UNIVERSITY OF LEEDS

Improving Multi-Tenancy Security by Controlling Resource Allocation in IaaS Public Clouds

Hussain Lafi H AlJahdali

Submitted in accordance with the requirements for the degree of

Doctor of Philosophy

The University of Leeds

School of Computing

December, 2017

Intellectual Property and Publication Statements

The candidate confirms that the work submitted is his/her own, except where work which has formed part of jointly-authored publications has been included. The contribution of the candidate and the other authors to this work has been explicitly indicated below. The candidate confirms that appropriate credit has been given within the thesis where reference has been made to the work of others.

Aljahdali, H., Townend, P. and Xu, J., 2013, March. Enhancing Multi-tenancy Security in the Cloud IaaS Model over Public Deployment. In Service Oriented System Engineering (SOSE), 2013 IEEE 7th International Symposium on (pp. 385-390). IEEE. In this paper, the problem was positioned up and the system model was presented. The work entirely carried out by the candidate (i.e. AlJahdali). The paper was reviewed and improved by Jie Xu and Paul Townend.

AlJahdali, H., Albatli, A., Garraghan, P., Townend, P., Lau, L. and Xu, J., 2014, April. Multi-tenancy in Cloud computing. In Service Oriented System Engineering (SOSE), 2014 IEEE 8th International Symposium on (pp. 344-351). IEEE. In this paper, a comprehensive understanding of Multi-Tenancy was presented, its origins, benefits and its risks. The system model was described and illustrated. The threat model was presented and discussed and the approach of how to tackle Multi-Tenancy was demonstrated. Finally, the results of attack model were presented and discussed. The data catalogue used to generate the results was produced by Peter Garraghan. The SQL codes and enquires were developed and reviewed by the candidate (i.e. Aljahadli) and Abdul-Aziz Albatli equally. The paper was reviewed and improved by Jie Xu, Paul Townend and Lydia Lau. The content of this paper contributed to Chapter 2, Chapter 3, Chapter 4 and Chapter 5.

Declaration

Some parts of the work presented in this thesis have previously appeared in the following papers:

Aljahdali, H., Townend, P. and Xu, J., 2013, March. Enhancing Multi-tenancy Security in the Cloud IaaS Model over Public Deployment. In Service Oriented System Engineering (SOSE), 2013 IEEE 7th International Symposium on (pp. 385-390). IEEE

AlJahdali, H., Albatli, A., Garraghan, P., Townend, P., Lau, L. and Xu, J., 2014, April. Multi-tenancy in Cloud computing. In Service Oriented System Engineering (SOSE), 2014 IEEE 8th International Symposium on (pp. 344-351). IEEE

Acknowledgments

I would like to take this opportunity to express my profound gratitude to a number of people who I believe have impacted me positively during the course of this journey.

The first acknowledgment is submitted to whom I became the man I am today, to my parents Lafi, Qamra and Fayza. To my father and my two mothers, I have been lucky to have two mothers, one who gave me birth and another who raised me. If I would write books about them, I would never express a tip of my deep feelings to them.

My second acknowledgment is to my waif Ghaliah, my son Lafi and my daughter Sarah. To my little family who always believed in me. To whom without, this journey became dark and lonely.

My third acknowledgment goes to my brothers, sisters and friends. Shujaa, Sultan, Abdulrahmn, Khalied, Mohamed, Wadha, Wafa, Ghader, Danah, Fahdah, Rwaished, Batil, Bunder, Osamah, Amjad, Hijab and Almuhamadi Khalid. And a special acknowledgment to Mohammed Bin Aied Alhafi, who supported me in different aspects.

My final acknowledgment is submitted to my supervisors, Prof Jie Xu and Dr Paul Townend. I would like to express my gratitude and sincere regard for their exemplary guidance, mentoring, and constant encouragement throughout the course of this thesis. My research skills have greatly improved because of their guidance.

Also, I would like take the opportunity to express my gratitude to the Distributed Systems and Services group as they ALL supported me during this journey.

Abstract

In a world where the requirements of computing systems are rapidly changing, the need for a dynamic, yet a cost-effective system becomes urgent. Besides, the need of dynamically scale-up and scale-down, mobility, and reduce both individuals and enterprises share costs and expenses. Thus, such needs and more are fulfilled by Cloud Computing.

Mostly, Cloud Computing is promoted as a new paradigm which offers a set of benefits for both providers and consumers. For service providers, it gives an ease of management, reduced maintenance and operational costs, better utilisation of resources, and extra profit. For customers, it offers on demand resources, mobility and effective scale-up, and scale-down.

Despite the benefits provided by Cloud Computing, some challenges are seen. For instance, security is a major challenge. Indeed, security is an obstacle to promoting public Clouds for large consumers (i.e. governments and enterprises). Therefore, more research on safety issues in Cloud Computing is required. For instance, the issues of access control could be found in traditional systems; however, Multi-Tenancy could be considered a unique issue related to Cloud Computing.

Nonetheless, the research shows for the first time the size of Multi-Tenancy as a security concern. Specifically, Multi-Tenancy could increase the probability of being under attack by 100%. Moreover, to enhance the safety of Multi-Tenancy, availability could compromise as well the Cloud provider’s profit. Although Multi-Tenancy is a complex issue due to its benefits to Cloud Computing, we develop a scheme to enhance the security of Multi-Tenancy while preserving its benefits.

Table of Contents

List of Tables

List of Figures

Chapter 1 Introduction

1.1 Research Motivation

In a world where the requirements of computing systems are rapidly changing, the need for a dynamic, yet a cost-effective system becomes urgent. The need of dynamically scale-up and/or scale-down, mobility and reduce costs and expenses are shared by both individuals and enterprises. Such needs and more are fulfilled by Cloud Computing.

Cloud Computing is promoted as a new paradigm which offers a set of benefits for both providers and consumers. For service providers, it offers an easy of management, reduced maintenance and operational costs, better utilisation of resources and extra profit. For consumers, it offers on demand resources, mobility and dynamic scale-up and scale-down.

However, along with the benefits offered by Cloud Computing come a number of challenges. One of the most important challenges is security; security becomes an obstacle to promote public Clouds for giant consumers (i.e. governments and enterprises). Such situation opens the opportunities for research as security issues in Cloud Computing vary based on its nature. For example, the issues of access control could be found in traditional systems. Whereas, the issues of Multi-Tenancy could be considered a unique issue related to Cloud Computing.

Indeed, multi-tenancy is the major drive for this study. Ideally, in a multi-tenant environment, the risks for integrity and confidentiality violations are present. However, for effective implementation of the multi-tenancy model, professionals need to understand the attack vectors and surfaces. Indeed, we will be keen to investigate the novel way of approaching multi-tenancy. Moreover, the study will identify an attack model using a Google data set.

1.2 Research Context

In controlled environments such as traditional IT systems, the implementations of security controls and measures are considered relatively easy. The basic strategy of Information Security is to separate and control where the idea is to define premises and implement security controls on the borders. Although it is an effective strategy, it fails with complex and emerging systems such as Cloud Computing [1]. With complex environments and systems such as Cloud Computing, the clear definition of premises becomes a challenge (ibid). This leads to the acceptance of the risks aroused by different vulnerabilities in Cloud Computing [2].

Furthermore, the nature of vulnerability increases the complexity of dealing with it. Usually, vulnerabilities open the doors for risks which make the decision of eliminating the vulnerability a wise decision and the first choice. Unfortunately, some of the Cloud Computing vulnerabilities are the Cloud competency such as Multi-Tenancy.

The elimination of Multi-Tenancy will lead to the elimination of some, if not most, Cloud Computing benefits such as overprovisioning which contributes directly toward the increase of Cloud providers' profits. On the other hand, the acceptance of the risks associated with Multi-Tenancy, for example, will hinder the promotion of public Clouds as enterprises and governments deploy private Clouds. The decision of deploying private Clouds rather than joining public Clouds is driven by the security issues associated with Cloud Computing public deployment.

In order to handle and propose a mitigation strategy for security issues in the Cloud Computing, an in depth understanding of each security issue must be reached. Along with an investigation of a real public Cloud system in order to quantify and measure the security problem and its circumstances. Such empirical investigation will give the chance to assess the size of the problem and the best strategy to deal with it. This will positively impact the process of decision making in terms of deploying a private Cloud or joining a public Cloud.

This research project offers an in-depth understanding of the Multi-Tenancy as a Cloud Computing feature, highlighting its origin, the benefits associated with it and how to mitigate it. Moreover, this project will quantify Multi-Tenancy and identify the factors correlated with it based on an empirical investigation of a public Cloud. Both the in-depth understanding of Multi-Tenancy and the empirical investigation will contribute toward an effective solution to enhance the security of Multi-Tenancy on public Clouds.

1.3 The Problem Statement

We aim at enhancing the security of Multi-Tenancy in Infrastructure-as-a-Service (IaaS) in the public Clouds. Indeed, it is important since the risks associated with Multi-Tenancy are driving the decision of adopting a private Cloud to preserve security. On the other hand, the adoption of public Clouds reduces the capital investment and operational costs. Precisely, we investigate a large-scale Cloud to quantify the security risks Multi-Tenancy through understanding the correlated factors. Indeed, the activity is vital since it clarifies the actual impact of Multi-Tenancy and contributes in proposing an effective solution to enhance Cloud security. Moreover, through findings of the research, a comprehensive knowledge about Multi-Tenancy and quantification of the risks will positively impact the Cloud markets.

1.3.1 The Objectives of the Project

Upon the completion of the project, we will have accomplished the following:

I. *Formalise an attack model while taking advantage of Multi-Tenancy in Cloud Computing*. There are different security attacks launched on Clouds utilising different vulnerabilities, yet no explicit attack highlights the attack sequence where Multi-Tenancy is shown as the main vulnerability. Therefore, this project formalises a specific attack model which highlights how Multi-Tenancy could be exploited. Accordingly, it will define the research approach and present the percentage of a successful attack that uses Multi-Tenancy as a vulnerability.

II. *Quantify the scale of Multi-Tenancy*. Ideally, this project is considered to be the first to quantify Multi-Tenancy in Cloud Computing. Consequently, it is vital since the majority of review papers in Cloud Computing highlight Multi-Tenancy as a vulnerability, yet no one shows how much is the scale of Multi-Tenancy.

III. *Develop a scheme to mitigate Multi-Tenancy from a security perspective*. The current proposed strategies are either to eliminate Multi-Tenancy or accept it. Both strategies do not provide a balance between security requirements and cost requirements; as they treat security as binary attribute either to preserve it or loss it. Such strategies affect the Cloud providers and consumers negatively. This research project proposes a new direction to tackle the problem where it mitigates the risks of Multi-Tenancy to reach an acceptable balance between security and other attributes such as cost. After understanding Multi-Tenancy in depth, in terms of the impact scale, and its environmental set up; an effective solution will be achievable.

IV. *Quantify the quality impact of Clouds after enhancing the security of Multi-Tenancy.* In practice, every solution must have trade-offs; hence, this research project security in Multi-Tenancy is the major concern. Therefore, we will capture the trade-offs, different behaviours, and interactions from the customer to the placement of virtual machines (VM) in order to reduce the impact of securing Multi-Tenancy.

1.4 Research Methodology

As discussed earlier, Multi-Tenancy seems to be an issue in Cloud Computing. However, little empirical attention has been paid to significantly identify and quantify the scale of Multi-Tenancy. Hence, there is a theoretical necessity to understand and specify the nature of Multi-Tenancy in relation to security issues before embarking or conducting an empirical investigation. Therefore, based on the stated objectives, the research methodology is divided and broken down into four components. Each component stems from at least one objective:

- First, the literature review of the scientific journals, conferences and data bases is conducted in the area of Cloud Computing security and Multi-Tenancy in Cloud Computing. Specifically, the review will be vital in understanding Multi-Tenancy and identifying the origin, pros, and characteristics. This procedure will reveal the theoretical nature of Multi-Tenancy in designing the threat model and mitigating the risks associated with Multi-Tenancy.

- Second, designing an attack model using Markov chain to define the best approach in securing Multi-Tenancy. This procedure aims to show a real attack model taking advantage of Multi-Tenancy as well as to reveal the best possible ways of counter-measures.
- Third, using the descriptive statistics to get accurate measures of Multi-Tenancy within physical machines (PMs). Ideally, the correlation tests will be used to identify the significance of the relations between different attributes that might affect Multi-Tenancy. Moreover, spatial analysis will be used to capture the different behaviours and interactions related to Multi-Tenancy within the Cloud environment.
- Fourth, designing a scheme to enhance the security of Multi-Tenancy using the Chines Wall Security Policy (CWSP) design principles. This procedure will highlight the real enhancement of Multi-Tenancy security. In addition, it will highlight the quality impact on Clouds after enhancing the security.

Clearly, each procedure is aligned with the outlined objectives. This constructive alignment between the objectives and the empirical procedures is a research necessity to highlight and identify causes of the problem as well as to specify requirements of the solution and promising directions.

1.5 Major Contributions

The major contributions generated out of this research project are different in nature ranging from identifying and quantifying the process to providing a proposed solution. These seemingly similar yet different (in methodology) contributions come from the outlined objectives and the empirical investigation. They are as follows:

I. *Investigation, analysis, and quantification of Multi-Tenancy in large scale Cloud*: focusing on a real Cloud environment to be analysed expands the understanding of the problem. In essence, the investigation was done on different aspects, for instance, the Multi-Tenancy within PMs. Likewise, the analysis of number of PMs, the number of VMs, the duration of VMs life, the start and end time of each VM, and the number of users in relation to number of VMs gives us a measurable scale of the Multi-Tenancy. In addition, the correlation analysis between different attributes such as number of PMs, number of VMs, duration of VMs, and different platforms highlight significantly the effect of Multi-Tenancy. Finally, the capture of different workloads behaviours will reveal their impact on Multi-Tenancy either on maximising or minimising the impact of Multi-Tenancy on the Cloud environment.

II. *The design of an attack model to exploit Multi-Tenancy: while utilising Markov chain to design the attack*, two goals were achieved. First, the highlight of the main role of Multi-Tenancy on an attack sequence and the dependants stages before and after being a Multi-Tenant. Second, the possibility of measuring the likelihood of being under an attack which takes advantage of Multi-Tenancy.

III. *The development of a scheme to mitigate Multi-Tenancy*: based on CWSP, a proposed solution was designed to effectively enhance the security of Multi-Tenancy on IaaS public Clouds.

IV. *The evaluation of the quality impact of the scheme to enhance the security of Multi-Tenancy:* a mixed approach was used (mathematical proof and experimental evaluation). Ideally, the use of the large-scale Cloud data set to measure against were utilised and the mathematical proof of the scheme was conducted to cover different aspects of the solution.

Crucially, the project has four main contributions; each contribution is in alignment with the overall objective of considering the already known securing issues of Cloud

Computing. This project attempts to provide answers to the emerging threats, and specify new vulnerabilities related to Cloud Computing architecture.

1.6 Thesis Organisation

The thesis is comprised of seven chapters where this is the first of them. Ideally, before the beginning of every chapter, the thesis's gives an introduction of the topic to be covered. Later, every chapter ends with a summary of what has been discussed in the chapter.

Chapter 2 covers all the literature review needed to establish a comprehensive understanding of the subjects of the research.

Chapter 3 presents the analysis of Google data as a case study.

Chapter 4 illustrates the threats associated with Multi-Tenancy and its attack model.

Chapter 5 describes the approach to enhance Multi-Tenancy security in Clouds.

Chapter 6 will be the evaluation and discussion of the work.

Chapter 7 will highlight the conclusions, summaries the findings and state the future directions of the research.

Chapter 2 Security in Cloud Computing

This chapter will prepare the stage to position up the research problem where in-depth knowledge of Cloud Computing and its important components will be given and illustrated. After that, the security issues in Cloud Computing will be discussed and in-depth knowledge of attack theory is given. Moreover, more details and knowledge of Multi-tenancy will be presented and described where its origins, importance and different arguments about it will be focused on. Then, the different security domains are highlighted where the research scope is identified. Finally, a focus on the security attacks and its mechanisms is discussed. Ideally, the literature review will help in setting up the basis for the analysis of the multi-tenancy, the gaps in security, and the improvements to the security model.

2.1 Virtualisation

Cloud Computing is seen as an emerging technology, and one of the vital technologies to enable Cloud Computing is virtualisation. Virtualisation is the technology to emulate physical computing components (i.e. CPU, memory and network adapters) to offer a physical computing resource in a virtual fashion to overcome the limitations of the actual resource. One of the major benefits of virtualisation is the isolation of hardware (HW) failure from the software (SW) failure, such feature has a positive impact on the availability of the overall computing system. With the evolution of the virtualisation technology, it becomes a vital technology to equip Cloud Computing with competitive features [3].

Virtualisation as a concept has been around since 1960s in some form as in IBM mainframes and with the time it expanded to cover CPU, memory, networking and storage [4]. Nowadays virtualisation is mature enough in both levels: market wise and technically where more products and companies are existing and competing [3], [4].

The two major components of virtualisation are the Virtual Machine Monitor (VMM) – aka hypervisor – and the Virtual Machine (VM). The purpose of the VMM is to isolate the guest Operating System (OS) from the underlying HW and give the opportunity to run more than one VM in the same physical machine (PM) [5]. Whereas the VM is the actual entity where the preferred OS is installed in and interactions take place. The VMM is usually owned and managed by the system administration – Cloud provider in the case of Cloud Computing –, whereas the VM is owned and managed by the customer.

2.2 Cloud Computing

This section will give the backgrounds for Cloud Computing where section 2.2.1 will define the Cloud Computing and its important components. And section 2.2.2 will list and describe the Cloud characteristics. Where section 2.2.3 and 2.2.4 will demonstrate the Cloud service models and deployments model respectively. Finally, section 0 will illustrate the Cloud structure and components.

2.2.1 Cloud Computing Definition

Currently, Cloud computing is recognized as one of the most popular technologies available- it can be seen as an instance of computing as a utility. In computing as a utility, customers utilize the concept of “pay-as-you-go” for applications, computing, and storage resources [6], [7]. Along with the pay-as-you-go concept, the elasticity in upgrading or downgrading resources makes Cloud computing a popular model for organizations [8]. Moreover, the cost effectiveness of Cloud computing is encouraging its adoption. Accordingly, the enterprises requiring a high level of elasticity and to decide whether to build up their own IT infrastructure or to utilize Cloud infrastructure may find that using a Cloud infrastructure and will give a better balance between cost and elasticity [6], [8]–[10].

Ideally, Cloud computing is defined as "*a system, where the resources of a data centre is shared using virtualization technology, which also provide elastic, on demand and instant services to its customers and charges customer usage as utility bill*" [7]. Figure 2.1 visualises the Cloud Computing definition. Pay-as-You-Go and Elasticity along with On-Demand, Broad network access, Scalability and Virtualization are considered as the essential characteristics of Cloud Computing Model.

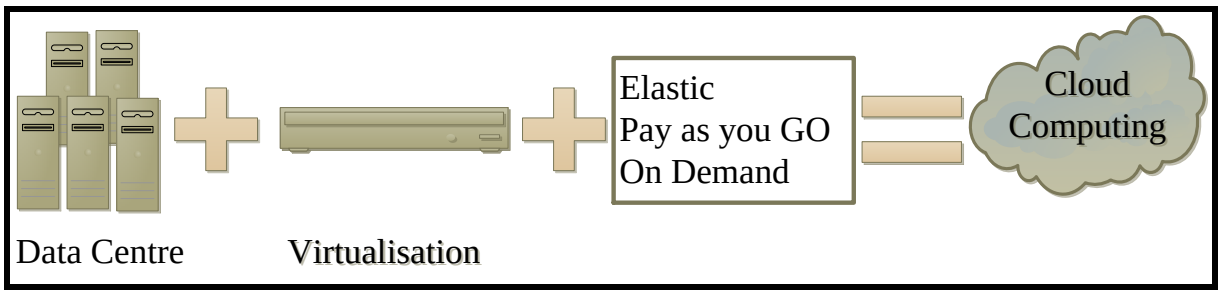

Figure 2.1: Cloud Computing Definition Visual Representation.

2.2.2 Cloud Computing Characteristic

As mentioned above *Pay-as-You-Go* and *Elasticity* along with *On-Demand, Broad network access, Scalability,* and *Virtualization* considered as the essential characteristics of Cloud Computing Model. The following is a brief description of each characteristic.

- ***Elasticity***: where resources are easily and rapidly provisioned by the customers in order to scale out or scale in; in some cases it is even automatically [7], [11]. Scale out will not take time and effort from the customer over Cloud infrastructure. Also, scale in will save resource waste and cost because whenever the customer wants to scale in the resource will be released immediately.
- ***On-Demand***: where the customers can individually provision resources as needed without any interaction with the service provider [7], [11].
- ***Measured services / Pay-as-You-Go***: where the resource use is monitored, controlled and reported by the provider [7], [11]. The charging of service is calculated by the use of resources within a period of time.
- ***Broad network access***: "*Capabilities are available over the network and accessed through standard mechanisms that promote use by heterogeneous*

thin or thick client platforms (e.g., mobile phones, laptops, and PDAs)" [1], [11].

- ***Scalability***: where adding or removing resources' nodes will not affect the setup of the Cloud and both horizontal and vertical scaling are supported by Cloud as needed [1]. For instance if a Cloud provider decided to expand the Cloud infrastructure by adding more servers that will not affect the existing customers and will not cause interruption for the service.
- ***Virtualization***: is used to achieve elasticity and cost efficiency by enabling the utilization of the hardware to its maximum capacity [3], [7]. Through virtualization Cloud providers can easily reallocate resources to obtain certain levels of utilizations (i.e. some level of utilizations can be used to balance between power consumptions and availability) [11]. Also, virtualization enables multi-tenancy in order to meet economic targets by sharing resources [4], [12].

From a technical point of view, Cloud Computing achieves scalable services delivery platform by utilizing virtualization and service oriented architecture (SOA) [3], [4], [13]. The utilization of service oriented architecture (SOA) in Cloud Computing is reflected by the different service models in Cloud Computing Model.

2.2.3 Cloud Computing Service Models

The popular models of Cloud Computing are Software as a Service (SaaS), Platform as a Service (PaaS) and Infrastructure as a service (IaaS) [1]. Each service model is described briefly as in the following:

- ***SaaS***: in this service model applications are provided as a service by the Cloud Service Provider (CSP) where the customer can't monitor or control the underlying infrastructure [11].
- ***PaaS***: in this model the CSP provides a configured environment to host the customer's application where the customer has a control over the deployed application only and possibly application hosting environment configurations [11].
- ***IaaS***: in this model the customer is capable of provisioning computing, storing and networking resources[11].

In addition, the importance of security in Cloud Computing has brought up into the scene a new service model which is Security as a Service (SecaaS). SecaaS refers to the provision of security applications and services via Cloud where these services could be provided either to Cloud Computer providers or to Cloud customers [14]. SecaaS is driven by but not limited to greater economies of scale, streamlined delivery mechanisms and focused services model [14], [15]. In a survey done by [14] security services can be categorised into ten categories based on the interest of Cloud customers and security professionals. These categories are as follows: Identity and Access Management (IAM), Data Loss Prevention (DLP), Web Security, Email Security, Security Assessments, Intrusion Management, Security Information and Event Management (SIEM), Encryption, Business Continuity and Disaster Recovery and Network Security. As an example for such services [15] from Information Systems Security Association (ISSA) suggests three possible views of services which are listed as follows:

- *Crypto co-processor:* this service is addressing confidentiality, integrity and key management in the Cloud where this is achieved by utilizing the capability of a processer called a cryptographic co-processor [2]. The protocol aims at encrypting data by dividing and distributing it within the Cloud as chunks.
- *IaaS assessment scanning:* since the control of the resources in IaaS lays on the Cloud customer an evaluation done by a third party specialized in security could do an assessment to the infrastructure such as evaluating security status and checking for vulnerabilities and misconfigurations. *This could be useful service model because the 2012 data breach investigations report shows that 92% of data breaches were discovered by a third party* [16].
- *Identity Management as a Service (IMaaS):* the idea is to introduce SecaaS as a trusted third party and play the role of Identity Federation Broker.

2.2.4 Cloud Computing Deployment Models

Security in Cloud Computing is affected by the deployment model where the security controls and measures are different from one model to another. Notably, private Clouds are considered trusted Clouds where public Clouds are considered untrusted Clouds [1], [17]. Moreover, implementing security measures and controls to solve a particular problem will be different from one deployment model to another [18]. For example implementing authentication mechanisms in community Clouds

will be different from implementing them in public Clouds; in community Clouds a trusted third party (TTP) could be used to achieve secure access where in public Clouds a scalable federated solution is needed to achieve secure access between Clouds [1], [19]. On another argument, that is not a valid statement for all security issues where physical security measures and controls are the same for all deployment models. So, in order to enhance security in Cloud Computing we need to put in our minds the different deployment models for Cloud Computing [1], [17]. And there are four deployment models for the Cloud and they are listed as follows [11]:

- ***Private Cloud**:* The Cloud infrastructure is operated and managed by it is own organization. It is possible also that a third party can manage the infrastructure instead of the organization. The Cloud infrastructure may exist on premise or off premise.
- ***Community Cloud**:* The Cloud infrastructure is shared by organizations and supports institutions like the banking that share the security requirements and compliance considerations. Moreover, the Cloud infrastructure may be managed by the organizations or a third party and may exist on premise or off premise.
- ***Public Cloud***: it is an infrastructure which is available to the public for sharing information.
- ***Hybrid Cloud***: a combination of several deployment models such as the public Cloud.

Table 2-1 shows the differences between different Cloud deployments in terms of infrastructure location, ownership, management and trustworthiness.

Table 2-1: Cloud Computing Deployment Models Differences.

Deployment Model	Infrastructure Managed by	Infrastructure Owned by	Infrastructure Located in	Trustworthy
Public Cloud	Service provider	Service provider	Off- Premise	Untrusted
Private/ Community Cloud	Organization or service provider	Organization Or service provider	On-Premise Or Off-Premise	Trusted
Hybrid Cloud	Both Organization and service provider	Both Organization and service provider	Both On-Premise And Off-Premise	Trusted and untrusted

2.2.5 Cloud Computing System Architecture

As Cloud defined in section 2.2.1, HW wise is a datacentre where its resources are shared using virtualisation technology. So, there is no special HW needed to build a Cloud Computing infrastructure. Yet, there is an advanced type of servers called Blade servers which is designed for virtual environments [3]. A blade server is a type of server with two parts: the blade enclosure where the cooling, hard drive, and power unit is located and the blades where the processers and RAMs are located. It is optional to use Blade servers for Cloud Computing for better performance and ease of management.

On the other hand, SW wise for any Cloud there is a Cloud Management Software (CMS) which is a layer just above the virtualisation and communicate directly with VMM. CMS usually takes care of the allocation of VMs and oversees the Cloud infrastructure [20]. Figure 2.2 shows an abstract Cloud Computing structure and how the components are connected.

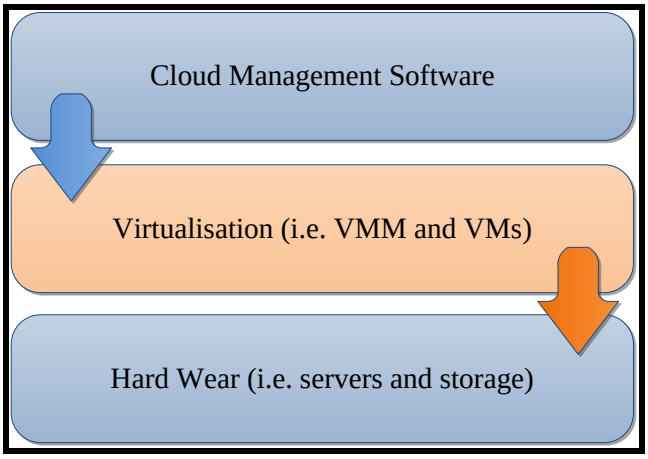

Figure 2.2: Abstract Cloud Computing Structure.

2.3 Security

In this section the security and its attributes will be defined and illustrated, the security in Cloud Computing will be highlighted and the attack theory will be demonstrated. Section 2.3.1 will define security and give its attributes. Where section 2.3.2 will show how security becomes a concern in Cloud Computing. Finally, section 2.3.5 will demonstrate the different aspects of an attack and give a statistics about them.

2.3.1 Security Definition and Attributes

Information Security refers to securing the information systems from unauthorized access, use, disclosure, disruption, modification, inspection, recording or destruction [3]. As Figure 2.3 shows, information security tries to maintain the confidentiality, integrity, and availability – aka CIA – of data where they also represent the core concepts of information security [21]. In practice, the confidentiality of data refers to limiting the access of an asset to the authorized parties (i.e. people, systems and processes).

Moreover, integrity refers to the protection of asset from unauthorized deletion, modification, or fabrication. Availability refers to an object of a system being accessible and useable upon demand by authorized entity. In addition to accountability, non-repudiation, and authenticity are the extended principles of information security (ibid). Accountability refers to holding responsibility upon an action where non-repudiation means that one party of a transaction cannot deny having received a transaction nor can the other party deny having sent a transaction. Likewise, authenticity refers to the validation process that insures both parties involved in a transaction are who they claim they are.

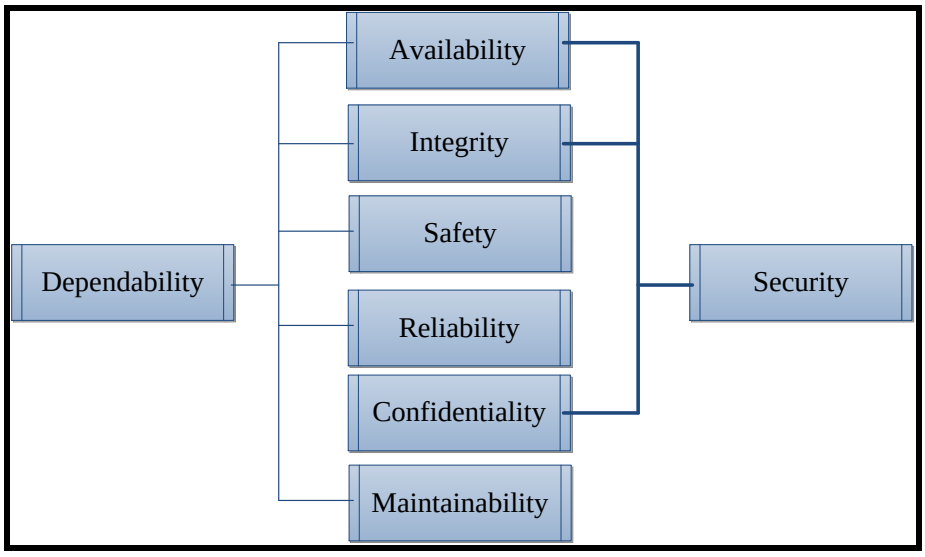

Figure 2.3: Security against Dependability attributes.

For the purpose of this project, a set of information security terminologies will be defined for a further use in the following sections such as; Threat, Attack, Vulnerability, Exploit, Risk, Attacker, Victim, Asset, Malware, and Hacking. Ideally, the threat covers the potential for violation of security, which arise when a vulnerability to the security is present. Again, an attack means any attempt to destroy, expose, alter, disable, steal or gain unauthorized access to/or make unauthorized use of an asset.

Additionally, the vulnerability is defined as a flaw or weakness in a system's design, implementation, or operation and management that could be exploited to violate the system's security policy or to reduce a system's information assurance; vulnerability is also known as the attack surface [4]. To exploit a vulnerability; an attacker must have a tool to exploit the weakness. Also, when the level of vulnerability and the level of threat are combined a risk is identified where it measures the likelihood of a successful attack. An attacker is anyone who generates an attack.

On the other hand, a victim is anyone under attack and has been damaged. An asset refers to any data, device, or other component of the environment that supports information related activities. Also, a malware is software used or created to disrupt computer operation, gather sensitive information, or gain access to private computer systems. Finally, hacking refers to a break into (a server, Web site, etc.) from a remote location to steal or damage data.

2.3.2 Security in Cloud Computing

With the benefits of Cloud Computing comes along challenges to the model; one of the most challenging of these aspects is security. Based on a study for the Cloud Security Alliance (CSA), there are seven top threats that organizations will face in adopting Cloud Computing [22]. These are *Abuse and Nefarious Use of Cloud Computing*, *Insecure Application Programming Interfaces (API)*, *Malicious Insiders*, *Shared Technology Vulnerabilities*, *Data Loss/Leakage*, *Account*, *Service and Traffic Hijacking,* and *Unknown Risk Profile* [5]. In addition, another study by Gartner has also identified seven Cloud Computing security risks, which are *Outsourcing Services*, *Regulatory Compliance*, *Data Location*, *Shared Environment*, *Business Continuity and Disaster Recovery*, *Hard Environment for Investigating Illegal Activity* and *Long Term Viability* [23]. Moreover, a survey of Cloud providers by the International Data Corporation (IDC) in 2008 to study the obstacles or concerns for adopting Cloud Computing in enterprises showed that security as a concern came first with 88.5% of the votes, whilst availability; which is one of information security principles; came third with 84.8% of the votes [10], [24].

Such concerns are driven by Cloud nature of shared resources and Multi-Tenancy. The threat of data compromise increases in the Cloud, due to the increased number of parties leading to an increase in the number of points of access [1]. Also, delegating data control to the Cloud leads to an increase in the risk of data compromise where outsourced services bypass the personal, logical and physical security controls of a consumer. A number of concerns emerge regarding the issues of Multi-Tenancy and data remanence where any resource object is reusable in the Cloud infrastructure. Reusable objects must be carefully controlled and managed since they create a serious vulnerability and violate confidentiality through possible data leakage. Data leakage in this context may be caused by the fact that hardware in Cloud Computing is not separated; there is a good level of separation in Cloud Computing at the application and virtual layer but not enough in the hardware layer [18]. Also, confidentiality could be breached due to the reusability of resource objects through data remanence, where a customer can request storage space from a Cloud provider and run a scan in order to search for sensitive data to other customers [1], [17], [18].

The most important challenge in studying security in Cloud Computing relies on the trade-off between security and cost, which is itself one of the important factors in shifting to Cloud Computing. Tim Watson, Head of the computer forensics and security group at De Montfort University notes:

> *"...although one provider may offer a wonderfully secure service and another may not, if the latter charges half the price, the majority of organizations will opt for it as they have no real way of telling the difference."* [7].

Also, George Wrenn, Security Solutions Director at Unisys recommends that customers must consider other factors more than price and top feature sets (i.e. feature sets will be different from one Cloud provider to another) before deciding to move critical systems and applications to Cloud [7]. From the previous quotations the trade-off between security and cost is obvious, where security is considered relatively costly.

2.3.2.1 Authentication in Cloud Computing

Since the triad (confidentiality, authentication and integrity) is vital in any system, then understanding the authentication mechanism in the Cloud is key. Indeed, authentication in the Cloud refers to ensuring the proper entity or person is getting access to the provided services or information granted by the Cloud technology provider. Therefore, when the user is being authenticated, then he/she is able to access information which belongs to them. In practice, the private and public type of Cloud use different model for authentication purposes. For instance, the use of multifactor authentication, access management, and the Amazon Web Services identity. Famous IT firms like Facebook, Microsoft, and Google are using such mechanism to authenticate their clients while on the Cloud [6].

The diagram (see Figure 2.4) shows an example of security mechanism used in the Cloud.

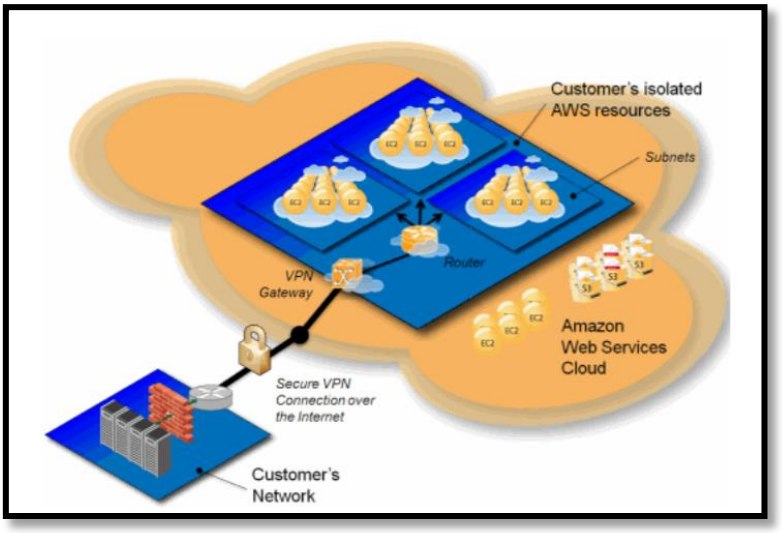

Figure 2.4: Multifactor Authentication Example

Indeed, the AWS uses the multifactor authentication mechanism since it allows for the identity and access management.

2.3.2.2 Confidentiality in the Cloud

Besides authentication, the confidentiality of data is of great significance. Ideally, the mechanism involves the use of cipher text in the process of storing data [7]. The technique is good in refraining the users and service providers from editing data that has been encrypted. Notably, the Dell corporation uses the method when data is stored on the external media or drive. Crucially, the technique is beneficial since the users do not need to bother with the enforce policies of the organizations' data encryption.

Moreover, the technique has been applied in the Waula Cloud [6]. In practice, the Waula Cloud ensures data has been encrypted before sending it to the Cloud. Such kind of confidentiality gives a high performance and great access control in the system. Thus, the methodology is important in the Cloud and the Cloud service providers are encouraged to employ.

2.3.3 Security at the Compute Level

Saini and Saini talk about the security levels in a Cloud environment. The two mention of the division of the security levels in the computing category within the Cloud infrastructure. First, we discuss the physical server security. Indeed, at the level, the users are expected to have user groups which can operate the server

using access privileges. Therefore, the protection measure must ensure a physical server security determines the authentication and the authorization mechanism [1].

Besides, we mentions about securing the hypervisor. Since the Hypervisors run all the VMs, an attack on it can lead to an impact on all the VMs in the network. Unfortunately, the Hypervisor exposes a single point of failure to the Cloud infrastructure. Hence, the security can be attained by tightening the firewalls between the management system and the main network.

2.3.4 The Security at the Operating System

Despite the security models in the Cloud, the operating system (OS) has a significant role in determining whom to trust. Therefore, the OS depends on the rings of protection to ensure the system is safe. The rings work like the family members, friends, co-workers, and acquaintances [3]. Accordingly, the people close to an individual, such as the wife or children, have a higher priority of trust. Whereas, those who are distant acquaintances have a lower level of trust. Likewise, the OS levels its environment in rings for it to gain trust in those components.

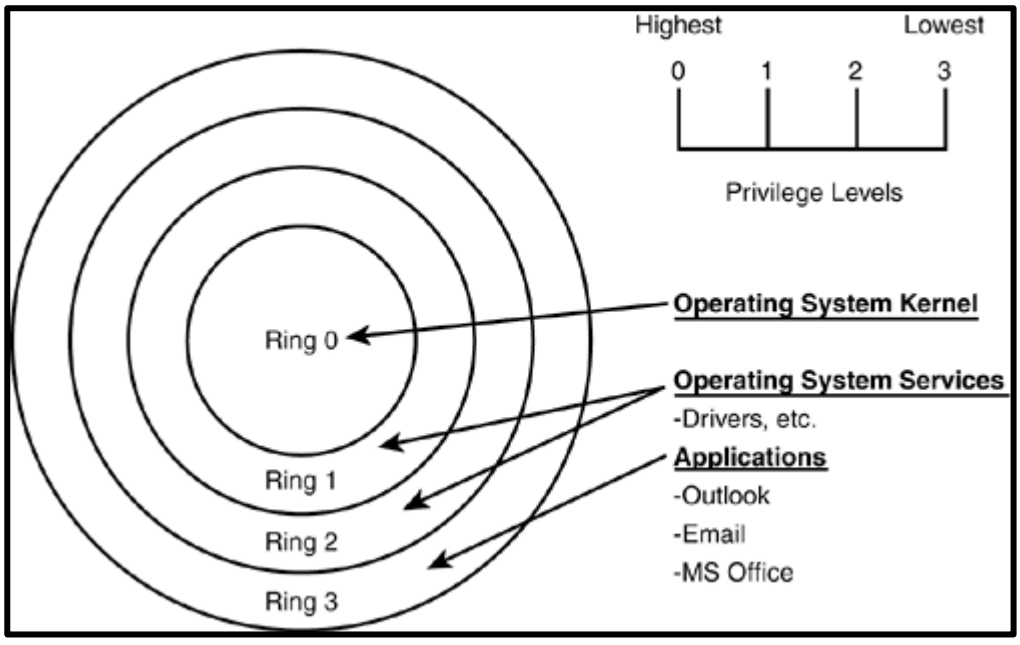

Figure 2.5: The Security Rings in the OS

The model above shows how the operating system determines who to trust. For instance, it determines the levels at which the operating system can allow a code to be executed. Crucially, components at level 0 are the most trusted. Therefore, anything running in ring 0 is considered to be running in the operating system privilege mode. Besides, those running in ring 1 are considered the non-privileged level of the OS [3]. The next level (ring 2) is preserved for the input and output parts of the processor. Essentially, the utilities and the drivers reside in that part. Finally, the ring 3 is meant for those applications that operate at the user levels. In practice, those levels are meant to bring in process isolation which improves the security of the devices in machines, especially in a compromised situation like the Cloud environment.

2.3.5 Attack theory

Security experts usually scan and assess a system to identify vulnerabilities; and to overcome a weakness, we must know how it can be exploited? Understanding the exploitation mechanism will lead to the solution of securing the vulnerability. This puts us in a position where we have to know about the attacks, their behaviours, and different levels that the attacks need to pass to achieve the target.

The nature of Information Security (InfoSec) for a specific vulnerability there could be many numbers of attacks to exploit it [2]. On the other hand, one successful attack against a system will identify most of the possible vulnerabilities that can be utilised. Moreover, the attacks vary in the sense of their behaviour; for example, it is easy to detect any distributed denial of service (DDoS) attack, and any attack consists of port scanning due to the unexpected increase in the traffic. Also, it is easy to identify viruses due to their unique signatures; whereas it is hard enough to detect iFrame attacks. The iframe attack is an attack where an HTML code is embedded inside another HTML code as a frame to collect credit card information [8].

Data Breach Investigations Report (DBIR) report has been generated since 2004, and all results are based on first-hand evidence collected during paid external forensic investigations conducted by Verizon from 2004 to 2011 [9]. All the results that will be listed below are taken from the DBIR 2012 [16].

The report of 855 incidents and 174 million records show that 98% of the attacks are generated by external agents (i.e. they are not related to the organization by any means). And only less than 1% of the attacks are generated by a business partner. On the other hand, the results show that about 81% of the attacks are generated by utilizing some form of hacking; Where 69% of the attacks are done by utilizing malware. Also, 79% of the victims were targets of opportunity; that means the attackers did not identify the victim to generate the attack but they identified a vulnerability to generate the attack. Moreover, 94% of the data breaches involved servers with an increase of 18% from the results of DBIR 2011. And 85% of the incidents took weeks or more to be discovered; where 92% of the incidents were discovered by a third party. Also, the report shows the relationship between the four A's which are Agents (attacker), Action (attack), Asset and Attributes (security principles); where 518 out of 855 of the incidents are done by external hackers against servers and violated confidentiality. Also, 422 of the incidents were violating integrity and authenticity in the same class (i.e. with the same attackers, attack and asset).

From another perspective, the difficulty of the attack may affect the threat model that means the levels and the security controls needed to be bypassed in order to accomplish the target. And for that purpose, the report classifies the difficulty of the attack into four classes as follows:

- "Very Low" difficulty of attack: no special skills or resources required, for example, the average user could have done it [9].
- "Low difficulty": basic methods, no customization, or low resources required, for example, automated tools and scripts [9].
- "Moderate difficulty": attacks where skilled techniques, some customization, or significant resources required [9].
- "High difficulty": attacks where advanced skills, significant customizations, and/or extensive resources required [9].

Moreover, the report provided separate results for the initial compromise and the subsequent actions that follow it. For example, the initial compromise talks about the unauthorized access to an asset, and the latter is what is done to compromise and infiltrate the data. Indeed, the data shows that 65% of the data breaches and 37% of the records were compromised by a low difficulty attacks in the initial compromise. Also, 24% of the data breaches and 16% of the records were compromised by a moderated difficulty attacks in the initial compromise. On the other hand, 4% of the data breaches and 61% of the records were compromised by a high difficulty attacks in the subsequent actions. Consequently, 39% of the data breaches and 37% of the records were compromised by a moderated difficulty attacks in the subsequent actions. In a similar context, about 29% of data breaches are done by a one action attack; that means the attacker had to bypass only one security measure. And the rest data breaches are done by two or more attack actions.

Based on the above, the vulnerability, penetration mechanism and the capability forms the core elements for a successful attack. Figure 2.6 shows the elements needed to have a successful attack.

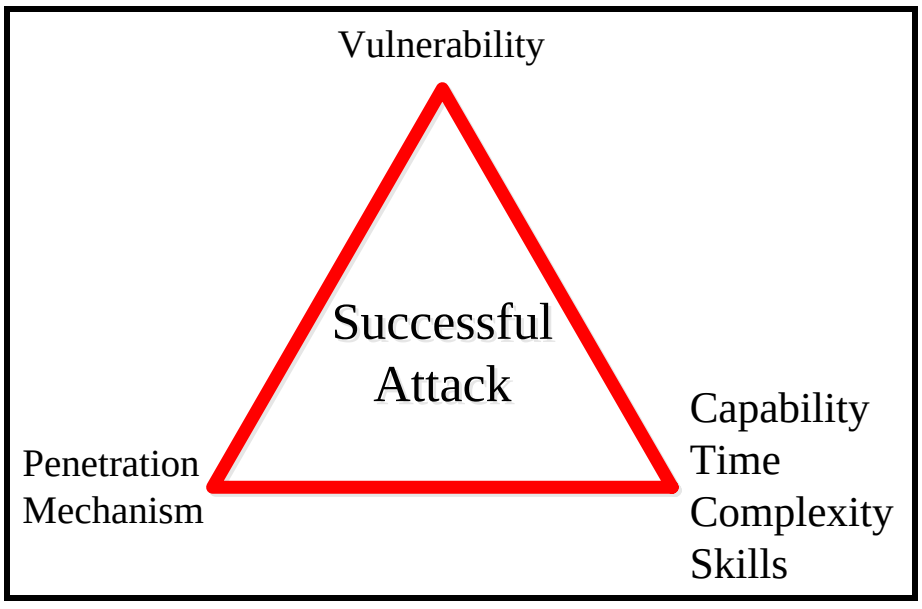

Figure 2.6: Attack Theory.

2.4 Multi-Tenancy

In this section the concept of Multi-Tenancy will be introduced, its importance and its security challenges will be discussed. Section 2.4.1 will define and describe Multi-Tenancy. Where section 2.4.2 will show its importance to Cloud Computing. Finally, section 2.4.3 will demonstrate the different arguments on Multi-Tenancy and its security challenges.

2.4.1 What is Multi-Tenancy?

Multi-Tenancy is a natural result of trying to achieve economic gain in Cloud Computing by utilizing virtualization and allowing resource sharing [2], [9]. Multi-Tenancy refers to resource sharing in Cloud Computing, but such a definition is still general in the context of Cloud Computing, where Multi-Tenancy is seen differently from different service models.

In Software as a Service (SaaS), applications are provided as a service by the Cloud Service Provider (CSP) where the customer cannot monitor or control the underlying infrastructure; here, Multi-Tenancy means that two or more customers utilize the same service or application provided by the CSP regardless of the underlying resources [4][12].

In Infrastructure-as-a-Service (IaaS), where the customer is capable of provisioning computing, storing, and networking resources and can control, but cannot manage the underlying infrastructure [3]. In essence, the clients in the IaaS environment can install and use an arbitrary operating system [4]. Consequently, Multi-Tenancy occurs when two or more virtual machines (VMs) belonging to different customers share the same physical machine [2].

$$Virtualization + ResourceSharing = Multi - Tenancy \quad (1)$$

As equation (1) shows, in order for Multi-Tenancy to occur – in IaaS – both virtualization and resource sharing must be allowed by the CSP.

2.4.2 Why Multi-Tenancy is important?

Figure 2.7 shows all the identified possible benefits of Multi-Tenancy and by looking into the tree's leaves, it is easily recognized that the origin of the benefits could be linked either to virtualization, resource sharing or by combining both of them.

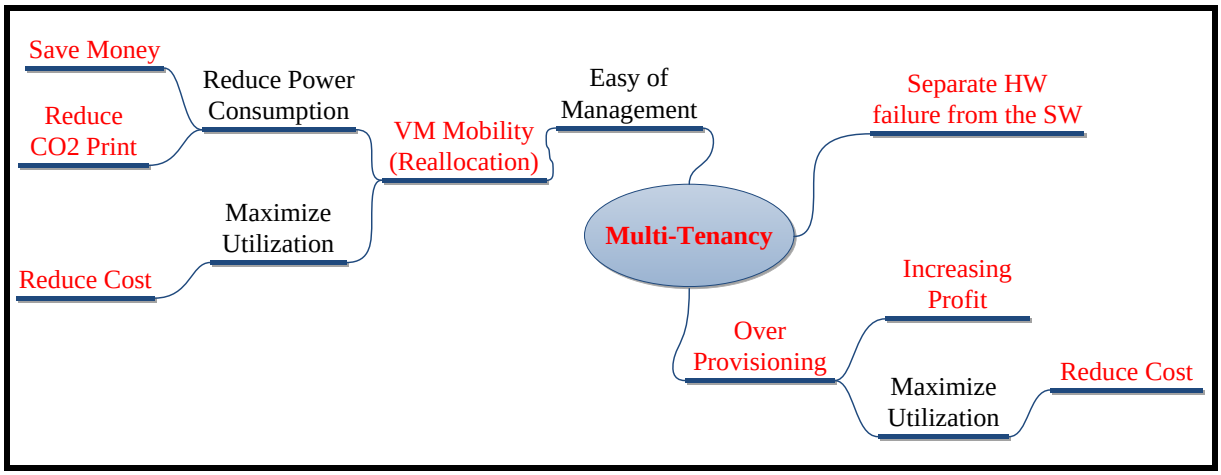

Figure 2.7: Multi-Tenancy Benefits' Tree.

For instance, separating the hardware failure from the software failure is achieved by virtualization. On the other hand, sharing the resource will increase the utilization which will lead to a reduction in cost by making the resource available for more than one customer.

In other cases such as over provisioning and VM mobility, both virtualization and resource sharing will amplify their impact. VM mobility can contribute in maximizing the utilization of the infrastructure or reducing the power consumption by reallocating VMs into clusters and minimizing the number of servers used. Whereas over provisioning is considered one of the major features of Cloud Computing since it gives the opportunity for the CSP to seal more than the capacity of his infrastructure. These features are important for Cloud Computing and any proposed solution must be added to them or at least try to keep them and not to eliminate a single one of them.

2.4.3 Arguments about Multi-Tenancy

Multi-Tenancy has been identified as a security issue in Cloud Computing by several researchers such as [18] who conducted a survey on security issues in service delivery models in Clouds and stated that Multi-Tenancy is a major Cloud Computing characteristic that may lead to confidentiality violation. In addition, [1] identified Multi-Tenancy as a major threat to both confidentiality and privacy when talking about Cloud Computing security. Furthermore, it highlighted shared technology vulnerabilities – hence Multi-Tenancy – as one of the top threats to Cloud

computing in a survey done on the existing literature [7]. Moreover, it recognizes Multi-Tenancy as a new source of threat in Cloud Computing infrastructure [9].

From another point of view, [21] linked between Multi-Tenancy as a form of shared environment and the attraction of malicious activities in the Clouds. Intel IT Centre [25] generated a document of best practices on building secure Clouds; yet it clearly highlighted Multi-Tenancy and shared technology issues as security challenges for a Cloud environment. Where [26] in his work proposed a layered security approach for Cloud Computing, and states that virtualization is one of the process hosting layer (i.e. servers) issues where competitors will have separate virtual machines in the same physical machine; hence Multi-Tenancy.

In [27] several areas were identified as danger in Clouds; under data governance the writer highlighted that Multi-Tenancy arrangements in Clouds are raising questions about data segregation. While NIST developed a report titled "*Guidelines on Security and Privacy in Public Cloud Computing*"; they identify Multi-Tenancy as of the security and privacy downsides in the Cloud [28]. In a totally different approach [29] interviewed five leading scientists from the Cloud community; Raghu Ramakrishnan the Chief Scientist for Search and Cloud Platforms at Yahoo! was one of them, where his response to the question of "On a related note, for a graduate student starting a PhD, what would you say are the key fundamental challenges of Cloud computing that should be addressed by new research in the field?" included Multi-Tenancy as a fundamental challenge of Cloud Computing. Again [30] raised questions in how Cloud Computing affecting security, privacy and trust; where he identified Multi-Tenancy as one of the security issues.

Cloud Security Alliance (CSA) released a document titled "*Security as a Service*" [14] where they attempted to define categories for services; they raised the question "*How does one assure data isolation in a multi-tenant environment?*". Also, CSA in the same document stated that Multi-Tenancy is creating new targets for intrusion. In a study done by [31] to identify the challenges of security and privacy in Cloud Computing; Multi-Tenancy is recognized as one of the unique implications of security and privacy in Cloud computing. In the same direction [32] defined Multi-Tenancy as a major characteristic of Cloud Computing and a major dimension in the Cloud security problem that needs a vertical solution from the Software-as-a-Service

(SaaS) down to Infrastructure-as-a-Service (IaaS). Where [33] highlighted the fact that Multi-Tenancy may enable information leakage and increase attack surface which will affect the security of the Clouds. Also, [34]–[36] considered Multi-Tenancy among the serious issues in Cloud security.

Multi-Tenancy has brought different arguments in Cloud Computing. While software developers see it as an opportunity, security experts see it as vulnerability [2], [12], [37], [38]. Even though security experts agree that Multi-Tenancy is a vulnerability that could lead to confidentiality or/and integrity being exposed, they vary in providing the solution for such vulnerability.

Whereas [38] suggested the elimination of the virtualization layer in order to prevent multi tenancy, [2] suggested that the provider should expose the risk of Multi-Tenancy to the customer and do nothing about it (i.e. give them the option of paying extra to avoid Multi-Tenancy). The first strategy seems very effective, but would eliminate great benefits for Cloud providers such as VM mobility and financial gain due to resource sharing.

VM mobility is one of these benefits where providers can easily reallocate VMs to achieve better utilization and save power consumption. Moreover, the paper did not mention any thing about the cost of change; the paper clearly stated that all the hardware needed is currently available. But such a specific hardware is implemented upon request because it is considered relatively costly when compared with other hardware that provides the same capability. In addition, the paper did not mention how big existing Cloud providers will manage to shift from the current practises into the proposed solution? And how much will be the cost of change? Also, how the new solution will affect the management of Cloud resources? In order to preserve the security Cloud providers have to spend a lot of money.

On the other hand, the second strategy will not enhance the Cloud security and customers especially enterprises are holding back investment in Cloud Computing because of security issues [9], [13], [18].

Moreover, current practice of UK enterprises is to deploy Private Clouds in order to cut costs and safeguard sensitive data [18]. We therefore identify that a solution securing Multi-Tenancy yet keeping its benefits is needed. So, a deep

understanding of Multi-Tenancy is required in order to identify all the possible benefits brought to Cloud Computing because of Multi-Tenancy.

2.5 Security Domains in Cloud Computing

What is unique about Multi-Tenancy in Cloud Computing is that both the attacker and the victim are sharing the same server – PM. Such a setup cannot be mitigated by traditional security techniques and measures, simply because it is not designed to penetrate inside servers and their monitoring techniques are limited to the network layer [13], [39].

To illustrate, Figure 2.8 shows the different cases of attacker and victim locations and the networking between them. In case one, the attacker and the victim both are regular Internet users; in order to defend against such attacks, traditional network security techniques and devices are efficient.

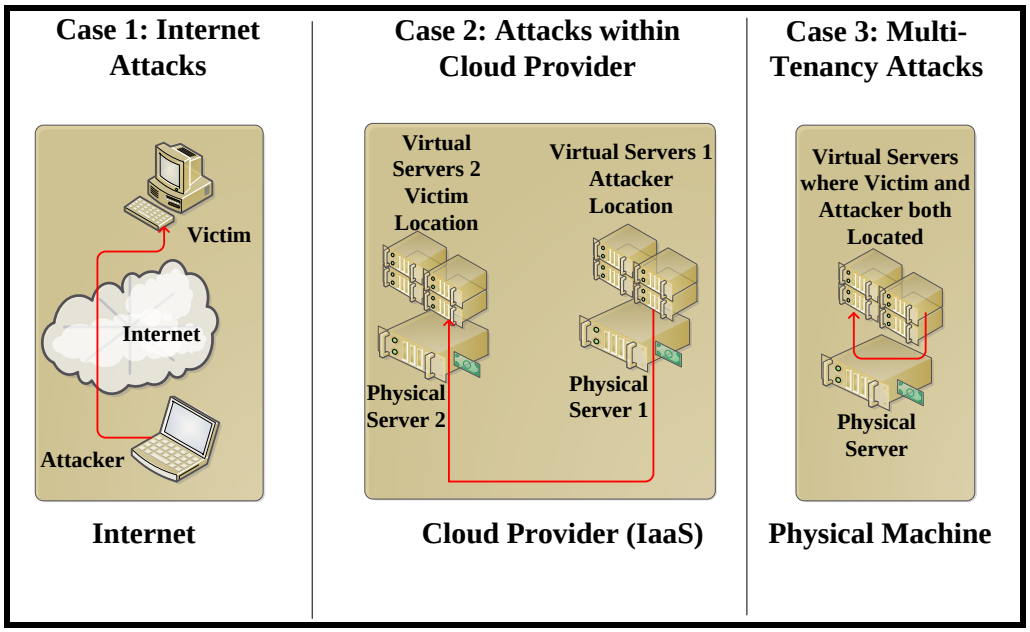

Figure 2.8: Different Scenarios for Security.

In case two, both the attacker and the victim are customers in the same Cloud provider but each one of them is located on a separate server. This kind of setup is due to the utilization of the virtualization layer in the Cloud Computing Model; to secure such setup, virtual network security devices and techniques must be implemented by Cloud providers [39].

Case three describes the problem that we intend to address in this project, where both the attacker and the victim are customers in the same Cloud and are sharing the same server. Such situation is due to Multi-Tenancy; securing such setup is not an easy task as network communication between the attacker's VM and the victim's VM is limited within the PM. Therefore, traffic will not leave the PM, which is harder to be mitigated by virtual network security defences as opposed to case two.

In order to secure such vulnerability, we must first answer the following question: how is Multi-Tenancy exploited? An answer can be found in [2], where an attack is generated over the Amazon EC2 Cloud to investigate data leakage. In order to carry out the attack, network probing is performed; following this, a brute force attack is generated to take advantage of the Multi-Tenancy effect by allocating the attacker's VM beside the victim's VM [3]. The results show that by spending just a few dollars, an attacker has a 40% chance to allocate his VM beside the victim's VM. After achieving Multi-Tenancy, a side channel attack – any attack takes advantage of the system characteristics – is generated to extract the data of the victims.

Obviously, any tenant can attack its neighbour because the type of attack that could be utilized, such as side channels, cannot be detected by the hypervisor or even the operating system. So, there is no way to eliminate the Multi-Tenancy effect in order to keep its benefits, yet the effect could be minimized. Multi-Tenancy cannot be eliminated, but a smart resource allocation technique will minimize the risk of Multi-Tenancy; in other words, a resource allocation technique will increase the level of difficulty of achieving Multi-Tenancy for customers, yet it is easily managed by Cloud providers. What is interesting of Multi-Tenancy is that in order to achieve it for targeted victims, the attacker needs to invest an effort, time and cost. So, by making Multi-Tenancy difficult to be achieved by customers, we are restricting the number of potential attackers.

2.6 Security Attacks

Attacks vary on their complexity, techniques used and their behaviour. Such variation is justified due to their aim and the nature of the attack vector been exploited. In Cloud Computing, some of the conventional attacks and techniques are still valid and new attacks have emerged. Some of well-known attacks that could be utilized efficiently over the Cloud infrastructure will be given.

One of the wide range attacks is Side Channel attack; a side channel attack is any attack that takes advantage of the physical characteristics of a system. Side channel attacks have several attack forms such as memory attacks, timing attacks and power attacks. There are many side channel attacks known in the field, the following show some of the well-known side channel attacks:

- Timing attacks are based on measuring the time it takes for a unit to perform operations. Observing the time variance will reveal how a system is designed and how to exploit it [40].
- Power Consumption attacks needs a good understanding of the system HW. The least component of any system's hardware is the transistor where the transistor is working as a voltage switch. So, by just analysing the power consumed by a unit while performing different operations; an attacker can identify the processes of a system. Knowing such information makes it possible to exploit the system (ibid).
- Differential fault analysis is the form of attacks when attackers study the behaviour of a system by injecting faults into it; such technique gives them the opportunity to understand the system and its flaws [40].

Another effective attack strategy is brute forcing. Brute forcing is an attack strategy or mechanism which could be applied over any kind of attacks. It is one of the simplest strategies in order to build an attack but it is one of the most common used strategies as mentioned in section 2.3.5. For instance if an attacker wants to find out a password of a system and utilizing brute force strategy that means the attacker will try every possible combination until the correct password is found. So, brute forcing could be defined as running an attack operation multiple times until a

successful breach is done. Brute forcing is identified as one of the top ten attacks by [16] where it forms 23% of data breaches attacks.

Besides, network probing is another mechanism that is vital. The technique is used to find out the physical topology of a network that consists of IPs and servers connected in the network. Such information could be utilized to identify possible targets and to design an attack for a sub group in the network.

In addition, Denial of Service attack (DoS) or distributed denial of service attack (DDoS) is an attempt to make a machine or network resource unavailable to its intended users [10]. Such attack generally consists of the efforts of one or more people to temporarily or indefinitely interrupt or suspend services of a host connected to the Internet.

On the other hand, Virtual Machine Escape is an exploit in which the attacker runs code on a VM that allows an operating system running within it to break out and interact directly with the hypervisor [11]. In such attack, an intruder can access other VMs after having an access to the host OS.

Finally, as this project focus on Multi-Tenancy a set of mechanisms could be used in order to detect if the VM is a tenant to another VM. Co-residency (which is another term used exchangeable with Multi-Tenancy) could be detected using the same way Multi-Tenancy is exploited by – hence side channels [2]. One of the used techniques is matching Dom0 IP in Xen hypervisors [41]. By running `trace rout` command to specific VM, if the Dom0 IP of the source VM matches the Dom0 IP of the destination VM then they are co-resident. Another easy way to detect co-residency is by measuring packet round trip time where the round trip time of any targeted VM is measured against a co-resident VM and calculate the difference in order to figure how far is the targeted VM.

2.7 Summary

This chapter grounded and established the required knowledge in order to propose a novel approach to tackle Multi-Tenancy and present a solution to enhance its security. The concept of virtualisation, Cloud Computing, Multi-Tenancy, and security and security attacks were presented.

Virtualisation as the main player of Cloud Computing was defined and its components were described in section 2.2. Then, Cloud Computing was defined and its characteristics were highlighted and discussed in section 2.2. Along with demonstrating its service and deployments models. Finally, an abstraction of the Cloud Computing structure was illustrated at the end of section 2.2.

Next, the definition of security and its attributes against dependability attributes were demonstrated in section 2.3. After establishing the grounds of Cloud Computing and security, the issue of security was highlighted and detailed to position up the research problem. Finally, the attack theory and some supporting statistics were presented and illustrated.

After that, the concept of Multi-Tenancy where presented in section 2.4. starting with illustrating what is Multi-Tenancy and how does it form in Clouds; and passing through its importance and ended by showing the different arguments about it and their grounds.

Finally, in section 2.5 and section 2.6 the security domains in Cloud Computing and security attacks were discussed respectively. The security domains in Cloud Computing were demonstrated and discussed to highlight the scope of the research project and its challenges. Where the security attacks were discussed and illustrated to highlight the visibility of the study and the importance of the solution.

The next chapter will present and describe Google dataset as the only big data related to Cloud Computing made available to the extent of our knowledge. Some of the important work that used the data will be presented and its details and descriptions will be also illustrated to lay down the needed knowledge for the analysis and results sections.

Chapter 3 Case Study: Analysis of Google Data

We will investigate the Google dataset through outlining the components of the machine events, attributes, job events, task constraints, and resource usage. Through the analysis, the quantification of the multi-Tenancy will be covered in details and the methodology outlined. Moreover, the section will describe the platform using the Spearman's correlation test to know the effect of multi-tenancy in the systems. Since the aim is to inspect the vulnerabilities in the Cloud, an attack model will be described in the chapter. Crucially, the section is useful in laying the foundation for the subsequent chapters.

3.1 Google Dataset

To better understand the issues and challenges in developing and adopting the Cloud, an analysis on real Cloud data is a crucial step. One of the biggest data released for public related to Cloud Computing is Google dataset. Specifically, Google has recently released two sets of data (7-day and 29-day sets) [42]. These sets have been investigated and analysed by many researchers in the literature [43]–[47]. Those studies were focusing on resources utilization, scheduling, relations with Grid/HPC systems, scalability, cluster management, and behaviour of workloads, but with little focus on user behaviour, security and the patterns of the workloads.

Moreover, the users of these trace logs have been identified as Google engineers and services [42], [46]. Reference [47] has concluded that there is a dependency/relationship between resource utilization, the number of tasks and user patterns. Another study by [45] which examined Google trace logs concerning workload characteristics stated that the most notable workload characteristic is heterogeneity. They stated that such heterogeneity leads to complications in resource allocations and utilizations.

Besides, [43] conducted a comparative study between Google data set and Grid/HPC systems, stating that Google workloads show that resource allocations are finer concerning CPU and Memory than that of Grid/HPC systems. Reference [46] conducted a study on the workload characteristics of Google Dataset. They concluded that machines are continuously taken offline and online to combat system

failures and to apply upgrades. Also, many of the submitted jobs are not latency sensitive as more jobs are killed before normal completion.

Therefore, having reviewed the related part of the literature and after examining the Google dataset, it is concluded that the workload consists of many patterns, depending on the angle of attention. In this project, Multi-Tenancy will be highlighted as the vulnerability and in-depth understanding related to different dimensions of Multi-Tenancy is required. So, we decided to empirically investigate the possibility of reconstructing the proposed attack model from the dataset released by Google. Such activity can be used as a monitoring tool where CSP can monitor some behaviour that can be linked to popular attack models. Also, a quantification of the Multi-Tenancy will be done using the dataset.

3.1.1 Dataset Tables and Descriptions

The dataset as released form Google was described in [42], it consists of six tables where two of them are related to the machines, one of them related to jobs, two are related to the tasks, and the last one is related to resource usage. Table 3-1 shows the dataset tables and its attributes.

Machine events is the first table described by Google where it captures the machine details. The timestamp attribute stores the time in microseconds where the beginning of the trace log is time 600 seconds. So, if and event happened 60 seconds after the beginning of trace log its timestamp would be 660 seconds. Any event started before the trace log would have the timestamp 0 and any event ended after the trace log end time would have the timestamp 2^{63}-1 (i.e. maximum integer as the timestamp is recorded as a 64 bit integer). Every machine in the dataset is a unique machine and has an ID which is stored in machine ID. Any event happened to the machine is recorded in the machine event. The event of the machine could be one of three possible events as follows:

- ADD (0): any machine that is attached to the infrastructure will have the event add. In other words all the machines in the trace log will have this event type.
- REMOVE (1): any machine fail or under maintenance will have this event type.
- UPDATE (2): whenever a machine capacity in terms of CPU or RAM is upgraded or downgraded will have this event type.

The platform ID is a string represents the different microarchitecture of the machines. Capacity of the CPU and memory represent the resource capacity of the machine in terms of CPU units and memory capacity.

Table 3-1: Dataset Tables and its Attributes.

Table Name	Machine events	Machine attributes	Job events	Task events	Task constraints	Resource usage
Attributes	*1. Timestamp*[1] *2. Machine ID* 3. Event type 4. Platform ID 5. Capacity: CPU 6. Capacity: memory	*1. Timestamp* *2. Machine ID* *3. Attribute name* 4. Attribute value 5. Attribute deleted	*1. Timestamp* 2. Missing info *3. Job ID* 4. Event type 5. User name 6. Scheduling class 7. Job name 8. Logical job name	*1. Timestamp* 2. Missing info *3. Job ID* *4. Task index* 5. Machine ID 6. Event type 7. User name 8. Scheduling class 9. Priority 10. Resource request: CPU 11. Resource request: RAM 12. Resource request: Disk 13. Different-machine constraint	*1. Timestamp* *2. Job ID* *3. Task index* *4. Attribute name* *5. Comparison operator* 6. Attribute value	*1. start and end time* *2. job ID* *3. task index* 4. machine ID 5. CPU usage (aka rate) - mean 6. memory usage 7. assigned memory 8. unmapped page cache memory usage 9. page cache memory usage 10. maximum memory usage 11. disk I/O time - mean 12. local disk space used - mean 13. CPU usage (aka rate) - max 14. disk IO time - max 15. cycles per instruction (CPI) 16. memory accesses per instruction (MAI) 17. sampling rate *18.* aggregation type

1 Any italic attribute is a key in the table.

Machine attributes is the second table related to the machine where timestamp and device ID are the same as computer event table. What is new is the attribute name, attribute value, and the attribute delete. Ideally, an attribute name represents a machine property such as kernel version and clock speed - attribute value is a representation of the attribute name number across all devices. Furthermore, "attribute delete" is to indicate if an attribute is previously deleted or not.

Job events are the third table and the only one to capture all the data about jobs. Each job is unique, and its ID is represented in the job ID attribute as well as the job name. Whereas, the logical job name is a combination of several inputs gathered from different fields. The scheduling class shows how sensitive a job or a task to latency where 0 represents a non-production and 3 is more latency sensitive job or work. Specifically, the priority ranks the job or task among the other jobs or tasks. Each task could consist of one or more task, and each task is a Linux container which is a form of virtualization (for instance, the task or VM will be used interchangeably in the document from now and forward). Figure 3.1 shows the state transition for jobs and tasks. For every job or task there are nine event types as shown in Table 3-2.

Table 3-2: Job and Task Event Type.

Event type	Description
Submit (0)	A task or job became eligible for scheduling.
Schedule (1)	A job or task was scheduled on a machine. For jobs, this occurs the first time any task of the job is scheduled on a machine.
Evict (2)	A task or job was de-scheduled due to a higher priority task or job or because a disk holding the task's data was lost.
Fail (3)	A task or job was de-scheduled due to a task failure.
Finish (4)	A task or job completed normally.
Kill (5)	A task or job was cancelled by the user or a driver program.
Lost (6)	A task or job was presumably terminated, but a record indicating its termination was missing from our source data.
Updated pending (7)	A task or job's scheduling class, resource requirements, or constraints were updated while it was waiting to be scheduled.
Updated running (8)	A task or job's scheduling class, resource requirements, or constraints were updated while it was scheduled.

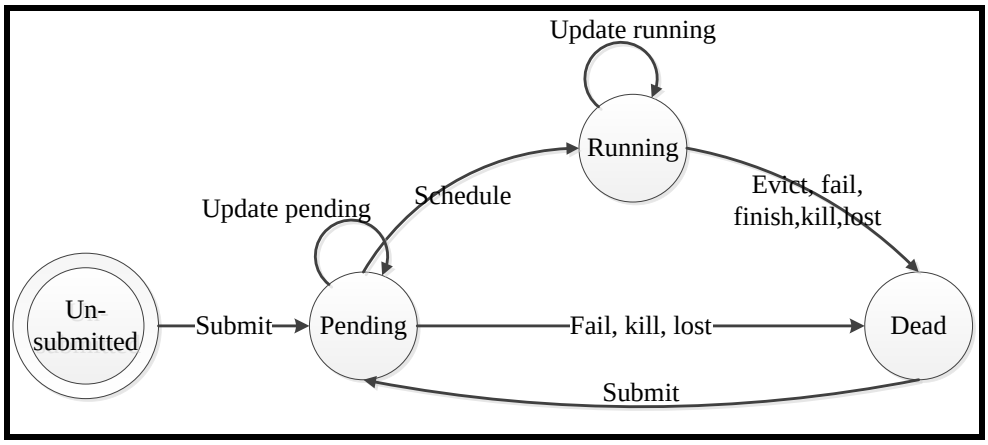

Figure 3.1: State Transition for Jobs and Tasks.

Task events table captures the details of the individual tasks. The task index, when combined with the job ID, creates a unique ID for the task in the whole dataset. Besides, the resource request for CPU, RAM, and local disk are the recourses estimated by the customer and not granted for sure.

Task constraints table captures any user requirement for a given task. Specifically, any constraint on machine attribute will be named on attribute name, and the comparison operator will reflect the difference in the attribute by comparing the attribute value here against the attribute value in machine attributes table.

Resource usage is the last table and the biggest among them where the requested and actual usage of the resources is captured.

3.2 Quantifying Multi-Tenancy

3.2.1 Methodology

Figure 3.2 shows the data processing pipeline which is the methodology used to study the dataset. Each phase will be described in details where the needed tools, requirements and outcomes will be discussed.

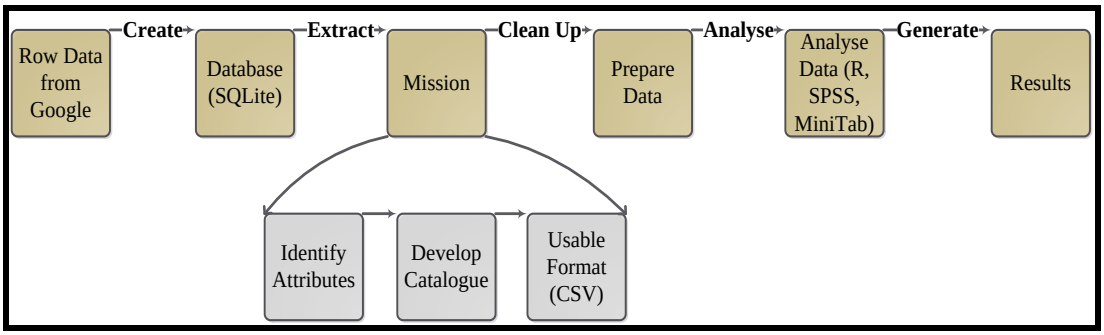

Figure 3.2: Data Processing Pipeline.

- ***Row Data from Google:*** the data as published by Google was in CSV[1] format where each table of the data was located in a directory along with its hash file in order to check the integrity of the data. As shown in Table 3-1, the dataset consisted of six tables. For the sake of this study, the data was downloaded and stored in a PC and replicated on ARC1[2] to allow group sharing and take advantage of both the storage capacity and the backup service.
- ***Database:*** row data is used to create a database in order to investigate it. The database consisted of six tables with its attributes as shown in Table 3-1. SQLite[3] is used as a database engine.
- ***Mission:*** using the database, a catalogue is extracted for each mission. A catalogue is a database table consists of selected attributes from the original dataset's tables focusing on one aspect for further investigation. This phase is called mission as for some cases more than one catalogue is needed for the same purpose. So, a mission is a collection of catalogues which cover one aspect from different dimensions. Each mission consists of three phases as follow:
    - *Attribute Identification:* a selected attributes of the original attributes are selected which relate to the element under investigation such as Multi-Tenancy.

1 Is a comma separated values file which allows data to be saved in a table structured format. It is widely used as most of the applications have the capability to import it and generate files in such format.

2 A High Performance Computing (HPC) facility owned and managed by University of Leeds.

3 Is a software library that implements a self-contained, server less, zero-configuration, transactional SQL database engine.

- *Catalogue Development:* after identifying the needed attributes, a set of SQL commands are coded to develop the needed catalogue or catalogues as needed.
- *Usable format:* the catalogue is developed in a database and for further analysis other softwares and tools must be used, for that a usable format is vital for easy of catalogue handling. As most of the tools accept CSV format, the catalogue then is generated in CSV format.

- ***Prepare Data:*** after developing the catalogue, a clean-up and preparation of the data is needed depending on the mission and the data analysis tool will be used. Data preparation done using MS Excel where additional columns are generated as needed, change on units if needed or coding names. The data then is formatted into txt[4], xlsx[5] or CSV depending on the analysis tool. If the data is going to be analysed using R[6], then a txt format will be used. And if the data going to be analysed using Excel or SPSS[7], then the xlsx format will be used. Otherwise CSV will be the default file format of the data.
- ***Analyse Data:*** after the preparation of the data, it is ready to be analysed. For this phase several tools were used for different missions. R, SPSS, Minitab and Excel were used to analyse different data and generate results and draw conclusions. Different tools were used as different methods were needed to get useful information from the missions.
- ***Results:*** the final phase is the generation of results based on the data analysis. Tables, pie charts, graphs and figures were generated to present the results of different missions.

3.2.2 Sampling method

The dataset is considered big data in terms of its volume. Table 3-3 shows the row data tables' sizes. The total size is 185.325 GB before creating the database, this size will be increased after creating the database as the

4 Is text format file.
5 Is an Excel file format.
6 Is a free software environment for statistical computing and graphics.
7 Is a software package used for statistical analysis.

relationships of tables and keys are also stored. The size of database is increased with each mission as catalogues are created.

Table 3-3: Row Data Tables' Sizes.

Table	Size
Machine Events	2 MB
Machine Attributes	1 GB
Job Events	323 MB
Task Events	16 GB
Task Constraints	2 GB
Resource Usage	166 GB
Total	**185.325 GB**

The database is vital in order to investigate the dataset and create catalogues. At the beginning the database were created on ARC1 to allow group access for the database and to distribute the SQL commands in order to save time. For example, A single join command between any table and the resource usage table in a single PC, took up to 21 days as there are more than 174 billion raw of data in the resource usage table. So, in order to develop a catalogue which consists usually of more than select and join command involving resource usage a time of three months may be needed which unpractical and unachievable due to the PhD study time constraint. Furthermore, if the time concerns could be ignored the memory limitation will pop up. As a result, a distributed solution was needed where the use of ARC1 and ARC2[8] then became vital. Unfortunately, due to technical limitations because of management decisions there was not a distributed database installed in ARC1 or ARC2.

The advantage of ARC1 and ARC2 were the storage capacity as the database could grow without any restrictions, yet the processing time were similar to using a single PC as the database could not be distributed. An effective solution is to use Hadoop[9] over a cluster, yet such facility is not

8 The second generation of ARC1 where the infrastructure were expanded and upgraded.

9 Is a framework that allows for the distributed processing of large data sets across clusters of computers using simple programming models.

available on the University of Leeds and is restricted by a tight management process.

Due to the limitation on computational power and the huge volume of data needed to be analysed, sampling the data is an effective and scientific approach to investigate huge populations. This approach is used with when there is a limitation in studying the whole population or there are technical limitations or constrains in studying the population. For example, [57] studied the first 48 hours from the dataset and drawn conclusions about the total population.

For this approach, the important question is how to know that the sample is representative? Once being able to answer this question in a scientific manner, the approach is then acceptable. This is important as some samples may deceive researchers which then lead to wrong conclusions.

For the sake of this study, a cross validation is used to confirm that the sample is representative for the population. Cross validation is a method where two samples are selected randomly from a population and then their descriptive statistics are compared against each other. If their statistics were close, then the sample is representative. Otherwise the sample is not representative.

Figure 3.3 shows the method used to sample the dataset for the sake of this study. Three samples were selected randomly from the population. The first sample was 50 machines, the second sample was 40 machines and the third sample was also 40 machines. Although one sample is usually considered enough and common practise, three samples were used in order to cross validate the sample representation. In cross validation two samples are enough, yet in this study three samples were used. This is to enhance the validation where the more samples used the more firm result generated. After measuring the statistics of the three samples, the samples then compiled to gather to form the sample population of the study.

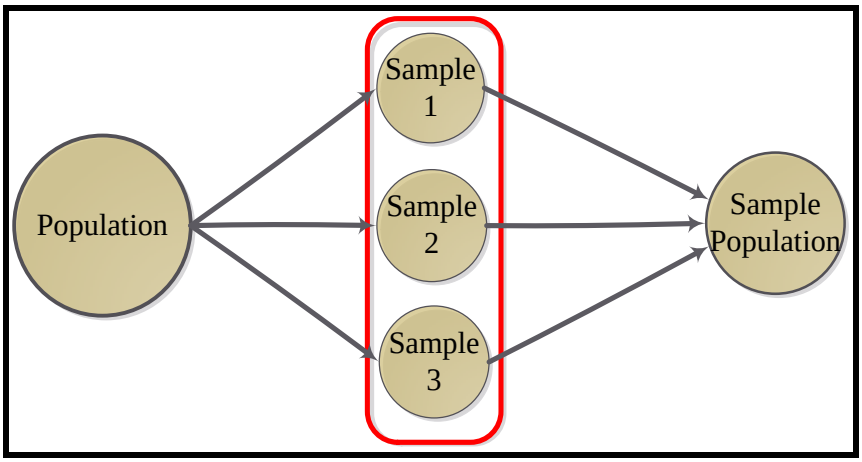

Figure 3.3: Sampling Technique and Cross Validation.

Since the investigation is about Multi-Tenancy and based on the definition on section 2.4.1, VMs are important for this investigation and what shape Multi-Tenancy. Then, the standard deviation[10] of the number of VMs in the dataset will be used to investigate the sample similarity. Table 3-4 shows the standard deviation of the three samples and the sample population. This tells us that the variation of the VMs number in the samples is similar across each platform in the dataset which support that any conclusion drawn from these samples could be valid on the population of the dataset.

10 Is a statistical measure reflects the degree to which the values in a distribution differ from the mean.

Table 3-4: Standard Deviation of the samples.

Platform	Sample 1	Sample 2	Sample 3	Sample Population
1	231.40	162.86	211.16	216.30
2	942.37	985.85	1009.00	976.34
3	637.46	708.96	643.93	648.86
4	605.23	607.32	587.56	625.14

Table 3-5: Average Readings for Platform 1 Attributes.

Sample	Multi-Tenancy %	No of Tasks	Duration (days)	No of Machines
Sample population	6.85	1260.16	24.06	33
Sample 1	6.99	1358.17	22.15	13
Sample 2	6.54	1170.67	23.74	10
Sample 3	6.96	1223.10	26.65	10

Table 3-6: Average Readings for Platform 2 Attributes.

Sample	Multi-Tenancy %	No of Tasks	Duration (days)	No of Machines
Sample population	286.88	2360.42	22.32	33
Sample 1	356.03	2568.77	24.37	13
Sample 2	246.80	2000.70	18.96	10
Sample 3	237.05	2449.30	23.03	10

Table 3-7: Average Readings for Platform 3 Attributes.

Sample	Multi-Tenancy %	No of Tasks	Duration (days)	No of Machines
Sample population	217.31	1910.44	23.45	32
Sample 1	98.76	1859.17	19.58	12
Sample 2	250.30	1823.80	24.90	10
Sample 3	326.58	2058.60	26.66	10

Table 3-8: Average Readings for Platform 4 Attributes.

Sample	Multi-Tenancy %	No of Tasks	Duration (days)	No of Machines
Sample population	60.27	1103.56	13.38	32
Sample 1	21.85	877.08	8.94	12
Sample 2	149.95	1423.00	21.53	10
Sample 3	16.69	1055.90	10.57	10

3.2.3 Statistics of Multi-Tenancy

Multi-Tenancy as the primary focus of this research project has not yet been investigated and quantified to the best of our knowledge. Also, as illustrated in the preceding chapters, there is a need to quantifying Multi-Tenancy. So, in this section, the methodology used to extract Multi-Tenancy information from Google data set will be demonstrated, and then the results will be presented.

Methodology

The methodology presented in section 3.2.1 will be used here, and further details on how the results were obtained will be given. Moreover, the row data and database phases will be explained. First, the modification starts from the Mission stage where Table 3-9 shows the attributes needed to form the mission catalogue. Second, the Job ID is to identify users where a unique user submits each task, and also, is used as a foreign key to join to the Task Events table and Resource Usage table. Third, the Task Index is used to identify every VM within a job where the combination of both Job ID and Task Index will represent the unique id for the VM. Fourth, the Machine ID is used to identify the PM hosting the VM- this is important to capture the Multi-Tenant VMs. Fifth, the Event type is the kind of the VM, for this mission, the focus will be on the hosted VMs (for example the task event type 1, the scheduled tasks). Sixth, the job event type 5 will be captured in this catalogue to be used for a further investigation related to the attack model. Seventh, the start and end time for each VM is obtained to identify the Multi-Tenant VMs.

Table 3-9: Multi-Tenancy Mission Catalogue

Attribute	Original Table
Job ID	Task Events
Task Index	Task Events
Machine ID	Task Events
Event type	Task Events
Start time	Resource Usage
End time	Resource Usage

After the extraction of the catalogue from the database in CSV format, a random sample of machines is selected from different platforms directly from the database (Machine Events table). Then, a mini catalogue for each machine is developed from the Multi-Tenancy mission record with only the Job ID, task index, start time, and end time using a SELECT statement. After that, the output is stored in a CSV format.

Consequently, for each mini catalogue, a sort command using Excel is done by sorting the start time in the ascending order. Next, the time is converted into minutes, as it is originally in microseconds. Consequently, this done by dividing the time by 600,000,000 as data set starts from 600 µs.

Notably, the table is ready to be analysed by R to identify the Multi-Tenant VMs and find out the percentage of Multi-Tenancy in each machine. The following procedure and codes are used to calculate the Multi-Tenancy percentage in each machine:

```
// To load the data into R:
temp = read.table("FILE_NAME",h=F,skip=1)
head(temp)
// Then, the start and end time will be stored in x.
x = temp[,c(2,3)]
// The following loop will be used to find the Multi-
Tenant VMs by creating a matrix and subtract the
start time of the second entry with the previous end
time. If the result is less than 0, then the VMs are
Multi-Tenant.
check = matrix(nrow=nrow(x),ncol=nrow(x))
for(i in 1:nrow(x)){
```

```
print(i)
for(j in 1:nrow(x)){
     check[j,i] = x[j,1]-x[i,2]
  }
}
// Since a matrix is used, the diagonal must be
zeroed to avoid counting a false measurement.
for(i in 1:nrow(x))check[i,i]=0
// Then, the lower half of the matrix is stored in a
vector to avoid duplicate readings.
temp = c()
for(i in 1:ncol(check)){
print(i)
temp=c(temp,check[i:nrow(check),i])
}
// Finally, the percentage of Multi-Tenancy in a
given machine is calculated as follows:
N = nrow(check)
inters = sum(temp<0)
perc = inters/N
perc*100
// The results then is stored in an Excel sheet
contain the Machine ID, Platform ID and the Multi-
Tenancy percentage.
```

Other measures such as the number of VMs hosted in the PM are calculated by a COUNT command for the individual VMs in the mini catalogue. Also, the duration where the PM is hosting VMs is calculated by subtracting the end time (for the last row) from the start time of the first row. Finally, the number of machines is known by running a COUNT command for each platform in the database directly (Machine Events table).

Results

Figure 3.4 shows the Multi-Tenancy percentage per machine for platform 1. The X axis is the machine ID, and the Y axis is the Multi-Tenancy in percentage. The methodology detailed in section 3.2.1 is used to generate this graph. What does it mean that machine ID (3631294705) scores about 5% of

Multi-Tenancy? It means that 5% of the up and running time of this machine, two or more VMs belong to different customers occupied it. From the figure it is obvious that no machine exceeded the limit of 10%.

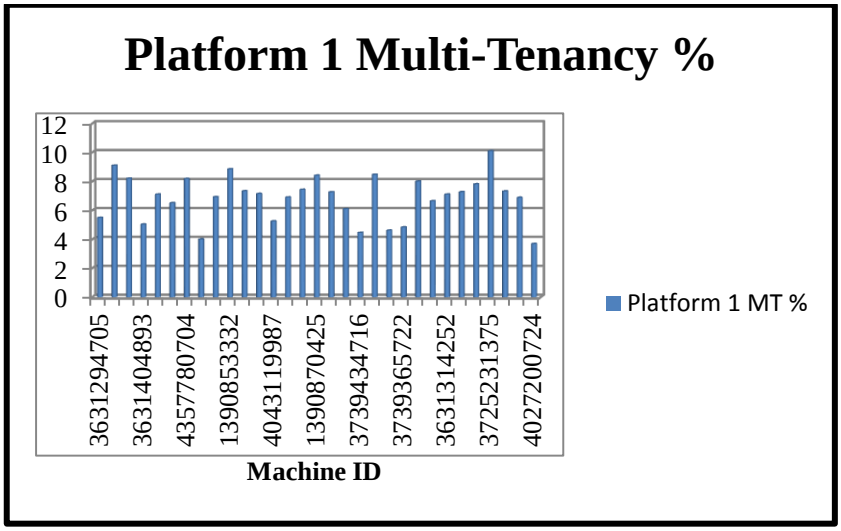

Figure 3.4: Multi-Tenancy % of Platform 1

As the previous figure, Figure 3.5 captures the Multi-Tenancy percentage of platform 2. Whereas, Figure 3.6 and Figure 3.7 captures the Multi-Tenancy percentage for platform 3 and platform 4 respectively. Unlike platform 1, platform 2, 3 and 4 are fluctuating, and no clear pattern captured.

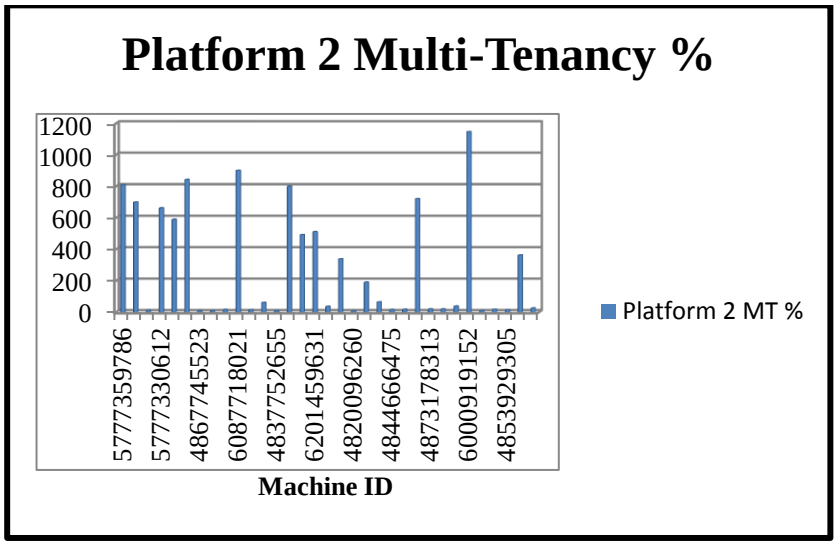

Figure 3.5: Multi-Tenancy % of Platform 2

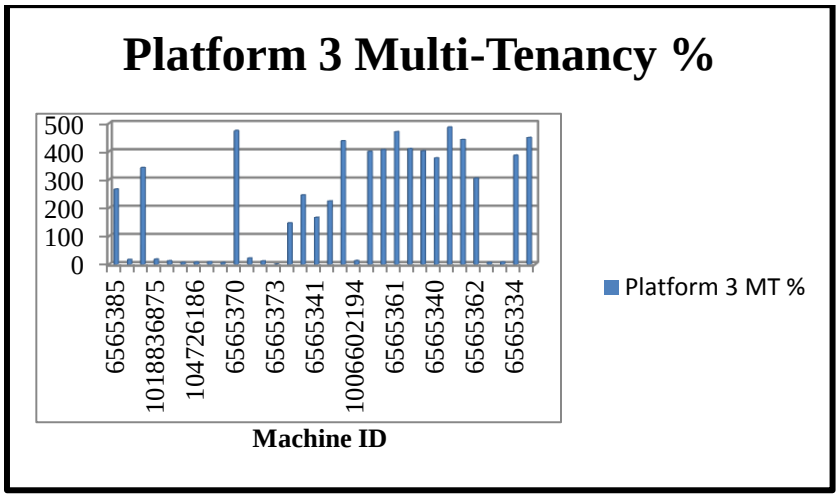

Figure 3.6: Multi-Tenancy % of Platform 3

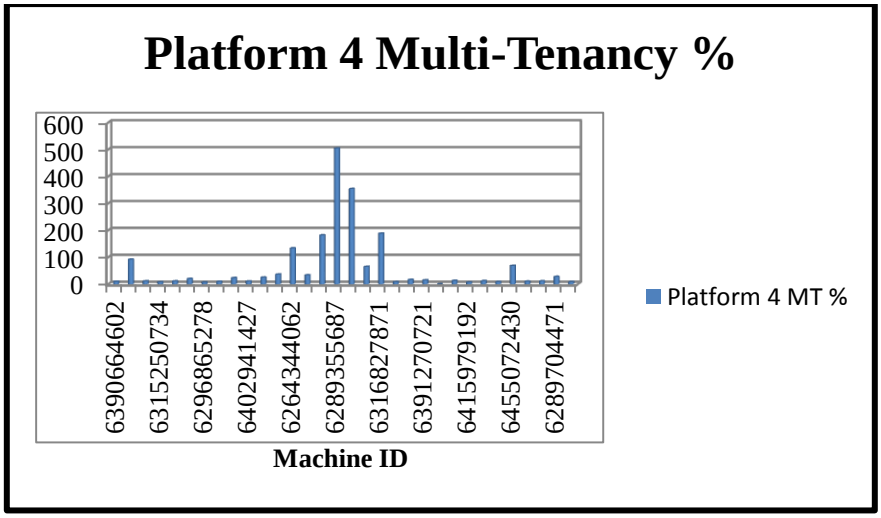

Figure 3.7: Multi-Tenancy % of Platform 4.

Figure 3.8 shows the three samples of platform 1 and their readings as a spider chart. The attributes are the Multi-Tenancy percentage, number of VMs, the number of PMs, and the durations of days. This data is corresponding to the readings in Table 3-5. Besides, the spider charts help to highlight any obvious correlation observed. Indeed, it is evident from the figure that the three samples are almost identical in the statistical readings.

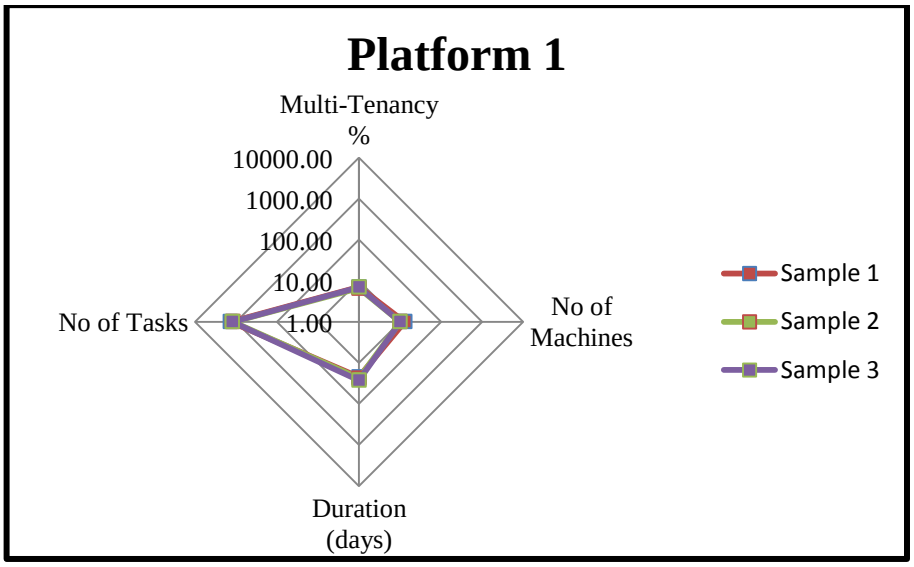

Figure 3.8: Samples' Measures for Platform 1

Figure 3.9 visualises the readings in Table 3-6 which is the readings of the three samples of platform 2. Similar to platform 1, the samples of platform 2 show a high similarity in the statistical features.

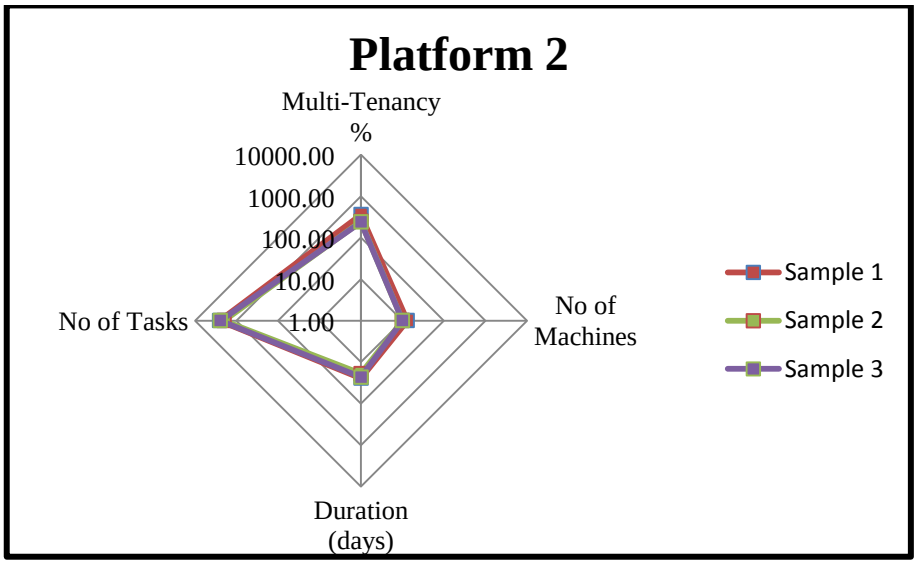

Figure 3.9: Samples' Measures for Platform 2

Figure 3.10 and Figure 3.11correspondence to Table 3-7 and Table 3-8 respectively. Figure 3.10 visualises the readings of platform 3, whereas Figure 3.11 visualises the readings of platform 4. It is evident that the samples of platform 3 are aligned together on some VMs, many PMs and duration, yet have a clear mismatch in Multi-Tenancy percentage, especially for sample 1. Unlike platform 1, 2 and 3, platform 4 shows a mismatch in the Multi-Tenancy percentage and the duration readings between the three samples.

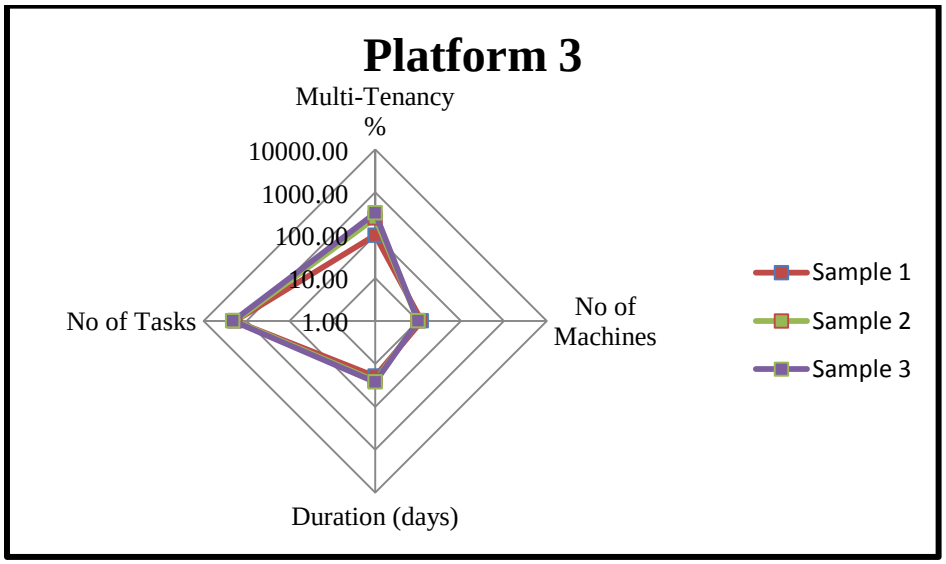

Figure 3.10: Samples' Measures for Platform 3

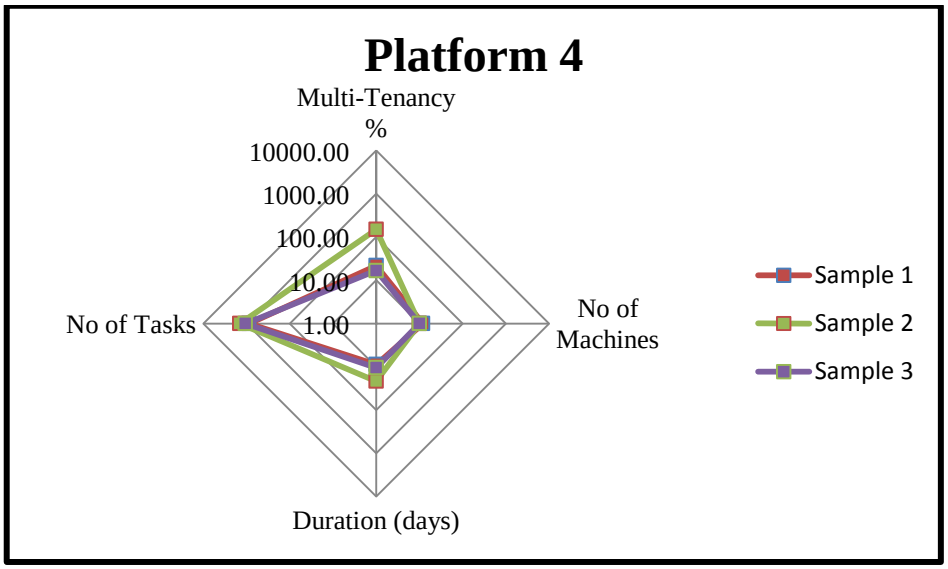

Figure 3.11: Samples' Measures for Platform 4

Figure 3.12 shows the platform ID against the Multi-Tenancy percentage per machine. That number aggregates Figure 3.4, Figure 3.5, Figure 3.6 and Figure 3.7 to visualise the pattern of each platform for further analysis.

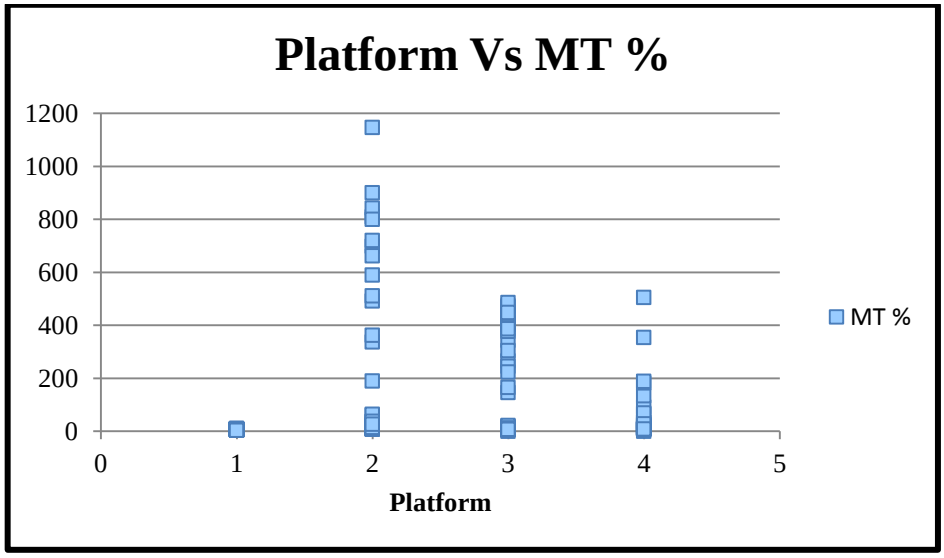

Figure 3.12: All Platforms Vs MT%.

Figure 3.13 examines the platform ID against the number of VMs hosted per machine. This highlights the clustering and shows the distribution of VMs.

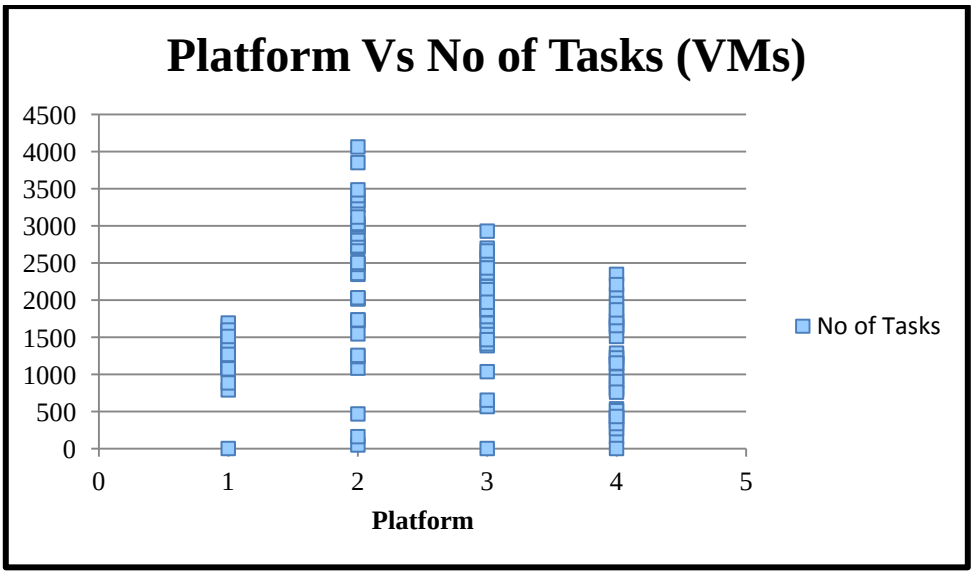

Figure 3.13: All Platforms Vs Number of VMs

Figure 3.14 shows the relationship between platform ID and the duration in days. The duration of PM being utilised by VMs, the maximum duration is 29 days as it is the length of the dataset period. Most of the machines are hosting VMs for two week or 29 days, with fewer number served below five days.

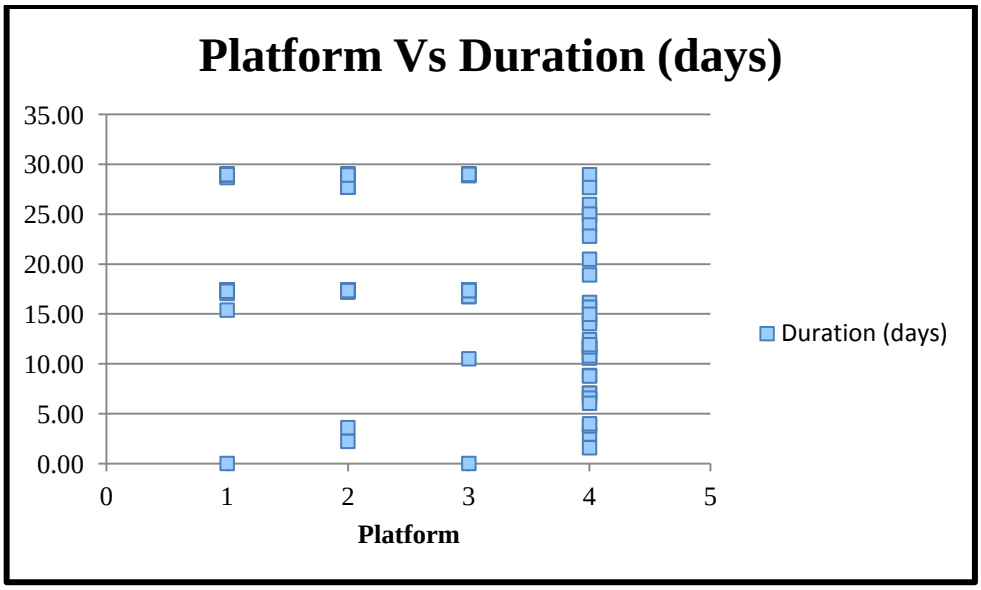

Figure 3.14: All Platforms Vs Duration

Figure 3.15 shows the number of PMs per platform. It is noticeable that platform 3 acquire most of the infrastructure's machines.

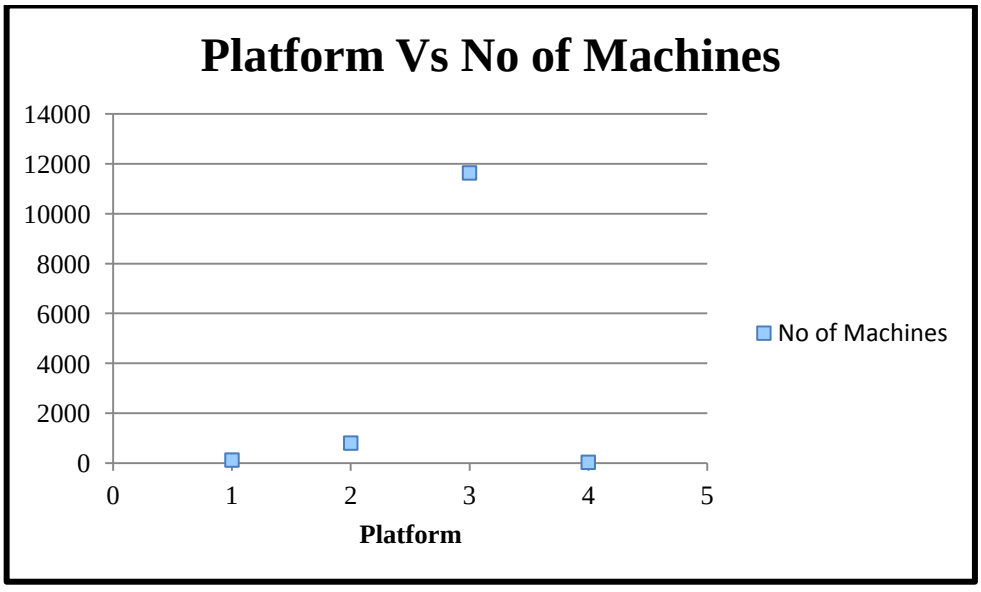

Figure 3.15: All Platforms Vs Number of Machines

Figure 3.16 shows the average of Multi-Tenancy percentage per platform. This helps in ranking the platforms according to their score. From the figure, it is obvious that platform 2 and 3 are close in the overall average of Multi-Tenancy. Then platform 4 comes next in the ranking and platform 1 is lowest among them in terms of Multi-Tenancy percentage.

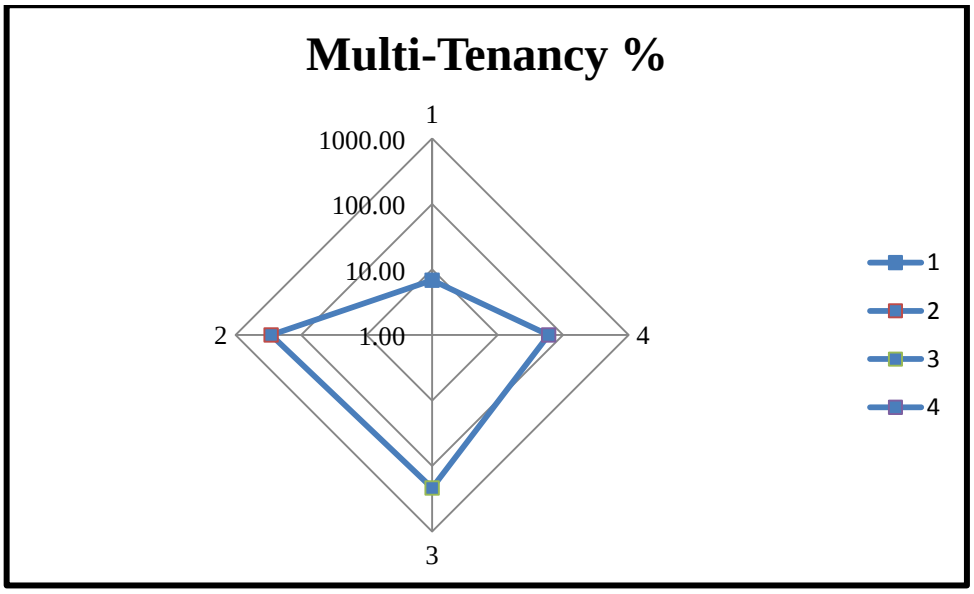

Figure 3.16: Spider Chart for MT%.

Figure 3.17 shows the average of number of VMs hosted by PMs per platform. From the figure, it is obvious that platform 2 is the highest and platform 4 is the lowest.

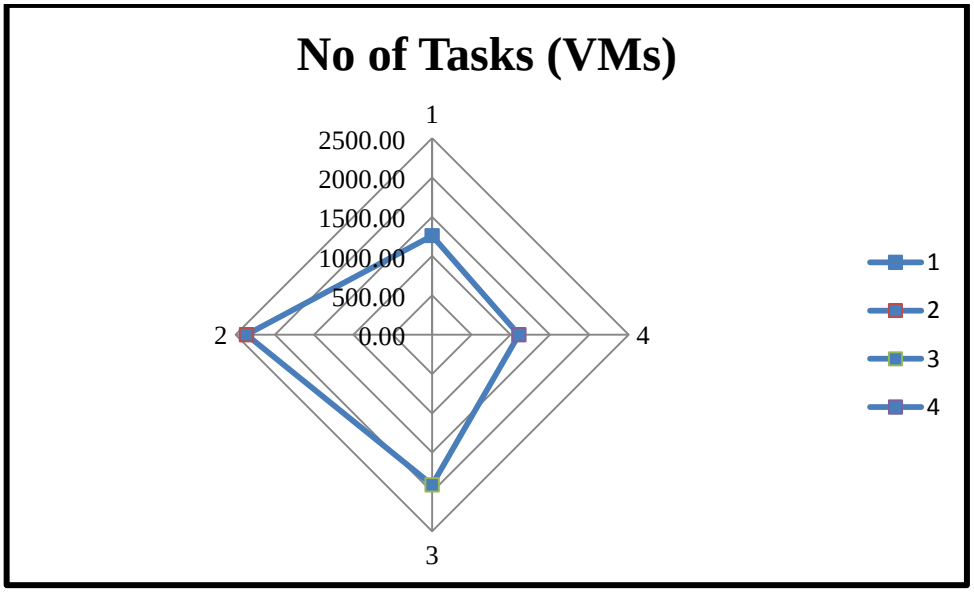

Figure 3.17: Spider Chart for Number of Tasks.

Figure 3.18 shows the average of duration per platform. From the figure, it is obvious that platform 1, 2 and 3 are close in the overall average of duration with just about 25 days. Whereas, platform 4 scores about 15 days which half the time of the dataset.

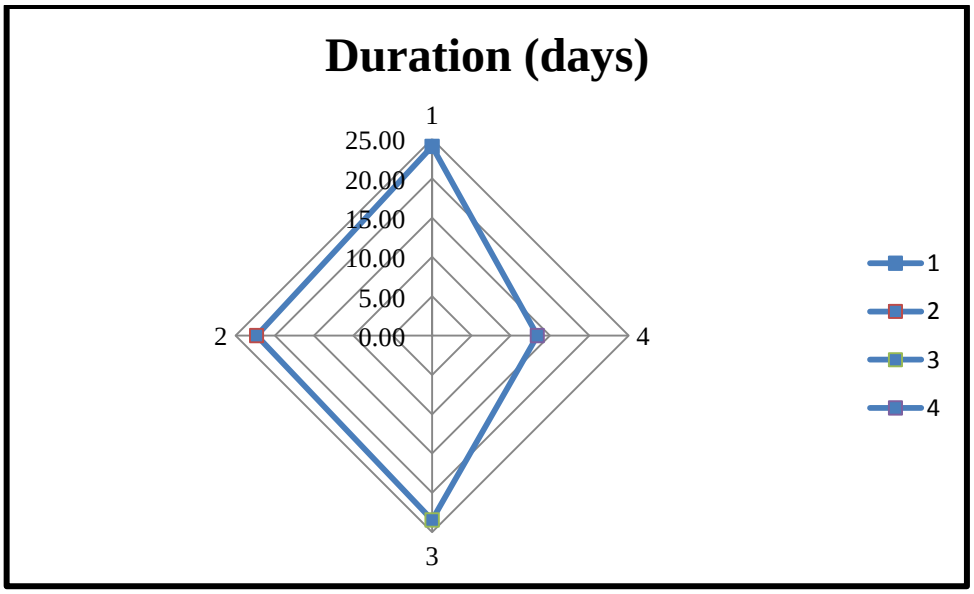

Figure 3.18: Spider Chart for Duration.

Figure 3.19 shows the number of PMs per platform. Platform 3 is the largest with more than 11000 machines. Whereas, platform 4 is the smallest with only 32 machines.

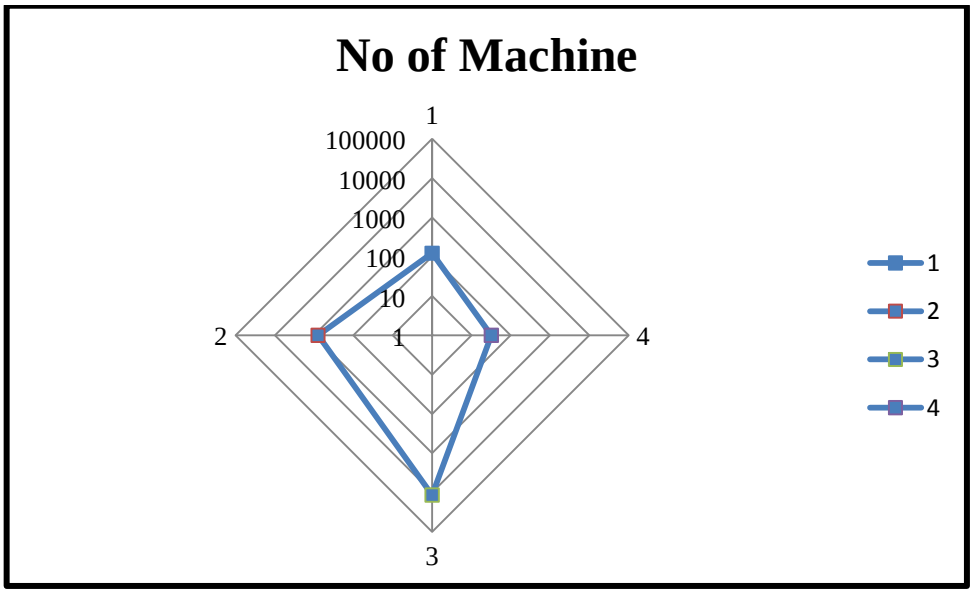

Figure 3.19: Spider Chart for Number of Machines.

Figure 3.20 aggregates the overall measures from the four preceding figures (Figure 3.16, Figure 3.17, Figure 3.18 and Figure 3.19) in order to visualise the correlation between the different attributes. There is a strong correlation between Multi-Tenancy percentage and number of VMs. Also, there is a strong correlation between the number of PMs and the duration.

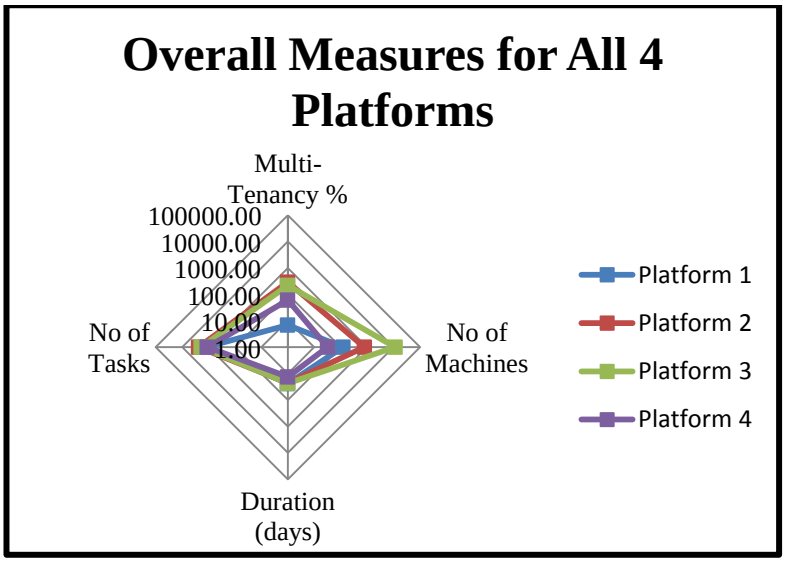

Figure 3.20: Spider Chart for Overall Measures for all Platforms.

3.3 Platform Analysis

As seen in section 3.2, there are different attributes in the system along with the Multi-Tenancy percentage. Thus, to know what quality affects the Multi-Tenancy, a correlation test is done. Spearman's test is used as the data is considered nonparametric, two tailed test is run as there is no specification on the direction of the effect.

Table 3-10: Correlation Analysis.

Correlations							
			Platform ID	Multi-Tenancy %	No of Tasks	Duration (days)	No of Machine
Spearman's rho	Platform ID	Correlation Coefficient	1.000	.331**	-.084	-.421**	-.200*
		Sig. (2-tailed)	.	.000	.343	.000	.024
		N	128	128	128	128	128
	Multi-Tenancy %	Correlation Coefficient	.331**	1.000	.595**	.248**	.318**
		Sig. (2-tailed)	.000	.	.000	.005	.000
		N	128	128	128	128	128
	No of Tasks	Correlation Coefficient	-.084	.595**	1.000	.428**	.518**
		Sig. (2-tailed)	.343	.000	.	.000	.000
		N	128	128	128	128	128
	Duration (days)	Correlation Coefficient	-.421**	.248**	.428**	1.000	.424**
		Sig. (2-tailed)	.000	.005	.000	.	.000
		N	128	128	128	128	128
	No of Machine	Correlation Coefficient	-.200*	.318**	.518**	.424**	1.000
		Sig. (2-tailed)	.024	.000	.000	.000	.
		N	128	128	128	128	128

**. Correlation is significant at the 0.01 level (2-tailed).

*. Correlation is significant at the 0.05 level (2-tailed).

3.4 User to Platform Behaviour

The behaviour of the attributes interactions is important as some responses may increase the risk of vulnerability by expanding the attack surface. In the following different scenarios are captured in the data.

Figure 3.21 shows two cases represent different behaviours caught in the dataset. Example 1 shows a customer submitting 23 VMs, and all of them are hosted on platform three, but in different machines. On the other hand, case 2 shows a client sending 70 VMs, and most of them are organised by platform three except for three VMs. Two of these VMs are hosted by platform 2 in two different PMs, and the third VM is organised by platform 4.

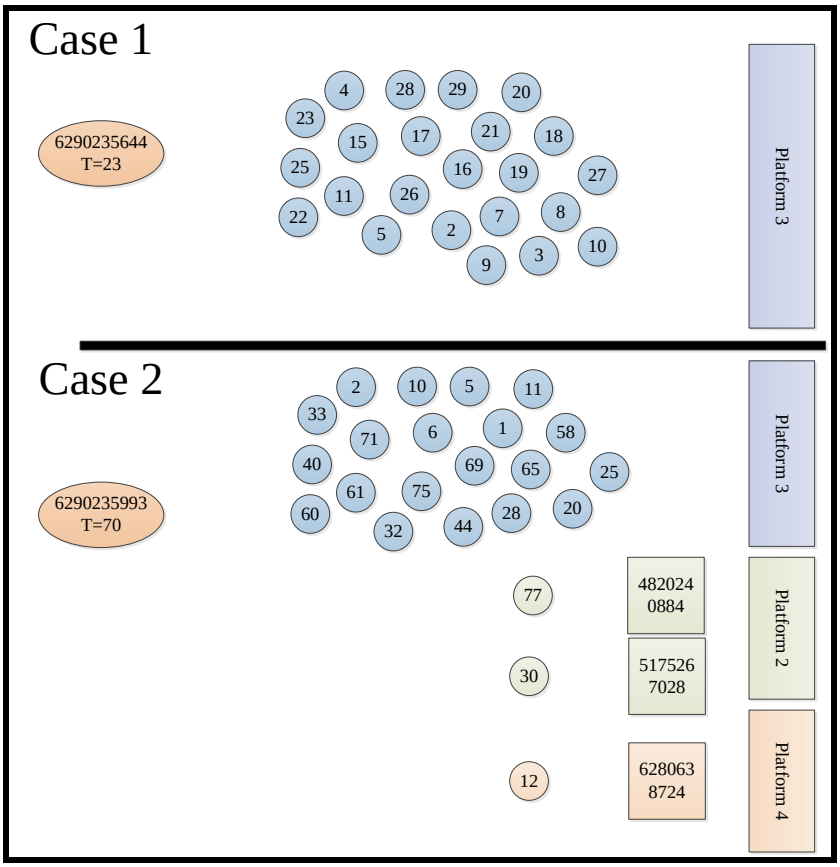

Figure 3.21: Different Cases of Job to Platform Allocation.

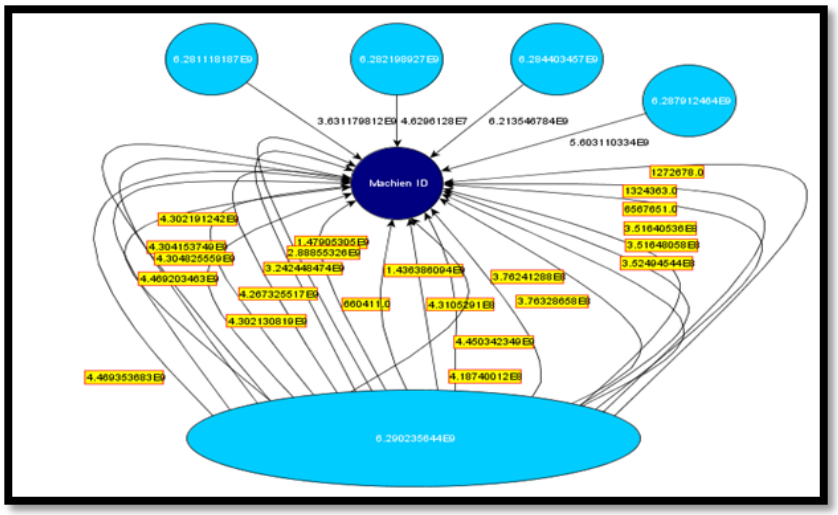

Figure 3.22: Job to Machine Allocation.

3.5 Attack Model

The thesis uses an attack model for specific vulnerabilities. Indeed, the nature for specific vulnerability is that there exist numerous possibilities that can be utilized. Besides, the attacks vary in the sense of their behaviour. For instance, for a denial of service attack, it is easy to investigate since any increase in traffic can raise eye-brows. Likewise, it is easy to identify the viruses because of the unique signatures; whereas, it is complicated to detect iFrame attacks- for instance, the vulnerability emanates inside an HTML code as a frame to collect credit card data.

Therefore, the attack model proposed to take advantage of Multi-Tenancy as a modified version of the attack model used in [2]. Moreover, the attack model proposed is generated in three phases as shown in Figure 3.23:

- In phase one, attacker register with the Cloud provider as a normal customer. This phase is mandatory for any customer needs to hire a Cloud services. As a natural step, there is no need to prove it as it is self-proved phase.
- In phase two, the attacker gathers information about the allocation technique and the Cloud infrastructure where network probing is utilized. The attacker can make sense of the allocation technique simply by requesting resources and then releasing them – hence brute forcing. This action will give the attacker knowledge of the allocation technique

for more targeted attack. Moreover, the attackers can take advantage of the information revealed by the Cloud provider about their infrastructure or any kind of systems or techniques they are using. After that, the attacker can utilize brute force techniques to generate VMs in order to achieve Multi-Tenancy. This phase has been proved in (ibid) where both network propping and brute forcing are considered a general attack mechanisms and relatively easy.

- In phase three, after the attacker achieved Multi-Tenancy, a side channel attack was generated to extract the victim's data. Different side channel attacks could be used as mentioned in section 2.6, and such attacks were proven by researchers in Cloud Computing [48]–[53]. For example, [52] lunched a successful timing attack taking an advantage of Multi-Tenancy by measuring the I/O clock for VMs. In order to mitigate two types of timing side channels attacks, 60% of the Cloud infrastructure must be sacrificed. Another form of side channel attacks is memory attacks, where [49] described a Cloud Internal Denial of Service attack (CIDoS) and memory side channel attack is used as a means of communication.

This attack model is designed to take advantage of Multi-Tenancy; thus, without Multi-Tenancy, the attack will not be applicable.

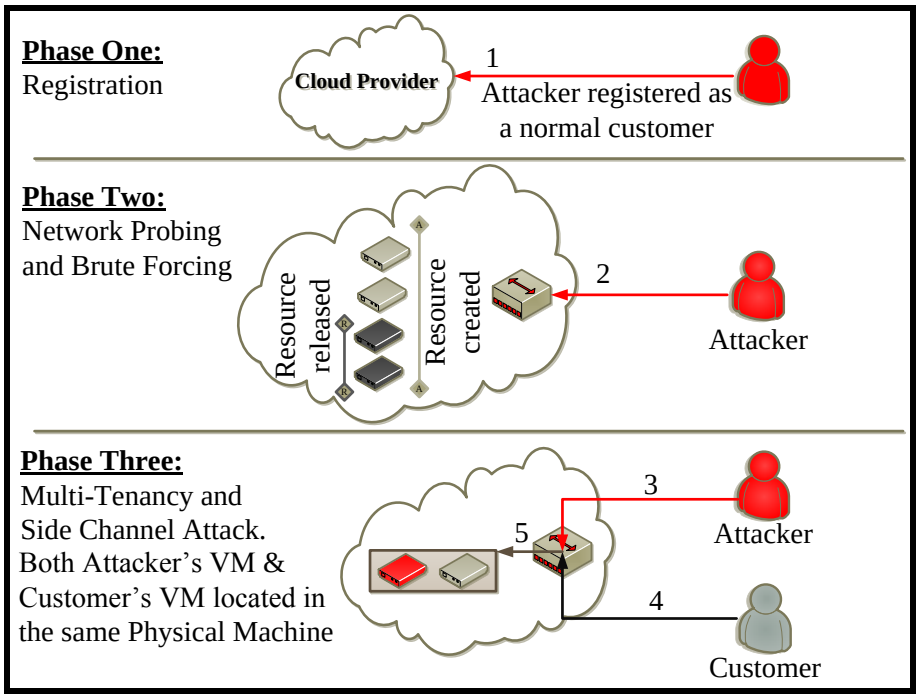

Figure 3.23: Attack Model Visualisation.

3.6 Security Trade Offs

In order to enhance the security of Multi-Tenancy other attributes may be affected. In this section, the different measures in the system will be tested against the threshold of Multi-Tenancy percentage set by a Cloud provider.

Figure 3.24 shows the number of VMs must be terminated in order to preserve a given Multi-Tenancy percentage. The general trend is the more the threshold is relaxed the lower the number of VMs needed to be terminated. The terminated VMs are considered waste. It is expected that for platform 2 and 3 to reach zero wasted VM, the threshold must be more than 200%. This is because platform 4 reached zero wasted VM just about 60% which is its Multi-Tenancy percentage average.

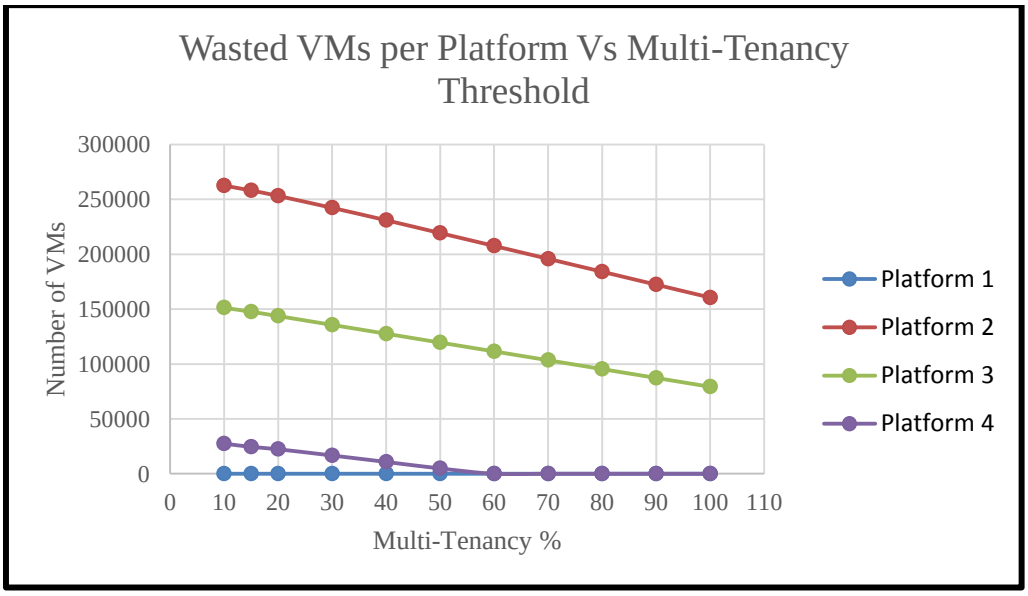

Figure 3.24: Wasted VMs per Platform.

Figure 3.25 presents the PMs that exceeded the corresponding Multi-Tenancy percentage. This is important because it may give us insights on the behaviour within the platform specially when combined with other figures such as Figure 3.24. platform 3 which is the largest in terms of number of PMs and Multi-Tenancy percentage steady still from the point of 30% and onwards.

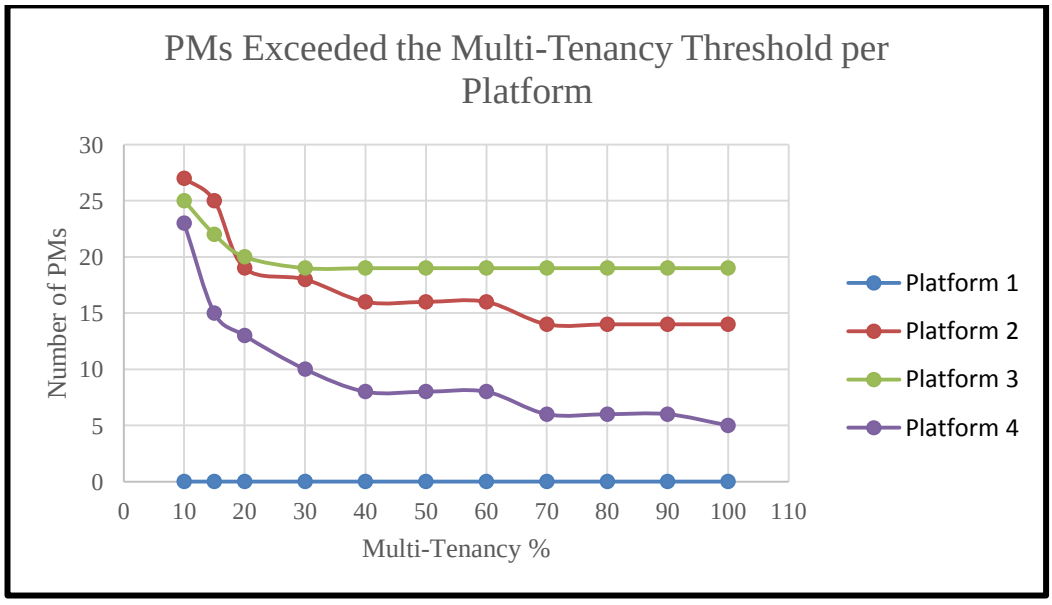

Figure 3.25: Number of PMs Exceeded the Multi-Tenancy Threshold.

Figure 3.26 calculates the needed PMs in order to accommodate the wasted VMs. This is important as it is will be used as decision making attribute. This figure reflects the amount of investment to upgrade the infrastructure.

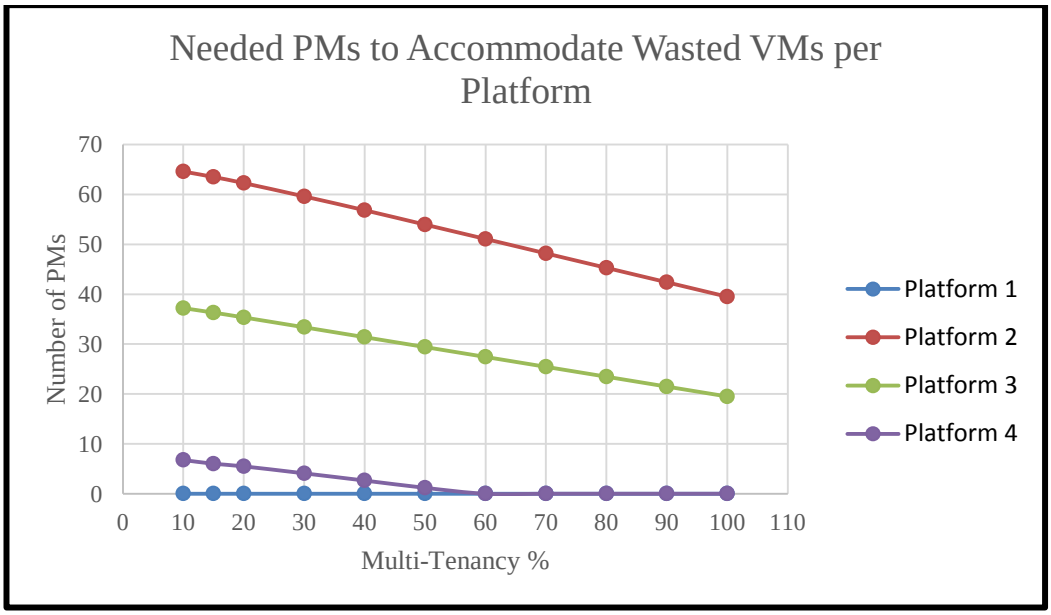

Figure 3.26: Number of PMs needed to Accommodate Wasted VMs.

Figure 3.27 presents the difference in number of wasted VMs when cross platforms allocation is enabled and when it is not. Instead of upgrading the infrastructure, some of the wasted VMs could be accommodated in different platforms without affecting the Multi-Tenancy percentage. This is a critical point as it may increase the attack surface.

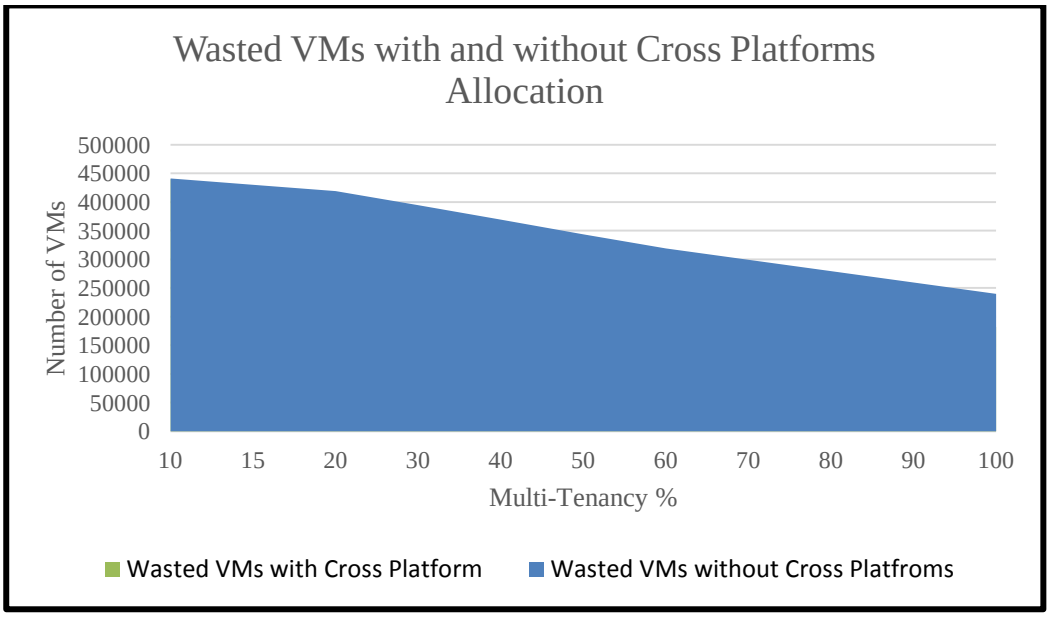

Figure 3.27: The Effect of Cross Platforms Allocation on the Number of Wasted VMs.

Figure 3.28 highlights the difference between the needed PMs to accommodate wasted VMs and the PMs that exceeded the Multi-Tenancy percentage for a given Multi-Tenancy percentage. It is noticeable that the needed PMs are always bigger than the affected PMs.

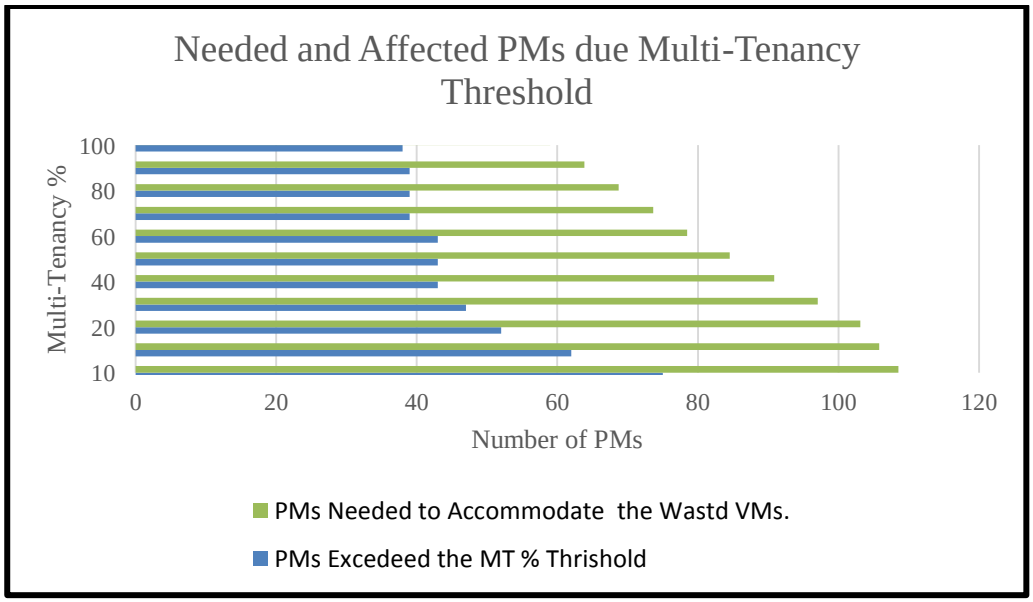

Figure 3.28: Needed and Affected PMs Against Different Multi-Tenancy Threshold.

Figure 3.29 shows the number of affected users by restricting the percentage of Multi-Tenancy. This is to consider the economical aspect in the decision-making process and security trade-offs. It is noticeable that with each 5% relaxation on Multi-Tenancy, 50% save on the users is gained.

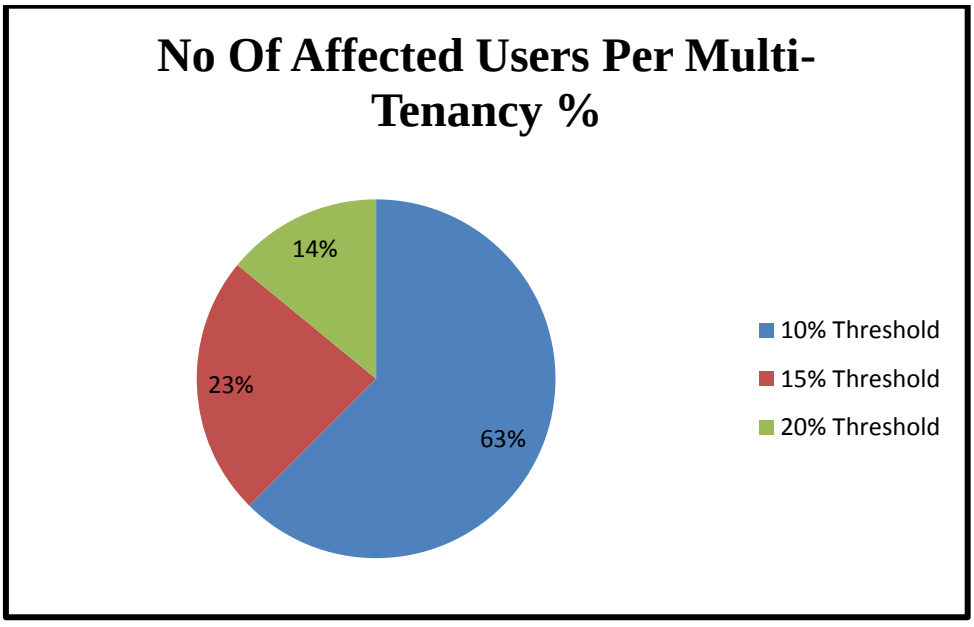

Figure 3.29: Number of Affected Users by Multi-Tenancy Threshold.

3.7 Summary

This chapter presented the experimental results for the research project and illustrated the methodology followed to get these results. Importantly, the results of quantifying Multi-Tenancy were shown. Moreover, the different attributes captured in the dataset were tested to highlight their correlations. Also, different behaviours linked to the allocation features and attack model were presented.

Moreover, the methodology, sampling technique, and statistics of Multi-Tenancy were submitted and illustrated in section 3.2. Additionally, the platform analysis where the correlation results of different attributes were shown in section 3.3. Again, different behaviours of the customer to VMs interaction were presented in section 3.4; whereas section 3.5 showed the results of the attack model. Finally, section 3.6 presented the trade-offs of securing Multi-Tenancy and the affected attributes.

Notably, in section 3.6, Figure 3.24 displayed the wasted VMs per platform against different Multi-Tenancy thresholds. Overall, the general trend is that the higher the Multi-Tenancy threshold, the lower the number of sacrificed VMs. Although the Multi-Tenancy threshold reached 100%, platforms 2 and 3 needed to sacrifice their VMs to meet the threshold. On the other hand, platform 1 remained unaffected by the different Multi-Tenancy threshold. Despite platform 3 having the highest Multi-Tenancy percentage, platform 2 sacrificed more VMs to meet the limits. Specifically, the statistics revealed the highest number of VMs sacrificed was 262,530 VMs at 10% threshold for platform 2. Whereas, platform 3 sacrificed 151,253 VMs at the same limit. At 100% threshold, platform 2 sacrificed 160,551 VMs and platform 3 sacrificed 79,257 VMs.

Moreover, Figure 3.25 illustrated the PMs that exceeded the Multi-Tenancy threshold per platform. Interestingly, the number of sacrificed VMs in platform 3 was lower than platform 2, and the affected PMs in platform 3 was higher than platform 2. Moreover, platform three still needed to sacrifice VMs to meet each Multi-Tenancy threshold, yet the affected PMs were steady from and after the 30% threshold. Accordingly, the observation highlights that individual

PMs are oversaturated with Multi-Tenancy. In other words, minimising the number of sacrificed VMs may not reduce the affected PMs. A similar scenario was seen in platform 4 where the number of sacrificed VMs was equal to zero from and after 60% threshold, yet the number of affected PMs was continued to be reduced until it reached 5 PMs.

Unfortunately, the sacrificing of VMs affects the availability of service or the profit of Cloud provider; however, it depends on the Cloud provider's actions in such a situation. Therefore, if the Cloud provider decides to keep the number of customers, then the availability of the service will be affected. Whereas, if the service provider decides to reduce the number of clients to maintain availability, then the profit will be impacted by the number of consumers decreases.

Figure 3.26 displayed the needed PMs to accommodate the sacrificed VMs per platform. Moreover, it highlights the opportunity of maintaining the same number of customers and availability at the same time. Again, the general trend of Figure 3.26 followed the behaviour of Figure 3.24 where the each of the new PMs was assumed to serve up to 4,065 VMs. Nonetheless, it was a conservative assumption based on the Google dataset as there was only one PM which reached that number of VMs. Figure 3.27 displayed the difference in the number of sacrificed VMs with cross platform allocation against allocation within the platform. It is evident that cross platform distribution saves VMs spicily if the Multi-Tenancy threshold is relaxed, but it brings security.

Likewise, Figure 3.28 revealed the needed PMs to accommodate the sacrificed VMs against the PMs which exceeded the Multi-Tenancy threshold for the entire infrastructure (for example, platforms 1, 2, 3, and 4). Although the number of PMs exceeded the threshold of 70%, 80%, and 90% was steady, the needed PMs to accommodate the wasted VMs was reduced. Hence, it reflects the amount of cost reduction when cross platform allocation is enabled. Apparently, the cost is represented as new servers are needed to scale up the infrastructure.

Furthermore, Figure 3.29 illustrated the number of customers affected by the Multi-Tenancy threshold - 4% of the customers were affected when the

Multi-Tenancy threshold was 10%. Consequently, a 5% increase in the Multi-Tenancy threshold saved 50% of the affected customers. With 20% Multi-Tenancy threshold, only 1% of the customers were affected which is about 10 customers out of 925 total clients.

To sum up, enhancing the security of Multi-Tenancy in to preserve confidentiality will affect either availability, profit, or increase costs (capital investment, maintenance, and operations). Depending on the Cloud provider strategy, one or more attributes may be affected. The decision is not easy since there is a consideration for some servers, customers, and VMs submitted. Hence, any change of the parameters leads to alterations in the Cloud provider’s decision.

Chapter 4 Multi Tenancy in Clouds: Threats and Attacks

4.1 Threat Model

The threat model is a model describing the environment to highlight the vulnerabilities and risks associated with them. The goal of this threat model is to highlight the required elements to take advantage of Multi-Tenancy and what is the needed environment for Multi-Tenancy to occur. As Multi-Tenancy occurs in shared Clouds and can only be a threat if the Cloud is untrusted, our threat model will describe the public Clouds as they considered untrusted shared Clouds as mentioned in Table 2-1. Although hybrid Clouds could be regarded as untrusted, the untrusted part is the part that is hosted in the public Clouds. As a result, public deployment for Clouds is the model under investigation.

As there are at least three service models, IaaS is the service model that is under investigation. As mentioned in section 2.4.1 Multi-Tenancy is the case when two or more virtual machines (VMs) belonging to different customers are sharing the same physical machine (PM). This situation can happen only on the IaaS, as the client is requesting VMs and pay for the resource capacity utilised by it.

Based on the above, IaaS public Clouds are the environment that is under investigation. As equation (1) in section 2.4.1 shows that Multi-Tenancy is a result of virtualization technology and allowing resource sharing, an in-depth description of these techniques is required to clarify the threat model.

From section 2.1, we know that virtualisation has two components virtual machine manager (VMM) and VM. As VMM could be used to lunch attacks as mentioned in section 2.6 in the case of VM escape attack, VMM in this threat model is assumed to be secure and cannot be hijacked. Because hijacking VMM is a problem related to virtualisation technology and not limited to Cloud Computing. In addition, such attacks related to VMM do not require Multi-Tenancy in order to take place. To clarify that, it could happen that a Cloud provider disables Multi-Tenancy by locating each customer's VMs in a separate PM. In this situation, there is no Multi-Tenancy, yet the virtualisation is enabled to feature other functions or gain extra benefits. Here the customer could hijack the VMM to gain extra privileges. As a result, the assumption of secure VMM is

justified since the main vulnerability is Multi-Tenancy and the goal is to highlight its risks. Since VMs are deployed over PMs where part of the PM resources (i.e. CPU, RAM and disk) is allocated to the VM, the assumption that VMs have access to RAMs, CPUs and disks is a realistic assumption.

Moreover, resource sharing as a common practice of Cloud providers is considered a risky environment, especially when combined with virtualisation – hence, Multi-Tenancy. If the situation of two VMs coexist in the same PM and in the same time (i.e. Multi-Tenant), and both can access the PM resources in the same time. Then, the assumption of launching a side channel attack such as memory attacks is considered a realistic assumption. Such an attack could not be possible if the resource sharing is not allowed. Also, if virtualisation is not implemented the attack will not happen. This is an important point as if resource sharing could be enabled on other forms such as time sharing where the PM is reused by different customers each given time and not in the same time. Therefore, virtualisation is important to allow same time coexistence VMs over a PM.

Based on the above, the assumption of both virtualisation and resource sharing are enabled in the Cloud is made which is also a common practice of Cloud providers. Such setup creates Multi-Tenancy which raises concerns on confidentiality and integrity of data.

To summaries the threat model, the environment that is considered a threat and highlights the risks of Multi-Tenancy are IaaS public Clouds which enable resource sharing over virtualisation. The following assumptions were made:

- The VMM is secure.
- VMs granted access to PMs resources.
- The customers do not know the infrastructure (i.e. allocation mechanism and any underlying technology or topology) of the Cloud provider.

4.2 Reconstructing the Attack model from the Google Dataset

The attack model presented in section 3.5 consisted of three phases as described. As phase one is self-proof, there is no need to proof it. Whereas, step three is difficult to investigate without direct access to the PM which is data not revealed in the Google dataset. Still, there is sufficient data to sense phase two of the attack model using the data set. Therefore, to find out a pattern of the attack model reflected on the dataset, we focuses on phase two. Specifically, the brute forcing on Google's dataset will be investigated.

So, to sense a brute forcing behaviour, the killed tasks in the dataset were targeted. Notably, nine event types can tag any task. Between the nine event types, a killed task event represents a task cancelled by the customer or a driver program. Moreover, the killed task event is the only task event consists of human interaction where the customer could terminate the task. So, in the analysis, we decided to utilise the killed tasks to sense any brute forcing behaviour. Furthermore, in the investigation, we accepted the fact that cyber-attacks may be generated by humans or software and brute forcing is not an exception.

Crucially, the data set consists of 25,000,000 tasks, and 6,608,917 of the total number of tasks were tagged as killed tasks which represent 26.4% of the total number of tasks. The total number of customers in the dataset was 925 active clients. However, only 735 of them committed a kill task event.

Figure 4.1 illustrates the number of killed VMs per user for the number of users. Whereas, Figure 4.2 displays the top thirty users with the highest number of killed VMs. We observed that most of the customers did not pass the threshold of killing tasks which were 200,000, but three customers passed that threshold. The customers' IDs were 390, 772 and 225 where 390 notably killed over 450,000 tasks and 773 killed around 350,000 tasks. Also, we observed that the highest eight customers after the top three could be grouped where they fall in the range between 150,000 and 200,000 killed jobs. The rest of customers committed a kill task event in a frequency below 100,000 times.

Consequently, we can highlight customers 390, 772, and 225 as their behaviour can give a strong indication of brute forcing technique. Such

behaviour can be linked to the proposed attack model where customers' confidentiality can be violated.

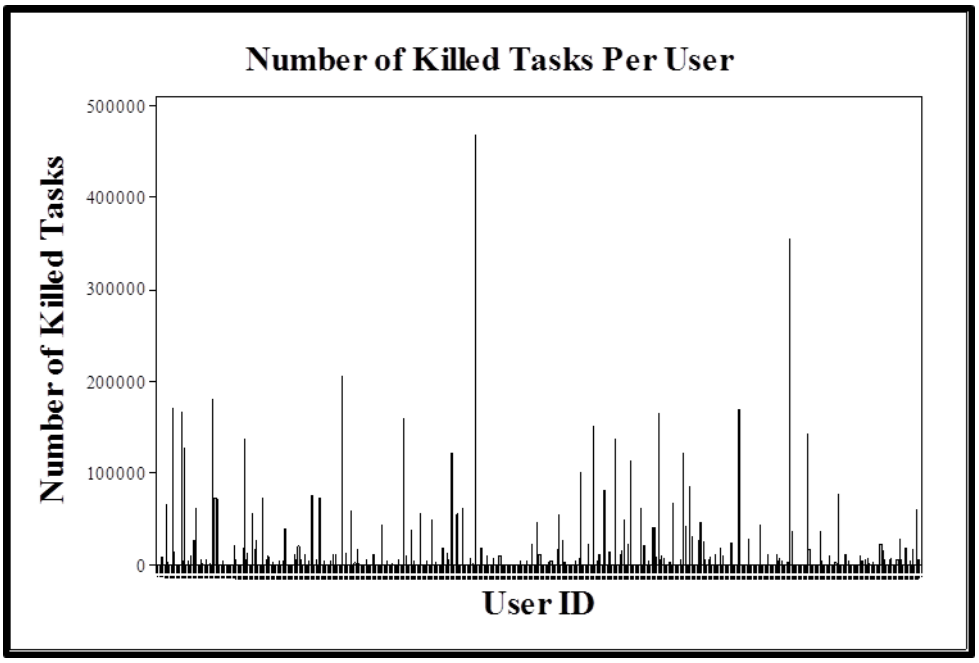

Figure 4.1: Number of Killed VMs per User.

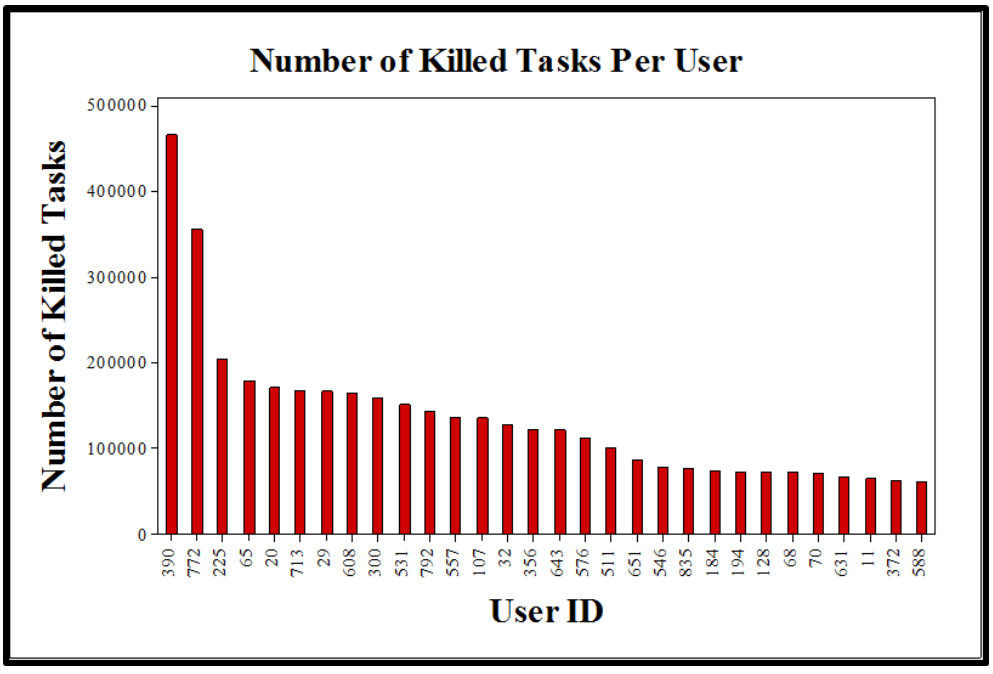

Figure 4.2: Number of Killed VMs per User (Top 30 Users).

4.2.1 Markov Chain

Markov chain is a model to show and highlight stage dependency and to calculate the probability of a successful attack based on the attack model described in section 3.5. Particularly, Figure 4.3 shows the Markov chain of the model and the likelihood of each transaction from one stage to another. To clarify the graph, MT stands for Multi-Tenancy, BF stands for brute force, and SCA stands for side channel attacks. Phase one in the attack model is represented by stage 1 in Markov chain, and phase two is represented by steps 2, 3, 4, and 5. Moreover, step 6 and 7 represent phase three of the attack model while staging 8 and 9 describe the failed and successful attacks respectively.

4.2.2 The Advantage of using the Markov Chain Model

Indeed, the model is useful in the predictions of the attacks in the Cloud computing because of the following reasons:

- The model is flexible: since most of the attacks in the Cloud can be simulated, then it is easy to find the probability of each attack happening from the past events. Indeed, the method is probabilistic, and it provides estimates of the probability distribution associated with the situation [12].
- The order of the Markov Chain: the methodology is vital since it gives the number of the past attacks and techniques that can influence the probability of the present condition. Apparently, in the first-order Markov chain, the present state only depends on the previous state [12].

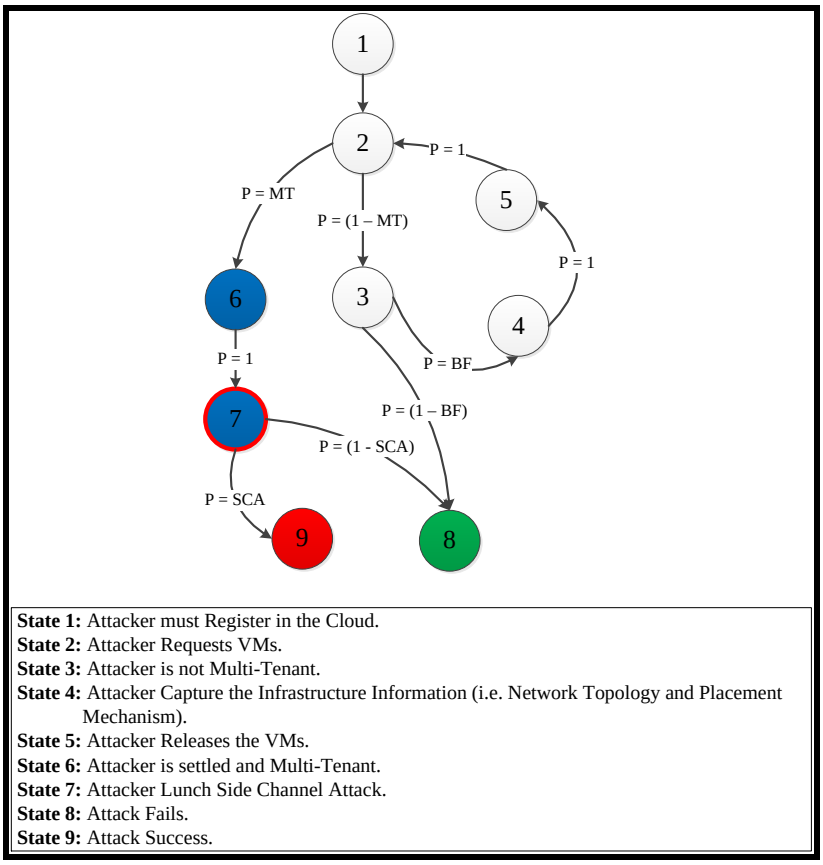

Figure 4.3: Markov Chain of the Attack Model.

Table 4-1 shows the probability from one stage to another. Therefore, to find out the probability from one stage to another stage while passing through several stages, the likelihood of each link must be multiplied altogether. For example, if we need to know the probability of a successful attack from the attack chain, the simplest and direct route is the following:

Thus, to calculate the probability of a successful attack, the likelihood of all links must be multiplied. From Table 4-1 we know the following:

$$theprobability(P)stage12 = P_{1-2} = 1$$

$$theprobabilitystage26 = P_{2-6} = MT$$

$$theprobabilitystage67 = P_{6-7} = 1$$

$$theprobabilitystage79 = P_{7-9} = SCA$$

Therefore, the probability of a successful attack would be as follows:

$$P_{1-9} = P_{1-2} \times P_{2-6} \times P_{6-7} \times P_{7-9}$$

$$P_{1-9} = 1 \times MT \times 1 \times SCA$$

$$P_{1-9} = MT \times SCA$$

Based on the calculations above, there is a need to know the probability of being Multi-Tenant and the possibility of side channel attacks to calculate the likelihood of a successful attack. The chances of being Multi-Tenant are unknown in the literature to the extent of our knowledge. Therefore, there is a need to quantify Multi-Tenancy to have – at least – an indication of its size. In the case of side channel attack, the extreme case would be that any attacker will launch a side channel attack once he becomes Multi-Tenant. Then, the maximum probability could be assumed for SCA. As a result, the above calculation would be:

$$assumeSCA = P_{MAX}$$

$$thenSCA = 1$$

$$asaresultP_{1-9} = MT \times 1$$

$$P_{1-9} = MT$$

In light of the above, the probability of a successful attack is equal to the likelihood of being Multi-Tenant. In other words, there is a need to know the probability of Multi-Tenancy to calculate the likelihood of successful attack.

Table 4-1: Probability from stage to another.

From	To	Probability (0 – 1)
1	2	1
2	3	1 – MT
3	4	BF
4	5	1
5	2	1
2	6	MT
6	7	1
7	8	1 – SCA
7	9	SCA
3	8	1 – BF

Another interesting calculation would be to calculate the effect of phase two of the attack model on the probability of successful attack. Phase two is represented in the following rout:

Then, the probability of successful attack with brute forcing would be as follows:

$$P_{successfulattack} = P_{1-9} + P_{2-2}$$

$$where P_{1-9} is the direct path for successful attack$$

$$previous we know that P_{1-9} = MT$$

$$P_{2-2} is the probability of brute forcing$$

$$P_{2-2} = P_{2-3} \times P_{3-4} \times P_{4-5} \times P_{5-2}$$

$$P_{2-2} = (1 - MT) \times BF \times 1 \times 1$$

$$P_{2-2} = (1 - MT) \times BF$$

From section 2.6, 23% of the attacks utilised brute force. Therefore, this percentage could be used as the probability of utilising brute forcing. Then, the P2-2 would be updated as follows:

$$P_{2-2} = (1 - MT) \times 0.23$$

$$P_{2-2} = 0.23 - 0.23MT$$

$$as a result$$

$$P_{successfulattack} = P_{1-9} + P_{2-2}$$

$$P_{successfulattack} = MT + 0.23 - 0.23MT$$

$$P_{successfulattack} = 0.23 + 0.77MT$$

Table 4-2 shows the probability of a successful attack using different probabilities for Multi-Tenancy. The Multi-Tenancy probabilities are assumed for the sake of capturing the effect of phase two on the likelihood of successful

attack. Figure 4.4 shows the effect of phase two on the probability of successful attack. It is clear that the relation between Multi-Tenancy probability and the effect of phase two (i.e. brute forcing) is an inverse relationship. Whenever the higher the Multi-Tenancy probability, the lower the force of brute forcing on the successful attack.

Table 4-2: Variation on the Probability of Successful Attack according to different Multi-Tenancy Probabilities.

P_{MT}	$P_{successful\ attack}$
0.1	0.307
0.2	0.384
0.3	0.461
0.4	0.538
0.5	0.615
0.6	0.692
0.7	0.769
0.8	0.846
0.9	0.923
1	1

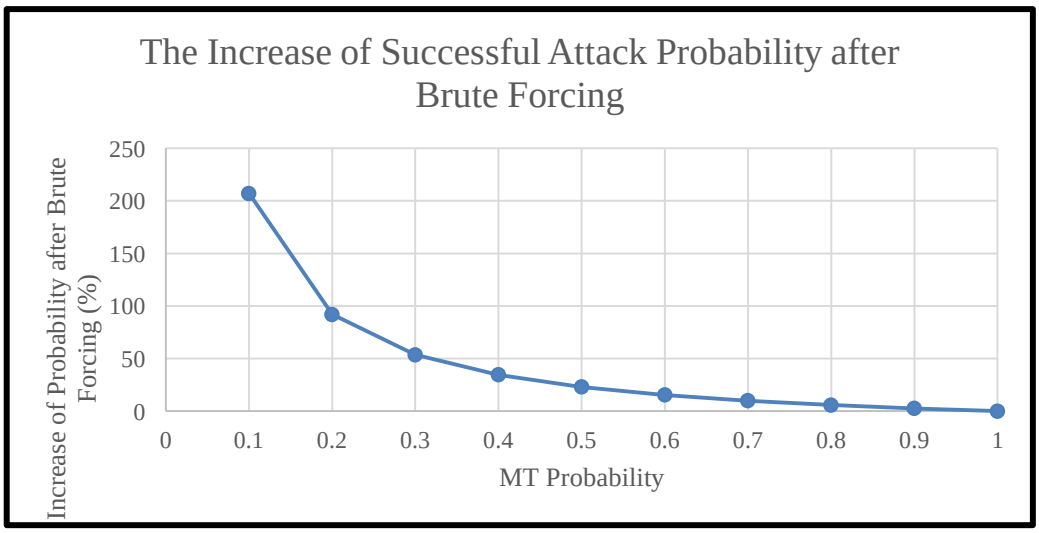

Figure 4.4: The Effect of Brute Forcing on the Successful Attack Probability.

To sum up, the probability of a direct successful attack is equal to the probability of Multi-Tenancy. Also, the brute forcing will increase the probability of a successful attack in general. Such increase is inverse with the probability of Multi-Tenancy. Since there is no known probability of Multi-Tenancy or any sort of quantification to measure its impact, there is a need to quantify Multi-Tenancy in order to know how big is the impact.

4.3 Wide-Band Delphi Method

As an alternative to the Markov Chain, the Delphi technique can be used to give an expert opinion on the security of the Cloud system. Indeed, the Delphi method is a forecasting technique used to collect expert opinion objectively. In the Delphi method, the moderator is used to control and facilitate the answers to provide solutions to a problem. Ideally, the method has the following advantages [8]:

- The subject matter expert (SME) interacts throughout an interactive synthesis which captures all the inputs while allowing the participants to iteratively revise their facts and opinions in light with the others. Therefore, the solution can be verified.
- Moreover, the anonymity of the participants will encourage good results since the members will be speaking their minds.

Indeed, the methodology will help to refine the recommendations about the security in the Cloud with multi tenancy.

4.4 Summary

The chapter discussed the Multi-Tenancy in the Clouds through the threat and attacks models. Besides, the Markov Chain was discussed since it is vital in predicting the dependency among the Multi-Tenant system. Moreover, the chapter illustrated the advantages of the Markov chain and how it helped in analysing the situation.

Equally, Chapter 5 will be helpful in dissecting the system models and a technique such as the Chinese Wall Security Policy. Moreover, the Multi-Tenancy harmful calculator will be covered.

Chapter 5 An Approach to Enhancement of Multi-Tenancy Security

5.1 How to Approach Multi-Tenancy

Once agreed that Multi-Tenancy is a vulnerability, then we need to choose one of the well-known risk strategies in order to control and minimize its impact [54]. There are four strategies to deal with risks and Figure 5.1 shows the action points in order to mitigate a risk; these four strategies are as follows:

- Eliminating the risk.
- Mitigating the risk.
- Transferring the risk.
- Accepting the risk.

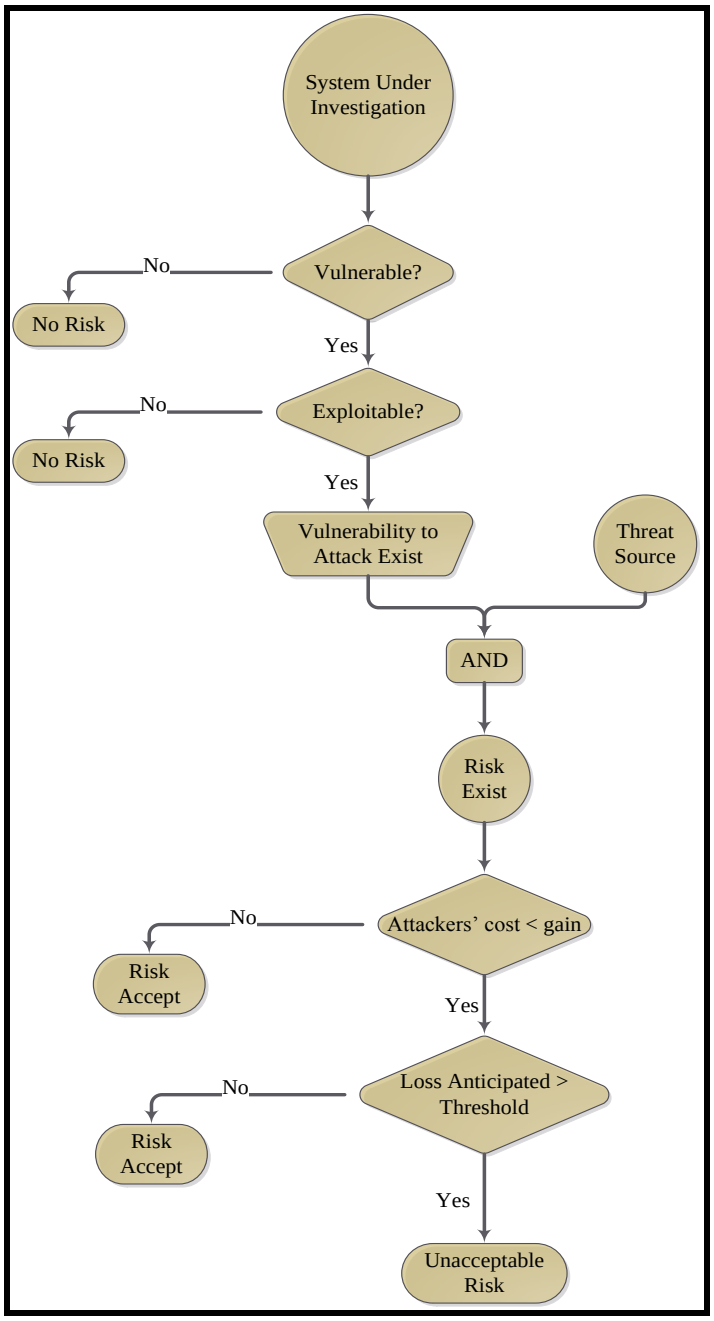

Figure 5.1: Risk Mitigation Action Points recommended by NIST [54].

Although eliminating the risk is considered the most powerful strategy, it is not possible to apply on Multi-Tenancy as most of the Cloud Computing benefits are linked to it, as illustrated in section 2.4.2.

Therefore, in order to eliminate the risk of Multi-Tenancy, we have to eliminate Multi-Tenancy and that can be achieved by either eliminating what makes Multi-Tenancy vulnerable or eliminating what forms Multi-Tenancy in the first place.

As mentioned previously in equation (1) in section 2.4.1 that Multi-Tenancy is a natural result of allowing resource sharing over virtualization. Therefore, in order to eliminate Multi-Tenancy, we need to either eliminate the use of virtualization or disable resource sharing.

In both cases that is not acceptable as that will eliminate most of the Cloud Computing main drivers and marketing strength features as shown in Figure 2.7.

The other direction is to try to eliminate what makes Multi-Tenancy vulnerable which is the possibility of taking advantage of the shared environment. Such possibility is valid because of the side channel attacks.

Side channel attacks by their nature are unlimited (i.e. in the sense it is hard to count them) in their number and they evolve with time which makes it hard to eliminate all the possible side channel attacks [40]. In addition, dealing with the known side channel attacks is not an easy task where the forms of such attacks are multiple.

For example, there are researchers dealing with the memory as the attack vector such as [55] where they proposed to eliminate the shared memory attack. Although, they achieved the goal but their solution was a hypervisor dependent where it worked only over Xen.

Another form of attack is the timing side channel attacks and [52] in his proposal was able to eliminate three forms of timing side channel attacks. However, in order to eliminate these three forms of timing side channel attacks the CSP must sacrifice 2/3 of his infrastructure. In other words 2/3 of the infrastructure is overhead and the solution will not eliminate all the timing side channel attacks without even mentioning the other forms of side channel attacks.

From the previous two examples, it is quite obvious that trying to eliminate all the side channel attacks will be at a very high cost if it is possible in the first place.

Furthermore, some of the known side channel attacks have a tight relationship where two forms of side channel attacks cannot be eliminated in the same time [40]. In other words, if the vulnerability of side channel attack A

is eliminated, then the vulnerability of side channel attack B cannot be eliminated. As a result, if attack A is blocked that means attack B will be successful.

Since we cannot eliminate the side channel attacks and cannot tolerate the loss of major features due to the elimination of either virtualization or resource sharing then the eliminating risk strategy is not acceptable and is not applicable when it comes to dealing with Multi-Tenancy risks.

The second-best strategy is to mitigate the risk and this is what we are after; where we try to balance between the benefits brought by Multi-Tenancy and the security putting in minds other factors such as performance and cost. Figure 5.2 illustrates the approach to secure Multi-Tenancy where we highlighted Multi-Tenancy as vulnerability and a security concern in Cloud Computing and we illustrated that layer 1 cannot be eliminated either partially or completely. Also, layer 3 cannot be eliminated totally as described earlier.

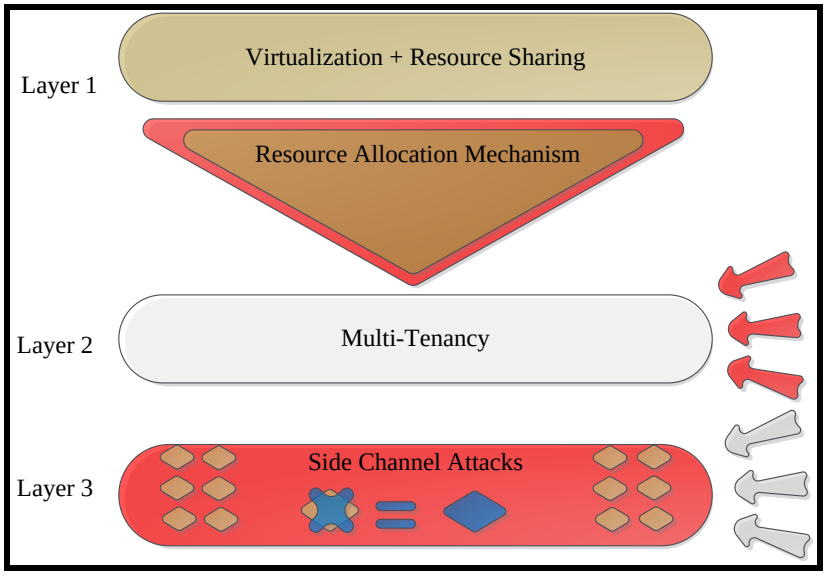

Figure 5.2: The Approach to Secure Multi-Tenancy.

Therefore, the only angle left is how to go from layer 1 into layer 2. In other words how does Multi-Tenancy happen in the IaaS Clouds? The answer is the resource allocation mechanism. Also, in order to mitigate the Multi-Tenancy risks, the resource allocation mechanism must be controlled. Making the resource allocation mechanism a security aware mechanism will enhance the total security for the CSP and will minimize the surface attack. In addition, minimizing the probability of being a Multi-Tenant by controlling the resource

allocation mechanism will have an impact on the underlying layer (i.e. layer 3) as layer 3 is dependent on layer 2.

5.2 System Model

Based on the approach from section 5.1, allocation mechanism must be controlled in order to secure Multi-Tenancy and minimise its impact. In this section, the proposed system model to secure Multi-Tenancy.

5.2.1 System Model Visualisation and Description

The proposed system model is shown in Figure 5.3, where all the solid gold links (1c, 2c, 3c, 4c, 5c, 6c and 7c) represent control channels which are under Cloud provider control and responsibility. Moreover, all the dashed gold links (1o, 2o and 3o) are operation channels where the letter (c) donates monitoring and letter (o) gives operation. The separation of these channels is vital to enhance system security and implement security in depth.

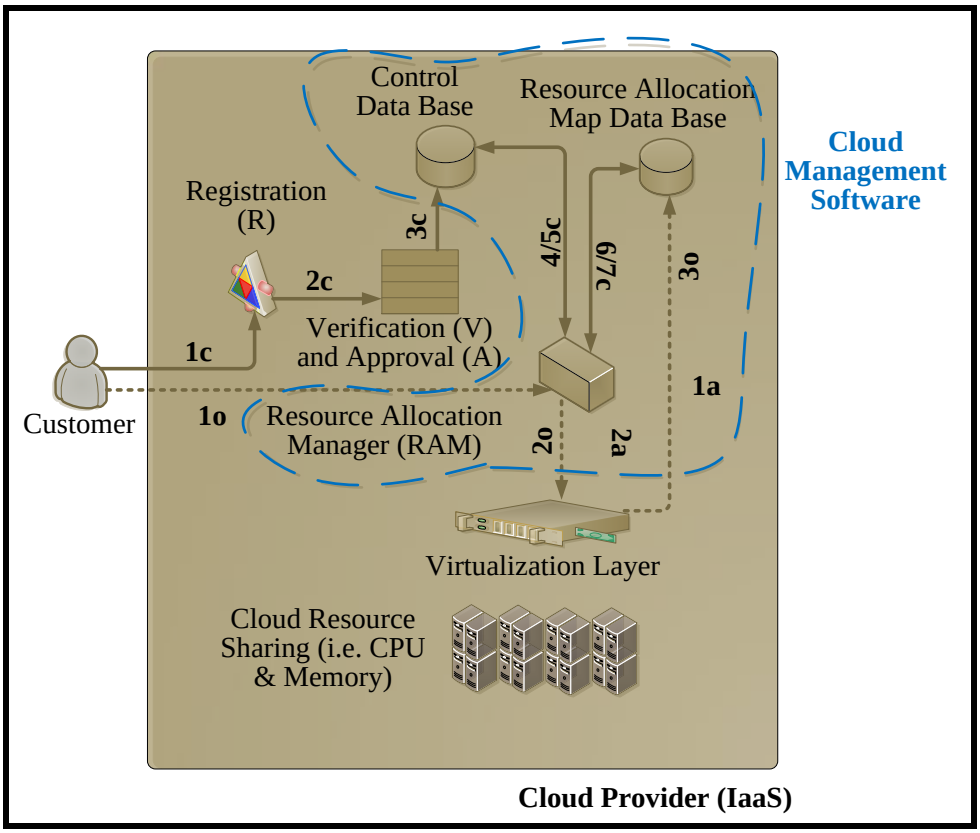

Figure 5.3: System Model Visualisation.

All the system components are defined below; following this process, a description of the system flow is given.

- ***Registration Unit:*** the initial contact between the customer and Cloud provider. Registration can be an online form or a contract signed by both

parties. In this phase, all relevant information that will define the allocation mechanism restrictions should be gathered.

- ***Verification and approval Unit:*** this is an important phase where the provider should verify and approve the information given by the customer. The importance of such process is to protect the vendor's image by avoiding any fraud possibility.
- ***Control Database (CDB):*** is the location where the parameters and restrictions of the resource allocation technique are stored. There are two contacts to this Database; the first is made by the Cloud provider in order to store the customer information to be utilized by the allocation technique; the second is made by the resource allocation manager in order to extract the customers' resource allocation requirements.
- ***Resource Allocation Manager Unit (RAMU):*** is responsible for allocating resources following a customer request. The RAMU is the only system component to access the control database.
- ***Resource Allocation Map Database (RAM-DB):*** this Database is responsible of keeping updated records of resource allocation.

5.2.2 Description of the Proposed System

To understand this system better, we describe the scenario of a customer joining a Cloud provider. The process starts when a customer needs to utilize a public IaaS Cloud as its infrastructure. First, the customer will register either online or by visiting the Cloud provider – this is represented by link 1c. Then, the provider will verify the information provided by the customer; official documents could be used for this purpose – this is represented by link 2c. If the customer passes the verification process, the provider will approve the process to the next stage. After the verification and approval processes, the customer's data will be stored in the control Database where all resource allocation restrictions will be specified by the CSP, this is reflected by link 3c.

Specifically, when the customer has been successfully registered and security restrictions specified, the system is ready to be utilised by the client. Besides, the request of the client to allocate resources shall be sent to the RAMU via link 1o. Then, the RAMU will require the security restrictions from the control database and the current resource allocation map from the resource

allocation map Database to allocate the customer's resources in the proper location – this process includes four transactions 4c, 5c, 6c, 7c, and 2o respectively. And whenever a client releases a resource, the resource allocation map Database is updated immediately via 3o. Notably, Figure 5.4 shows the algorithm and the flow of the system model. In such a setup, an attacker will not have the chance to take advantage of the resource allocation mechanism, and the benefits of Multi-Tenancy are preserved. The yellow coloured process in Figure 5.4 which is the actual distribution mechanism will be detailed in the next section.

In a real production Cloud environment, the adoption of this system model is achievable as long as the necessary requirements are fulfilled. First, Cloud environment should utilise virtualization technology or Container technology. In relation to the definition of Multi-Tenancy in this investigation, virtualization is an important component because without this requirement, the definition of Multi-Tenancy is not applicable. Second, the scheduler should be modified to accommodate the new set of policies in order to schedule any new request. Third, a new database capturing the current allocation of resources should be introduced to the system to close the loop for the scheduler. The availability of these requirements in a real production Cloud environment is essential to implement the proposed system model.

The overhead of adopting these requirements is considered minimal because the changes are made to the software (i.e. the scheduler's policies). No change to the hardware is required to adopt the system model.

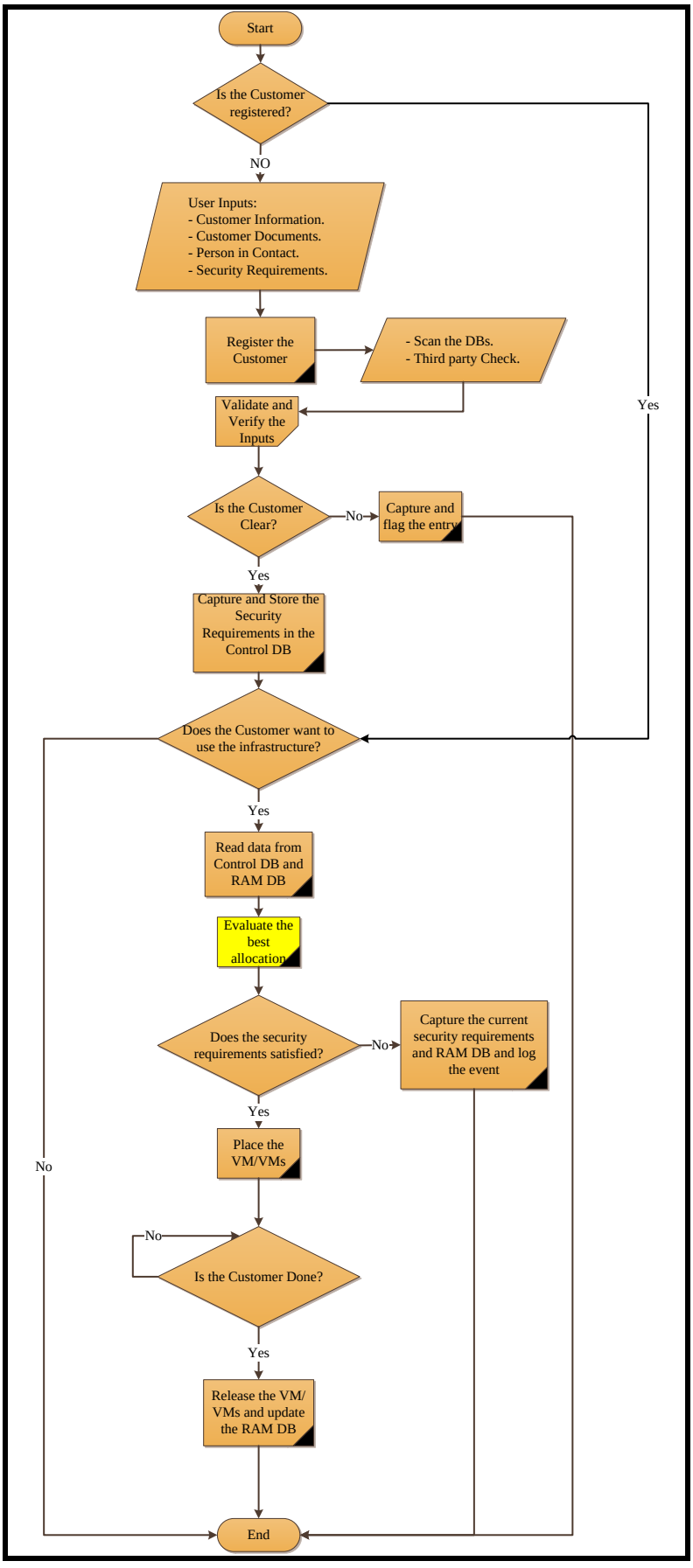

Figure 5.4: System Model Algorithm.

An extra benefit of this system is that the control Database can have a different resource allocation method, whereby the system model will not be changed. This advantage will give the Cloud provider the opportunity to define security restrictions based on their business strategy. Also, it gives the provider the chance to implement their resource allocation methods if needed. Moreover, it could be a security best practice to change the resource allocation process periodically to raise the system difficulty and make it hard to be predicted. Figure 5.5 summarises the system model advantages, challenges and disadvantages.

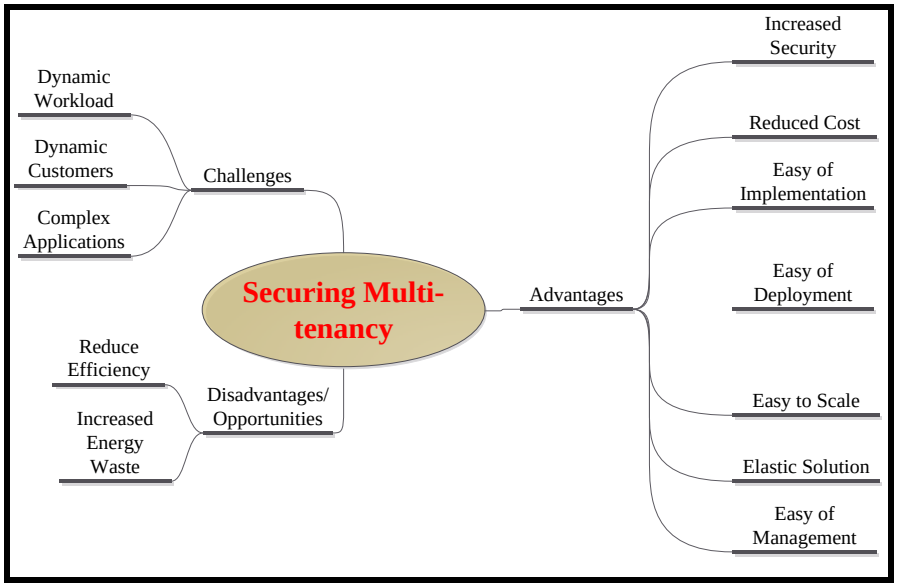

Figure 5.5: System Model Advantages, Challenges and Disadvantages.

5.2.3 Chinese Wall Security Policy

The allocation mechanism is based on a model called the Chinese Wall Security Policy (CWSP). The original model was for military uses in communications systems [56]. This model then modified and proposed to solve the conflict of interest for enterprises by (ibid). In their model, they had three entities: subject, object, and label. The subject is the organisation, the object is the file, and the label is file tag. Particularly, the model is building a tree for subjects who may have a conflict of interest; thus, many subjects coexist in the same tree are having conflict of interest. Then, each object is unlabelled unless accessed by a subject. Once a subject accessed the object, it becomes labelled and stored in the tree under the acted subject. Indeed, the label will prohibit the object for any subject that is sibling to the acted subject.

The concept proposed to control access in Clouds for VMs by [13] where they had three entities in their model subjects, objects, and access operation. Subjects were the organisations, objects were the VMs, and access operation was the read and write operation. The model will start by building a tree for subjects who have a conflict of interest and store their objects under them. Any VM can be accessed by its creator or any subject does not coexist in the tree holds the creator subject.

The model will be used to secure allocation for VMs in IaaS public Clouds. For this purpose, there will be one tree to store all subjects and their objects – called security tree. Moreover, the model will consist of three entities: subjects, objects, and transactions. Specifically, the subjects are the Cloud customers and objects are the VMs and transactions are the request and release of VMs. Therefore, the security tree will consist of two tiers which will lead to the separations during the allocation of VMs.

Notably, the Figure 5.6 shows the security tree where subjects are grouped by the business domain. Therefore, the first-tier separation is done by separating individuals from enterprises. Moreover, the second-tier separation happens between competitors. Accordingly, the first-tier separation will reduce the poll of attackers by separating individuals from enterprises; hence, it will have a positive impact on the surface of attack. Furthermore, the poll of attackers will be reduced by the second-tier separation as competitors will be separated from each other. Indeed, most of the attacks are generated by individuals either working alone or part of an organised crime, or others are generated by competitors to gain business advancement [16].

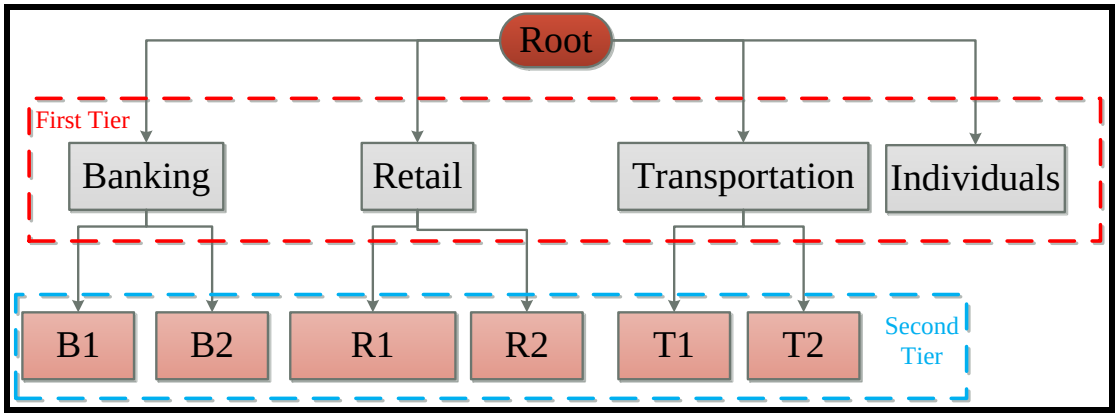

Figure 5.6: Security Tree.

To illustrate, Figure 5.7 shows the layout of VMs before applying CWSP where both the companies of T1 and T2 (transport companies) are sharing physical machine 3. However, the situation is avoided in Figure 5.8, where the CWSP is applied. Also, the possibility of having all VMs belonging to the same customer in the same physical machine is valid. Notably, Figure 5.8 shows the scenario with a physical machine 1 (hosted by B1).

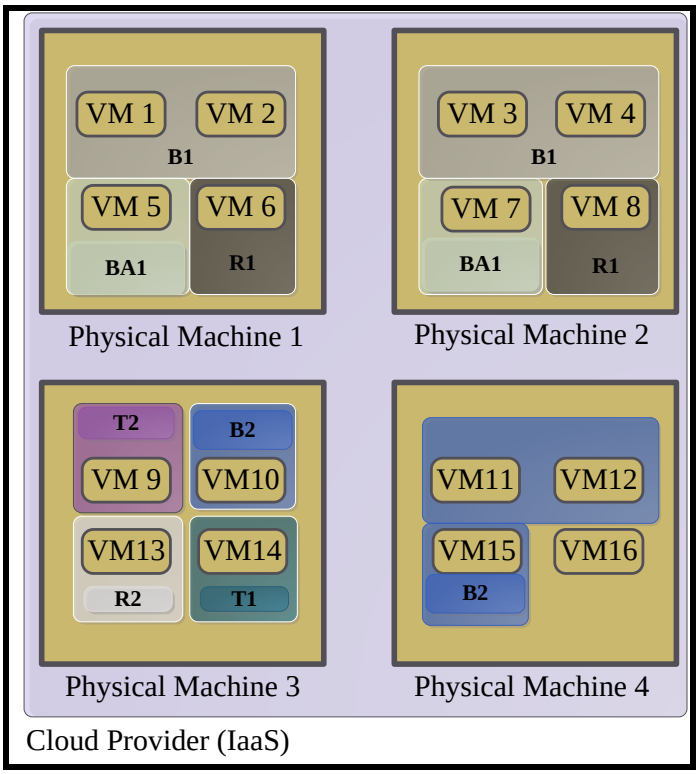

Figure 5.7: Resource Allocation without CWSP.

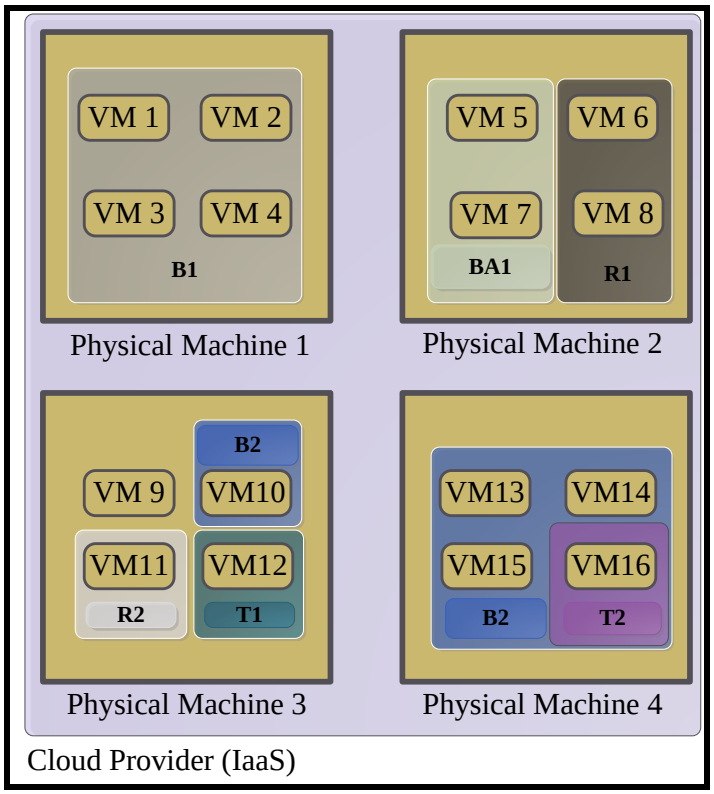

Figure 5.8: Resource Allocation with CWSP.

In the following steps, a formalisation of the model is shown and described through the formation and proofing of theories:

$$Let X the set of servers, X = \{x_1, \dots\dots, x_n\}$$

$$Let O the set of objects, O = \{v_1, \dots\dots, v_n\}$$

$$Let S the set of subjects, S = \{s_1, \dots\dots, s_n\}$$

$$Let B the set of business domains, B = \{b_1, \dots\dots, b_n\}$$

$$Let T the set of transactions, T = \{R_1, R_2\}$$

$$where R_1 = VMrequest \wedge R_2 = VMrelease$$

Policy One:

If two VMs are located in the same server, then it is one of the following cases:

- Both VMs belong to the same subject;
- VMs belong to the individual group;
- VMs belong to a subject from different business domain.

$$if\, o_x \in x_m \vee o_y \in x_m$$

$$then\, o_x \vee o_y \in s_x$$

$$o_x \vee o_y \in b_i where b_i is the individual group$$

$$o_x \in b_x \vee o_y \notin b_x$$

Allocation Matrix:

The allocation matrix consists of two Boolean matrixes. If there is an object allocated in a server, then the corresponding cell will return *true*. Furthermore, we use $A_1(s,x)$ to investigate whether an object already exists in the server for a given subject. Hence, it will reduce the time of allocating the object since there is already an object which has passed the investigation for the same subject. Besides, $A_2(b,x)$ investigates the allocation possibility when no object belongs to the same subject in the server. Specifically, the following equations illustrate the scenarios:

$$A_1(s_x, x) = boolean$$

$$true = if\, o_x \in s_x is already allocated$$

$$false = if\, there is no object already allocated \in s_x$$

$$A_2(b_x, x) = boolean$$

$$true = if\, o_x \in b_x is already allocated$$

$$false = if\, there is no object already allocated \in b_x \vee b_i$$

$$where\, b_i is the individual group$$

Policy Two:

The allocation of an object is granted only and only if:

$$A_1(s, x) = true$$

$$A_1(s, x) = false \vee A_2(b, x) = false \vee s \notin b_i$$

$$A_1(s, x) = false \vee A_2(b, x) = true \vee s \in b_i$$

Policy Three:

$$if\, A_1(s, x) = false for all (s, x)$$

$$A_2(b, x) = false for all (b, x)$$

Hence, it represents an initial secure state.

Policy Four:

The request of an object is granted only and only if:

$$T(s, R_1) = true \Rightarrow one statement is valid below$$

$$A_1(s,x) = true$$

$$A_1(s,x) = false \vee A_2(b,x) = false \vee s \notin b_i$$

$$A_1(s,x) = false \vee A_2(b,x) = true \vee s \in b_i$$

Theory One:

Once an object is allocated in a server, then the only object that can coexist with it either belongs to the same subject or to a subject from another business domain, but not an individual group.

Proof of theory one:

Let us assume an object *A* located in a server *X* belongs to a subject *M* who belongs to business domain *Y*. Thus, If theory one is false, then it is possible for an object *B* belonging to a subject *F* of an individual group or to a subject *N* of the same business domain *Y* to coexist with an object *A* in server. Therefore, if object *A* is already hosted in server *X,* then for object *B* to coexists, policy two must be satisfied. Thus, the following holds:

$$for A_1(s_F, x_X) = false \vee A_2(b_i, x_X) = false \vee s \in b_i$$

$$the allocation will be rejected due policy two violation$$

$$for A_1(s_N, x_X) = false \vee A_2(b_Y, x_X) = true \vee s \notin b_i$$

$$the allocation will be rejected due policy two violation$$

Theory Two:

A server can host one subject of each business domain except an individual group or more than one subject from the individual group.

Proof of theory two:

If server *M* is occupied by a subject *X* belonging to a business domain *F*, then by policy two, the only subjects can be granted allocation in server *M* are either belonging to business domain other than *F* and don't belong to the individual group. Equally, if server *N* is occupied by a subject *Y* belonging to an individual group, then by policy two, the only subjects that can be granted allocation in server *N* are belonging to the individual group.

Theory Three:

For a business domain consisting of N subjects, then the minimum number of servers to accommodate at least one object for each subject is N.

Proof of theory three:

Let us assume there are N subjects belonging to business domain *B*, and each subject requesting one object. Therefore, by theory two, the Cloud provider needs N servers to accommodate the request.

Back to Figure 5.7 which depicted the VMs allocations without CWSP. Since the CWSP is not enabled, no allocation restrictions are assumed. As a result, any VM can be in the infrastructure. Thus, the configurations in Figure 5.7 are possible and valid. On the other hand, theory 1 will secure and restrict the allocation mechanism which leads to the configuration in Figure 5.8.

5.2.4 System Model Simulation

In order to reflect the impact of the system on a Cloud, a simulation was conducted using Opnet (https://www.riverbed.com/gb/) as a simulation tool. Opnet mainly simulates data and telecom networks, yet the introduction of systems was made to simulate applications and capture user and traffic behaviours. The tool gives you the power to add traffic policies, system attributes and limitations. Since the goal of this investigation is to capture the VMs allocation based on the policy applied and its impact on the system; Opnet is a very good choice to use. The first objective of this investigation is to demonstrate the impact of CWSP allocation method compared with a well-known method (i.e. round robin). The second objective is to identify the impacted attributes of the system. The final objective is to measure the impact and the trade-off capacity. For the sake of this investigation, the environment consists of four racks where each rack consists of 25 PMs. Each PM has eight cores where each VM is designed to use two cores. The groups in Figure 5.6 were used to generate the traffic and apply the CWSPs. The maximum number of PMs is 100, where the maximum number of VMs that could be accommodated is 400. The common practice in Clouds with regard to scheduling – allocating resources – is the round robin technique. Therefore, the system is configured to use the round robin as a scheduling method when it is not using CWSP. The configuration of the allocation method is done through the traffic profile which is a component designed in the Opnet to characterise the traffic form end to end. Round robin is well-known algorithm and already implemented, whereas the CWSP algorithm was implemented using the rules

in section 5.2.3 and Figure 5.4. the user profile which is a component designed in the Opnet to characterise the user requirements, priority and much VMs to request was configured utilising Figure 5.6.

Figure 5.9 shows the connection between clients and CSP, where (I) refers to individuals requesting Cloud resources while Figure 5.10 shows the systems within the CSP. It is clear that the CSP accepted requests from all the users as shown in Figure 5.11. On the other hand, when CWSP is implemented the CSP does not accept any requests from individuals – Group I – and inforce the system policies as shown in Figure 5.12. As a result, the utilisation is degraded in exchange of the security enhancement as shown in Figure 5.13. The results show the trade of between the number of VMs accommodated (i.e. profit and utilisation) and security level. With CWSP the utilisation is 89% compared to 100% utilisation without the CWSP. In terms of the number of VMs accommodated, there is a drop of 10% when CWSP is activated. In other words, this could mean a drop of 10% out of the profit in exchange with security level.

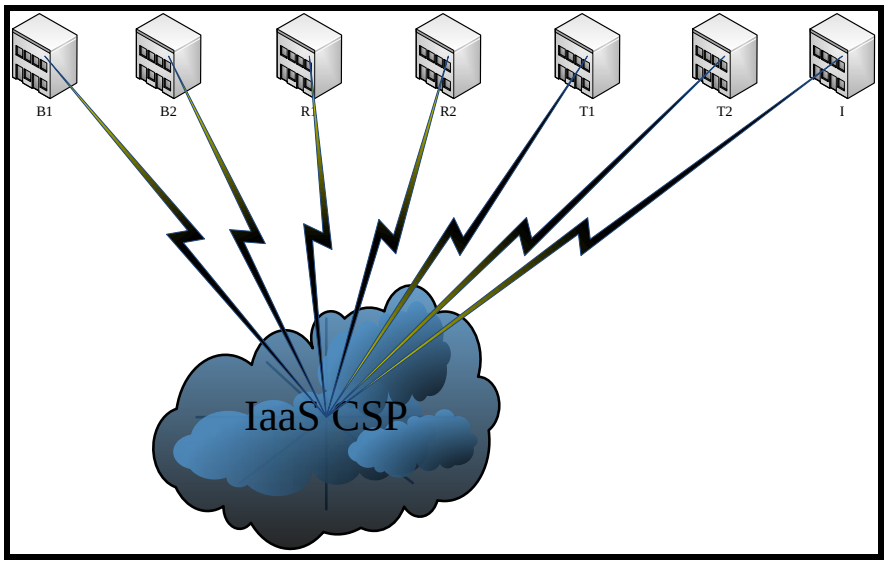

Figure 5.9: Clients to CSP connection.

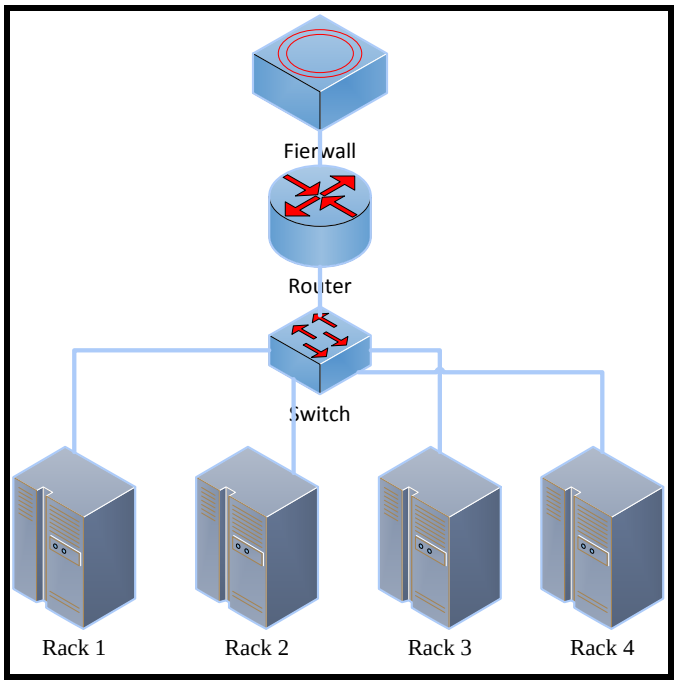

Figure 5.10: CSP's infrastructure.

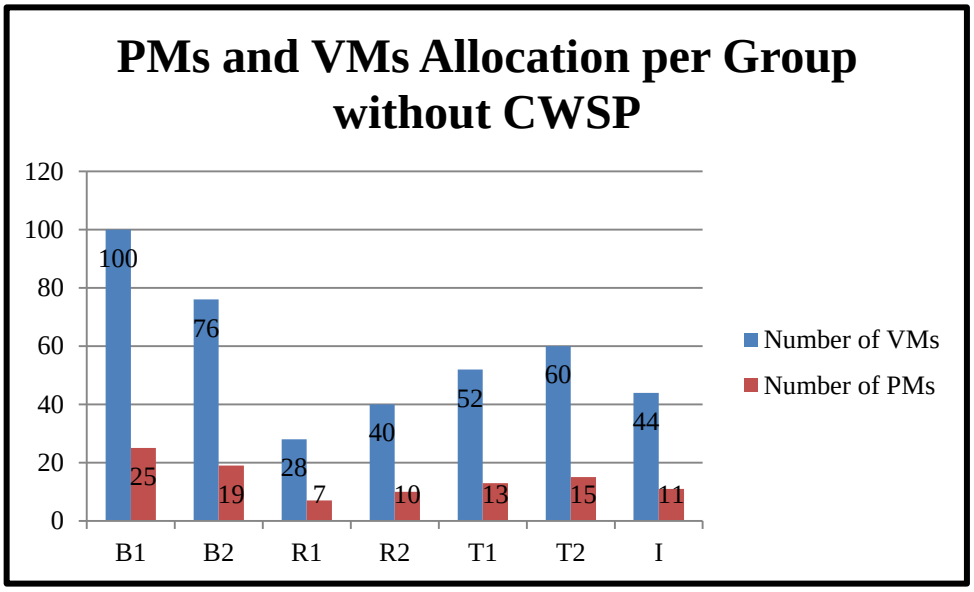

Figure 5.11: Group Allocation without CWSP.

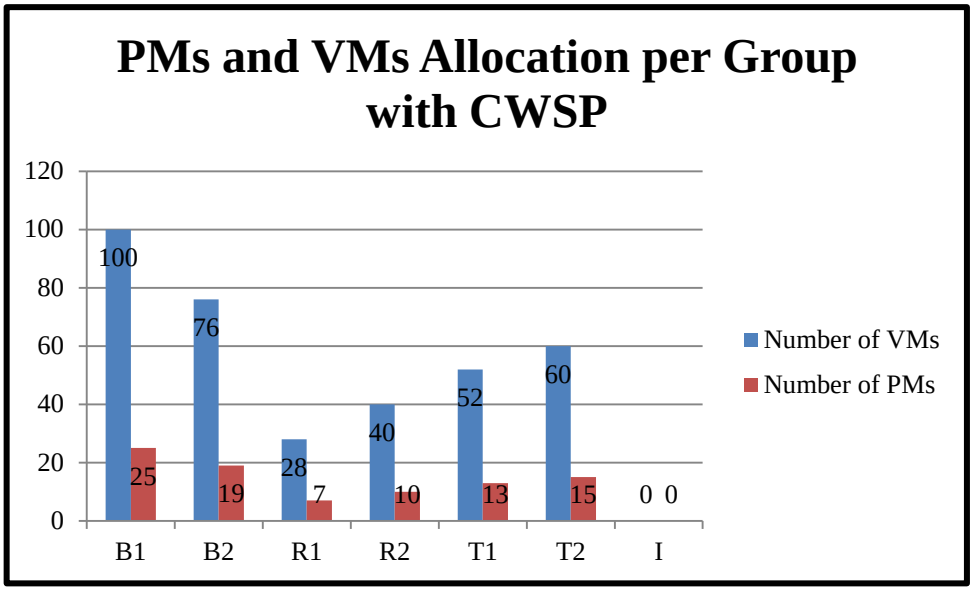

Figure 5.12: Group Allocation with CWSP.

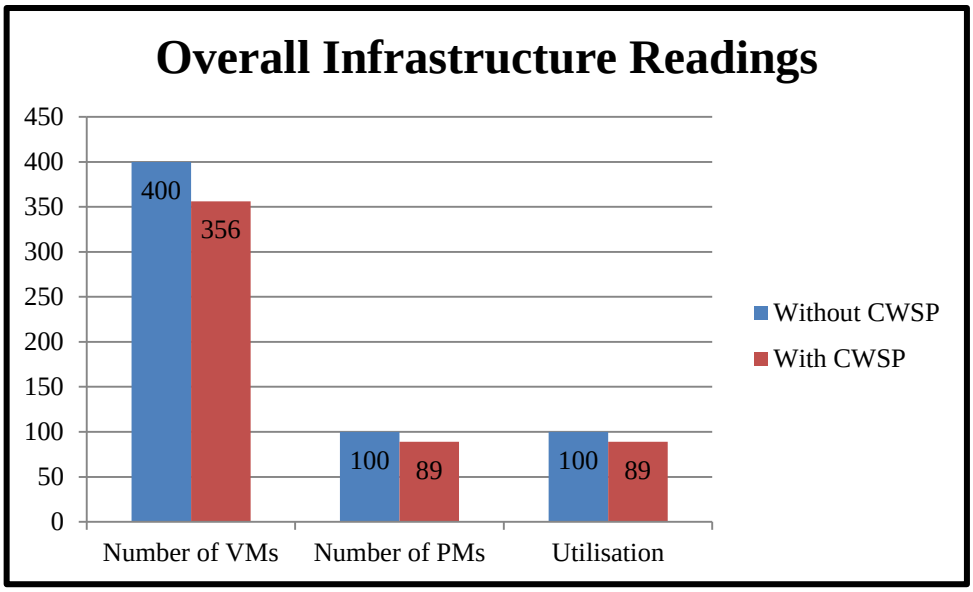

Figure 5.13: Overall Utilisation.

5.3 Why Use the CWSP Model?

As noted from the previous chapters, the major issues in the Cloud are confidentiality and integrity of data. Therefore, the Chinese Wall Security Policy is vital for such an infrastructure. Ideally, the policy allows the users to pass through a secure Discretionary Access Control (DAC) [9]. The DAC is the basis of the CWSP where it allows for the equivalence relation: transitivity, reflexivity, and symmetricity [10]. Moreover, it permits a flow of information and data if the

system is secure after an addition of a new user or VM. Hence, the Cloud users are sure of their confidentiality while in the Cloud environment.

5.4 Alternative Method for Security

We propose the use of Bell LaPadula (BLP) model to provide security in the Cloud environment [10]. Ideally, the BLP does not place any constraints on the objects; in particular, it does not require them to be hierarchically arranged into an organisation's datasets and conflict of interest classes. Instead, it imposes a model upon the security parameters. On the contrary, the BLP attaches the safety policy to the subjects as well, not like the CWSP. Unfortunately, the BLP works only when the subjects are not given the freedom to choose the data sets they wish to access. Therefore, the BLP ignores the freedom subjects and objects enjoy while working with the CWSP.

Nonetheless, the freedom that is ignored in the BLP can be restored with the modification to have a "subject need-to-know" to cover all the company datasets. Crucially, the Chinese Wall Security Policy is functional and valuable in its domain.

5.5 Multi-Tenancy Harmful Calculator

In this section, we will propose a statistical model to calculate the probability of being Multi-Tenant with a malicious VM in a given Cloud. The model will estimate the likelihood that a given customer will share the server – physical machine – with a hacker by knowing the number of servers provided by a Cloud provider to serve the Cloud customers. The model is built using numerical modelling which is a technique used to describe system behaviour, in this case, the IaaS Cloud.

This model is best to serve IaaS providers because it includes the number of servers into the calculation. This model gives more accurate estimates because the scope of interaction is limited to the servers within the Cloud provider. The importance of this model is that we can measure the probability of being a Multi-Tenant with a hacker and this measurement could be aggregated with the risk assessment process for the customer to decide whether contract or not with a specific Cloud provider.

To calculate the probability of being a Multi-Tenant of a hacker, we need to know either number of customers or number of VMs in a given time. Also, we need to know the number of hackers or suspicious VMs. On top of that, we need to know the number of servers that are available to be used by customers. Equation (2) is proposed to calculate the probability of multi-tenancy.

$$let the probability of being a Multi-Tenant a malicious VM = P_{MTm}$$

$$P_{MTm} = \sum_{i=1}^{H-1} \frac{H}{X^{(i+1)}} + \frac{1}{X^{(H+1)}} \qquad (2)$$

The equation (2) *H* represents the expected number of hackers from the total number of customers, if we are interested in looking at the user level and VMs, *H* will represent the total number of suspicious VMs in a sample. Finally, *X* represents the total number of servers provided by a Cloud provider.

5.6 Summary

This chapter described and discussed the threat model of Multi-Tenancy in IaaS public Clouds. This is important to identify the attack model that takes advantage of Multi-Tenancy as vulnerability. Moreover, the approach to secure Multi-Tenancy was presented and illustrated, as it shaped the proposed system model. Besides, from this chapter, the need for quantifying Multi-Tenancy was highlighted ones more.

We, discussed the Chinese Wall Security Policy (CWSP) which is key in defining the rules and theories that govern the allocation of resources within multiple VMs. Apparently, for any company to store data in the Cloud, a secure security policy must be enacted; hence, the CWSP provides the assurance. Besides, the CWSP, we proposed the BLP model as an alternative to the CWSP. Although the model ignores the freedom to choose which is provided by the former method, it is still functional in its domain and can be used to develop secure tunnels in the Cloud since it provides security on individual subjects and objects using a secure DAC.

Chapter 6 Evaluation

6.1 Results and Analysis

We found two major issues in the research about the multitenant architectures. First, multitenant architectures must balance between sharing and security. Therefore, for such a system to deliver a cost saving and scalable solution, it must manage the dynamic resource consumption by the tenants without affecting the security. Evidently, the attack model described the situation using the killed VMs. Indeed, Brute force technique is closely related to the procedures of killing the tasks by the users. However, with the CWSP model, the threats of data integrity in a multitenant environment can be reduced and controlled.

Second, the research is successful in showing how defining the resources can reduce the vulnerabilities, where a tenant can access in as a dedicated architecture within a multi-tenant solution. However, the solution reduces the flexibility of the nodes in the network. Crucially, using the BLP model, objects in a data sharing environment, can be controlled on what they can access. Thus, the model combined with the CWSP can yield great results in managing vital resources against attacks.

6.2 Comparison with Other Solutions

Evidently, many CSPs in the market have employed the technique of reducing access. Often, the use of a service provider immediately introduces the shared responsibility and must be understood. In practice, the CSP and the Cloud subscribers should define and document in any service level agreement (SLA) their roles and responsibilities. In essence, everyone should be acquitted with the knowledge of operations to reduce the risks of data and rights exposure. In fact, the solution presented in the thesis is better since more policies have been proposed to curb such intrusion in the IaaS, SaaS, and PaaS.

Table 6-1 summarises the comparison of the proposed system in this investigation and similar systems that aim to enhance the security of Clouds. The two solutions selected are the most relevant to this work in terms of all of them take advantage of Multi-Tenancy to lunch attacks. Moreover, all of them utilize side-channels attacks in order to penetrate the vulnerability. The criteria consists of nine attributes as follows:

- Scalable: this attribute is to reflect if the proposed solution could be scalable in theory and practice.
- Secure: this attribute is about the scope of attacks which the system can countermeasure after applying the proposed solution.
- Resources Affected: this attribute shows the effect of the system on the Cloud infrastructure after implementing the solution.
- Cost of implementation: this attribute is the cost of implementing the solution that could be originated from the software modifications, hardware modifications or wasted capital investment.
- Service type: this attribute is to specify which Cloud service type is targeted.
- Deployment type: this attribute illustrates which Cloud deployment type is targeted.
- Methodology: this attribute specifies the methodology used to evaluate the solution.
- Attack vector: this attribute identifies the attack vector used to design the threat and attack models.
- Affected attribute: this attribute highlights which security attribute was contained by implementing the solution.

Table 6-1: Comparison of Similar Systems.

Criteria of Comparison	Proposed Solutions		
	This Research	**Solution of [55]**	**Solution of [52]**
Scalable	Yes	Limited to Xen	Yes
Secure	Against side channel attacks	Limited to memory attacks	Limited to time attacks
Resources Affected	Processing units might be affected	None	2/3 of the infrastructure must be sacrificed
Cost of Implementation	Low	Low	High
Service Type	IaaS	IaaS	IaaS
Deployment Type	Public	Public	Public
Methodology	Case study and simulation	Implementation	Simulation
Attack Vector	Multi-Tenancy	Memory	Time
Affected Attribute	Confidentiality and Integrity	Confidentiality and Integrity	Confidentiality and availability

6.3 Limitation of the Study

Although the investigation yielded a number of promising results, the following challenges were experienced:

- Theoretical analysis: some facts were analysed using assumption and theories to gain understanding. For instance, when studying the attack model, it was not easy to differentiate a normal user from a brute force attacker; hence, complicated mathematical models and assumption had to be used to make inferences.
- Data Set: in real environments, there are different deployments of Clouds. There is a deployment utilising the resource sharing technologies either by creating VMs or Containers. The opposite deployment does not take advantage of any virtualization technologies. Therefore, for this investigation, the data set should match and be from the first deployment. Hence, if the data set is from the second deployment, the investigation of Multi-Tenancy will be hardly achievable because the concept of Multi-Tenancy is not activated or does not exist. For other future investigations related to Multi-Tenancy, the data set should come only from the first deployment of Clouds. This is a real research challenge since there is a limitation of available Cloud data sets publicly.

Chapter 7 Conclusion and Future Work

This chapter concludes the thesis with section 7.1 summarising the thesis. Consequently, the research contributions will be revisited to link the work done. Later, the overall research evaluation will be presented, and the future work and research directions will be discussed.

7.1 Summary

The work in this thesis presents the empirical analysis and the quantification of Multi-Tenancy to improve its security in IaaS public Clouds. Indeed, the origins of Multi-Tenancy, its importance, capacity, approach, and improvement of its security is discussed and illustrated. Specifically, this work, for the first time, presents a comprehensive understanding of Multi-Tenancy and its quantification. Moreover, understanding is leveraged to enhance the Cloud environment security, improve the decision-making process for Cloud service providers (CSPs) concerning providing secure Clouds and quantifying the impact of security on Cloud environment.

Chapter 2 introduced the concept of virtualisation and described the enabling technology for Cloud Computing. Besides, the concept of Cloud Computing was discussed and explored in details where its definition, service and deployment models, and system architecture were presented.

Next, the concept of Multi-Tenancy and its importance to Cloud Computing was demonstrated. Specifically, the different arguments about Multi-Tenancy were detailed and demonstrated. Moreover, the concept of security was introduced and its challenges in Cloud environment was discussed and detailed. In essence, the different security domains were demonstrated in details and the attack theory was presented. Besides, the security attacks were illustrated and described in details. Finally, the dataset was presented and described and its related work was discussed.

We further demonstrated the threat model. Accordingly, a further elaboration details of the attack model was done, and the implementation of the attack model using Markov chain, highlighted the attack steps and probability of happening. Again, the approach to tackle Multi-Tenancy was demonstrated and

discussed in details to justify for the new argument of securing Multi-Tenancy. Furthermore, the implementation of the system model using Chinese Wall Security Policy was presented and the alternative method elaborated. Finally, a mathematical model to calculate Multi-Tenancy was explained and presented.

Besides, we presented experimental results of Multi-Tenancy. Specifically, the sampling method was discussed and illustrated in details. Again, the results of quantifying Multi-Tenancy were presented where a machine level analysis, platform level description, and different attributes captured were demonstrated. After that, the correlation analysis of platform ID, Multi-Tenancy percentage, number of VMs, duration and number of servers was presented. Equally, the behaviours from the user to platform were captured and presented. Finally, the results of the attack model were shown and the security trade-offs was presented.

7.2 Research Contributions

We positively contributed to the following:

- *The design of an attack model to exploit Multi-Tenancy*. The thesis utilises Markov chain to design the attack model and two goals are achieved. First, the highlight of the main roll of Multi-Tenancy on an attack sequence, and the stages before and after being a Multi-Tenant. Second, the possibility of measuring the likelihood of being under an attack which takes advantage of Multi-Tenancy. Indeed, the results illustrate that the higher the Multi-Tenancy probability, the higher the likelihood of the user being under attack.
- *The development of a scheme to mitigate Multi-Tenancy*. The thesis develops a scheme based on Chinese Wall Security Policy which is designed to effectively enhance the security of Multi-Tenancy on IaaS public Clouds. Besides, the system model is proven mathematically.
- *The evaluation of the quality impact of the scheme in order to enhance the security of Multi-Tenancy*. The thesis utilises the experimental evaluation to demonstrate the trade-offs of enhancing the security of Multi-Tenancy. In fact, the results show that availability, Cloud provider profit, or infrastructure cost are affected by enhancing the security of

Multi-Tenancy. Equally, the cost of up to 70 servers or the loss of 4% of the customers is the decision facing Cloud providers when deciding to enhance the security of their Clouds.

7.3 Overall Research Evaluation

The four research objectives of this thesis are discussed in Chapter 1, and the success criteria of the research in relation to achieving the proposed research objectives are listed as follows:

- To formalise an attack model takes advantage of Multi-Tenancy in Cloud Computing: The thesis developed an attack model utilising Markov chain. In fact, the work has described the threat model where the environment is defined and the vulnerability is specified. Notably, Chapter 2 discussed the attack theory and illustrated different security attacks.
- To quantify the scale of Multi-Tenancy. The thesis has quantified Multi-Tenancy since Chapter 2 established an in-depth knowledge of Multi-Tenancy in Cloud Computing through showing how to approach Multi-Tenancy and highlighting its challenges.
- To develop a scheme to mitigate Multi-Tenancy from a security perspective. This thesis has developed a scheme to enhance the security of Multi-Tenancy.
- To quantify the quality impact of Clouds after enhancing the security of Multi-Tenancy. Notably, the thesis has evaluated the impacts of enhancing the security of Multi-Tenancy.

In summary, it can be observed that all four main research objectives have been positively completed.

7.4 Future Work

There are several ways that the work presented in this thesis could be enhanced and improved. One of the areas of improvement is the dataset. Another dimension is related to the proposed scheme. Finally, the analytics of the dataset requires improvements.

Although the Google dataset is one of the biggest Cloud data made public, it would be interesting to investigate other set of data for another Cloud provider. Indeed, it is important because the reality that is known about Multi-Tenancy is limited to Google's Cloud. Indeed, another dataset will open the opportunity to generalise the observations and compare different behaviours of the interactions between users and VMs. Although one could claim that Google is one of the most efficient companies in the world of information technology, Cloud Computing is a paradigm that is implemented in different countries and with different visions.

Moreover, the scheme proposed in this thesis has been proven mathematically, but it would be interesting if it is implemented in a real Cloud. Evidently, the security problems are satisfactory problems in their nature, yet once they mature they become optimisation problems. Thus, this work comes to satisfy the security of Multi-Tenancy, and the next natural progression is to optimise the solution.

7.5 Conclusion

Multi-tenancy is a benefit to the Cloud users, but it has associated risks. Indeed, when security issues arise, it is natural to eliminate or reduce the problem. However, the cost of such changes is a big issue. Hence, the matter remains a technicality which needs to be handled by all the Cloud providers. Besides multitenancy being a problem, the opportunity must be utilized without dwelling on its setbacks. Therefore, handling the issues that come with multitenancy will enhance the usability of the Cloud.

Indeed, we succeeded in studying multi-tenancy in details including its benefits and uniqueness. Moreover, the proposed system model and allocation methodology will bring in the balance for both the security and gains from the architecture. Additionally, the proposed model of handling multi-tenancy in a more balanced manner is a great milestone in the research

References

[1] V. Gupta, “Chinese Wall Security Policy,” *Master's Project*, p. 54, 2009.

[2] S. Rajarajeswari and K. Somasundaram, “Data confidentiality and privacy in Cloud computing,” *Indian Journal of Science and Technology*, vol. 9, no. 4, 2016.

[3] S. N. Kumar and A. Vajpayee, “A survey on secure Cloud: security and privacy in Cloud computing,” *American Journal of Systems and Software*, vol. 4, no. 1, pp. 14-26, 2016.

[4] S. Aldossary and W. Allen, “Data security, privacy, availability and integrity in Cloud computing: issues and current solutions,” *(IJACSA) International Journal of Advanced Computer Science and Applications*, vol. 7, no. 4, pp. 485-498, 2016.

[5] P. K. Tiwari and B. Mishra, “Cloud computing security issues, challenges and solution,” *International Journal of Emerging Technology and Advanced Engineering*, vol. 2, no. 8, pp. 306-310, 2012.

[6] K. Jakimoski, “Security techniques for data protection in Cloud computing,” *International Journal of Grid and Distributed Computing*, vol. 9, no. 1, pp. 49-56, 2016.

[7] N. Chidambaram, P. Raj, K. Thenmozhi and R. Amirtharajan, “Enhancing the security of customer data in Cloud environments using a novel digital fingerprinting technique,” *International Journal of Digital Multimedia Broadcasting*, vol. 2016, p. 6, 2016.

[8] P. Saripalli and B. Walters, “QUIRC: A quantitative impact and risk assessment framework for Cloud security,” *2010 IEEE 3rd International Conference on Cloud Computing*, pp. 280-288, 2010.

[9] Verizon Risk Team, “2012 data breach investigations report,” Verizon, 2012.

[10] H. Saini and A. Saini, “Security mechanisms at different levels in Cloud infrastructure,” *International Journal of Computer Applications*, vol. 108, no. 2, 2014.

[11] J. Ros, “Security in the Cloud: The threat of coexist with an unknown tenant on a public environment,” *Royal Holloway University of London*, 2012.

[12] T. Pesch, S. Schröders, H. J. Allelein and J. F. Hake, “A new Markov-chain-related statistical approach for modelling synthetic wind power time series,” *New Journal of Physics*, vol. 17, May 2015.

[13] ASM educational center, INC. (ASM), “CISSP security mechanisms,” 3 March 2016. [Online]. Available: http://asmed.com/cissp-security-mechanisms/.

[14] S. Fehis, O. Nouali and M.-T. Kechadi, “A new distributed Chinese Wall Security Policy model,” *Journal of Digital Forensics, Security and Law*, vol. 11, no. 4, 2016.

[15] Zissis and D. Lekkas, “Addressing Cloud computing security issues,” Futur. Gener. Comput. Syst., vol. 28, no. 3, pp. 583–592, Mar. 2012.

[16] T. Ristenpart, E. Tromer, H. Shacham, and S. Savage, “Hey, you, get off of my Cloud: exploring information leakage in third-party compute Clouds,” Comput. Commun. Secur., vol. 25, no. 3, pp. 199–212, 2009.

[17] M. a. Vouk, “Cloud computing — Issues, research and implementations,” ITI 2008 - 30th Int. Conf. Inf. Technol. Interfaces, pp. 31–40, Jun. 2008.

[18] Harauz, L. Kaufman, and B. Potter, “Data Security in the World of Cloud Computing ?,” IEEE Secur. Priv. Mag., vol. 7, no. 4, pp. 61–64, 2009.

[19] P. Barham, B. Dragovic, K. Fraser, S. Hand, T. Harris, A. Ho, R. Neugebauer, I. Pratt, and A. Warfield, “Xen and the art of virtualization,” ACM SIGOPS Oper. Syst. Rev., vol. 37, no. 5, p. 164, 2003.

[20] M. Armbrust, A. D. Joseph, R. H. Katz, and D. A. Patterson, “Above the Clouds : A Berkeley View of Cloud Computing,” Science (80-.)., vol. 53, no. UCB/EECS-2009–28, pp. 07–013, 2009.

[21] M. T. Khorshed, a. B. M. S. Ali, and S. a. Wasimi, “A survey on gaps, threat remediation challenges and some thoughts for proactive attack detection in Cloud computing,” Futur. Gener. Comput. Syst., vol. 28, no. 6, pp. 833–851, Jun. 2012.

[22] P. Saripalli and B. Walters, “QUIRC: A Quantitative Impact and Risk Assessment Framework for Cloud Security,” Cloud Comput. (CLOUD), 2010 IEEE 3rd Int. Conf., pp. 280–288, Jul. 2010.

[23] W. a Jansen, “Cloud Hooks: Security and Privacy Issues in Cloud Computing,” 2011 44th Hawaii Int. Conf. Syst. Sci., pp. 1–10, Jan. 2011.

[24] F. Gens, “IT Cloud services user survey, pt. 2: Top benefits & challenges,” 2008.

[25] P. Mell and T. Grance, “The NIST Definition of Cloud Computing,” 2009.

[26] A. Azeez, S. Perera, D. Gamage, R. Linton, P. Siriwardana, D. Leelaratne, S. Weerawarana, and P. Fremantle, “Multi-tenant SOA Middleware for Cloud Computing,” 2010 IEEE 3rd Int. Conf. Cloud Comput., pp. 458–465, Jul. 2010.

[27] R. Wu, G.-J. Ahn, H. Hu, and M. Singhal, “Information flow control in Cloud computing,” in 2010 6th International Conference on Collaborative Computing: Networking, Applications and Worksharing (CollaborateCom), 2010, pp. 1–7.

[28] SecaaS Working Group, “Defined Categories of Service 2011,” Security, p. 27, 2011.

[29] M. Carvalho, “SECaaS – Security as a Service,” ISSA J., no. October, pp. 20–24, 2011.

[30] Verizon, “2012 data breach investigations report,” 2012.

[31] Cloud Security Alliance, “Security Guidance for Critical Areas of Focus in Cloud Computing V2.1,” Security, vol. 1, no. December, pp. 1–76, 2009.

[32] S. Subashini and V. Kavitha, “A survey on security issues in service delivery models of Cloud computing,” J. Netw. Comput. Appl., vol. 34, no. 1, pp. 1–11, Jan. 2011.

[33] A. Celesti, F. Tusa, M. Villari, and A. Puliafito, “Three-phase Cross-Cloud federation model: The Cloud SSO authentication,” Proc. - 2nd Int. Conf. Adv. Futur. Internet, AFIN 2010, pp. 94–101, Jul. 2010.

[34] L. Malhotra, D. Agarwal, and A. Jaiswal, “Virtualization in Cloud Computing,” J. Inf. Technol. Softw. Eng., vol. 04, no. 02, pp. 2–4, 2014.

[35] D. Teneyuca, “Internet Cloud security: The illusion of inclusion,” Inf. Secur. Tech. Rep., vol. 16, no. 3–4, pp. 102–107, Aug. 2011.

[36] Cloud Security Alliance, “Top Threats to Cloud Computing,” Security, no. March, pp. 1–14, 2010.

[37] J. Brodkin, “Gartner: Seven Cloud-computing security risks,” InfoWorld, vol. July, pp. 2–3, 2008.

[38] T. Dillon, C. Wu, and E. Chang, “Cloud Computing: Issues and Challenges,” 2010 24th IEEE Int. Conf. Adv. Inf. Netw. Appl., pp. 27–33, 2010.

[39] Intel IT Centre, “Planning Guide Cloud Security,” 2011.

[40] M. Yildiz, J. Abawajy, T. Ercan, and A. Bernoth, “A Layered Security Approach for Cloud Computing Infrastructure,” 2009 10th Int. Symp. Pervasive Syst. Algorithms, Networks, pp. 763–767, 2009.

[41] S. Mansfield-Devine, “Danger in the Clouds,” Netw. Secur., vol. 2008, no. 12, pp. 9–11, Dec. 2008.

[42] W. Jansen and T. Grance, “Guidelines on Security and Privacy in Public Cloud Computing,” Director, vol. 144, no. 7, pp. 800–144, 2011.

[43] G. Blair, F. Kon, W. Cirne, D. Milojicic, R. Ramakrishnan, D. Reed, and D. Silva, “Perspectives on Cloud computing: Interviews with five leading scientists from the Cloud community,” Journal of Internet Services and Applications, vol. 2, no. 1. Springer London, pp. 3–9, 03-Jun-2011.

[44] S. Pearson and A. Benameur, “Privacy, Security and Trust Issues Arising from Cloud Computing,” 2010 IEEE Second Int. Conf. Cloud Comput. Technol. Sci., vol. 8, no. 6, pp. 693–702, Nov. 2010.

[45] H. Takabi, J. B. D. Joshi, and G.-J. Ahn, “Security and Privacy Challenges in Cloud Computing Environments,” IEEE Secur. Priv. Mag., vol. 8, no. 6, pp. 24–31, 2010.

[46] M. Almorsy, J. Grundy, and I. Müller, “An analysis of the Cloud computing security problem,” the proc. of the 2010 Asia Pacific Cloud …. 2010.

[47] A. S. Ibrahim, J. Hamlyn, and J. Grundy, “Emerging Security Challenges of Cloud Virtual Infrastructure,” in Proceedings of APSEC 2010 Cloud Workshop, 2010.

[48] R. Chakraborty, S. Ramireddy, T. S. Raghu, H. R. Rao, and B. Buffalo, “The Information Assurance Practices of Cloud Computing Vendors,” IEEE Comput. Soc., vol. 12, no. 4, pp. 1–8, 2010.

[49] Z. Chen and J. Yoon, “IT Auditing to Assure a Secure Cloud Computing,” 2010 6th World Congr. Serv., pp. 253–259, Jul. 2010.

[50] S. Bleikertz and M. Schunter, “Security Audits of Multi-tier Virtual Infrastructures in Public Infrastructure Clouds,” Proc. 2010 ACM Work. Cloud Comput. Secur. Work., pp. 93–102, 2010.

[51] H. Al-Aqrabi, L. Liu, and J. Xu, “Governance Compliant Service Oriented Computing,” .

[52] E. Keller, J. Szefer, J. Rexford, and R. B. Lee, “NoHype: Virtualized Cloud mInfrastructure without the Virtualization,” in Proceedings of the 37th annual international symposium on Computer architecture, 2010, p. 350.

[53] A. Ciuffoletti, “Monitoring a virtual network infrastructure: an IaaS perspective,” ACM SIGCOMM Comput. Commun. Rev., vol. 40, no. 5, pp. 47–52, 2010.

[54] H. Bar-El, “Introduction to Side Channel Attacks.”

[55] D. Chisnall, The Definitive Guide to the Xen Hypervisor. Prentice Hall, 2008.

[56] C. Reiss, J. Wilkes, and J. J. L. Hellerstein, “Google cluster-usage traces: format+ scheme,” Google Inc., …, pp. 1–14, 2011.

[57] S. Di, D. Kondo, and W. Cirne, “Characterization and Comparison of Google Cloud Load versus Grids,” Int. Conf. Clust. Comput. (IEEE Clust., pp. 230–238, 2012.

[58] C. Reiss, a Tumanov, and G. Ganger, “Towards understanding heterogeneous Clouds at scale: Google trace analysis,” … Cent. Cloud …, 2012.

[59] C. Reiss, A. Tumanov, G. R. Ganger, R. H. Katz, and M. a. Kozuch, “Heterogeneity and dynamicity of Clouds at scale,” Proc. Third ACM Symp. Cloud Comput. - SoCC ’12, pp. 1–13, 2012.

[60] Z. Liu and S. Cho, “Characterizing machines and workloads on a Google cluster,” Proc. Int. Conf. Parallel Process. Work., pp. 397–403, 2012.

[61] I. S. Moreno, P. Garraghan, P. Townend, and Jie Xu, “An Approach for Characterizing Workloads in Google Cloud to Derive Realistic Resource Utilization Models,” 2013 IEEE Seventh Int. Symp. Serv. Syst. Eng., vol. 60, no. 12, pp. 49–60, 2013.

[62] Y. Zhang, A. Juels, M. Reiter, and T. Ristenpart, “Cross-VM side channels and their use to extract private keys,” Proc. 2012 …, p. 305, Oct. 2012.

[63] S. Alarifi and S. D. Wolthusen, “Mitigation of Cloud-internal denial of service attacks,” in Proceedings - IEEE 8th International Symposium on Service Oriented System Engineering, SOSE 2014, 2014, pp. 478–483.

[64] Y. Zhang, A. Juels, A. Oprea, and M. K. Reiter, “HomeAlone: Co-residency detection in the Cloud via side-channel analysis,” in Proceedings - IEEE Symposium on Security and Privacy, 2011, pp. 313–328.

[65] S. Yu, X. Gui, and J. Lin, “An Approach with Two-stage Mode to Detect Cache-based Side Channel Attacks,” in Proceedings of the International Conference on Information Networking 2013 (ICOIN), 2013, pp. 186–191.

[66] P. Li, D. Gao, and M. Reiter, “Mitigating Access-Driven Timing Channels in Clouds Using StopWatch,” in Proceedings of the IEEE/IFIP International Conference on Dependable Systems and Networks (DSN), 2013, pp. 1–12.

[67] F. Liu, L. Ren, and H. Bai, “Mitigating Cross-VM Side Channel Attack on Multiple Tenants Cloud Platform,” J. Comput., vol. 9, no. 4, pp. 1005–1013, Apr. 2014.

[68] G. Stoneburner, A. Goguen, and A. Feringa, “Risk Management Guide for Information Technology Systems Recommendations of the National Institute of Standards and Technology,” 2002.

[69] F. Rocha, T. Gross, and A. Van Moorsel, “Defense-in-depth against malicious insiders in the Cloud,” Proc. IEEE Int. Conf. Cloud Eng. IC2E 2013, pp. 88–97, 2013.

[70] D. F. C. Brewer and M. J. Nash, “The Chinese Wall security policy,” in Proceedings. 1989 IEEE Symposium on Security and Privacy, 1989, pp. 206–214.

[71] J. O. Iglesias, L. Murphy, M. De Cauwer, D. Mehta, and B. O’Sullivan, “A Methodology for Online Consolidation of Tasks through More Accurate Resource Estimations,” 2014 IEEE/ACM 7th Int. Conf. Util. Cloud Comput., pp. 89–98, 2014