.88	1.68	1.23
23	4	4.26	-0.26	1.80	37.14	1.61	1.11
24	1	1.22	-0.22	1.58	37.36	1.56	1.01
25	2	2.06	-0.06	1.52	37.42	1.50	1.02
26	1	1.91	-0.91	0.61	38.33	1.47	0.41
27	6	3.99	2.01	2.62	40.34	1.49	1.75
28	2	1.14	0.86	3.48	41.20	1.47	2.37
29	1	1.92	-0.92	2.56	42.12	1.45	1.76
30	0	1.78	-1.78	0.78	43.90	1.46	0.53
31	4	3.72	0.28	1.06	44.18	1.43	0.74
32	0	1.06	-1.06	0.00	45.24	1.41	0.00
33	0	1.79	-1.79	-1.79	47.03	1.43	-1.26
34	0	1.65	-1.65	-3.44	48.68	1.43	-2.40
35	3	3.45	-0.45	-3.89	49.13	1.40	-2.77
36	1	0.99	0.01	-3.88	49.14	1.37	-2.84
37	0	1.65	-1.65	-5.53	50.79	1.37	-4.03
38	4	1.53	2.47	-3.06	53.26	1.40	-2.18
39	7	3.18	3.82	0.76	57.08	1.46	0.52
40	0	0.91	-0.91	-0.15	57.99	1.45	-0.10

The prediction errors are distributed with Mean of 0.00. A three σ spread from the Mean is equivalent to having a control chart with the UCL and LCL set at 5.88 and -5.88, respectively. Table 6.6 provides the Mean values, UCL, and LCL for all the quarters.

Table 6.6: Control Limits to Measure Accuracy of Apache Prediction

Time(t)	**Vuls**	**Predictions**	**Error** E_t	**MAD**	**TS**	**Mean**	**UCL**	**LCL**
1	8	2.87	5.13	5.13	1.00	0.00	5.80	-5.80
2	2	2.67	-0.67	2.90	1.54	0.00	5.80	-5.80
3	2	5.61	-3.61	3.14	0.27	0.00	5.80	-5.80
4	0	1.62	-1.62	2.76	-0.28	0.00	5.80	-5.80
5	1	2.73	-1.73	2.55	-0.98	0.00	5.80	-5.80
6	3	2.54	0.46	2.20	-0.93	0.00	5.80	-5.80
7	5	5.34	-0.34	1.94	-1.23	0.00	5.80	-5.80
8	2	1.54	0.46	1.75	-1.10	0.00	5.80	-5.80
9	6	2.6	3.40	1.94	0.76	0.00	5.80	-5.80
10	3	2.42	0.58	1.80	1.14	0.00	5.80	-5.80
11	2	5.07	-3.07	1.92	-0.53	0.00	5.80	-5.80
12	0	1.46	-1.46	1.88	-1.32	0.00	5.80	-5.80
13	1	2.46	-1.46	1.85	-2.13	0.00	5.80	-5.80
14	4	2.29	1.71	1.84	-1.21	0.00	5.80	-5.80
15	3	4.8	-1.80	1.83	-2.19	0.00	5.80	-5.80
16	6	1.38	4.62	2.01	0.30	0.00	5.80	-5.80
17	4	2.33	1.67	1.99	1.14	0.00	5.80	-5.80
18	1	2.16	-1.16	1.94	0.57	0.00	5.80	-5.80
19	5	4.53	0.47	1.86	0.85	0.00	5.80	-5.80
20	1	1.3	-0.30	1.79	0.72	0.00	5.80	-5.80
21	2	2.19	-0.19	1.71	0.64	0.00	5.80	-5.80
22	3	2.03	0.97	1.68	1.23	0.00	5.80	-5.80
23	4	4.26	-0.26	1.61	1.11	0.00	5.80	-5.80
24	1	1.22	-0.22	1.56	1.01	0.00	5.80	-5.80
25	2	2.06	-0.06	1.50	1.02	0.00	5.80	-5.80
26	1	1.91	-0.91	1.47	0.41	0.00	5.80	-5.80
27	6	3.99	2.01	1.49	1.75	0.00	5.80	-5.80
28	2	1.14	0.86	1.47	2.37	0.00	5.80	-5.80
29	1	1.92	-0.92	1.45	1.76	0.00	5.80	-5.80
30	0	1.78	-1.78	1.46	0.53	0.00	5.80	-5.80
31	4	3.72	0.28	1.43	0.74	0.00	5.80	-5.80
32	0	1.06	-1.06	1.41	0.00	0.00	5.80	-5.80
33	0	1.79	-1.79	1.43	-1.26	0.00	5.80	-5.80
34	0	1.65	-1.65	1.43	-2.40	0.00	5.80	-5.80
35	3	3.45	-0.45	1.40	-2.77	0.00	5.80	-5.80
36	1	0.99	0.01	1.37	-2.84	0.00	5.80	-5.80
37	0	1.65	-1.65	1.37	-4.03	0.00	5.80	-5.80
38	4	1.53	2.47	1.40	-2.18	0.00	5.80	-5.80

continued ...

Control Limits to Measure Accuracy of Apache Prediction ...continued

Time(t)	Vuls	Predictions	Error E_t	MAD	TS	Mean	UCL	LCL
39	7	3.18	3.82	1.46	0.52	0.00	5.80	-5.80
40	0	0.91	-0.91	1.45	-0.10	0.00	5.80	-5.80

Figure 6.3 shows that TS is well within control limits. Therefore, the prediction model does not exceed threshold and is accurate.

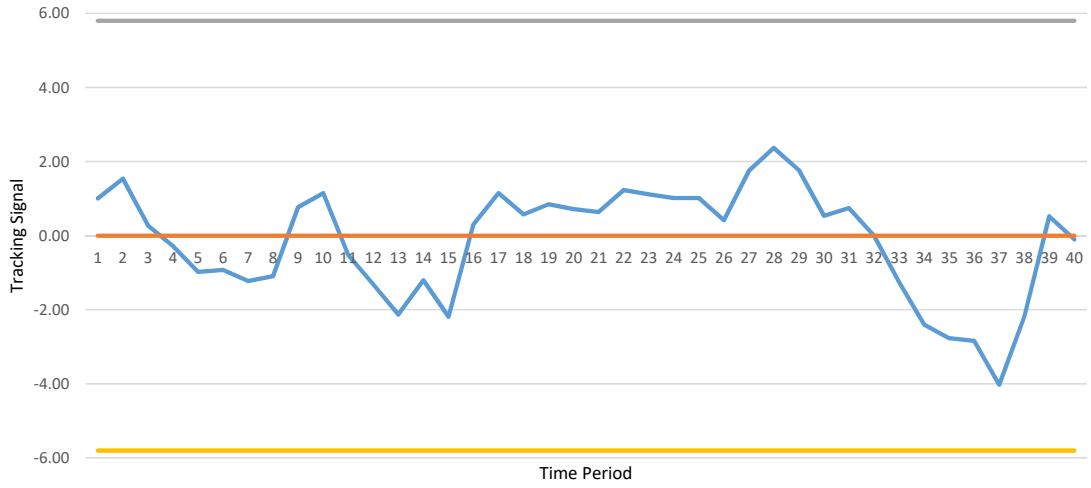


Figure 6.3: Accuracy of Apache Vulnerability Prediction

6.2.2 Prediction of Unknown Apache Vulnerabilities with regard to the Impact Levels

In this section, the impact ratings of unknown Apache vulnerabilities are predicted. The reported Apache vulnerabilities using Local and Remote Network AVs are considered as independent variables X_1 and X_2, respectively. High, Medium, and Low impact ratings of reported vulnerabilities are used as dependent variables Y_1, Y_2, and Y_3, respectively. The predicted impact ratings of unknown vulnerabilities are calculated as $\bar{Y_1}$, $\bar{Y_2}$, and $\bar{Y_3}$ for High, Medium, and Low impact ratings, respectively.

Prediction of High Impact Unknown Apache Vulnerabilities

Correlation coefficients are calculated using X_1, X_2, and Y_1 (High Impact). The P-values of both the independent variables are calculated. However, it was observed that P-value of Local AV is higher than 0.15. Therefore, it was not considered for the prediction. So, to make the prediction using Equation 6.10, 9 vulnerabilities would appear exploiting the remote network during 2018. The prediction results show that out of those 9 vulnerabilities, at least 1.48 vulnerabilities will be of High Impact. This result is an excellent prediction as it is almost equal to the per year mean value of High Impact reported vulnerabilities of 1.50.

$$\begin{aligned} \bar{Y_1} &= a + b_2(X_2) \\ &= 0.04 + 0.16(9) \\ &= 1.48 \end{aligned} \tag{6.10}$$

Prediction of Medium Impact Unknown Vulnerabilities

Correlation coefficients X_1, X_2, and Y_2 (Medium impact) are used to predict Medium Impact unknown vulnerabilities. Both independent variables are considered for the prediction as P-values are less than 0.15. So, to make the prediction using Equation 6.11 during 2018, 1 vulnerability would appear to exploit Local AV and 9 vulnerabilities exploit Remote Network AV. The prediction shows that out of those 10 vulnerabilities, at least 7.29 vulnerabilities will be of Medium Impact. This result is an excellent prediction as it is very close to the per year Mean value of Medium impact reported vulnerabilities, 7.40.

$$\begin{aligned} \bar{Y_2} &= a + b_1(X_1) + b_2(X_2) \\ &= 0.10 + 0.71(1) + 0.72(9) \\ &= 7.29 \end{aligned} \tag{6.11}$$

Prediction of Low Impact Unknown Vulnerabilities

Correlation coefficients X_1, X_2, and Y_3 (Low Impact) are used to predict Low Impact unknown vulnerabilities. Both independent variables are considered for the prediction as P-values are less than 0.15. So, to make the prediction using Equation 6.12 during 2018, 1 vulnerability would appear to exploit Local AV and 9 vulnerabilities exploit Remote Network AV. The prediction result shows that out of those 10 vulnerabilities, at least 1.71 vulnerabilities will be of Low Impact. This result is an average prediction as it is relatively close to the per year Mean value of Low Impact reported vulnerabilities, 0.90.

$$\begin{aligned} \bar{Y_3} &= a + b_1(X_1) + b_2(X_2) \\ &= -0.14 + 0.49(1) + 0.12(9) \\ &= 1.71 \end{aligned} \tag{6.12}$$

6.3 Risk Assessment of Apache HTTP Server

This section covers risk assessment of Apache. Vulnerability impact ratings, threat likelihood levels, and severity levels of risk to Apache are determined.

6.3.1 Determination of Apache Vulnerability Impact Ratings

An exploitation scenario for each Physical, Local, and Network vulnerability is presented. CVSS (Section 3.4.3) is used to score example scenarios to determine impact ratings. Table 6.7 provides Apache risk, vulnerabilities, and the AVs.

Table 6.7: Apache Risk, Vulnerabilities, and AVs

Risk No.	Risk Type	Apache Vulnerabilities	Attack Vector
R1	DoS	Physical Security Vulnerability (V1)	Physical
R2	Information Leakage	Discovered Web Directories (V2) HTTP Server Version Information (V3)	Local
R3	Unauthorised Access	Bypass Access Restriction (V4) Configuration Vulnerabilities (V5) System Settings Vulnerabilities (V6) Etag Header Information (V7) Discovered HTTP Methods (V8)	Network

Example Scenario 1

Figure 6.4 provides the impact rating of a vulnerability exploited by a PU TA who has physical access to the Apache web server. A PU with malicious intent can exploit a physical vulnerability by misusing physical access, for example, shutdown the server by unplugging the power or network cable. He can also steal the customers' data by copying it onto a removable drive and later use it for financial benefits. Such an exploitation scenario is categorised as:

- Attack vector is PHYSICAL
- Attack complexity is low
- Physical access privileges are required by the attacker to exploit vulnerability
- End user's interaction is not required to realise the exploitation
- Scope would change (exploitation would affect other components)
- Vulnerability exploitation would impact the confidentiality and availability security objectives

6.8 is the base score calculated, and the vulnerability impact rating is labelled as Medium.

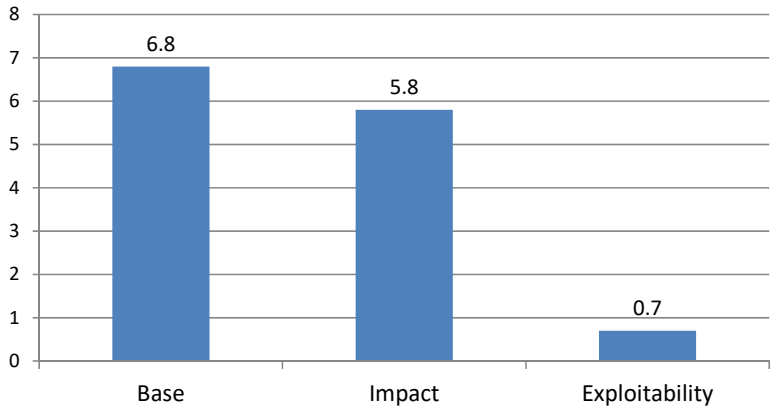


Figure 6.4: Apache Vulnerability Exploited through Physical AV

Example Scenario 2

The impact rating of a Local vulnerability exploitation scenario is determined here. An NU TA can misuse local access provided to manage website directories on the Apache web server. An NU can exploit a vulnerability, for example, a directory browsing feature is disabled, and Apache is running with a default root user. The attacker can use Malware or crafted attacks to gain more privileges and later use these privileges to access the root, Bin, and Conf directories and manipulate the sensitive server data. Such an exploitation scenario is categorised as:

- Attack vector is LOCAL
- Attack complexity is high
- Low access privileges are required by the attacker to exploit vulnerability
- End user's interaction is not required to realise the exploitation
- Scope would not change (exploitation would not affect other components)
- Vulnerability exploitation would impact the confidentiality and integrity security objectives

Figure 6.5 provides the impact rating of a local vulnerability exploited by a malicious NU. 6.3 is the base score calculated, and the vulnerability impact rating is labelled as

Medium.

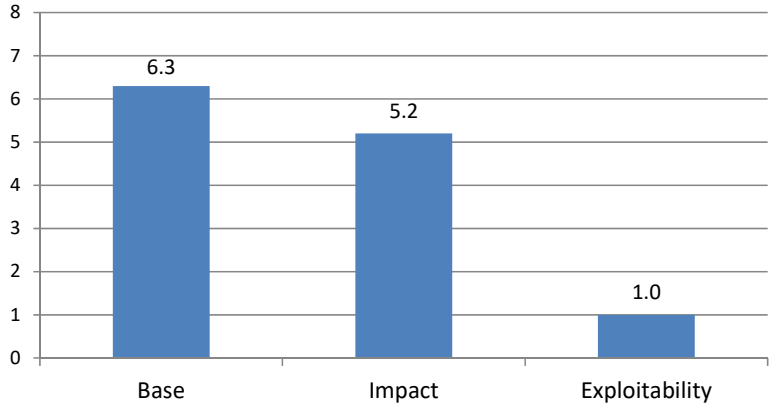


Figure 6.5: Apache Vulnerability Exploited through Local AV

Example Scenario 3

An SC with malicious intent can exploit a vulnerability that exists due to an insecure host server configuration such as unnecessary open ports. The attacker can use these type of flaws in the server configuration to gain unauthorised access to compromise the host system on which Apache web server is running and later, use this access to leverage further attacks. Such an exploitation scenario is categorised as:

- Attack vector is NETWORK
- Attack complexity is high
- High access privileges are required by the attacker to exploit vulnerability
- End user's interaction is not required to realise the exploitation
- Scope would change (exploitation would affect other components)
- Vulnerability exploitation would impact the confidentiality, integrity and availability security objectives

Figure 6.6 provides the impact rating of a network vulnerability exploited by an SC by accessing Apache services through the internet. 8.5 is the base score calculated, and the vulnerability impact rating is labelled as High.

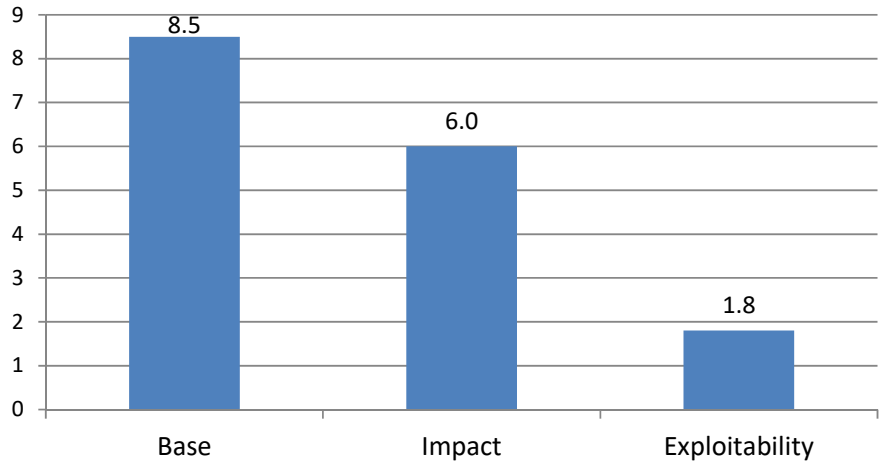


Figure 6.6: Apache Vulnerability Exploited through Network AV

AVs are considered to assign impact ratings to a set of vulnerabilities that leverage to same risk type. For example, R2 (Information Leakage) has two corresponding vulnerabilities (V2 and V3) which can be exploited through Local AV. Therefore, a Medium impact level (Figure 6.5) is assigned to this set of vulnerabilities belonging to same risk. Table 6.8 provides a summary of Apache risk, vulnerability type, corresponding AV, and the impact ratings. There can be more Apache risk and corresponding vulnerabilities. However, only three risk type are considered here to limit the scope of the assessment process.

Table 6.8: Summary of Apache Risk, Vulnerabilities, AVs, and Impact Ratings

Risk No.	**Risk Type**	**Apache Vulnerabilities**	**Attack Vector**	**Impact Rating**
R1	DoS	Physical Security Vulnerability (V1)	Physical	Medium
R2	Information Leakage	Discovered Web Directories (V2) HTTP Server Version Information (V3)	Local	Medium
R3	Unauthorised Access	Bypass Access Restriction (V4) Configuration Vulnerabilities (V5) System Settings Vulnerabilities (V6) Etag Header Information (V7) Discovered HTTP Methods (V8)	Network	High

6.3.2 Threat Likelihood Assessment of Apache HTTP Server

In this section, a threat likelihood assessment of Apache is performed. Figure 6.7 provides an Apache attack tree. In this case, a PU is a web administrator who manages the Apache server. An NU is a customer who uses the Apache web server to host website(s). An SC is an unregistered user who can access Apache web services through the internet, as provided in Figure 6.7

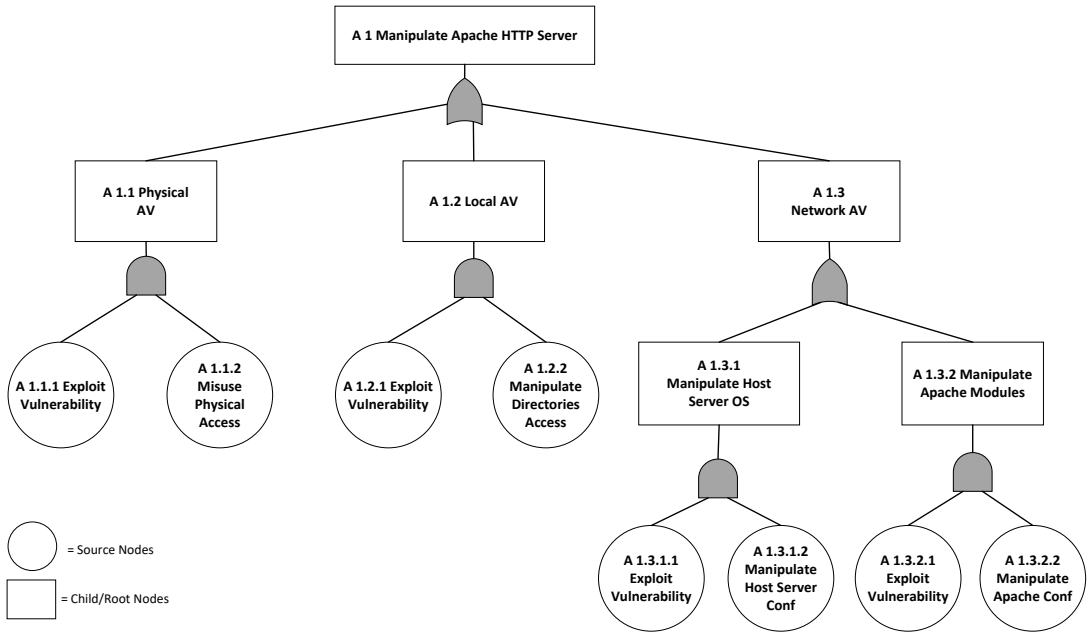


Figure 6.7: Apache Attack Tree

Assigning Threat Levels to Source Nodes of Apache Tree

In this section, threat levels are assigned to the source nodes of the Apache attack tree from PU, NU, and SC TAs.

A 1.1.1 and A 1.1.2 The source nodes connected with A 1.1 (Physical Attack) are considered. For attack step A 1.1.1 (Exploit Physical Vulnerability) minimum technical knowledge is required to exploit a physical vulnerability using physical access, for example, unplugging the power or network cable. Therefore, all three

TAs are capable enough to exploit physical vulnerability.

To assign a threat level to A 1.1.2 (Physical Access) source node, it is assumed that Apache Web server is provided with an acceptable physical access protection, such as locked data center, video surveillance and scanning system. However, the web administrator may have direct or physical access to manage the Apache web server. So, if the web administrator is influenced by a malicious party to exploit the server, then he can either damage the server physically or steal the data from it by using a removable disk. The PU is the only TA who is physically present on the premises where the host server is installed and has the privileges to access it. Therefore, both the capability and motivation (if influenced) levels are considered as high for the PU TA. Table 6.9 provides the threat levels assigned to A 1.1.1 and A 1.1.2.

Table 6.9: Threat Levels to A 1.1.1 and A 1.1.2

TA	**Capability**	**Motivation**	**TL to A 1.1.1**	**Capability**	**Motivation**	**TA to A 1.1.2**
PU	4	2	3	5	4	5
NU	4	2	3	1	4	2
SC	4	2	3	1	3	1

A 1.2.1 and A 1.2.2 A NU is an authorised user of the Apache web server who has access to manage his website directories provided on the Apache server. However, a NU user with malicious intent can exploit a vulnerability such as a directory browsing feature to gain more privileges. Therefore, an NU TA is more capable compared to other two, to exploit any such vulnerability through attack step A 1.2.1.

An NU who is capable of exploiting a local vulnerability, if motivated by an

internal or external malicious party, may pose a serious threat to the Apache server. An NU can use different methods such as Malwares and crafted attacks to exploit weak access controls to gain more privileges on the Apache server and later, use these high level privileges to exploit a vulnerability to manipulate data directories such as Root, Conf and Bin through attack step 1.2.2. Table 6.10 provides the threat levels assigned to A 1.2.1 and A 1.2.2 attack steps.

Table 6.10: Threat Levels to A 1.2.1 and A 1.2.2

TA	**Capability**	**Motivation**	**TL to A 1.2.1**	**Capability**	**Motivation**	**TA to A 1.2.2**
PU	3	2	2	2	3	2
NU	4	3	4	4	5	5
SC	2	2	1	1	3	1

A 1.3.1.1 and A 1.3.1.2 In the A 1.3.1.1 attack step, an SC can exploit vulnerabilities such as unnecessary open ports to compromise the host server. These vulnerabilities may exist when server hardening techniques are not followed. An SC is assumed to be an expert attacker who is capable enough to exploit Apache by using advance hacking tools and techniques. In the A 1.3.1.2 attack step, a motivated SC can exploit host vulnerabilities that exist due to an insecure host server configuration. An SC attacker can use brute force attack to retrieve the host server password due to weak security controls. Once the attacker gains the server's credentials, the host server can be compromised. Table 6.11 provides the threat levels assigned to A 1.3.1.1 and A 1.3.1.2 attack steps.

Table 6.11: Threat Levels to A 1.3.1.1 and A 1.3.1.2

TA	Capability	Motivation	TL to A 1.3.1.1	Capability	Motivation	TA to A 1.3.1.2
PU	2	1	1	2	2	1
NU	3	2	2	3	3	3
SC	4	3	4	4	5	5

A 1.3.2.1 and A 1.3.2.2 In the A 1.3.2.1 attack step, an SC (remote attacker) can exploit known vulnerabilities that exist in the optional Apache modules, for example, if the administrator has installed optional Apache modules and did not change the default settings. In the A 1.3.2.2 attack step, a malicious SC can exploit vulnerabilities that exist due to an improper Apache configuration. For example, the SSL Apache module is not enabled, and the CGI module is not disabled. The *Mod_Security* (Web Application Firewall) module is also not enabled by the web administrator against attacks such as SQL Injection, Session Hijacking, Cross Site Scripting, and Malware. Table 6.12 provides the threat levels assigned to A 1.3.2.1 and A 1.3.2.2 attack steps.

Table 6.12: Threat Levels to A 1.3.2.1 and A 1.3.2.2

TA	Capability	Motivation	TL to A 1.3.2.1	Capability	Motivation	TA to A 1.3.2.2
PU	2	2	1	2	3	2
NU	2	2	1	3	3	3
SC	5	3	5	5	4	5

Propagating Threat Levels Through to the Apache Attack Tree

In this section, threat levels are propagated to the child nodes. First of all, threat levels are propagated to A 1.3.1 and A 1.3.2 attack steps.

A 1.3.1 Manipulate Apache Host Server OS As the source nodes to A 1.3.1 are connected with a logical *AND* operator, the minimal threat levels of source nodes are propagated to A 1.3.1. Table 6.13 provides the threat levels to A 1.3.1 attack step.

Table 6.13: Manipulate Apache Host Server OS

Type:AND	**TL for A 1.3.1.1**	**TL for A 1.3.1.2**	**TL for A 1.3.1**
PU	1	1	1
NU	2	3	2
IC	4	5	4

A 1.3.2 Manipulate Apache Modules Source nodes to A 1.3.2 are connected with a logical *AND* operator. Therefore, the minimal levels of threat of source nodes are propagated to A 1.3.2. Table 6.14 provides the threat levels to A 1.3.2 attack step.

Table 6.14: Manipulate Apache Modules

Type:*AND*	**Likelihood for A 1.3.2.1**	**Likelihood for A 1.3.2.2**	**Likelihood for A 1.3.2**
PU	1	2	1
NU	1	3	1
IC	5	5	5

Now, threat levels are propagated to child nodes A 1.1, A 1.2, and A 1.3 to determine threat likelihood levels to AVs. Threat levels are also propagated to the A 1 root node to

determine overall threat likelihood level to Apache.

A 1.1 Apache Physical Attack The source nodes are connected to A 1.1 with a logical *AND* operator. Therefore, the minimal threat levels of the two source nodes are propagated to A 1.1.

Table 6.15 provides the threat levels to the A 1.1 attack step.

Table 6.15: Threat Likelihood Level at Physical AV

Type:*AND*	**Likelihood for A 1.1.1**	**Likelihood for A 1.1.2**	**Likelihood for A 1.1**
PU	3	4	3
NU	3	2	2
SC	3	1	1

A 1.2 Apache Local Attack As the source nodes are connected to A 1.2 node with a logical *AND* operator, the minimal threat levels of the two source nodes are propagated to A 1.2.

Table 6.16 provides the threat levels to A 1.2.

Table 6.16: Threat Likelihood Level at Local AV

Type:*AND*	**Likelihood for A 1.2.1**	**Likelihood for A 1.2.2**	**Likelihood for A 1.2**
PU	2	2	2
NU	4	5	4
SC	1	1	1

A 1.3 Apache Remote Network Attack As the child nodes are connected to A 1.3

with a logical *OR*, the maximal threat levels of the two source nodes are propagated to A 1.3.

Table 6.17 provides the threat levels to the A 1.3 attack step.

Table 6.17: Threat Likelihood Level at Network AV

Type:*OR*	**Likelihood for A 1.3.1**	**Likelihood for A 1.3.2**	**Likelihood for A 1.3**
PU	1	1	1
NU	2	1	2
SC	4	5	5

A 1 Manipulate Apache HTTP Server All three child nodes are connected to root node A 1 with a logical *OR*. Therefore, the maximal levels of threat of these child nodes are propagated to A 1. Table 6.18 provides the threat levels to the A 1 attack step by propagating threat levels from Tables 6.15, 6.16, and 6.17.

Table 6.18: Threat Likelihood Levels to Apache

Type:*OR*	**Likelihood for A 1.1 (Physical AV)**	**Likelihood for A 1.2 (Local AV)**	**Likelihood for A 1.3 (Network AV)**	**Likelihood for A 1 (Overall)**
PU	3	2	1	**3**
NU	2	4	2	**4**
SC	1	1	5	**5**

Table 6.19 provides qualitative threat likelihood levels from PU, NU, and SC.

Table 6.19: Qualitative Threat Likelihood Levels to Apache

Risk No.	Attack Vector	TL from PU	TL from NU	TL from SC
R1	Physical	Moderate	Low	Negligible
R2	Local	Low	Substantial	Negligible
R3	Network	Negligible	Low	Severe

Threat levels are assigned to the Apache attack tree to determine threat likelihood levels from PU, NU, and SC. Figures 6.8, 6.9, and 6.10 provide the threat likelihood levels to the Apache HTTP server from PU, NU, and SC, respectively. The threat levels are critical-6 (dark red), severe-5 (red), substantial-4 (Orange), moderate-3 (dark yellow), low-2 (yellow) and negligible-1 (green).

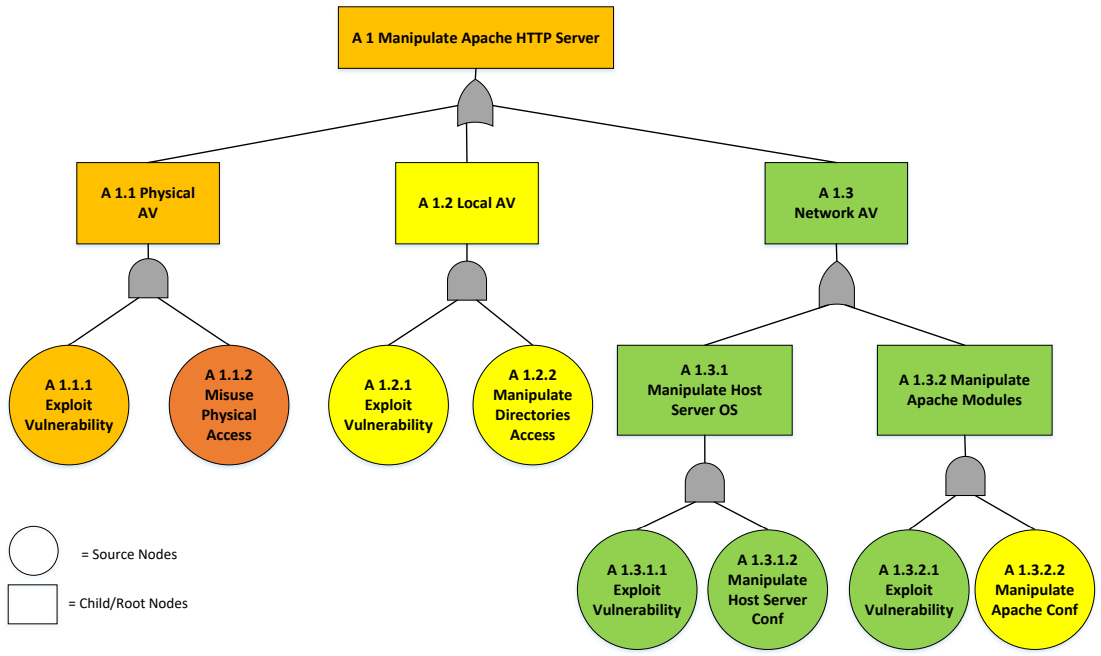


Figure 6.8: Apache Attack Tree for PU

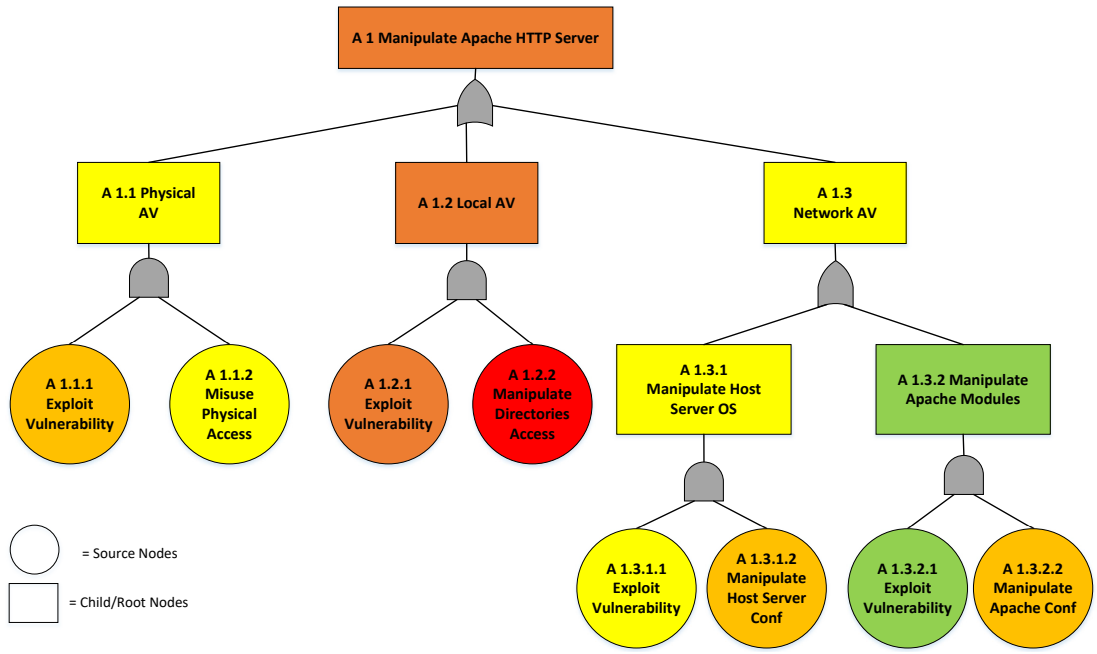


Figure 6.9: Apache Attack Tree for NU

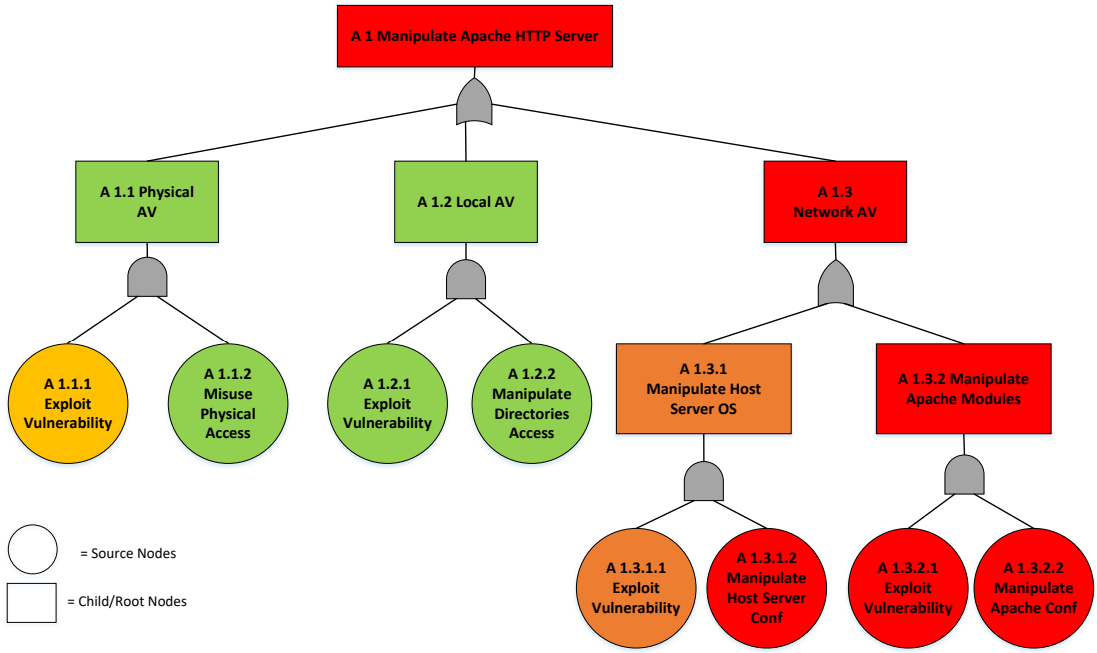


Figure 6.10: Apache Attack Tree for SC

A PU is likely going to exploit a physical vulnerability and poses a Moderate likelihood level. Low, and Negligible threat likelihood levels are determined when a PU

is likely to exploit Local and Network AVs, respectively.

An NU poses a Substantial threat likelihood level to exploit a vulnerability through Local AV. However, Low threat likelihood levels are observed to Physical and Network AVs.

An SC poses a Severe threat likelihood level to exploit a vulnerability through Network AV. However, Negligible threat likelihood levels are observed to Physical and Local AVs.

6.3.3 Severity Levels of Risk to Apache

In this section, severity levels of risk to Apache are determined by combining vulnerability impact ratings and threat likelihood levels. Medium risk severity levels are determined for R1, R2, and R3 from both PU and NU. An SC poses High risk severity level for R3 by exploiting the Network AV. However, Low risk severity levels for R1 and R2 are observed. Table 6.20 provides the risk severity levels to Apache HTTP server from PU, NU, and SC.

Conclusion

Apache is a software package with different architecture and functionality compared to Xen. The generalisability and applicability of the process is determined by applying it to Apache that is a different type of software from Xen. The evaluation results are near accurate and show that the research methods are not limited to performing only Xen vulnerability prediction and risk assessment. The same processes, data types, and methods are used to perform vulnerability prediction and risk assessment of Apache. To further ensure the applicability of the process, it is applied to another different and common software package (Squid Proxy).

Table 6.20: Severity Levels of Risk from PU, NU, and SC

Risk No.	**Attack Vector**	**Vul Impact Rating**	**TL from PU**	**Level of Risk from PU**	**TL from NU**	**Level of Risk from NU**	**TL from SC**	**Level of Risk from SC**
R1	Physical	Medium	Moderate	Medium	Low	Medium	Negligible	Low
R2	Local	Medium	Low	Medium	Substantial	Medium	Negligible	Low
R3	Network	High	Negligible	Medium	Low	Medium	Severe	High

6.4 Vulnerability Prediction of Squid Proxy Server

This section covers the vulnerability prediction process for the Squid server. The vulnerability prediction process is applied to Squid to predict unknown vulnerabilities for 2018. The objective of vulnerability prediction and risk assessment of Squid Proxy server is to highlight the vulnerabilities and risks to Squid to raise awareness to the organisations by identifying the risks with high severity levels which may be associated with Squid service.

6.4.1 Prediction of Unknown Squid Vulnerabilities

During the search for reported vulnerabilities in the Squid proxy server, 56 vulnerability records appeared from 2008 to 2017. Reported vulnerability data is distributed quarterly to predict unknown vulnerabilities for 2018. Table 6.21 provides the reported Squid vulnerabilities from 2008 to 2017.

Table 6.21: Reported Squid Vulnerabilities

Time(t)	Year	Quarter	Vulnerabilities
1	2008	1	2
2		2	3
3		3	0
4		4	0
5	2009	1	2
6		2	0
7		3	3
8		4	4
9	2010	1	2
10		2	0
11		3	1
12		4	1
13	2011	1	0
14		2	0
15		3	1
16		4	1
17	2012	1	0

continued . . .

Reported Squid Vulnerabilities ...continued

Time(t)	Year	Quarter	Vulnerabilities
18		2	1
19		3	2
20		4	1
21	2013	1	1
22		2	0
23		3	3
24		4	0
25	2014	1	0
26		2	1
27		3	2
28		4	2
29	2015	1	1
30		2	1
31		3	1
32		4	1
33	2016	1	4
34		2	11
35		3	1
36		4	0
37	2017	1	3
38		2	0
39		3	0
40		4	0

The Holt-Winters method is used to predict the unknown vulnerabilities for the first quarter of 2018.

$$\begin{aligned} F_1 &= (L_0 + T_0) \times S_1 \\ &= (0.73 + 0.03) \times 1.16 \\ &= 0.89 \end{aligned} \tag{6.13}$$

Now that a value is predicted for the first quarter, the L_t, T_t, and S_t values can be updated. Therefore, L_1 is updated with an assumption that $\alpha = 0.00$.

$$\begin{aligned} L_1 &= \alpha \times \frac{D_1}{S_1} + (1-\alpha) \times (L_0 + T_0) \\ &= 0.00 \times \frac{2}{1.16} + (1-0.00) \times (0.73 + 0.03) \\ &= 0.77 \end{aligned} \tag{6.14}$$

T_1 is also updated with an assumption that $\beta = 0.03$.

$$\begin{aligned} T_1 &= \beta \times (L_1 - L_0) + (1-\beta) \times T_0 \\ &= 0.03 \times (0.77 - 0.73) + (1-0.03) \times (0.03) \\ &= 0.03 \end{aligned} \tag{6.15}$$

After updating L_t and T_t using α and β, S_t values (S_5 to S_40) are updated with an assumption that γ = 0.00

$$\begin{aligned} S_5 &= \gamma \times \frac{D_1}{L_1} + (1-\gamma) \times (S_1) \\ &= 0.00 \times \frac{2}{0.77} + (1-0.00) \times (1.16) \\ &= 1.16 \end{aligned} \tag{6.16}$$

After calculating L_t, T_t and S_t values for all 40 periods, the vulnerabilities are predicted for 2018.

$$\begin{aligned} F_{41} &= [L_{40} + (T_{40} \times 1)] \times S_{37} \\ &= [2.12 + ((0.03) \times 1)] \times 1.16 \\ &= 2.50 \end{aligned} \tag{6.17}$$

$$\begin{aligned} F_{42} &= [L_{40} + (T_{40} \times 2)] \times S_{38} \\ &= [2.12 + ((0.03) \times 2)] \times 1.14 \\ &= 2.50 \end{aligned} \tag{6.18}$$

$$
\begin{aligned}
F_{43} &= [L_{40} + (T_{40} \times 3)] \times S_{39} \\
&= [2.12 + ((0.03) \times 3)] \times 1.04 \\
&= 2.32
\end{aligned}
\tag{6.19}
$$

$$
\begin{aligned}
F_{44} &= [L_{40} + (T_{40} \times 4)] \times S_{40} \\
&= [2.12 + ((0.03) \times 4)] \times 0.80 \\
&= 1.81
\end{aligned}
\tag{6.20}
$$

Table 6.22 provides all the L_t, T_t, S_t, and predicted values.

Table 6.22: Prediction of Unknown Squid Vulnerabilities

Time(t)	D_t	$\bar{S}_t$	S_t Values	L_t Values	T_t Values	Prediction
0	-	-	**0.73**	**0.03**	-	
1	2	2.61	1.16	0.77	0.03	0.89
2	3	3.74	1.14	0.80	0.03	0.91
3	0	0.00	1.04	0.84	0.03	0.87
4	0	0.00	0.80	0.87	0.03	0.70
5	2	2.21	1.16	0.91	0.03	1.05
6	0	0.00	1.14	0.94	0.03	1.07
7	3	3.07	1.04	0.98	0.03	1.02
8	4	3.96	0.80	1.01	0.03	0.81
9	2	1.91	1.16	1.05	0.03	1.21
10	0	0.00	1.14	1.08	0.03	1.23
11	1	0.90	1.04	1.11	0.03	1.16
12	1	0.87	0.80	1.15	0.03	0.92
13	0	0.00	1.16	1.18	0.03	1.37
14	0	0.00	1.14	1.22	0.03	1.39
15	1	0.80	1.04	1.25	0.03	1.31
16	1	0.78	0.80	1.29	0.03	1.03
17	0	0.00	1.16	1.32	0.03	1.53
18	1	0.74	1.14	1.36	0.03	1.55
19	2	1.44	1.04	1.39	0.03	1.45
20	1	0.70	0.80	1.43	0.03	1.14
21	1	0.68	1.16	1.46	0.03	1.70
22	0	0.00	1.14	1.50	0.03	1.71
23	3	1.96	1.04	1.53	0.03	1.60

continued ...

Prediction of Unknown Squid Vulnerabilities ...continued

Time(t)	D_t	$\bar{S}_t$	S_t Values	L_t Values	T_t Values	Prediction
24	0	0.00	0.80	1.57	0.03	1.26
25	0	0.00	1.16	1.60	0.03	1.86
26	1	0.61	1.14	1.64	0.03	1.86
27	2	1.20	1.04	1.67	0.03	1.74
28	2	1.17	0.80	1.70	0.03	1.37
29	1	0.57	1.16	1.74	0.03	2.02
30	1	0.56	1.14	1.77	0.03	2.02
31	1	0.55	1.04	1.81	0.03	1.89
32	1	0.54	0.80	1.84	0.03	1.48
33	4	2.13	1.16	1.88	0.03	2.18
34	11	5.75	1.14	1.91	0.03	2.18
35	1	0.51	1.04	1.95	0.03	2.03
36	0	0.00	0.80	1.98	0.03	1.59
37	3	1.49	1.16	2.02	0.03	2.34
38	0	0.00	1.14	2.05	0.03	2.34
39	0	0.00	1.04	2.09	0.03	2.18
40	0	0.00	0.80	2.12	0.03	1.70
41	Prediction	-	-	-	-	**2.50**
42	Prediction	-	-	-	-	**2.50**
43	Prediction	-	-	-	-	**2.32**
44	Prediction	-	-	-	-	**1.81**

Squid Vulnerability Prediction Accuracy

Table 6.23 provides a summary of prediction error and TS calculated for all *t* periods.

Table 6.23: MAD and Tracking Signal of Squid

Time(t)	**Vuls**	**Predictions**	**Error** E_t	**Bias**	A_t = $\lvert E_t \rvert$	**MAD**	**TS**
1	2	0.89	1.11	1.11	1.11	1.11	1.00
2	3	0.91	2.09	3.20	3.20	1.60	2.00
3	0	0.87	-0.87	2.33	4.07	1.36	1.72
4	0	0.70	-0.70	1.63	4.77	1.19	1.37
5	2	1.05	0.95	2.58	5.72	1.14	2.26
6	0	1.07	-1.07	1.51	6.79	1.13	1.33
7	3	1.02	1.98	3.49	8.77	1.25	2.79

continued ...

MAD and Tracking Signal of Squid ...continued

Time(t)	**Vuls**	**Predictions**	**Error** E_t	**Bias**	$A_t = \lvert E_t \rvert$	**MAD**	**TS**
8	4	0.81	3.19	6.68	11.96	1.50	4.47
9	2	1.21	0.79	7.47	12.75	1.42	5.27
10	0	1.23	-1.23	6.24	13.98	1.40	4.46
11	1	1.16	-0.16	6.08	14.14	1.29	4.73
12	1	0.92	0.08	6.16	14.22	1.19	5.20
13	0	1.37	-1.37	4.79	15.59	1.20	3.99
14	0	1.39	-1.39	3.40	16.98	1.21	2.80
15	1	1.31	-0.31	3.09	17.29	1.15	2.68
16	1	1.03	-0.03	3.06	17.32	1.08	2.83
17	0	1.53	-1.53	1.53	18.85	1.11	1.38
18	1	1.55	-0.55	0.98	19.40	1.08	0.91
19	2	1.45	0.55	1.53	19.95	1.05	1.46
20	1	1.14	-0.14	1.39	20.09	1.00	1.38
21	1	1.70	-0.70	0.69	20.79	0.99	0.70
22	0	1.71	-1.71	-1.02	22.50	1.02	-1.00
23	3	1.60	1.40	0.38	23.90	1.04	0.37
24	0	1.26	-1.26	-0.88	25.16	1.05	-0.84
25	0	1.86	-1.86	-2.74	27.02	1.08	-2.54
26	1	1.86	-0.86	-3.60	27.88	1.07	-3.36
27	2	1.74	0.26	-3.34	28.14	1.04	-3.20
28	2	1.37	0.63	-2.71	28.77	1.03	-2.64
29	1	2.02	-1.02	-3.73	29.79	1.03	-3.63
30	1	2.02	-1.02	-4.75	30.81	1.03	-4.63
31	1	1.89	-0.89	-5.64	31.70	1.02	-5.52
32	1	1.48	-0.48	-6.12	32.18	1.01	-6.09
33	4	2.18	1.82	-4.30	34.00	1.03	-4.17
34	11	2.18	8.82	4.52	42.82	1.26	3.59
35	1	2.03	-1.03	3.49	43.85	1.25	2.79
36	0	1.59	-1.59	1.90	45.44	1.26	1.51
37	3	2.34	0.66	2.56	46.10	1.25	2.05
38	0	2.34	-2.34	0.22	48.44	1.27	0.17
39	0	2.18	-2.18	-1.96	50.62	1.30	-1.51
40	0	1.70	-1.70	-3.66	52.32	1.31	-2.80

The prediction errors are distributed with a Mean of -0.09. A three σ spread from the Mean is equivalent to having a control chart with the UCL and LCL set at 5.69 and -5.87, respectively. Table 6.24 provides the Mean values, UCL, and LCL for all the

quarters.

Table 6.24: Control Limits to Measure Accuracy of Squid Prediction

Time(t)	**Vuls**	**Predictions**	**Error** E_t	**MAD**	**TS**	**Mean**	**UCL**	**LCL**
1	2	0.89	1.11	1.11	1.00	-0.09	5.69	-5.87
2	3	0.91	2.09	1.60	2.00	-0.09	5.69	-5.87
3	0	0.87	-0.87	1.36	1.72	-0.09	5.69	-5.87
4	0	0.70	-0.70	1.19	1.37	-0.09	5.69	-5.87
5	2	1.05	0.95	1.14	2.26	-0.09	5.69	-5.87
6	0	1.07	-1.07	1.13	1.33	-0.09	5.69	-5.87
7	3	1.02	1.98	1.25	2.79	-0.09	5.69	-5.87
8	4	0.81	3.19	1.50	4.47	-0.09	5.69	-5.87
9	2	1.21	0.79	1.42	5.27	-0.09	5.69	-5.87
10	0	1.23	-1.23	1.40	4.46	-0.09	5.69	-5.87
11	1	1.16	-0.16	1.29	4.73	-0.09	5.69	-5.87
12	1	0.92	0.08	1.19	5.20	-0.09	5.69	-5.87
13	0	1.37	-1.37	1.20	3.99	-0.09	5.69	-5.87
14	0	1.39	-1.39	1.21	2.80	-0.09	5.69	-5.87
15	1	1.31	-0.31	1.15	2.68	-0.09	5.69	-5.87
16	1	1.03	-0.03	1.08	2.83	-0.09	5.69	-5.87
17	0	1.53	-1.53	1.11	1.38	-0.09	5.69	-5.87
18	1	1.55	-0.55	1.08	0.91	-0.09	5.69	-5.87
19	2	1.45	0.55	1.05	1.46	-0.09	5.69	-5.87
20	1	1.14	-0.14	1.00	1.38	-0.09	5.69	-5.87
21	1	1.70	-0.70	0.99	0.70	-0.09	5.69	-5.87
22	0	1.71	-1.71	1.02	-1.00	-0.09	5.69	-5.87
23	3	1.60	1.40	1.04	0.37	-0.09	5.69	-5.87
24	0	1.26	-1.26	1.05	-0.84	-0.09	5.69	-5.87
25	0	1.86	-1.86	1.08	-2.54	-0.09	5.69	-5.87
26	1	1.86	-0.86	1.07	-3.36	-0.09	5.69	-5.87
27	2	1.74	0.26	1.04	-3.20	-0.09	5.69	-5.87
28	2	1.37	0.63	1.03	-2.64	-0.09	5.69	-5.87
29	1	2.02	-1.02	1.03	-3.63	-0.09	5.69	-5.87
30	1	2.02	-1.02	1.03	-4.63	-0.09	5.69	-5.87
31	1	1.89	-0.89	1.02	-5.52	-0.09	5.69	-5.87
32	1	1.48	-0.48	1.01	-6.09	-0.09	5.69	-5.87
33	4	2.18	1.82	1.03	-4.17	-0.09	5.69	-5.87
34	11	2.18	8.82	1.26	3.59	-0.09	5.69	-5.87
35	1	2.03	-1.03	1.25	2.79	-0.09	5.69	-5.87

continued . . .

Control Limits to Measure Accuracy of Squid Prediction ...continued

Time(t)	Vuls	Predictions	Error E_t	MAD	TS	Mean	UCL	LCL
36	0	1.59	-1.59	1.26	1.51	-0.09	5.69	-5.87
37	3	2.34	0.66	1.25	2.05	-0.09	5.69	-5.87
38	0	2.34	-2.34	1.27	0.17	-0.09	5.69	-5.87
39	0	2.18	-2.18	1.30	-1.51	-0.09	5.69	-5.87
40	0	1.70	-1.70	1.31	-2.80	-0.09	5.69	-5.87

Figure 6.11 shows that the TS is well within control limits except for period 32, where TS exceeded the threshold by 0.22 (from -5.87 to -6.09). However, the following predicted values are within control limits.

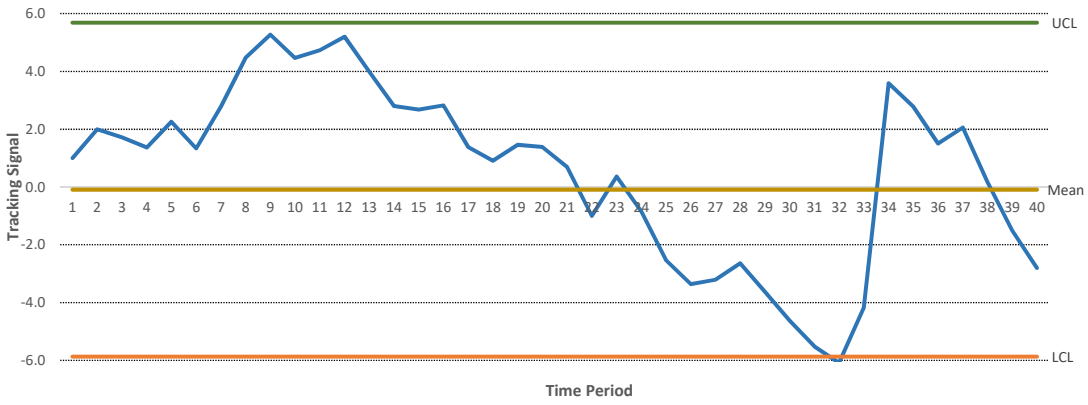


Figure 6.11: Tracking Squid Vulnerability Prediction Accuracy

6.4.2 Prediction of Unknown Squid Vulnerabilities with regard to the Impact Levels

In this section, unknown Squid vulnerabilities are predicted with regard to the impact levels. Reported Squid vulnerabilities using Remote Network AV are considered as the independent variable X_1. High, and Medium Impact ratings of reported vulnerabilities are used as dependent variables Y_1, and Y_2, respectively. The predicted unknown vulnerabilities are calculated as $\bar{Y_1}$, and $\bar{Y_2}$ for High, and Medium Impact ratings, respectively.

Prediction of High Impact Unknown Vulnerabilities

The prediction result shows that out of 6 vulnerabilities, 0.80 vulnerabilities will be of High Impact during 2018. This result is an excellent prediction as it is precisely the same as the per year Mean value (0.80) of High impact reported vulnerabilities.

$$\begin{aligned} \bar{Y_1} &= a + b_1(X_1) \\ &= 0.02 + 0.13(6) \\ &= 0.80 \end{aligned} \tag{6.21}$$

Prediction of Medium Impact Unknown Vulnerabilities

The prediction result shows that out of 6 vulnerabilities, at least 5 vulnerabilities will be of Medium Impact during 2018. This result is also an excellent prediction as it is very close to the per year Mean value of Medium Impact reported vulnerabilities, 4.80.

$$\begin{aligned} \bar{Y_2} &= a + b_1(X_1) \\ &= -0.02 + 0.87(6) \\ &= 5.24 \end{aligned} \tag{6.22}$$

6.5 Risk Assessment of the Squid Proxy Server

In this section, risk assessment of the Squid is performed.

6.5.1 Determination of Squid Vulnerability Impact Ratings

Physical, Local, and Network exploitation scenarios for the Squid proxy server are considered to score the vulnerabilities. Table 6.25 provides Squid risk, vulnerabilities, and the AVs.

Table 6.25: Squid Risk, Vulnerabilities, and AVs

Risk No.	Risk Type	Vulnerabilities	Attack Vector
R1	DoS	Physical Security Vulnerability (V1)	Physical
R2	Unauthorised access to application servers	Unauthorised access to the proxy server, and then attaining access to the internal application and database servers (V2)	Local
R3	Unauthorised URL access	Unauthorised access to incorrect URLs (V3)	Network
R4	Unauthorised URL redirection	Unauthorised access to the proxy server and adding and/or redirecting URLs to unintended URLs (V4)	Network

Example Scenario 1

Figure 6.12 provides the impact rating of a physical vulnerability exploited by a PU with physical access to a host server on which Squid proxy is running. 6.8 is the Base score calculated, and the vulnerability impact rating is labelled as Medium.

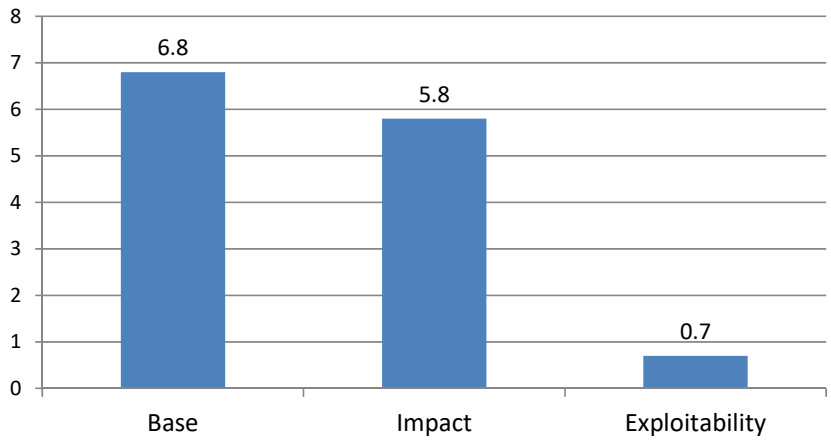


Figure 6.12: Squid Vulnerability Exploited through Physical AV

Example Scenario 2

Figure 6.13 provides the impact rating of a Local vulnerability exploited by an NU with malicious intent. 8.4 is the Base score calculated for this scenario, and the vulnerability

impact rating is labelled as High.

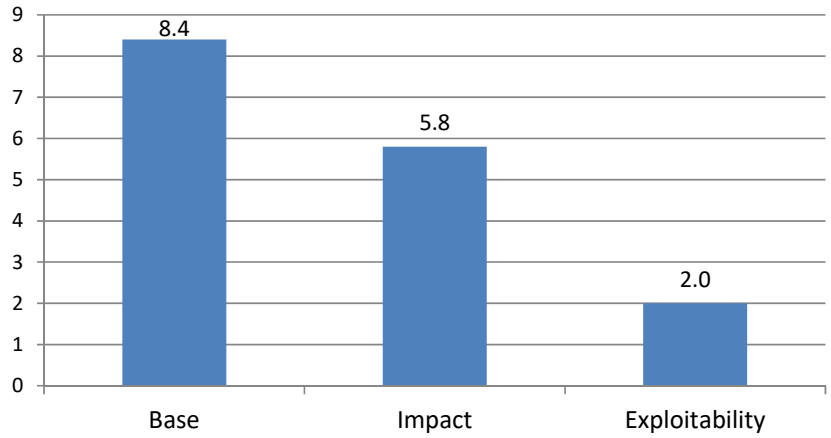


Figure 6.13: Squid Vulnerability Exploited through Local AV

Example Scenario 3

Figure 6.14 provides the impact rating of a Network vulnerability exploited by an IC attacker. 6.4 is the Base score calculated for this scenario, and the vulnerability impact rating is labelled as Medium.

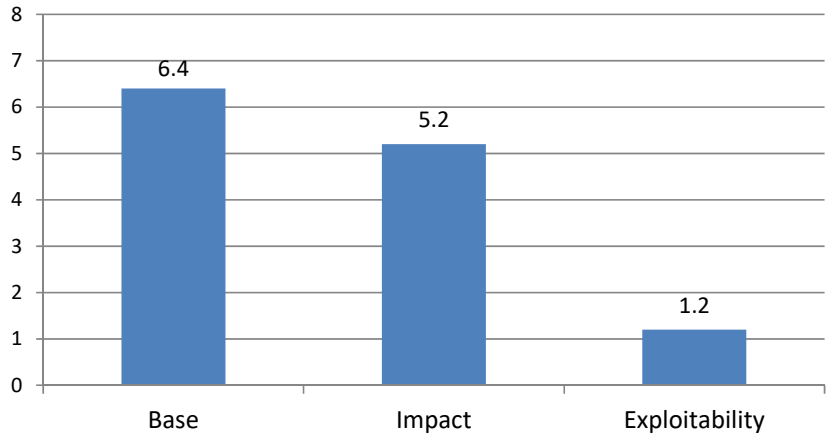


Figure 6.14: Squid Vulnerability Exploited through Network AV

The impact ratings are assigned to corresponding set of vulnerabilities and risk type. Table 6.26 provides the summary of Squid risk, vulnerability type, corresponding AV, and the impact ratings.

Table 6.26: Summary of Squid Risk, Vulnerabilities, AVs, and Impact Ratings

Risk No.	**Risk Type**	**Vulnerabilities**	**Attack Vector**	**Impact Rating**
R1	DoS	Physical Security Vulnerability (V1)	Physical	Medium
R2	Unauthorised access to application servers	Unauthorised access to the proxy server, and then attaining access to the internal application and database servers (V2)	Local	High
R3	Unauthorised URL access	Unauthorised access to incorrect URLs (V3)	Network	Medium
R4	Unauthorised URL redirection	Unauthorised access to the proxy server and adding and/or redirecting URLs to unintended URLs (V4)	Network	Medium

6.5.2 Threat Likelihood Assessment of Squid

In this section, threat likelihood assessment of the Squid proxy server is performed. A Squid attack tree is presented in Figure 6.15.

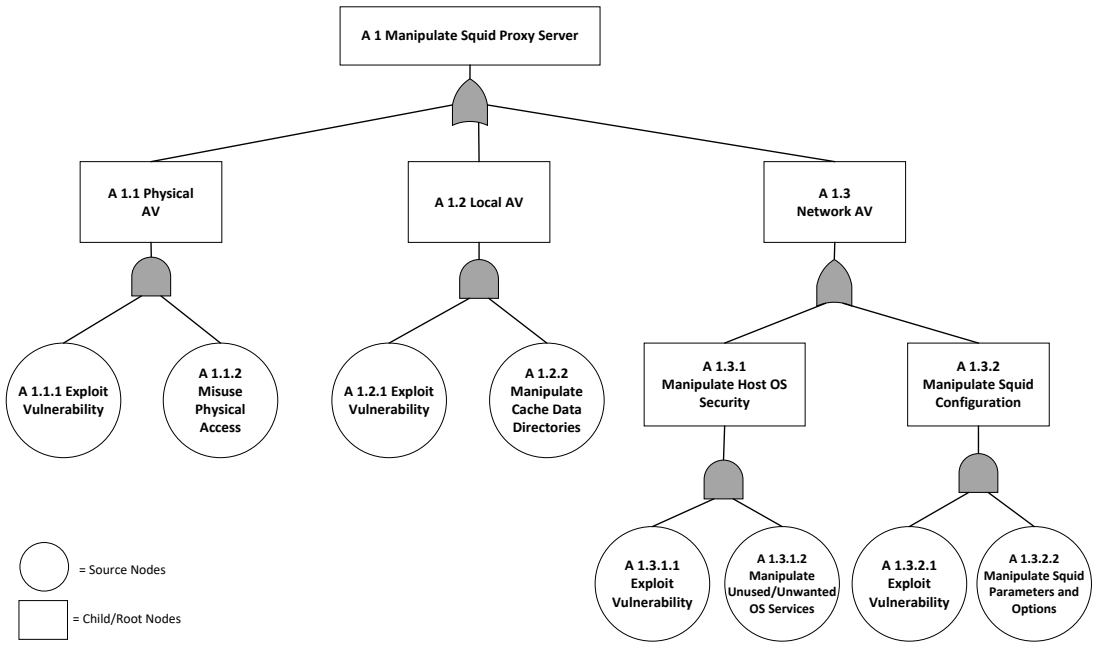


Figure 6.15: Squid Attack Tree

Assigning Threat Levels to Source Nodes

In this section, threat levels are assigned to the source nodes of Squid attack tree from PU, NU, and IC TAs. A PU is an administrator of Squid proxy server and an NU is an internal user who can assess Squid as a proxy from the same sub-network. Whereas, an IC is an illegitimate external user and is assumed to be an attacker for threat likelihood assessment.

A 1.1.1 and A 1.1.2 Table 6.27 provides the threat levels assigned to A 1.1.1 and A 1.1.2.

Table 6.27: Threat Levels to Squid A 1.1.1 and A 1.1.2

TA	Capability	Motivation	TL to A 1.1.1	Capability	Motivation	TA to A 1.1.2
PU	4	2	3	4	3	4
NU	4	1	2	2	3	2
IC	4	2	3	1	4	2

A 1.2.1 and A 1.2.2 An NU with malicious intent can use different tools to exploit local network vulnerability to sniff network traffic and retrieve server credentials. These credentials can then be used to manipulate data directories of other network users. Table 6.28 provides the threat levels assigned to A 1.2.1 and A 1.2.2.

Table 6.28: Threat Levels to Squid A 1.2.1 and A 1.2.2

TA	Capability	Motivation	TL to A 1.2.1	Capability	Motivation	TA to A 1.2.2
PU	3	1	2	5	1	3
NU	2	1	1	4	3	4
IC	2	3	2	1	3	1

A 1.3.1.1 and A 1.3.1.2 An IC (remote attacker) can manipulate the security of a host OS by exploiting a vulnerability that exists due to an insecure OS configuration. An expert IC can use advanced tools and techniques to exploit OS configuration flaws to gain control of the host server OS. The attacker can also exploit unused or unwanted services running on the host OS. For example, by default, the Sendmail service is running on Linux with the default settings which are known. The attacker can use these default settings to exploit a reported vulnerability of an older version of Sendmail. Table 6.29 provides threat levels assigned to A 1.3.1.1

and A 1.3.1.2.

Table 6.29: Threat Levels to Squid A 1.3.1.1 and A 1.3.1.2

TA	Capability	Motivation	TL to A 1.3.1.1	Capability	Motivation	TA to A 1.3.1.2
PU	1	1	1	1	1	1
NU	2	2	1	3	2	2
IC	3	4	3	4	3	4

A 1.3.2.1 and A 1.3.2.2 A remote attacker can manipulate the Squid Proxy configuration by exploiting a Squid vulnerability that exists due to default Squid settings. A remote attacker can also exploit default parameters, and options configured for Squid.

A highly motivated and focused attacker using advanced hacking tools and techniques can exploit the default parameters and options settings. For example, the default `http_port` (3128) which squid uses to listen to incoming requests is not changed, and `maximum_object_size` and `minimum_object_size` parameters are configured to manage the size of a cached object. It can then be manipulated by the attacker to cause DoS by requesting numerous large cache objects.

Moreover, `authenticate_ttl`, and `authenticate_ip_ttl` parameters should be configured carefully to set the client Time To Live (TTL) period, and authentication bounding to a particular IP, respectively. Table 6.30 provides threat levels assigned to A 1.3.2.1 and A 1.3.2.2.

Table 6.30: Threat Levels to Squid A 1.3.2.1 and A 1.3.2.2

TA	Capability	Motivation	TL to A 1.3.2.1	Capability	Motivation	TA to A 1.3.2.2
PU	1	1	1	1	1	1
NU	1	2	1	2	2	1
IC	4	3	4	5	4	5

Propagating Threat Levels Through Squid Attack Tree

In this section, threat levels are propagated to A 1.3.1 and A 1.3.2 attack steps. Later, threat levels are propagated to A 1.1, A 1.2, and A 1.3 attack steps by using the same process followed for Apache threat levels propagation (Section 6.3.2). Therefore, threat levels propagation to the A 1 root node only is presented in this section.

A 1 Manipulate Squid Proxy Server All three child nodes A 1.1, A 1.2, and A 1.3 are connected to the root node A 1 by a logical *OR*. Therefore, the maximal level of threat of these child nodes is propagated. Table 6.31 provides the threat levels propagated to the A 1 attack step.

Table 6.31: Threat Likelihood Levels to Squid

Type:*OR*	Likelihood for A 1.1 (Physical AV)	Likelihood for A 1.2 (Local AV)	Likelihood for A 1.3 (Network AV)	Likelihood for A 1 (Overall)
PU	3	2	1	**3**
NU	2	1	1	**2**
IC	2	1	4	**4**

Table 6.32 provides qualitative threat likelihood levels to Squid from PU, NU, and IC.

Table 6.32: Qualitative Threat Likelihood Levels to Squid

Risk No.	Attack Vector	TL from PU	TL from NU	TL from IC
R1	Physical	Moderate	Low	Low
R2	Local	Low	Negligible	Negligible
R3	Network	Negligible	Negligible	Substantial
R4	Network	Negligible	Negligible	Substantial

Threat levels are assigned to the Squid attack tree to determine the threat likelihood levels from PU, NU, and IC. Figures 6.16, 6.17, and 6.18 provide threat levels to the Squid proxy server from PU, NU, and IC, respectively. Figure 6.18 shows that an IC poses a Substantial threat level to realise a remote network attack by exploiting a vulnerability that may exist due to improper Squid Parameters, and Options configuration.

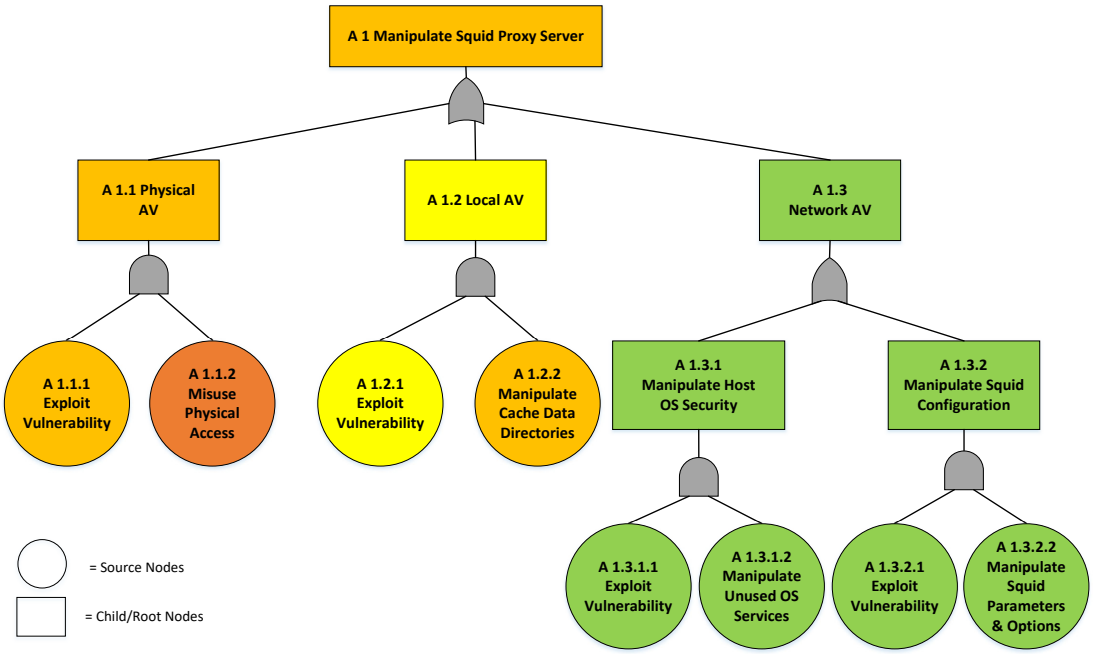


Figure 6.16: Squid Attack Tree for PU

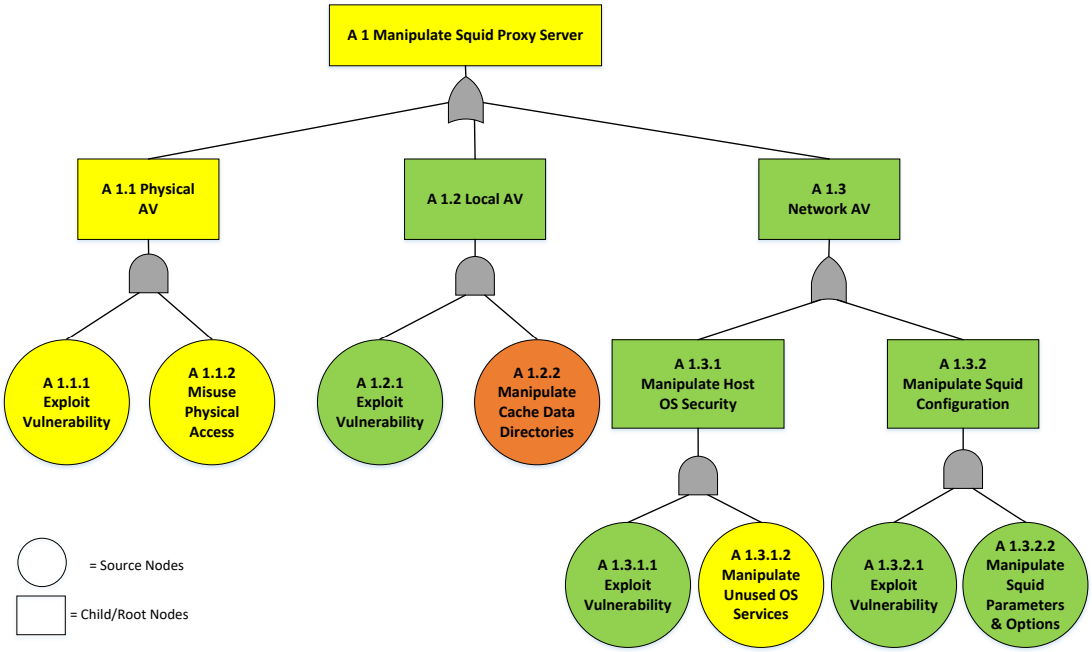


Figure 6.17: Squid Attack Tree for NU

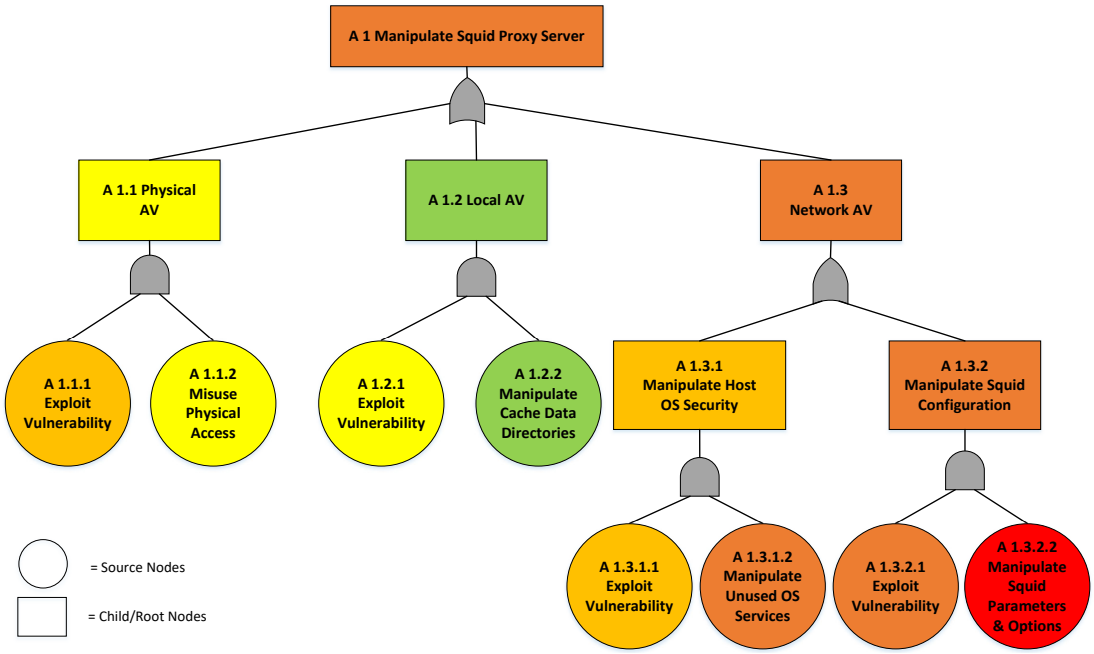


Figure 6.18: Squid Attack Tree for IC

6.5.3 Severity Levels of Risk to Squid

In this section, severity levels of risk to Squid are determined by combining vulnerability impact ratings and threat likelihood levels. Medium risk severity levels are determined for R1 and R2, and Low risk severity levels for R3 and R4 from both PU and NU. An IC poses Medium risk severity level for all the four risk types. Table 6.33 provides the risk severity levels to Squid from PU, NU, and IC.

Conclusion

Squid vulnerability prediction and risk assessment results highlight the flexibility of the research methods and the process. As in the case of Apache, research methods did not show limitations when applied to the Squid Proxy server. However, unknown Squid vulnerability prediction results were found to be a bit inaccurate (not close to the average reported vulnerabilities of the last ten years). The reason can be the missing factors, lack of optimised vulnerability reporting process, and smoothness of the reported vulnerability data (Roumani et al., 2015). On the other hand, threat likelihood and risk severity levels are determined accurately which concludes the generalisability of the Xen vulnerability prediction and risk assessment process to Squid.

6.6 Conclusion

The vulnerability prediction and risk assessment process was successfully evaluated by applying it to Apache HTTP and Squid Proxy servers. The applicability and generalisability of the risk assessment process was observed through near accurate results. NVD reported vulnerability datasets were used to predict unknown Squid and Apache vulnerabilities similar to the case of Xen. This shows that NVD reported vulnerability

Table 6.33: Severity Levels of Risk from PU, NU, and IC

Risk No.	**Attack Vector**	**Vul Impact Rating**	**TL from PU**	**Level of Risk from PU**	**TL from NU**	**Level of Risk from NU**	**TL from IC**	**Level of Risk from IC**
R1	Physical	Medium	Moderate	Medium	Low	Medium	Low	Medium
R2	Local	High	Low	Medium	Negligible	Medium	Negligible	Medium
R3	Network	Medium	Negligible	Low	Negligible	Low	Substantial	Medium
R4	Network	Medium	Negligible	Low	Negligible	Low	Substantial	Medium

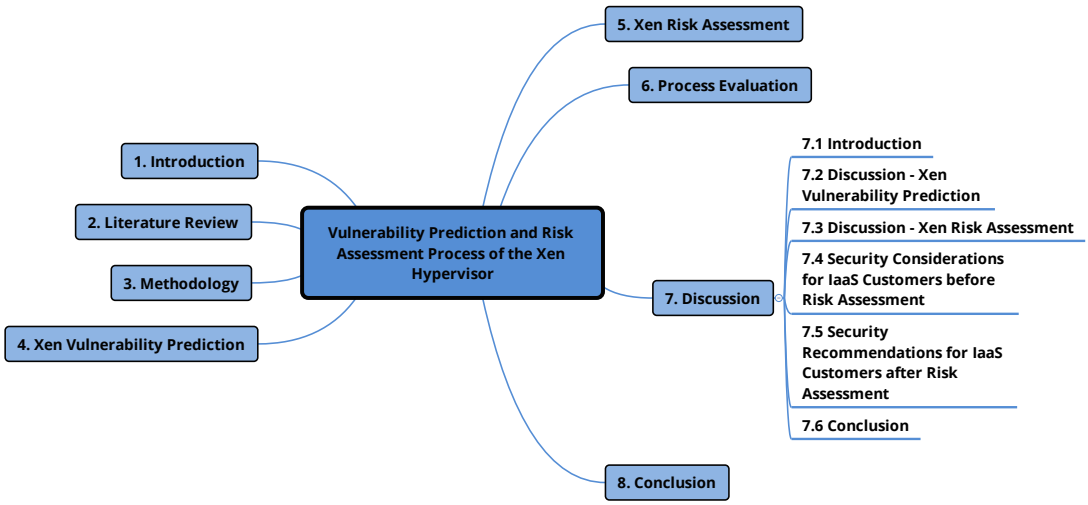
data could be effectively used to predict unknown vulnerabilities using the Holt-Winters method.

For risk assessment, the vulnerability impact ratings were not adopted like Xen. CVSS is used to score three example scenarios to determine impact ratings. Threat likelihood levels were determined by creating attack trees for both Apache and Squid. However, a detailed study was conducted to understand the architecture of both Apache and Squid to create attack trees and identify AVs.

A comprehensive search was conducted to identify and list the risk to both Apache and Squid separately. Severity levels of risk to both these servers were determined by using the same risk estimation matrix. Chapter 7 next provides a discussion of the Xen vulnerability prediction and risk assessment results.

Chapter 7

Discussion



7.1 Introduction

In Chapter 6, the evaluation of the vulnerability prediction and risk assessment process was conducted by applying it to Apache HTTP and Squid Proxy servers. This chapter reiterates the research problem and the research methods used to address the problem. It also covers the discussion of the Xen vulnerability prediction and risk assessment findings, and details how results support to address the research questions. Security

considerations and recommendations are provided for the IaaS customers to make informed security decisions to adopt IaaS and select an appropriate CSP.

This chapter is organised as follows: Section 7.2 covers the discussion of the Xen vulnerability prediction process. Section 7.3 covers discussion of the Xen risk assessment process. Security recommendations for customers before the risk assessment are covered in Section 7.4. Section 7.5 provides security recommendations for customers after performing the risk assessment of Xen. The conclusion of this chapter is provided in Section 7.6.

7.2 Discussion - Xen Vulnerability Prediction

This section provides a discussion of the vulnerability prediction process. The Holt-Winters quantitative prediction method is used to predict unknown vulnerabilities using the process of exponential smoothing. Using qualitative methods for the prediction can lead to inaccurate results when a quantitative approach is available and can be more appropriate. The qualitative prediction methods often result in inaccuracy (Newberne, 2006). On the other hand, quantitative methods such as the Holt-Winters method provides more accurate prediction without any substantial cost. The Holt-Winters method uses regression techniques, exponential smoothing parameters, and moving averages methods. It is a powerful prediction method which provides accurate prediction results. It is also easy to use and widely available to use through software such as Microsoft Excel. Usually, every organisation uses Microsoft Office suite. Thus, the Holt-Winters prediction method does not add extra cost in terms of buying specialised software tools for prediction.

Prediction models offer accuracy very rarely, but, accuracy is critical because inaccurate results can be costly in terms of time and resources. Organisations must carefully select the prediction model on the basis of their problem, and the type and

size of the data available to be used for the prediction. The selection of an accurate prediction model should be based on the Level, Trend, and Seasonal components of the available dataset. Moreover, some prediction models are not accurate for short or long-term prediction. However, the Holt-Winters method uses smoothing parameters to reduce the irregularities in the given dataset. It provides a more precise and effective technique to predict values for the future. An essential aspect of using exponential smoothing technique is that weights are applied to existing data values. Weights are set by giving more weight to recent and relevant observations compared to observations in the past. The exponential smoothing technique is very constructive to update the prediction results each time a new value becomes available at the end of a given dataset. It is observed as a useful method to predict future values using the most recent data. Vulnerability prediction for large software applications like the Xen hypervisor makes accurate and real-time prediction more critical.

In this research, to predict unknown Xen vulnerabilities, the Holt-Winters method is extended with the concept of exponentially weighted moving averages to predict different components of data variables. Holt-Winters is selected for prediction because the Xen reported vulnerabilities data contains Trend and Seasonal components. Holt-Winters has two methods: one is additive, and another is multiplicative. The multiplicative method is used here because the seasonal characteristics of the data depend on the current mean level of the time series data. The exponential smoothing parameters used for prediction using Holt-Winters methods required a smoothing constant set in the range of $0 < a < 1$. The constant value is considered to assign weights to the data observations. The final and optimal value of the smoothing constant depends on the time series data used for prediction. The value of α is normally between 0.05 and 0.3. However, α can be estimated by minimizing the sum of squared errors. The value of the weight for each component can be changed on the basis of the data used for the time series. When the value of α is high, more weight is assigned to the most recent

observations. On the other hand, the low value of α means that observations further in the past will be assigned more weight or given more importance. The multiplicative Holt-Winters method involves three smoothing parameters for the Level, Trend, and Seasonal component. Mathematically it is written as:

$$L_t = \alpha \times \frac{D_t}{S_t} + (1 - \alpha) \times (L_{t-1} + T_{t-1}) \tag{7.1}$$

$$T_t = \beta \times (L_t - L_{t-1}) + (1 - \beta) \times T_{t-1} \tag{7.2}$$

$$S_{t+p} = \gamma \times \frac{D_t}{L_t} + (1 - \gamma) \times (S_t) \tag{7.3}$$

The Xen reported vulnerability data time series is assigned three smoothing parameters. However, the method is adjusted by changing the values of smoothing parameters to determine the output with lower Mean Absolute Percentage Error (MAPE). Initially, the default values of smoothing parameters are used. Default values are set at 0.10 for Level, Trend, and Seasonal components. Afterwards, the weights that produced the lowest MAPE value are noted.

The Xen reported vulnerability data is selected from 2013 to 2017 and the seasonal period is four (quarterly for each year). The smoothing parameters are applied to 20 observations to predict the number of unknown Xen vulnerabilities for the first two quarters of 2018 (periods 21 and 22). The forecast error is then examined, and the one with the lowest MAPE is applied to predict the unknown vulnerabilities for the last two quarters of 2018 (periods 23 and 24), basically creating a new fitting period. The final predicted values are compared to the actual data values for the same time periods.

7.2.1 Reliability of the Prediction Model

The reliability of the prediction model is checked to determine the consistency of the prediction results. Initially, unknown Xen vulnerabilities are predicted for periods 21 and 22. Later, another prediction sample is created when the actual data values of these two periods are added to the original data set. Now, the calibration with the lower MAPE is applied to new data set with 22 periods instead of 20 periods to predict the unknown vulnerabilities for periods 23 and 23 (the last two quarters of 2018). The reliability of the prediction model is measured by assessing the extent to which the MAPE of the first data set (20 periods) is similar to the MAPE of the second data set (22 periods) when the same calibration of the prediction model is applied to both data sets.

7.2.2 Security Recommendations to Mitigate Hypervisor Vulnerability Exploitations

Like any other software package, the Xen hypervisor contains vulnerabilities. However, the concern from customers is what security tools and techniques CSPs are using to harden the hypervisor to mitigate the risk of vulnerability exploitation. Like the architectural and configuration risk, customers must ensure that CSPs use measures to protect the hypervisor code base. The hypervisor provides and manages the whole virtualised infrastructure and a vulnerability in the hypervisor leverages threats that may result in exploitation of VMs. Some of the security considerations for customers against software vulnerabilities are:

- Protection against single point of failure – The dynamic nature of the hypervisor can be a cause for concern if a malicious VM runs a code to compromise hypervisor instance. A single event of malware can also exploit other hypervisors in the same network environment resulting in a single point of failure.

- Like any other software, the hypervisor should be patched and updated regularly to help fix vulnerabilities.
- Vendor's website should be visited regularly to see the news or updates about the hypervisor.
- Vulnerability databases should also be checked regularly to know about new zero-day exploits.
- Controlled access to VMs should be ensured by properly managing privileges and access to VMs. The controlled access to VMs would help reduce the code base and its exploitation through malicious tools.

7.3 Discussion - Risk Assessment

In Section 7.2, the discussion of the Xen vulnerability assessment process was provided. This section provides a discussion of the Xen risk assessment process. Despite the many benefits of virtualised infrastructure, virtual systems are not risk free. In fact, virtualised infrastructure introduce new risk. Many of the vulnerabilities can appear due to architectural and configuration errors in the hypervisors. Compromise of virtualised infrastructure results in loss of CIA security objectives and can be a nightmare for customers using the compromised infrastructure. Exploitations by sophisticated threats lead to risks such as Loss of Business Reputation Due to Co-Tenant Activities, Isolation Failure, Malicious Insider, Intercepting Data in Transit, Data Leakage within Cloud, Undertaking Malicious Probes or Scans, Privilege Escalation, and Hypervisor Management Interface Compromise (Ruiz & Pedraza, 2016). However, the risks vary between cloud service delivery models, depending upon the type of hypervisor, security controls, security procedures, and risk management methods used to protect the hypervisors.

Selecting an appropriate CSP to move to cloud-based services is very important for organisations. However, in most of the cases, organisations may not have sufficient

knowledge to understand the risk to cloud virtualised infrastructure and perform the appropriate selection. The reason can be that they may not have required security experts and specialists for this important job and lack of visibility about the operations and security considerations of the CSPs. Therefore, for some organisations selecting an appropriate CSP is difficult based on their security requirements especially when there are many CSPs available for selection.

Customers should understand the risk after the move to cloud infrastructure and what security controls they need from the proposed CSP to ensure the security of their assets. They should ask the CSP where they stand against these risks and what security controls are in place. If the CSP is complying with the security considerations, a report of their risk assessment should be provided to customers showing how the controls are implemented, and the risks are being mitigated. If the CSP cannot provide an assessment report or does not correctly explain the security controls implementation, then this should be a cause for concern.

To make an appropriate selection of CSPs, customers need a platform to perform a risk assessment to understand the risk and security controls that should be in place by the CSPs to mitigate these risk. Therefore, the Xen risk assessment process enables customers to identify the risks and their severity levels to their data and information. Once the risk assessment results are understood by customers, CSPs should be asked questions about the risk control and mitigation strategies. However, instead of approaching CSPs directly, customers can use CSA's Cloud Controls Matrix (CCM) (STAR Registry, 2017), which is a robust framework available for organisations to help them select a CSP. CCM is based on different regulatory compliance and security standards. It provides details of the security controls that should be implemented by the CSP to meet the security requirements of customers. If they can find their proposed CSP in CCM, then they can look for the answers they are seeking. In case they do not find the CSPs, then customers should directly approach the CSP.

7.3.1 Implications of the Risk Assessment Process

Once the risk assessment is complete, customers can refer to the CSA Security, Trust & Assurance Registry (STAR) database (STAR, 2017). Customers can use the CSA Consensus Assessments Initiative Questionnaire (CAIQ) and CCM to determine which security controls are needed by the CSPs to manage these risk. Both CAIQ and CCM provides a way to prepare a proposal request for those security controls. It allows customers to quickly determine which security controls are compulsory and should be implemented by the CSPs to mitigate risk with high severity levels.

The CSA STAR database (STAR, 2017) offers a complimentary registry that provides the self-assessment results from more than 100 CSPs where more than 200 questions were answered by each CSP (STAR Registry, 2017). The registry entries document the detail of security controls implemented by the CSPs. This online registry database allows cloud customers to assess and select an appropriate CSP based on their risk assessment results. The CSA STAR is based on the research of the CSA Governance, Risk Management and Compliance (GRC) Stack. It provides four initiatives for customers, CSPs, and other key stakeholders. The GRC Stack components include CAIQ, CCM, Cloud Audit, and Cloud Trust Protocol (CTP). However, CAIQ and CCM are very useful for customers to assess the CSPs' security controls and procedures before making adoption decisions.

7.3.2 Xen Hypervisor Risk Assessment

The risk assessment process enables customers to accept or reject the risk based on their severity levels. Risk acceptance is a very sensitive stage, and it must be considered carefully. Risk decisions are critical for customers, and they should follow a formal process such as security cost-benefit analysis and benchmarking (Alruwaili & Gulliver, 2014). Customers should also discuss risk assessment results with internal and external

information security auditors to develop an understanding of risk.

Through the Xen risk assessment process, nine risk categories are identified along with the relevant vulnerabilities, and assets. Three example scenarios are presented to score Xen vulnerabilities from two different TAs to determine the impact ratings. However, all Xen vulnerability exploitation scenarios can be identified and scored to determine the impact ratings. Since it is not practicable to cover all possible exploitation scenarios, vulnerability impact ratings are adopted from ENISA's risk framework (Catteddu & Hogben, 2009). For example, Risk 1 is related to vulnerability V1 (Lack of Resource Isolation), V2 (Hypervisor Code Vulnerabilities), and asset A1 (Company reputation), A2 (Personal sensitive data), A3 (Personal data), and A4 (Service delivery). High impact rating is determined as the exploitation of both V1 and V2 can result in loss of service delivery and data leakage. The exploitation can also affect the reputation of the organisations. Table 7.1 provides a summary of the Xen risk, relevant vulnerabilities, and their impact ratings.

Table 7.1: Summary of Xen Risk, Vulnerabilities, and Impact Ratings

Risk Type	**Vulnerabilities**	**Impact Rating**
Loss of Business Reputation Due to Co-Tenant Activities	V1,V2	High
Isolation Failure	V1,V2,V3,V4	Very High
Malicious Insider	V5,V6	Very High
Intercepting Data In Transit	V3,V4,V5	High
Data Leakage	V3,V4,V5	High
Undertaking Malicious Probes or Scans	V3,V4	Medium
Compromise Hypervisor	V1,V2	Very High
Privilege Escalation	V2,V5	High
Management Interface Compromise	V5,V7	Very High

V1 = Lack of resource isolation, V2 = Hypervisor code vulnerabilities
V3 = Possibility of internal network probing, V4 = Possibility of co-residence checks
V5 = AAA Vulnerabilities, V6 = Inadequate physical security procedures
V7 = Remote access to management interface

Threat identification and the determination of their likelihood levels are an essential characteristic of the risk assessment process. Threats result in compromise of CIA security objectives. The loss of security objectives may happen when a threat exploits one or more assets. It is necessary for customers to know the threats and their likelihood levels. However, the threat likelihood levels may differ between customers. In this research, a qualitative threat likelihood assessment of Xen is performed. Attack trees are combined to generate a cyclic attack tree for Xen. The analysis results show that out of two TAs, an NU is most likely going to realise a network attack by exploiting a vulnerability in the virtual network tool stack to pose a **Severe** threat level. Whereas, a PU is going to exploit a physical vulnerability to compromise hypervisor. Table 7.2 provides threat likelihood levels to Xen from both PU and NU.

Table 7.2: Threat Likelihood Levels from PU and NU Threat Actors

Risk Type	**Attack Vector**	**PU Threat Likelihood Level**	**NU Threat Likelihood Level**
Loss of Business Reputation Due to Malicious Co-Tenant	Local	Negligible	Moderate
Isolation Failure	Local	Negligible	Moderate
Malicious Insider	Physical	Moderate	Negligible
Intercepting Data in Transit	Network	Negligible	Severe
Data Leakage	Local	Negligible	Moderate
Undertaking Malicious Probes or Scans	Network	Negligible	Severe
Compromise Hypervisor	Physical	Moderate	Severe
Privilege Escalation	Local	Negligible	Moderate
Management Interface Compromise	Network	Negligible	Severe

Vulnerability impact ratings and threat likelihood levels are mapped together to determine the severity levels of risk to hypervisors. Table 7.3 provides a summary of the risk severity levels from the PU and NU TAs.

Table 7.3: Severity Levels of Risk to Xen

Risk Type	Levels of Risk from PU	Levels of Risk from NU
Loss of Business Reputation Due to Malicious Co-Tenant	Medium	Medium
Isolation Failure	Medium	High
Malicious Insider	High	Medium
Intercepting Data in Transit	Medium	High
Data Leakage	Medium	Medium
Undertaking Malicious Probes or Scans	Low	High
Compromise Hypervisor	High	High
Privilege Escalation	Medium	Medium
Management Interface Compromise	Medium	High

7.4 Security Recommendations for Customers Before the Risk Assessment

This section provides security considerations for customers who are planning to move to cloud infrastructure and want to go through a risk assessment process to make informed adoption decisions.

7.4.1 Critical Assets

Before assessing the risk to virtualised infrastructure, customers need to understand the criticality and security of the data and information which will move to the cloud infrastructure. It is also important for the customers to know the sensitivity of the data. Otherwise, customers will end up placing their critical data in the cloud infrastructure where appropriate security controls and procedures are not in place to mitigate risk. On the other hand, if the criticality of the data is exaggerated, then it could lead to the implementation of additional security controls that would add more cost and require

more resources to manage. So, it is important for customers to classify the security of data carefully to ensure the appropriate protection.

7.4.2 Data Privacy

Organisations who want to move to cloud infrastructure need to perform a Privacy Impact Assessment (PIA) to ensure the privacy of their personal information and risk associated with cloud infrastructure along with the security controls and procedures required to mitigate those risks. In a typical scenario, CSPs normally have guidelines to define how personal information of customers will be gathered and used. Therefore, organisations must identify and consider the implications of accepting a CSP's privacy policy and guidelines.

7.4.3 Data Confidentiality

Data stored in the cloud infrastructure are more vulnerable to unauthorised access compared to the case where data are stored on a local server. However, the factors that may allow unauthorised access to data in the cloud varies between service delivery models. Implementing and managing security controls to ensure the confidentiality of the data depends entirely on the delivery model. Moreover, the type of cloud deployment also affects the implementation and management of control requirements.

7.4.4 Data Integrity

CSPs provide better security and protection against data loss or corruption compared to an organisation who manages their own data centre environment. Some CSPs offer data backup and recovery services as optional services along with core cloud services. However, some CSPs offer data backup services at additional cost and even some CSPs do not offer backup services at all. Therefore, it is better for the customers to assess and

select a CSP which provides them better data backup and recovery services to avoid data loss and corruption. Furthermore, it is necessary for customers to determine the data protection level by assessing how CSPs protect against data loss and corruption. Customers should also determine the level of granularity they have to restore the data after an incident happens. Therefore, customers should assess the CSPs data backup and recovery processes to ensure the compliance of their organisational security practices.

7.4.5 Data Availability

SLA is an important attribute of availability. SLA usually provide the level of availability a CSP is going to provide for a cloud service. The customer should have a clear understanding of the percentage of availability they need. Customers should also be able to assess whether or not these availability levels meet their organisational requirements.

The risk of DoS attacks is not entirely different in CC compared to a traditional data centre environment. However, the increased rate of adoption of cloud services has increased the level of DoS risk because DoS attack happens due to the aggregation of multiple VMs sharing the same hardware and this may be more attractive for an attacker. A customer can be affected by DoS attack if it is launched against the customer itself, CSP, and even other tenants.

7.5 Security Recommendations for Customers After the Xen Risk Assessment

Section 7.4 provide the security considerations for the customers before the risk assessment. In this section, security recommendations are provided for customers to make informed security decisions after the risk assessment. The risks with High severity are

explained along with the security considerations relevant to each risk type. It would guide customers to seek answers from the proposed CSPs that how they mitigate these types of risk.

7.5.1 Security Recommendations for Risk Severity Levels Posed by a PU

This section details the reasons for risk severity levels from a PU TA and security recommendations for customers. Table 7.4 provides the risk with High severity levels from a PU TA. Risk types where the threat level is Negligible does not result in High severity levels. Therefore, the risks with High severity levels are considered here. Risk assessment results show that R3 (Malicious Insider) and R7 (Compromise Hypervisor) pose a High severity level to Xen from a PU.

Table 7.4: Severity Levels of Risk to Xen from a PU

Risk Type	**Levels of Risk from PU**	**Impact on Security Objective**
Malicious Insider	High	Integrity and Availability
Compromise Hypervisor	High	Integrity and Availability

Recommendations to Mitigate a Malicious Insider Risk

A high risk severity level is determined for this risk type. Malicious insider risk is one of the most common risks for many customers who are planning to move to cloud infrastructure services. A malicious insider can access and manipulate sensitive information that belongs to different cloud customers. Customers should ascertain whether CSPs follow appropriate procedures to manage this risk. To mitigate the risk of a malicious insider, the CSPs should consider the following points:

- A proper background check should be performed by the CSPs before hiring their employees. However, the amount of detail that customers can have about the CSPs' hiring process depends on the physical location of the CSP and services offered. An efficient hiring process may help CSPs to hire a person with no previous record of being untrustworthy. However, scrutinizing may not help the CSPs to identify candidates who are untrustworthy but do not have criminal records or history. Moreover, it would be difficult to identify a malicious employee who has been trustworthy for a long time and is now untrustworthy.
- The CSPs should monitor employees' activities and enforce separation of duties to reduce the chances of a malicious insider executing unauthorised activities. Monitoring employees' activities can help the CSPs to a large extent to manage malicious insider risk. Monitoring should include all the day-to-day activities that all the employees perform in their job such as logical access use, physical access use, and accessing media or content that contains customers' data. CSPs should maintain a log of these activities to identify any malicious activity. The activity logs should be protected from alteration or deletion by ensuring separation of duties among employees. For example, the administrator of one hypervisor and host OS should not be given access to delete or change the log file of that system. Log files should also not be saved on the local server where administration has access.
- Just like an efficient hiring process, the CSPs should use appropriate practices when terminating employees. Their access to the servers and network should be monitored carefully during the last days of their job. Moreover, their access to the servers and network should be revoked immediately after they leave the organisation.
- The CSPs should collect and regularly analyse the server and network logs and make sure that the IT staff is aware of that. This will make the IT staff realise

that their access activities are monitored which would eventually decrease the chances of a malicious insider attack.

Recommendations to Mitigate the Compromise Hypervisor Risk

This type of risk also poses a High severity level. Compromising the hypervisor would be useful to escape the isolation between different VMs. It would result in unauthorised access to data stored on different VMs. Moreover, unauthorised access can lead to monitoring and modifying the data residing on the VMs. A compromised hypervisor can also lead to a reduction of the hardware resources to the VM, resulting in DoS.

Therefore, due to this High severity risk, CSPs should implement appropriate controls to ensure that the hypervisor or host server are physically protected from a malicious insider or third-party contractors. Physical security of CSPs' data centres where hypervisors and host servers are running, is vital to ensure the overall security of the virtualised infrastructure.

Physical access to the host server on which the hypervisor is running should be monitored and controlled by the CSP. All unused hardware such as removable disks should be disconnected from the server after use. Extra network cards should also be removed from the host server if not in use and only connected to a network.

In addition to data theft by misusing physical access, a malicious administrator can alter the security settings of the host server to access the hypervisor later remotely. Therefore, remote access to the hypervisor and the host server should also be controlled and monitored to mitigate this risk type.

Using the above security considerations, customers can ask questions from the CSPs. However, sometimes it would be impossible for the customers to understand the physical controls that are implemented by a CSP. In such as case, customers should rely on a third party's audit report that covers a physical security assessment of a proposed CSP.

7.5.2 Security Recommendations for Risk Levels posed by an NU

This section details the reasons for risk severity levels from an NU TA and security recommendations for customers. Table 7.5 provides the summary of the risk with High severity levels from an NU TA.

Table 7.5: Severity Levels of Risk to Xen from an NU

Risk Type	**Levels of Risk from NU**	**Impact on Security Objective**
Isolation Failure	High	Confidentiality and Integrity
Intercepting Data in Transit	High	Confidentiality and Integrity
Undertaking Malicious Probes or Scans	High	Integrity and Availability
Compromise Hypervisor	High	Integrity and Availability
Management Interface Compromise	High	Confidentiality and Integrity

Recommendations to Mitigate the Isolation Failure Risk

This risk type poses a High severity level. It includes the failure of security controls and mechanisms which CSPs have implemented to separate storage, memory, and network between different VMs running on the same shared hardware.

The hypervisor allows CSP to take a snapshot or copy of the memory and disk of a running VM at a point in time to use it for backup and recovery purposes. If CSPs do not have security controls implemented to protect these snapshots adequately, then an attacker can gain unauthorised access to the data which is stored on the VM's local drives and also the encryption keys which are stored in the memory. Therefore, customers should be concerned about the security of their data as there is lack of visibility about the CSPs' security controls together with the patch and vulnerability

management policies to ensure the security of data stored on the VMs.

Moreover, to mitigate this type of risk, CSPs should disable all unnecessary services which are not in use by the hypervisor. For example, file sharing between guest VMs and host OS should be disabled if not required. File sharing can be used as an attack vector where the same folder on the host server is shared between host OS and different VMs.

The hypervisor's introspection capabilities should be used effectively to monitor the security of guest VMs. It should also be used to monitor the security of the hypervisor's functionalities shared between VMs.

Recommendations to Mitigate the Intercepting Data in Transit Risk

The distributed and dynamic nature of virtualisation implies more data in transit than the conventional IT environment. Different network attacks such as sniffing, spoofing, man-in-the-middle, and replay attacks should be considered as possible threat sources.

The hypervisor connects multiple guest VMs together through virtual switches. However, it can be a security concern for some organisations where there are policies to monitor all the networks in different ways. Virtual network switches in most hypervisors do not have capabilities to monitor network traffic between VMs compared to physical network switches which can be managed to monitor network traffic between physical servers. However, some hypervisors allow the creation of VLANs using virtual switches. They also provide firewall capabilities to keep the traffic separate that belongs to different VMs. Therefore, CSPs should implement additional security appliances to inspect, control and monitor VMs' network communication in a central location.

VMs should be protected with both host-level and network-level security measures. In the case of VMs, the virtual network configuration plays a vital role in the security of VMs. Four virtual network configuration areas, network segmentation, network path redundancy, traffic control using a firewall, and VM traffic monitoring, should be

considered carefully by the CSPs.

Moreover, the network should be segmented, for example, the hypervisor management traffic should be in a different VLAN. VMs and the hypervisor management traffic in the same network can allow an attacker to exploit a network vulnerability to attack the hypervisor for a malicious VM. The segregation of a network to keep the VMs traffic segmented from hypervisor management traffic should mitigate the risk.

Recommendations to Mitigate the Undertaking Malicious Probes or Scans Risk

Though Malicious Probes or Scans are indirect risks to the virtualised infrastructure, they pose a High severity level when an attacker tries to collect information to launch further attacks.

The hypervisor should not be placed in the same network where all the VMs are located to mitigate this risk. A separate network such as VLAN should be created to separate the hypervisor management and remote access traffic. The ping and traceroute replies should also be blocked for the network/subnet where the hypervisor is running.

Moreover, the management of the hypervisor from a network location should be disabled most of the time. Also, the management interfaces should be appropriately configured. Inaccurate configuration can expose the hypervisor to attacks from the network.

Recommendations to Mitigate the Compromise Hypervisor Risk

This risk also poses a High severity level and to mitigate it, CSPs should install the hypervisor with proper security considerations and vendor recommended best practices to ensure its security. The CSP should follow best practices when managing the hypervisor:

- Extra hardware such as external hard disk drives should not be left connected

after a task is completed such as backup.

- Clipboard or file sharing services should not be enabled that allow a guest OS to access the file from the host OS.
- Vendor website or notifications should be considered for important news of bulletins.
- Management interfaces for remote access should be properly secured or monitored.
- Access to the hypervisor and management software should be restricted.
- Proper access controls to manage administrative access should be implemented.

Recommendations to Mitigate the Management Interface Compromise Risk

Securing the management interface of the hypervisor is critical to protecting against attacks that use network access as AV. The management interface should be secured from both local and remote network attacks. If an administrator needed to access the hypervisor through the network and required access to management interfaces, then access to management interfaces, network traffic must be encrypted using Virtual Private Network (VPN) to encapsulate the traffic. The network traffic should also be secured through the firewall. Security policies should be applied to restrict unauthorised access to these management interfaces. VLANs can also be considered where there should be a separate management network which can only be accessed by the administrator.

Moreover, CSPs should use token-based authentication to allow access to the hypervisor through management interfaces. The access to the hypervisor should be restricted on the basis of the policies specifying who can or cannot access the hypervisor.

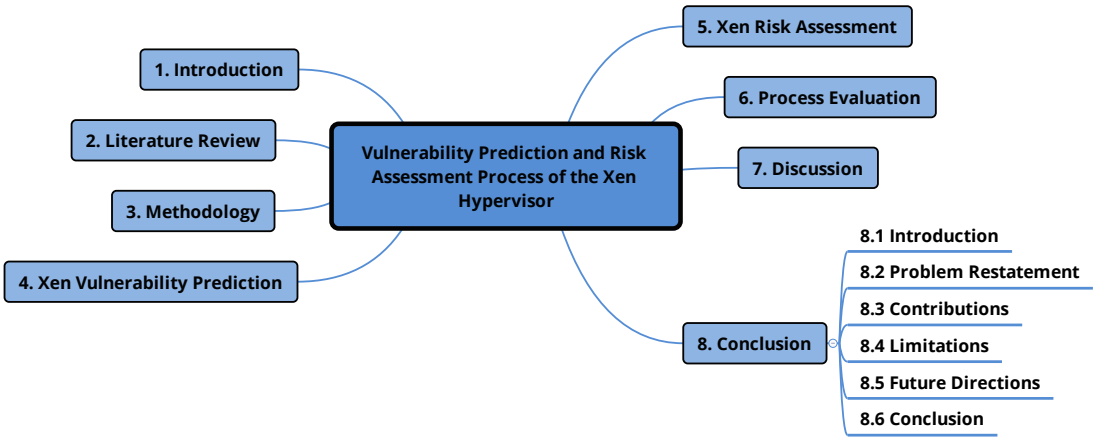
7.6 Conclusion

The broad characteristics of cloud services such as multi-tenancy, elasticity, data residency, wide network access, and shared responsibilities between customers and CSPs makes them very dynamic and increases their scale. Cloud virtualised infrastructure leverages more threats and risks that can have a serious impact on the security and privacy of customers' data. To preserve the security level of the data and information, customers need to know the risks and their severity levels, and security controls to mitigate these risks.

Through the Xen risk assessment process, the customers can identify the risk to the Xen hypervisor. Once the risks and their severity levels are determined, the customers can ask questions of CSPs about their security controls and procedures. The security recommendations are also provided for customers for the risk with High severity levels. Customers can use these recommendations to develop their understanding of the required security controls of a proposed CSP to mitigate these risks. This would eventually result in an increased cloud infrastructure adoption rate. CSPs would also realise the sensitivity of the stored data and place proper security controls to minimise the security risk to encourage customers to adopt cloud infrastructure. Chapter 8 next provides the conclusion, problem re-statement, limitations, and future research directions of this research.

Chapter 8

Conclusion



8.1 Introduction

In the Chapter 7, a discussion of the Xen vulnerability prediction and risk assessment process was provided. This chapter provides the conclusion of the thesis, research limitations, and directions to extend this research in the future. The chapter briefly restates the problem area and highlights the functionalities of the vulnerability prediction and risk assessment process. It also covers how the process addresses the research problem.

This thesis presented work that contributes to the security of the Xen hypervisor which provides and manages virtualised infrastructure. The literature review conducted in Chapter 2, identified the research gaps and raised research questions to drive this research. Chapter 3 provided the detail of DSR methodology that was adapted to conduct this research. The Time Series Holt-Winters Method, Regression Analysis, ENISA's risk framework, CVSS, Asset-driven Structured Analysis, and Risk Estimation Matrix were used to test the hypotheses and address the research questions. The major contributions of this research were presented in Chapters 4, 5 and 6. A discussion on the Xen vulnerability prediction and risk assessment process was presented in Chapter 7.

This chapter is organised as follows: Section 8.2 restates the research problem. Section 8.3 provides the limitations. Future directions of this research are provided in Section 8.4. The conclusion of this chapter is provided in Section 8.5.

8.2 Problem Re-statement

CC is an emerging computing paradigm that offers benefits to organisations such as unlimited hardware resources, energy savings, ease of maintenance, management of systems and data, and affordability. In other words, it provides an efficient, flexible, scalable, and cost-effective platform to organisations who consume IT services to deliver their businesses. It offers three service delivery models to organisations, such as infrastructure, platform and software as a service. However, despite offering many advantages, CIA of customers' data and the information is the biggest concern. Moreover, each of the delivery models faces different risks that impact service delivery to its customers. Furthermore, vulnerability and risk assessment of CC is difficult due to multiple threats from different sources such as CSPs, technological vulnerabilities, and malicious activities from co-tenants.

CC introduces new risks to assets compared to the risks of conventional IT environments. New and unknown risks raise security concerns and result in a lack of adoption of cloud services by many organisations. Much research has been conducted regarding risk assessment of CC, but it assesses risk from a broader perspective rather than focusing on CC service delivery models. The fact is, threats and risks vary between these service delivery models. The existing research lacks implementation details and inherits common limitations such as assumptions to determine impact ratings and the probability of threat occurrence. Therefore, these limitations highlight gaps in existing research and drive the need for an optimised vulnerability and risk assessment platform.

Risk assessment is a process of risk management which provides a platform to examine the vulnerabilities, analyse threats, and determine overall risk to assets (Alturkistani & Emam, 2014; Kiran, 2014). The risk assessment process can lead to recording and prioritising risk to make security decisions. The risk assessment can be qualitative or quantitative, or a combination of both. For qualitative risk assessment, the output of the assessment process is a self-descriptive value such as high, medium and low risk. However, for quantitative risk assessment, the result is a numeric value such as a probability, a proportion, or an expected rate (Cayirci, 2015). Risk assessment has a solid relationship with the analysis of vulnerability impact and the likelihood of threat occurrence (X. Zhang et al., 2010).

8.3 Limitations

Apache and Squid reported vulnerability data is less smooth due to inconsistent trend and seasonal factors. The number of reported vulnerabilities vary in all quarters of the last ten years. Some quarters do not have even a single vulnerability reported. Furthermore, the vulnerability datasets are very small as only 100 Apache, and 56 Squid vulnerabilities were reported to NVD. Therefore, inconsistent data resulted in

less accurate results. Symmetric Mean Absolute Percent Error (SMAPE) was used to measure the accuracy of the predicted results (Equation 8.1). MAPE was not considered because the data that contains zero values may have twisted the overall prediction error. SMAPE has a lower bound of 0% and an upper bound of 200% to avoid problems of zero values when determining prediction error.

$$SMAPE = \frac{200}{N} \times \sum_{i=1}^{N} |\frac{ActualData - Prediction}{ActualData + Prediction}| \tag{8.1}$$

Apache prediction results produced 77.61% SMAPE. Squid prediction results are also less accurate with 105% SMAPE (SMAPE upper bound is 200%). However, Xen prediction results are better with 50.30% SMAPE because Xen reported vulnerability data is relatively smoother than Apache and Squid.

A structured analysis approach using attack trees was used to determine threat likelihood levels. However, attack trees tend to be used at high levels of abstraction. Attack trees are also limited to viable threat-only vectors that contain implicit information such as assets, vulnerabilities, and TAs (Hutle et al., 2015).

The CSA STAR database can be searched for answers to make security decisions. More than 100 CSPs answered the questionnaire, and each CSP answered more than 200 questions. The answers from CSPs in most cases were not just Yes or No, but were descriptive. Therefore, the search would require a significant amount of time from customers to find the relevant answers to make adoption decisions.

The vulnerability prediction process uses reported vulnerability data from online vulnerability databases. However, there is another limitation related to proprietary software where vendors do not report their vulnerabilities to databases.

8.4 Future Directions

This section provides the directions to extend this research in the future.

8.4.1 Vulnerability Prediction

Currently, the number of reported vulnerabilities is used as an input dataset to predict vulnerabilities for future. However, the reported vulnerability data which is used for prediction treats all the reported vulnerabilities equally. The accuracy of the prediction model could be improved by including additional factors such as an access complexity score and the impact level of each reported vulnerability to the input dataset. The prediction model could be then further optimised to predict other dependent variables such as *type* and *frequency* of the reported vulnerabilities instead of only predicting the *number* of unknown vulnerabilities.

Another future direction would be to use other prediction models such as ARIMA prediction models to predict unknown vulnerabilities using the same input dataset. Roumani et al. (2015) uses both ARIMA and exponential smoothing time series methods to predict the number of vulnerabilities for five web browsers. ARIMA could be considered to predict the unknown vulnerabilities of large software applications like Xen. The prediction results of ARIMA could be compared with the Holt-Winters prediction to see which model provides more accurate results with the given input dataset.

8.4.2 Threat Likelihood Assessment

Threat likelihood assessment is an essential phase of the Xen risk assessment process. The accuracy of risk assessment results depends on the accuracy of the threat likelihood assessment process. In this research, a structured analysis approach is used for threat likelihood assessment. It provides a broader and high-level view of threats using attack

trees. However, attack trees become unwieldy on devices or at implementation specific lower levels. Therefore, an optimised threat modelling technique such as Semantic Threat Graphs (STGs) could be used to ensure an in-depth threat analysis at a lower technical level. STGs are better for the assessment of specific components and implementations on the lower level that includes identification of suitable countermeasures. However, STGs are complex and lack methods to model weak entry points of the system and between different components of the system.

Therefore, to have an optimised threat assessment process, attack trees and STGs could be combined to develop a more comprehensive and understandable map of the potential weak entry points of the system. This hybrid process would also provide detailed knowledge and protection strategies for the system and components such as STGs make the information implicit which is explicit in an attack tree. Therefore, it is suggested that STGs could be used along with attack trees to identify not only the threats but also the countermeasures for the identified threats.

Hutle et al. (2015) proposes the use of STGs to explicitly present the relationships between the threats and the security controls. The attack graphs are used as a generalisation of attack trees, a Smart Grid Architectural Model (SGAM) based model, and attack patterns to identify threats to key information assets in a smart grid. The purpose is to construct comprehensive graphs that describe threats to large software applications like the Xen hypervisor.

8.5 Conclusion

Cloud computing due to its dynamic nature leverages different security risks to each of its service delivery models. There is a long list of factors on which risks are dependent upon such as data or information belonging to customers, scalable and flexible architecture, and immature security controls. The lack of security controls to mitigate

risk in CC results in lack of adoption of cloud services by the organisations. This research presents a vulnerability prediction and qualitative inductive risk assessment process that targets the Xen hypervisor. A hypervisor is a core component of the IaaS service delivery model. Security management is mostly the responsibility of customers in this delivery model; however, CSPs are only responsible for the security of the hypervisor. Customers are unaware of the security of the virtualised infrastructure where their data and information will be stored after the move to cloud infrastructure. This raises concerns from many customers resulting in a lack of adoption of cloud infrastructure services. Therefore, a platform for customers to analyse vulnerabilities and risks to these hypervisors is desirable.

Prediction of vulnerabilities can help minimise the damage therefore, it is desirable to identify the vulnerabilities earlier in software systems to help reduce the cost of damage and also the loss of reputation which can be caused by a successful exploitation. This research presented an unknown vulnerability prediction process that would enable the organisations to identify the security tools to harden the hypervisor to mitigate the risks of vulnerability exploitations. Time Series Holt-Winters method is used to predict the number of unknown vulnerabilities as it is a good fit for the vulnerability datasets (Roumani et al., 2015). This research used the reported Xen vulnerability dataset retrieved from the NVD. The prediction model predicted 41.85 vulnerabilities for 2018. The average reported Xen vulnerabilities from 2013 to 2017 were 43.80 vulnerabilities per year. The validity and reliability of the prediction model was determined by using MAPE. MAD was used to measure the accuracy of the result using control chart by tracking the prediction. Prediction model did not exceed thresholds levels, as tracking signal was well under the upper and lower control limits. Security recommendations were also provided to mitigate vulnerability exploitation scenarios for example, the hypervisor should be patched and updated regularly, the vendor's website should be visited regularly to see the news or updates, vulnerability databases should also be

checked regularly to know about new vulnerabilities regarding the hypervisors.

Risk assessment of the cloud and effective management is a challenging research problem. To minimise the security concerns of customers and to convince them to adopt IaaS, risks and their severity levels must be known to make informed security decisions. The risk assessment process in this research would allow customers to have a comprehensive risk assessment platform to analyse Xen security and make decisions to adopt Xen based virtualised infrastructure. The risk assessment would allow them to avoid and reduce threats, adverse actions, and attacks by quantifying the risks and implementing mitigation strategies. Through risk assessment, nine technical risks of Xen were identified along with the relevant vulnerabilities, and assets. The results show that R3 (Malicious Insider) and R7 (Compromise Hypervisor) pose a High severity level to Xen from a PU threat actor. The assessment results also show that an NU threat actor poses a High risk severity levels for R2 (Isolation Failure), R4 (Intercepting Data in Transit), R6 (Undertaking Malicious Probes or Scans), R7 (Compromise of Hypervisor), and R9 (Management Interface Compromise). Later, security recommendations were made for the organisations to mitigate the risks with high severity. However, customers are encouraged to identify and add new risks in the assessment process that may be specific to their services, data, and information. The customers can consider the security recommendations made in this research to improve the security of their Xen based virtual environment or optimise selection process of a cloud service provider after analysing the security controls and procedures.

The vulnerability prediction and risk assessment process was evaluated by applying it to the Apache HTTP and Squid Proxy software packages. Apache and Squid were chosen to demonstrate the generalisability and applicability of the process to other open source infrastructure level software packages. Moreover, Apache and Squid were targeted for their wide level of usage around the computer world as a web cache and proxy servers respectively. Furthermore, being open source software packages, complete

reported vulnerability data was available through vulnerability databases. The evaluation process did not show limitations and indicates that it is not explicitly developed for the Xen hypervisor. It can also be applied to other open source infrastructure level software packages. The process would not require changes in analysis methods, but a thorough study would be required to identify the new risk that may be introduced due to advancements in cyber-attacks. Also, new types of vulnerabilities and threats may appear due to upgradations of Xen.

References

Acunetix. (2017). *Apache web server security: Apache security* [Internet web page]. Retrieved from `https://www.acunetix.com/websitesecurity/apache-security/`

Adhikari, R. & Agrawal, R. (2013). An introductory study on time series modeling and forecasting. *arXiv preprint arXiv:1302.6613*.

Albakri, S. H., Shanmugam, B., Samy, G. N., Idris, N. B. & Ahmed, A. (2014). Security risk assessment framework for cloud computing environments. *Security and Communication Networks*, *7*(11), 2114–2124.

Alhazmi, O. H. & Malaiya, Y. K. (2005a). Modeling the vulnerability discovery process. In *16th ieee international symposium on software reliability engineering (issre'05)* (pp. 10–pp).

Alhazmi, O. H. & Malaiya, Y. K. (2005b). Quantitative vulnerability assessment of systems software. In *Proc. annual reliability and maintainability symposium* (pp. 615–620).

Alruwaili, F. F. & Gulliver, T. A. (2014). Safeguarding the cloud: An effective risk management framework for cloud computing services. *International Journal of Computer Communications and Networks (IJCCN)*, *4*(3), 6–16.

Alturkistani, F. M. & Emam, A. Z. (2014). A review of security risk assessment methods in cloud computing. In *New perspectives in information systems and technologies, volume 1* (pp. 443–453). Springer.

Alva, A., Caleff, O., Greg, E., Lum, A., Pasley, K., Sudarsan, S. & Venkitaraman, R. (2013). The notorious nine: cloud computing top threats in 2013. *Cloud Security Alliance*.

Arora, A., Nandkumar, A. & Telang, R. (2006). Does information security attack frequency increase with vulnerability disclosure? an empirical analysis. *Information Systems Frontiers*, *8*(5), 350–362.

Ayala, I. D. C. L., Vega, M. & Vargas-Lombardo, M. (2013). Emerging threats, risk and attacks in distributed systems: Cloud computing. In *Innovations and advances in computer, information, systems sciences, and engineering* (pp. 37–51). Springer.

Barrowclough, J. P. & Asif, R. (2018). Securing cloud hypervisors: A survey of the threats, vulnerabilities, and countermeasures. *Security and Communication Networks*, *2018*.

Bazargan, F., Yeun, C. Y. & Zemerly, M. J. (2012). State-of-the-art of virtualization, its security threats and deployment models. *International Journal for Information*

Security Research (IJISR), *2*(3/4), 335–343.

Bibi, S., Tsoumakas, G., Stamelos, I. & Vlahavas, I. P. (2006). Software defect prediction using regression via classification. In *Aiccsa* (pp. 330–336).

Brohi, S. N., Bamiah, M. A., Brohi, M. N. & Kamran, R. (2012). Identifying and analyzing security threats to virtualized cloud computing infrastructures. In *Cloud computing technologies, applications and management (iccctam), 2012 international conference on* (pp. 151–155).

Catteddu, D. & Hogben, G. (2009). Benefits, risks and recommendations for information security. *European Network and Information Security*.

Cayirci, E. (2015). Models for cloud risk assessment: A tutorial. In *Accountability and security in the cloud* (pp. 154–184). Springer.

Cayirci, E., Garaga, A., De Oliveira, A. S. & Roudier, Y. (2016). A risk assessment model for selecting cloud service providers. *Journal of Cloud Computing*, *5*(1), 14.

CESG. (2009). Hmg ia standard no. 1 technical risk assessment. *CESG recommendation*.

Chhabra, S. & Dixit, V. (2015). Cloud computing: State of the art and security issues. *ACM SIGSOFT Software Engineering Notes*, *40*(2), 1–11.

Chou, Y. (2010). *Cloud computing primer for it pros* [Internet web page]. Retrieved from `https://blogs.technet.microsoft.com/yungchou/2010/11/15/cloud-computing-primer-for-it-pros/`

Daskala, B. & Le Metayer, D. (2012). *Methodology for privacy risk management* [Internet web page]. Retrieved from `https://www.cnil.fr/sites/default/files/typo/document/CNIL-ManagingPrivacyRisks-Methodology.pdf`

Dawoud, W., Takouna, I. & Meinel, C. (2010). Infrastructure as a service security: Challenges and solutions. In *Informatics and systems (infos), 2010 the 7th international conference on* (pp. 1–8).

Djenna, A. & Batouche, M. (2014). Security problems in cloud infrastructure. In *Networks, computers and communications, the 2014 international symposium on* (pp. 1–6).

Edge, K., Raines, R., Grimaila, M., Baldwin, R., Bennington, R. & Reuter, C. (2007). The use of attack and protection trees to analyze security for an online banking system. In *System sciences, 2007. hicss 2007. 40th annual hawaii international conference on* (pp. 144b–144b).

Edge, K. S., Dalton, G. C., Raines, R. A. & Mills, R. F. (2006). Using attack and protection trees to analyze threats and defenses to homeland security. In *Military communications conference, 2006. milcom 2006. ieee* (pp. 1–7).

Elahi, G., Yu, E. & Zannone, N. (2010). A vulnerability-centric requirements engineering framework: analyzing security attacks, countermeasures, and requirements based on vulnerabilities. *Requirements engineering*, *15*(1), 41–62.

Elhage, N. (2011). Virtunoid: Breaking out of kvm. *Black Hat USA*.

Fitó, J. O. & Guitart, J. (2014). Business-driven management of infrastructure-level risks in cloud providers. *Future Generation computer systems*, *32*, 41–53.

Gallon, L. & Bascou, J. J. (2011). Using cvss in attack graphs. In *Availability, reliability and security (ares), 2011 sixth international conference on* (pp. 59–66).

Geng, J., Ye, D. & Luo, P. (2015). Forecasting severity of software vulnerability using grey model gm (1, 1). In *Advanced information technology, electronic and automation control conference (iaeac), 2015 ieee* (pp. 344–348).

Haque, S., Keffeler, M. & Atkison, T. (2017). An evolutionary approach of attack graphs and attack trees: A survey of attack modeling. In *Security and management, 2017 international conference on* (pp. 224–229).

Hashizume, K., Rosado, D. G., Fernández-Medina, E. & Fernandez, E. B. (2013). An analysis of security issues for cloud computing. *Journal of Internet Services and Applications*, *4*(1), 1–13.

Henver, A. & Chatterjee, S. (2010). *Design research in information systems.* Springer.

Hevner, A. R., March, S. T., Park, J. & Ram, S. (2004). Design science in information systems research. *MIS Quarterly*, *28*(1), 75–105.

Hussain, M. & Abdulsalam, H. (2011). Secaas: security as a service for cloud-based applications. In *Proceedings of the second kuwait conference on e-services and e-systems* (p. 8).

Hutle, M., Hansch, G. & Fitzgerald, W. (2015). D2. 2 threat and risk assessment methodology. *Tunneling and Underground Space Technology*, *24*(3), 269–277.

Hyndman, R. J. & Koehler, A. B. (2006). Another look at measures of forecast accuracy. *International journal of forecasting*, *22*(4), 679–688.

Ibrahim, A. S., Hamlyn-harris, J. H. & Grundy, J. (2010). Emerging security challenges of cloud virtual infrastructure.

Ingoldsby, T. R. (2010). Attack tree-based threat risk analysis. *Amenaza Technologies Limited*, 3–9.

Kalekar, P. S. (2004). Time series forecasting using holt-winters exponential smoothing. *Kanwal Rekhi School of Information Technology*, *4329008*, 1–13.

Kazim, M., Masood, R., Shibli, M. A. & Abbasi, A. G. (2013). Security aspects of virtualization in cloud computing. In *Computer information systems and industrial management* (pp. 229–240). Springer.

Khan, A. U. (2014). *Data confidentiality and risk management in cloud computing* (Unpublished doctoral dissertation). University of York.

Khan, A. U., Oriol, M., Kiran, M., Jiang, M. & Djemame, K. (2012). Security risks and their management in cloud computing. In *Cloud computing technology and science (cloudcom), 2012 ieee 4th international conference on* (pp. 121–128).

Khorshed, M. T., Ali, A. S. & Wasimi, S. A. (2012). A survey on gaps, threat remediation challenges and some thoughts for proactive attack detection in cloud computing. *Future Generation computer systems*, *28*(6), 833–851.

Kim, S. & Kim, H. (2016). A new metric of absolute percentage error for intermittent demand forecasts. *International Journal of Forecasting*, *32*(3), 669–679.

Kinsta. (2018). *What is apache web server? a basic look at what it is and how it works* [Internet web page]. Retrieved from `https://kinsta.com/knowledgebase/what-is-apache/`

Kiran, M. (2014). A methodology for cloud security risks management. In *Cloud computing* (pp. 75–104). Springer.

Kitchenham, B., Brereton, O. P., Budgen, D., Turner, M., Bailey, J. & Linkman, S. (2009). Systematic literature reviews in software engineering–a systematic literature review. *Information and software technology*, *51*(1), 7–15.

Kortchinsky, K. (2009). Cloudburst: A vmware guest to host escape story. *Black Hat USA*.

Kotzé, P., van der Merwe, A. & Gerber, A. (2015). Design science research as research approach in doctoral studies. In *21st americas conference on information systems (amcis)*.

Last, D. (2015). Using historical software vulnerability data to forecast future vulnerabilities. In *Resilience week (rws), 2015* (pp. 1–7).

Lawson, N. (2009). Side-channel attacks on cryptographic software. *Security & Privacy, IEEE*, *7*(6), 65–68.

Leitold, F. & Hadarics, K. (2012). Measuring security risk in the cloud-enabled enterprise. In *2012 7th international conference on malicious and unwanted software*.

Litchfield, A. & Shahzad, A. (2017). A systematic review of vulnerabilities in hypervisors and their detection. In *Proceedings of the 23rd americas conference on information systems*.

Liu, B., Shi, L., Cai, Z. & Li, M. (2012). Software vulnerability discovery techniques: A survey. In *2012 fourth international conference on multimedia information networking and security* (pp. 152–156).

Luo, S., Lin, Z., Chen, X., Yang, Z. & Chen, J. (2011). Virtualization security for cloud computing service. In *Cloud and service computing (csc), 2011 international conference on* (pp. 174–179).

Matis, J. (2017). *How physically secure is your data center?* [Internet web page]. Retrieved 12 December, 2017, from `https://www.digitalrealty.com/blog/how-physically-secure-is-your-data-center/?_ga=2.138811191.509472991.1515105941-1566856645.1515105941`

Mell, P., Kent, K. A. & Romanosky, S. (2007). *The common vulnerability scoring system (cvss) and its applicability to federal agency systems*. US Department of Commerce, National Institute of Standards and Technology.

Mell, P., Scarfone, K. & Romanosky, S. (2006). Common vulnerability scoring system. *IEEE Security & Privacy*, *4*(6).

Modi, C., Patel, D., Borisaniya, B., Patel, A. & Rajarajan, M. (2013). A survey on security issues and solutions at different layers of cloud computing. *The Journal of Supercomputing*, *63*(2), 561–592.

Newberne, J. H. (2006). *Holt-winters forecasting: A study of practical applications for healthcare managers* (Tech. Rep.). MIKE O'CALLAGHAN FEDERAL HOSPITAL LAS VEGAS NV.

Nightingale, A. (2009). A guide to systematic literature reviews. *Surgery (Oxford)*, *27*(9), 381–384.

NIST. (2017). *National vulnerability database (nvd)* [Internet web page]. Retrieved 31 December, 2017, from https://nvd.nist.gov/

Offermann, P., Levina, O., Schönherr, M. & Bub, U. (2009). Outline of a design science research process. In *Proceedings of the 4th international conference on design science research in information systems and technology* (p. 7).

Palmer, P. B. & O'connell, D. G. (2009). Regression analysis for prediction: understanding the process. *Cardiopulmonary physical therapy journal*, *20*(3), 23.

Pearce, M., Zeadally, S. & Hunt, R. (2013). Virtualization: Issues, security threats, and solutions. *ACM Computing Surveys (CSUR)*, *45*(2), 17.

Peffers, K., Tuunanen, T., Rothenberger, M. A. & Chatterjee, S. (2007). A design science research methodology for information systems research. *Journal of Management Information Systems*, *24*(3), 45–77.

Pék, G., Buttyán, L. & Bencsáth, B. (2013). A survey of security issues in hardware virtualization. *ACM Computing Surveys (CSUR)*, *45*(3), 40.

Perez-Botero, D., Szefer, J. & Lee, R. B. (2013). Characterizing hypervisor vulnerabilities in cloud computing servers. In *Proceedings of the 2013 international workshop on security in cloud computing* (pp. 3–10).

Rackspace. (2017). *Cloud security gets physical* [Internet web page]. Retrieved 12 December, 2017, from https://blog.rackspace.com/cloud-security-gets-physical

Ristenpart, T., Tromer, E., Shacham, H. & Savage, S. (2009). Hey, you, get off of my cloud: exploring information leakage in third-party compute clouds. In *Proceedings of the 16th acm conference on computer and communications security* (pp. 199–212).

Roumani, Y., Nwankpa, J. K. & Roumani, Y. F. (2015). Time series modeling of vulnerabilities. *Computers & Security*, *51*, 32–40.

Ruiz, M. D. M. L. & Pedraza, J. (2016). Privacy risks in cloud computing. In *Intelligent agents in data-intensive computing* (pp. 163–192). Springer.

Rutkowska, J. & Wojtczuk, R. (2008). Preventing and detecting xen hypervisor subversions. *Blackhat Briefings USA*.

Sabahi, F. (2011). Virtualization-level security in cloud computing. In *Communication software and networks (iccsn), 2011 ieee 3rd international conference on* (pp. 250–254).

Saini, V., Duan, Q. & Paruchuri, V. (2008). Threat modeling using attack trees. *Journal of Computing Sciences in Colleges*, *23*(4), 124–131.

Samani, R., Reavis, J. & Honan, B. (2014). *Csa guide to cloud computing: Implementing cloud privacy and security*. Syngress.

Saripalli, P. & Walters, B. (2010). Quirc: A quantitative impact and risk assessment framework for cloud security. In *2010 ieee 3rd international conference on cloud computing* (pp. 280–288).

Schneier, B. (1999). Attack trees. *Dr. Dobb's journal*, *24*(12), 21–29.

Schulze, H. (2015). *Cloud security survey report 2015* [Internet web page]. Retrieved from https://media.scmagazine.com/documents/114/cloud-security-spotlight-repor_28381.pdf

Schwartz, M. (2011). *Fired employee indicted for hacking gucci network* [Internet web page]. Retrieved from https://www.networkcomputing.com/networking/fired-employee-indicted-hacking-gucci-network

Shah, S. & Mehtre, B. M. (2015). An overview of vulnerability assessment and penetration testing techniques. *Journal of Computer Virology and Hacking Techniques*, *11*(1), 27–49.

Shin, Y., Meneely, A., Williams, L. & Osborne, J. A. (2011). Evaluating complexity, code churn, and developer activity metrics as indicators of software vulnerabilities. *IEEE Transactions on Software Engineering*, *37*(6), 772–787.

Shoaib, Y. & Das, O. (2014). Pouring cloud virtualization security inside out. *arXiv preprint arXiv:1411.3771*.

Sinanc, D. & Sagiroglu, S. (2013). A review on cloud security. In *Proceedings of the 6th international conference on security of information and networks* (pp. 321–325).

Space, S. (2017). *Os/linux distributions using apache* [Internet web page]. Retrieved from https://secure1.securityspace.com/s_survey/data/man.201705/apacheos.html

Squid-cache. (2009). *Squid-cahe* [Internet web page]. Retrieved from https://wiki.squid-cache.org/WhySquid

Srinivasan, M. K., Sarukesi, K., Rodrigues, P., Manoj, M. S. & Revathy, P. (2012). State-of-the-art cloud computing security taxonomies: a classification of security challenges in the present cloud computing environment. In *Proceedings of the international conference on advances in computing, communications and informatics* (pp. 470–476).

STAR, C. (2017). *Csa security, trust and assurance registry (star)* [Internet web page]. Retrieved from https://cloudsecurityalliance.org/star/#_overview

STAR Registry. (2017). *Star registry entries* [Internet web page]. Retrieved from https://cloudsecurityalliance.org/star/#_registry

Szefer, J., Jamkhedkar, P., Perez-Botero, D. & Lee, R. B. (2014). Cyber defenses for physical attacks and insider threats in cloud computing. In *Proceedings of the 9th acm symposium on information, computer and communications security* (pp. 519–524).

Tanimoto, S., Hiramoto, M., Iwashita, M., Sato, H. & Kanai, A. (2011). Risk management on the security problem in cloud computing. In *Computers, networks, systems and industrial engineering (cnsi), 2011 first acis/jnu international conference on* (pp. 147–152).

Tianfield, H. (2012). Security issues in cloud computing. In *Systems, man, and cybernetics (smc), 2012 ieee international conference on* (pp. 1082–1089).

Tirkes, G., Guray, C. & Celebi, N. (2017). Demand forecasting: A comparison between the holt-winters, trend analysis and decomposition models. *Tehnicki Vjesnik-Technical Gazette*, *24*(S2), 503–510.

Vaquero, L. M., Rodero-Merino, L. & Morán, D. (2011). Locking the sky: a survey on iaas cloud security. *Computing*, *91*(1), 93–118.

Walden, J., Stuckman, J. & Scandariato, R. (2014). Predicting vulnerable components: Software metrics vs text mining. In *Software reliability engineering (issre), 2014 ieee 25th international symposium on* (pp. 23–33).

Wang, H., Liu, F. & Liu, H. (2012). A method of the cloud computing security management risk assessment. *Advances in Computer Science and Engineering*, 609–618.

Wojtczuk, R. (2008). Subverting the xen hypervisor. *Black Hat USA*, *2008*.

Wooley, P. S. (2011). *Identifying cloud computing security risks* (Unpublished doctoral dissertation). University of Oregon.

You, P., Peng, Y., Liu, W. & Xue, S. (2012). Security issues and solutions in cloud computing. In *Distributed computing systems workshops (icdcsw), 2012 32nd international conference on* (pp. 573–577).

Zhang, S., Caragea, D. & Ou, X. (2011). An empirical study on using the national vulnerability database to predict software vulnerabilities. In *International conference on database and expert systems applications* (pp. 217–231).

Zhang, T. & Lee, R. B. (2014). New models of cache architectures characterizing information leakage from cache side channels. In *Proceedings of the 30th annual computer security applications conference* (pp. 96–105).

Zhang, X., Wuwong, N., Li, H. & Zhang, X. (2010). Information security risk management framework for the cloud computing environments. In *Computer and information technology (cit), 2010 ieee 10th international conference on* (pp. 1328–1334).

Zhang, Y., Juels, A., Reiter, M. K. & Ristenpart, T. (2012). Cross-vm side channels and their use to extract private keys. In *Proceedings of the 2012 acm conference on computer and communications security* (pp. 305–316).

Appendix A

Vulnerability Types and Threat Actors Properties

A.1 Vulnerability Types

Reported and unknown vulnerabilities are the two vulnerability categories which are mentioned as follows:

Known or Reported Vulnerabilities Known vulnerabilities are those which are reported to vulnerability databases. The security community discovers and analyses the vulnerabilities and shares the vulnerability information over the internet through databases such as NVD, CVE, and SANS. This online vulnerability information helps the community to identify and truly understand the attack type.

Unknown Vulnerabilities Unknown vulnerabilities are those which are not discovered yet. In another case, the vulnerabilities which are discovered but not reported to the vulnerability databases are also referred as unknown vulnerabilities.

A.1.1 Threat Actor Capabilities

This section details the capabilities of TAs to exploit a vulnerability.

Formidable A TA with formidable capability can take several months to exploit a target system. A TA can use specially developed tools and equipment for some specific targets. Moreover, TAs with such capability have expert level knowledge about computers and security.

Significant A TA with significant capability can spend few months to exploit a target system. A TA can use publicly available tools and large equipment for some specific targets. Moreover, TAs with such capability have professional level knowledge about computers and security.

Limited A TA with limited capability can spend few weeks or days to exploit a target system. A TA can use publicly available tools with some equipment. Moreover, a TA with limited capability is a trained computer or a network user.

Little A TA with little capability can spend few hours or days to exploit a target system. A TA can try different hacking tools with some equipment. Moreover, a TA with little capability is an average computer or a network user.

Very Little A TA with minimal capability can spend few hours to exploit a target system. A TA can use the equipment already connected to the target system such as plug and play, and removable devices.

A.1.2 Threat Actor Motivation Levels

This section details the motivation levels of TAs to exploit a vulnerability.

Very High (Focused) The primary objective of a focused TA is to exploit the target system by all means necessary.

High (Committed) A TA who is highly committed, will try to exploit the target system

on a frequent or constant basis. However, they may require formal clearance and could be deterred.

Medium (Interested) A TA who is interested, will try to exploit the target system if an opportunity exists, and the attack takes less effort. However, they may require taking formal clearance and could be deterred.

Low (Curious) A TA who is curious, will investigate the target system casually and attack if there is any weakness. However, TA may require formal clearance and could be deterred.

Very Low (Indifferent) A TA with indifferent motivation level will not attack the system and does not pose any risk.

A.2 Qualitative Threat Likelihood Level from Threat Actors

Threat likelihood levels can be determined as Severe, Substantial, Moderate, Low, and Negligible.

Severe A severe threat level can be expected when a TA behaves very severely, ignores all the security policies, and is an unreliable person. This threat level can be observed in organisations which do not meet IA Maturity Model Level 1.

Substantial The level of threat will be substantial if a TA does not behave well, sometimes ignores all the security policies and procedures, and occasionally is an unreliable person. This threat level can be observed in organisations which do not meet IA Maturity Model Level 1.

Moderate The threat level will be moderate if a TA is a reliable and trustworthy person, and behaves well. They also follow security policies and procedures and do not try to bypass security controls to perform job-related tasks. This threat

level can be observed in organisations which do not meet IA Maturity Model Level 2.

Low The threat level will be low if a TA behaves exceptionally well, is a reliable and trustworthy person. They strictly follow security policies and have complete awareness of consequences of bypassing security controls. This threat level can be observed in organisations which do not meet IA Maturity Model Level 3.

Negligible Organisations can ignore this threat level as it does not pose any risk.

Appendix B

Common Vulnerability Scoring Systems

B.1 Common Vulnerability Scoring Systems

This appendix provides the details of Common Vulnerability Scoring Systems (CVSS) used score the vulnerabilities to determine their impact ratings. Section B.1.1 provides the equation used to calculate Base score. Section B.1.2 presents the rubric used to calculate the score. Section B.1.3 and B.1.4 covers the metrics that determine the sub score to finally determine the Base score to determine the impact of vulnerability exploitation.

B.1.1 Base Scoring Equation

The Impact and Exploitability sub score equations are combined to calculate the Base Score. The Base score is calculated as,

If (sub score of impact =< 0) 0 else,

Scope Unchanged Round up (Minimum [(Impact + Exploitablity), 10])^3

Scope Changed Round up (Minimum [1.08 × (Impact + Exploitablity), 10])

and the Impact sub score (ISC) is defined as,

Scope Unchanged $6.42 \times ISC_{Base}$

Scope Changed $7.52 \times [ISC_{Base} - 0.029] - 3.25 \times [ISC_{Base} - 0.02]^{15}$

Where,

$ISC_{Base} = 1 - [(1 - Impact_{Conf}) \times (1 - Impact_{Integ}) \times (1 - Impact_{Avail})]$

And the Exploitability sub score is,

8.22 × Attack Vector × Attack Complexity × Privilege Required × User Interaction

Base Scoring Equation

B.1.2 Scoring Rubrics

CVSS 3.0 provide guidelines to score vulnerabilities. To figure it out that when to score the impact of the vulnerability, constraint impacts should be considered to an understandable final impact which an attacker is going to achieve. The attacker's ability level to cause impact on an asset should be supported by the exploitability sub score as a minimum. But the vulnerability's description should also be considered to include details to determine the impact.

The below Figures from 4.6 to 4.13 provides the CVSS 3.0 scoring rubrics to score base group metrics as provided in above Table 4.18.

Attack Vector

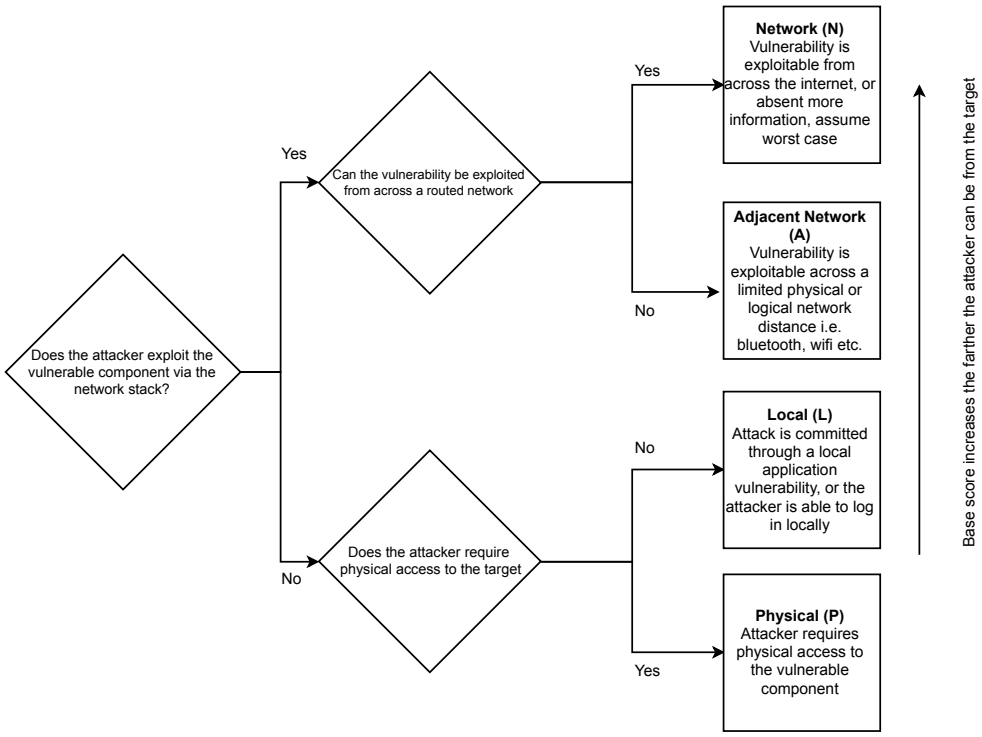


Figure B.1: Scoring Rubric for Attack Vector Metric

Attack Complexity

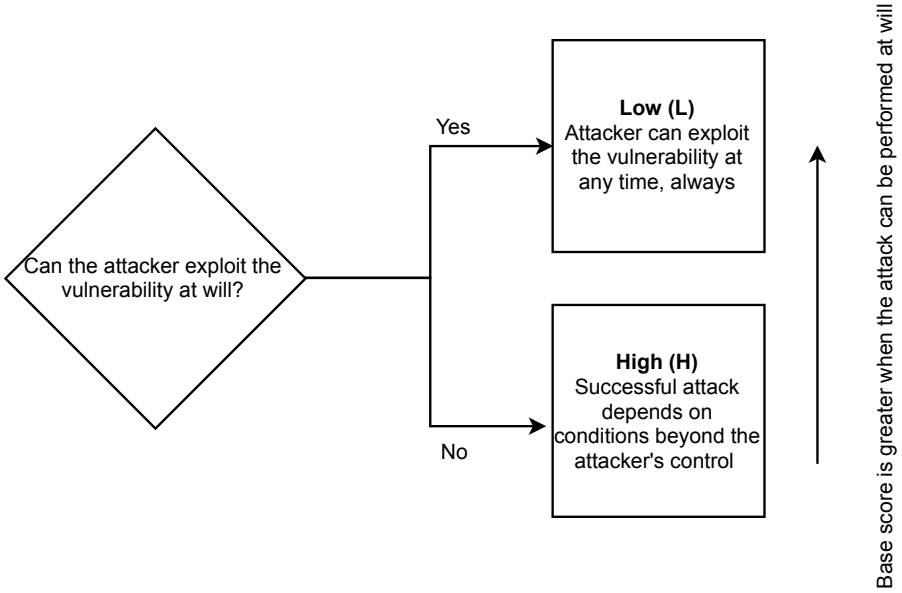


Figure B.2: Scoring Rubric for Attack Complexity Metric

Privileges Required

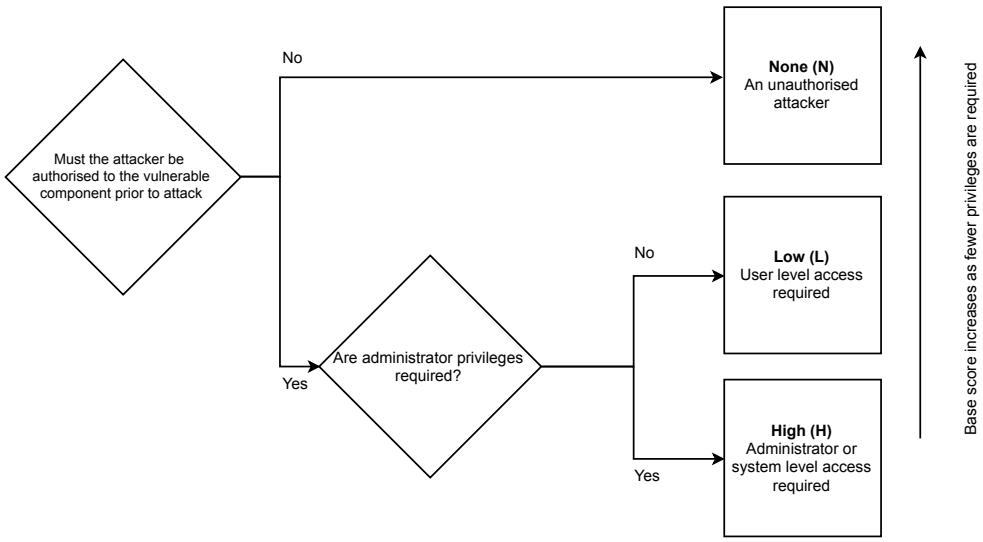


Figure B.3: Scoring Rubric for Privileges Required Metric

User Interaction

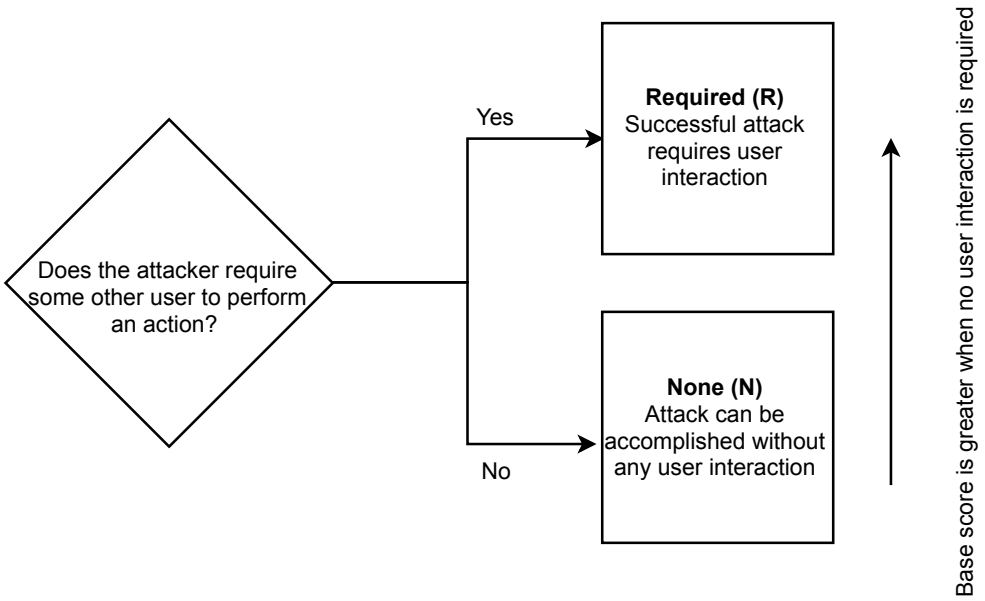


Figure B.4: Scoring Rubric for User Interaction Metric

Scope

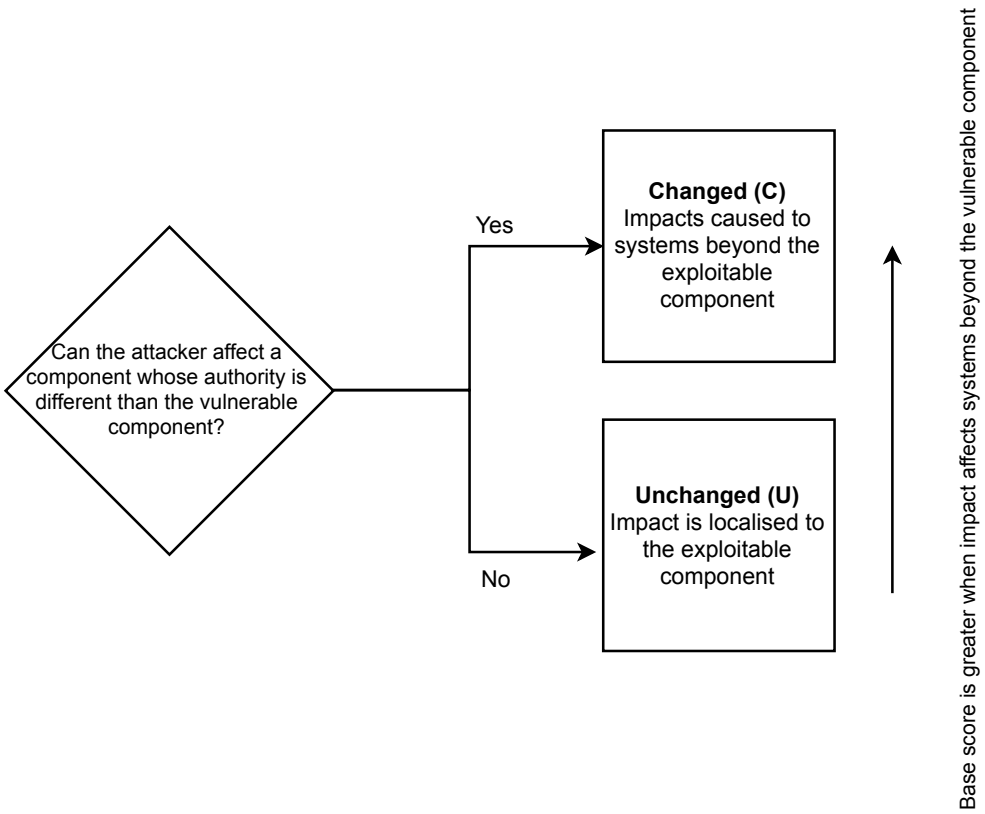


Figure B.5: Scoring Rubric for Scope Metric

Confidentiality Impact

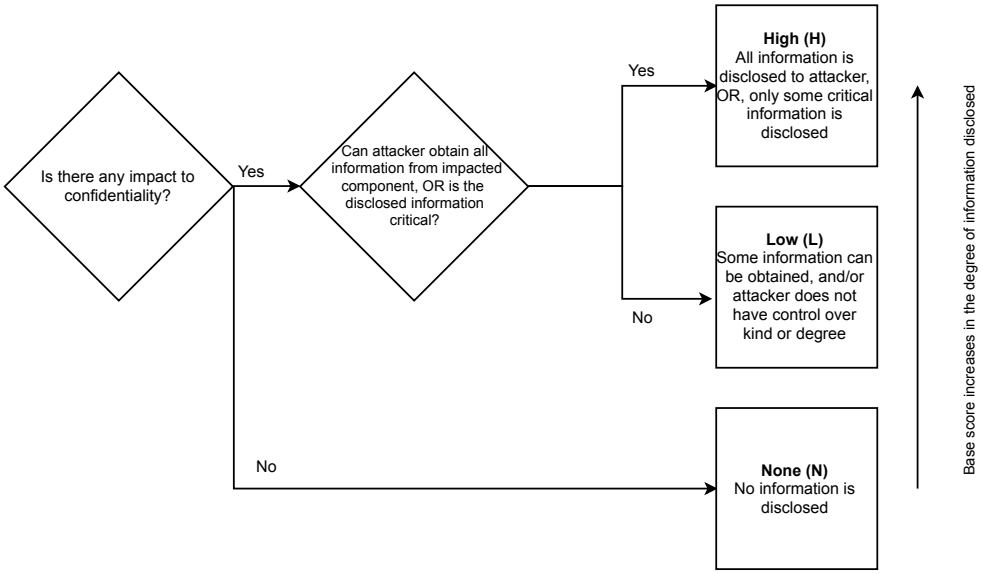


Figure B.6: Scoring Rubric for Confidentiality Impact Metric

Integrity Impact

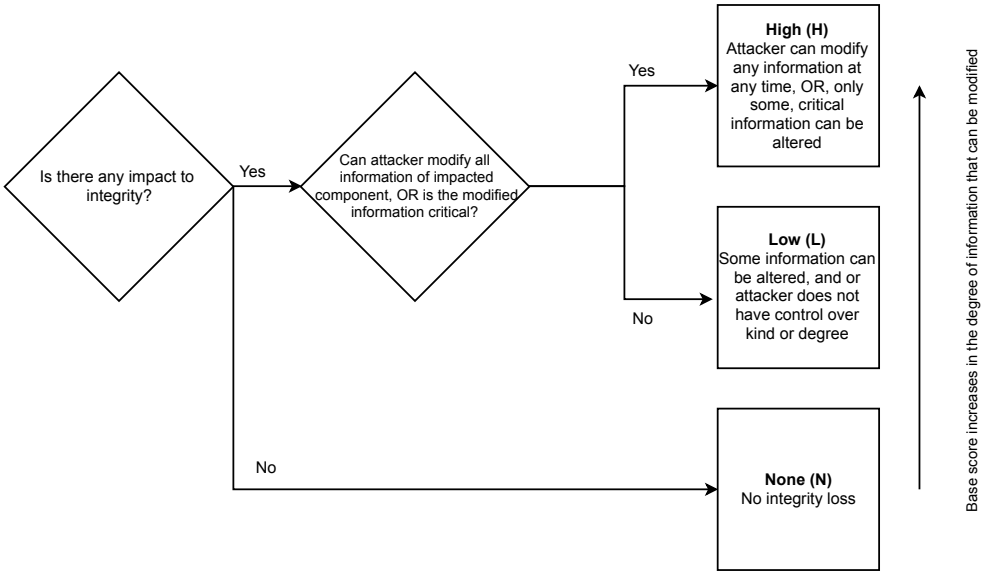


Figure B.7: Scoring Rubric for Integrity Impact Metric

Availability Impact

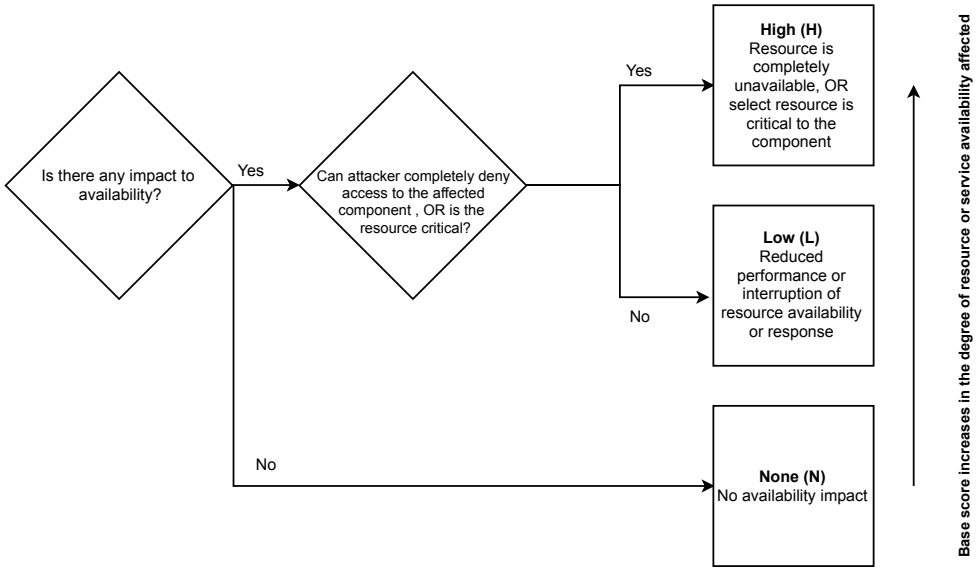


Figure B.8: Scoring Rubric for Availability Impact Metric

B.1.3 Exploitability Metrics

The Exploitability metrics provide characteristics of a vulnerability and formally refers to a vulnerable component. These metrics are categorised as Attack Vector (AV), Attack Complexity (AC), Privileges Required (PR), User Interaction (UI) and Scope (S). All these metrics should be scored relative to the vulnerable component and provide the qualities of the exploited vulnerability.

Attack Vector (AV)

AV provides the context of vulnerability exploitation. This metric value is based on the attacker's way to reach to the target system through Network (N), Adjacent Network (A), Local (L) and Physical (P). The metric value will result in more score if the attacker is exploiting the target system across the internet as compared to exploiting it by requiring physical access to it. Table 3.2 provides the AV metric values and their description.

Table B.1: Attack Vector Metric Values

Metric Value	**Description**
Network (N)	This metric value refers to the vulnerable system which can be compromised by exploiting a network vulnerability. Such a vulnerability is considered as "remotely exploitable" and the vulnerable component is one or more hops away from the attacker.
Adjacent Network (A)	This metric value means that the vulnerable component is bound to network layer but the attacker should have network access using the same physical network device and must be within the same subnet as the vulnerable component. This attack can happen when both the attacker and target are in the same broadcast domain.
Local (L)	This metric value means that the vulnerable component is not bound to the network layer and the attacker uses read/write/execute capabilities. The attack can happen in two ways, either the attacker is logged in locally and have access to the system to exploit the vulnerability or attacker require end user's interaction to execute the required commands.
Physical (P)	This metric value means that the attacker is required to physically access and manipulate the vulnerable component.

Attack Complexity (AC)

This metric provides the conditions which are beyond the control of the attacker. For example, higher this metric value is, higher is the complexity of the attack. The conditions offered by this metric forces the attacker to learn more about the target system, configuration settings and policies. The detail of AC metric values and their description is provided in Table 3.3.

Table B.2: Attack Complexity Metric Values

Metric Value	Description
Low (L)	This metric value allows the attacker to achieve repeatable success again the vulnerable component. The attacker does not need to put too many efforts to learn about the target system and its configurations.
High (H)	This metric value forces the attacker to learn more about the vulnerable component and its configuration settings etc. The attacker needs to spend more time and efforts in preparation and exploitation of the vulnerable component. For example, the attacker must put extra efforts to learn about target system's configuration settings, shared secrets and exploit mitigation techniques. The attacker may also need to launch a man-in-the middle attack to learn or modify network communications.

Privileges Required (PR)

This metric value is the level of privileges an attacker must possess to exploit a vulnerable component. For example, higher the value of this metric is, lower the privileges an attacker require. Means, if the attacker has no or low privileges, then it would be difficult for him to exploit the vulnerability as he needs to obtain access privileges. However, if the attacker has a high level of privileges, then it would be easy for him to exploit the vulnerability. Table 3.4 provides PR metric values and their relevant description.

Table B.3: Privileges Required Metric Values

Metric Value	Description
None (N)	The attacker has no privileges and not authorised to access settings or configuration information to exploit the vulnerability.

continued ...

. . . continued

Metric Value	Description
Low (L)	This metric value means that the attacker has some level of privileges and authorised to access some file and configuration settings. Therefore, this metric makes the attacker capable to exploit a vulnerability resulting in an impact to non-sensitive resources.
High (H)	The attacker has high privileges to access files and configuration settings. Making the attacker very capable to exploit the vulnerable system and cause damage to component-wide settings and files belongs to other users.

User Interaction (UI)

This metric value determines the level of participation from the end user to launch a successful attack and exploit the vulnerable component. This metric value would be high if no interaction is required from the end user. The detail of UI metric values and relevant description is provided in the Table 3.5.

Table B.4: User Interaction Metric Values

Metric Value	Description
None (N)	No interaction or participation from the end user is required to exploit the vulnerable component.
Required (R)	The end user must participate and take some actions for the attacker to exploit the vulnerable system. For example, the vulnerability exploitation is only possible at the time when administrator is installing an application on the target system.

Scope (S)

This metric means when the attacker exploits a vulnerability which is under one authorisation scope can affect resources managed by another authorisation scope. For example, in the context of virtual environment, the change of scope occurs when an

attacker runs malicious code on guest Virtual Machine (VM) to bypass the hypervisor layer and access or delete some files of the host OS (Dom0 in case of Xen hypervisor). So, there are two scopes involved in this example. One scope that is, authorises and controls VM and its user's privileges and second scope that is, authorises and controls host OS privileges. This metric value would be high if the scope change has occurred. Table 3.6 provides the values and relevant description for S metric.

Table B.5: Scope

Metric Value	**Description**
Unchanged (U)	This metric value means that the exploited vulnerability can only affect the resources under the same authorisation scope. Exploitation does not affect resources authorised by another scope.
Changed (C)	This metric value means that the exploited vulnerability can affect resources which are under another authorisation scope. Exploitation does affect resources authorised by another scope.

B.1.4 Impact Metrics

The impact metrics provide the properties of the exploited component. It refers to the component that is affected by a successful attack and reflects Confidentiality, Integrity, and Availability (CIA) impact to the exploited component. However, if the exploitation resulted in a change of Scope, then this metric should reflect the CIA impact to the exploited component or the impacted component under another authorisation scope. If the exploitation resulted in no change of Scope, then this metric should only reflect the CIA impact to the exploited component.

Confidentiality (C)

This metric value provides the impact to the confidentially of the data or information managed by the vulnerable component. If the value of this metric is high, then the loss of confidentially is also very high. Table 3.7 provides the C metric values along with the relevant description.

Table B.6: Confidentiality Impact Metric Values

Metric Value	**Description**
High (H)	This metric value means a complete violation of the confidentiality and disclosure of all the information such as administrator's password and shared encryption keys to the attacker.
Low (L)	This metric value refers to the loss of confidentiality but, to some extent. A successful exploitation results in access to some information but not complete control of the information and does not directly affect the target system or its users.
None (N)	This metric value refers to no loss of confidentiality.

Integrity (I)

This metric value provides the impact to the integrity of the information after a successful exploitation. Integrity refers to the reliability and exactness of the information. If the value of this metric is high, then the loss of integrity is also very high. Table 3.8 provides the I metric values along with the relevant description.

Table B.7: Integrity Impact Metric Values

Metric Value	Description
High (H)	This metric value is a complete violation of the integrity and loss of protection of all the information. For example, unauthorised modification of all the files protected by the vulnerable system. High metric value refers to the direct impact of the loss of integrity to the users of the systems.
Low (L)	This metric value refers to the loss of integrity but, to some extent. A successful exploitation results in a unauthorised modification to some information but not complete control of the information. Low metric value refers to the level of unauthorised modification that does not have a direct impact on the vulnerable system or its users.
None (N)	This metric value refers to no loss of integrity.

Availability (A)

This metric value provides the impact to the availability of the information to the legitimate users after a successful attack. Availability refers to the accessibility of the required information by the users of the target system. If the value of this metric is high, then the consequences are also very high. The metric values and their description for A metric are provided in the Table 3.9.

Table B.8: Availability Impact Metric Values

Metric Value	Description
High (H)	This metric value means complete loss of the availability of information to legitimate users of the target system. A successful exploitation of the target system results in loss of some availability, but High metric value refers to the direct impact of the loss of availability to the users of the vulnerable system.

continued ...

. . . continued

Metric Value	Description
Low (L)	This metric value refers to the loss of integrity but, to some extent. A successful exploitation results in a unauthorised modification to some information but not complete control of the information. Low metric value refers to the level of unauthorised modification that does not have a direct impact on the vulnerable system or it's users.
None (N)	This metric value refers to no loss of availability.