INSIDER THREAT REDUCTION MODEL FOR THE CLOUD ENVIRONMENT

LUCKY NKOSI

(200711510)

(B.Sc. Hons. Computer Science)

(University of Zululand)

A dissertation submitted in fulfillment of the requirements for the degree of

Master of Science in Computer Science

Department of Computer Science,

Faculty of Science and Agriculture

University of Zululand

RSA

Supervisor: Mr. P Tarwireyi

Co-supervisor: Dr. J Oladosu

2013

DECLARATION

I acknowledge that I have read and understood the University's policies and rules applicable to postgraduate research, and I certify that I have, to the best of my knowledge and belief, complied with their requirements. I declare that this dissertation is saving for the supervisory guidance received the product of my own work and effort. I have, to the best of my knowledge and belief, acknowledged all sources of information in line with normal academic conventions. I further certify that this dissertation is original, and that the material has not been submitted, either in whole or in part, for a degree at this or any other university. I have subjected the document to the University's text-matching and/or similarity-checking procedures. Parts of this work were published and presented at SATNAC 2011 in South Africa, and at ISSA Conference 2013 in South Africa. Another part of this work will be published in ICAST 2013 Conference in South Africa.

Signature:……..

Print Name:.....

Date: ……..

DEDICATION

I dedicate this work to my parents, Ndaloyomdali and Nqoba Kuhlekonke Zibusiso Nkosi.

ACKNOWLEDGEMENT

I would like to thank the Head of the Centre, Prof. M.O. Adigun, for the opportunity and his guidance, advice and support. I would also like to extend my gratitude to my supervisor, Mr Paul Tarwireyi for guidance and encouragement he provided throughout the course of my study and my co-supervisor, Dr John Oladosu for his support and advice. I would like to give many thanks to Mr. E. Jembere, who has been very supportive and has sacrificed his time in guiding me throughout my work. I would also like to extend my gratitude to my family for their understanding and support. I wish to give many thanks to my Sponsors, Telkom for funding me throughout my research.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

ABSTRACT

Cloud computing is a growing paradigm that offers a lot of benefits to cloud users. Despite the potential benefits that cloud computing could offer to business and individuals, security threat remains one of the growing concerns that are hindering the adoption of this paradigm. Specifically, the Insider threat is of greatest concern. In the traditional systems, insiders are regarded as current employees, former employees and any stakeholders that have access to the system. However, in the cloud computing environment the scope of insider expands to include contractors, administrator and employees of the cloud service provider. All these have the potential to compromised customer data stored in the cloud. As a result, customers are not comfortable with the idea of moving their critical information to the cloud service provider. The challenge then becomes, how to ensure that malicious insiders do not compromise the security of customer data and applications. Solutions are still needed to ensure that the data stored in the cloud is secure from malicious insiders of the cloud service provider. In an effort to address this problem, this work presents an insider threat reduction model for the cloud environment. The model uses sequential rule mining techniques to reason about the behaviour patterns of the user and predict whether a user is a normal user or a malicious user who has masqueraded in the system. A rule learning algorithm was developed and used in learning the behavior pattern of users, in order to build user profiles. Matching algorithm was also developed and then used to match the historical behavior of the user with the current behavior, in order to identify users that masquerade in the system as normal user. The result obtained proved that the proposed insider threat reduction model of the cloud environment maybe an effective solution in reducing insider attacks that originated from malicious users by accurately predicting whether a user was normal user or malicious user based on the behaviour patterns.

Chapter 1

INTRODUCTION

1.1. Overview

Cloud Computing has recently emerged as a new computing paradigm which has the potential to reach greater heights as reached by the Internet revolution (*Rocha et al. 2011*). In this type of computing everything is dynamically delivered as a service based on user demand, it uses a pay-per-use model for enabling available, convenient, on-demand network access to a shared pool of configurable computing resources that can be rapidly provisioned and released with a smallest service provider interaction (*Casola et al. 2010, Syam Kumar et al. 2010*).

The cloud provides lots of benefits to individuals, business and SMMEs, by allowing them to pay for what they use only and also help them not to build their own infrastructure from scratch. This provides users with the infrastructure they can use in order to reduce the expenditure by migrating their business storage containing confidential information into the cloud. This means that, just like water and electricity, subscribers only pay for the platform, software and infrastructure services they use, and as a result, this reduces the need for infrastructure investments, which are usually high.

The evolution of Grid computing, Service-Oriented Architecture and virtualisation technology has led to a new computing paradigm that provides various resources such as platform, infrastructure and software as a service (*Sundararajan et al. 2011*). Cloud computing takes advantage of Grid computing, web services, service-oriented architecture and virtualization

technology as building blocks to provide a smooth development where users can deploy their applications as a result, makes cloud to be a globally acceptable in many organisations.

Despite the global acceptance of the cloud and its widespread, individuals and businesses are still reluctant to migrate their business data and applications to the cloud due to security issues (*Kuyoro et al. 2011*). Security is one of the growing concerns that hinder the adoption of the cloud and is the highest ranked challenging issue in the cloud computing environment (*Zhang et al. 2010, Saltan, 2010*). Although other scholars have identified many security threats in the cloud computing environment, the malicious insider represent a significant concern (*Rocha et al. 2011, Brunette et al. 2009*). As a result, many organisations are not comfortable with the idea of moving their data and applications to systems they do not control. Because, the migration of workloads to such shared infrastructure usually widens the security threat surface, resulting in an increase in the probability of unauthorised access and exposure. Moreover, there is no guarantee that companies will not lose their leading edge against competitors when their trade secrets are exposed due to security breaches (*Manifesto, 2010*).

An insider is regarded as an employee of the cloud provider with the privileges to access an IT system. A malicious insider is an entity or administrator of the cloud service provider who intentionally surpasses or misuse privileges in a way that negatively affects the confidentiality, availability or integrity of the organisations information system (*Eom et al. 2011*). Although insider attacks originating from cloud administrator may not occur frequently, they have a higher rate of success, can go undetected and pose a much greater risk to the information stored in the cloud. This is due to the fact that insiders enjoy certain important advantages of having high privileges accessing in the system, insiders are familiar with their target and security measures in place to prevent unauthorised access to the system. This type of users can steal credential of the

legitimate users and masquerade in the system by pretending as if there are real normal users. As a result, making it difficult for cloud service providers to differentiate between the behaviour patterns of the normal user and masquerader in order to verify whether the users, who claim to be, is the one based on the patterns generated (*Chinchani et al. 2005*).

A masquerader is the user who pretends to be an authorised user through the use of stolen credentials in order to gain access to the system. A masquerader has two particular features that make it difficult for cloud service providers to prevent and detect. One is that, a masquerader has legitimate access to the system, and the other, is that malicious access sequence of events performed by a masquerader can be similar to those performed by normal user.

In view of the insider threat problem, preventive controls, detective controls and response controls have been used to mitigate insider attacks in the cloud (*Claycomb et al. 2012*). Organisations have implemented different preventative controls in order to mitigate the threat originating from individuals with the legitimate access to the system. However, preventative controls are less effective because malicious insiders are familiar with the security controls that are used to prevent unauthorised user from accessing the system. Different detective controls have been implemented to detect a sequence of events that are performed by malicious users that pretend to be a normal user in the system. Lastly, the organisation has an incident response plan to mitigate the damage resulting from malicious insider actions.

The CERT model given in Figure 1 shows the technical controls and non-technical controls that are used in mitigating malicious insider actions.

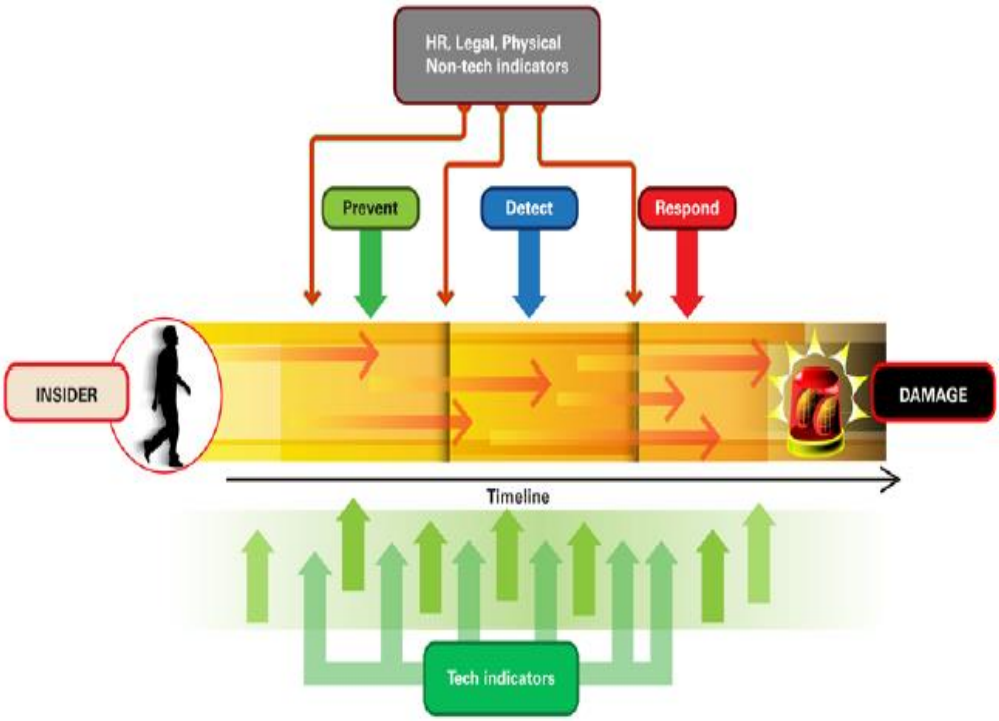


Figure 1: Controls for mitigating insider (CERT model) (*Claycomb et al. 2012*)

From Figure 1, the top level represents non-technical controls such as human resource (HR) physical etc, while the lower level represents the technical controls such as remote access log. From the Figure presented above, the main focus of this research work is in detective controls because it is assumed that all preventative controls have been compromised or bypassed and the intruder have potentially entered into the system. Hence, detective control is used as a second line of defense to mitigate insider attacks. Monitoring and detection are often used exchangeable. The malicious insider can be identified by monitoring or observing their patterns of information use.

Different monitoring techniques have been proposed in order to address the problem of insider threat whereby a malicious user pretends be a legitimate user in the system such techniques

include predictive pattern generation (*Teng et al. 1990*), intrusion detection systems (*Nguyen et al. 2003*) and composite role based monitoring (*Park et al. 2004*).

Predictive pattern generation monitoring technique has been used to predict the next action to be executed by the insider by relying on the previous set of actions performed by the user (*Teng et al. 1990*). However, relying on the previous actions that are not coming from a training dataset to judge whether a user is malicious or not is not sufficient to clearly state whether a user is malicious.

Intrusion detection systems are one of the monitoring techniques that is mostly implemented in security system, because it's widely accepted as a promising approach in monitoring behaviour patterns of the user (*Nguyen et al. 2003*). This approach uses rules for discovering the anomalous behaviour of the user. It uses events generated by the system such as login time and sequence of events performed by the user. However, intrusion detection system is not effective when countering a professional masquerader and this approach monitoring technique is not capable of detecting attacks whose rules are not available. As a result, it misclassifies rules that lead to an attack as the normal behaviour of the user. In that way a sequence of events performed by masquerader will be regarded as normal actions.

One of the monitoring approaches that have been proposed in addressing the insider threat problem is the composite role based monitoring (*Park et al. 2004*). The widely acceptance of composite role based monitoring it is based on the fact that, it provides the advantage of scalable administration and implements the concept of least privilege and separation of duties amongst different administrators. This approach ensures that users are monitored by comparing the current task with the expected behaviour in the system. Despite the advantages that the

composite role based monitoring can provide in addressing insider threat problem, it still has some pitfalls, it cannot be able to differentiate the sequence of events performed by normal user and a masquerader. As a result, malicious administrators who pretend to be normal users in the system are responsible for the largest proportion of data breaches within organisations, such as misuse of information assets (*Baker et al. 2010, Brodkin et al. 2008*). This is due to the fact that, the behaviour patterns or actions performed by masquerader are misclassified as normal actions from legitimate users, which results in the increase of insider attacks which this research work seeks to address in the cloud environment. There is a necessity of the insider threat reduction model to deal with the unexpected behaviour pattern that originates from the malicious administrators who pretend to be normal user in the system. This work was specifically aimed at providing a monitoring approach by exploring the use of the sequential rule mining technique in order to predict the behaviour of the normal user and the behaviour of the user who pretend to be a legitimate user when accessing the system.

1.2. Statement of the Problem

Cloud computing is gaining global acceptance and has potential to bring many advantages to business and individuals, however, security remains one of its major concerns (*Casola et al. 2010, Sundararajan et al. 2011*). Cloud computing widens the insider threat surface by adding a whole new group of people such as cloud administrators and cloud contractors beyond the original organisations employees. This means that, besides the inherent organisational threats, the cloud provider's employees will pose additional insider threats such as disclosing and stealing data that is property of the cloud user. The biggest challenge that faces the cloud service

provider is the inability to detect malicious users when accessing the system using the credentials of the legitimate user. Because current monitoring techniques such as intrusion detection and composite role based monitoring cannot differentiate between actions performed by a masquerader and the normal user in the system (*Park et al. 2004, Nguyen et al. 2003*). The study, therefore, developed an insider threat reduction model in order to identify masqueraders and normal users based on the sequence of events generated by the system.

1.3. Research Questions

The main question that the research work should answer is:

How could insider attacks originating from the cloud service provider be minimised in the cloud environment?

To answer the question, the answers to the following sub-questions are investigated and a bottom up approach was used to answer the main question.

i. What are the currently existing insider threats and vulnerabilities in the cloud computing environment?
ii. What are the current solutions pertaining to the threats and vulnerabilities and their limitations?
iii. How can a solution to minimise insider threats in cloud environments be crafted?

1.4. Rationale of the Study

Cloud computing is gaining popularity due to its ability to provide dynamic scalability and elasticity of resources. The cloud reduces the need for SMMEs to own the infrastructure thereby helping them to cut down on expenditure by migrating storage and hosting on to the cloud. Even though cloud computing offers a lot of benefits, the major concerns that hinder the adoption of the cloud include security issues such as insider threats. The insider threat problem is an on-going research in cloud computing (*Glott et al. 2011*). This problem arises when the storage containing sensitive information about the organisationis migrated to the cloud service provider. Where the storage owner does not have control over their personal information stored in the cloud in terms of who access it and when. This may result administrator of the cloud service provider acting in the malicious way by masquerading in the system as a normal user and performing an operation that are not associated with their job role.

According to the Verizon 2010 data breach report (Baker, 2010), there is a 26% increase in data breaches by malicious insiders and 52% of the organisations surveyed characterized the incidents arising from insider threats as primarily deliberate and only 19% believed that insider threat incident were predominantly accidental and 26% reported to be the equal combination. This indicates that the insider threat problem is a major issue. Combined findings indicate that malicious insider data breaches are the most costly data breach incidents. Countries such as United State and German continue to incur the most expensive data breach incidents caused by malicious insiders at $277 and $214 per compromised record (*Romanosky et al. 2011*). This also indicates that there is a need for a security mechanism to deal with malicious that deviates from expected behaviour patterns.

The solution approach taken in this research in addressing the insider threat problem would help in identifying the sequence of events performed by normal user and actions performed by a malicious user who pretends to be a normal user when using the system through monitoring behaviour patterns of the user who manages the cloud. This would go a long way in improving the adoption of the cloud.

1.5. Research Goal and Objectives

1.5.1. Goal

The main goal of the research was to develop an insider threat reduction model that would minimise insider attacks in the cloud environment by monitoring behaviour pattern of insiders.

1.5.2. Objectives

The specific objectives of this research were:

1. To conduct a literature survey of existing insider threats and how they occur in the cloud environment.
2. To survey existing techniques used to solve insider threat problem with the view of finding ways in which insider threat can be minimised based on the strength of the reviewed techniques.
3. To develop a model aimed at minimising insider attack in a cloud environment.
4. To evaluate the proposed model. This will include implementation and evaluation of the model proposed in objective (3).

1.6. Research Methodology

The above research objectives were accomplished by using the following methods, namely literature survey, model development, and proof of concept. These methods are discussed in details in the following sub-sections:

1.6.1. Literature Survey

The aim of literature survey was to provides the current state of the art on the current security mechanism that are currently employed to mitigate the malicious insider in different areas of security including traditional system and in the cloud environment. This work explores the encryption techniques, access control approaches and monitoring approach on how they deal with the insider threat problem. The result of the literature survey was also used to propose a model that was suitable for solution approach that was provided. Thereafter, a survey of the metrics needs to be considered when evaluating insider threats approach to mitigate malicious insider.

1.6.2. Model Formulation

The model development needed a critical approach of evaluating what other researchers have done in the field of security in relation to insider threat, in view of finding strong points of the existing techniques. The knowledge gained here provided the platform for the development of the model.

1.6.3. Proof of Concept

A prototype of the proposed insider threat reduction model for cloud environment was implemented and evaluated as a proof of concept. Appropriate accuracy parameters were used for evaluation.

1.7. Organisation of the Dissertation

The rest of the dissertation is organised as follows:

Chapter 2 discusses a background on the important concepts which include cloud computing, insider threat and malicious insider. Chapter 3 presents the results of the literature review by exploring the different existing security mechanism proposed by other researchers in addressing the insider threat problem and identify the gaps. In chapter 4 we present the description of the insider threat reduction model development. This chapter discusses design requirement of our solution and present solution approach adopted. Finally, the formulated model aimed at minimising the insider threat is presented. Chapter 5 deals with the implementation of our model. The evaluation and result discussion of our insider threat reduction model are presented in Chapter 6. Chapter 7 concludes the research and also presents some direction for future work.

Chapter 2

BACKGROUND

2.1 Introduction

The growth of day-to-day business operations for small and medium business has led to the adoption of cloud computing, as a means of storing of critical data about their business. Cloud computing is an exciting technology that shows the most significant shifting in today IT infrastructure. However, there is still a lot of work to be done in the area of security, because cloud inherits all security issues from traditional systems and those that are presented by its unique architecture. In most cases, security policies depend on how much of the computing resources do you own and is under your control. If someone else is running your computing resources, you need to implement strategies to stay secure. Moving critical data into the cloud environment means that you will have less control. The issue now is how you will be able to deal with security threats such as insider threat that are even difficult to mitigate in systems which you control (*Bouchahda et al. 2010*).

In section 2.2 we discuss cloud computing, its deployment models and delivery models. Section 2.3 gives an overview of security issues that is related to the new computing paradigm while Section 2.4 presents the security threats associated with information stored in the cloud. Section 2.5 presents identified threats and vulnerability associated with the malicious insider. Section 2.6 presents the classification of insider threat. Section 2.7 introduces the concept of sequential rule mining. Section 2.8 gives a summary of the chapter.

2.2 Cloud Computing

Cloud computing has gained global acceptance because of its five major characteristics which include on-demand self-service, measure service, rapid elasticity, location-independent resource pooling and ubiquitous network access. These characteristics are geared towards achieving seamlessly cloud (*Takabi et al. 2010*)

The major key attributes of cloud computing that differentiates it from other computing models include (i) the provisioning of services in a remote manner rather than locally, (ii) provisioning of computer power dynamically in a cost effective way and the ability to scale. This is because everything is being managed by the cloud service provider by allowing cloud user to pay for only what they use only. Cloud architecture is categorised into three different deployment models and three delivery models or service layers.

As Figure 2.1 depicts the cloud architecture with its own entities involving the service delivery models and deploys models in the whole cycle of cloud deployment.

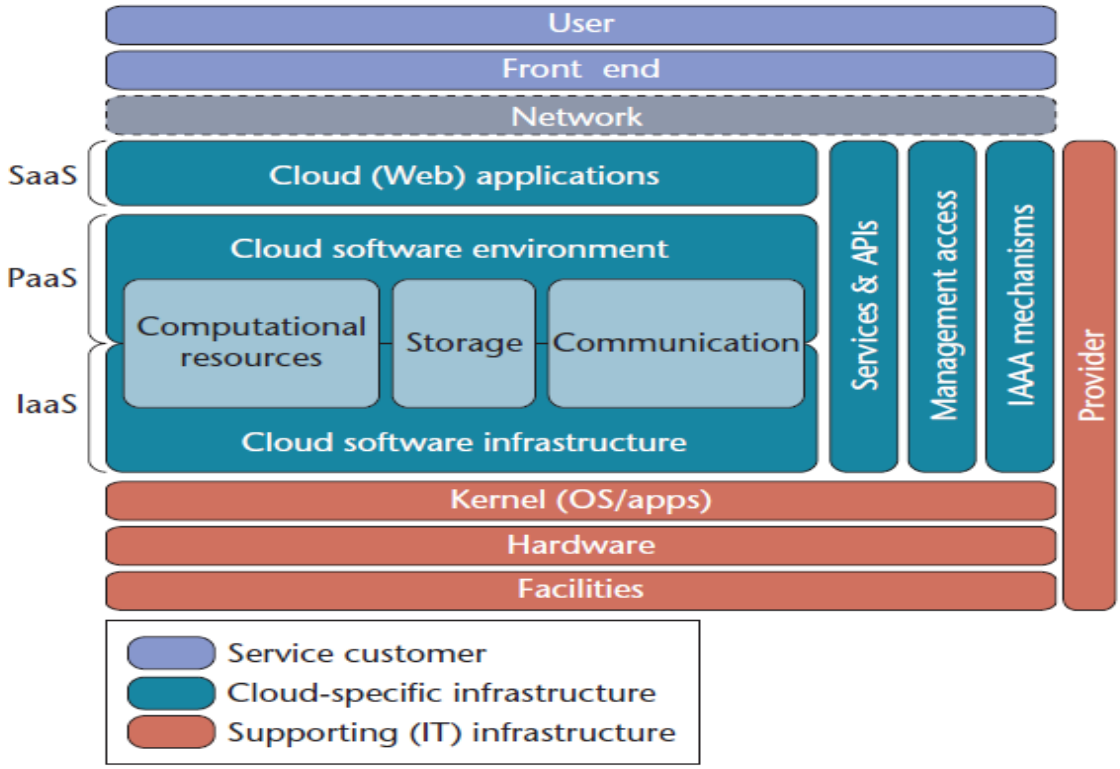


Figure 2.1: Cloud Reference Architecture (*Grobauer et al. 2011*)

The cloud encompasses two entities with different roles which are as follows:

- **Cloud Service Provider/Cloud Provider/Provider -** provide services on the cloud that can be accessed by cloud customer such as Infrastructure as-a-Service (IaaS), Platform as-a-Service (PaaS) and Software as-a-Service (SaaS).
- **Cloud Customer/Client/Cloud User** – uses services that are provided by the cloud provider.

In some cases a service provider can also be a cloud customer of another service provided in the cloud for example when Platform is a service. Cloud computing has three different deployment models namely public cloud, private cloud and hybrid cloud. In public cloud services are managed off-premises by third-party provider (*Ramgovind et al. 2010*). Services in private cloud

are managed on-premises by one organisation (*Bamiah et al. 2011*). Hybrid cloud is the combination of both private and public cloud (*Sandhu et al. 2000*). Cloud computing has three delivery models which include Software-as a Service (SaaS), Platform-as a Service (PaaS) and Infrastructure-as a Service (IaaS)

2.2.1 Benefits of the Cloud Computing

a. **Reduced cost of IT investment:** Using cloud computing allows business to reduce IT investment or expenditure like hardware and software that are needed for the functionality of the business. Using the cloud helps in reducing the cost in a way that business outsource everything from the cloud provider.
b. **Not a centralised working environment:** Cloud computing takes away the disadvantage for restricting employee to be on the centralised working environment by providing the flexibility for the employee to have in and outdoor workspace for working environment.
c. **Scalability:** Cloud computing provides flexibility to allow small business to scale up while decreasing the maintenance of IT infrastructure.
d. **Information back-up:** Cloud provides an advantage to small business that their business information is backed up by the cloud service provider. However, backing up information raises some concern around data protection and threats in a cloud environment.

2.3 Security in the Cloud Computing

Cloud computing has been acknowledged as a key promising technology and market development for small and medium business. However, security is one of the key contributing factors that hinder the uptake of the cloud. Organisations and businesses are still reluctant about deploying services in shared environments and are not comfortable with the idea of storing their

critical data in systems they do not control (Manifesto, 2010). Cloud still suffers from security risk and threats that prevent cloud customers from trusting it. In addition, cloud also adds a level of risk in the data that is stored on the cloud provider side, because data storage is outsourced to the external third-party provider. Maintaining security objectives such as integrity, confidentiality and availability of critical data make it harder (*Zhang et al. 2010, Choubey et al. 2011*). The cloud provider promises to ensure integrity and confidentiality of information by singing Service-Level Agreement (SLA) with the customer that information is secured according to what service provider claim in SLA. The Service-Level Agreement is a service contract between the cloud consumer and provider. One of the security requirements that must be met in the cloud by provider are discussed as below:

2.3.1 Confidentiality

In cloud computing, confidentiality is one of the security objectives that must be met by the cloud service provider to ensure that protected information is accessed by legitimate users who are authorised to access information. Securing confidential of sensitive information in order to ensure that is not seen by unauthorised users is a challenging task. Hence encrypting the data in order ensure information is not viewed by unauthorised users is seen as a potential security mitigating mechanism (*Takabi et al. 2011*).

2.3.2 Integrity

Integrity is the requirement of assuring that critical information hosted by cloud provider both in transit and at rest is protected against any unauthorised access that results in improper modification.

2.3.3 Availability

Availability is another security objective that ensures a timely and reliable access of cloud resource by cloud customer. The cloud resource needs to be available at any time to cloud customer regardless of any failure when required by authorised users.

Figure 2.2 illustrates some challenges associated with the new computing paradigm. In this figure, security is cited as a most difficult challenges associated with the cloud (*Zhang et al. 2010*).

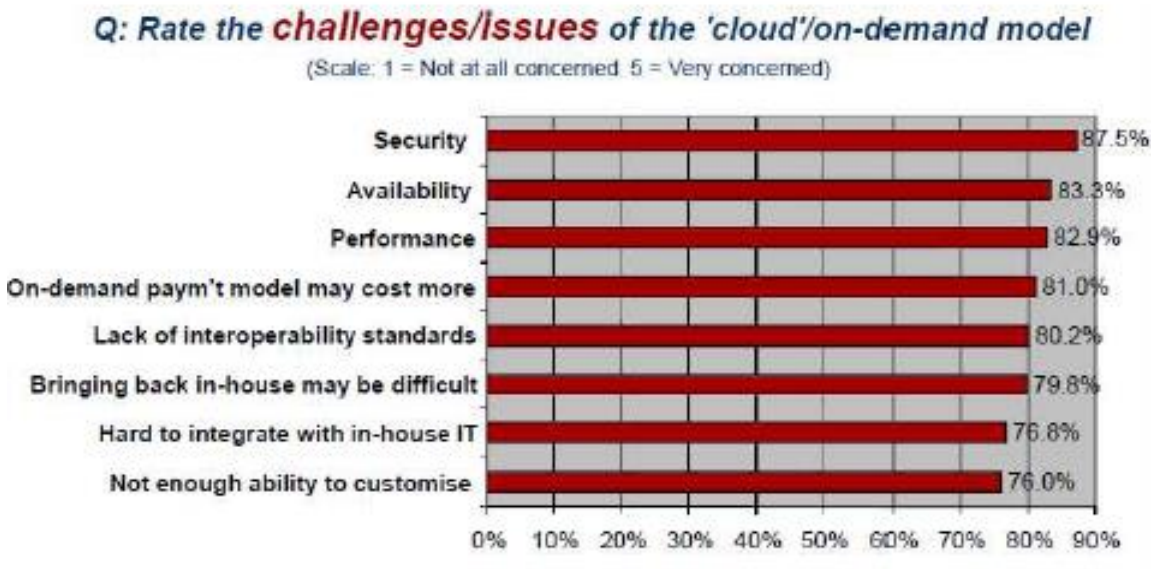


Figure 2.2: Challenges around cloud Computing Environment (*Zhang et al. 2010*)

Cloud computing is associated with many security challenges (*Garter et al. 2006*), such as:

- **Privilege user access:** because of the data being processed outside the organisation, security perimeters that amplify risk and threats of data being accessed by unauthorised users.

- **Data location:** data is hosted in different places with different security employed to secure such data as a result, data owner is left to trust the cloud provider about their information.
- **Data segregation:** increases the chance of data being compromised because it is stored in a shared environment.
- **Long-term viability**: cloud provider ensures that data remain available in case business loses market value.
- **Recovery:** Cloud providers are faced with the challenge of ensuring that data is available even after some disaster.

2.4 Security Threats in the Cloud Environment

One of the key factors in the cloud that amplify security threats and risk are the vulnerability that exists in the cloud computing environment. Before we go into detail about different types of threats that are found in the cloud, we need to take a close look on what is defined as risk, threat and vulnerability in relation to data that is stored by the cloud service provider. Moving data to a cloud environment introduces much vulnerability by creating a new surface of attacks. According to *Bernd et al. (2011*), **risk** is defined as the potential that a given threat can exploit vulnerabilities of an asset (in this case a data that is hosted in the cloud) and thereby cause harm to the organisation, by quantifying it in terms of probability of an event and consequences. **Threat** is a potential occurrence that can cause harm in the data stored in the cloud. **Vulnerability** is the weakness in the system, that the system cannot be able to resist the actions of an attacker when trying to cause harm. In any case vulnerability must be described in terms of certain attack that can either originate within the organisation or outside the organisation.

As indicated in section 2.3 cloud providers are concerned about ensuring that security objectives such as confidentiality, integrity and availability of data stored in the cloud are secured. If these security services are violated in the cloud service provider result in the privacy issues that raise concern about the data of the organisation that is hosted on the cloud. This type of computing environment takes new security dimension and introduces a lot of privacy risk of data stored in the cloud and it is associated with many threats and vulnerability that makes cloud to be an un-trusted computing platform. Threats that are introduced by this computing paradigm come in different format and pose serious challenges to the data belonging to cloud users. There are many attacks that are derived from potential threats that can cause loss of confidentiality, integrity and availability of information in the cloud computing, these attacks can occur ranging from application level when user login into the system until the infrastructure layer and cause the discomfort to the system operations.

The threats posed to an organisation can be categorised into two (*Jeffrey et al. 2008*)

- Outsider threats
- Insider threats

Threats that originate from outsiders are known as outsider threats, these types of threats are performed by individuals who do not have sufficient knowledge about the system and have no privilege of accessing the organisation assets, while those that originate from insider are known as insider threats (*Chinchani et al. 2005*). However, many organisations are aware of the threat that originates outside and have proper security mechanism to reduce attack originate from outsiders. But much focus has not been put on the insider threat. However, the focus of this study is not concerned much about the outsider threat but insider threat.

Insiders	**Outsiders**
Have knowledge about the system currently in place	Does not have knowledge about the system
Privilege users	Not privilege users
Trusted users have to critical information.	Not trusted users because they are not part of the organisation.

Table 2.1: Key Differences Between Insiders and Outsiders

The insider threat is a human centric issue that is more complex to solve using one technique due to some large amount of factors involved and the inherent unpredictable human behaviour. Many researchers define insider and insider threat in different ways depending on the domain or the context in which are considered. In our case, we consider the definition of insider provided by *Jeffrey et al.* (2009), *Marianthi et al.* (2005), *Rocha et al.* (2011), *Eberle et al.* (2010). An insider is defined as any entity with the legitimate access or trusted individual with the knowledge about the system. What is important is that once legitimate users have been granted any authorised explicit right to the information system, they are then considered as insiders. In the traditional systems, insiders are regarded as current employees, former employees and any stakeholders that have access to the system. However, in the cloud computing environment the scope of insider expands to include contractors, administrator and employees of the cloud service provider.

The growth of critical data being managed by the cloud service provider has resulted in the increase in the number of threats in the cloud environment. Some of these security threats are listed below:

- Abuse and Nefarious use of cloud
- Insecure Interfaces and APIs
- Malicious Insider
- Virtualised Technology
- Data Loss or Leakage
- Account or Service Hijacking
- Unknown Risk Profile

1. Abuse and Nefarious Use of the Cloud

Abuse of nefarious use of the cloud threat usually emanates due to the fact that the cloud provider allows their users to have unlimited access to storage and bandwidth capacity. As a result, users can hack the cloud, since some cloud provider free trial that gives opportunity to malicious users to crack password and launch potential attack (*Brunette et al. 2009*).

2. Insecure Interface and APIs

Exposing weak APIs to cloud service consumer to interact with cloud services has given rise to some security issues and amplify chances of information being accessed accidentally or intentionally (*Brunette et al. 2009*). Organisations need to ensure confidentiality, integrity and availability of information through providing strong security in APIs because security availability depends on APIs provided by the cloud provider.

3. Malicious Insider

Malicious insider originates from authenticated, an authorized and trusted employee of the cloud service provider. Employee of the cloud service provider can violate private information by breaking trust that is given to them. The lack of transparency increases the chances of the data being compromised by employees of the cloud provider (*Brunette et al. 2009*).

4. Shared Technology Issues

Sharing of infrastructure so that services can be delivered in a scalable raises some challenges in managing Virtual Machine (VM) that is hosted by cloud providers (*Brunette et al. 2009*).

5. Data Loss or Leakage

Storing data without employing encryption mechanism results in data loss or leakage. This may be a result of deletion or alteration of information done intentionally or accidentally by cloud administrators (*Brunette et al. 2009*). Another contributing factor to the data loss is the failure to provide the backup of the original data content and may be due to any operational failure and unreliable storage.

6. Account or Service Hijacking

Account hijacking refers to unauthorised access from attackers who want to gain access to information belonging to cloud customer through employing attacking methods such as phishing, fraud and to control user accounts (*Brunette et al. 2009*).

7. Unknown Risk Profile

The unknown risk profile is a low rated threat among the seven threats, this threat arises from the compliance of the internal security procedures. It is infeasible to review all available security threats in the cloud (*Brunette et al. 2009*). However, in this research we narrow down to one particular threat which is malicious insider threat.

In this research work we have assessed the techniques being used to mitigate such threats and its implication to the cloud environment. Since the focus of this research is more on insider threat, in the following section we will discuss identified threats and vulnerabilities that are associated with the malicious insider.

2.5 Identified Threats and Vulnerabilities

A threat caused by a legitimate insider of the cloud service provider is costly for the organisation that operates in the cloud. The malicious insider can explore privileges and knowledge they have acquired about the business and pose serious threat to valuable confidential information hosted in the cloud (*Duan et al. 2011*). We have identified threats and vulnerabilities that are most common to be exploited by malicious insiders of the cloud service provider. These threats include integrity violation, unauthorised access, and denial of services and disclosure of confidential data. Vulnerabilities include inadequate separation of privileges, availability of services, failing limit access to internal resource and insecure cryptography.

2.5.1 Integrity Violation

Integrity violation refers to the dishonest of the administrator that has been given a certain level of access to the system. This type of threat poses a serious challenge to cloud user privacy since malicious administrators of the cloud provider have access to the internal resource. A malicious administrator can easily alter or delete data belonging to the cloud user. The improper modification of cloud user data results in the loss of confidentiality and can make data to be completely unusable to the cloud user (*Park et al. 2006*). Improper modification occurs because the system that is currently implemented by the cloud provider permit the over privilege account to administrator that are managing the cloud. As a result, a malicious administrator can perform malicious actions to sensitive data of the cloud users. It also gives a malicious administrator advantage of performing malicious actions freely since no one is accountable for any damage that can occur to cloud user data since administrators are given access in the coarse grained format (*Li et al. 2010*).

2.5.2 Unauthorised Access

Unauthorised access refers to the viewing of accounts or files when the administrator has not been given access to perform such operations by the owners. Based on the knowledge that each administrator has acquired over time about the security system that is currently implemented to detect and prevent unauthorised access. As a result, the knowledge that has been acquired by malicious administrator gives the malicious administrator privilege to carry malicious activities in the sensitive data stored in the cloud (*Rocha et al. 2011*). Malicious administrators can breach the system in order to gain access to internal resource of the cloud without being detected and violate cloud user privacy. Using weak authentication mechanisms when authenticating

administrator, allows an administrator to access cloud user data and perform malicious actions that violate confidential information and integrity of information.

2.5.3 Denial of Service

Denial of service is a deliberate attempt of making cloud services not to be available to the intended users. A malicious administrator of the cloud provider can make the system unavailable to its intended cloud user by simply overloading the system with many requests making the system non responsive. This type of user can access encrypted data or enter the system as a lower administrator and elevates privileges by trying to access cloud resources that are managed by other administrators. When cloud users fail to access cloud services because of denial of service attack originating from malicious administrator this breaks the trust between cloud user and cloud provider (*Basescu et al. 2011*).

2.5.4 Information Disclosure

Information disclosure refers to unauthorised access to sensitive data gained by a malicious administrator of the cloud provider that causes the breach in the cloud user privacy by leaking the sensitive information. For example, a malicious administrator can access sensitive data about the organisation and sell it to competitors. The increase of information disclosure at cloud service provider side has changed many individuals and business perception about how their critical data are managed and accessed in the cloud service provider (*Asma et al. 2012*). The main vulnerability that allows a malicious administrator to access sensitive data of the cloud user, is because of the system in place that promotes the inadequate separation of privileges

among different administrators, in terms of who have access to what, under what conditions should one access the sensitive information belonging to the cloud user.

2.5.5 Inadequate Separation of Privileges

Having more than one administrator of the cloud service provider assigned to the same role in order to perform tasks in the system, allows malicious administrators to carry their malicious activities because there is no proper separation of duties between different administrators. Malicious administrators perform malicious activities knowing that it is difficult to trace who performed what and where because many administrators are clustered in one role (*Brunette et al. 2009, Asma et al. 2012*).

2.5.6 Availabilities of Services

Availability of service vulnerability is associated with the cloud storage infrastructure. If the infrastructure went down for some hours, this would result in the data loss and could lead to access issues that cause legitimate cloud customer not being able to access cloud services (*Basescu et al. 2011*). This raises so many questions such as "in case of failure what is the responsibility of cloud service provider in ensuring availability of service to cloud customer".

2.5.7 Failing to Limit Access to Internal Resources

A cloud service provider lacks the aspect of limiting access to internal resources. Malicious administrators are able to perform the escalation of privileges that will see a normal administrator being able to access files that are accessed by the manager. As a result, malicious administrators are able to alter, delete and view confidential information that is not required by their normal job activities (*Rocha et al. 2011*).

2.5.8 Insecure Cryptography

Storing sensitive information in the plaintext allows malicious administrator of the cloud service provider who is managing the cloud to steal confidential information. Using weak encryption techniques before the data is outsourced allow malicious administrator to decode the cryptographic mechanism and read the encrypted data easily.

With the insight gained after reviewing some threats and vulnerabilities, we then matched vulnerability to the threats that originated from trusted administrators of cloud service provider. Table 3.1 below presents the match of threats and vulnerabilities.

Threat	**Vulnerabilities**
Unauthorised Access	Using weak authentication mechanisms which result in allowing the malicious administrator to access the cloud user private keys that are stored in the plaintext.
Improper Modification of Data	Permitting over privilege account to administrators who are managing the cloud that can delete or modify cloud user private information.
Denial of Service	Elevation of privilege that give access to

	malicious administrators to make the system unavailable to cloud user.
Information Disclosure	Permitting more than one administrator to manage sensitive data without limiting the access to critical data and using inadequate separation of privileges among the cloud administrator.

Table 2.2**:** Matching Threats and Vulnerabilities

Table 2.2 shows a number of threats that had been identified that posed serious threat to cloud user data stored in the cloud.

2.6 Insider Threats

The occurrence of insider threat can be classified into two categories based on the impact they have on the organisation data. Insider can either be accidental or malicious. The occurrence of accidental threat usually originates from currently authorised and trusted individual within the organisation. However, the occurrence of such threat has no malicious intent in terms of violating security policy defined by organisation for securing the data. We argue that, even though accidental threat may pose a serious threat in the cloud computing environment, but stopping these types of threat from happening is not a difficult task compared with when dealing

with the malicious insider threat emanates from trusted individuals (*Theoharidou et al. 2009*). One way of preventing these types of threat from occurring when cloud administrators performs their operations with cloud user data, is to conduct a proper training awareness program with the new and current administrators of the cloud provider as a way to making them to be aware what is required of them (*Basescu et al. 2011*). Secondly, we also argue that providing an alert system as a way of preventing the elevation of privileges of the authorised cloud administrator unknowingly that might result in the violation of cloud user can be one of the solutions that can be adopted when dealing with the accidental threat (*Marianth et, al 2005, Malek et al. 2010*) .

The focus of this research is not solely based on accidental threat that can occur in the cloud environment, but is based on the malicious threat that can be performed by the administrator of the cloud provider. Preventing threat from a malicious insider in the cloud environment is one of the biggest challenges that cause various organisations to be nervous to adopt and host their confidential data on the cloud (*Grobauer et al. 2011, Brodkin et al. 2008*).

Insider threat originates from a current employee, administrator, contractor or former employee who deliberately exceeds or escalates his or her access privilege in the manner that can affect the confidentially, integrity and availability of information stored on the cloud. Some of the reasons that motivate insiders to perform malicious actions include financial gain, sabotage and disgruntlement (*Band et al. 2006*).

The Figure 2.1 below depicts the set of elements or attributes that define a malicious insider. In order to have much Insight about the whole cycle of insider, one needs to understand how a malicious insider operates. Access, privileges, knowledge, skills and motivation are key elements that amplify chances for insider to perform malicious action.

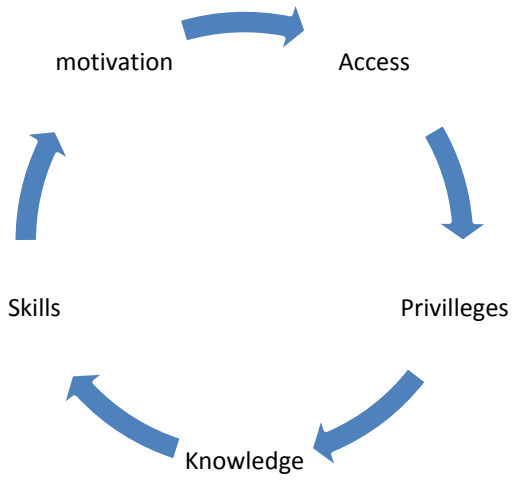


Figure 2.3: Key elements of an insider (*Basescu et al. 2011*).

If all the set of steps ranging from access to skills are mastered successfully by insiders, it creates a lot of problem to the cloud providers. Preventing someone who has access and privileges for a set of operations within the system is a challenging task, because after user have been given access and privileges their level of knowledge about the system increase over time. Once these types of users have acquired a sufficient knowledge about the system they can bypass security measures designed to prevent and detect or react to unauthorised access (*Basescu et al. 2011*). As a result, insiders become skillful enough and motivated to perform their malicious action with the understanding that their malicious actions go undetected.

An insider threat is regarded as any operation performed by legitimate users who are already trusted, authenticated and privileged who can intentionally exceed or misuse their level of trust and access in the manner that adversely affect the cloud user (*Brodkin et al. 2009*). A malicious insider is categorized into two classes, namely, a traitor and a masquerader. A traitor is defined as a user who has been given a certain level of access in the system but whose actions are encountered by policies, and whose goal is to negatively affect confidentiality, integrity and

availability of information assets (*Mark et al. 2005*). A traitor may exhibit the normal behaviour of the user who has access to the system and still perpetrates malicious activities. In this case, profiling behaviour of the user is less effective in identifying slight changes in a normal behaviour.

A masquerader is an attacker who succeeds in stealing credentials of the legitimate user and impersonates another user (*Mark et al. 2005).* This type of insider has less knowledge and is expected to perform inconsistent actions with the victim typical behaviour. In this research work we focus more on masquerader detection.

2.7 Sequential Rule Mining Knowledge

Sequential rule is defined as a relationship between two itemsets X, Y $\subseteq$ I such that X$\cap$Y = Ø and X, Y are not empty. These rules are in a form of X$\Rightarrow$Y, that is to say, if X appears, Y is most likely to appear afterward in the same set of sequence. The adaptation of sequential rules is typically used in sequential rule mining (*Das et al. 1998*, *Harms et al. 2002*). Sequential rule mining is concerned more about finding sequential rules from the database that are respecting higher or equal minimum support and confidence specified as a threshold. Minimum support is used to control the minimum number of data cases a rule must cover while minimum confidence is used to control the predictive strength of the rule. Rules that are generated are said to be valid rules in terms of how each user is expected to perform some actions. This means that the goal of the sequential rule mining technique is to intelligently identify which event follows just immediately after another event in order to predict the behaviour of the user in the system. For example, these rules indicate that if event A occurs, event B is expected to occur immediately after A with the specified minimum support and confidence. Minimum support represents a

percentage of transactions from a transaction database that a given rule satisfies. This is taken to be a probability P(X∪Y), where, X ∪ Y, indicate that a transaction contains both X and Y, that is the union of itemsets X and Y. This is to say, from the whole database we check how many times X and Y appears. The formula for computing support is given below:

$$\text{Support } (X \rightarrow Y) = P(X \cup Y)$$

The minimum confidence assesses the degree of certainty of the detected association. This is taken to be a conditionally probability $P(X|Y)$, that is, the probability that a transaction containing X also contain Y. Confidence ensure in finding how many times X appears and followed by Y in the same set of sequence in the database (*Harms et al. 2002*). The formula for computing confidence is given below:

$$\text{Confidence } (X \rightarrow Y) = P(Y|X)$$

Both the minimum support and the minimum confidences are used to prune the search space and limit the number of rules that are generated. If support and confidence are set too low many rules will be generated causing a combinatorial explosion, because frequent itemsets will be associated with one another in all possible ways and many of them are meaningless. The inverse of it is that, if support and confidence is set too high fewer rules will be generated. So observing the behaviour patterns from the user profile that comes from training data and compare it with the testing data is one of the promising ways to identify users who masqueraded in the system using the credentials of the legitimate user to perform malicious activities (*Das et al. 1998, Harms et al. 2002*). Training data is a set of events that have been trained in order to create a user profile while testing data is a sequence of events that are used to test where sequences of events in the

test data set can trigger and fire rules in the profile as a way of judging whether a user is a malicious user or not. (*Das et al. 1998*)

2.8 Summary

In this chapter we have highlighted the basic background concepts including cloud computing, cloud architecture, security issues around cloud, security threats, classification insider threat and sequential rule. We have described the necessity for maintaining the security information within the cloud. Since there are different security threats that exist in the cloud, we focused particularly on the malicious insider. In the following chapter, we critically examine the existing security mechanisms being used to reduce threats from the malicious insiders of the cloud service provider.

Chapter 3

LITERATURE REVIEW

3.1 Introduction

The most challenging task in a cloud computing environment is the provision of relevant security in mitigating insider threat. Fundamentally, cloud computing should have security mechanisms. These mechanisms may include detection mechanisms to detect users who masquerade in the system with the intention of violating confidentiality of information (*Hema et al. 2010*). There are many different detection security mechanisms that have been proposed to address the issue of malicious behaviour of insiders that emanates from the cloud service provider. These security mechanisms employed by the cloud service provider to mitigate malicious insider include activity logging (Shenk et al. 2008), system calls (*Nguyen et al. 2003*), intrusion detection systems (*Lundin et al. 2000, Srivastava et al. 2006*) and data mining techniques (*Agrawal et al. 1996*). This chapter brings to light the outcome of the descriptive problem analysis phase of our research

This chapter presents related works that have been done by other scholars in view of the problem being investigated in this research. Section 3.2 discusses log management tools for discovering behaviour pattern. Section 3.3 presents the system calls technology. Section 3.4 reviews the intrusion detection systems approaches. Section 3.5 present data mining techniques. We then provide the summary of the chapter in Section 3.6.

3.2 Activity Logging

There are many monitoring techniques that have been proposed by many scholars to address the behaviour of an individual within the system. One of the recent solutions that have gained popularity in monitoring the behaviour pattern of the user is log management tools (*Myers et al. 2009*). Log management tools have been recognised as a useful mitigating strategy for event detection and compliance regulation. Most of the organisations consider log tools as the first line of detection that can be used to trace events or actions that have been performed by a user within the system in order to determine the behaviour of the user (*Shenk et al. 2008*).

Employing logs for detecting malicious behaviour provides a better understanding of how a user behaves in the system. Even though using log management tools may seem to be a better solution towards reducing the malicious behaviour of legitimate insider in the system, however, log management tools are associated with many challenges including the cost of implementing the log tool. Employing log management tools without intrusion detection system cannot render a good solution to address insider threat issue. The log may contain lots of information that is not relevant to what a user is interested in, analysing one log file is time consuming.

One of the most techniques that have been used as an input in many Web Usage Mining techniques in addressing insider threat in relation to the masquerader problem is Web Server logs (*Eirinaki et al. 2003*). Web server logs technique, in determining whether a user is legitimate user based on the behaviour patterns, it matches the active user session (or previously stored profiles) to usage patterns of the user. However, web server logs do not provide enough information due to caching mechanisms of web browser and basic principles of HTTP protocols.

Which is a low-level of stateless protocols without clearly defined semantics of performed actions, because of GET and POST do not provide enough information.

3.3 System Calls

System calls are one of the approaches that have been used to build a profile for monitoring the behaviour of an insider within the system with the goal of enhancing the intrusion detection techniques (*Nguyen et al. 2003*). Intrusion detection systems are commonly based on rule for discovering the anomalous behaviour of the user and consider events generated by system based on login. As a result, intrusion detection system is not an effective approach for countering skillfully insider. Nguyen (2003) builds user-oriented models and process-oriented models using file systems and process related systems call exploiting the regularity in the patterns of the file access and process-calling by programs and users.

The approach proposed by Nguyen (2003), present unique way when used to counter malicious behaviour of individuals within the system, as it analyses the system trace performed by the user and lately become a profile that can be used for monitoring purposes. The drawback of this approach mainly focuses on the system call level only and it can be effective for some attacks that are known to the system because the user has to perform operations that form a trace so that behaviour can be analysed based on the last sequence that was made by a legitimate user. Another issue associated with this approach is that false alarm rates are often high because of the many possibilities of system calls made by the user.

3.4 Intrusion Detection Systems

Over the past years, intrusion detection systems were originally designed to identify and mitigate external intruders. Intrusion detection systems are now widely used to combat malicious insider problem (*Cappelli et al 2006*, Wood, 2000, *Scalora et al. 2007, Hofmeyr et al 1998, Sekar et al. 1999*). There are two types of intrusion detection systems, namely misuse detection and anomaly detection.

3.4.1 Misuse Detection

A promising technique that has been used mostly by the current generation of commercial intrusion detection systems is the misuse detection system (*Smaha et al. 1998, Neol et al. 2002*). The approach used in misuse detection in order to discover an intruder is that, it makes decisions based on the comparisons of active user sessions with the rules of attacks previously used by attackers when carrying malicious activities.

One of the advantages that misuse detection can provide is that it can effectively and accurately detect occurrence of known attacks performed by the attacker, while the disadvantage of this approach is that, it can only detect intrusions that follow pre-defined patterns of the user. However, misuse detection technique is not able to detect attacks whose rules are not pre-defined, because it assumes that every operation executed by the user whose rules are not pre-defined operations are normal. As a result, leads to high false positive rates.

3.4.2 Anomaly Detection

In recent years, anomaly detection systems have emerged as a promising technique in detecting malicious insider and against novel attacks (*Neol et al. 2002, Chen et al. 2011*). In discovering the malicious insider, the anomaly detection system bases its decision on the profile of the user normal behaviour. The behaviour of the user, whose session is active, is compared with the user profile representing his or her normal behaviour. As a result, if the deviation is found when comparing user session data and user profile an alarm is raised.

The disadvantage of the anomaly detection approach is that, it cannot reason about an attack in terms of describing what the attack is from malicious actions performed by malicious insiders. Lastly, anomaly detection produces high false positive rates. Other approaches that have been proposed for anomaly detection systems are discussed as follows**:**

3.4.2.1 Predictive Pattern Generation

One of the most promising techniques in addressing the insider problem is a predictive pattern generation (*Teng et al. 1990*). In addressing the insider problem, predictive pattern generation makes use of the axioms of conditional probability in order to predict the future scenarios based on the events that have already occurred. This technique uses a dynamic set of rules for detecting intrusions and it is also highly adaptable to profile changes. Rules that are generated are not static. But the generation of rules is based on the sequential relationship and observed events. The identification of normal patterns of events allows the predictive generation algorithm that is used to infer in terms of specifying that some events are more likely to occur next in the sequence of events than others.

With predictive pattern generation technique, the prediction that event will occur, the algorithm assigns the probability to each mostly likely to occur event as a way to predict which event will follow after the other event. However, the predictive pattern generation technique has two major drawbacks, This approach introduces high false positive rates and false negative rates which are not desirable. Secondly, is that, the effectiveness of the solution depends on the training system using well thought the scenario of abnormal behaviour of which is time consuming to build.

3.4.2.2 Composite Role Based Monitoring

Park (2004) introduced composite role based monitoring technique to alleviate challenges presented by predictive pattern generation (Park et al. 2004). The acceptance of composite role based monitoring has been seen as a promising monitoring approach that is used in encounter malicious insider of the organisation. The composite role base monitor ensures that roles are directly mapped to the domain a user belongs and each role is assigned to a set of expected behaviour in the domain where users belong. The mapping of roles prevents legitimate user from misusing his or her role across different domains within the organisation. The figure below presents the composite role based monitoring architecture. The architecture presents three different role domains, which include organisation role, application role and operating system role.

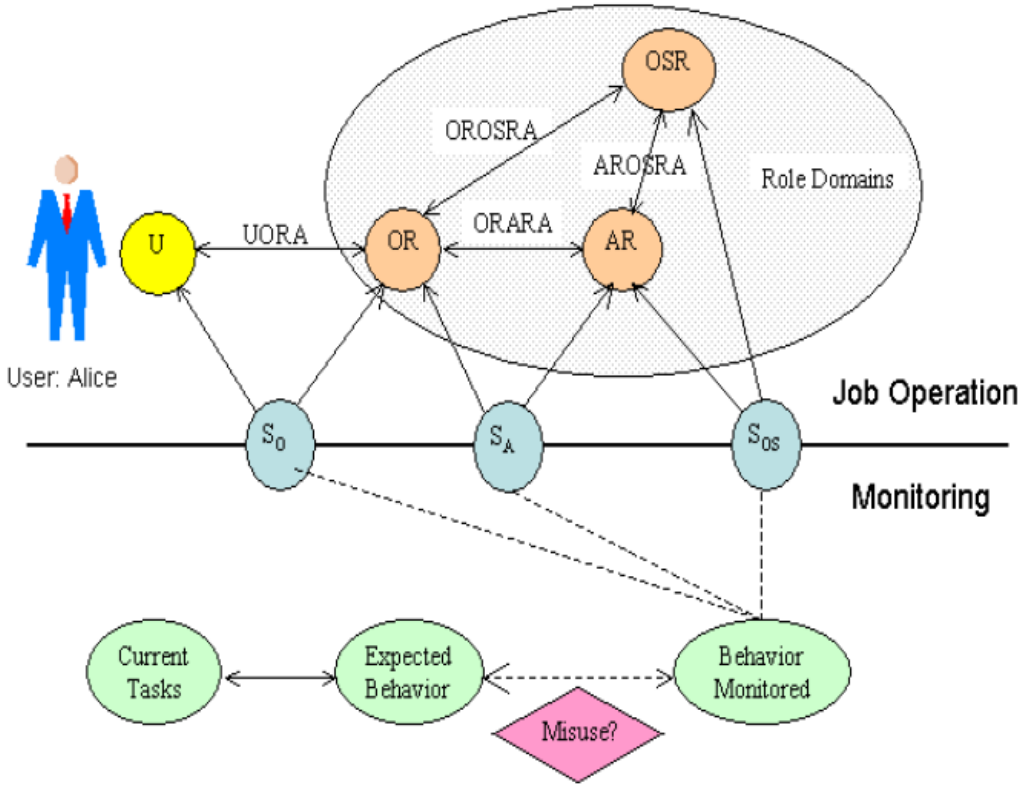


Figure3.1: Composite Role Based Monitoring Architecture (*Park et al. 2004*)

Figure 3.1 shows how monitoring is done in order to predict whether a user is behaving normal or not. The process of monitoring is done by comparing the current task executed by the user against the expected behaviour of the user when using the system. If the expected behaviour defines in a domain where a user belong does not match with the current task, a user is most likely to be malicious insider who wants to launch attacks. If the current task matches the expected behaviour a user is deemed to be a normal user (*Park et al. 2004*). Despite of benefits composite role based monitoring provides in terms countering malicious administrator, assigning user to a role in the domain a user belongs in order to provide adequate monitoring cannot counter unlawful behaviour of the potential masquerader. This is because user behaviour by nature is not fixed since it is a human centric problem and secondly relying on role assignment in a domain a user belong does not provide accurate behaviour pattern since user is monitored on

the role is assigned too. Using role to monitor user does not prove enough evidence because the legitimate user can still perform actions that are in his domain role accidentally. We argue that profiling a user behaviour is a potential solution to address challenges presented in composite role based monitoring technique.

3.4.2.3 Time Series Analysis

In the past years, time series analysis approach has shown positive signs in detecting a malicious insider who attempts to use the credentials of a legitimate user to perform malicious activities (Denning, 1986). Time series was proposed in order to dynamically adapt statistical profiles that keep on changing over time, because profile maybe abused by attackers who want to perform their malicious actions by gradually training the profile and thereby avoiding mechanisms of anomaly detection.

The order and time intervals are more important for observations and predicting user behaviour patterns. Based on the time intervals that are observed, an observation is said to be abnormal if the probability of the occurring event at the specified time interval is too high or too low (*Denning, 1986*). However, the main disadvantage of the time series technique is that, it requires a big amount of computational resources such as CPU and memory usage and time series does not scale well.

Apart from anomaly detection techniques that have been reviewed so far, anomaly detection is categorised into two, namely knowledge based methods and statistical methods. These methods are discussed in details in the following sections:

3.4.2.4 Knowledge-based Methods

Knowledge based methods are one of the approaches that have been used in addressing the problem of intruders (*Neol et al. 2002, Colombe et al. 2004*). The process of detecting a malicious insider in the knowledge based method is done by comparing activities that are performed by the user against predefined rules of the normal behaviour. One of the main goals of the knowledge base approach is the representation of normal behaviour patterns, from which anomalous behaviour is identified as a possible attack.

Knowledge based methods ensures that when particular activities are found in the profile of the user, the system gains the knowledge that the activity performed by the user is not an attack if the representation in the profile matches with the predefined rules (*Neol et al. 2002, Colombe et al. 2004*). Despite the effort of knowledge based approach in detecting the intruder, it has some few pitfalls of obtaining knowledge to prove that user is behaving in a normal way. It is expensive because it is time consuming in terms of crafting rules that represent the normal behaviour.

3.4.2.5 Statistical Methods

Recently statistical methods have emerged as a promising technique in detecting malicious insider and overcome challenges that are introduced by knowledge based methods (*Neol et al. 2002*). In statistical methods profiles of normal behaviour are generated by statistical criteria rather than being hand crafted. Knowledge based methods and statistical methods share the same goal of creating a representation of normal behaviour.

The difference in these two approaches is that, in statistical methods a representation is done automatically as parameters of predefined statistical models. As a result, the automatic generation avoids the more costly handcrafting of profiles observed in knowledge based methods. However, since the profile model is statistical it is prone to error. As a result, we observe the tradeoff in the quality versus cost. Statistical approach has its own disadvantage, profile that are generated are less reliable compared to profile with handcrafting rules in knowledge based methods. Lastly, statistical approach is able to identify normal behaviour not in the profile as a possible attack leading to high false alarm rates. To overcome challenges presented by anomaly detection data mining techniques have also been explored in addressing the challenge of a malicious insider who may intrude using the credentials of the legitimate user (*Schultz et al. 2001*).

3.4.2.6 Sequence Matching Approach

Scholars like Lane and Brodley (1997) developed a sequence matching approach to identify a malicious insider in the system. The key idea behind the sequence matching approach is that, the similarity of new actions performed by legitimate insider is matched against 10 most recent actions in the user profile. The limitation of the sequence matching approach comes from the fact that, if a malicious insider performs three actions that are similar to legitimate user actions in the profile he or she will not be detected as a malicious insider, because such can still be performed by legitimate users. The sequence matching approach in this case is less effective.

3.4.2.7 Host-Based Profiling

David *et al* (1998) used host based profiling as one of the techniques for understanding the intent of some user actions in order to mitigate the insider attack. The basic idea behind this approach is that, once the attacks have occurred in the system, the investigator needs a way to reconstruct the intent of the attacker from audit sources. Even though this process is effective, this is a slow and manual process which cannot be generated to pre-attack analysis. The rules might be crafted to cover known attacks, but sophisticated attackers will always find means to fly under radar. Keeping rules or a profile of the user updated to the latest threat is a significant challenge when using the host based approach. Das *et al.* (1998) and Neol *et al.* (2002) used sequential rule mining as one of the techniques for detecting an attack originating from a malicious insider who has access to the system.

3.5 Data Mining Techniques

Over the past years, data mining has been used to extract implicit, previously unknown and potential useful information from data so that users can use for decision making. Recently, scholars have started looking into the possibility of using data mining in the emerging field of computer security, especially in the challenging problem of intrusion detection in addressing the insider threat problem (*Srikant et al. 1996, Han et al. 2006*). A lot of data mining techniques have been developed, some of which are discussed in the following sections. This section will give a brief overview of some few data mining techniques which are: audit data analysis and

mining, recursive mining, sequential pattern mining and sequential rule mining. Much more emphasis will be put more on sequential rule mining, which is the only data mining technique that will be used in this study.

3.5.1 Audit Data Analysis and Mining

The applicability of data mining approach to anomaly detection includes Audit Data Analysis and Mining (ADAM) (*Valdes et al. 2000*). To discover the attacks from the user behaviour patterns audit data analysis and mining approach uses the combination of association rules mining and classifiers and also build a normal repository of normal frequent itemsets that hold during the attacks. It does that, by mining data that is known and employ a sliding algorithm in order to discover frequent itemsets and compare them with those stored in the normal itemsets repository excluding those that are not normal.

The advantage provided by audit data mining and mining is that, it can learn and discover novel attacks without necessary depending on the training data. However, Audit data analysis and mining approach is less effective in detecting the masquerader problems of the user who has obtained credentials and entered the system, this approach is said to be more effective in countering denial of service attack.

3.5.2 Recursive Mining

Another approach that many scholars have acknowledged in addressing insider threat problem specifically the masquerader is recursive mining (*Szymanski et al. 2004*). In recursive mining approach, it is assumed that the habit of each user in executing a sequence of commands may reflect his identity and thus can be used as a signature. An input in recursive mining approach is first encoded into input symbols and then recursively mined for frequent patterns. Dominant

patterns in recursive mining are encoded with new symbols and the input is rewritten by replacing each dominant pattern with its symbols. Signatures for each user are generated many times until process stop when no dominant patterns in the transformed input could be discovered.

The recursive mining approach uses one-class SVM classifier for the masquerader detection. After the dominant signatures are mined containing attributes of the user, signatures are compared to attributes generated from currently monitored string of the potential masquerader. As a result, if normal and intrusion activities are sufficiently distinct, attribute generated from legitimate user actions will be more similar to user signatures than those generated from the masquerader session. However, the idea of the recursive mining approach seems to be promising but such approach demands mixing user data and as a result may not be ideal or easily implemented.

3.5.3 Sequential Rule Mining

The sequential rule mining technique is one of the techniques that have been used to discover patterns from the sequence of events in order to predict the future behaviour (*Fournie-Viger et al. 2011*). While sequential pattern mining is one of the most popular data mining techniques that are used to discover temporal relation between events in discrete time (*Agrawal et al. 1996*). This data mining technique aims at finding a sequence of events that appear frequently in the database in order to predict the behaviour pattern of the user. However, dealing with the knowledgeable malicious insider knowing that sequence of events appears frequently is not sufficient enough to predict whether a user is deviating from the normal behaviour or not. Because it is possible that a malicious insider can perform event 1 before executing event 5 and at the same time events 5 can appear before event 1, claiming that event 1 is followed by 5 or 5 is

followed by 1 is a pattern of the user could lead in the increase of false negative because of dynamic nature of the user when performing operations in the system (*Agrawal et al. 1996*).

Thus, for prediction purposes, it is desirable from the set of events executed by the insider to indicate how many times event 5 appears before 1 and how many times 1 appeared and 5 did not. Adding to this information, sequential pattern mining cannot be done easily, because it does not provide a good prediction in terms of learning behaviour pattern. The main reason is that sequential pattern mining is concerned more with the list of events that contains several events.

Minnila (1997), Hamilton (2005) and Hsieh (2006), introduced sequential rule mining as an alternative approach. The applicant of sequential rule mining technique has been applied in weather observation and stock analysis to observe and predict the future behaviour analysis. The well-known sequential rule mining technique is that of Minnila proposed (1997), this approach discovers or find rules that are respecting minimum support and confidence.

In sequential rule approach, rules are of the form X→ Y where X and Y are two sets of events, and are interpreted as if events X appears, events Y are most likely to occur with the support and confidence afterward. However, this approach of sequential rule mining can only discover rules in a single sequence of events (*Fournier-Viger et al. 2010*).

Das *et al.* (1998) extended the work done by Minnila by using time series to discover where the left side can have multiple events while the right part still has to contain single event. However, the algorithm used by Das (1998) is highly ineffective because it test all the possible rules without any strategy in place for pruning the search space.

The problem introduced by the Das (1998) algorithm, resulted Harms *et al.* (2002) discovering sequential rules with constraints and time lags in multiple sequences. In this approach, sequential rules are discovered from sequence database and do not restrict the number of events contained in each rule generated. This technique, it searches for rules with support and confidence higher or equal to a user specified threshold to prune rules does not meet support and confidence. In this work, we adopted the sequential rule mining technique in order to predict the behaviour of malicious insider who has the potential to masquerade as a legitimate user, with the aim of observing the behaviour pattern of the user who has stolen the password of the legitimate user in order to carry malicious actions.

3.6 Summary

Monitoring the behaviour of a malicious insider of the cloud service provider is a very critical issue in ensuring the confidentiality, integrity and availability of information. Rocha *et al.* (2011), state that cloud is a new computing paradigm challenged by security threats such as insider threat and is one of the dangerous security threats due to the impact it has on data stored in the cloud environment. What mostly increases chances of administrator to perform their malicious actions is based on the fact that they are familiar with the security controls in place used to detect and prevent threats to information stored in the cloud. As a result, this means the monitoring behaviour pattern of the knowledgeable insider is a growing concern in the cloud environment.

In this chapter, we have presented the state of the art of different monitoring approaches that have been previously proposed in an attempt to address the problem of the insider threat raised in this research. We started by presenting logging techniques and system calls in relation to the insider threat problem. We then presented some intrusion detection system techniques and methods that are relevant to this study. Lastly, we have presented some data mining techniques that can be applied in the monitoring behaviour of the user who masquerader using valid credentials of legitimate users. In the following chapter we present the design of our model that seeks to provide monitoring in order to minimise the insider threat problem.

Chapter 4

THE INSIDER THREAT REDUCTION MODEL

4.1 Introduction

As more and more organisations continue adopting the cloud, there is an increase in the shifting of sensitive information from local premises to the cloud. However, ensuring the security of this information is still a matter of concern among many cloud adopters. The major concern with the data that is outsourced to cloud service provider are the administrators who need to be granted privileged access to various resources to do their jobs. To ensure that cloud administrators do not compromise data security, their activities need to me monitored. This prompted us to address issues of malicious insider who masquerade in the system using credentials of legitimate users. This study provides a way of monitoring user activities when accessing critical information hosted in the cloud.

In Section 4.2, we present the domain specific scenario used in this work. Section 4.3 discusses design requirements, this is followed by Section 4.4 where we present an insider threat model that is driven by the design requirements presented in Section 4.3. In Section 4.5 we present critical success factors. Section 4.6 follows, where we provide the summary of the chapter.

4.2 Domain Specific Usage Scenario

Let consider a medical center which stores all the medical records for its patients in the cloud. Each employee of the service provider has access to all the files related his or her job role and each employee is supposed to access just the documents related to his or her work. One of the

employees of the cloud service provider may gain access to the system by stealing credentials of the legitimate user and perform tasks that impersonate a legitimate user in the system thus compromising the confidential of the sensitive information. If the malicious users leak the celebrity or politician information into the public, since disclosure is a requirement that the organisation must notify individuals of data security incidents involving their personal information. In that case, organisation will be then required to compensate parties that are involved to maintain the trust. The main security concern in this case is that, sensitive data must be protected from being accessed by unauthorised users.

This is due to the fact that such actions are performed by skillful and knowledgeable employee that are familiar with their target and the security measures put in place. Actions that are performed by malicious users can also be performed by legitimate users. This kind of malicious behaviour of employees trying to impersonate as a legitimate user in order to perform operations that are not associated with his or her job role need to be detected to ensure that sensitive data is secure from malicious insiders. The sensitive data stored in the cloud infrastructure is bound to increase over time, because many small businesses may want to leverage in the cloud. As a result, the knowledge and skills of the cloud administrator increases over time, monitoring the behaviour pattern of the privilege users have become one of the essential aspects to ensure that data stored on the cloud is secured from being accessed by unauthorised users. Based on the scenario presented above we then bring out security requirements that form the bases of our model.

4.3 Design Requirements

From the scenario presented in Section 4.2 we have identified the problem that is facing cloud service provider, and then we provide some requirements that need to be taken into consideration when designing a security mechanism to alleviate the problem of users that pretend to be normal users in the system. Such requirements include management of user identities, monitoring and event logging. The study however, focuses more on monitoring.

Monitoring behaviour patterns helps in identifying malicious users. This is done through examining the sets of actions performed by the user and comparing the generated behaviour pattern with the historical behaviour patterns. Comparing the historical behaviour helps in identifying malicious users that used the credentials of legitimate user in order to access the system, it must be noted that we cannot neglect the fact that legitimate users who have been given access to the system can be malicious at the same time by deviating from their normal behaviour

Monitoring malicious users who are accessing sensitive data stored in the cloud service provider side is one of the important aspects to ensure that data remain secure from unauthorised users, who might want to use data for malicious purposes by masquerading in the system as a normal user. Hence, there is a pressing need for security mechanism to ensure that user data stored in the cloud remain secured anytime. Enforcing security policies at cloud service provider become a requirement.

In chapter 3 we have reviewed different approaches used to address the insider threat problem. This section, presents the design criteria taken into consideration when designing the insider

threat reduction model for minimising attack originating from malicious insiders of the cloud service provider. The design criteria are as follows:

i. **Management of user identities-** Cloud providers want to ensure that all users who want to access the system are authenticated and authorised users in order to limit access of unauthorised users to sensitive data stored in the cloud. This ensures that administrators of a cloud service provider possess valid credentials and have access to appropriate cloud resources. Since different administrators fall in different domains and require different roles to perform tasks in the system. We do not cluster many users in one role but we decentralised the process of role assignment by mapping one role to a specific account for a user. In order to ensure the segregation of duties and also eliminates the challenge of different administrators associated with one job at the time that result in the escalation of privilege, thereby ensuring that different privileged administrators have access to only one cloud resource that are required by their job roles. As part of managing user identities, roles and privileges should always change to match the current job profile at any given point in time of administrator.

ii. **Monitoring-** Cloud providers require continuous monitoring of the behaviour of the insider, on how data is used once it is accessed. Monitoring should ensure that malicious insider who masquerades in the system using the credentials of the legitimate insider are detected through analysing behaviour patterns. This will ensure that potential malicious insiders are identified on their behaviour patterns when using the system, because behaviour cannot be stolen from the normal user and it is most likely that a masquerader will not be consistent when performing some actions in the system. Profiling behaviour patterns of the user in order to learn how each user is

behaving in the system then become a necessity in detecting malicious user with the intention of performing malicious actions in the system.

iii. **Event logging-** Logging events of the cloud administrators help in collectively providing documented evidence of actions performed by the administrator and behaviour of a single administrator within the system. Audit trails and log ensure that each user is accountable for a malicious action. Securing event logs are one of the growing concerns since unauthorised users might edit log to cover up their actions.

The above mentioned design requirements helped us in designing the security model for reducing threats originating from insiders of the cloud service provider. This reduction model also entails how design requirements are achieved. The model ensures that users who masquerade in the system are detected based on their behaviour patterns when accessing the system.

4.4 Insider Threat Reduction model

To bridge the gap that causes business and individuals to be reluctant about adopting the cloud when sensitive data is migrated into the cloud environment, there was a necessity to develop a model that reduces attacks that originate from the legitimate user of the cloud service provider. The development of the insider threat reduction model was driven by the design requirements mentioned above. We have firstly analyse a typical scenario which was used as a method for collecting or elicitation of functional requirement when developing a security system (*Basescu et al. 2011*). Describing a domain specific usage scenario helped in providing a clear

understanding about the domain that was suited for requirements analysis for the design of our model.

The domain specific usage scenario considered in Section 4.2 had been used to illustrate any behaviour or actions performed by the users of the cloud service provider. Based on the usage scenario provided above, we concluded in our research by inheriting some of the advantages provided by composite rule based monitoring. Composite role based monitoring implements least privilege and separation of duty that make it the most widely accepted technology (*Salem et al. 2008*). Before dwelling much on the proposed model, we outlined the advantages of the preferred approach taken by this research work as a building block of our model. Composite role based monitoring implements least privilege access to control user who are managing internal resources and ensure that permits are granted to role based group instead of user account assigned to the user. Composite role based monitoring ensure that users are separated from unneeded permission.

It also makes users of the cloud service provider to think about minimising access needed for each user within the organisation to perform his or her task as its implement segregation of duties for different user that fall in different domains. We used the concept implemented in the composite role based monitoring, since it is a widely accepted technology in the distributed system composite role based monitoring was seen as a promising technique to address the problem of insider threat based on the advantages that were provided by this solution, which motivated us to consider it when developing our insider threat reduction model. We were not benchmarking with composite role based monitoring technique but we inherited advantages that were provided by this monitoring technique in our solution approach.

From the theoretical point of view assigning user in the role based group instead of user account might be a viable way when mitigating insider threat posed by legitimate users of the organisation. But taking that into practice may create a lot of problems when trying to monitor the behaviour of each user when accessing the system. Since a knowledgeable user may violate security policy and start performing malicious actions that compromise integrity and confidentiality of sensitive data stored in the cloud, knowing that it is difficult to trace who perform what since many users are clustered in one role based group. We examined the composite role based monitoring and pointed out the weakness of providing adequate monitoring when users had been given access to the system. Sequential pattern mining technique was then used in this work to predict the behaviour pattern of the user when using the system. The theory behind this data mining technique in predicting the behaviour of the user was that, it was able to intelligently tell or learn that if events "ABC" occurred event "F" was mostly likely to occur afterward with the specified minimum support and minimum confidence. This theory was then applied in the research in predicting the behaviour patterns of the user.

In Figure 4.1, we depict our insider threat reduction model comprised of six main components. It was believed that, taking into consideration the management of a user identities, monitoring and log events during the crafting of our model ensured an appropriate solution for reducing the insider threat originating from the cloud service provider side. We believed that this would ultimately enhance the uptake of the cloud. As indicated in Section 1.3 that, how could a solution to minimize insider threat in the cloud environment be crafted. To address this concern, we proposed the insider threat model that initially enforced policy in the **Policy base** component. We assumed that insider threat could not be effectively minimised by technical controls only but the combination of technical and non-technical yields a better approach to reduce insider threat.

The policy base enforces non-technical controls that are enforced before and after users have access to the system. The formulation of the model is crafted from the design criteria. **Management of user identity component** is responsible for ensuring that an administrator of cloud service provider that poses valid credentials has access to cloud resources. Since different administrators fall in different domains and require different roles in performing tasks in the system that necessitate the model that can provide continuous monitoring in order to detect a malicious insider. We did not cluster many users in one role but we decentralized the process of role assignment by mapping one role to a specific account for a user. Management of user identity eliminated the challenge of different administrators associated with one job at the time that result in the escalation of privilege, thereby ensuring that different privileged administrators had access to only one cloud resource that was required by their job roles. Management of user identity ensured that operation was done by authorised user that comply with the rules and regulations specified in the role of the user.

Our approach needed to support monitoring in order to accurately predict whether a user was a malicious user. **Monitoring component** act as a main component, that is responsible for providing continuous monitoring in order to detect user who masquerade in the system with the aim of violating confidentiality and integrity of information. The details of how monitoring is achieved in order to differentiate between normal users and malicious user is provided under the description of components. **Cloud resource** is a service that is accessed by insiders in order to perform daily operations in managing sensitive data. The database is responsible of storing sensitive data belonging to cloud customer and protect data from being accessed by a malicious insider that might cause inappropriate disclosure or misuse.

One of the main benefits our model delivers is the use of sequential patterns in detecting masqueraders that use valid credentials of a legitimate user in order to perform their malicious actions. The order in which actions are executed is more important in providing how each user is expected to behave in order to reduce chances of a malicious user performing malicious actions.

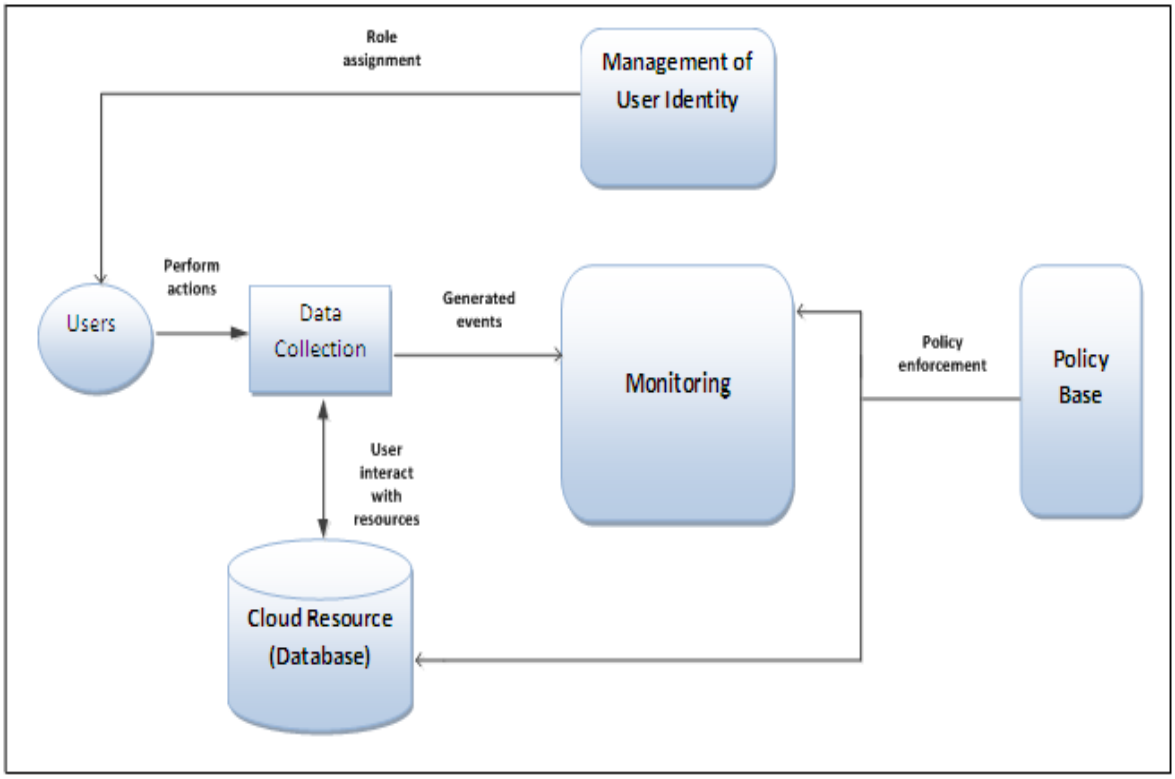


Figure 4.1: Insider Threat Reduction Model for Cloud Environment.

From Figure 4.1 we can see that the monitoring component is presented at a high-level without any details on how it works in order to achieve the desired goal of detecting a user who masquerade in the system using valid credentials of the legitimate user.

4.4.1 Description of Components

4.4.1.1 Policy Base Component

Policy base component deals with non-technical solutions in order to minimizing the attack originate from a malicious insider. Since employing technical controls only without enforcing policies before users access the system cannot effectively reduce insider attack. The significance of this component in our model is to ensure that procedures and policies take into consideration during the process of designing our solution. Automating decision policies that ensure that once a user leave the organisation his or her access should be terminated from the physical system and also in the remote login. To reduce chances of malicious activity being performed by a user who is no longer part of the organisation. Enforcing policy, critically reduce the insider threat on the basis that the user knows what is expected of him or her when using the system. The integration of policy based components into monitoring and cloud resource ensure proper usage of the system. The use of policy base help in controlling access to cloud resources as a way to reduce an insider threat, because the insider threat cannot be successfully reduced by technical controls only and policy base plays a vital role in minimising insider attacks

4.4.1.2 Management of User Identities Component

This component encompasses some of the non-technical solutions that have to be in place to ensure that the right employees are hired in the first place. The cloud HR departments should conduct pre-employment background checks of all new administrators to get more information on each candidate to make informed decisions (*Randazzo et al. 2005*)**.** This might include checking credit reports, criminal records and school medical reports. The ISO27001 standard emphases that, legal department must craft policies and procedures to government access and the use of IT resources. Once the right candidate has been hired, they have to be assigned to the

proper roles and privileges. Health Insurance Portability and Accountability Act (HIPAA) standard, the management of user identity must be done in both pre-admission and post-admission to avoid user violating sensitive information that belong to patients (*Dwyer et al. 2004*).

This means that privileges should be terminated when the administrator leaves the company. The literature has shown most insider attacks are performed by disgruntled employees or employees who have resigned and are currently serving their notice (*Randazzo et al. 2005*)**.** The HR department should work closely with the IT department to ensure that user identities are monitored closely under these circumstances and take appropriate action. There should be proper policies and procedures to govern this.

Having many users assigned to one role has made it difficult to determine a malicious user among other users who are assigned in the same role under the same domain. Benefits that are provided in the role based access control model are integrated in this component. Since we are not benchmarking with any solution, but we inherit advantages that are provided by role based access control. We enforce separation of duties and assign one user to a specific role at a time that helps in increasing the protection of internal resources. The role that is assigned to the user then becomes the integral part of the user profile.

4.4.1.3 Monitoring Component

The monitoring act as a core component in our model, we assumed that users after being assigned to proper roles, a malicious insider could still masquerader by using other users credential to perform malicious activities that result in violating user confidential data. In this component, we provided more details on rule learning and pattern matching sub-components

because it's where the contributions of this research was found. From the monitoring component we developed two main algorithms namely rule learning algorithm and pattern matching algorithm. The rule learning algorithm learns how each user performs actions in the system in order to come up with the user profile and pattern matching algorithms match the user profile with the set of single sequences from the test data. We employed a constant monitoring of user events that are performed by the user to ensure that user pattern was consistent with the one stored in the user profile by using sequential rule mining techniques. The way each user behaved was different compared to other users when using the system, that helped us to accurately predict based on the pattern that is generated from a profile to dynamically judge whether a user was a legitimate user or not. This component was implemented for proof of concept in order to detect potential masquerader who wanted to violate the integrity and confidentiality of information using the credentials of the normal user.

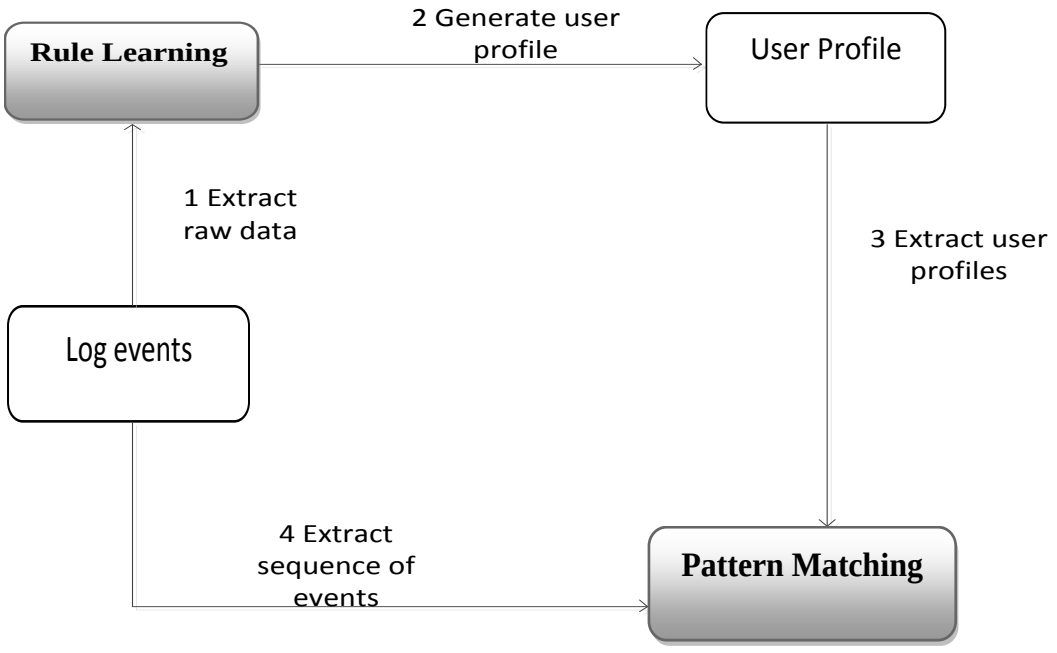


Figure 4.2: Monitoring Component for Insider Threat Reduction Model.

Figure 4.2 shows the monitoring component in details on how each user is monitored when accessing the system. Input in our monitoring component was a set sequence of events generated when the user was performing actions in the system. The generated sequence of events were then logged as raw data and extracted into rule learning algorithm. The output from learning algorithm was used to build the profile of the user and from the profile we could learn the sequence of patterns that identified each user when using the system. The monitoring component has sub components and is discussed as follows:

4.4.1.4 Event Logging Component

The **Log event** component was responsible for capturing and storing all sequences of events that were performed by the user when accessing the system. After sequential rules that were mined the user profile was then generated.

4.4.1.5 Rule Learning Algorithm

The rule learning component was used to learn from the raw data how each user was executing tasks in the system in order to generate sequential rules that eventually form a user profile that uniquely identify each user in the system. The learning process helped in characterising the behaviour patterns of each user in the system. The rule learning component was used to learn from the raw data how each user was executing tasks in the system in order to generate sequential rules that eventually formed a user profile that uniquely identify each user in the system. The learning process helps in characterising the behaviour patterns of each user in the system. In order to generate user profile, we developed a rule learning algorithm that showed the processes involved in order to create a user profile. Figure 4.3 presents the rule learning algorithm.

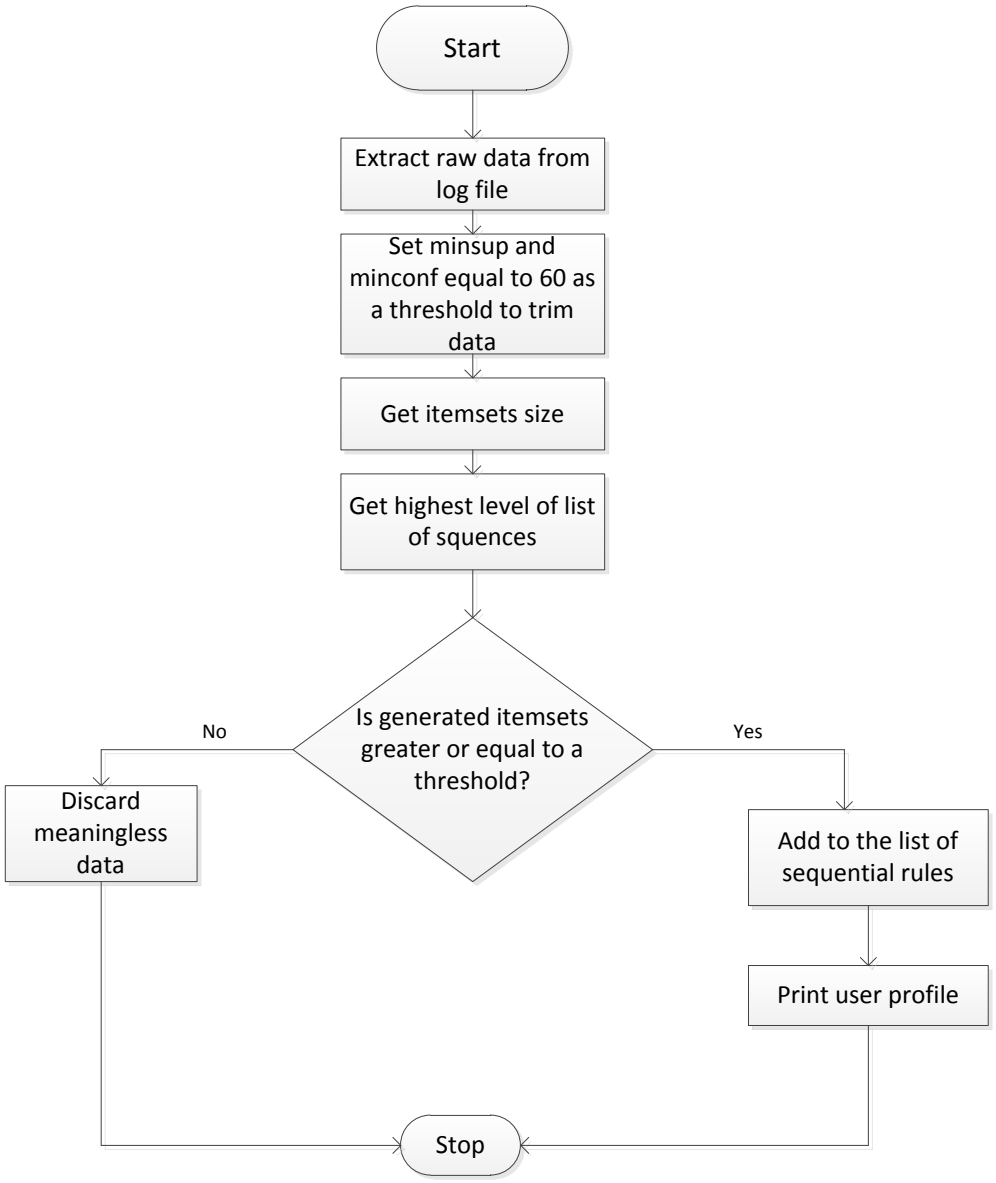


Figure 4.3: Rule Learning Algorithm

The first step of rule learning algorithm was that, a raw data was extracted from log file for learning and it created itemsets from raw data that form a sequence. Itemsets that eventually form a sequence were in a form of [2, 4, 6, 9, 2]. We set minimum support (minsup) and minimum confidence (minconf) as a user specified threshold in order to trim the frequent

patterns. After minimum support and minimum confidence was set, we got the itemset size and the highest level of list of sequences. From the sequence, a list of sequences was created and each list of sequences storing a different sequence of events indicating how each user was executing a set of action in the system. As a way to learning the interesting sequential rules in order to generate user profile, the algorithm checked whether generated itemsets met the specified threshold in order to trim the frequent patterns. Every data from the sequence of events that was learned and said to meet the specified threshold was added to the list and form part of rule generation and the data that did not met specified threshold is said to be uninteresting and was discarded because it did not form part of rule generation. For more information see appendix A for reference purpose.

4.4.1.6 User Profile

A user profile was created after the learning process had been performed and it was used to represent the user identity within the system. Every user needed to have a user profile that could be used to differentiate the behaviour patterns of each user from other user within the organization. To ensure that the behaviour of the user could be judged against the behaviour patterns that identify a user when accessing the system.

4.4.1.7 Pattern Matching Algorithm

The pattern matching component was used to identify malicious users by comparing the current generated sequence of events from the testing data with the user profile to find out whether the user was a malicious user. In the process of identifying a malicious user, we developed a pattern matching algorithm. The pattern matching algorithm predicted whether a user was malicious or not based on the sequence of events in the testing data whether their trigger and fired rules in the user profile. If the behaviour pattern stored in the user profile was not the same with the current sequence of events there was a probability that a user was a malicious user or else if the behaviour pattern was consistent with the one stored in the user profile, we assumed that the user was a normal user. Figure 4.4 present pattern matching algorithm and steps that were involve during the process of identifying whether a user was malicious or not.

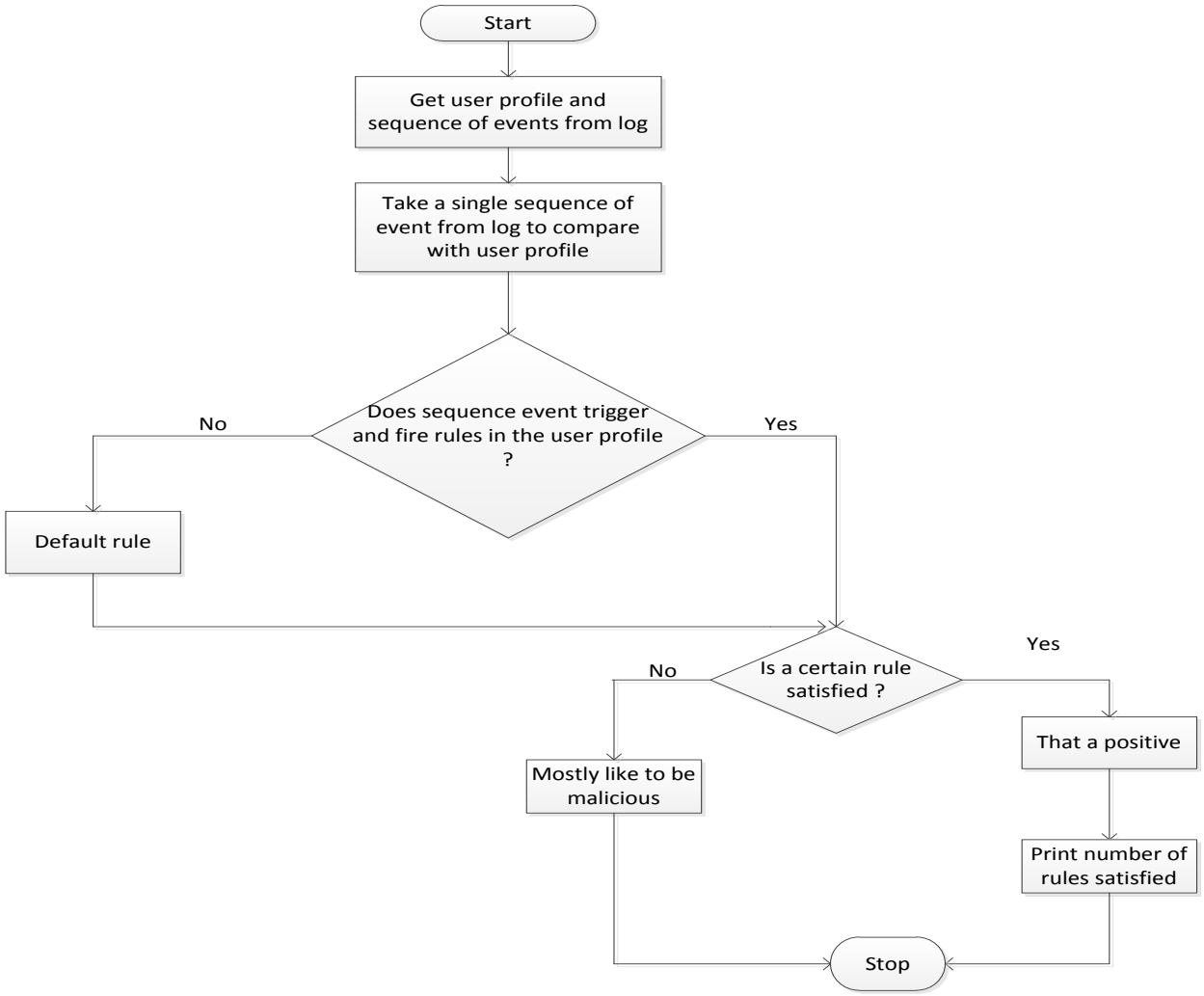


Figure 4.4: Pattern Matching Algorithm

The first step of a pattern matching algorithm both user profile and sequence of events served as an input to the algorithm. The algorithm read the single sequence from the begging to the end and noted itemsets and compared the first inputted single sequence with the user profile. The algorithm then looped over inside the user profile to ensure every sequential rule in the profile is compared with the single sequence. The algorithm checked whether the single sequence triggers and fired the rules, if the single sequence did not fire any rule that a default rule. If the sequence of events in the single sequence that had been inputted triggers and fires rules in the profile the

user was a normal user. The algorithm then checks whether a certain rule is satisfied, if rule in the profile is not satisfied that is most likely to be a malicious user. If the rule is satisfied that is a positive and the algorithm out all the number of positives. Appendix B can be used for reference purpose for the detail algorithm.

4.5 Critical Success Factors for Minimizing Insider Threat

In order to ensure that the model presented above is more effective in dealing with the malicious insider, some of the critical success factors that must be taken into consideration are presented as follows:

- A security awareness training program that shows the proper usage of the system to new employees of the cloud service provider on what is acceptable and what is not when using the system in order to effectively reduce the chance of threat from occurring within the organisation that could result in the greater harm in the system. Training awareness should educate new users about creating strong password usage and not allowing two different users sharing the same password in order to reduce ineffective of user password.

- The other aspect that needs to be considered before new insiders or employees have access to the system is to perform some background check and screening. This helps in providing a historical behaviour of each user because the cloud service provider may run a risk of employing a user with a criminal record. For users who have bad record can cause breaches to sensitive data stored in the cloud. As a result, that could affect the reputation of the cloud service provider and decrease the level of trust of the service provider to their customers.

- Enforcing policy such as due diligence is one of the critical success factors that can be used to reduce threat caused by authorised user in the system. This policy helps in providing guideline on how a user must behave in the system and also collect user actions or activities that are performed by the user that can be used later for auditing purposes in order to make sure that each user is accountable for his or her actions.

- Revoking access from physical systems and remote access after a user has left the organisation should be considered as another set of policy to be enforced in order to reduce insider threat. Because many security data breaches are initiated by users who used to have access to some internal resources within the organisation and different password known to the user must be changed to ensure integrity and confidentiality of information. If the user no longer need passwords in order to perform certain tasks in the system, those credentials must be deactivated to ensure that users can no longer use in order to access the system.

- Sensitive data stored in the storage media in the cloud service provider must be encrypted to ensure confidentiality of information, because a malicious insider can leak sensitive data if it is stored in the plaintext by taking it off premise. As a result, encrypting data at rest and in transit is another critical success factor to reduce the insider threat.

We have outlined five critical success factors that need to be considered when dealing with insider threat in the cloud environment. In this research we are not claiming that the above

mentioned critical success factor completely mitigate insider threat originating from privilege users of the cloud service provider, but we are saying that if these critical success factors are taken into consideration insider threat can be reduced effectively in the cloud environment.

4.6 Summary

In this chapter, we have described the design of our model for reducing insider threat in the cloud environment that originated from users who could masquerade using valid credentials of the legitimate user. From the design of our model we successfully bridged the gap to the solution approach adopted in the work, by ensuring that user behaviour pattern was closely monitored based on the user behaviour patterns to ensure that insider threat reduction model enabled better monitoring. As stated in the previous sections, the goal of this research was to develop an insider threat reduction model to minimise attacks originating from legitimate user of the cloud service provider by monitoring behaviour patterns. This has been achieved in this chapter with detailed description of model design, algorithms, and component interaction in this research.

Chapter 5

MODEL IMPLEMENTATION

5.1 Introduction

To prove the concept being discussed in this research, this chapter presents the implementation of the model. We also use UML diagram to illustrate the design of the model which include the use case diagram and sequence diagram. This chapter is structured as follows: in Section 5.2 present the assumption that are considered during the implementation process, in Section 5.3 then present the implementation design. In Section 5.4 provide implementation details. Section 5.5 summarises the chapter.

5.2 Implementation Assumptions

The following assumptions were made during the implementation of our solution approach.

1. In order to effectively monitor malicious insiders all event(s) performed by the user are analysed after a user has completed all tasks, we neglect the aspect of analysing event(s) in real time.
2. Users perform predefined tasks associated with their role throughout when using the system.
3. It is fine for a user to deviate slightly from usual behaviour because of new circumstances as long as their deviations are within set thresholds.

Based on the assumptions that were made during the implementation process, in the following section we then present the prototype design of the insider threat reduction model.

5.3 Implementation Design

This section present the prototype design of our model using UML models. We start by providing illustration using a use case diagram, followed by an activity diagram that helps in clearly painting the picture about the prototype design.

5.3.1 Use Case Modeling

In the design of the implementation we first present a use case which is selected as a mechanism for evaluating the prototype against the design criteria presented in Section 4.3 and while considering assumptions that were made in Section 5.2. Figure 5.1 depicts the use case diagram and its actors, which include an information web service that verifies credentials, web security service and requester.

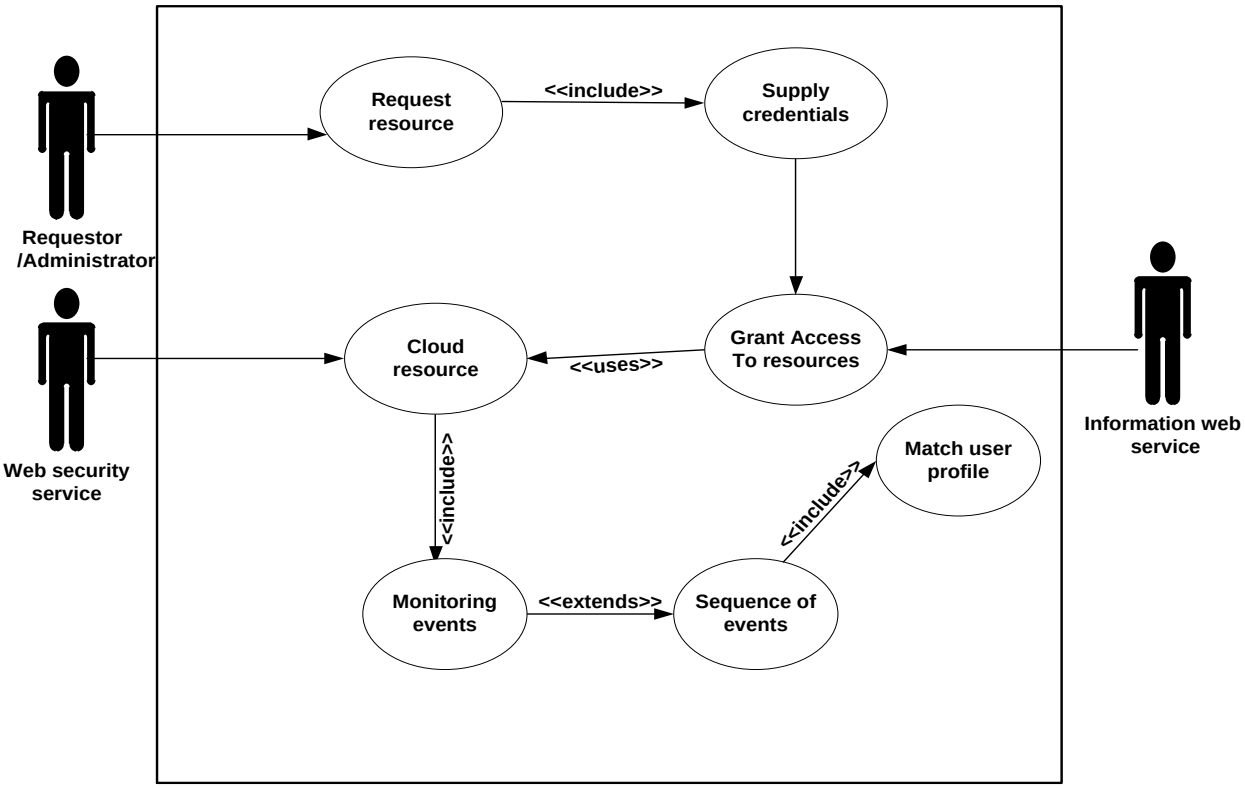


Figure 5.1: Use case diagram for monitoring component

The above use case diagram illustrate the flow of information between different components of the system. In this case, if a requester is requesting to access protected resource that is valuable assets, the requester supplies the valid credentials that are already known to service provider because the requester is a legitimate user. Information web service is responsible verifying the supplied credentials before access is given to the requester. Once authenticated the requester is given access to the protected cloud resource. The web security service then monitors all events performed by the requester in the cloud resources and log those events performed by the requester.

i. Information web service

The information web service acts as a filter that ensures that only known users that possess valid credentials will have access to the resources. During the process of verification, the information web service sends supplied credentials as a SOAP message to check the existence of credentials in the database together with the role assigned to the user. If the credentials match, the user is then given permission to perform operations in the system if not the user is denied access.

ii. Web security service

Web security service provides the monitoring of all events performed by the requester when accessing the system. The Web security service also communicates with the logging event component that stored all events performed by the user.

iii. Cloud resource

Cloud resource is a service that is being utilised by insiders in order to perform their daily business processes. All events that are performed by users are stored in the database and extracted to pattern matching algorithm and rule learning for reasoning about the behaviour pattern of the user.

5.3.2 Sequence Diagram for Insider Threat Reduction Model

The sequence diagram in Figure 5.3.2 illustrates how the messages flow from one component to another. The diagram shows the vital role played by each component to achieve a smooth monitoring of user action when interacting with the system. This sequence diagram provides an understanding when a user wants to access resources by supplying credentials in order to execute a certain task.

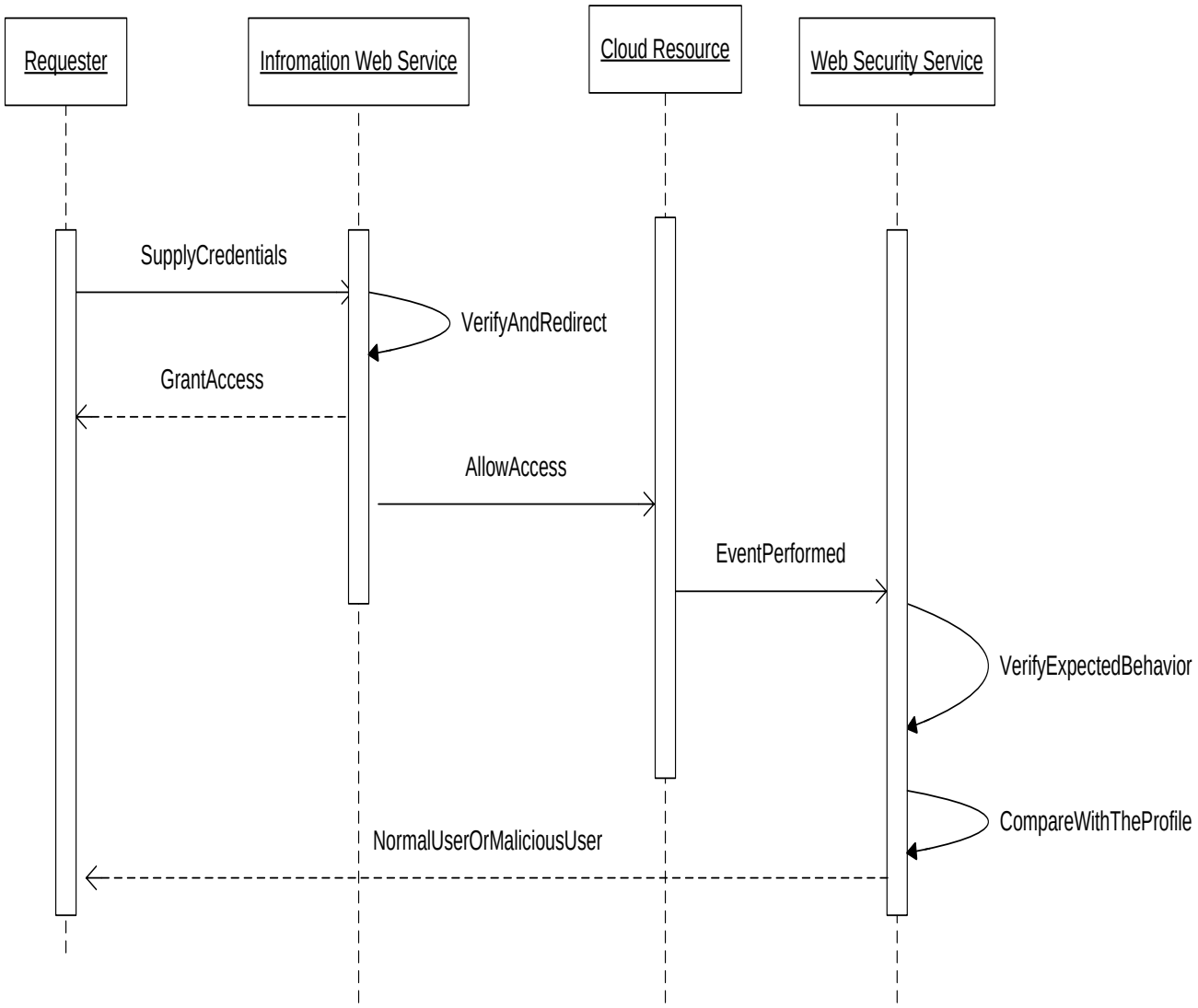


Figure 5.2: Sequence diagram for the insider threat reduction model

Figure 5.2 illustrates how the message flow from a requester who wants to execute different tasks on the system that a most like to be associated with his or her job role by supplying valid credentials that are known to the system. After the credentials are provided by the requester. As a result, we can be able to recognise that it is the information web service that does the verification of the existence of the credentials supplied by the user.

If the credentials are valid the user is given access to perform permitted operations in the system. The web security service does the monitoring by storing events that are performed by the user in the sequence in which events occurred in order to determine the behaviour of the user. Those events match with the profile of the user in order to reason about the behaviour of a user. The predefined threshold was used to judge whether a user was malicious user or not.

5.4 Implementation Details

For the solution implemented in this research work to be suitable for cloud environment, OpenStack cloud environment was used. OpenStack cloud is an open source cloud that allows users to build Amazon Web Services (AWS) and deploy their applications. OpenStack was selected as a cloud environment to be used on the basis that, it support features of Role-Based Access Control and support many new technologies by allowing user to develop their applications and plug-in into OpenStack cloud. This type of cloud provides some benefit such as controllability, flexibility, scalability, openness and compatibility with Amazon Web Services such as Amazon EC2 and Amazon S3. OpenStack has three major components which are: image service, storage and compute. Compute component in the cloud is responsible for provisioning and managing large networks for virtual machines (VM), whereas object storage component is responsible for storing large amount of data and is in support of block storage. Lastly, the image service component does the registration of virtual disk images. The conceptual architecture of OpenStack cloud is shown in Figure 5.3:

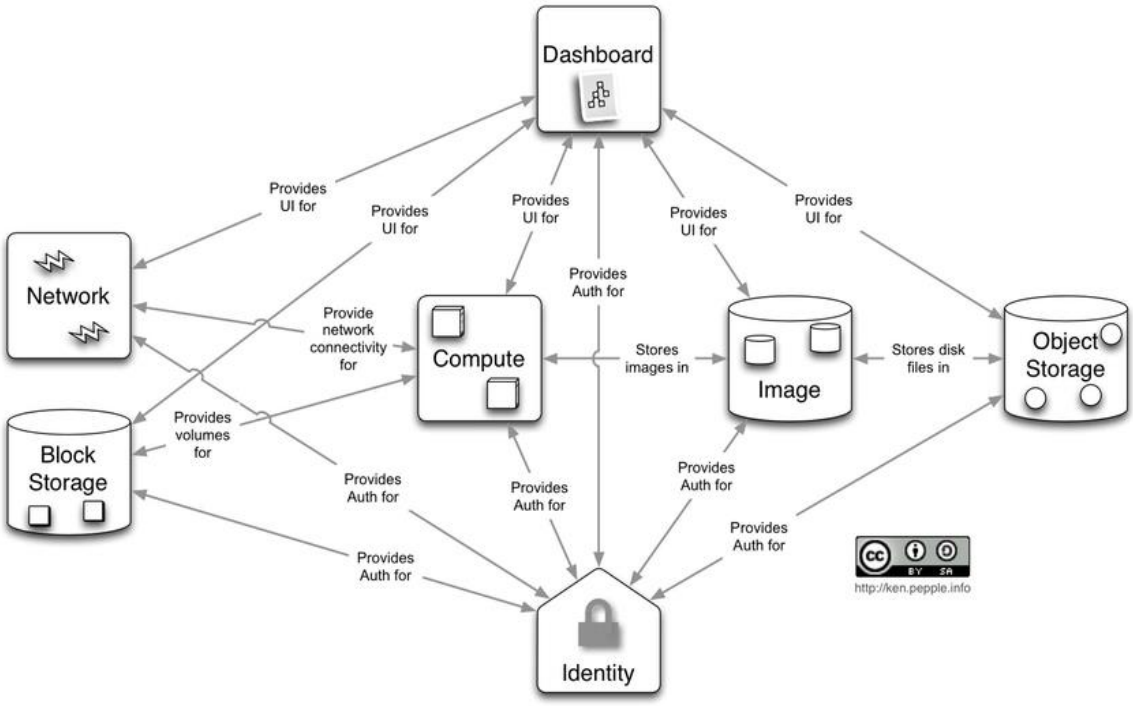


Figure 5.3: Conceptual architecture of the OpenStack cloud (OpenStack, 2011).

Figure 5.3 illustrates the different components of the cloud. What must be noted is that, for the purposes of our demonstration, The test our implementation was conducted on the Software as a Service (SaaS) layer of our cloud. Therefore, Platform as a Service (PaaS) and Infrastructure as a Service are out of the scope of this work (*Nkosi et al. 2013*). As part of this application, there was also a service to monitor and log all the user events for the various activities in a session. In order to ensure that employees of cloud service provider who were managing the data stored in storage component were behaving in the normal way when performing different operations in the data there was a need to monitor user actions. Because of the human-centric problem facing cloud service provider, there was a necessity to protect the integrity of data hosted in the cloud. The dashboard is a front-end of the cloud or web-based interface that is exposed to users in order to interact with the system. Identity service provides roles that are attached to the user with specific operations that a user can perform when managing data stored in the cloud. See appendix B for cloud setup.

The approach of using Web Services technology was also considered during the implementation of our solution and later we deployed our solution into the cloud. Java was used as a programming language and Netbeans 7.1.1 was used as a development environment while Glassfish as an application server.

The approach that was considered in achieving the desired goal was to consider monitoring user behaviour patterns. The process of monitoring user actions performed in order to find a sequence of events that appear frequently with the given minimum support and confident. We considered using a real user that acted as a requester with our dashboard in order to generate the sequence of events that represent user actions in the system. The sequence of events enabled us to differentiate between users who were deviating from normal behaviour and malicious users.

In order to perform the mining of sequence of events, we used Sequential Pattern Mining Framework (SPMF) _ (*Fournier-Viger, 2008*). Because is an open-source data mining platform written in Java and it offers the implementation of 50 data mining algorithms. It also provides the flexibility that any source code of each algorithm can be integrated in any Java platform. Moreover, SPMF is a well-developed API for mining the data and analyze the behaviour pattern of the user in a distributed environment.

Since our goal was to find sequences of events that form a rule from user actions in order to create a user pattern. We implemented rule learning algorithm in order to cater for our desired goal to get a sequence of events that were interesting to a user, based on the predefined minimum support and confidence specified by the user as a threshold. All sequences of events performed by the user was kept in the database component.

5.4.1 Environment Setup

To test the application that was developed, Three machines were used in order to allow cloud administrators to interact with cloud with the purpose of learning behaviour pattern of each administrator. The latest version of Cloud OpenStackm, Folsom, was deployed on the virtual machines that support Virtualisation technology to power up virtual machines. Two machines have *i*5 gen processor and Virtualisation technology enabled with 8 GB of RAM DDR3 and other machine was an *i*7processor with a processing speed of 2.10 GHz and 8GB of RAM machine. We needed to perform all the necessary configurations and installation before the cloud was deployed on the virtual machines to ensure proper running of the cloud. To show and give a better understanding of the cloud after had been deployed on the virtual machines. Figure 5.4**:** illustrates the overview of the installation.

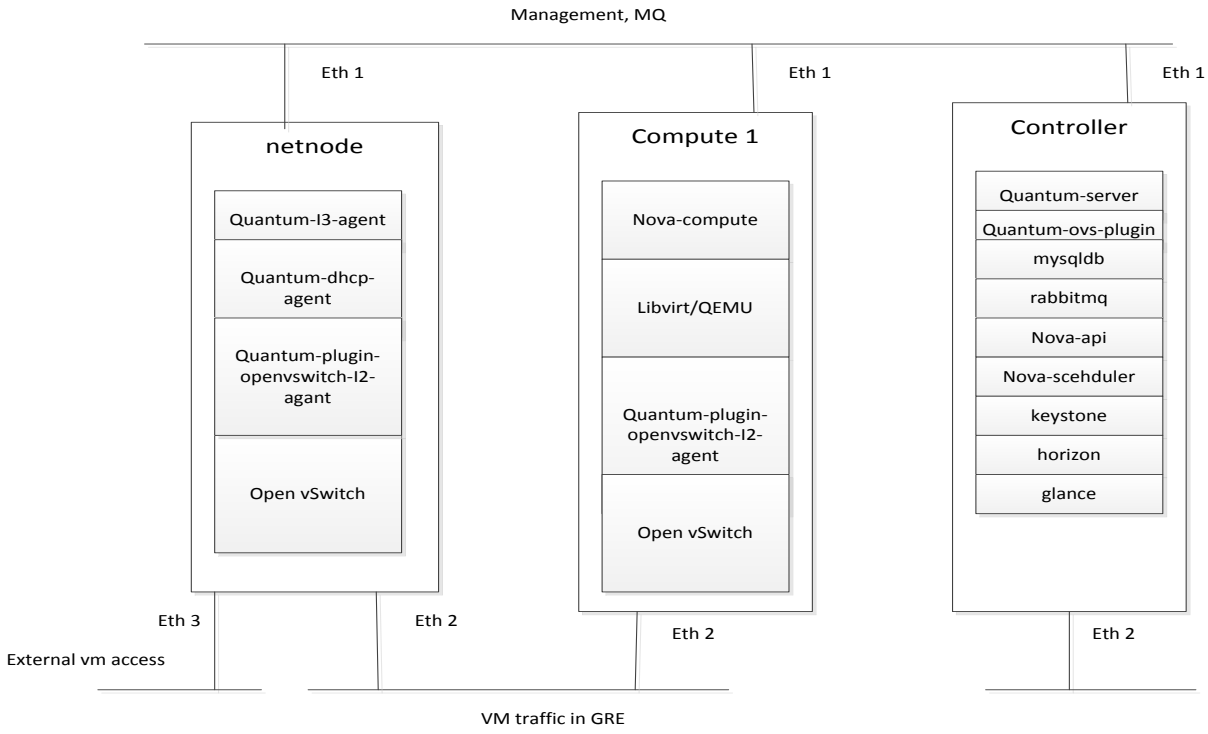


Figure 5.4: Overview of the installation (OpenStack, 2011).

Figure 5.4 illustrates the overview installation that was carried out in the setup of Folsom cloud with different project that were used for smooth running of the cloud. Starting from setting up network to allow cloud administrator to access cloud that had been deployed in three machines. The host machine was only used since it provides internet connection between the host machine and virtual machines The first virtual machine acted as a controller to run the following components to ensure the smooth running of the cloud. This components includes Nova which is responsible for providing resources, while Keystone provides the central authentication mechanisms. The Glance is responsible for delivering ready to use images. The quantum-server runs the detailed virtual network. Horizon provides a web based interface that is used for configuring and managing the virtual machine from the web interface. Cinder is responsible for providing permanent data storage for instances. The Mysql database stores sequences of events that are generated after the administrator has finished interacting with the cloud while Glance shows instances that are running. The RabbitMQ ensures the reliability of the message that is sent from one component of the cloud to the other. The second virtual machine or node in the setup of the cloud provides the computing resources. The first virtual machine which is a controller coordinates the node actions. As a result, node require nova-compute which manage instances and Quantum, the open vSwitch that runs virtual machine.

In the process of setting up, the controller uses network interface 1 to manage both virtual machines. The interfaces were configured to communicate through the same local area network, while network interface 2 is used to handle traffic between two virtual machines and was also dedicated for instances. Network interface 3 was used by Quantum for routing the traffic from instances to the internet, making them publicly reachable. In all three machines we used Web service information in order to perform verification of user credentials when trying to access the

system. The Web security service was developed to monitor events that were performed by the user on the virtual machines.. After each administrator had completed the task he or she was executing in the system, the Web security service logged all those events. Every event performed by the administrator was then sent to the server. We then employed rule learning algorithm to generate sequential rules and used matching algorithm determine the user behaviour by comparing the user profile with the first training dataset user profile. The interface that was exposed to allow users to interact with the system is presented below:

Figure 5.5: Login interface

Figure 5.5 depicts the login page, where all users of cloud service provider are prompted to provide valid credentials that are known to the service provider. If the credentials provided by the user are valid, the second interface in Figure 5.6 shows up, where a user can now perform operations that related to his job role.

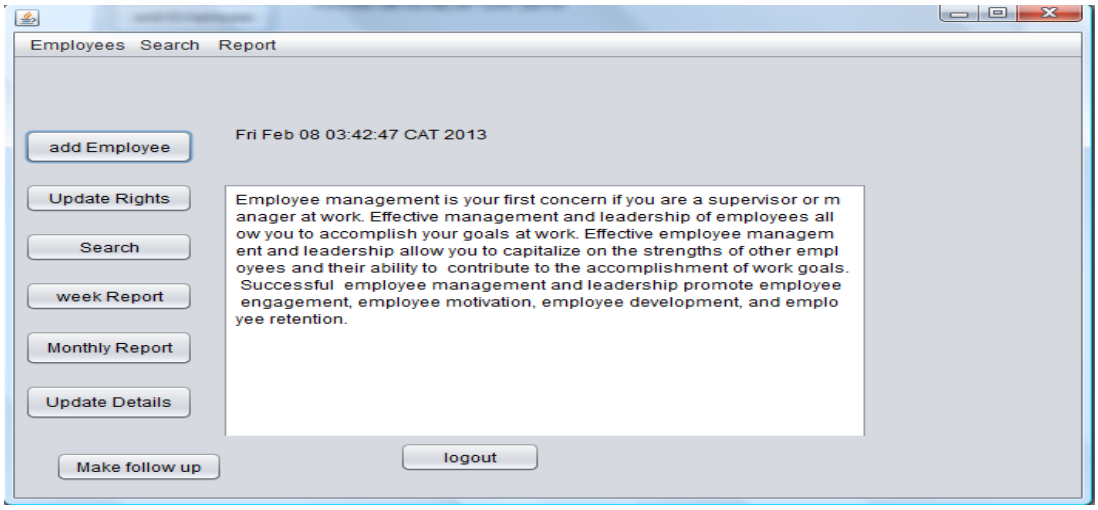


Figure 5.6 interface

During the process of mining sequence of events to discover sequential rules, administrators were allowed to interact with the cloud. Once a administrator completed the task, sequence of events was sent to the database as SOAP messages by Web service security. Since the process of discovering sequential rules was done offline, every sequence of event that were generated after administrator interacted with the cloud were then extracted to rule learning algorithm to discover sequential rules that form a user profile from the training data set.

5.5 Summary

In chapter 5, we have presented the implementation of our model. The design was achieved through the use of use case, sequence diagrams. The environmental setup shows how the preparation for development our model was setup, environmental setup indicates the number machines that were also used, that allowed us to deploy our solution. In the next chapter, we present an experiment conducted, results obtained and results discussions.

Chapter 6

EVALUATION AND DISCUSSION OF RESULTS

6.1 Introduction

This chapter reports the experiment that was conducted to prove the concept developed in this work. Section 6.2 discusses the processes involved in detecting a malicious insider. Section 6.3 provides the evaluation of the insider threat reduction model. Section 6.4 presents the experiments that investigate the effect of increasing minimum support on the quality of profiles mined based on specificity, sensitivity and precision. Section 6.5 presents the effect of increasing minimum support when observing the true positive rate. Section 6.6 presents type I error with the aim to observe whether increasing minimum support decreases or increases false positive rate. Section 6.7 presents type II error with the aim of observing whether increasing minimum support decrease or increase the false negative rate. Section 6.8 presents the effect of increasing minimum support on the number rules generated. Section 6.9 presents the scalability of the learning process with the reduction of minimum support. Section 6.10 presents the effect of reducing minimum support on sensitivity and specificity .The results are discussed in Section 6.11 whilst section 6.12 summarises the chapter.

6.2 The Process of Identifying Malicious Insider

Under ideal circumstances the solution was supposed to be evaluated at runtime, when users were actually using it to support some daily business processes. However, owing to some constraints the system was evaluated otherwise. Such constraints included not having many users in order to generate profile in the runtime environment that could represent one user in the system. The second constraint was of not having many users in generating enough data that could be used for testing. An alternative way of achieving the same goal for classification problems was to do the evaluation experiments offline, provided that there is a way of collecting data with a known class labels attribute values. The data were divided into two; training dataset and testing data set (*Wang et al. 2003*). The training dataset was used to build the model and the test dataset was used to evaluate the quality of the model developed. In our case the training dataset was used to create the user profiles, which would be in the form of a set of sequential association rules. To evaluate the effectiveness of our approach in detecting masqueraders the testing dataset was used. The hypothesis here was a given user profile should be able to uniquely identify each user in the system and could be used to detect masqueraders.

The process of generating the training data was as follows: This was done by first assigning each user to a specific role in the system that was different from other users. After a role assignment, each user was then allowed to perform operations in the system in order to generate raw data. The purpose of allowing users to perform operations in the system was to form known class label first that represented a user in the system, because our experiments were conducted offline not at runtime. We considered the first ten operations with the purpose of deriving the training dataset in order to form the profile of the user in the system. After the training dataset was generated it was then later serve as an input into the rule learning algorithm to generate sequential patterns

for the purpose of generating user profile. After user profile was created we then knew beforehand how each user was expected to behave in the system. Figure 6.1 shows the nature of how both training and testing dataset look like.

UserID	Training/Testing Dataset
User 1	(9, 10, 1, 1, 1, 1) (9, 10, 1, 1,1,11) (9, 10, 1, 1, 32, 9 10) (9, 10, 1, 11, 32, 1) (9, 1, 9, 1, 1 9, 1) (9, 1, 9, 1, 32, 10 1) (9, 1, 9, 11, 9, 11) (9, 1, 9, 11, 32, 1)

Figure 6.1: Shows Snapshot for Both Training and Testing Dataset

The testing dataset was generated by considering operations that were performed by a different user that was using valid credentials with the aim of observing patterns based on the raw data that later form a testing dataset. All the sequence of events that were generated from the original user was then used as a baseline for testing purposes in the system.

What must be noted is that, the use of term malicious actions and a malicious user are used in a different manner in this research. Malicious actions are defined as a sequence of events that lead to an attack while a malicious user is defined as someone who has been found or proven to be malicious based on his or her behaviour patterns.

Figure 6.3 shows the snapshot sequential rules generated from training dataset that form a profile of one of the users.

UserID	Sequential rules	Support	Confidence
User 1	[9, 10, 1, 1, 1 → 1]	0.7	0.8749999999999
	[9, 10, 1, 1, 1 → 1 1]	0.7	0.8466666666666
	[9, 10, 1, 1, 32 → 9 10]	0.7	1.0
	[9, 10, 1, 11, 32 → 1]	0.7	1.0
	[9, 1, 9, 1, 1 9 → 1]	0.7	0.8749999999999
	[9, 1, 9, 1, 1 9 → 1 1]	0.7	0.8749999999999
	[9, 1, 9, 1, 32 → 10 1]	0.7	0.7777777777777
	[9, 1, 9, 1, 32 → 32]	0.7	0.7777777777777
	[9, 1, 9, 1 1, 9 → 1 1]	0.7	0.8274444444444
	[9, 1, 9, 1, 1 9 → 1]	0.7	0.8945555555555
	[9, 1, 9, 1 1, 32→ 1]	0.7	0.8499999999999

Figure 6.2: Snapshot of the user profile

Figure 6.3 shows the profile of the user generated from training dataset. What can be observed is that from the first row in the user profile is that, when event 9, 10, 1, 1, 1 occur event 1 is mostly to occur next with a confidence of 87 %. Each sequential rule in the profile indicates how each user most likely to behave in the future is given a set of events in the rule antecedent.

All sequences of events that were generated that form testing dataset were then later extracted to matching algorithm in order to reason about the behaviour pattern of the user, whether a user was deviating from expected patterns. A behaviour of the user is said to be malicious if the execution of events in the testing data are not the same with the user profile. In this case, a user was performing actions that are not defined as a normal behaviour in the profile of the user. Comparing the profile of the user and testing dataset helped us in detecting a malicious user who pretended to be a legitimate user by comparing profiles of the user and testing dataset. If there were no events from the testing dataset that trigger and fire sequential rule in the profile of the

user, the system returned false indicating that no sequential rule matched and the system immediately pick up that user as a malicious user. The super administrator was alerted about malicious activities that were carried out by such malicious user.

6.3 Evaluation of insider threat reduction model

The evaluation was conducted to prove the concept put forward in this work and to support our claim that malicious users can be detected by observing their behaviour patterns when using the system. In this work, to prove the effectiveness of the insider threat reduction model, the model was evaluated based on the following metrics:

1. Accuracy: accuracy in this work was evaluated using sensitivity, specificity and precision.
2. The effect of increasing minimum support on the number of true positives rate.
3. Type 1 error and Type 2 errors
4. The effect of increasing minimum support (minsup) on the number of rules that were generated.
5. The scalability of learning process with the reduction of the minimum support (minsup).

Accuracy was measured in terms of sensitivity and specificity. Both specificity and sensitivity are statistical measures of performance of the binary classification test. The measures are defined as follows:

$Sensitivity = Number\ of\ True\ Positive\ (TP) / Number\ of\ Positives$..... [1]

Number of true positives (non-malicious): refer to the number of non-malicious actions that were correctly identified non malicious. That is, sensitivity is defined as the probability that the test say a user is normal when a user is really normal.

$Specificity = Number\ of\ True\ Negative / Number\ of\ Negative$..... [2]

Number of true negative (malicious): refer to events that were correctly labeled as negative (malicious). That is, specificity measures the proportion of malicious actions that were correctly identified.

Having high sensitivity and specificity indicates a lower false positive rate and false negative rate. For a normal user, who has a prior knowledge on using the system and performing predefined operations it is expected to generate less false negative rate. Our main concern is in detecting user who masquerade in the system as a normal user and reduce the false positives rate and false negatives rate. Once false negatives and false positives are reduced, it is an indication of the reduction of insider threats emanating from malicious administrator or insider of the cloud service provider.

False positive and false negative measures are defined as follows:

$False\ Positive\ Rate\ (FPR) = 1 - specificity$..... [3]

Where false positives are negative tuples that were incorrectly labelled

$False\ Negative\ Rate\ (FNR) = 1 - sensitivity$ ..... [4]

Where false negative refer to positive events that were incorrectly identified

In our evaluation minimum support was used as a controlled factor throughout in our experimentation to trim event of a sequence that did not meet minimum support.

A sequential rule X→Y is a relationship between two itemsets X, Y such that X, Y⊆ I, where X ⊂ I, Y ⊂ I, X∩Y=∅. We used the standard formula defined in the field of data mining for calculating the sequential support (*Fournier-Philippe et al. 2011*) and the formula is defined as follows:

$Sup(X \rightarrow Y) = \sup(X * Y) / |S|$..... [5]

Where S is the total size of the database X, Y appears.

Precision was also considered. Precision is defined as follows

$Precision = True\ Positives / (True\ Positives + False\ Positives)$ ..... [6]

6.4 Effect of Increasing Minimum Support on the Quality of the Profiles Mined.

6.4.1 Experimental Setup

This experiment was carried out with the aim of assessing the effect of increasing minimum support on the quality of the profiles mined. Assessing the quality of profile helped us in measuring accuracy of the system, in terms of detecting malicious users that deviate from expected behaviour patterns. The quality of profile was measured using specificity, sensitivity and precision. In this experimental setup, we compared testing dataset and user profile from the same user to see how well our solution could accurately detect the deviation from expected behaviour patterns. Instead of using one dataset, we used three datasets in order to prove the

concept that was put forward in this research. Using many datasets or experimenting with different datasets help in ensuring results that are obtained or conclusions that are made from different datasets in order to support our claim are sufficient enough. Both user profile and testing dataset served as an input to the matching algorithm in order to reason about the behaviour patterns of the user. Sequence of events in the test data set was then compared to the rules in the user profile.

Every sequence of events in the testing dataset that triggered and fired some sequential rule in the user profile was labelled as a false negative and the rest of the events in the testing data were labelled as true negatives. To observe events that are labelled as true positive and false positive, we took the same sequential rules in the user profile and some sequence of events that were not used in the generation of the user profile. Every sequence of events that trigger and fired rules in the user profile were labelled as true positive while the rest were labelled as false positive. After obtaining the number of false positive, false negative, true positives and true negatives sensitivity, specificity and precision were computed. Formulas for compute sensitivity, precision and specificity are provided in Section 6.2.1. See appendix for the matching algorithm that was developed. Results obtained by running this experiment is presented

Table 6.1 Data Obtained from User 1

Minsup	Sensitivity	Specificity	Precision
0.6	0.75	0.84	0.20
0.7	0.80	0.88	0.50
0.8	0.84	0.92	0.66
0.9	0.87	0.99	0.72

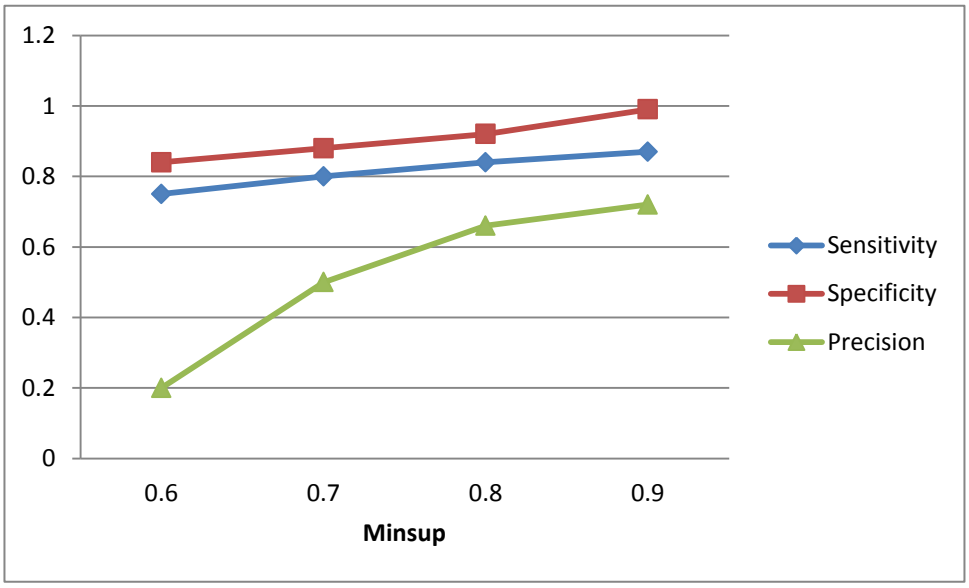


Figure 6.3: Effect of Increasing Minsup on Sensitivity, Specificity and Precision

Table: 6.2 Data Obtained from User 2

Minsup	Sensitivity	Specificity	Precision
0.6	0.66	0.67	0.81
0.7	0.83	0.93	0.86
0.8	0.85	0.94	0.91
0.9	0.94	0.98	0.94

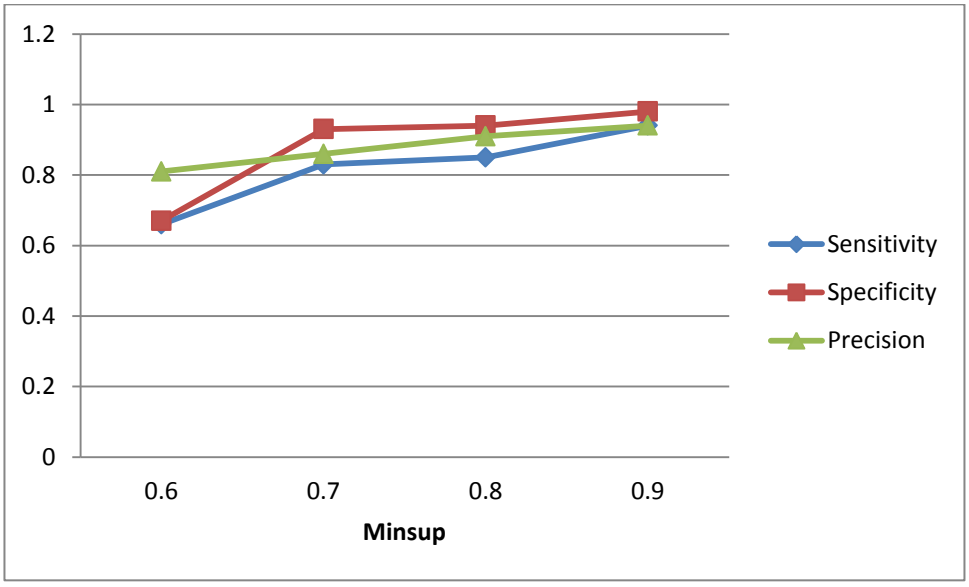


Figure 6.4 Effect of Increasing Minsup on Sensitivity, Specificity and Precision

Table 6.3 Data Obtained from User 3

Minsup	Sensitivity	Specificity	Precision
0.6	0.76	0.78	0.89
0.7	0.85	0.94	0.90
0.8	0.86	0.96	0.91
0.9	0.89	0.98	0.95

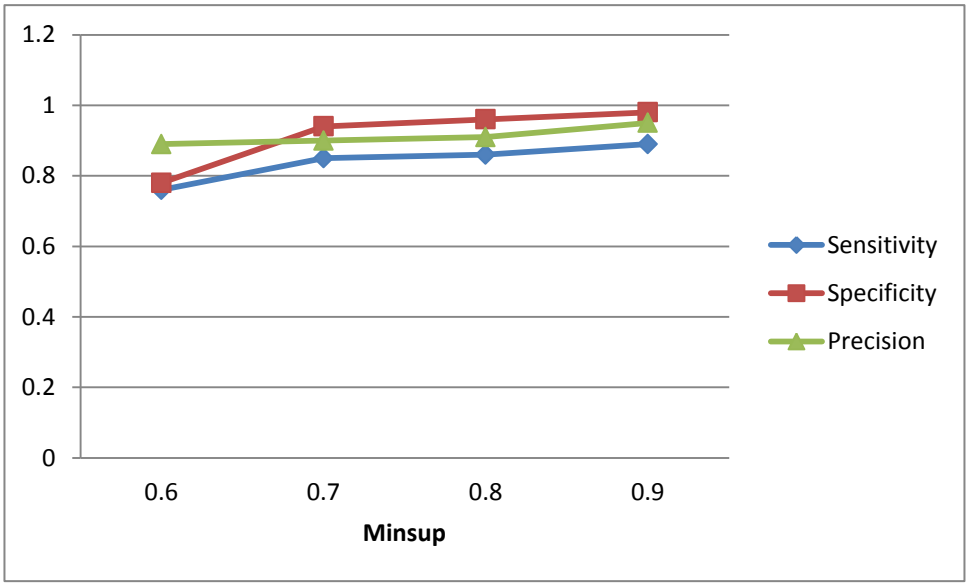


Figure 6.5 Effect Increasing Minsup on Sensitivity, Specificity and Precision

6.4.2 Testing Result

From Figures 6.3 – 6.5 show results obtained when investigating the quality of profile mined in terms of specificity, sensitivity. From the results that were obtained as minimum support was

increased. We observed the increase in sensitivity about non-malicious actions that were correctly identified originating from a normal user that behaved in a normal way. At the same time we also observed the increase in specificity about the malicious actions were correctly identified that were higher than non-malicious actions that were correctly identified. As a result, this proved that our solution had a capability of detecting user that deviate from expected behaviour patterns. Since malicious actions were higher than non-malicious actions this indicated the overall ability of the algorithm in discriminating malicious users from normal users based on the threads of the graphs. This mean that, the algorithm managed to accurately detect malicious users that deviated from expected behaviour patterns efficiently and did not wrongly classify the normal user.

This result indicated that every user that was deviating from expected behaviour patterns the system was able to pick that user and classify as a malicious user, since malicious actions represent unexpected behaviour patterns were higher than non-malicious actions that represented expected behaviour patterns. As a result, this proved the deviation of the user from the expected behaviour patterns. This is to say, the way each user was behaving in the system in terms of executing task was different from the way a normal user usually behaves when executing task in the system. The developed model shows that, it has that ability or capability of identifying a malicious insider based on their patterns when observing malicious actions (specificity) and non-malicious actions (sensitivity). Having high number malicious actions and non-malicious actions were an indication that the number of false positive and number false negative generated were reduced. This means the profiles that were mined were of a high quality in terms of accurately predicting the actual behaviour of the user.

6.5 Effect of increasing Minimum Support on the Rate of True Positives

From the same set of experiment we conducted, we then measure the effect of increasing minimum support on the rate of true positive. The main purpose of measuring the detection rate was to observe whether increasing minimum support reduced or increased detection rate malicious actions performed by malicious insiders. Hence, the detection rate of malicious actions of malicious insiders was expected to increase in the increase of minimum support.

Table 6.4: Data Obtained When Investigating True Positive.

Minsup	TPR	TPR	TPR	Average
0.6	0.20	0.81	0.89	0.63333
0.7	0.50	0.86	0.9	0.75
0.8	0.66	0.91	0.91	0.826667
0.9	0.72	0.94	0.95	0.87

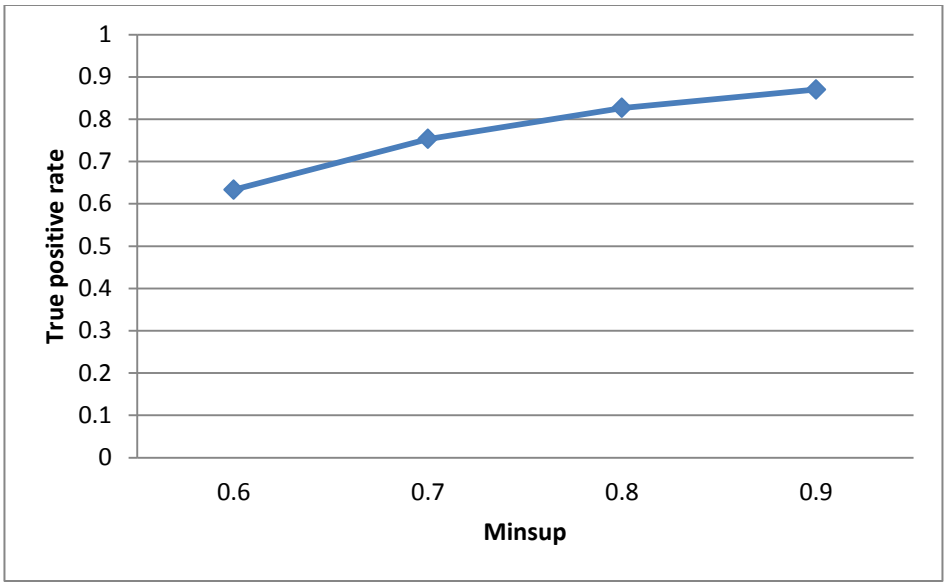


Figure 6.6: Increasing Ninsup on TPR

6.5.1 Test Result

Figures 6.6 indicate results obtained when observing the effect of true positives rate when minimum support was increased. From the graphs obtained we observe an increase in the detection rate as minimum support was increased. From the results obtained, this is an indication that was accurate in terms of malicious actions or users that deviate from the expected behaviour patterns in the system. The main reason for the increase in the detection rate was that, the system was accurate in terms of classifying malicious actions as malicious that deviate from expected behaviour patterns that originates from malicious insiders. As a result, this indicates that the chances of catching malicious user that deviate from expected behaviour are high. Hence, this result also proves the concept of detecting malicious insiders that deliberately deviate from expected behaviour patterns. This also shows that, the solution approach provided this research in reducing malicious insider has a capability of detecting malicious actions.

6.6 Type 1 Error

The first experiment also allowed us to measure type 1 error. A Type 1 error is another metric that is used to measure accuracy of the system. The main purpose was to observe increasing minimum support reduces or increases false positives rate. False positive rate occurs when the system misclassifies non-malicious actions to be malicious actions. Hence, if the false positives rate is decreased by increasing minimum support that gives assurance that the system is accurate in terms of not misclassifying non-malicious actions to be malicious. As a result, behaviour is expected of misclassifying non-malicious actions as malicious. The computation of type 1 error depends on malicious actions that were obtained in the first experiment. If malicious actions are high, the chances of misclassifying non-malicious actions as malicious actions are less.

Table 6.5: Data Obtained for Type 1 Error

Minsup	(1 – Specificity) = FPR (testing dataset 1)	(1 – Specificity) = FPR (testing dataset 2)	(1 – Specificity) = FPR(testing dataset 3)	Average
0.6	0.16	0.33	0.22	0.236667
0.7	0.12	0.07	0.06	0.083333
0.8	0.08	0.06	0.04	0.06
0.9	0.04	0.02	0.02	0.026667

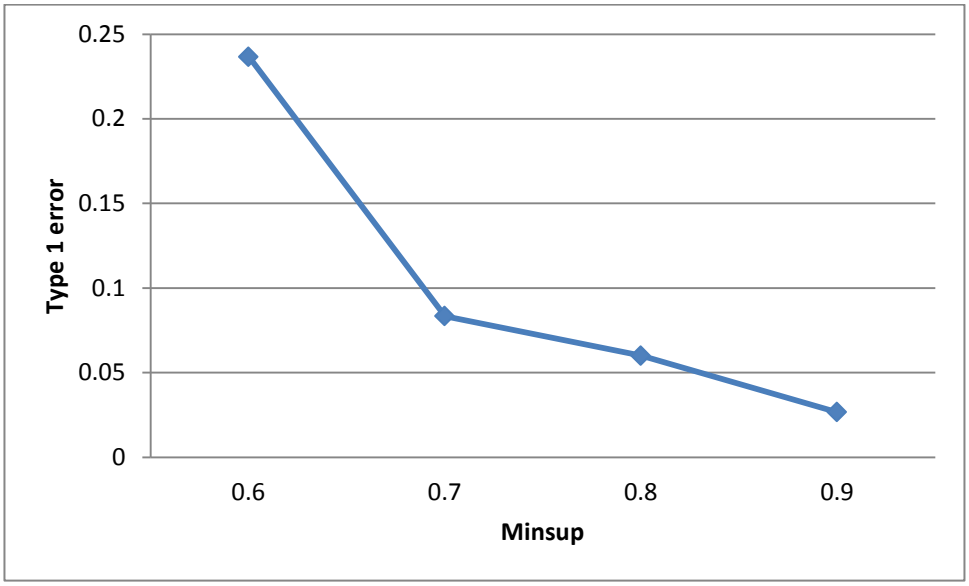


Figure 6.7: Type 1 Error.

6.6.1 Test Result

Figure 6.7 indicates the result that was obtained when investigating the effect of increasing minimum support on the rate of false positives. From the pattern of the graph obtained we observed the decrease in the misclassification of non-malicious action as malicious actions (false positive rate), this means that, the system was able to accurately classify non-malicious actions from legitimate insider not to be malicious. This result also showed that the rate at which normal user was deviating from expected behave was very low. As a result, that contributed in the decrease of false positives rate generated in the system. The main reason for the decrease in false positive rates depends on the specificity (malicious actions), the high the specificity (malicious actions) the lower the rate of false positives was observed, and this was also the evidence from the results obtained. Reducing the number of misclassifying non-malicious to be malicious (false positives rate) from insider indicated that a model was able to reduce insider attacks originating from legitimate insiders. As a result, this proved that the solution had a capability of reducing

insider attacks and this also showed that the developed model was able to effectively reduce false positive rate.

6.7 Type 2 Errors

A Type 2 error is a metric that was used to measure accuracy of the system. Measuring type 2 error depended on the first experiment that was conducted. The main aim was to observe whether increasing minimum support would decrease or increase the false negatives rate. False negatives rate occurs when the system classifies malicious actions from an insider not be malicious actions. The aim was to observe whether increasing minimum support decrease or increase the rate of misclassifying malicious actions to be non-malicious. The computation of type 2 error depends on the sensitivity. If the misclassification of malicious actions as non-malicious (false negative rate) is reduced, the better the model that was developed to reduce false negative. The data obtained is also presented below:

Table 6.6: Data Obtained for Type 2 Errors

Minsup	(1 – Sensitivity) = FNR (user 1)	(1 – Sensitivity) = FNR (user 2)	(1 – Sensitivity) = FNR (user 3)	Average
0.6	0.25	0.34	0.24	0.276667
0.7	0.2	0.17	0.17	0.18
0.8	0.16	0.15	0.14	0.15
0.9	0.13	0.06	0.11	0.1

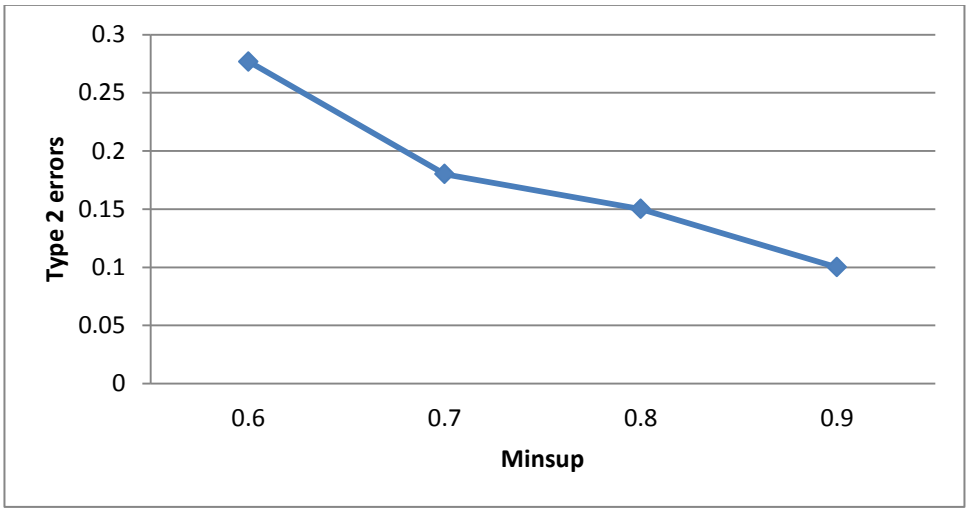


Figure 6.8: Type 2 Errors

6.7.1 Test Result

From the graph obtained what can be recognised is that, when minimum support is increased the pattern of the graph decreases indicating that the number of misclassifying malicious as non-malicious is reduced. This indicates that, the system was accurate in terms classifying malicious actions as malicious not either. This is also evident from the results obtained that the solution

approach that was provided has the capability of reducing insider attacks originating from insiders within the organisation. The reason for the decrease in false negative depends also on the sensitivity (non-malicious actions), this means that the higher the sensitivity (non-malicious actions) of sequential rules the lesser the number of false negative rate is experienced. This shows that, the model that was developed proved to be effective in reducing attacks that originates within the organisation based on the result obtained.

6.8 Effect of Increasing Minimum Support on the Number of Rules Generated.

The first quality attribute of our solution also depended on the first experiment that was conducted. The training dataset depicted the behaviour of normal user that perform a sequence of events in the same way every time when using the system. Three users were given legitimate credentials to access the system in order to generate a sequence of events that depict behaviour pattern of each user when using the system. Each user executes some tasks in the system in order to generate raw data. Once the raw data is generated we then derive the training dataset, after the training dataset is created each user have its own training dataset. Later, all training dataset for each user serve as an input into the rule learning algorithm. The rule learning algorithm was then used to generate sequential rules from the training dataset and level 5 was used as a threshold to start generating patterns to the last level. What must be noted is that, the process of generating sequential is independent from other processes, because this experiment was conducted offline. All the sequential rules that were generated from the algorithm were stored as a user profile in the database. Each users was allowed to perform 10 operations in the system in total. The

purpose of this test was to investigate the effect of increasing minimum support on the number of rules generated holding minimum confidence at 70 percent. Minimum confidence was held at 70 percent because we were interested in seeing the effect of increasing minimum support on the number of rules generated. Experiments were conducted in all three datasets that were used at the same time varying minimum support from 60%, 70%, 80 % and 90%. Data that was gathered is presented below.

Table 6.7: Data Obtained for Computing Sequential Rules Obtained in Different Minsup When Minsup is Increased.

Minsup	**User 1**	**User 2**	**User 3**	**Average**
0.6	3666	7500	6378	5848
0.70	3149	1130	1050	1776.333
0.8	517	830	577	701.3333
0.9	8	85	79	57.3333

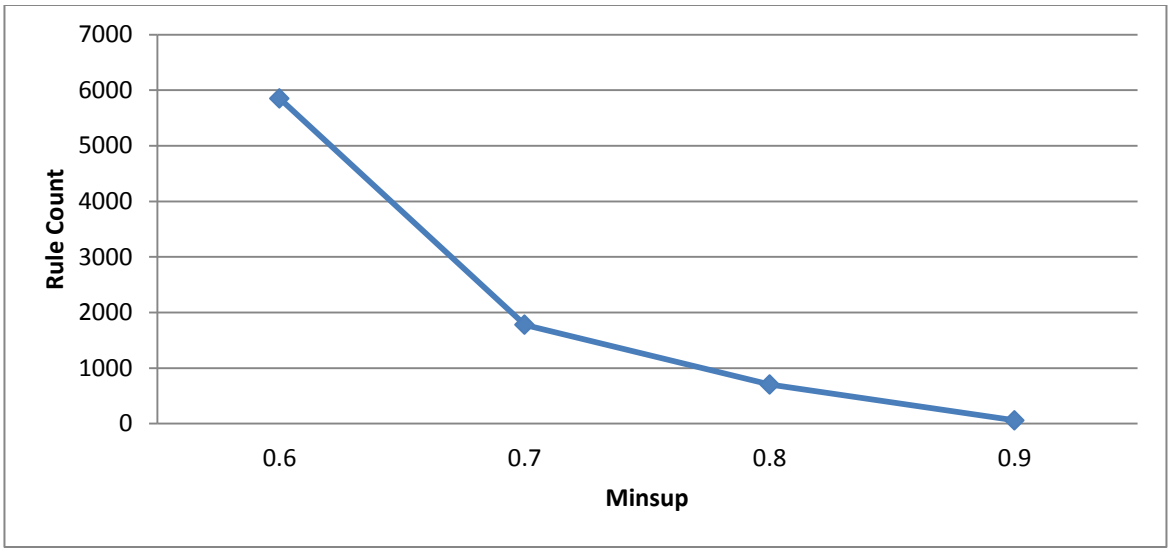


Figure 6.9: The Effect of Increasing Minsup on the Number of Rules Generated

6.8.1 Test Result

The results in Figure 6.9 and Table 6.7 show the effect of increasing minimum support on the number of rules generated. What can be observed from the graph is that, a number of sequential rules generated decreases as minimum support is increased. It is seen that at the minimum support of 0.6 a higher number of sequential rules were generated, while increasing minimum support decreases the number sequential rules generated. The reason for this decrease in the number of sequential rules generated was that, sequential rules that were previously above the minimum support do not form part of a sequential rules generation. As a result, this indicated that the results that were obtained in assessing the first quality attribute of our solution were significant. Because it was expected that in 0.6 there were many sequential rules that were generated as minimum support was increased, less number of sequential rules that were generated.

6.9 Scalability of the Learning Process with Reduction of Minimum Support

The second attribute of our solutions also depended on the first experiment. The attribute was conducted to evaluate the scalability of the learning process with the reduction of minimum support. It was observed that increasing minimum support decreases number rules generated. For scalability of the learning process, the aim was to investigate the performance while reducing minimum support to see whether learning process scale well or not. To obtain results that are not biased, we ensured that only the experiment was running all other applications were closed to provide fairness in the result obtained. The purpose of this test was to observe the processing time in milliseconds when minimum support was reduce from 0.9%, 0.8%, 0.7%, and 0.6% for each test. The data that was collected by running this experiment presented in Table 6.8.

Table 6.8: Data Obtained for Computing Average Processing Time in Milliseconds

Minsup	**User 1**	**User 2**	**User 3**	**Average**
0.9	704	865	785	784.6667
0.8	1521	2153	1742	1805.333
0.7	3001	4727	4585	4104.333
0.6	7327	14021	1291	11443

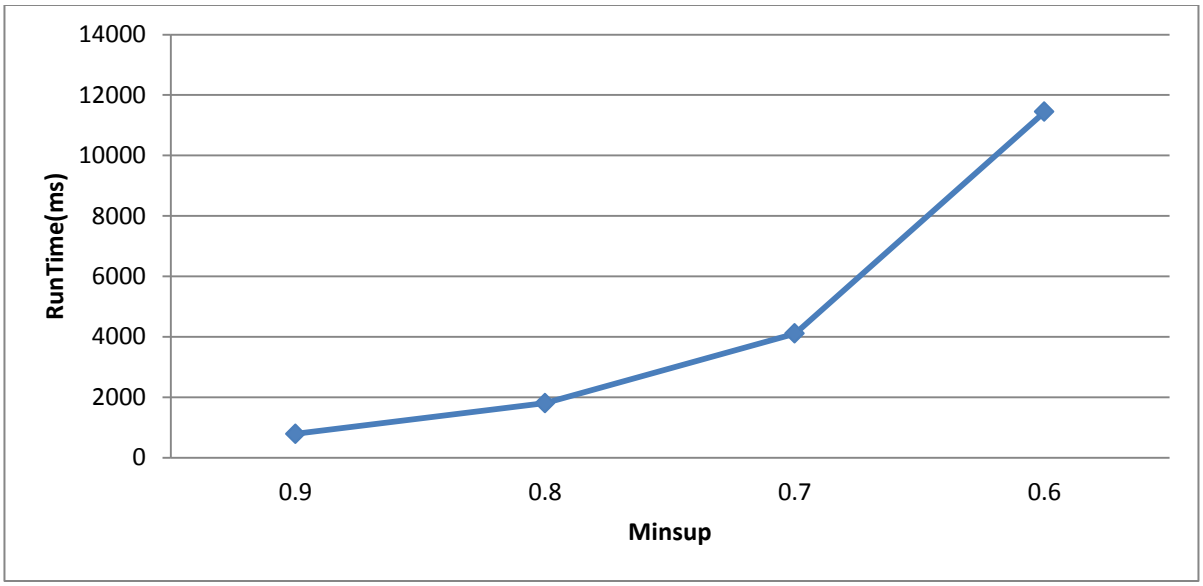


Figure 6.10: Reduction of Minsup on the Learning Process in Runtime

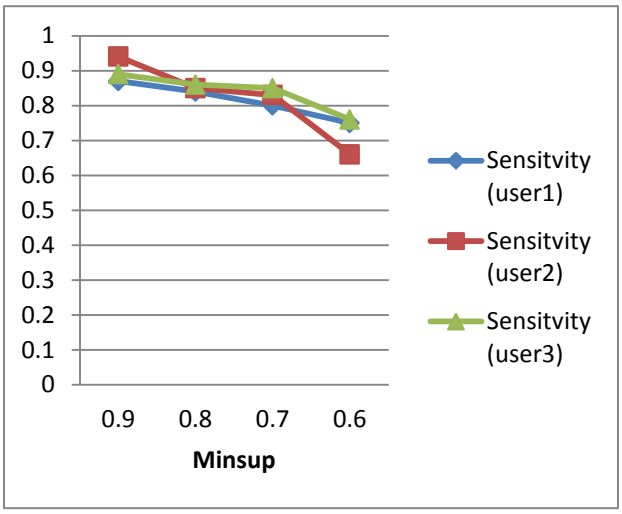
6.9.1 Test Result

Figure 6.10 and table 6.8 shows the processing time when investigating the scalability of the learning with the reduction of minimum support. What can be observed from the pattern of the graphs is that, at 0.9 % there are few sequential rules that were generated, so the time it took to execute the learning process was lower compared to the one when support is 0.6. From the results obtained, it can be seen that as minimum support decreases, the execution time for learning process increases in an exponential manner. The main reason is that, the pattern in the graph shows that the sequential rule mining algorithm is not scalable with the decrease in the minimum support.

6.10 Effect of Reducing Minimum Support on Sensitivity and Specificity

The third attribute of our solution was evaluated based on the effect of reducing minimum support on sensitivity and specificity. In assessing third attribute we were mainly concentrated

on seeing whether reducing minimum support, the accuracy would decrease drastically or it would decrease at the reasonable rate.



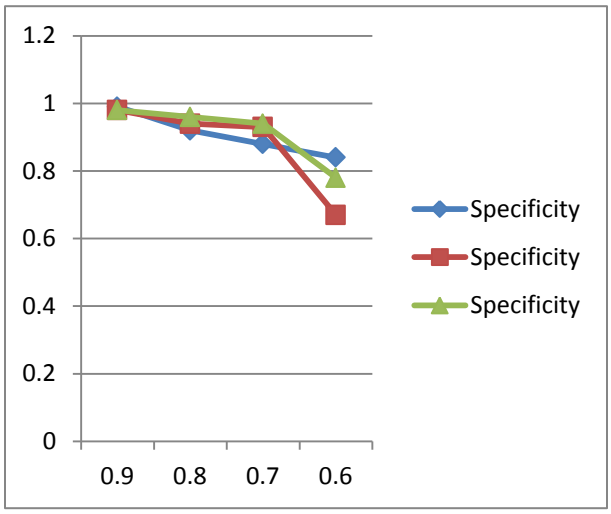


Figure 6.11-1: Reducing Minsup on Sensitivity Figure 6.11-2: Reducing Minsup on Specificity

6.10.1 Test Result

From the result obtained when minimum support was reduced, we observed the decrease in the accuracy of the system, as a result this indicated that, even though minimum support was reduced the pattern of the graphs decreased at a low rate. This also shows that the quality of the solution was from the result obtained when observing accuracy is not compromised at a high rate. Since, the decreased ranges between 98 to 67 percent.

6.11 Result Discussion

From the foregoing experiments, the main basic conclusion that can be made is that our insider threat reduction model has the capability reducing the number of insider attacks originating from legitimate users. The literature reviewed states that, though composite role based monitoring and

intrusion detection system are widely adopted, they still lack the aspect of monitoring as they generate high rate of false negative alarm and do not provide adequate monitoring because they classify unknown attacks as normal behaviour patterns. This became a concern because the cloud service providers cannot guarantee their security system. This thus necessitates an insider threat reduction model that is flexible in terms of detecting users that deviate from expected behaviour patterns that lead to an attack.

The experimental results on quality of profile mined in Section 6.4 proved the concept that was put forward in this work of detecting the deviation from expected behaviour patterns. The quality of profile mined experimental results showed that the both sensitivity (non-malicious) and specificity (malicious) increased as minimum support was increased. When observing non-malicious actions and malicious events, malicious events were higher than non-malicious actions that indicate that the user was deviating from the expected behave pattern. Our results show that when we kept on increasing minimum support would definitely see users that deviate from expected behaviour when observing sensitivity and specificity.

The desirable properties of the monitoring approach should reduce the false positives rate and false negatives rate. Results on type 1 error in Section 6.6 proved that the number of false positive rate is decreased. This is to say, there are lesser number of false positive alarm. This shows the strength of the model that was developed in reducing insider attack. In Section 6.7 the result of type 2 errors proved that the number of false negative rate was reduced. Having a small number of false positive and false negatives rate indicated that threat emanating from a malicious insider was reduced. As a result, this helped us in answering our research question.

The results in the first quality attribute of our solution on the number of sequential rules that are generated in Section 6.8 proved that as minimum support was increased the lesser the number of rules that were generated. As the number of rules generated decreases at the same time rules that are generated in high support are strong rules than those rules that were generated at a lower support. The results of the scalability of the learning process in Section 6.9 which the second attribute of our solution proved that the rule learning algorithm was not scalable with the decrease of minimum support. From the results obtained when assessing the third quality attribute of our solution, the result also proved that, there is a decreased in the accuracy of the system but not in a drastic manner. This shows the quality that our solution could provide when mitigating insider threat. We then conclude by saying insider threat reduction model that was developed for monitoring malicious users in the system, proved to be adequate in terms of detecting masquerader within the system through observing their behaviour patterns and reducing insider attacks that originates within the cloud service provider side.

6.12 Summary

In this chapter we have presented results for insider threat reduction model for monitoring the behaviour pattern of the user. The rule learning algorithm was used to learn the sequence of events from raw data in order to build the profile of the user. One experiment was carried out and quality attributes of the solutions were also derived from the same experiment. Quality attributes included the effect of increasing minimum support on the number of rules generated, scalability of the learning process with the reduction of minimum support and the effect of reducing minimum support on sensitivity and specificity.

The effect of increasing minimum support when observing sensitivity, specificity and precision, the effect increasing minimum support on rate of true positives, type 1 and type 2 errors was used to measure the accuracy of the system. The main goal of the insider threat reduction model that was proposed by reducing insider attack in the cloud service provider was achieved in terms of reducing false positives rate, false negatives rate that were generated and observing the sensitivity of sequential rules generated. From the literature reviewed it is stated that, the good model should produce less number of false positive rate and false negative rate, while the detection rate is expected to increase in order to catch malicious user in the system (*Srivastava et al. 2006, Zhang et al. 2004*). Our results also showed what is expected to be the properties of the good model. These were achieved by training real users and assigned them on a different role in the system and also give other users legitimate access to the system to see whether the algorithm developed will reason about behaviour patterns when comparing test data against user profile.

Chapter 7

CONCLUSION AND FUTURE WORK

7.1 Introduction

In this research, we have tried to address the challenges of insider attack that usually originate from legitimate users of the cloud service provider by monitoring user behaviour pattern which is a problem faced by the cloud service provider. The challenge of insider attack for monitoring user behaviour brought the need for security model to ensure cloud resources were secured. This results in a lot of information being compromised at the cloud service provider side, because of unexpected behaviour performed by insiders who have access to the system. As a result, of unexpected behaviour of entities involved, the insider threat reduction model becomes a high necessity in ensuring that users who are masquerading in the system through their behaviour patterns are recognised. Objective 1 and objective 2 in this research work were achieved by reviewing the literature and pointed out the advantages of the adopted approach. Objective 3 was achieved by developing an insider threat reduction model and evaluated its effectiveness in terms of detecting a masquerader within the system.

This work specifically focused on the number of sequential rules generated, scalability of the learning process, effect on increasing minimum support on the quality of sequential mined observing sensitivity, specificity and precision, the effect of increasing minimum support on the rate of false positive, false positive, type 1 and type 2 errors in order to judge the behaviour pattern of the user. This is based on the fact that many users want to leverage on the cloud because of advantages cloud can provide to cloud users by hosting their application and sensitive

data. Hence, if a malicious user masquerades in the system as a normal user and his or her malicious behaviour is not detected, a malicious user can violate confidentiality and integrity of information stored in the cloud. This, thus lead to many users being reluctant about hosting their application in the cloud if there is no security model put in place to monitor such behaviour.

The overall goal of this research work was to come up with the insider threat reduction model for cloud environment that would reduce inside attack at the cloud service provider side by monitoring the user behaviour and differentiate between normal users and malicious users based on the behaviour pattern. To this end, we have managed to model, implement and evaluated an insider threat reduction model based on the metrics mentioned above. This was to ensure that the model can reason about the pattern of sequential rules generated by normal users and the sequence of events generated from malicious users. Experiments were carried as indicated in chapter 5 to prove the argument that was put forward in this research work in answering our research question stated in chapter 1. This chapter wraps the whole research that was carried out and summaries it in Section 7.2. Section 7.3 we conclude by suggesting a limitation of our work and suggest envisaged future work.

7.2 Conclusion

The global acceptance of the cloud has led many individuals and business willing to leverage on it, because of the advantages that are provided by the cloud for hosting their sensitive information about their business. As a result, business and individual are required to trust the cloud service provider with their information. However, security is the major issue facing the cloud service provider. Because a malicious user of cloud service provider can steal credentials of the credentials of legitimate users and perform malicious actions and violate the integrity and

confidentiality of information by behaving in unexpected ways. This brings the system to be dealing with the unexpected behaviour that originates from the user that have access the system using the credentials of the normal user. To provide solution for such environment, there was a need to use data mining approach that mines the sequential pattern of the user (*Fournier-Phillip et al. 2011*). The rule learning algorithm was used to learn patterns from raw data in order to generate user profiles and matching algorithm was used to match sequence of events with the rules in the user profiles.

From the literature reviewed we discovered that composite role based monitoring is promising technology used in addressing the issue of the insider by assigning users to appropriate roles to reduce the inconstancy that occurs when two users are assigned to the same role. Even though it has been accepted as promising technology but fails to monitor behaviour patterns of the user once a user has been given access to the system and composite role based monitoring increase the rate of false positive alarm that are generated in the system. This has become a major problem when business host sensitive data in the system that they do not have control over their data and they do not know who have access to what in their sensitive information. This resulted in the development of insider threat reduction model in order to monitor the behaviour pattern of malicious user when accessing the system.

Sets of experiments were carried out as indicated in chapter 6, that show the behaviour pattern between a malicious user and a normal user when using the system. An analysis of the results obtained from the implementation revealed that our insider threat reduction model managed to identify a masquerader and a normal user based on their behaviour patterns. In essence, the evaluation concluded that our insider threat reduction model as proposed in this research provides the required monitoring in detecting a potential masquerader in the system that has

performed malicious activities in order to full-fill his or her goals. This thus means there would not be any malicious user that cannot be detected because of the security system implemented.

7.3 Limitation and Future Work

Even though our proposed insider threat reduction model proved to be an applicable approach for reducing insider attack by monitoring behaviour pattern of malicious user in the dynamic environments such as cloud, it has some limitations which could be recommended for future enhancements. During the process of testing our system CPU utilisation and memory consumption was not tested. In future we will like to know whether the model can be able to recognise a malicious user and a normal user based on the CPU utilisation and memory consumption. Another limitation of our work is that, experiments were conducted in the homogenous environment in the future we would like to deploy our model in the real cloud environment that can be accessed in the heterogeneous environment. The process of discovering the behaviour pattern of the user was done offline, in future will like to see the impact it might have when discovering behaviour patterns of the user in running time environment.

BIBLIOGRAPHY

Agrawal, R., & Srikant, R. (1995). *Mining sequential patterns*. In Data Engineering, 1995. Proceedings of the Eleventh International Conference on (pp. 3-14). IEEE.

Asma, A., Chaurasia, M. A., & Mokhtar, H. (2012). *Cloud Computing Security Issues*. International Journal.

Baker, W., Goudie, M., Hutton, A., Hylender, C., Niemantsverdriet, J., Novak, C., & Tippett, P. (2010). *Verizon 2010 Data Breach Investigations Report*. Verizon Business.

Basescu, C., Carpen-Amarie, A., Leordeanu, C., Costan, A., & Antoniu, G. (2011, March). *Managing data access on clouds: A generic framework for enforcing security policies*. In Advanced Information Networking and Applications (AINA), 2011 IEEE International Conference on (pp. 459-466). IEEE.

Bamiah, M. A., & Brohi, S. N. (2011). *Seven deadly threats and vulnerabilities in cloud computing*. International Journal of Advanced engineering sciences and technologies.

Band, S. R., Cappelli, D. M., Fischer, L. F., Moore, A. P., Shaw, E. D., & Trzeciak, R. F. (2006). *Comparing insider IT sabotage and espionage: A model-based analysis* (No. CMU/SEI-2006-TR-026). CARNEGIE-MELLON UNIV PITTSBURGH PA SOFTWARE ENGINEERING INST.

Bouchahda, A., Le Thanh, N., Bouhoula, A., & Labbene, F. (2010). *Enforcing Access Control to Web Databases*. In Computer and Information Technology (CIT), 2010 IEEE 10th International Conference on (pp. 612-619). IEEE.

Brodkin, J. (2008). Gartner: *Seven cloud-computing security risks*.

Brodley, C. E. & Lane, T. (1997). *Sequence matching and learning in anomaly detection for computer security.* In AAAI Workshop: AI Approaches to Fraud Detection and Risk Management (pp. 43-49).

Brunette, G., & Mogull, R. *Security Guidance for critical areas of focus in Cloud Computing V2. 1. CSA* (Cloud Security Alliance), USA (2009).

Casola, V., Rak, M., & Villano, U. (2010). *Identity federation in cloud computing*. In Information Assurance and Security (IAS), 2010 Sixth International Conference on (pp. 253-259). IEEE.

Cappelli, D. M., Moore, A. P., & Trzeciak, R. F. (2012). *The CERT Guide to Insider Threats: How to Prevent, Detect, and Respond to Information Technology Crimes (Theft, Sabotage, Fraud)*. Addison-Wesley Professional.

Colombe, J. B., & Stephens, G. (2004). *Statistical profiling and visualization for detection of malicious insider attacks on computer networks*. In Proceedings of the 2004 ACM workshop on Visualization and data mining for computer security (pp. 138-142). ACM.

Choubey, R., Dubey, R., & Bhattacharjee, J. (2011). *A survey on cloud computing security, challenges and threats*. International Journal on Computer Science and Engineering (IJCSE*)*, *3*(3), 1227-1231.

Crampton, J., & Huth, M. (2009). *Detecting and countering insider threats: Can policy-based access control help*. In Proceedings of the 5th International Workshop on Security and Trust Management.

Chen, Y., & Malin, B. (2011). *Detection of anomalous insiders in collaborative environments via relational analysis of access logs.* In Proceedings of the first ACM conference on Data and application security and privacy (pp. 63-74). ACM.

Chinchani, R., Iyer, A., Ngo, H. Q., & Upadhyaya, S. (2005). *Towards a theory of insider threat assessment.* In Dependable Systems and Networks, 2005. DSN 2005. Proceedings. International Conference on (pp. 108-117). IEEE.

Claycomb, W. R., & Nicoll, A. (2012). *Insider Threats to Cloud Computing: Directions for New Research Challenges*. In Computer Software and Applications Conference (COMPSAC), 2012 IEEE 36th Annual (pp. 387-394). IEEE.

Davison, B. D., & Hirsh, H. (1998). *Predicting sequences of user actions*. In *Notes of the* AAAI/ICML 1998 Workshop on Predicting the Future: AI Approaches to Time-Series Analysis.

Das, G., Lin, K. I., Mannila, H., Renganathan, G., & Smyth, P. (1998*). Rule Discovery from Time Series.* In KDD (Vol. 98, pp. 16-22).

Denning, D. E. (1987). *An intrusion-detection model*. Software Engineering, IEEE Transactions on, (2), 222-232.

Dwyer III, S. J., Weaver, A. C., & Hughes, K. K. (2004). *Health Insurance Portability and Accountability Act.* Security Issues in the Digital Medical Enterprise.

Eom, J. H., Park, M. W., Park, S. H., & Chung, T. M. (2011). *A framework of defense system for prevention of insider's malicious behaviours.* In Advanced Communication Technology (ICACT), 2011 13th International Conference on (pp. 982-987). IEEE.

Eberle, W., Graves, J., & Holder, L. (2010). *Insider threat detection using a graph-based approach.* Journal of Applied Security Research, *6*(1), 32-81.

Eirinaki, M., & Vazirgiannis, M. (2003). *Web mining for web personalization*. ACM Transactions on Internet Technology *(TOIT)*, *3*(1), 1-27.

Grobauer, B., Walloschek, T., & Stocker, E. (2011). *Understanding cloud computing vulnerabilities*. Security & Privacy, IEEE, *9*(2), 50-57.

Glott, R., Husmann, E., Sadeghi, A. R., & Schunter, M. (2011). *Trustworthy clouds underpinning the future internet.* In The future internet (pp. 209-221). Springer Berlin Heidelberg.

Hamilton, H. J. & Karimi, K. (2005). *The TIMERS II Algorithm for the Discovery of Causality.* Proc. 9 th Pacific Asia Conference on Knowledge Discovery and Data Mining , 744-750.

Fournier-Viger, P., Faghihi, U., Nkambou, R., & Nguifo, E. M. (2010). *CMRULES: An Efficient Algorithm for Mining Sequential Rules Common to Several Sequences*. In FLAIRS Conference.

Fournier-Viger, P., Nkambou, R., & Tseng, V. S. M. (2011). *RuleGrowth: mining sequential rules common to several sequences by pattern-growth*. In Proceedings of the 2011 ACM Symposium on Applied Computing (pp. 956-961). ACM.

Jeffrey , H,., & Probst, C. W. (2011). *Insiders and insider threats—an overview of definitions and mitigation techniques*. Journal of Wireless Mobile Networks, Ubiquitous Computing, and Dependable Applications, 2(1), 4-27.

Han, J., Kamber, M., & Pei, J. (2006). *Data mining: concepts and techniques*. Morgan kaufmann.

Harms, S. K., Deogun, J., & Tadesse, T. (2002). *Discovering sequential association rules with constraints and time lags in multiple sequences*. In Foundations of Intelligent Systems (pp. 432-441). Springer Berlin Heidelberg.

Hofmeyr, S. A., Forrest, S., & Somayaji, A. (1998). *Intrusion detection using sequences of system calls*. Journal of computer security, *6*(3), 151-180.

Hsieh, Y. L., Yang, D.-L. & Wu, J. (2006). *Using Data Mining to Study Upstream and Downstream Causal Realtionship in Stock Market*. In Proc. 2006 Joint Conference on Information Sciences Laxman, S. & Sastry, P. 2006. A survey of temporal data mining. Sadhana 3: 173-198

Kuyoro, S. O., Ibikunle, F., & Awodele, O. (2011). *Cloud Computing Security Issues and Challenges.* International Journal of Computer Networks (IJCN),3(5), 247-255.

Khorshed, M. T., Ali, A. S., & Wasimi, S. A. (2011). *Monitoring insiders activities in cloud computing using rule based learning*. In Trust, Security and Privacy in Computing and Communications (TrustCom), 2011 IEEE 10th International Conference on (pp. 757-764). IEEE.

Liu, D., Wang, X., & Camp, J. (2008). *Game-theoretic modeling and analysis of insider threats*. International Journal of Critical Infrastructure Protection, *1*, 75-80.

Li, M., Yu, S., Ren, K., & Lou, W. (2010). *Securing personal health records in cloud computing: Patient-centric and fine-grained data access control in multi-owner settings*. In Security and Privacy in Communication Networks (pp. 89-106). Springer Berlin Heidelberg.

Lundin, E., & Jonsson, E. (2000). *Anomaly-based intrusion detection: privacy concerns and other problems*. Computer networks, *34*(4), 623-640.

Mannila, H., Toivonen, H., & Verkamo, A. I. (1997). *Discovery of frequent episodes in event sequences*. Data Mining and Knowledge Discovery, *1*(3), 259-289.

Manifesto, O. C. (2010). *Open cloud manifesto*.

Menascé, D. A. *The Insider Threat Security Architecture 2010. The Insider Threat Security Architecture*. International Conference on Computational Science and Engineering, 2009. CSE '09.

Myers, J., Grimaila, M. R., & Mills, R. F. (2009). *Towards insider threat detection using web server logs*. In Proceedings of the 5th Annual Workshop on Cyber Security and Information Intelligence Research: Cyber Security and Information Intelligence Challenges and Strategies (p. 54). ACM.

Nguyen, N., Reiher, P., & Kuenning, G. H. (2003). *Detecting insider threats by monitoring system call activity*. In Information Assurance Workshop, 2003. IEEE Systems, Man and Cybernetics Society (pp. 45-52). IEEE.

Nkosi, L., Tarwireyi, P, & Adigun, M. O (2013). *Insider Threat Detection Model for the Cloud*. In Information Security for South Africa (ISSA), 2013 (pp. 1-7). IEEE.

Park, J. S., & Ho, S. M. (2004). *Composite role-based monitoring (CRBM) for countering insider threats.* In Intelligence and Security Informatics (pp. 201-213). Springer Berlin Heidelberg.

Park, J. S., & Giordano, J. (2006). *Role-based profile analysis for scalable and accurate insider-anomaly detection*. In Performance, Computing, and Communications Conference, 2006. IPCCC 2006. 25th IEEE International (pp. 7-pp). IEEE.

Ramgovind, S., Eloff, M. M., & Smith, E. (2010). *The management of security in cloud computing*. In Information Security for South Africa (ISSA), 2010 (pp. 1-7). IEEE.

Rocha, F., & Correia, M. (2011). *Lucy in the sky without diamonds: Stealing confidential data in the cloud*. In Dependable Systems and Networks Workshops (DSN-W), 2011 IEEE/IFIP 41st International Conference on (pp. 129-134). IEEE.

Romanosky, S., Telang, R., & Acquisti, A. (2011*). Do data breach disclosure laws reduce identity theft?*. Journal of Policy Analysis and Management, *30*(2), 256-286.

Salem, M. B., Hershkop, S., & Stolfo, S. J. (2008). *A survey of insider attack detection research*. In Insider Attack and Cyber Security (pp. 69-90). Springer US.

Sandhu, R., Ferraiolo, D., & Kuhn, R. (2000). *The NIST model for role-based access control: towards a unified standard*. In ACM workshop on Role-based access control (Vol. 2000).

Sekar, R., Bowen, T. F., & Segal, M. E. (1999). *On Preventing Intrusions by Process Behaviour Monitoring*. In Workshop on intrusion detection and network monitoring (Vol. 1999, pp. 29-40).

Scalora, M., & Bulling, D.(2007) *Developing Threat Assessment Best Practice Standards: Leveraging Behavioural Science Strategies to Enhance Decision-Making, February 2007*. Open source literature review in partial fulfillment of Subcontract Number: MSMA-07-00001

Shenk, J. (2008). *Demanding more from log management systems*. SANS Institute Whitepaper.

Schultz, M. G., Eskin, E., Zadok, F., & Stolfo, S. J. (2001). *Data mining methods for detection of new malicious executables*. In Security and Privacy, 2001. S&P 2001. Proceedings. 2001 IEEE Symposium on (pp. 38-49). IEEE.

Sundararajan, S., Narayanan, H., Pavithran, V., Vorungati, K., & Achuthan, K. (2011). *Preventing Insider attacks in the Cloud*. In Advances in Computing and Communications (pp. 488-500). Springer Berlin Heidelberg.

Sultan, N. (2010). *Cloud computing for education: A new dawn*?. International Journal of Information Management, *30*(2), 109-116.

Syam Kumar, P., Subramanian, R., & Thamizh Selvam, D. (2010). *Ensuring data storage security in cloud computing using Sobol sequence.* In Parallel Distributed and Grid Computing (PDGC), 2010 1st International Conference on (pp. 217-222). IEEE.

Srinivasamurthy, S., & Liu, D. Q. (2010). *Survey on Cloud Computing Security*. In Proc. Conf. on Cloud Computing, CloudCom (Vol. 10).

Srikant, R., & Agrawal, R. (1996). *Mining sequential patterns:* Generalizations and performance improvements (pp. 1-17). Springer Berlin Heidelberg.

Srivastava, A., Sural, S., & Majumdar, A. K. (2006). *Database intrusion detection using weighted sequence mining*. Journal of Computers, 1(4), 8-17.

Zhang, Y, Szymanski, B. K., &. (2004). *Recursive data mining for masquerade detection and author identification*. In Information Assurance Workshop, 2004. Proceedings from the Fifth Annual IEEE SMC (pp. 424-431). IEEE.

Takabi, H., Joshi, J. B., & Ahn, G. J. (2010). *Securecloud: Towards a comprehensive security framework for cloud computing environments*. In Computer Software and

Applications Conference Workshops (COMPSACW), 2010 IEEE 34th Annual (pp. 393-398). IEEE.

Teng H., Chen K., Lu S. (1990). *Security Audit Trail Analysis Using Intrusive Generated Predictive Rules*. In Proceedings of the 11th IEEE National Conference on Artificial Intelligence Applications, March 1990, pages 24-29.

Theoharidou, M., Kokolakis, S., Karyda, M., & Kiountouzis, E. (2005). *The insider threat to information systems and the effectiveness of ISO17799*. Computers & Security, 24(6), 472-484.

Valdes, A., & Skinner, K. (2000). *Adaptive, model-based monitoring for cyber attack detection*. In Recent Advances in Intrusion Detection (pp. 80-93). Springer Berlin Heidelberg.

Wood, B. (2000). *An insider threat model for adversary simulation*. SRI International, Research on Mitigating the Insider Threat to Information Systems, 2, 1-3.

APPENDIX A

Rule learning Algorithm for Generating Sequential Rules

```
package ca.pfv.spmf.sequentialpatterns.prefixspan;

import java.util.ArrayList;

import java.util.List;

import java.util.HashMap;

public class Sequences { public final List<List<Sequence>> levels = new ArrayList<List<Sequence>>(); // itemset class par taille

   public int sequenceCount = 0;

   public double minConf = 0.0;

   private final String name;

   HashMap <String , Double> level4blast =new HashMap<String, Double>(); public void setMinConf(double min) {

      minConf = min;

   }  public double getMinConf() {

      return minConf;

   }  public Sequences(String name) {

      this.name = name;

      levels.add(new ArrayList<Sequence>());  }

   public void printSequencesFrequentes(int nbObject) {

      System.out.println(toString(nbObject));

   }  public List<List<Sequence>> lastTwoLevels(List<List<Sequence>> lvls) {

      List<List<Sequence>> LastTwoLevels = new ArrayList<List<Sequence>>();

      lvls = this.levels;

      int k = lvls.size();

      LastTwoLevels.add(lvls.get(k - 2));
```

```
    LastTwoLevels.add(lvls.get(k - 1));

    return LastTwoLevels;   }

public void SecondLastLevel(int nbObject, int m)

{     List<List<Sequence>> LevelB4Last = new ArrayList<List<Sequence>>();

//int k = levels.size();

    LevelB4Last.add(levels.get(m));

     for (List<Sequence> level : LevelB4Last) {

       for (Sequence sequence : level) {

        //String                    seq=                    "{"+sequence.getItemsets().get(0).toString().trim()+
"}{"+sequence.getItemsets().get(1).toString().trim()+"}{"+sequence.getItemsets().get(2).toString().tri
m()+"}";              String seq ="";

        for(int i=0;i<sequence.size();i++)

        {          seq=seq+"{"+sequence.getItemsets().get(i).toString().trim()+"}";          }

          double sup=Double.parseDouble(sequence.getRelativeSupportFormated(nbObject));

          if (( sup<1)&&( !LevelB4Last.contains(seq)))

          System.out.println(seq+ " "+ sup+ "  "+ LevelB4Last.contains(seq));

         level4blast.put(seq, sup);        }    } }

  public String toString2(int nbObject) {

    List<List<Sequence>> currentlevel = new ArrayList<List<Sequence>>();

   // int k = levels.size();       //LastLevel.add(levels.get(k - 1));

    //SecondLastLevel(nbObject);

    int lowerlevel = 5;

    StringBuffer print = new StringBuffer(200);       for(int k = lowerlevel; k <10; k++)

    {     currentlevel.add(levels.get(k));

    SecondLastLevel(nbObject, k - 1);

   // StringBuffer print = new StringBuffer(200);

   //print.append("level sjyhojshojdthjdfhjhjdfljhfdhj" +k);
```

```
        print.append("       \n");

        print.append("       \n");

        print.append("ItemSet ");

        print.append("       \t");

        print.append("       \t");

        print.append("       \t");

        print.append("       \t");

        print.append("Support ");       print.append("       \t");

        print.append("       \t");

        print.append("       \t");

        print.append("       \t");

        print.append("Confidence");

        print.append("       \n");

        print.append("");

        for (List<Sequence> level : currentlevel) {

           for (Sequence sequence : level) {

              String item2 = sequence.getItemsets().toString();

            String seq ="";

            for(int i=0;i<sequence.size()-1;i++)

            {           seq=seq+"{"+sequence.getItemsets().get(i).toString().trim()+"}";           }

//String                seq                =                "{"+sequence.getItemsets().get(0).toString().trim()+
    "}{"+sequence.getItemsets().get(1).toString().trim()+"}{"+sequence.getItemsets().get(2).toString().tri
    m()+"}";

         double sup2 = Double.parseDouble(sequence.getRelativeSupportFormated(nbObject));

         System.out.println(seq+level4blast.containsKey(seq));

              double sup1=level4blast.get(seq);
```

```
            if (sup2 < 1) {
                double conf = sup2 / sup1;

                print.append("       \n");
                print.append(item2);
                print.append("\t");
                print.append("\t");
                print.append(sup2);
                print.append("\t");
                print.append("\t");
                print.append(conf);
                print.append("\t");          }            }        }
    print.append("       \n");       }
    return print.toString();
}    public String toString(int nbObject)    {    StringBuffer r = new StringBuffer(200);
    r.append(" ----------");
    r.append(name);
    r.append(" -------\n");
    int levelCount = 0;
    int patternCount = 0;
    for (List<Sequence> level : levels) {

        r.append("  L");
        r.append(levelCount);
        r.append(" \n");
        for (Sequence sequence : level) {               patternCount++;
```

```
            r.append("  pattern ");
            r.append(patternCount);
            r.append(":  ");
              r.append("(");
            r.append(sequence.toString());
              r.append(") ");
            r.append("support :  ");
            r.append(sequence.getRelativeSupportFormated(nbObject));
           // r.append(" (");
            r.append(" ");
            r.append(sequence.getAbsoluteSupport());
            r.append('/');
           r.append(nbObject);
           r.append("\n");
            //r.append(") \n");
         }          levelCount++;       }
      r.append(" --------------------------------- Patterns count : ");
      r.append(sequenceCount);
      r.append(toString2(nbObject));
      return r.toString();    }
public void addSequence(Segment sequence, int k) {
      while (levels.size() <= k) {
         levels.add(new ArrayList<Segment>());       }
      levels.get(k).add(sequence);
      sequenceCount++;    }
  public List<Segment> getLevel(int index) {
```

```
        return levels.get(index);    }
  public int getLevelCount() {       return levels.size();   }
    public List<List<Sequence>> getLevels() {
              return levels;
    }   }
```

Matching Algorithm for Comparing (USER PROFILE and TESTING DATASET)

```
* @author lucky
 import org.apache.commons.lang.StringUtils;
import java.io.*;
import java.util.Scanner;
import java.net.*;


public class  Main
    public static void main(String[] args) throws FileNotFoundException, UnsupportedEncodingException
      {
        // Input files from with testing dataset and training set
        Scanner input = new Scanner(new FileReader(fileToPath("input.txt")));
       Scanner input2 = new Scanner(new FileReader(fileToPath("second.txt")));


//        Two arrays instantiated
String[][] array1 = new String[2000][6];
        String[][] array2 = new String [560][6];
        int i = 0;
              while (input.hasNextLine())
        {           String line = input.nextLine();
```

```
        String t = StringUtils.substringBetween(line, "[", "]");
        String[] z = StringUtils.split(t, ",");
        System.out.print("length" + z.length);
                for (int j = 0; j < z.length; j++)
        {               System.out.print(z[j]);
           array1[i][j] = z[j];
        }           System.out.println();
       i++;        }
     int a = 0;
     while (input2.hasNextLine())
     {           String line = input2.nextLine();
        String t = StringUtils.substringBetween(line, "(", ")");
       String[] z = StringUtils.split(t, ",");
        for (int j = 0; j < z.length; j++) {
           System.out.print("  " + z[j]);
           array2[a][j] = z[j];
        }           System.out.println();
       a++;        }
    int count = 0;
     for (int b = 0; b < array1.length; b++)
     {           boolean state = states(b, array1, array2);
        System.out.println("------------------------------" + state + "----------------------------------------");
                if (state)
        {               count++;           }       }
    System.out.println("Number of match :" + count);
}    public static String fileToPath(String filename) throws UnsupportedEncodingException
```

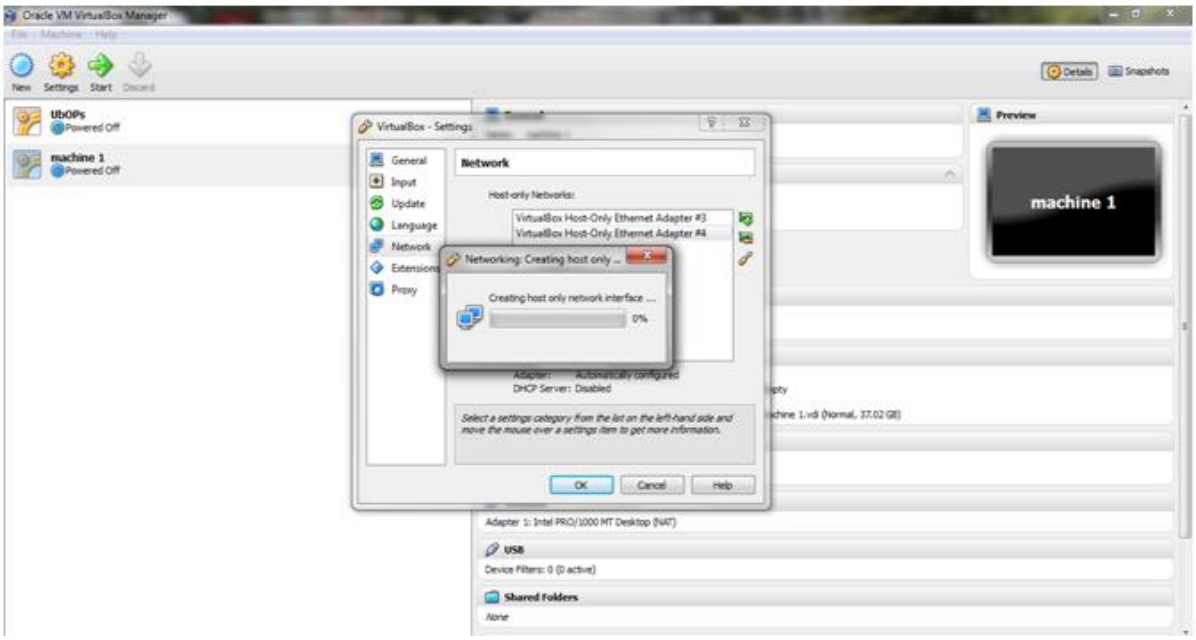
```
{       URL url = Main.class.getResource(filename);
   return java.net.URLDecoder.decode(url.getPath(), "UTF-8");
}    public static boolean states(int v, String array1[][], String array2[][])
{ boolean s = false;
          int count = 0;
          for (int k = 0; k < array2.length; k++)
   {            System.out.println("      ");
       count=0;
       for (int z = 0; z < array2[0].length; z++)
       {               System.out.print("[" + v + "][" + z + "] ");
          System.out.print("[" + k + "][" + z + "]");
          System.out.print("      ");
                      System.out.print(array1[v][z] + "--" + array2[k][z]);
                      if ((array1[v][z].trim()).equals(array2[k][z].trim()))
          {                   count++;              }
          System.out.println("count : " + count);
                      if (count == 6)               {                   return true;
          }           }       }       return s;    }}
```

APPENDIX B

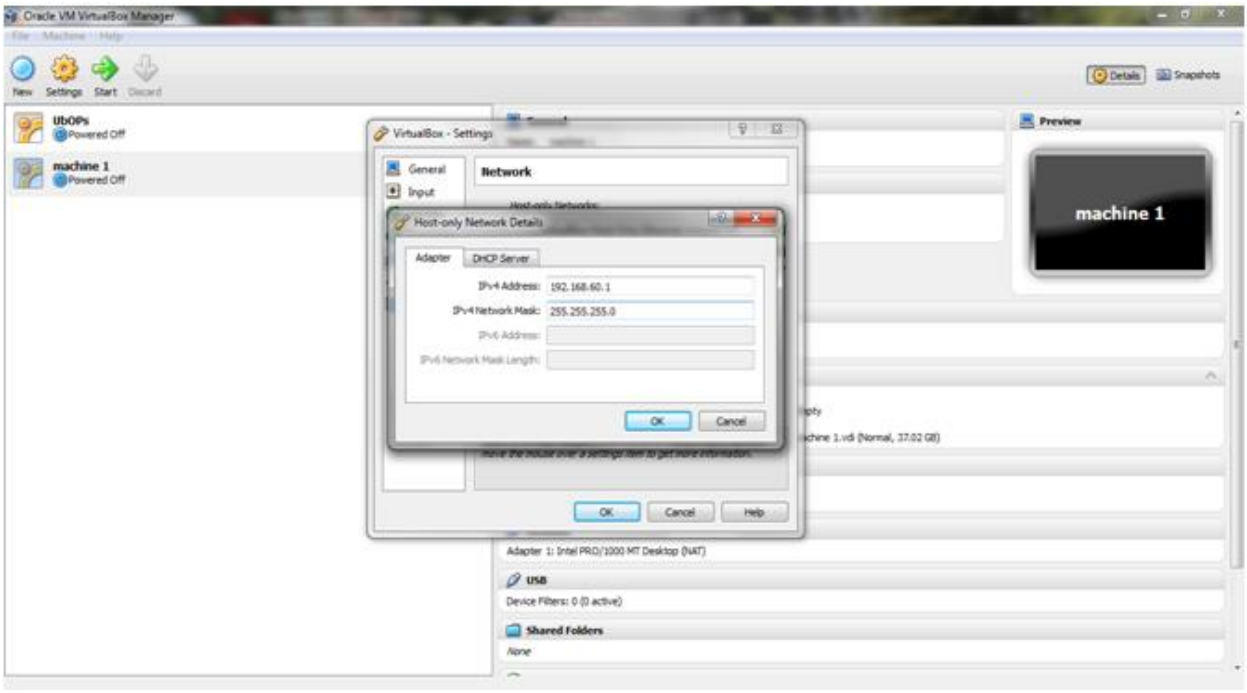
Steps involved in setting up virtual networks

Step 1: Start virtualbox, after starting virtualbox then go to step 2

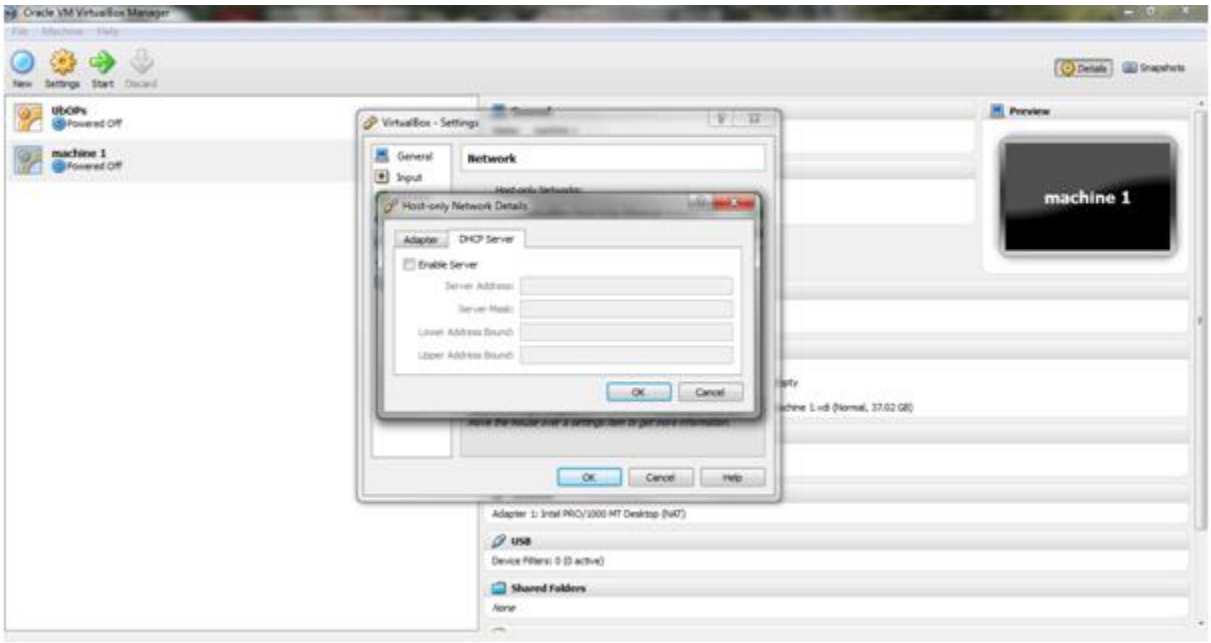
Step 2: File then **Preference** select **Network** option



Illustrating the creation of Host Only network and adding vboxnet



Showing setup of IPV4



setting the dhcp server

Configuring database to all incoming request in the clients

Sed –I ‘s/127.0.0.1/0.0.0.0/g’ /etc/mysql/my.cnf

This command help in configuring mysql database .

Configuring Keystone

Mysql –u root –p

Once keystone is configured, user can be added in the database with certain roles and level of access.

Installing glance

Before new user can be added, image storage service must be install to allow the smooth running

Apt-get install glance

This command install glance with it all necessary properties.

Installing quantum server

Apt-get install quantum server quantum-plugin-opevswitch

This command install the quantum server

Dashboard installation

Apt-get install openstack-dashboard memcached

This command installs the web interface that is customizable.