MODELLING AND MITIGATING MINOR-THREATS IN NETWORK THREAT MANAGEMENT

OLUWAFEMI ORIOLA

(MATRIC NO: 146284)

JULY, 2015

MODELLING AND MITIGATING MINOR-THREATS IN NETWORK THREAT MANAGEMENT

BY

OLUWAFEMI ORIOLA

(MATRIC NO: 146284)

B.Sc. (Hons.) Computer Science (Akungba), M.Sc. Computer Science (Ibadan)

A Thesis in the Department of Computer Science

Submitted to the Faculty of Science

In partial fulfillment of the requirement for the Degree of

DOCTOR OF PHILOSOPHY

of the

UNIVERSITY OF IBADAN

JULY, 2015

CERTIFICATION

This is to certify that this research work was carried out by ORIOLA OLUWAFEMI with matriculation number 146284 in the Department of Computer Science, Faculty of Science, University of Ibadan, Ibadan, Nigeria.

……………………..

Dr. A.B. Adeyemo
B. Sc. (Ife), PGD, M.Tech, Ph.D. (Akure) Computer Science
Supervisor

……………………..

Dr. O. Osunade
B.Sc. (Ife), M.Sc., Ph.D. (Ibadan) Computer Science
Co-Supervisor

……………………....................................

Dr. A.B. Adeyemo
B. Sc. (Ife), PGD, M.Tech, Ph.D. (Akure) Computer Science
Acting Head
Department of Computer Science
University of Ibadan, Ibadan

CERTIFICATION

This is to certify that this research work was carried out by ORIOLA OLUWAFEMI with matriculation number 146284 in the Department of Computer Science, Faculty of Science, University of Ibadan, Ibadan, Nigeria.

……………………………………………………

Dr. A.B. Adeyemo
B. Sc. (Ife), PGD, M.Tech, Ph.D. (Akure) Computer Science
Supervisor

……………………………………………………

Dr. O. Osunade
B.Sc. (Ife), M.Sc., Ph.D. (Ibadan) Computer Science
Co-Supervisor

……………………………………………………

Dr. A.B. Adeyemo
B. Sc. (Ife), PGD, M.Tech, Ph.D. (Akure) Computer Science
Acting Head
Department of Computer Science
University of Ibadan, Ibadan

DEDICATION

I dedicate this work to the Almighty God who has given me the grace to achieve this feat. May His name be praised.

ACKNOWLEDGEMENT

Honour, majesty and praise be to the Almighty God, my maker and sustainer, a true potentate. The whole journey started through Him; He commissioned the beginning of the programme particularly this research. When progress looked impossible and the road seemed blocked, He made a way.

Also worthy to appreciate is my amiable supervisor, Dr Adesesan Barnabas Adeyemo. He did not only supervise this work by giving moral and intellectual support but his actions, advice and love were fatherly. In the same wise, I express my sincere appreciation to Dr Oluwaseyitanfunmi Osunade, my co-supervisor. You have been of great encouragement to me. I also give kudos to Dr 'Lekan Akinola, the Postgraduate Coordinator in the Department of Computer Science. To my 'departmental mother', Dr Omowumi Adeyemo and all other lecturers in the Department of Computer Science, thank you so much for your mentorship.

I express my gratitude to the research team at the Centre for Security, Communication and Networks Research, Plymouth University, United Kingdom who contributed immensely to this work. In particular, I appreciate Dr Maria Papadaki for given up time out of her busy schedule to supervise me as a visiting research student for about four months. I thank Associate Professor Bogdan Ghitta and the Head of School of Computing and Mathematics, Professor Steve Furnell for the assistance in carrying out the experiment.

I also express my gratitude to my parents, Mr Kolawole Oriola and Mrs Beatrice Oriola for their moral, financial and spiritual support. Appreciation also goes to my siblings: Deji Oriola, Titilayo Oriola-Ojo, Gbenga Oriola and my loving wife, Titilope Oluwaseun Oriola.

I appreciate Mr Sola Kudehinbu and Mr Gabriel Olatunbosun, who assisted in actualising this dream from the beginning; your aspiration in life will be fulfilled.

I want to end this section by appreciating God once again. He lives!

TABLE OF CONTENTS

APPENDICES

LIST OF FIGURES

Figures

LIST OF TABLES

Tables

ABSTRACT

Network Threat Management (NTM) is used to model and mitigate network threats classified as major-threats and minor-threats without exceeding Cost of Detection (CD), Time of Detection (TD) and False Positive Rate (FPR) limits. Existing network threat modelling and mitigation frameworks focused on major-threats because until recently, only major-threats are usually harmful, while minor-threats were perceived non-harmful. Recent studies however have shown that some minor-threats are harmful. This study was designed to model and mitigate minor-threats in NTM.

The Threat Prediction Model (TPDM) and Threat Prioritisation Model (TPRM) were used for modelling while Threat Mitigation Model (TMTM) was used for mitigation. The TPDM was modified to identify minor-threats by incorporating actionable attributes. The modified TPDM accuracy was compared with TPDM based on confidence, with 1.0 benchmark. The TPRM was modified to rate minor-threats using Dempster-Shafer Method and compared with snort-classifier and Common Vulnerability Scoring System (CVSS) as standards. The rating range between 0 and 5 was 'less harmful' while rating above 5 was 'moderately harmful'. The modified TPDM and TPRM were implemented using java. The TMTM was modified using Hillson's risk mitigation model. The CD based on number of rules, TD and FPR were used to compare modified TMTM and TMTM for snort and suricata implementations. Real life minor-threats known as Plymouth University Advanced Persistent Threats (PUAPT) were developed using metasploit for analysis. Existing Lincoln Lab Denial of Service (LLDOS) minor-threats were also analysed for standardisation. The CD, TD and FPR limits for PUAPT analysis were set at 5_rules, 60_seconds and 25% respectively while LLDOS were 5_rules, 90_seconds and 25%. Data were analysed using descriptive statistics.

In PUAPT analysis, modified TPDM was accurate with confidence of 1.0 compared to 0.0 of existing TPDM. The modified TPRM rated harmful minor-threats as moderately harmful while non-harmful as less harmful. The snort-classifier rated both harmful and non-harmful minor-threats as less harmful while CVSS rated none of the minor-threats. With modified TMTM for snort implementation, CD, TD and FPR of

5_rules, 1_second and 2.7% respectively were incurred compared to 19082_rules, 240_seconds and 99.1% of existing TMTM. With modified TMTM for suricata implementation, CD, TD and FPR of 5_rules, 1_second and 1.2% respectively were incurred compared to 18701_rules, 240_seconds and 99.8% of existing TMTM. The modified TPDM for LLDOS was accurate with confidence of 1.0 compared to 0.1 of existing TPDM. The modified TPRM rated harmful minor-threats as moderately harmful while non-harmful as less harmful, snort-classifier rated both harmful and non-harmful minor-threats as less harmful and CVSS rated only minor-threats with vulnerabilities. With modified TMTM for snort implementation, CD, TD and FPR of 5_rules, 3_seconds and 21.1% respectively were incurred compared to 19082_rules, 480_seconds and 99.9% of existing TMTM. With modified TMTM for suricata implementation, CD, TD and FPR of 5_rules, 75_seconds and 1.3% respectively were incurred compared to 18701_rules, 480_seconds and 99.0% of existing TMTM.

The modified models accurately modelled and mitigated minor-threats without exceeding cost of detection, time of detection and false positive rate limits. The modified models are recommended for modelling and mitigating minor-threats in network threat management.

Keywords: Network threat management, Minor-threat, Threat modelling, Threat mitigation.

Word count: 500

CHAPTER ONE

INTRODUCTION

1.1 Background of the Study

Internet is one of the greatest innovations that has benefitted human race since the nineteenth century. It has eliminated the boundary among groups in the societies. Now, it can be accessed everywhere via web, phones or cloud. In particular, hackers have been using internet media to access unauthorised resources on the internet. A hacker or attacker could steal confidential information or commit financial fraud through the internet. Some of the methods employed include phishing, masquerading, spoofing and crypto-analysis.

Apart from the individual usefulness of internet in causing harms to resources online, groups of users do benefit from it. Some Attackers collaborate and cooperate via internet to exploit the vulnerability of victim systems and cause damage in Organised manner. According to Global Agenda Council on Organised Crime Report for 2011/2012 in Weforum (2012) and Internet Organised Crime Threat Assessment Release for 2014 inEuropol (2014), these kinds of threats are referred to as Internet-facilitated Organised Crime Threats and they make use of three common methods in achieving their missions: Botnet, Worm Propagation and Advanced Persistent Threats (APT).

Botnet otherwise known as zombies are set of interconnected computers that could attack a single host or multiple hosts (Banday *et al.*, 2009). They may be Organised as Server-Client or Peer-to-Peer computers. Each of the computers is known as bots and nowadays, distributed hosts are being taken over as slaves without authorisation by master remotely in order to improve complexity and sophistication of their exploits. The Worm is a malicious program that could replicate itself to damage other useful programs in single host, different hosts or multiple networks with various manifestations (CAIDA, 2003). Initially, APT was used to describe nation-states stealing of data or damage to other nation-states for strategic gain. But the definition has now been expanded by security vendors and media to include similar attacks carried out by cybercriminals stealing data from businesses for profit (Websense, 2011). It is 'Advanced' because it is targeted and sophisticated and 'Persistent' because it usually continues over some period until the aim is achieved.

The emergence of these Internet-facilitated Organised Crime Threats has increased the violation of network security policies, disruption of assets' services and loss of assets. Symantec (2012) and Symantec (2013) reported that majority of these threats are discovered in large organisations. According to Symantec (2013), only thirty one per cent (31%) of the Internet Threats were targeted at organisations with less than two hundred and fifty personnel in 2012. It was also reported that a single Threat was discovered in 2011 to have infected six hundred thousand (600,000) mac machines in 2012. Arbor Networks (2012) reported that *distributed denial of service* (DDoS) caused by Bots was the most perpetrated between the period of October 2010 and September 2011. Kaspersky (2009) identified about fifteen million unique samples of malware specimens in 2009, which means that one unknown sample was discovered roughly every two seconds. This high level of occurrence and distribution might be attributed to Internet-facilitated Organised Crime Threats. In the reports by Symantec (2013), the insurgence and sophistication of these threats have also been justified. It was reported that as at 16th March 2011, approximately 88.2 per cent of all spam was distributed by spam-sending botnets. Also Worms (including viruses) were accounted for more than 70 per cent of the malicious codes discovered in 2012. In recent years, perpetrations of Advanced Persistent Threats have continued to increase. For example, Malwares such as Stuxnet, Duqu, and Flamer & Disttrack in 2010, 2011 and 2012 respectively have persistently showed high levels of sophistication and danger.

In Risk Management, threats are quantified by their severity (potential harm) and the likelihood of experiencing an incident within a given timeframe (Computer Economics, 2009). The risk of this threats could be significant or less-significant, critical or less-critical (Jumaat, 2012). Reconnaisance, scanning and public user-level threats have been examined as less significant and less critical compared to super-user level, malware threats and denial of service; however, the less could inflict serious harm on service delivery by causing denial of service (DoS). The concept of categorizing threats into Major and Minor Threats was introduced by Symantec Corporation in Symantec (2005), where the reported significant threats were referred to as Major Threats while the less significant ones were tagged the Minor Threats. In the reports, the Major Threats pose great risks to organisations while the Minor Threats pose lesser risks to organisations. Recently, most of the Denial of Service (DoS) and Distributed Denial of Service (DDoS) attacks have been linked to udp flood, icmp (ping) flood and syn flood (Incapsula, 2014), which are in the Minor Threats category. These threats are categorised or assigned to low priority groups in Caswell and Roesch (1998), Snorby (2011), Porras *et al.* (2002), Albushi *et al.* (2009) and Jumaat (2012). Often, Minor Threats require little effort to be carried out while Major Threats require much effort. Hence, attacker failure in gaining super-user level access may force them to exploit denial of service, which may be disruptive in critical system (Wang and Zhao, 2006). Most lower attack stages constitute Minor Threats while higher level attack stages constitute Minor Threats.

In critical Network Threat Management Systems, multiple Information Security sensors are required to combat the influx of the Internet-facilitated Organised Crime Threats. Staniford *et al.* (2002) and Cardenas *et al.* (2004) showed that the aggregation of evidences from different sensors and clients would lead to efficient and accurate detection of DDoS and worm attacks. Even though the techniques were effective in those scenarios, huge costs were incurred in managing the threats. Hence, only the supposedly harmful threats referred to as ‘Major Threats’ were mitigated while the perceived less-harmful threats known as ‘Minor Threats’ were often accepted.

According to Ntouskas *et al.* (2011), Security Management is a continuous and systematic process of identifying, analysing, handling, reporting and monitoring operational risk of an organisation. And for these processes to be achieved, Network Threat Management Practices are required to provide a manageable Enterprise Security System (Scott, 2002). The base component of a good Network Threat Management is Threat Modelling which is: “a systematic, non-provable, internally consistent method of modelling a system, enumerating risks against it, and prioritising them” (SensePost, 2011). It involves steps such as identification of critical assets, decomposition of the system to be assessed, identification of possible points of attack (vulnerability), identification of threats, categorization and prioritisation of the threats, and mitigation of threats (Olzak, 2006). But, Network Threat Management and Threat Modelling have not been explored for the purpose of identifying, prioritising and mitigating Minor Threats. The motivations below underscore the need for this research.

1.2 Research Motivations

The first motivation for this research is that existing works on Threat Modelling focused on Mitigating Major Threats. The harmful Minor Threats’ risks are *accepted* instead of being *mitigated* during Threat Mitigation. This is as a result of modelling of entire threats, which caused biased prioritisation of minor threats, which are harmful. M-correlator by Porras *et al.* (2002) classified probe and suspicious usage as Minor Threats, Fuzzy-Met by Alshubi *et al.* (2009) classified reconnaissance, scanning and SQL overflow as Minor Threats, Incident Prioritisation by Jumaat(2012) classified snmp public access udp as Minor Threats and Snort by Caswell and Roesh (1998) classified icmp events and network scan as Minor Threats. Based on the fact that the values of threat likelihood and consequence for Major Threats were usually higher than Minor Threats, it might be appropriate to model Minor Threats separately from the Major Threats. However, the type of variables to determine the level of risk of Minor Threats could be different from Major Threats. Therefore, this work focuses on modelling of Minor Threats using strategically selected risk determination factors that rely on asset, defence and attacks. Also, the major-threat mitigation function of Risk Mitigation Model might need to be extended to cater for significantly harmful minor threats without affecting the scope of the Network Threat Management. The existing Risk Mitigation Model is adapted to mitigate significantly harmful Minor Threats.

Another motivation for the research is drawn from the fact that the localized approach to Threat Modelling cannot sustain security in the continuously emerging dynamic threats' world most especially in developing world. In Nigeria with very scarce expertise knowledge and high internet penetration market for instance, which contributed about ten per cent of World Internet Threats as early as the period between 2006 and 2008 according to Federal Bureau of Investigation: Internet Crime Complaint Centre report presented in Doyle (2010), the usefulness of Collaboration cannot be overemphasized. Doyle's report had it that phishing, identity theft and file damage, which are now popularly perpetrated by bots, worms and APTs in advanced nations were the exploits usually used by Nigeria cyber-criminals. Because of high availability of exploit tools and high internet penetration rate in Nigeria, the sophisticated threats might soon be used by Nigeria Cybercriminals to perpetrate their attacks. Hence, this thesis focuses on Threat Modelling from global perspective. Recently, Chen *et al.* (2011) and Chen *et al.* (2013) have applied Unified Threat Management System in Collaborative Network Security Management with specific focus on Forensic Analysis using a Cloud-based Security Centre for in-depth analysis of attack. The studies showed that Collaborative Network Security Management is reliable for in-depth analysis of threat. Therefore, Collaborative Network Security Management System is used as model for the global analysis.

Moreover, the fact that Internet-facilitated Organised Crime Threats have accounted for most of the attacks in organisation and continued to increase despite advances in Threat Analysis and continuous platform upgrades suggest that a new reputable approach must be applied to mitigate the threats with specific focus on in-depth analysis of threat and cost-effective management of threat. Attack-centric Threat modelling have been studied in RVA (2010), Danta *et al.* (2007), Bhattacharya *et al.*, (2008), Ritchey and Amman (2000) & Sheyner *et al.* (2002); Defence-centric Threat

Modelling in Killourhy *et al. (*2004); and Asset-centric Threat Modelling in Common Vulnerability Scoring System by Mell et al. (2009). Only few works such as Ha *et al.* (2006), Hasan and Myagmar (2005) & McHugh *et al.* (2001) have suggested Hybrid-centric Threat Modelling but no research has explored it for Minor Threat modelling. Hence, this research applies Hybrid-centric Threat Modelling in modelling Minor Threats because of its ability to ensure in-depth analysis of Threats.

In a typical organisation operating over the internet, the Demilitarized Zone (DMZ) is protected by limited security configurations while the inside zone is protected with more effective security configurations (Paquet, 2013). A successful compromise of security in the DMZ could lead to security compromise in the inside zone. Also, the vulnerability of one asset could lead to the vulnerability of other assets in the same zone or different zones depending on the configuration. This means that attack in organisations could be better described as multistage phenomenon (Gomez, 2011) instead of single stage phenomenon. This means that a Network Security domain is violated by multiple stages of attack referred to as Scenario Attack while a Collaborative Network Security domain has potential of being violated by Complex Scenario Attack. However, the existing Threat Identification Models have been focused on static and single attack recognition, static and multistage attack recognition as well as dynamic and single scenario attack pattern recognition. They lack the ability to predict complex scenario attack patterns with high degree of accuracy, which this study proposes to address.

The following studies on Collaborative Threat Analytics: Cyber Threat Intelligence CIF (2009) reported in Moriarty (2013); Collaborative Intrusion Detection by Chen *et al.* (2007); Peng *et al.* (2007) and Chen and Malin (2011); and Distributed InfoSec Alert Management by Porras *et al.* (2002) and Alsubhi *et al.* (2009) have shown that Incident Sharing and Analysis suffer from privacy, interoperation, quality, uncertainty, multidimensionality and distrust issues. Hence, such problems, which might affect the proposed Collaborative Network Security Management framework are addressed in this work.

A review has been carried out and the main problem to be addressed in this study is: *"How can Minor Threats in the context of Internet-facilitated Organised Crime Threats be modelled and mitigated in Network Threat Management aspect of Collaborative Network Security Management without the influence of the problems of Incident Sharing and Analysis?"*

This is presented in the following posers as:

a. How can actionable Threat Paths of Internet-facilitated Organised Crime Threats be predicted with highest possible accuracy without bias in the context of Collaborative Network Security Management?

b. How can Minor Threats from Internet-facilitated Organised Crime Threats be prioritised without bias in the context of Collaborative Network Security Management?

c. How do we integrate the Minor Threat Mitigation with the Traditional Network Threat Mitigation Frameworks in Network Threat Management without compromising its scope or requirements?

d. How can privacy, distrust, interoperation, quality, multidimensionality and uncertainty problems associated with Incident Sharing and Analysis in Collaborative Network Security Management be avoided?

1.3 Research Aim and Objectives

The aim of this research work is to model Minor Threats for the purpose of mitigating harmful Minor Threats, without adverse effect on the scope of Network Threat Management, where the threats are Internet-facilitated Organised Crime Threats and the Network Threat Management are carried out from the perspective of Collaborative Network Security Management without the effect of Privacy, Interoperability, Quality, Trust, Multidimensionality and Uncertainty Issues. The specific objectives of the research are:

a. Conceptualisation of Collaborative Network Security Management System Framework for Event Sharing, Analysis and Security Configuration.

b. Development of a Prediction, Prioritisation and Mitigation Models for Modelling and Mitigating Minor Threats.

c. Design and Implementation of Prototypes of Minor Threat Modelling and Mitigation Tools.

d. Development of Internet-facilitated Organised Crime Threats and Collaborative Network Security Management System.

e. Evaluation of the Performance of Modelling and Mitigation Models.

1.4 Organisation of Thesis

In Chapter One, background of the study, motivations and research problems are raised. The Aim and Objectives are presented.

Chapter two discusses Network Threats and Minor Threats. Network Security Management Systems are reviewed. Strategic review of Collaborative Network Security Management Systems with emphasis on Network Threat Management is done. Remarks on research direction were given.

Chapter three presents both conceptualisation of Collaborative Network Security Management Framework and Predictive Modelling techniques for Minor Threat Modelling and Mitigation. The modelling tools, experimental designs and performance evaluation metrics are also presented.

Chapter four presents the results of Threat Prediction, Threat Prioritisation and Threat Mitigation Models. It also presents the Comparison of the Models with existing models and discusses them.

In Chapter five, the thesis is completed by presenting the summary, conclusion, postulations, contribution to knowledge and recommendations for future research.

CHAPTER TWO

LITERATURE REVIEW

2.1 Internet-facilitated Threats

There are different types of threats and they pursue different goals (Michalski *et al.*, 2012). Internet Threats are the threats that are associated with internet environments. They are mostly discovered in organisations that make use of internet in their day-to-day business activities (Farnham, 2013), otherwise referred to as Internet-dependent Organisations. The forms of threats reported to have compromised the security of IS resources include Malware, Denial of Service, Cross Site Scripting, SQL Injection, Probe, Spoofing, Bufferoverflow, Distributed Denial of Service, Botnet, Worm, Advanced Persistent Threats, Spam and Phishing.

2.1.1 Internet-facilitated Organised Crime Threats

Internet-facilitated Organised Crime Threats refers to threats that are perpetrated via internet media. These Threats are explored via web, mobile, or cloud in multistage and coordinated manner. Examples of such Threats are Botnet, Worm and Advanced Persistent Threat. They often involve many simple attacks or complex attack scenario (Wang and Zhao, 2006) and do inflict more hazards on IS because they are always targeted. In the next sections, the categories of the threats are discussed with their modes of operation.

2.1.1.1 Botnet

According to Banday *et al.* (2009), the term bot, derived from “ro-bot” in its generic form is used to describe a script or set of scripts or a program designed to perform predefined functions repeatedly and automatically after being triggered intentionally or through a system infection. Although bots originated as a useful feature for carrying out repetitive and time consuming operations but they are being exploited for malicious intent. Bots that are used to carry out legitimate activities in an automated manner are called benevolent bots and those that are meant for malicious intent are known as malicious bots.

Benevolent bots among various other activities are used by search engines to spider online website content and by online games to provide virtual opponent. The first bot program Eggdrop created by Jeff Fisher in 1993 originated as a useful feature on Internet Relay Chat (IRC) for text-based conferencing on many machines in a distributed fashion. In a typical IRC setup, a user running an IRC client program connects to an IRC server in an IRC network. An IRC host computer running an IRC bot malware program becomes a Zombie or a drone.

The first malicious IRC bot, Pretty Park Worm that appeared in 1999 contained a limited set of functionality and features, such as the ability to connect to a remote IRC server, retrieve basic system information e.g. operating system version, login names, email addresses, etc.

Bot malware uses FTP, TFTP, HTTP protocol based services to infect computers and spread it until a desired strength of botnet is assembled. Botnets are also created by other botnets called seed botnets. In this section we discuss three different categories of command and control techniques; namely: centralized, peer to peer and random.

a. Centralized Command & Control (C&C) Technique

This C&C technique uses a central high bandwidth host called C&C server to forward messages between various bots. The C&C server in a botnet is a compromised computer that runs certain network services like IRC, HTTP, etc and which rallies the commands issued by the botmaster to each host in the botnet that join the C&C server channel. Many bots including *AgoBot*, *RBot*, *SDBot*, *SpamThru* and *Zotob* use this C&C technique.

b. P2P Command & Control (C&C) Technique

The peer to peer C&C technique uses P2P communication with no real central server to forward messages between botnets which makes it more resilient to failures in the network. Unlike centralized C&C technique, P2P C&C technique is much harder to discover and destroy; even if one or more bots are neutralized, the botnet still continues to operate. Further, an anonymous P2P technique may be used to make it even more difficult to detect. However the botnet size supported by P2P systems is generally very low in comparison to centralized systems, which makes profit oriented botmasters to avoid using P2P technique. Also the propagation latency and guaranteed message delivery is lacking in P2P systems. Some examples of botnets that use P2P C&C technique include *Phatbot* and *Sinit*.

c. Random Command & Control (C&C) Technique

The idea of random C&C technique has been presented by Evan Cooke, however no botnet has been reported to have used this C&C technique. In this C&C technique no bot can know about the existence of more than one other bot thus making the detection of the botnet very difficult. A botmaster or a bot can send an encrypted message randomly which may be intercepted by other bot and a conversation could begin. In this command and control technique message latency is very high, however; unlike other command and control techniques it lags guaranteed message delivery.

2.1.1.2 Worm

A computer worm is a program that self-propagates across a network exploiting security or policy flaws while computer virus requires some sort of user action to abet their propagation (CAIDA, 2003).

Computer worms primarily replicate on networks, but they represent a subclass of computer viruses. Interestingly enough, even in security research communities, many people imply that computer worms are dramatically different from computer viruses. In fact, even within CARO (Computer Antivirus Researchers Organisation), researchers do not share a common view about what exactly can be classified as a "worm." We wish to share a common view, but well, at least it is agreed that all computer worms are ultimately viruses.

The network-oriented infection strategy is indeed a primary difference between viruses and computer worms. Moreover, worms usually do not need to infect files but propagate as standalone programs. Additionally, several worms can take control of remote systems without any help from the users, usually exploiting a vulnerability or set of vulnerabilities. These usual characteristics of computer worms, however, do not always hold. Worms usually spread through internet. A popular example of worm is slammer worm discovered in 2003, which performed fifty five million scan per seconds, infecting 74,855 hosts (CAIDA, 2003). Figure 2.1 presents the map showing the spread of Slammer in 30Minutes.

2.1.1.3 Advanced Persistent Threats (APT)

APT: A buzzword or an imminent threat? Advanced Persistent Threats (APTs) have become a major concern for IT security professionals around the world, and for good reason. Recent attacks targeting Canadian government officials, French government officials, RSA, and elements of the European Union have all been linked to APTs. But what exactly is an APT? Too much hype has clouded the facts surrounding a very real danger for organisations *of all sizes*. The following are the characteristics of APT:

a. *Targeted:* APTs target specific organisations with the purpose of stealing specific data or causing specific damage. This stands in direct contrast to most historical malware, which wreaks havoc on any randomly infected system. First, any organisation, large or small, with valuable data is subject to APT methods. Second, the more valuable your data, the more likely you are to be targeted. The cybercrime economy is well Organised and funded, with attackers investing more to achieve bigger paybacks.

b. *Persistent:* APTs play out in multiple phases over a long period of time. Prior to the actual attack, attackers only know the target organisation and objective. They do not know where their target data resides, what security controls are in place, or what vulnerabilities exist that might be exploited. To steal the data, the attacker must identify vulnerabilities, evaluate existing security controls, gain access to privileged hosts within the target network, find target data, and finally, extract data from the network. The entire process may take months or even years. The lesson here is that attack detection cannot rely on any single event, but should look for patterns of events that are characteristic of APT methodologies.

c. *Evasive:* APTs are systematically designed to evade the traditional security products that most organisations have relied on for years. For example:

i. To gain access to hosts within the target network while avoiding network firewalls, the attacker delivers threats within *content* carried over commonly allowed protocols (http, https, smtp, etc.).

ii. To install malware on privileged hosts while avoiding antivirus programs, the attacker writes code designed for the specific target environment. This code has never been seen before and therefore, no AV signatures exist to provide protection.

iii. To send data out of the target network, while again avoiding firewalls, the attacker uses custom encryption and tunnels content within protocols that are allowed outbound by the firewall.

iv. Complex: APTs apply a complex mix of attack methods targeting multiple vulnerabilities identified within the organisation. A given APT may involve 1) telephone-based social engineering to identify key individuals within the target organisation, 2) phishing emails sent to those key individuals with links to a website that executes custom JavaScript code to install a remote access tool, 3) binary command-and-control code (either custom code or code generated by commonly available malware kits) and, 4) custom encryption technology. Clearly, no single security control provides coverage against all of these vectors. Any successful APT defense strategy must take

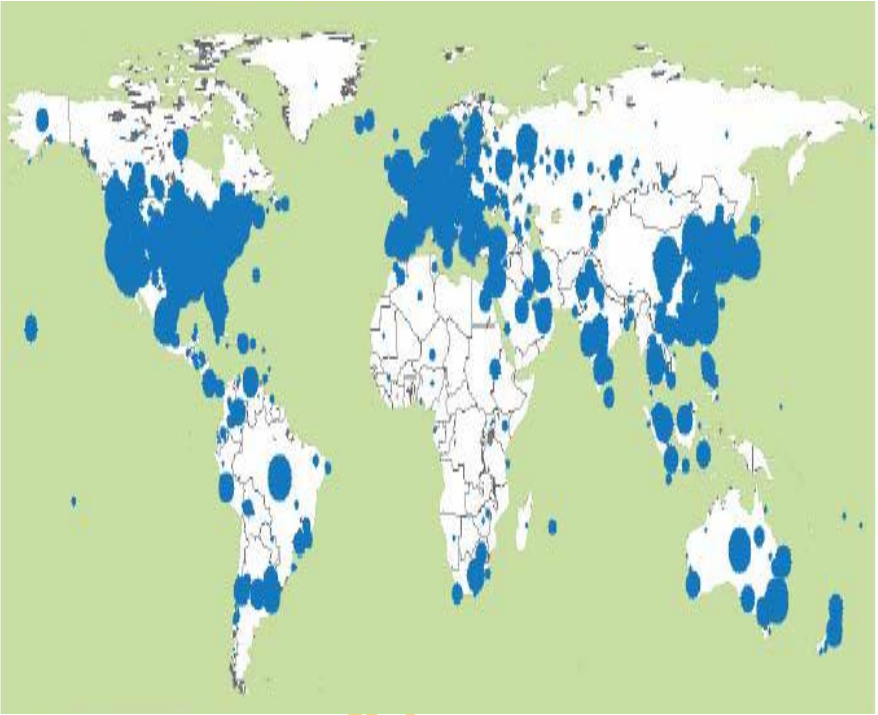


Figure 2.1: Map showing the spread of Slammer Worm within 30minutes in 2003 (www.caida.org)

a multi-layered approach in which multiple detection mechanisms work together to identify complex patterns of evasive behaviour.

The APT process includes three major phases that occur over a period of months.

i. *Phase 1 - Reconnaissance, Launch, and Infect:* The attacker performs reconnaissance, identifies vulnerabilities, launches the attack, and infects target hosts.

ii. *Phase 2 - Control, Update, Discover, Persist*: The attacker controls infected hosts, updates code, spreads to other machines, and discovers and collects target data.

iii. *Phase 3 - Extract and Take Action*: The attacker extracts data from the target network and takes action.

2.2 Threat Phases

An Attack-focused Threat usually inflicts great and mild damage on enterprise systems. Generally, attack phases inflict different levels of damage in these regards.

Snort by Caswell and Roesh (1998) has attacks classified into four priority groups in the default priority setting. A priority of 1 (high) is the most severe and 4 (very low) is the least severe. Only one attack is classified into the very low priority group. Table 2.1 presents snort's threat classification by attack priority.

Another popular tool, Common Vulnerability Scoring System (CVSS, 2014) ranks risk between 0 and 10. It can qualitatively described risk as low, medium and high. CVSS is a relatively new approach used to quantitatively analyse vulnerabilities. It is a product of the National Infrastructure Advisory Council (NIAC) effort to introduce an open standard for vulnerability scoring.

The CVSS approach is based on three basic metric groups defined as follows:

Base Metric group b: This defines the characteristics of some aspects of a vulnerability that do not change with time, nor in different target environments.

These characteristics are as follows

Table 2.1: Threat Classification by Attack Priority

S/N	**Class Type**	**Description**	**Priority**
1	attempted-admin	Attempted Administrator Privilege Gain	high
2	attempted-user	Attempted User Privilege Gain	high
3	inappropriate-content	Inappropriate Content was Detected	high
4	policy-violation	Potential Corporate Privacy Violation	high
5	shellcode-detect	Executable code was detected	high
6	successful-admin	Successful Administrator Privilege Gain	high
7	successful-user	Successful User Privilege Gain	high
	trojan-activity	A Network Trojan was detected	high
8	unsuccessful-user	Unsuccessful User Privilege Gain	high
9	web-application-attack	Web Application Attack	high
10	attempted-dos	Attempted Denial of Service	medium
11	attempted-recon	Attempted Information Leak	medium
12	bad-unknown	Potentially Bad Traffic	medium
13	default-login-attempt	Attempt to login by a default username and password	medium
14	denial-of-service	Detection of a Denial of Service Attack	medium
15	misc-attack	Misc Attack	medium
16	non-standard-protocol	Detection of a non-standard protocol or event	medium
17	rpc-portmap-decode	Decode of an RPC Query	medium
18	successful-dos	Denial of Service	medium
19	successful-recon-largescale	Large Scale Information Leak	medium
20	successful-recon-limited	Information Leak	medium
21	suspicious-filename-detect	A suspicious filename was detected	medium
22	suspicious-login	An attempted login using a suspicious	medium

		username was detected	
23	system-call-detect	A system call was detected	medium
24	unusual-client-port-connection	A client was using an unusual port	medium
25	web-application-activity	Access to a potentially vulnerable web application	medium
26	icmp-event	Generic ICMP event	low
27	misc-activity	Misc activity	low
28	network-scan	Detection of a Network Scan	low
29	not-suspicious	Not Suspicious Traffic	low
30	protocol-command-decode	Generic Protocol Command Decode	low
31	string-detect	A suspicious string was detected	low
32	Unknown	Unknown Traffic	low
33	tcp-connection	A TCP connection was detected	very low

(a) Confidentiality impact (CI) metric measures the impact on confidentiality of a successful exploit of the vulnerability on the target asset. The possible scores for this metric are as follows:

i. none: No impact

ii. partial: There is significant informational disclosure

iii. complete: A total compromise of critical system information

(b) Integrity impact (II) metric measures the impact on integrity a successful exploit of a vulnerability will have on the target asset. The possible scores for this metric are as follows:

i. none: No impact

ii. partial : Significant breach of integrity

iii. complete: A total compromise of system integrity

(c) Availability impact (AI) metric measures the impact on availability a successful exploit of the vulnerability will have on the target asset. The possible scores for this metric are as follows:

i. none: No impact

ii. partial : There is significant resource interruption

iii. complete: A total shutdown of the resource

(d) Impact bias (IB) metric bib gives a stronger weighting to one of the impact metrics over the other two. This allows for distinctions to be made on the importance of CIA functionalities and services on the asset. The possible scores for this metric are as follows:

i. Normal: Weights on "Impact scores" for CIA are all equal

ii. Confidentiality: confidentiality impact (CI) is assigned greatest weight

iii. Integrity: integrity impact (II) is assigned greatest weight

iv. Availability: availability impact (AI) is assigned greatest weight

(e) Access complexity (AC) metric measures the complexity of attack required to exploit the vulnerability once an attacker has access to the target system. The possible scores for this metric are as follows:

i. High: Specialised access conditions exist

ii. Low: System always exploitable

(f) Authentication (Au) metric measures whether or not an attacker needs to be authenticated to the target system in order to exploit the vulnerability.

The possible scores for this metric are as follows:

i. Required: Authentication required to exploit the vulnerability

ii. Not Required: Authentication not required to exploit the vulnerability

(g) Access vector (AV) metric measures whether or not the vulnerability is locally or remotely exploitable. The possible scores for this metric are as follows:

i. Local : For local exploitation

ii. Remote: For remote exploitation

Temporal metric group z: These are metrics which give an indication of events that may occur which affect the urgency of the threat posed by the vulnerability. These metrics are as follows:

(a) Exploitability metric "attempts" to measure the current state of exploit technique or code availability and suggests a likelihood of exploitation. This assumes that there are more unskilled attackers than there are attackers who are skilled enough to research vulnerabilities and then create their own version of exploit code. The possible scores for this metric are as follows:

i. Unproven: No exploit code available yet

ii. Proof of Concept: The code or technique is not functional in all situations and may require substantial hand tuning by a skilled attacker

iii. Functional : Functional exploit code available

iv. High: The code works in every situation where the vulnerability is exploitable.

(b) Remediation Level (RL) metric zrm gives an indication of the effectiveness of the safeguards put in place. The possible scores for this metric are as follows:

i. Official Fix : A complete vendor solution is available

ii. Temporary Fix : An temporary official fix is available

iii. Workaround: An unofficial, non-vendor solution available

iv. Unavailable: No solution available or the solution is impossible to apply.

(c) Report Confidence (RC) metric measures the degree of confidence in the existence of the reported vulnerability and the credibility of the known technical details. The possible scores for this metric are as follows:

i. Unconfirmed: There is little confidence in the validity of the report e.g. rumours.

ii. Uncorroborated: Multiple, non-official sources. There may be conflicting reports.

iii. Confirmed: Vendor of the affected technology has acknowledged that the vulnerability exists.

The environmental metric group e: The metrics in this group give an indication of the risk posed to different operational environments by a vulnerability. The metrics are as follows:

(a) Collateral Damage potential (CD) metric measures the potential for a loss in physical equipment, property damage or loss of life or limb. The possible scores for this metric are as follows:

i. None: There is no potential for property or physical damage

ii. Low: There is light property or physical damage if the vulnerability
is exploited

iii. Medium: There is significant property or physical damage if the vulnerability
is exploited

iv. High: There is catastrophic property or physical damage if the vulnerability
is exploited

(b) Target distribution (TD) metric measures the relative size of the field of target systems susceptible to the vulnerability. The possible scores for this metric are as follows:

i. None: No target systems exist

ii. Low: Between 1% − 15% of the total environment is at risk.

iii. Medium: Between 16% − 49% of the total environment is at risk.

iv. High: Over 50% of the environment is at risk

The overall score is given as:

Score = $10b_{av}b_{ac}b_{au}((b_{ci}b_{cib}) + (b_{ii}b_{iib}) + (b_{ai}b_{aib}))+ bz_{ex}z_{rm}z_{rc\,+}\, e = z + ((10 - z)e_{cd})e_{td}$
… 2.1

The scores in the category of 0-4 are classified Low, 4-7 as Medium and 8-10 as High Based on the Snort and CVSS Classification, Wang and Zhao (2006) Attack Phases are discussed under the subjects of Minor and Major Threats.

a. Minor Threats: They are usually found in the Preparation and Access Gaining Phases of the Attack.

i. The Preparation Phase involves pinging and information gathering using such techniques as footprinting, scanning and enumeration.

ii. The Access Gaining Phase is where attackers gain user-level access using the information obtained from previous phase. The attackers at this phase employ password eavesdropping, file sharing, bruteforce and bufferoverflow.

b. Major Threats: These threats are usually found in the Privilege Escalation, Pilfering, Track Covering and Backdoor Phases of Attack.

i. The Privilege Escalation Phase is where the super-user level is gained by the attacker in order control the system. Password Cracking and Vulnerability Exploitaions are some of the techniques used.

ii. Pilfering Phase is used to gather more information about system to compromise other systems using the trusted connections.

iii. Track Covering Phase is where the intrusive traces are covered up by disabling the auditing, cleaning the event log and hiding the attacking toolkits.

iv. Backdoor Creation Phase is used to re-establish direct connections for future intrusions. Rogue accounts can be created, start-up files may be modified to activate secret services at boot time and application may be trojanized at the phase.

Other tools include M-Correlator by Porras et al. (2002), Fuz-Met by Albushi *et al.* (2009) and Multi-strategic Incident Prioritisation by Jumaat (2012). They all prioritised reconnaissance, scanning and public-user access as minor threats having assigned low priority to threat. However, Jumaat (2012) was dynamic as the priority increases continuously with time.

2.3 Information Security

Information Security (InfoSec) is the protection of information and its critical elements including the systems and the hardware that use, store, and transmit that information (Whitman and Mattord, 2004). The Committee on National Security

Standard (CNSS) model of information security has been the industry standard for computer security since the development of the mainframe computer. This consists of key objectives described as C.I.A. (Whitman and Mattord, 2004).

The "C" stands for Confidentiality which is an Information Security objective that is used to ensure that only those with sufficient privileges and a demonstrated need may access certain information. To protect the confidentiality of information, a number of measures are used such as information classification; secure document storage; application of general security policies; education of information custodians and end users; and cryptography.

Also, the "I" stands for Integrity. It is the quality or state of being whole, complete ad uncorrupted. The integrity of information is threatened when it is exposed to corruption, damage, destruction or other disruption of its authentic state. Systems could employ error control techniques to compensate for external and internal threats to information integrity. Some other techniques may be by looking out for changes in file state as indicated by the operating system.

The "A" stands for the Availability, which is the characteristics of information that enables user access to information without interference or obstruction, in a usable format. A user in this definition may be either a person or another computer system. Availability does not imply that the information is accessible to any user; rather it means availability to authorized users.

Information Security is composed of Network Security, Computer and Data Security, Management of Information Security and Policy (Whitman and Mattord, 2004). The network security addresses the protection of an organisation's data networking devices, connections, contents and the ability to use the network to achieve organisation's data communication functions while the Computer and Data Security deals a whole lot with the protection of computer hardware, software, database and file system in Information System. Also, Management of Information Security is the process of achieving information security objectives of organisation using a given set of resources such as Network Security and Data Security devices. The Policy refers to the set of rules, standards or procedures used to achieve the information security aim of an organisation. In InfoSec, there are agents in form of human, software or hardware that manages the protection tasks referred to as Information Security Manager.

2.3.1 Information Security Tools

In order to ensure protection of network resources in organisations, various categories of Information Security mechanisms such as Patching, Vulnerability Assessment, Intrusion Detection, Firewall Protection, Encryption and Antimalware are used. The next subsections discuss some Information Security popular network security tools.

a. *Vulnerability Scanning Tool:* They are capable of scanning networks for very detailed information. As a class, they identify exposed user names and groups, show open network shares and expose configuration problems and other server vulnerabilities. An example is Nmap, a professional freeware utility available from www.insecure.org/nmap.

b. *Intrusion Detection and Prevention System:* Intrusion Detection System is software that detects violation of networks, hosts and security configurations. It can report alerts to the administrator or prevent violation by blocking abnormal traffics. The variants of Intrusion Detection System that block abnormal traffics is usually interoperated with Intrusion Prevention System.

There are two main categories of Network Intrusion Detection System: Signature-based Network Intrusion Detection System and Anomaly-based Network Intrusion Detection System. The Signature-based Network Intrusion Detection System makes use of predetermined policy (intrusion signature) to detect threat. The main limitation is that it often results in high false negative rate. The Anomaly-based Network Intrusion Detection System learns from the normal state of network activities to determine intrusion. The main limitation is that it often results in high false positive rate.

c. *Firewall:* Firewalls have made significant advances since their earliest implementations. The first generation firewall devices were router that could perform one simple packet filtering operations. The first generation of firewall are known as Packet Filtering Firewall that filters packet by examining every incoming and outgoing packet header. They can selectively filter packets based on values in the packet header, accepting or rejecting packets as needed. These devices can be configured to filter based on IP addresses, type of packet, port request and other elements present in the packets. The filtering examines packets for compliance with or violation of rules configured into the firewall database. In Figure 2.2, the basic set-up of Firewall is presented.

An application firewall is a form of firewall which controls input, output, and/or access from, to, or by an application or service. It operates by monitoring and potentially blocking the input, output, or system service calls which do not meet the configured policy of the firewall. The application firewall is typically built to control all network traffic on any OSI layer up to the application layer. It is able to control applications or services specifically, unlike a stateful network firewall which is - without additional software - unable to control network traffic regarding a specific application.

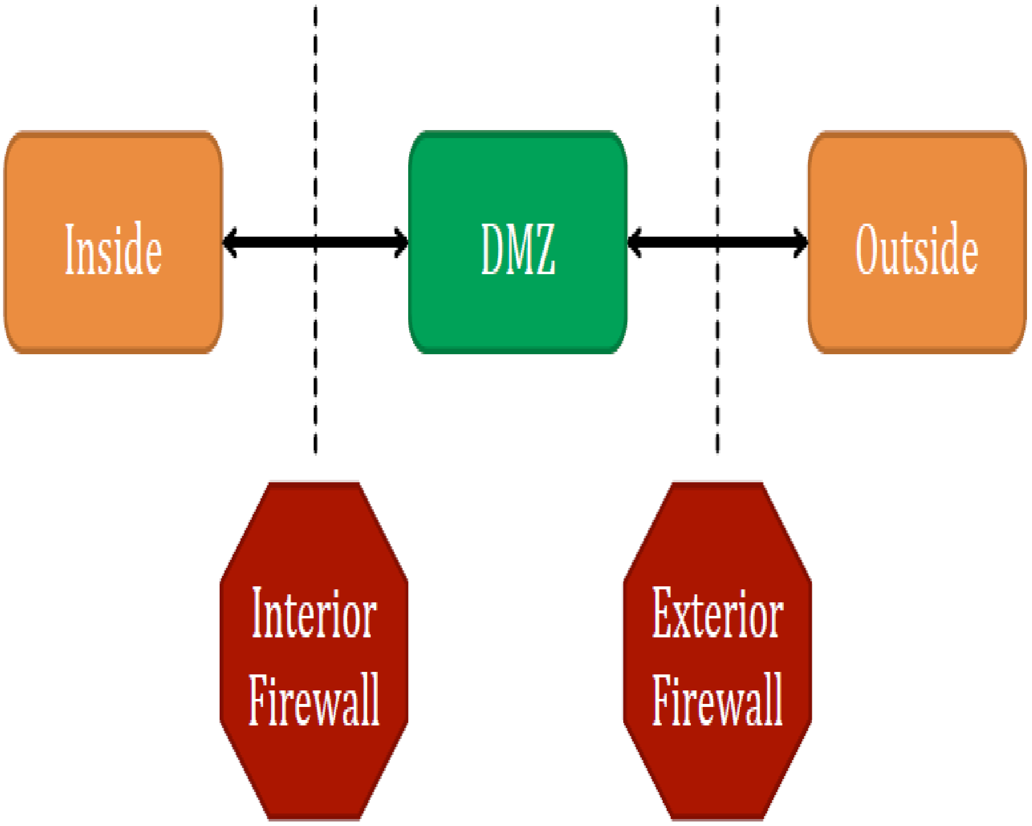


Figure 2.2: Set-up of Packet Firewall (Brumley, 2012)

d. *Encryption Tools:* Encryption is one part of cryptography. It is the process of converting an original message into a form that cannot be understood by unauthorized individuals. The encryption tools are the algorithms that are used to convert unencrypted message to encrypted message. There are two types:

e. Symmetric Encrytion and Asymetric Encryption. When same key are used to encipher and decipher, it is called Symmetric or Private Key Encryption. Examples of such tools include Triple DES (3DES), Advanced Encryption Standard (AES). Also, when different keys are used to encipher and decipher, it is known as Asymetric or Public Key Encryption. Examples include Rivest Shamir Aldermann (RSA) Algorithm and Elliptic Curve.

f. Antimalware: Antimalware otherwise known as Antivirus is a program that is used to scan and disinfect systems of malicious codes. They protect networks, systems and applications from damage caused by malicious codes such as virus, Trojan and worms.

2.4 Information Security Management

According to ENISA (2006), Information Security Management can be described based on the following facts:

a) information technology security administrators should expect to devote approximately one-third of their time addressing technical aspects. The remaining two-thirds should be spent developing policies and procedures, performing security reviews and analysing risk, addressing contingency planning and promoting security awareness;

b) security depends on people more than on technology;

c) employees are a far greater threat to information security than outsiders;

d) security is like a chain. It is as strong as its weakest link;

e) the degree of security depends on three factors: the risk you are willing to take, the functionality of the system and the costs you are prepared to pay;

f) security is not a status or a snapshot but a running process.

These facts inevitably lead to the conclusion that:

Security administration is a management and NOT a purely technical issue

Therefore the establishment, maintenance and continuous update of Information Security Management System provide a strong indication that a company is using a systematic approach for the identification, assessment and management of information security risks. Furthermore such a company will be capable of successfully addressing information confidentiality, integrity and availability requirements which in turn have implications for business continuity; minimization of damages and losses; competitive edge; profitability and cash-flow; respected organisation image and legal compliance.

The main objective of Information Security Management is to implement the appropriate measurements in order to eliminate or minimize the impact that various security related threats and vulnerabilities might have on an organisation. In doing so, Information Security Management will enable implementing the desirable qualitative characteristics of the services offered by the organisation (i.e. availability of services, preservation of data confidentiality and integrity etc.). The Information Security Management System (ISMS) framework is presented in Figure 2.3. and contains the following six steps:

Step 1. Definition of Security Policy,

Step 2. Definition of ISMS Scope,

Step 3. Risk Assessment (as part of Risk Management),

Step 4. Risk Management,

Step 5. Selection of Appropriate Controls and

Step 6. Statement of Applicability

Steps 1 and 2 produce the security policy documents and the scope of the ISMS. Steps 3 and 4, the Risk Assessment and Management process, comprise the heart of the ISMS and are the processes that "transform" on one hand the rules and guidelines of security policy and the targets; and on the other to transform objectives of ISMS into specific plans for the implementation of controls and

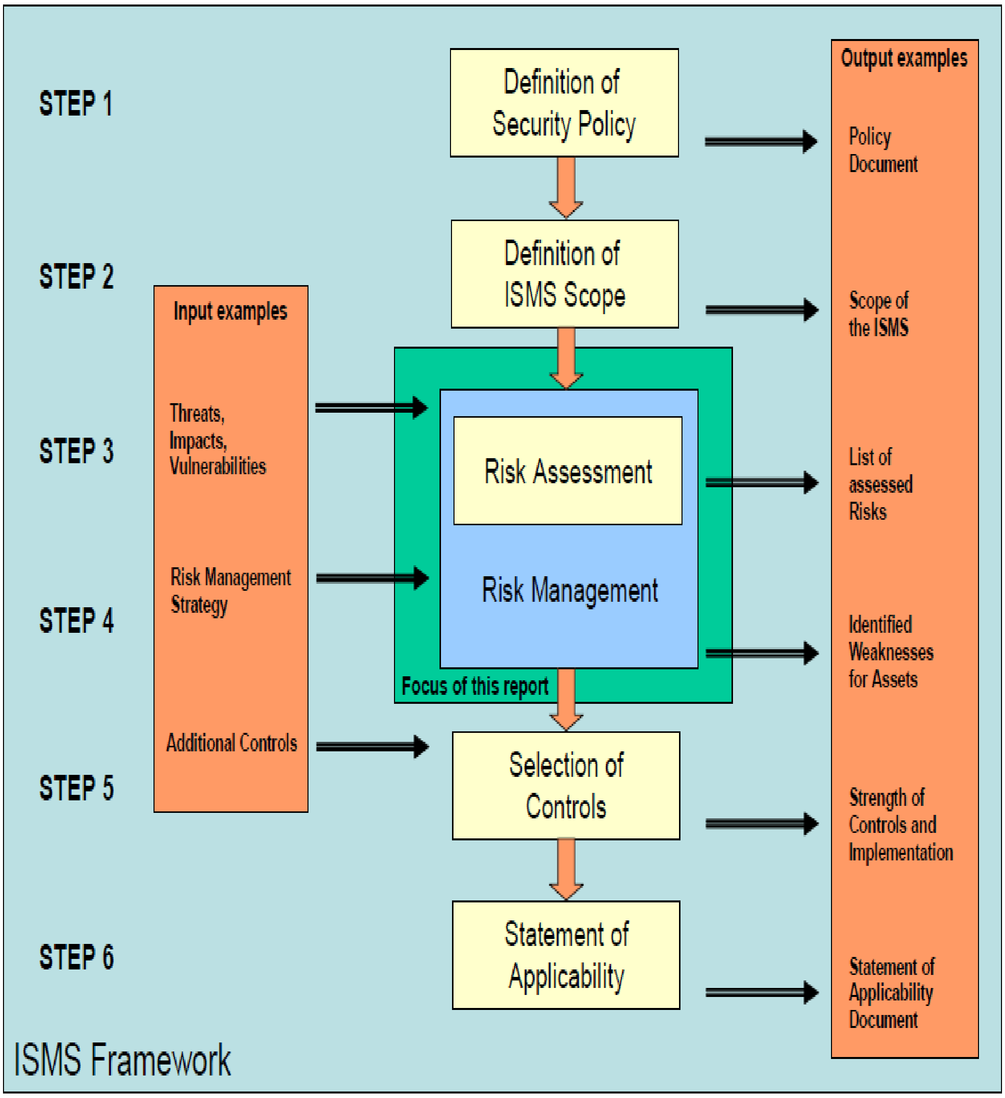


Figure 2.3 Information Security Management System Framework (ISO17799, 2000 & 2005)

mechanisms that aim at minimizing threats and vulnerabilities. The processes and activities related to the steps 5 and 6 do not concern information risks. They are rather related to the operative actions required for the technical implementation, maintenance and control of security measurements.

The International Organisation for Standardization (ISO) established in 1947 is a non-governmental international body that collaborates with the International Electro-technical Commission (IEC) and the International Telecommunication Union (ITU) on Information and Communications Technology (ICT) standards. The organisation has developed many standards for Information Security Management. The first amongst them is ISO/IEC 17799: 2005 which has now been replaced by ISO/IEC 27002:2005. The following ISO standards are currently

used in Information Security Management (ISO, http://www.iso27001security.com).

a. *ISO/IEC 27002:2005 (Code of Practice for Information Security Management)*

   ISO/IEC 27002:2005 is an international standard that originated from the BS7799-1, one that was originally laid down by the British Standards Institute (BSI). ISO/IEC 27002:2005 refers to a code of practice for information security management, and is intended as a common basis and practical guideline for developing organisational security standards and effective management practices. This standard contains guidelines and best practices recommendations for these ten security domains: (i) security policy; (ii) organisation of information security; (iii) asset management; (iv) human resources security; (v) physical and environmental security; (vi) communications and operations management; (vii) access control; (viii) information systems acquisition, development and maintenance; (ix) information security incident management; (x) business continuity management and compliance. Among these 10 security domains, a total of 39 control objectives and hundreds of best-practice information security control measures are recommended for organisations to satisfy the control objectives and protect information assets against threats.

b. *ISO/IEC 27001:2005 (Information Security Management System: Requirements)* The international standard ISO/IEC 27001:2005 has its roots in the technical content derived from BSI standard BS7799 Part 2. It specifies the requirements for establishing, implementing, operating, monitoring, reviewing, maintaining and improving a documented Information Security Management System (ISMS) within an organisation. It is designed to ensure the selection of adequate and proportionate security controls to protect information assets9. This standard is usually applicable to all types of organisations, including business enterprises, government agencies, and so on. The standard introduces a cyclic model known as the "Plan-Do-Check-Act" (PDCA) model that aims to establish, implement, monitor and improve the effectiveness of an organisation's ISMS. The PDCA cycle has these four phases:

i. "Plan" phase – establishing the ISMS

ii. "Do" phase – implementing and operating the ISMS

iii. "Check" phase – monitoring and reviewing the ISMS

iv. "Act" phase – maintaining and improving the ISMS

Often, ISO/IEC 27001:2005 is implemented together with ISO/IEC 27002:2005. ISO/IEC 27001 defines the requirements for ISMS, and uses ISO/IEC 27002 to outline the most suitable Information Security controls within the ISMS.

c. ISO/IEC 15408 (Evaluation Criteria for IT Security)

The international standard ISO/IEC 15408 is commonly known as the "Common Criteria" (CC). It consists of three parts: ISO/IEC 15408-1:2005 (introduction and general model), ISO/IEC 15408-2:2005 (security functional requirements) and ISO/IEC 15408-3:2005 (security assurance requirements). This standard helps evaluate, validate, and certify the security assurance of a technology product against a number of factors, such as the security functional requirements specified in the standard. Hardware and software can be evaluated against CC requirements in accredited testing laboratories to certify the exact EAL (Evaluation Assurance Level) the product or system can attain. There are seven EALs: EAL1 - Functionally tested, EAL2 - Structurally tested, EAL3 - Methodically tested and checked, EAL4 - Methodically designed, tested and reviewed, EAL5 - Semi-formally designed and tested, EAL6 - Semi-formally verified, designed and tested, and EAL7 - Formally verified, designed and tested.

d. *ISO/IEC 13335 (IT Security Management)*

ISO/IEC 13335 was initially a Technical Report (TR) before becoming a full ISO/IEC standard. It consists of a series of guidelines for technical security control measures:

i. ISO/IEC 13335-1:2004 documents the concepts and models for information and communications technology security management.

ii. ISO/IEC TR 13335-3:1998 documents the techniques for the management of IT security.

iii. ISO/IEC TR 13335-4:2000 covers the selection of safeguards (i.e. technical security controls).

iv. ISO/IEC TR 13335-5:2001 covers management guidance on network security.

Other standards include:

e. *Payment Card Industry Data Security Standard*

(www.pcisecuritystandards.org/tech)

The Payment Card Industry (PCI) Data Security Standard (DSS) was developed by a number of major credit card companies (including American Express, Discover Financial Services, JCB, MasterCard Worldwide and Visa International) as members of the PCI Standards Council to enhance payment account data security. The standard consists of twelve core requirements, which include security management, policies, procedures, network architecture, software design and other critical measures. These requirements are organised into the following areas:

i. Build and Maintain a Secure Network

ii. Protect Cardholder Data

iii. Maintain a Vulnerability Management Program

iv. Implement Strong Access Control Measures

v. Regularly Monitor and Test Networks

vi. Maintain an Information Security Policy

f. *COBIT (Control Objectives for Information and related Technology)* (www.isaca.org/COBIT)

COBIT was developed by the IT Governance Institute (ITGI) in 1995. It is "a control framework that links IT initiatives to business requirements, organises IT activities into a generally accepted process model, identifies the major IT resources to be leveraged and defines the management control objectives to be considered". The latest update is version 4.1, published in 2007. COBIT 4.1 consists of seven sections, which are (i) Executive overview, (ii) COBIT framework, (iii) Plan and Organise, (iv) Acquire and Implement, (v) Deliver and Support, (vi) Monitor and Evaluate, and (vii) Appendices, including a glossary. Its core content can be divided according to the thirty four IT processes. COBIT is increasingly accepted internationally as a set of guidance materials for IT governance that allows managers to bridge the gap between control requirements, technical issues and business risks. Based on COBIT 4.1, the COBIT Security Baseline focuses on the specific risks around IT security in a way that is simple to follow and implement for small and large organisations.

2.4.1 How to ensure Effectiveness of Information Security Management System?

For the ISMS to be effective, it must:

i. have continuous, unshakeable and visible support and commitment of the organisation's top management;

ii. be managed centrally, based on a common strategy and policy across the entire organisation;

iii. be an integral part of the overall management of the organisation related to and reflecting the organisation's approach to Risk Management, the control objectives and controls and the degree of assurance required;

iv. have security objectives and activities based on business objectives and requirements and led by business management;

v. undertake only necessary tasks and avoiding over-control and waste of valuable resources;

vi. fully comply with the organisation philosophy and mind-set by providing a system that instead of preventing people from doing what they are employed to do, it will enable them to do it in control and demonstrate their fulfilled accountabilities; be based on continuous training and awareness of staff and avoid the use of disciplinary measures and "police" or "military practices;

vii. be a never ending process.

2.4.2 Cost-Effective Information Security Management

The task of ensuring profit in Information Security expenditure is one of the duties of an Information Security Manager. The existing works on investment and budgeting decision processes that are currently prevailing in large and midsize businesses with respect to information security agreed that security funding should be managed from an economic perspective. That is, a firm should invest resources into security controls up to the point at which the last dollar of information security investment yields a dollar of savings (Su, 2006). The Gordon *et al.* (2005) review presented these economic approaches: Return on Investment for Security (ROSI), Net Present Value (NPV), and Annualized Loss Expectancy models, Security Attribute Evaluation (SEAM) and Cost-effectiveness Analysis methods and each approach used a specific form of a cost-benefit analysis.

In practice, however, it is rarely achievable to set up a budget decision making process that rests solely on results of these economic models due to the fact that both the cost and benefit components often are difficult to estimate. To keep cost-benefit analysis practical, we resolve to the concept of Information Security Project Management.

The Project Management Institute defined *Project* as a temporary endeavour undertaken to create a unique product, service, or result. The temporary nature of projects indicates a definite beginning and end. The end is reached when the project's objectives have been achieved or when the project is terminated because its objectives will not or cannot be met, or when the need for the project no longer exists. *Project Management* is the application of knowledge, skills, tools and techniques to project to meet project requirements (Project Management Institute, 2008). Project Management is accomplished through the use of processes such as: initiating, planning, executing, controlling and closing.

The following are some of the proven benefits of Project Management:

a. Provides clear roles, responsibilities, activities and schedules for team efforts.

b. Includes a method for considering the consequences of decreasing or increasing funds, resources, time, or quality.

c. Specifies a detailed plan of how to achieve our objectives.

d. Assists in the realistic assignments of tasks and responsibilities to team members according to the skills and resources available.

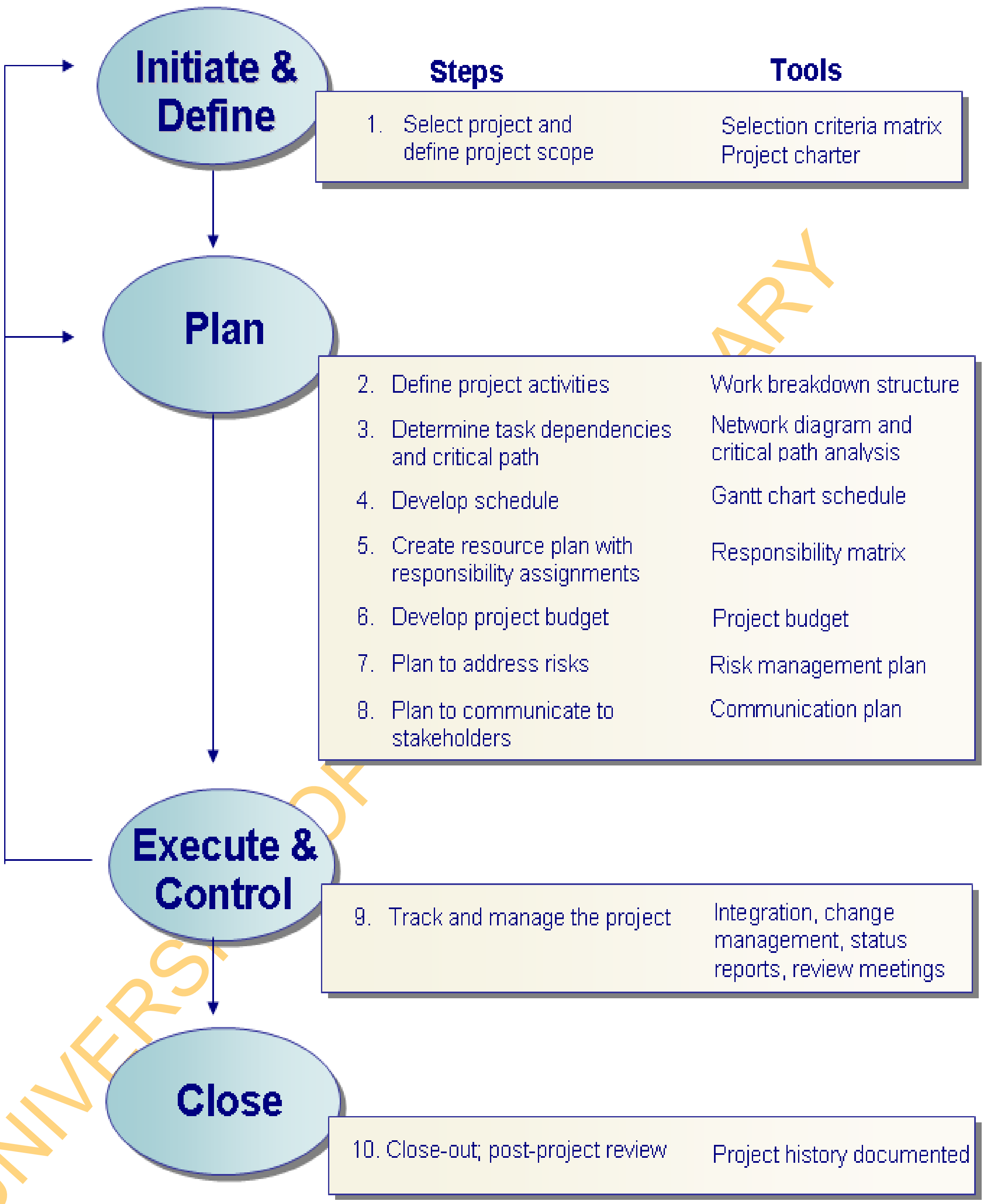
e. Gives structure to communicating the progress of projects.

f. Allows teams to identify potential problems and take preventive action early.

g. Keeps management officers and project stakeholders well-informed and supportive.

h. Helps manage pressure for expanding the scope of projects without proper decision criteria and analysis of changes.

Project Management is addressed using some set of steps as presented in Project Management Method in Figure 2.4.



Steps	Tools
1. Select project and define project scope	Selection criteria matrix Project charter
2. Define project activities	Work breakdown structure
3. Determine task dependencies and critical path	Network diagram and critical path analysis
4. Develop schedule	Gantt chart schedule
5. Create resource plan with responsibility assignments	Responsibility matrix
6. Develop project budget	Project budget
7. Plan to address risks	Risk management plan
8. Plan to communicate to stakeholders	Communication plan
9. Track and manage the project	Integration, change management, status reports, review meetings
10. Close-out; post-project review	Project history documented

Figure 2.4: Project Management Method (Institute for Development Management, www.bms.com/documents/STF)

In order to apply Project Management to Information Security, we apply the definition that Project Management is the application of a collection of tools and techniques (such as the CPM and matrix organisation) to direct the use of diverse resources toward the accomplishment of a unique, complex, one-time task within time, cost and quality constraints (Oizen, 1971) described by the Iron Triangle in Figure 2.5.

In order to apply Project Management to Information Security therefore, the three factors: Time, Cost and Quality shall be the focus as reflected in the Project Management Knowledge Areas.

Time – This refers to the actual time required to produce a deliverable, which in this case, would be the end result of project. Naturally, the amount of time required to produce the deliverable will be directly related to the amount of requirements that are part of the end result (scope) along with the amount of resources allocated to the project (cost).

Cost – This is the estimation of the amount of money that will be required to complete the project. Cost itself encompasses various things, such as: resources, labour rates for contractors, risk estimates, bills of materials, et cetera. All aspects of the project that have a monetary component are made part of the overall cost structure.

Quality – It is an emergent property of peoples, different attitudes and beliefs, which often change over the development life-cycle of a project. It determines the success of a project based on the project scope.

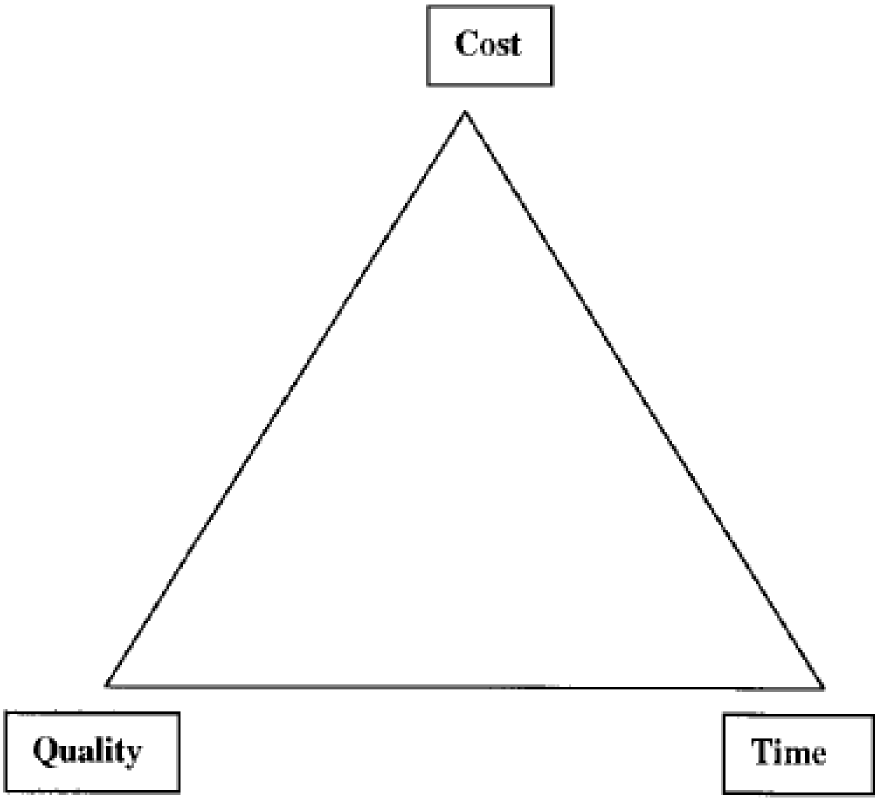


Figure 2.5: The Iron Triangle (Atkinson, 1999)

2.5 Security Design Patterns

Under the Security Design Patterns, Security Design Life-cycle and Secure-focused Configuration Management are discussed.

2.5.1 Security Design Lifecycle

The current Security Design Lifecycle Threat Modelling methodology is a four step process, designed to enable engineers with a measure of security expertise to model threat and have reasonable confidence that they have found important threats. The goals of the process are to improve the security of designs, to document the security design activity and to teach about security as people work through the process.

Like any other Information Technology process, security can follow a lifecycle model. The model presented here follows the basic steps of IDENTIFY – ASSESS – PROTECT – MONITOR. This lifecycle provides a good foundation for any security program. Using this lifecycle model provides you with a guide to ensure that security is continually being improved. A security program is not a static assessment or a finished product. Rather it requires constant attention and continual improvement. Figure 2.5 presents the diagram of Security Lifecycle.

a. *Identify:* The very first step in any security program is to know what is to be protected. The Identification Phase needs to start at the high level and drill down. You need to have a good understanding of the resources that you are trying to protect. Here are some questions to consider when trying identifying enterprise resources.

i. Where are the assets are physically located? Are they in a secured data centre or scattered about multiple office locations?

ii. How many servers, firewalls and routers do you have?

iii. What flavour of OS is running on each system?

iv. What applications and services are running on each server?

v. Who is the customer for each system? Does the application support the HR, finance or the marketing department?

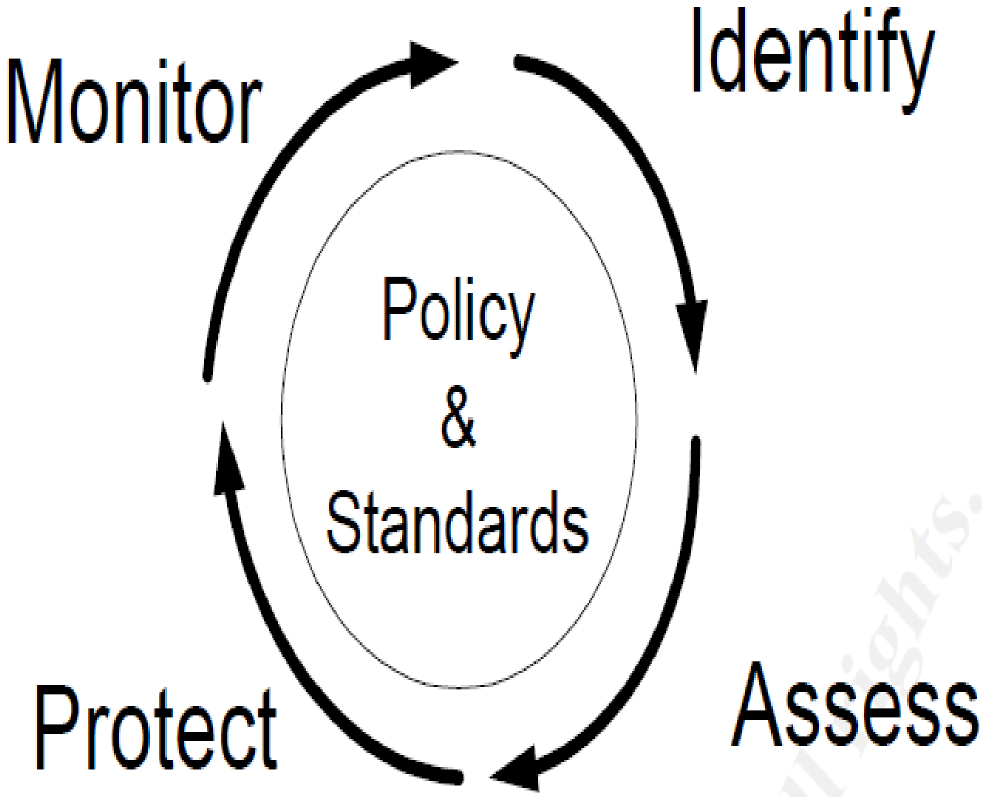


Figure 2.6: Security Life Cycle (PFau, 2003 in SANs Institute)

vi. What is the priority of the application? Is this a front end customer application or an internal, third tier application?

b. *Assess:* The assessment phase of the Security Lifecycle builds on the identification phase. Once the assets have been identified, the next step is to perform a thorough security assessment. The assessment phase can encompass many different aspects from reviewing processes and procedures to vulnerability scanning.

c. *Protect:* Once the network and systems are mapped out and some vulnerabilities are identified, the systems is brought in-line with corporate security policy and standards. Essentially it is at this time that the system is protected. This phase of the lifecycle is sometimes referred to as the ‘mitigation’ phase, since the objective is to mitigate any risks identified during the assessment phase. The focus of this phase is to configure and update each system and network component, so that its security is strengthened and complies with corporate policy. Thus, eliminating some vulnerabilities and mitigating others.

d. *Monitor:* The last phase of the security lifecycle is to monitor the security that has been established. For instance, when the security of servers, firewalls and routers is strengthened, it is necessary to ensure that those changes remain in place. Additionally, you need to monitor the compliance of new systems that are introduced into the enterprise. Computer systems are dynamic and are continually being updated and modified by administrators, developers and anyone else that has access to them. A process needs to be implemented that monitors and measures the status of security across the enterprise. There are several key goals for the monitoring phase: security compliance and verification and validating the security posture of the enterprise.

2.5.2 Secured-focused Configuration Management

In managing security risks, Configuration Management activities are carried out in secured manner (Johnson *et al.*, 2011). Figure 2.7 presents the phases of Secured-focused Configuration Management (SecCM).

a. *Planning:* As with many security activities, planning can greatly impact the success or failure of the effort. As a part of planning, the scope or applicability of SecCM processes are identified. Planning includes developing policy and procedures to incorporate SecCM into existing information technology and security programs, and then disseminating the policy throughout the organisation. Policy addresses areas such as the implementation of SecCM plans, integration into existing security program plans, Configuration Control Boards (CCBs), configuration change control processes, tools and technology, the use of common secure configurations and baseline configurations, monitoring, and metrics for compliance with established SecCM policy and procedures. It is typically more cost-effective to develop and implement a SecCM plan, policies, procedures, and associated SecCM tools at the organisational level.

b. *Identifying and Implementing Configurations:* After the planning and preparation activities are completed, a secure baseline configuration for the information system is developed, reviewed, approved, and implemented. The approved baseline configuration for an information system and associated components represents the most secure state consistent with operational requirements and constraints. For a typical information system, the secure baseline may address configuration settings, software loads, patch levels, how the information system is physically or logically arranged, how various security controls are implemented, and documentation. Where possible, automation is used to enable interoperability of tools and uniformity of baseline configurations across the information system.

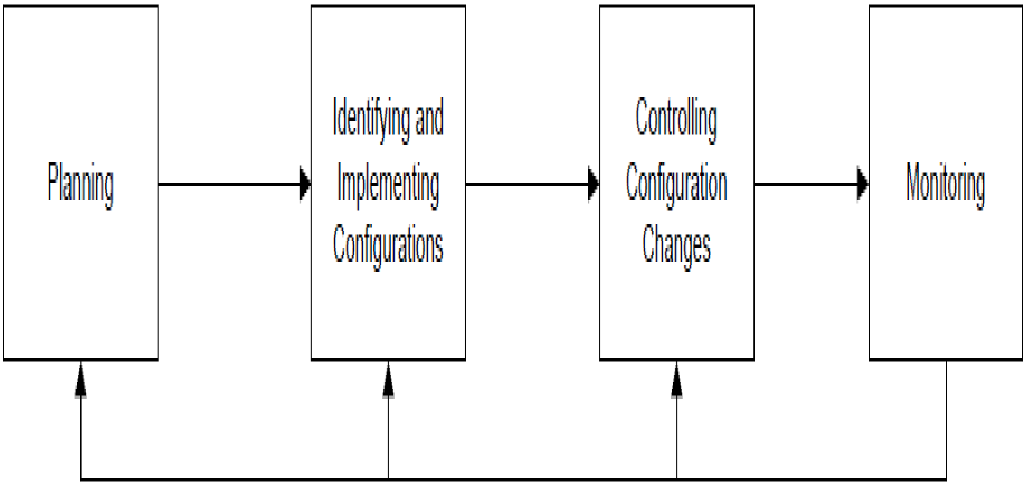

Planning
Identifying and Implementing Configurations
Controlling Configuration Changes
Monitoring

Figure 2.7: Secure-focused Configuration Management Phases

(Johnson *et al.*, 2011)

c. *Controlling Configuration Changes:* Given the continually evolving nature of an information system and the mission it supports, the challenge for organisations is not only to establish an initial baseline configuration that represents a secure state (which is also cost-effective, functional, and supportive of mission and business processes), but also to maintain a secure configuration in the face of the significant waves of change that ripple through organisations. In this phase of SecCM, the emphasis is put on the management of change to maintain the secure, approved baseline of the information system. Through the use of SecCM practices, organisations ensure that changes are formally identified, proposed, reviewed, analyzed for security impact, tested, and approved prior to implementation. As part of the configuration change control effort, organisations can employ a variety of access restrictions for change including access controls, process automation, abstract layers, change windows, and verification and audit activities to limit unauthorized and/or undocumented changes to the information system.

d. *Monitoring:* Monitoring activities are used as the mechanism within SecCM to validate that the information system is adhering to organisational policies, procedures, and the approved secure baseline configuration. Planning and implementing secure configurations and then controlling configuration change is usually not sufficient to ensure that an information system which was once secure will remain secure. Monitoring identifies undiscovered/undocumented system components, misconfigurations, vulnerabilities, and unauthorized changes, all of which, if not addressed, can expose organisations to increased risk. Using automated tools helps organisations to efficiently identify when the information system is not consistent with the approved baseline configuration and when remediation actions are necessary. In addition, the use of automated tools often facilitates

situational awareness and the documentation of deviations from the baseline configuration Processes and requirements within all four SecCM phases do not remain static thus all processes in all four phases are reviewed and revised as needed to support organisational risk management. SecCM monitoring is done through assessment and reporting activities. Reports address the secure state of individual information system configurations and are used as input to Risk Management Framework information security continuous monitoring requirements. SecCM monitoring can also support gathering of information for metrics that can be used to provide quantitative evidence that the SecCM program is meeting its stated goals, and can be used to improve SecCM processes in general.

2.6 Network Threat Management

Network Threat Management involves a whole lot of activities. An important aspect is Network Intrusion Detection and Prevention Management.

2.6.1 Network Intrusion Detection and Prevention Management

Network Intrusion Detection and Prevention Management borders with improving the efficiency and the effectiveness detection and filtering as presented below:

Existing studies on Network Intrusion Detection and Prevention Management have focused on reducing alerts, identifying critical threats, predicting intrusion and reducing false alarms. Alsubhi *et al.* (2008) categorise the research studies into two types: low-level and high-level alert operations. High-level operations apply aggregation, clustering, correlation, and/or fusion to sets of alerts in order to identify trends and abstractions within them, while low-level operations aim to identify the contextual information of each alert individually, and rate it based on its potential risk. As such, high-level operations aim to reduce alerts and improve detection efficiency, whereas low-level operations aim to enable a response mechanism by making informed decisions based on the contextual information and risk of each incident. However, a thorough analysis of threat must include both high and low-level operations, which is possible with Threat Modelling. *On Alarm reduction,* Alert Correlations (Ning *et al.*, 2002; Kruegel *et al.*, 2004a; Alserhani *et al.*, 2010; Ning *et al.*, 2001; Valdes and Skinner, 2001), which all aim to reduce the number of alerts and false alarms have been extensively studied. For Incident Management which involves low-level operation, we have such works as Low-level operations aim to improve the process of managing incidents and selecting appropriate responses. They are used to examine a large number of incidents and prioritise them by identifying which incidents are important, which are urgent and which are critical based on the potential risk. Example include: alert or incident prioritisation (Porras *et al.*, 2002; Lee and Qin, 2003; Alsubhi *et al.*, 2008; Dondo, 2008), risk assessment of incidents (Mu *et al.*, 2008) and Security Information and Event management (SIEM) (Alberts and Dorofee, 2004).

In Packet Firewall Management, the existing works mainly focus on Policy Configuration for efficiency and effectiveness improvement. They include Model-based Filtering Rules Analysis (Al-Shaer and Hamed , 2004; Matousek *et al.* 200; Jeffrey and Samik, 2009), which focuses on only filtering and High Level Language-based Filtering Rules Analysis (Eppstein and Muthukrishnan, 2001; Hari *et al.*, 2000; Bartal *et al.*, 1999; Hazelhurst, 1999; Mayer *et al.*, 2000), which focuses on rules representation and filtering.

These tasks have involved mostly analysis of threats. Michalski (2010) defined Threat Analysis as the method that is used to determine threats of interest using

a. *Leverage open and closed source data* to better quantify the level of threat in terms that are meaningful to the asset owners.

b. *Analyze and evaluate Data* from plausible data associations. What kind of information can be found in the data sources about a specific vulnerability/topic? What kind of "chatter" can be found on the internet**.**

c. *Review viable scenarios*, Identify Scenarios or attack vectors that leverage viable attack paths that can be realized by the level of capability of the threat.

d. *Provide mitigation* strategies

However, a reputable threat analysis should take into threat modelling to understand what the attack goals are, who the attackers are, what attacks are likely to occur, security assumptions of a system and where to best spend a security budget? (Schneier,1999). Threats are generally much easier to list than to describe, and much easier to describe than to measure. As a result, many organisations list threats, but fewer describe them in useful terms and still fewer measure them in meaningful ways. Several advantages ensue from the ability to measure threats accurately and consistently. Good threat measurement, for example, can improve understanding and facilitate analysis. It can also reveal trends and anomalies, underscore the significance of specific vulnerabilities, and help associate threats with potential consequences. In short, good threat measurement supports good security management. Unfortunately, the practice of defining and applying good threat metrics is still far reached. This is particularly true in the dynamic and complex domain of *Internet-facilitated Organised Crime* Threats. The following steps referred to as Threat Modelling steps are usually carried out to measure threats and ensure good security management. The steps include:

a. Identify potential threats and the conditions that must exist for an attack to be successful

b. Provide information about how existing safeguards affect required attack conditions

c. Provide information about which attack condition and vulnerability remediation activities add the most value

d. Help you understand which conditions or vulnerabilities, when eliminated or mitigated, affect multiple threats; this optimizes your security investment.

2.7 Threat Modelling

Threat modelling allows network security analyst to systematically identify and rate the threats that are most likely to affect their system. By identifying and rating threats based on a solid understanding of the architecture and implementation of application, one can address threats with appropriate countermeasures in a logical order, starting with the threats that present the greatest risk.

Threat modelling has a structured approach that is far more cost efficient and effective than applying security features in a haphazard manner without knowing precisely what threats each feature is supposed to address. With a random, "shotgun" approach to security, how does a security manager knows when his application is "secure enough," and how does he know the areas where his application is still vulnerable? The underline fact is that until threats are known, system security is unguaranteed.

The following terminologies are common in threat modelling:

a. *Asset.* A resource of value, such as the data in a database or on the file system (Meier *et al.*, 2003). A system resource. Assets are mostly categorized based on their significance. In fact, this affects the level of security that is devoted to a particular asset. Some classification of Assets include Demilitarized Zone and Inside Zone Assets, Network and System Assets, Security and Service Assets, etc.

b. *Vulnerability.* A weakness in some aspect or feature of a system that makes a threat possible (Meier *et al.*, 2003). Vulnerabilities might exist at the network, host, or application levels. Known vulnerability are indexed in Vulnerability repositories such as Common Vulnerability Exposure, National Vulnerability Database, Open Source Vulnerability Database, etc.

c. *Attack (or exploit).* An action taken by someone or something that harms an asset (Meier *et al.*, 2003). This include Malware flooding, Denial of Service, SQL Injection, Sniffing, etc.

d. *Countermeasure.* A safeguard that addresses a threat and mitigates risk (Meier *et al.*, 2003). The security measures that are used to safeguard include Patch, Encryption, Intrusion Detection and Prevention System, Firewall Filtering and Antimalware

e. *Threats:* Different definitions exist for threats. Michalski et al. (2012) defined a *threat* as a person or organisation that intends to cause harm. Duggan *et al.* (2007) defined threat as a malevolent factor, whether an organisation or an individual, with specific political, social or personal goal and some level of capability and intention to oppose an established government, a private organisation or an accepted social norm". Also, Reeshil (2013) defined threat as security incident that analyse network and gain information in order to eventually crash or corrupt networks.

Threat has been also defined as:

“capabilities, intentions and attack methods of adversaries, or any circumstance or event, whether originating externally or internally, that has the potential to cause harm to information or a program or system or cause those to harm others.”) (ISO-21827, 2007]

“A potential cause of an incident which may result in harm to a system or organisation.” (ISO-17799, 2005; ISO-13335-1, 2004]

“A potential violation of security.” (ISO-7498-2, 1989)

Also, there is threat agent which is defined as:

“a system entity that performs a threat action, or an event that results in a threat action.” (RFC4949, 2007)

“the originator and/or the initiator of deliberate or accidental man-made threats.” (ISO-21827, 2007)

Based on the above definitions, we conclude that:

“a threat can be defined as an action caused by threat agent
that can inflict harm or damage on the system normal
state through exploitation of the system’s vulnerability.”

2.7.1 Perspectives of Threat Modelling

Different perspectives exist for Threat Modelling. In terms of purpose, Threat Modelling has mostly been deployed at the Software Development Stage (Swiderski and Synder, 2004 & Myagmar and Yurcik, 2005). In addition, quite a number of works have been done in trying to understand Threat for the purpose of Information Security and Risk Management.

In the following categories of the Threat Modelling, the works have been done:

a. *Attack-centric Threat Model.* This is a case where system is evaluated from the point of view of an attacker and how they will go about exploiting the system and what they could possibly try to attack. Examples of such include RVA (2010), Danta *et al.* (2007), Bhattacharya *et al.*, (2008), Ritchey and Ammann (2000) and Sheyner *et al.* (2002).

b. *Asset-centric Threat Model:* In this case, the system is evaluated from the perspective of asset classification. For instance: personal information database, web server and mail-server. Example is CVSS (Mell *et al*., 2009),

c. *Defence-centric Threat Model:* The model evaluates weakness in security controls and looks for attacks against each element of the model. An example is Killourhy *et al. (*2004) Defence-centric Threat Taxonomy based on the Manifestation of Attack in Intrusion Detection.

d. *Hybrid-centric Threat Model:* The model evaluates the system security from the point of view of part or all the perspectives to look for possible attacks against each asset. Some of the works are ontological like Hasan and Myagmar (2005) & McHughes *et al*. (2001). Works such as Ha *et al*. (2006) Insider Threat Model and Kruegel *et al*. (2004b) Alert Verification Model are empirical.

2.7.2 Methods of Threat Modelling

There are two methods used for modelling threats. They include:

a. static analysis

b. predictive analysis

a. *Static Analysis*

This is the commonest method used for modelling threats. The analysis involves associating threats to predefined categories of threats. Examples include Microsoft STRIDE Model by Hernan *et al.* (2006), DREAD Model (Meier *et al.,* 2007) and Snort Severity (Caswell and Roesh, 1998).

b. Predictive Analysis

The concept of Attacker-based Threat Modelling is used to understand the mind and motivation of attackers and figure out how they might attack. Some consider this to be the opposite of asset-based threat modelling. Attacker-based Threat Modelling focuses not only on preparing friendly forces for defence (and offense), but also examines adversary capabilities and intent. It focuses on what an opponent may want and try to do. This leads to the concept of Predictive Analysis.

The concept of predictive analysis involves using statistical models and decision tools that analyze current and historical data to make predictions about future events. A well-known example of this is credit scoring. Based on a person's past behaviour, banks can make risk-based decisions on how much credit to extend and on what terms. To effectively conduct predictive analysis in the cyber-security space, sensors, data and trends are required.

Predictive analysis originated in malware identification—when a worm or virus was released, copycat authors often tried to modify the successful ones and re-release the malware for their own reasons. By identifying those worms or viruses that had the greatest potential for modification, vendors could develop signatures or heuristics that would likely stop copycats, even if the copycat malware had not yet been seen and analyzed.

In the enterprise environment, predictive analysis involves assimilating data from a number of sources, weighing them against historical patterns, and building a set of scenarios that can be used to identify and predict hostile actors and actions. The more information available, the more likely it is that the threat models will mirror reality and therefore, the more accurate the predictions.

According to an article in Government Computer News (GCN) magazine reported in SANS (2012), predictive analysis involves a number of steps:

a. Understanding the Problems.

b. Tying prediction variables to the problems

c. Selecting appropriate statistical model relevant to the problems

d. Preparing input data for application of the model

e. Validating the model with test data

f. Applying the models to production data, observing the accuracy over time and making the adjustments necessary.

A reputable tool used for predictive analysis is Data Mining. Data Mining refers to the non-trivial extraction of implicit, previously unknown and potentially useful information from data in databases (Zaiane, 1999). It is a key step of knowledge discovery in databases (KDD) which is the non-trivial process of identifying valid, novel, potentially useful, and ultimately understandable patterns in data (Fayyad *et al.*, 1996). In other words, data mining involves the systematic analysis of large data sets using automated methods. The Knowledge Discovery in Databases contains the following steps as presented in Figure 2.8.

i. *Developing an understanding of the application domain.* This is the initial preparatory step. It prepares the scene for understanding what should be done with the many decisions (about transformation, algorithms, representation, etc.). The people who are in charge of a KDD project need to understand and define the goals of the end-user and the environment in which the knowledge discovery process will take place (including relevant prior knowledge). As the KDD process proceeds, there may be even a revision of this step. Having understood the KDD goals, the pre-processing of the data starts.

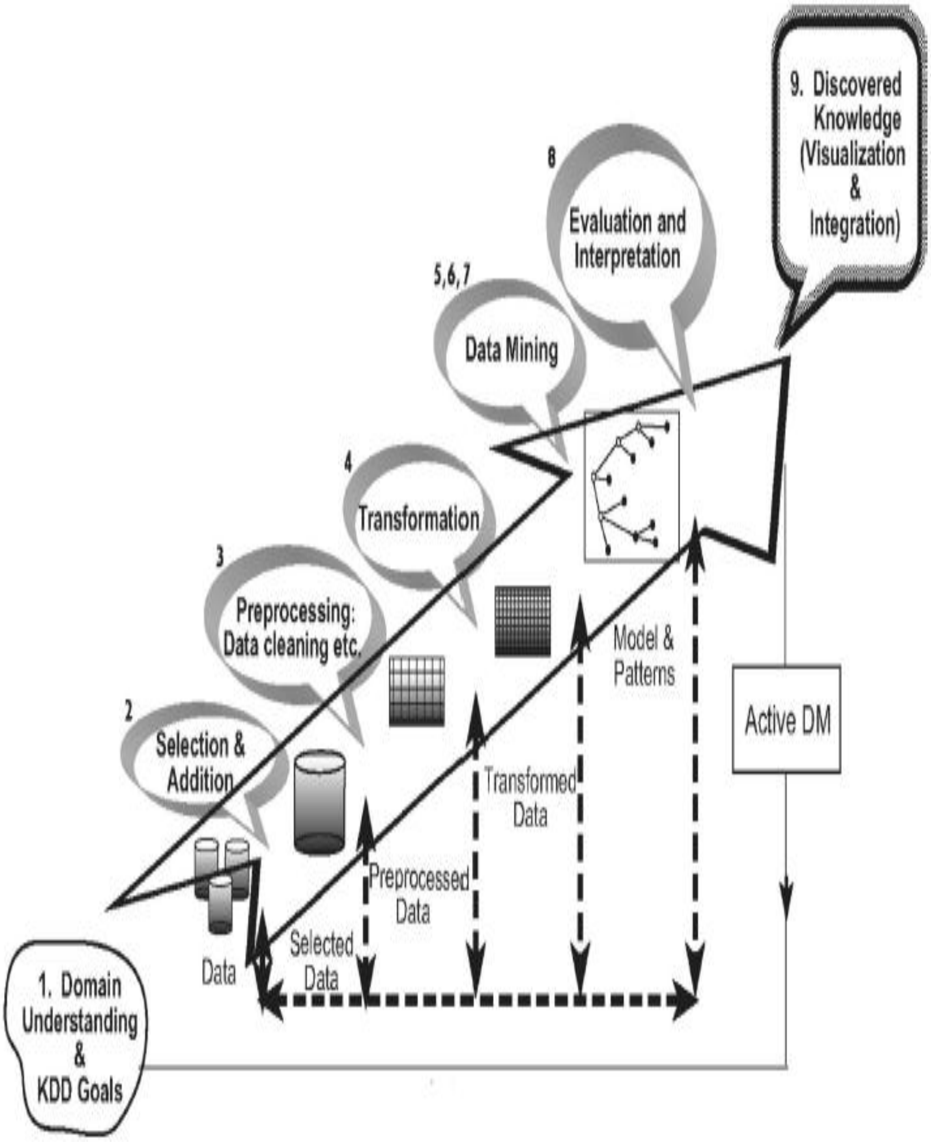

9. Discovered Knowledge (Visualization & Integration)
8
Evaluation and Interpretation
5,6,7
Data Mining
4
Transformation
3
Preprocessing: Data cleaning etc.
2
Selection & Addition
Model & Patterns
Active DM
Transformed Data
Preprocessed Data
Selected Data
Data
1. Domain Understanding & KDD Goals

Figure 2.8: Processes in Knowledge Discovery in Databases for Data Mining (Maimon and Rokash, 2009)

ii. *Selecting and creating a data set on which discovery will be performed*. Having defined the goals, the data that will be used for knowledge discovery should be determined. This includes finding out what data is available, obtaining additional necessary data, and then integrating all the data for the knowledge discovery into one data set, including the attributes that will be considered for the process. This process is very important because the Data Mining learns and discovers from the available data. This is the evidence base for constructing the models. If some important attributes are missing, then the entire study may fail.

iii. *Pre-processing and cleansing.* In this stage, data reliability is enhanced. It includes data clearing, such as handling missing values and removal of noise or outliers. It may involve complex statistical methods or using a Data Mining algorithm in this context. For example, if one suspects that a certain attribute is of insufficient reliability or has many missing data, then this attribute could become the goal of a data mining supervised algorithm. A prediction model for this attribute will be developed, and then missing data can be predicted.

iv. *Data transformation.* In this stage, the generation of better data for the data mining is prepared and developed. Methods here include dimension reduction (such as feature selection and extraction and record sampling), and attribute transformation (such as discretization of numerical attributes and functional transformation). This step can be crucial for the success of the entire KDD project, and it is usually very project-specific. Having completed the above four steps, the following four steps are related to the Data Mining part, where the focus is on the algorithmic aspects employed for each project.

v. *Choosing the appropriate Data Mining task.* We are now ready to decide: which type of Data Mining to use, for example, classification, regression, or clustering? This mostly depends on the KDD goals, and also on the previous steps. There are two major goals in Data Mining: prediction and description.

Prediction is often referred to as supervised Data Mining, while descriptive Data Mining includes the unsupervised and visualization aspects of Data Mining. Most Predictive Data Mining techniques are based on inductive learning, where a model is constructed explicitly or implicitly by generalizing from a sufficient number of training examples. The underlying assumption of the inductive approach is that the trained model is applicable to future cases. The strategy also takes into account the level of meta-learning for the particular set of available data.

vi. *Choosing the Data Mining algorithm.* Having the strategy, we now decide on the tactics. This stage includes selecting the specific method to be used for searching patterns (including multiple inducers). This approach attempts to understand the conditions under which a Data Mining algorithm is most appropriate. Each algorithm has parameters and tactics of learning (such as ten-fold cross-validation or another division for training and testing).

vii. *Employing the Data Mining algorithm***.** Finally, the implementation of the Data Mining algorithm is reached. In this step we might need to employ the algorithm several times until a satisfied result is obtained, for instance by tuning the algorithm's control parameters.

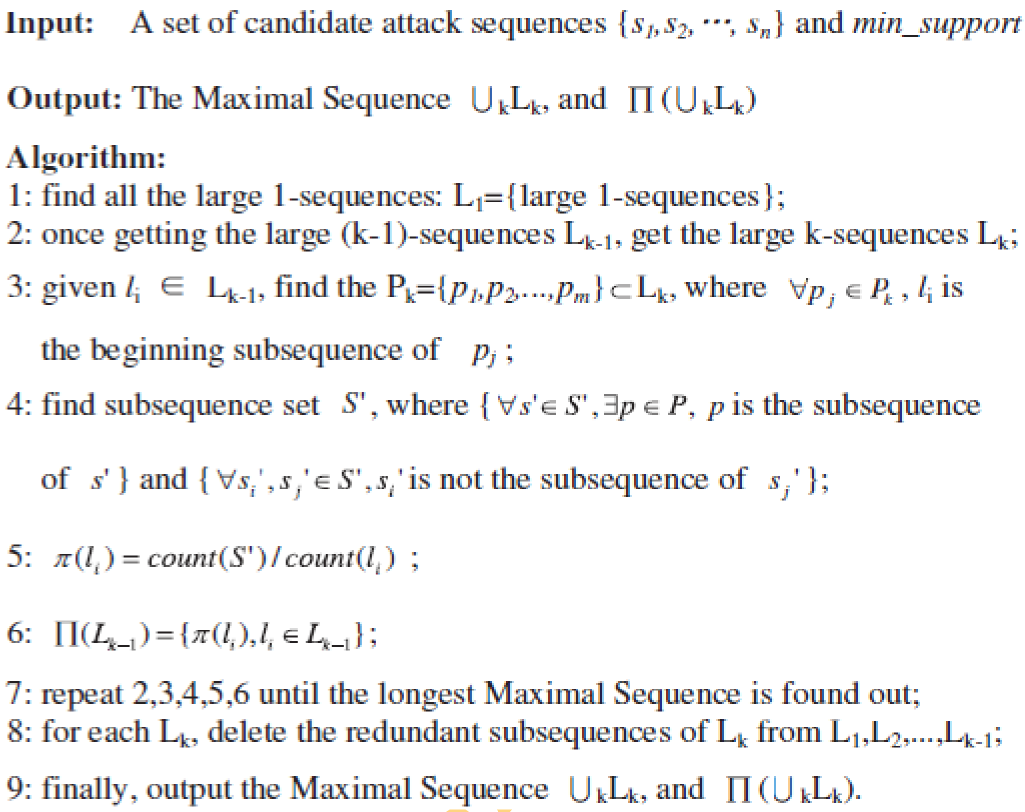
viii. *Evaluation.* In this stage, we evaluate and interpret the mined patterns (rules, reliability etc.), with respect to the goals defined in the first step. Here we consider the pre-processing steps with respect to their effect on the Data Mining algorithm results (for example, adding features in Step 4, and repeating from there). This step focuses on the comprehensibility and usefulness of the induced model. In this step the discovered knowledge is also documented for further usage. The last step is the usage and overall feedback on the patterns and discovery results obtained by the Data Mining:

ix. *Using the discovered knowledge***.** The knowledge is now ready to be incorporated into another system for further action. The knowledge becomes active in the sense that we may make changes to the system and measure the effects. Actually the success of this step determines the effectiveness of the entire KDD process.

In data mining, especially predictive data mining, associations among data are mined in form of rule using Association Data Mining techniques (or Association Mining techniques). The goal of mining association rule is to derive attribute correlation or interesting relationships among items in a given data set. Given a set of records, where each record is a set of items, support (x) is defined as the percentage of records that contain item set X. An association rule is an implication of the form *X => Y*, [c, s]. Here X n Y, *s = support (X u Y)* is the support of the rule, ands *c = support (X u Y) / support (X)* is the confidence. Rule support and confidence are two measures of rule interestingness that respectively reflect the usefulness and certainty of discovered rules (Han and Kamber, 2000). The earliest algorithm used for Association Mining is Apriori Algorithm, Agrawal and Srikant (1994) shown below.

```
procedure Apriori (T, minSupport)
L1= {frequent items};
for (k= 2; Lk-1 !=Ø; k++) {
        Ck= candidates generated from Lk-1
        // cartesian product Lk-1 x Lk-1 and eliminating any k-1 size itemset
         //that is not frequent
        for each transaction t in database do{
                #increment the count of all candidates in Ck contained in t
                Lk = candidates in Ck with minSupport
        }//end for each
}//end for
return __ __;
}
```

In Threat Identification using Predictive Analysis, Li *et al.* (2007) developed a Sequential Association Mining Method which could mine frequent sequence of attacks from candidate attack sequence generated using Time-based Window Size Approach.



Input: A set of candidate attack sequences $\{s_1, s_2, \cdots, s_n\}$ and *min_support*

Output: The Maximal Sequence $\cup_k L_k$, and $\prod(\cup_k L_k)$

Algorithm:

```
1: find all the large 1-sequences: $L_1$={large 1-sequences};
2: once getting the large (k-1)-sequences $L_{k-1}$, get the large k-sequences $L_k$;
3: given $l_i \in L_{k-1}$, find the $P_k=\{p_1,p_2,...,p_m\} \subset L_k$, where $\forall p_j \in P_k$, $l_i$ is
   the beginning subsequence of $p_j$;
4: find subsequence set $S'$, where $\{\forall s' \in S', \exists p \in P$, $p$ is the subsequence
   of $s'\}$ and $\{\forall s_i', s_j' \in S', s_i'$ is not the subsequence of $s_j'\}$;
5: $\pi(l_i) = count(S')/count(l_i)$ ;
6: $\prod(L_{k-1}) = \{\pi(l_i), l_i \in L_{k-1}\}$;
7: repeat 2,3,4,5,6 until the longest Maximal Sequence is found out;
8: for each $L_k$, delete the redundant subsequences of $L_k$ from $L_1,L_2,...,L_{k-1}$;
9: finally, output the Maximal Sequence $\cup_k L_k$, and $\prod(\cup_k L_k)$.
```

Other similar works include Inter-transaction Association Rules Mining for Intrusion Detection (Berberidis *et al.*, 2004) and Intrusion Detection using Fuzzy Association Mining (Luo and Bridges, 2000). However, none of the algorithm was developed paid attention to the association step.

2.7.3 Threat Modelling Steps

Different Threat Modelling frameworks exist. In this section, Threat Modelling is discussed based on Olzak (2006) and Meier *et al.* (2003) Threat Modelling Frameworks because of the inclusiveness of their views.

The Threat Modelling steps are six; namely:

a. *Categorization and Selection of Assets*

This is the step in which all assets in Information Systems are mapped to categories and critical ones may be selected for decomposition.

b. *Decomposition of the System*

This is the step in which threat model is created from the selected system. A system is defined as an environment within a network that provides a specific set of related functions. For example, a human resources application, with all related servers, routers, switches, operating systems, user workstations, etc. is a system. It may require creation of Security Profile. Table 2.2 presents a typical example of System Decomposition model.

c. *Identification of Possible Points of Attacks*

The first step in the identification of attack points is designating trust boundaries. A trust boundary separates processes, system components, and other elements that have different trust levels. Examples of entry points include sockets, interfaces between application components and user workstations, some application protocols such as http, ftp and ssh and some system vulnerabilities. At each trust boundary and the types of safeguards that provide access controls are identified. This information is required when completing attack trees.

Till date, Kruegel *et al.* (2004) Alert Verification for Intrusion Success, Heymann *et al.* (2006) Intrusion Detection through Alert Verification and Bolzoni et al. (2007) ATLANTIDES Architecture for Alert Verification in Network Intrusion Detection Systems. are the works that have proposed automatic verification method for identifying Intrusion Attempt by matching vulnerability with intrusion.

Table 2.2: Example of System Decomposition Model

Category	**Considerations**
Input validation	Is all input data validated? Could an attacker inject commands or malicious data into the application? Is data validated as it is passed between separate trust boundaries (by the recipient entry point)? Can data in the database be trusted?
Authentication	Are credentials secured if they are passed over the network? Are strong account policies used? Are strong passwords enforced? Are you using certificates? Are password verifiers (using one-way hashes) used for user passwords?
Authorization	What gatekeepers are used at the entry points of the application? How is authorization enforced at the database? Is a defence in depth strategy used? Do you fail securely and only allow access upon successful confirmation of credentials?
Configuration management	What administration interfaces does the application support? How are they secured? How is remote administration secured? What configuration stores are used and how are they secured?
Sensitive data	What sensitive data is handled by the application? How is it secured over the network and in persistent stores? What type of encryption is used and how are encryption keys secured?
Session management	How are session cookies generated? How are they secured to prevent session hijacking? How is persistent session state secured?

	How is session state secured as it crosses the network?
	How does the application authenticate with the session store?
	Are credentials passed over the wire and are they maintained by the application?
	If so, how are they secured?
Cryptography	What algorithms and cryptographic techniques are used?
	How long are encryption keys and how are they secured?
	Does the application put its own encryption into action?
	How often are keys recycled?
Parameter manipulation	Does the application detect tampered parameters?
	Does it validate all parameters in form fields, view state, cookie data, and HTTP headers?
Exception management	How does the application handle error conditions?
	Are exceptions ever allowed to propagate back to the client?
	Are generic error messages that do not contain exploitable information used?
Auditing and logging	Does your application audit activity across all tiers on all servers?
	How are log files secured?

d. *Identification of Threats*

Three approaches are used for threat identification; namely:

i. Single-step Identification Approach

ii. Step-by-Step Identification Approach

iii. Attack Pattern Identification Approach

i. *Single-step Identification Approach:* This is made up of threat identification models in which attacks are recognized as independent phenomena and predetermined. Examples include STRIDE (Hernan *et al*., 2006) and KDD 1999 Intrusion (MIT Lincoln Lab., http://www.ll.mit.edu/ attack model. It is a very simple approach for threat identification. Table 2.3 presents STRIDE Model while Table 2.4 presents the KDD 1999 Model. The DoS attacks deny legitimate requests to a system. Probe involves scanning and probing for getting confidential data. R2L is unauthorized access from a remote user and U2R is unauthorized access to local super-user privileges.

ii. *Step-by Step Threat Identification Approach:* Using a step-by-step analysis typically produces a more complete threat list. In this step, one identify threats that might affect system and compromise assets. To conduct this identification process, two basic approaches are used:

- *Team brainstorming on likelihood of Threat.* For example, could an attacker spoof an identity to access your server or Web application? Could someone tamper with data over the network or in a store? Could someone deny service?
- *Classification of Threats:* With this approach, a laundry list of common threats grouped by network, host, and application categories is produced. Next, apply the threat list to application architecture and any vulnerability identified earlier in the process.

This may involve the following tasks: Identifying network threats, host threats, application threats.

Table 2.3: STRIDE Model

Letter	**STRIDE (MSDN)**
S	Spoofing Identity – Impersonating someone else to the computer
T	Tampering with Data – The malicious modification of data
R	Repudiation – Involves users who can deny performing an action without other parties having any way to prove otherwise
I	Information Disclosure – Involves the exposure of information to individuals who are not supposed to have access to it.
D	Denial of Service
E	Elevation of Privilege- An unprivileged user gains privileged access and thereby has enough access to compromise or destroy the system

Table 2.4: KDD 1999 Threat Model

S/N	Main Attack Classes	Attack Classes
1	Denial of Service (DoS)	back, land, neptune, pod, smurt, teardrop
2	Remote to User (R2L)	ftp_write, guess_passwd, imap, multihop, phf, spy, warezclient, warezmaster
3	User to Root (U2R)	buffer_overflow, perl, loadmodule, rootkit
4	Probing	ipsweep, nmap, portsweep, satan

✓ *Identify Network Threats*

This is a task for network designers and administrators. It analyzes the network topology and the flow of data packets, together with router, firewall, and switch configurations, and look for potential vulnerabilities. Top network threats to consider during the design phase include:

- Using security mechanisms that rely on the IP address of the sender. It is relatively easy to send IP packets with false source IP addresses (IP spoofing).
- Passing session identifiers or cookies over unencrypted network channels. This can lead to IP session hijacking.
- Passing clear text authentication credentials or other sensitive data over unencrypted communication channels. This could allow an attacker to monitor the network, obtain logon credentials, or obtain and possibly tamper with other sensitive data items.

✓ *Identify Host Threats*

The approach used throughout this guide when configuring host security (that is, Microsoft Windows 2000 or Microsoft Windows Server™ 2003 and .NET Framework configuration) is to divide the configuration into separate categories to allow you to apply security settings in a structured and logical manner. This approach is also ideally suited for reviewing security, spotting vulnerabilities, and identifying threats. Common configuration categories applicable to all server roles include patches and updates, services, protocols, accounts, files and directories, shares, ports, and auditing and logging. For each category, identify potentially vulnerable configuration settings.

✓ *Identify Application Threats*

Having created a security profile that describes how the application handles core areas, such as authentication, authorization, configuration management, and other areas. STRIDE threat categories and predefined threat lists can be applied to scrutinize each aspect of the security profile of your application.

Step-by-step models include:

- *EC-Council's Certified Ethical Hacker Threat Model (http://www.eccouncil.org).*

They identify five stages: Reconnaissance, Scanning, Gaining Access, Maintaining Access, and Covering Tracks.

Phase 1 - Reconnaissance

Reconnaissance is probably the longest phase, sometimes lasting weeks or months. The black hat uses a variety of sources to learn as much as possible about the target business and how it operates, including

- Internet searches
- Social engineering
- Dumpster diving
- Domain name management/search services
- Non-intrusive network scanning

The activities in this phase are not easy to defend against. Information about an organisation finds its way to the Internet via various routes. Employees are often easily tricked into providing tidbits of information which, over time, act to complete a complete picture of processes, organisational structure, and potential soft-spots.

Phase 2 - Scanning

Once the attacker has enough information to understand how the business works and what information of value might be available, he or she begins the process of scanning perimeter and internal network devices looking for weaknesses, including

- Open ports
- Open services
- Vulnerable applications, including operating systems
- Weak protection of data in transit
- Make and model of each piece of LAN/WAN equipment

Scans of perimeter and internal devices can often be detected with intrusion detection (IDS) or prevention (IPS) solutions, but not always. Veteran black hats know ways around these controls.

Phase 3 - Gaining Access

Gaining access to resources is the whole point of a modern-day attack. The usual goal is to either extract information of value to the attacker or use the network as a launch site for attacks against other targets. In either situation, the attacker must gain some level of access to one or more network devices.

Phase 4 - Maintaining Access

Having gained access, an attacker must maintain access long enough to accomplish his or her objectives. Although an attacker reaching this phase has successfully circumvented your security controls, this phase can increase the attacker's vulnerability to detection.

Phase 5 – Covering Tracks

After achieving his or her objectives, the attacker typically takes steps to hide the intrusion and possible controls left behind for future visits.

- Cox and Gerg (2004) Threat Models

It involves stages such as:

Phase 1: Probe

In this phase, the attacker gathers information on a potential target. In a targeted attack, the scanning may be limited to your allocated range of IP addresses. In an untargeted attack, it might be against a wide range of addresses. Often, the initial activities of this information-gathering will not send a single packet to your network. A surprising amount of information can be gathered from information stores on the Internet. The goal of this phase is to map out network and determine details about the systems on the network, permitting the attacker to tailor an attack to exploit known vulnerabilities in the software version running on your system, or perhaps to a configuration error.

Phase 2: Penetrate

Once the systems and potentially vulnerable services have been discovered, the next step is an attack. The attack can take a variety of forms. The attack may cause a system to execute code of the attacker's choice. If the attacker has access as an unprivileged user, the attack may escalate the user account to an administrator-level access. The attack may simply crash a service or entire system.

There are a myriad of penetration methods and the vast bulk of Snort signatures are built to detect them in progress. Automated attacks such as worms or scripts actually combine the Probe and Penetrate phases by simply launching attacks against a range of addresses (which fail against systems that are not vulnerable). If a rule exists that is designed to recognize one of these attacks, Snort will certainly detect these attempts.

Sometimes the attack is hidden in a trojan horse and mdash; usually an attack program hidden in another. The attack sometimes contains a remote control utility that calls back to an attacker, giving the attacker a point of presence inside your network. An entire class of rules exist to watch for Trojan horse traffic.

Phase 3: Persist

Once an attacker goes through the trouble of finding a vulnerable system, locates or builds the attack, and then successfully attacks the machine, it would be a nuisance to have to repeat the process every time he wants to access the system. It may be that between visits, the system gets attention from an administrator and is no longer vulnerable. Launching an attack multiple times against a system increases the chances of being noticed.

As a result, one of the first things an attacker does once a machine has been "owned" is make it easier to get back onto the system. The attacker may create an administrator-level user with a password that only he knows. He may simply acquire the username and password database from the system and crack the passwords using a password cracking utility (like *John the Ripper* or *L0phtcrack*) to decrypt the passwords on their system. Once the passwords are cracked, the attacker can login as whomever he wants.

The attacker may install some remote-control software, too. This makes it easier to work remotely on the system. The most common of these tools is a utility called *netcat*. Netcat is a very flexible remote command-shell utility that is easy to install remotely and can be configured to run on any network port, making it possible to access through a firewall.

Most serious attackers attempt to hide evidence of their activity at this point by altering or deleting system and firewall logs. They may use utilities that hide the directories that hold their attack tools from the eyes of administrators. If the attacker was an automated tool or a network worm, it may copy itself to system files, and ensure that it will survive past a reboot. It may go so far as take steps to hide itself, too.

Phase 4: Propagate

Once the attacker has an established presence on a system, the next move is to see what else is available. The attack phases begin anew with the compromised system as the source of the activity. The attacker will try to map the internal network (or the network that contains the compromised system). The newly enumerated machines will then be attacked, if they are interesting to the attacker. If the attack was a worm, this phase is sometimes the most damaging. The worm attempts to infect (probe and penetrate) other systems on the local network (or systems on the public Internet).

There is a concept called *implied trust*, in which a username and password that works on one system (or group of systems) works on another system. For example, if the system that was compromised is a Linux system, the username and password that works on this system may also work on the organisation's Windows systems, as well. While the concept of "single sign-on" is an administrative aid, it can be a detriment in the event of a successful attack.

The only good news is that, if an attack gets to this point, it may be possible to detect this second round of attacks with your Snort systems.

Phase 5: Paralyze

This is the ultimate goal of a targeted attack, in which the attacker goes after your environment with a goal in mind. The goal may be to steal or destroy data, bring your systems down, or attack another organisation from one of your systems, making you look guilty. The attacker looks for what I call the "soft, chewy center" of your network. This is most often a database that hosts your organisation's proprietary data, financial information, inventory, or even email.

- *Haslum (2010) Threat Model*

The model described different states of systems outside normal behaviour of system. These states include:

State 1: Intrusion Attempt (IA): indicates that one or more attackers are trying to gather information about the system (for possible use in a future intrusion attempt).

State 2: Intrusion in Progress (IP): indicates that one or more attackers have started an attack against the system. The system is still functioning correctly and no confidentiality or integrity breaches have occurred.

State 3: Successful Attack (SA): indicates that one or more attackers have broken into the system and may have full control over the system.

ii. Attack Patterns (Multi-Step-Dynamic-Method)

Attack patterns are the primary tools that security professionals use. They are mostly used in Attack Path and Goal Recognition. They allow in depth analysis of threats, going beyond what are already known. Attack Patterns are a formalized approach to capturing attack information in enterprise systems. Attack patterns are generic representations of commonly occurring attacks that can occur in a variety of different contexts. The pattern defines the goal of the attack as well as the conditions that must exist for the attack to occur, the steps that are required to perform the attack, and the results of the attack.

A category of such tools are Attack Trees, which are conceptual diagrams showing how an asset, or target, might be attacked. Attack trees have been used in a variety of applications. In the field of information technology, they have been used to describe threats on computer systems and possible attacks to realize those threats. However, their use is not restricted to the analysis of conventional information systems. Works on Attack Tree in Threat Modelling include Schneier (2000) Attack Tree Analysis that quantifies the security or vulnerability of a system based on the goals of the attacker.

How Attack Trees works?

While several approaches can be used in practice, the accepted method is to identify goals and sub-goals of an attack, as well as what must be done so that the attack succeeds. When defining the attack trees, security analysts first evaluate the vulnerabilities of the systems and networks, then pretend to be attackers and work out attack plans to achieve the intrusion goals. In this process, an attack tree is extended and branches are built to identify the different sub-goals of the attacker and penetration points available to the attacker. The process continues by decomposing or expanding the means of penetration to the lowest level of intrusion, known as the leaves. An attack tree can represent each opportunity for an attack against a computer system or network. Start building an attack tree by creating root nodes that represent the goals of the attacker. Then add the leaf nodes, which are the attack methodologies that represent unique attacks. Attack trees are multileveled diagrams consisting of one root, leaves, and children. From the bottom up, *child nodes* are conditions which must be satisfied to make the direct parent *node* true; when the *root* is satisfied, the attack is complete. Each *node* may be satisfied only by its direct *child nodes*. Figure 2.9 presents a simple Attack Tree.

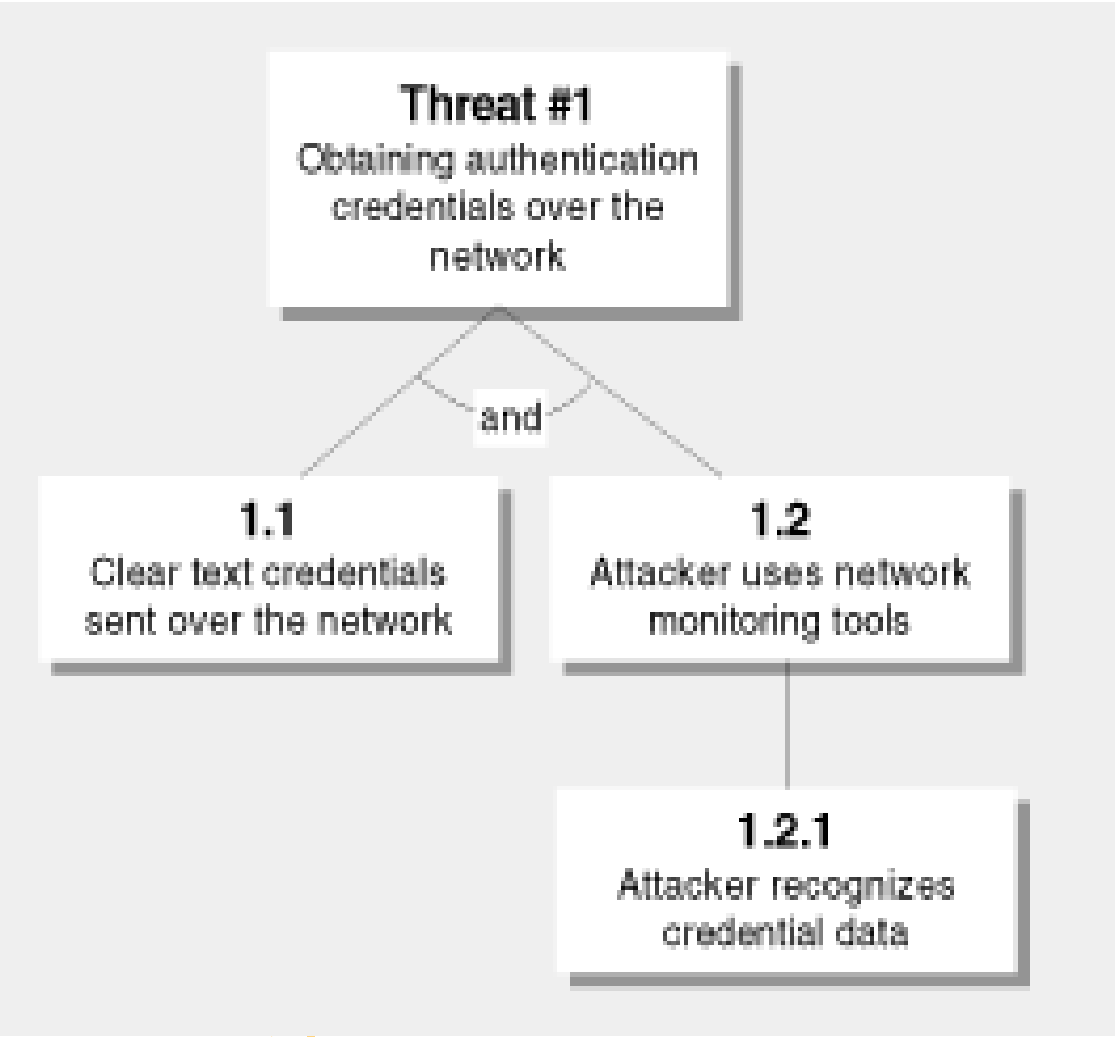

Threat #1
Obtaining authentication credentials over the network
and
1.1
Clear text credentials sent over the network
1.2
Attacker uses network monitoring tools
1.2.1
Attacker recognizes credential data

Figure 2.9: An Attack Tree (Olzak, 2006)

Another tool is Attack Graph. *Attack graphs* depict ways in which an adversary can exploit vulnerabilities to break into a system. System administrators analyze attack graphs to understand where their system's weaknesses lie and to help decide which security measures will be effective to deploy. In practice, attack graphs are produced manually by Red Teams. Construction by hand, however, is tedious, error-prone, and impractical for attack graphs larger than a hundred nodes. There are basically two types of attack graphs. In the first type, each vertex represents the entire network state and the arcs represent state transitions caused by an attacker'sactions. Example is Sheyner's scenario graph based on model checking Sheyner (2004). In the second type of attack graph, a vertex does not represent the entire state of a system but rather a system condition in some form of logical sentence. The arcs in these graphs represent the causality relations between the system conditions. We call this type of attack graph a dependency attack graph. Example is the graph structure used by Ammann et al. (2002).

Attack Graph (Geib and Goldman, 2001; Jha et al. 2002; Ammann et al., 2002; Cuppens *et al.,* 2002; Ning, 2003; Qin and Lee, 2004, and Li *et al.* 2007) in the Network Security Domain largely concentrates on correlation of observed actions and alerts produced by intrusion detection systems. Geib and Goldman (2001) presented a probabilistic model of plan recognition for recognizing and predicting the intentions of the agents based on the construction of execution traces from raw security alerts. This method requires predefined attack plan library and lacks support for reasoning about deceptive actions of an adversary. Ammann *et al.* (2002) developed an algorithm that found l shortest path that can be used to reach goal.

Jha *et al*. (2002) provided a formal characterization of this problem: we prove that it is polynomially equivalent to the minimum hitting set problem and we present a greedy algorithm with provable bounds. By interpreting attack graphs as Markov Decision Processes we can use the value iteration algorithm to compute the probabilities of intruder success for each attack the graph. Cuppens et al. (2002) proposed a method for detecting various steps of an intrusion scenario, which is seen as a planning activity based on a declarative description of actions, goals, and plans.

The method does not provide additional information to distinguish between the most and least plausible scenarios, which is an important feature since the number of possible scenarios can be large. Ning (2003) presented an approach that described vulnerability and IDS alerts. They also presented an approach that group alerts into attack graph scenarios. Benferhat et al. (2003) extended Cuppens' approach by providing the ability to rank possible scenarios. Qin and Lee (2004) proposed a graph-based technique to correlate isolated attack scenarios derived from low-level alerts. Attack trees define attack plan libraries used to correlate isolated alert sets that are converted into causal networks with assigned probability distributions to evaluate the likelihood of attack goals and predict future attacks. Another work was Li *et al.* (2007) Data Mining Approach to Generating Network Attack Graph for Intrusion Prediction. It based the identification of the Threat on Minimum Support Requirements and Longest Path of Attack.

A code-injection attack pattern that is used to describe code injection attacks in a generic way may be abstractly described as presented in Table 2.5.

Table 2.5: Code Injection Attack Pattern

Pattern	**Code injection attacks**
Attack goals	Command or code execution
Required conditions	Weak input validation Code from the attacker has sufficient privileges on the server.
Attack technique	1. Identify program on target system with an input validation vulnerability. 2. Create code to inject and run using the security context of the target application. 3. Construct input value to insert code into the address space of the target application and force a stack corruption that causes application execution to jump to the injected code.
Attack results	Code from the attacker runs and performs malicious action.

e. Prioritisation of Threat

Threat Prioritisation is the process that is used to rank threats based on threat ratings in order to determine security investment on such threat. A very popular threat Prioritisation model is DREAD (Meier et al., 2007), which is presented in Table 2.6. The ratings can fall in the range of 5–15. Then you can treat threats with overall ratings of 12–15 as High risk, 8–11 as Medium risk, and 5–7 as Low risk. By using the DREAD model, we can arrive at the risk rating for a given threat by asking the following questions:

i. **D**amage potential: How great is the damage if the vulnerability is exploited?
ii. **R**eproducibility: How easy is it to reproduce the attack?
iii. **E**xploitability: How easy is it to launch an attack?
iv. **A**ffected users: As a rough percentage, how many users are affected?
v. **D**iscoverability: How easy is it to find the vulnerability?

Another approach to Threat Prioritisation is based on the criteria of Prioritisation. In this, Risk Assessment has been used. The corporate bodies involved in this process of Risk Assessment Standardization include: Consultative Objective Risk Analysis System (CORAS, 2000), ISO/IEC-27005 (2005), Operationally Critical Threat, Asset, and Vulnerability Evaluation (OCTAVE, www.cert.org), COSO-Enterprise Integrated Framework (2004), Australia/New Zealand Standard Risk Management: ISO 31000 (2009), and A Risk Management Standard by the Federation of European Risk Management Associations (FERMA, 2002).

Risk is generally referred to as vulnerability exposed to threat (Naqvi, 2011). It implies that, threat is not necessarily a danger, but an indication of a dangerous situation which could result in willful or unwillful act of exercising /playing on the weakness (vulnerability) of the system in question. A significant threat will be the one that would exploit vulnerability. The classification work performed by Koller, (2000) grouped the risk assessment techniques into five parts: Discriminant Function Analysis (DFA); Bayesian Analysis; Decision Tree

Table 2.6: DREAD Threat Rating

	Rating	High (3)	Medium (2)	Low (1)
D	Damage potential	The attacker can subvert the security system; get full trust authorization; run as administrator; upload content.	Leaking sensitive information	Leaking trivial information
R	Reproducibility	The attack can be reproduced every time and does not require a timing window.	The attack can be reproduced, but only with a timing window and a particular race situation.	The attack is very difficult to reproduce, even with knowledge of the security hole.
E	Exploitability	A novice programmer could make the attack in a short time.	A skilled programmer could make the attack, then repeat the steps.	The attack requires an extremely skilled person and in-depth knowledge every time to exploit.
A	Affected users	All users, default configuration, key customers	Some users, non-default configuration	Very small percentage of users, obscure feature; affects anonymous users
D	Discoverability	Published information explains the attack. The vulnerability is	The vulnerability is in a seldom-used part of the product, and only a few users	The bug is obscure, and it is unlikely that users will work out

	found in the most commonly used feature and is very noticeable.	should come across it. It would take some thinking to see malicious use.	damage potential.

techniques; Factor Analysis (FA) and Neural Nets (NN). Work by individuals and organisation on risk assessment dated to time immemorial. Popular amongst which is the work of Carroll (1983) where the author proposed a ‘familiar risk analysis model’ employing the Annual loss Expectancy (ALE)’ to calculate the risk of an alternative via multiplication of the Annual Rate Occurrence (ARO) with the single loss Expectance (SLE), and Exposure Factor (EF) with the value of assets. Risk Assessment has also been studied in Haslum (2010) Fuzzy-based Incident Risk Assessment, Ahmed et al. (2010) Vulnerability-based Risk Assessment, Dondo (2009) Fuzzy-based Vulnerability Risk Assessment and Jumaat (2012) Multi-strategic Incident Risk Assessment.

Mathematically, Risk can be defined as

$$\textit{Risk (Threat)} = \textit{(Likelihood of Threat)} * \textit{(Consequence of Threat)} \quad \ldots (2.2)$$

where

$$\textit{Threat} = \textit{Intent} * \textit{Capability} \quad \ldots (2.3)$$

Unfortunately, no risk assessment technique has been able to show expressly Threat Intent and Capability in their risk estimation. However, intent and capability have been explored in works such as Ha *et al.* (2006) Insider Threat Model and Bhattacharya *et al.*, (2008) Information-centric Adversary Model. The existing Threat Prioritisation approaches are deployed in Vulnerability Scoring Systems and Alert Prioritisation Systems

f. *Mitigation of Threat*

The information gathered in the previous step is used as input into the mitigation step. What action to take, if any, is based on the severity of the risk scores. If management is evaluating how to apply resources to mitigating risk to multiple systems, the threat risk scores play a large role. Avoid-Transfer-Mitigate-Accept Model (Hillson, 1999) is the most popular. This model is incorporated into Attack Tree in certain cases. Other risk mitigation models include Elimination-Transfer-Retention-Reduction (Baker *et al.*, 1999) and Absorption-Prevention-Contingency (Ben-David and Raz, 2001).

2.8 Collaborative Network Security Management

Collaborative Network Security Management has been proven to be the right tools to address the dynamics of Internet-facilitated Organised Crime Threats (Chen *et al*, 2007; Ntoukas *et al.*, 2007; and Chen *et al.*, 2013), but it has not been applied in modelling and mitigating threats.

However, in other fields of InfoSec, Collaborative-based Threat Modelling have been applied. In Radio Frequency Identification Security, the need for Collaborative Threat Modelling was developed. Ahmadi (2008) developed a platform for logical determination of an RFID system's privacy and security risks based on the assumption that rather than viewing security objective from potentially biased perceptions of each individual part, there was need to work together to view a security objective in a contextual environment. The model's collaborative nature allowed for the input of various factions (vendor, security expert, company executives, consultant and consumer), based on what each perceived to be valid points, and enabled all collaborators to view and comment on all inputs. The model however did not consider Threat Modelling in the context of Internet-facilitated Organised Internet Management or provided mechanism to address uncertainty and multidimensionality of Collaborative Security Architecture.

Recently, Adam Shostack, a Microsoft Threat Modelling Expert emphasized the importance of group meetings involving developers, architect, tester and security professionals for Threat Modelling in Shostack (2014). He pointed out that Threat Modelling could be part of project management and that a mechanism must be provided for managing trust among members of team. However, no mechanism was provided for interoperating information from different team members.

2.8.1 Incident Sharing and Analysis

Within the domain of Collaborative Network Security Management, as far as we know, there is no plausible Incident Sharing and Analysis framework for modelling threats. However, in Cyber Threat Intelligence and Security Information and Event Management, certain frameworks and tools have been developed for Incident Sharing and Analysis. The review below examines the studies.

2.8.1.1 Cyber Threat Intelligence Standards and Tools

Before these tools and standards are reviewed, some basic principles of Cyber Threat Intelligence are discussed.

Cyber threat intelligence (CTI) is threat intelligence related to computers, networks and information technology. Intelligence as defined by Edward Waltz is, "the information and knowledge about an adversary obtained through observation, investigation, analysis, or understanding, is the product that provides battlespace awareness" (Waltz, 1998). Clark (2010) described intelligence as being actionable information. Additionally, cyber threat intelligence can be strategic or tactical. Strategic intelligence includes things like motivation of adversaries. Tactical intelligence includes things like 'tactics, techniques and procedures (TTP)' and 'indicators of compromise (IOCs)'. IOCs are one of the most easily actionable types of CTI and are often the focus standards and tools. Some of the most commonly used IOCs are IP addresses, domain names, uniform resource locators (URLs) and file hashes.

The information and knowledge Collection Strategies are divided into three categories:

a. *Internal Collection Strategies:* The internal threat category encompasses any CTI that is collected from within the organisation. This can included reported information from security tools such as firewalls, intrusion prevention systems (IPS) and host security systems like anti-virus. A valuable source of threat intelligence information comes from computer forensic analysis. The analysis can yield intelligence that is not readily visible and may be very useful in detection of other attacks.

b. *Community Collection Strategies:* The community category includes any CTI shared via a trusted relationship among multiple members with a shared interest. This can be an informal group with member organisations that are in the same industry sector or that have other common interests. There are formal community groups such as the Information Sharing and Analysis Centres (ISACs) Organised under the

National Council of ISACs (NCI, 2013). ISACs are formed for specific sectors such as higher education or financial services. There are over a dozen ISACs under the National Council of ISACs. One example of a community sharing group is Research and Education Networking (REN) ISAC. REN-ISAC is a trusted community for research and higher education. They are the main organisation behind the Collective Intelligence Framework. Another example of a community group is the Defense Industrial Base Collaborative Information Sharing Environment (DCSIE). This group provides a hub for CTI sharing between U.S. government defense contractors.

c. *External Collection Strategies:* The external category includes CTI from sources outside an organisation and not part of a community group. There are two types of external sources. The first is public sources. Public sources are available to anyone and generally there is no cost associated with access. While public feeds can be available at no cost, there can be problems. Amoroso points out data quality problem with volunteered data (Amoroso, 2011). An example of a public CTI feeds is MalwareDomains (MalwareDomains, 2013). MalwareDomains provides a list of domains known to be involved in malicious activity. The lists are available in multiple formats and can be used to block access to the malicious domains.

The other type of an external CTI source is private. Private sources are typically only available on a paid basis. An organisation can subscribe to a threat feed from a vendor to receive regularly updated CTI. These feeds have the advantage in that there may be a service level agreement on data quality. Many security products include some type of cyber threat intelligence update mechanism. CTI services can also be purchased separately. One example is the Emerging Threats ETPro Ruleset (EmergingThreats, 2013). Emerging threats offers subscription services for IDS rules and IP reputation.

On Cyber Threat Intelligence Exchange standards, the most popular are the Internet Engineering Task Force standards: Incident Object Description Exchange Format and Real-time Inter-network Defense. Others include MITRE standards. In deciding the appropriate Cyber Threat Intelligence Exchange Standard to be used in Threat Modelling, the following indicators are used (Saklikar, 2013).

- *Information Leakage:* This has to do with the ability of CTI tool and standard to manage of integrity and privacy of information.
- *Interoperability:* This has to do with the ability of CTI tool and standard to provide rich semantics that support both human and machine parsing.
- *Validation of Information Quality and Reliability:* This concerns the ability of CTI tool and standard to provide quality and trustworthy information.

The standards include:

a. Incident Object Description Exchange Format, RFC (Danyliw et al., 2007)

The Incident Object Description Exchange Format (IODEF) defines a data representation that provides a framework for sharing information commonly exchanged by Computer Security Incident Response Teams (CSIRTs) about computer security incidents.

An attribute is defined as an enumerated value with a default value of "private". In other classes where this attribute is used, no default is specified.

i. public. There are no restrictions placed in the information.
ii. need-to-know. The information may be shared with other parties that are involved in the incident as determined by the recipient of this document (e.g., multiple victim sites can be informed of each other).
iii. private. The information may not be shared.
iv. default. The information can be shared according to an information disclosure policy pre-arranged by the communicating parties.

The IncidentID class represents an incident tracking number that is unique in the context of the CSIRT and identifies the activity characterized in an IODEF Document. This identifier would serve as an index into the CSIRT incident handling system. The combination of the name attribute and the string in the element content must be a globally unique identifier describing the activity. Documents generated by a given CSIRT must not reuse the same value unless they are referencing the same incident.

The Incident class has three attributes:

Name: Required. STRING. An identifier describing the CSIRT that created the document. In order to have a globally unique CSIRT name, the fully qualified domain name associated with the CSIRT MUST be used.

Instance: Optional. STRING. An identifier referencing a subset of the named incident.

Restriction. Optional. ENUM.

IODEF is used in a number of projects and vendor products. A successful implementation of IODEF is used by the Anti-Phishing Working Group. They have extended the IODEF standard to support the reporting of phishing and other email incidents. It is used as a storage format in the Collective Intelligence Framework (CIF Project, 2009). IODEF is also used in products from DFLabs, Arcsite and Foundstone (Moriarty, 2013). The problems with the tools are that some of the default information may disclose certain level of privacy; it does not provide mechanism for ensuring trust among the exchange partners and cater for some additional information. The incident class is presented in Figure 2.10.

b. IODEF for Structured Cyber Security Information, RFC 7203 (Takahashi, 2013)

IODEF for Structured Cyber Security Information" (IODEF-SCI) is an extension to the IODEF standard that supports additional data. It is a standard proposed by the MILE working group (Takahashi, 2013). The additional information includes: attack pattern, platform information, vulnerability, weakness, countermeasure instruction, computer event log, and severity. IODEF-SCI supports the additional data by embedding existing standards within the IODEF document.

```
+------------------+
| IncidentID       |
+------------------+
| STRING           |
|                  |
| STRING name      |
| STRING instance  |
| ENUM restriction |
+------------------+
```

Figure 2.10: The IODEF Incident Class (Danyliw *et al., 2007)*

```
+---------------+
| Incident      |
+---------------+
| ENUM purpose  |<>---------[IncidentID]
| STRING        |<>--{0..1}-[AlternativeID]
|   ext-purpose |<>--{0..1}-[RelatedActivity]
| ENUM lang     |<>--{0..1}-[DetectTime]
| ENUM          |<>--{0..1}-[StartTime]
|   restriction |<>--{0..1}-[EndTime]
|               |<>---------[ReportTime]
|               |<>--{0..*}-[Description]
|               |<>--{1..*}-[Assessment]
|               |<>--{0..*}-[Method]
|               |      |<>--{0..*}-[AdditionalData]
|               |         |<>--{0..*}-[AttackPattern]
|               |         |<>--{0..*}-[Vulnerability]
|               |         |<>--{0..*}-[Weakness]
|               |<>--{1..*}-[Contact]
|               |<>--{0..*}-[EventData]
|               |      |<>--{0..*}-[Flow]
|               |      |  |<>--{1..*}-[System]
|               |      |     |<>--{0..*}-[AdditionalData]
|               |      |        |<>--{0..*}-[Platform]
|               |      |<>--{0..*}-[Expectation]
|               |      |<>--{0..1}-[Record]
|               |         |<>--{1..*}-[RecordData]
|               |            |<>--{1..*}-[RecordItem]
|               |               |<>--{0..*}-[EventReport]
|               |<>--{0..1}-[History]
|               |<>--{0..*}-[AdditionalData]
|               |      |<>--{0..*}-[Verification]
|               |      |<>--{0..*}-[Remediation]
+---------------+
```

Figure 2.11: The IODEF-SCI Incident Class

The following standards are proposed to be included in IODEF-SCI: Common Attack Pattern Enumeration and Classification (CAPEC), Common Event Expression (CEE), Common Platform Enumeration (CPE), Common Vulnerability and Exposures (CVE), Common Vulnerability Reporting Format (CVRF), Common Vulnerability Scoring System (CVSS), Common Weakness Enumeration (CWE), Common Weakness Scoring System (CWSS), Open Checklist Interactive Language (OCIL), Open Vulnerability and Assessment Language (OVAL), Extensible Configuration Checklist Description Format (XCCDF), Distributed Audit Service (XDAS) and ISO/IEC 19770.

This standard defines eight extension classes, namely Attack Pattern, Platform, Vulnerability, Scoring, Weakness, Event Report, Verification, and Remediation. The UML representation of the Incident classes is presented in Figure 2.11. The standard has not been applied to any public tool. The problems with the standard are that some of the default information may disclose certain level of privacy and it does not provide mechanism for ensuring trust among the exchange partners. However, it caters for additional security risk-related information that could ensure quality.

c. *Real-time Inter-network Defense (RID), RFC (Moriarty, 2012)*

Real-time Inter-network Defense (RID) outlines a proactive inter-network communication method to facilitate sharing incident-handling data while integrating existing detection, tracing, source identification, and mitigation mechanisms for a complete incident-handling solution. Combining these capabilities in a communication system provides a way to achieve higher security levels on networks. Policy guidelines for handling incidents are recommended and can be agreed upon by a consortium using the security recommendations and considerations. RID functions via five message types: Request, Acknowledgement, Result, Report and Query. The RID standard includes a Policy Class which would allow different policies to be applied based on the relationship with the sharing parties. Some of the relationships considered are Client-to-SP (Service Provider), SP-to-Client, Intra-Consortium, Peer-to-Peer and Between-Consortiums. This flexibility would allow for direct organisation to organisation sharing via the Peer-to-Peer relationship or within a community using the Intra-Consortium relationship. The problems with the standard are that some of the default information may disclose certain level of privacy; it does not provide mechanism for ensuring trust among the exchange partners and caters for additional security risk-related information that could ensure quality.

d. MITRE Standards: CybOX, STIX, TAXII (in Farnham, 2013)

MITRE developed three standards that each fills different needs for a Cyber Threat Intelligence. The first is Cyber Observable eXpression (CybOX), which provides a standard for defining indicator details known as observables. The second is Structured threat Information Expression (STIX) which provides a standard to define patterns of observables in context. The third is Trusted Automated eXchange of Indicator Information (TAXII) which provides a standard to exchange Cyber Threat Intelligence. It has been adopted as a planned standard by Microsoft as part of its 'Microsoft Active Protections Program' (MAPP) (Bryant, 2013). TAXII is also in use by Financial Services Information Sharing Analysis Centre (FS-ISAC) (Connolly, 2013).

In Figure 2.12, a Use-Case of STIX is presented. The problems with the tools are that some of the default information may disclose certain level of privacy; it does not cater for additional security risk-related information. In Figure 2.13, we present an example of Mapping of Cyber Threat Intelligence to Predictive Modelling Mark-up Language.

2.8.1.2 Security Information and Event Management Systems (SIEM)

Security Information and Event Management (SIEM) makes use of the Event Management Tools to monitor, identify, document and respond to security threats and reduce false positive incidents (Miller *et al.*, 2010). SIEM is a technology which provides real-time monitoring of multiple security appliances and historical reporting of security events from networks, systems and/or applications (Nicolett and Kavanagh, 2009). It can be seen as a new approach for enhancing the IDS and Firewall technologies.

AlienVault (http://alienvault.com) is a Security Iinformation and Event Management System. AlienVault's Unified SIEM provides SIEM, vulnerability assessment, network and host intrusion detection, and file integrity monitoring functions via software or appliance options. AlienVault Unified SIEM is composed of proprietary and open-source components. Open Source Security Information Monitoring (OSSIM) is an open-source security management platform. It provides support for NetFlow. Its unified SIEM lacks native support for Database Active Management (DAM) and there is no integration with third-party technologies.

CorreLog (https://correlog.com) integrates log management and SIM functions and provides basic capabilities. It targets midsize businesses, and have been validated with small deployments in the range of 50 to 75 servers. The solution includes agent-based event filtering and file integrity monitoring for Windows, Unix, and Linux platforms. CorreLog does not provide event source integration for packaged applications. CorreLog does not provide event source integration for third-party DAM technologies, but there is limited support for monitoring

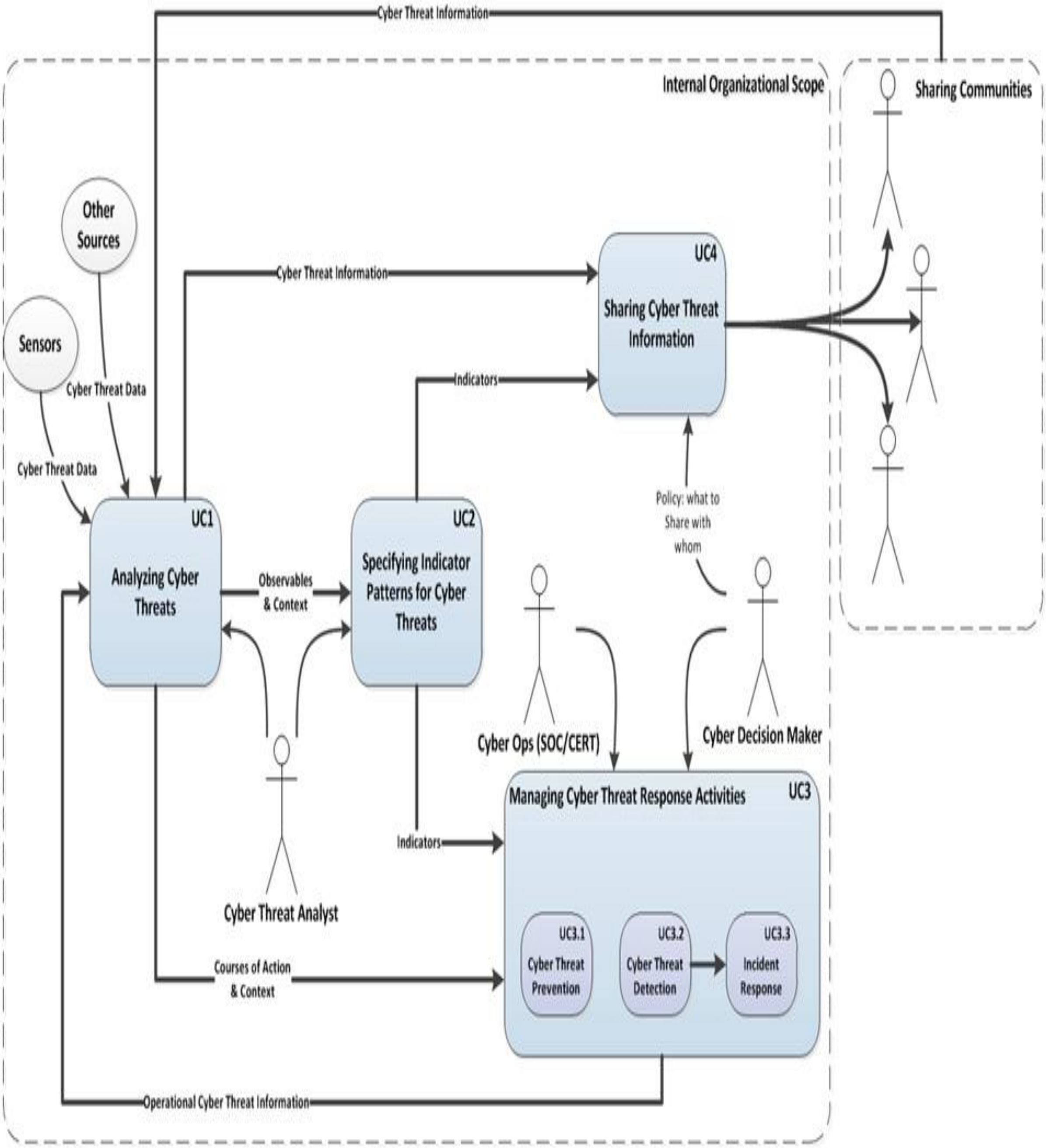


Figure 2.12: STIX Use Case Model (Farnham, 2013)

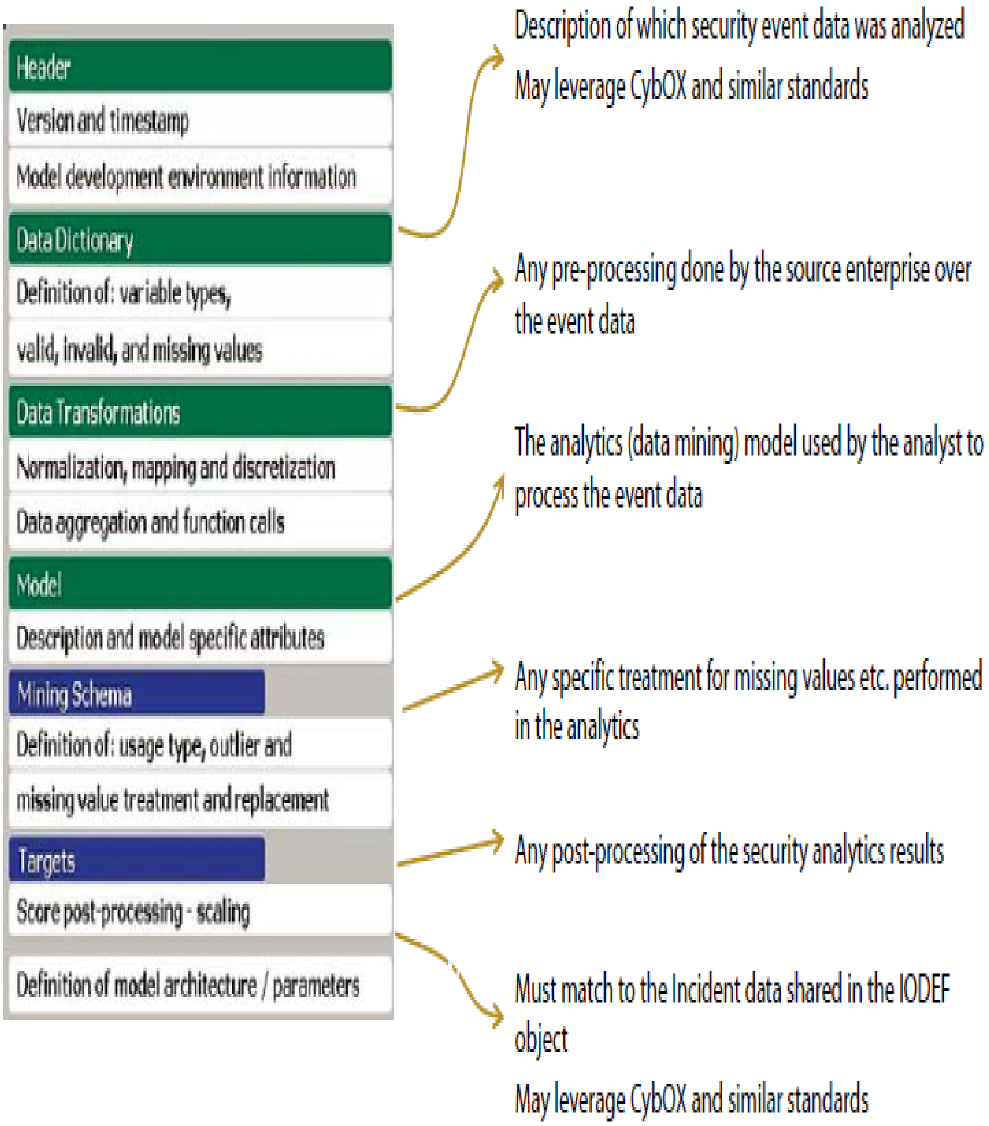

Header
Version and timestamp
Model development environment information
Data Dictionary
Definition of: variable types,
valid, invalid, and missing values
Data Transformations
Normalization, mapping and discretization
Data aggregation and function calls
Model
Description and model specific attributes
Mining Schema
Definition of: usage type, outlier and
missing value treatment and replacement
Targets
Score post-processing - scaling
Definition of model architecture / parameters
Description of which security event data was analyzed
May leverage CybOX and similar standards
Any pre-processing done by the source enterprise over the event data
The analytics (data mining) model used by the analyst to process the event data
Any specific treatment for missing values etc. performed in the analytics
Any post-processing of the security analytics results
Must match to the Incident data shared in the IODEF object
May leverage CybOX and similar standards

Figure 2.13: Mapping of Sample Threat Modelling to Cyber Threat Intelligence (Farnham, 2013)

database activity through native audit functions. In fact, CorreLog's predefined compliance reporting is limited to Payment Card Industry (PCI) only.

IBM's Tivoli Security Information and Event Manager (TSIEM) (http://www.ibm.com) software provides SIM and Security Event Monitoring functionality, and allows customers to have a starting point with log management. TSIEM provides capabilities for privileged user monitoring, compliance reporting, log management and basic real-time SEM. A typical deployment is focused on user activity monitoring and involves 100 or fewer servers. TSIEM integrates with a wide set of IBM and third-party Integrity and Access Monitoring technologies and applications. The technology is not well-suited for moderate or large deployments that require network security monitoring.

OSSEC is an open source host-based intrusion detection system(HIDS) (http://www.trendmicro.co.uk). It is a scalable, multi-platform, open source Host-based Intrusion Detection System (HIDS). It has a powerful correlation and analysis engine, integrating log analysis, file integrity checking, Windows registry monitoring, centralized policy enforcement, rootkit detection, real-time alerting and active response. It runs on most operating systems, including Linux, OpenBSD, FreeBSD, MacOS, Solaris and Windows." The OSSEC HIDS can be installed as a stand-alone tool to monitor one host or can be deployed in a multi-host scenario, one installation being the server and the others as agents. The server and agents communicate securely using encryption. OSSEC also has intrusion prevention features, being able to react to specific events or set of events by using commands and active responses. The system allows the creation of new commands which can be bound to events. The system comes with some predefined active response tools, but the administrator can add others.

McAfee IntruShield network security products (http://www.mcafee.com) delivers an integrated hardware and software solution, which delivers comprehensive detection *and* protection from known, first strike (unknown), DoS, and DDoS attacks from several hundred Mbps to multi-gigabit speeds. The architecture integrates patented signature, anomaly, and Denial of Service detection on a single purpose-built appliance. This not only enables highly accurate detection, but also empowers administrators with smart tools and processes, and enables flexible and scalable deployment for global businesses and vital government agencies. The IntruShield architecture employs a combination of threshold-based and patented self-learning, profile-based detection techniques that delivers unmatched intelligence to intrusion detection. With straightforward threshold-based detection, administrators can configure data traffic limits to ensure their servers will not become unavailable due to overload. Its self-learning methodologies enable studying of the patterns of network usage and traffic over time. It did not address most of the challenges of incident sharing and analysis.

Ning *et al.* (2003) presented the development of TIAA, a visual toolkit for intrusion alert analysis. TIAA is developed to provide an interactive platform for analyzing potentially large sets of intrusion alerts reported by heterogeneous intrusion detection systems (IDSs). To ensure timely response from the system, TIAA adapts main memory index structures and query optimization techniques to improve the efficiency of intrusion alert correlation. TIAA includes a number of useful utilities to help analyze potentially intensive intrusion alerts, including alert aggregation/disaggregation, clustering analysis, focused analysis, frequency analysis, link analysis, and association analysis. Moreover, TIAA provides several ways to visualize the analysis results, making it easier for a human analyst to understand the analysis results. It did not address uncertainty and trust issues affecting incident sharing and analysis.

In Ullrich (2004), DShield was discussed. DShield aggregates firewall and intrusion detection system logs from networks throughout the global Internet. Each log entry provided by a network represented one or more packets that violated a local rule. DShield transforms all of the logs into a normalized form. Each entry in the DShield trace includes: time-detected, submitter's ID, count, source IP, source port, destination IP, destination port, protocol exploited, and flags. The source IP can be used for identifying a malicious/infected scanning source if the IP address is not spoofed. Broadly speaking, the DShield trace provided a unique opportunity to extract the spatial-temporal characteristics of attacking machines. It did not state how it ensured trust.

Kang *et al.* (2004) provided the design, evaluation, and deployment of Sequoia, a robust communication architecture for distributed Internet-scale security monitoring systems. Sequoia supports a rich set of communication patterns for regional and global sharing of monitor observations, collaborative decision-making among monitors, and timely delivery of security information to monitors. Highly secure communication is achieved through a comprehensive set of security mechanisms for trust management of participating monitors and trust-based routing. In addition, Sequoia offers high-quality and reliable communication services using a scalable self-organizing structure that is resilient and adaptive. Sequoia's communication architecture supports aggregation, integration, and dissemination of blacklists using a publisher-subscriber paradigm. Sequoia comprises three key protocols through which monitors self-organize into a two-level hierarchy on which scalable, fast and trustworthy message delivery can be achieved: The Monitor Neighbour Discovery Protocol (MND) is used to form a topology-aware flat overlay among monitors, with every monitor connected to nearby nodes as its neighbours. The goal of the Distributed Dominator Selection Protocol (DDS) is to form a two-level communications hierarchy from the flat neighbour overlay constructed by MND. A monitor in the higher level of this hierarchy (dominators) must meet minimum requirements regarding trustworthiness and routing performance. The Communication Path Discovery Protocol (CPD) discovers multiple delivery paths from one or more senders to one or more destinations, considering both efficiency and security constraints. This is achieved by mapping the highly trusted dominator nodes into a structured overlay network. Sequoia addressed some of the challenges of incident sharing and analysis, especially trust and interoperation.

Yegneswaran *et al.* (2004) described and evaluated DOMINO, a cooperative intrusion detection system. DOMINO was designed to enable intrusion information sharing in a globally distributed network consisting of: trusted axis nodes Organised in a peer-to-peer overlay, satellite nodes associated with each axis node that are hierarchically arranged, terrestrial nodes, which are deployed at the leaves of the infrastructure, that provide daily intrusion summaries. DOMINO's design was based on heterogeneous data collection through NIDS, firewalls and active-sinks. This architecture enables DOMINO to be secure, scalable, fault tolerant, and facilitates data sharing. The evaluation clearly demonstrated the utility of sharing information between multiple nodes in a cooperative infrastructure. They used an information-theoretic approach to show that perspective on intrusions can be greatly enhanced by cooperation of a relatively small number of nodes. Using the 2002 and 2003 SQL-worm outbreaks, it was demonstrated that false-alarm rates can be significantly reduced in DOMINO and that reaction time for outbreak detection can be similarly reduced. Finally, the work provided an initial evaluation of the effectiveness of active-sinks in discriminating between types of attacks based on examining payload data. The results clearly demonstrate that active-sinks provide important insight in this regard. DOMINO offered a significant opportunity to improve intrusion mitigation using collaborative peer-to-peer nodes. However, it did not address information uncertainty management.

Locastor *et al.* (2005) presented Worminator, which extracts relevant information from alert streams and encodes it in Bloom Filters. This information forms the basis of a distributed watchlist. The watchlist can be distributed via a choice of mechanisms ranging from a centralized trusted third party to a decentralized P2P-style overlay network. They adopted two mechanisms in order to cope with the difficulties of distributed correlation and the potential volume of data being correlated. The Bloom filters by Worminator is employed to protect the confidentiality of the data being exchanged between domains. Second, efficient information exchange is accomplished with a distributed correlation scheduling algorithm. The scheduling algorithm dynamically calculates subsets of correlation peers that should communicate to exchange Bloom filters. Since information is also compacted by the Bloom filter, correlation between peers becomes extremely cost-effective in terms of bandwidth and processing power. The Worminator addressed privacy and interoperability.

Chen *et al.* (2007) presented a new distributed approach for detecting DDoS flooding attacks at the traffic flow level. The defence system was suitable for efficient implementation over the core networks operated by Internet Service Providers (ISP). At the early stage of a DDoS attack, some traffic fluctuations were detectable at Internet routers or at gateways of edge networks. A Distributed Change-point Detection (DCD) architecture was developed using Change Aggregation Trees (CAT). The idea was to detect abrupt traffic changes across multiple network domains at the earliest time. Early detection of DDoS attacks minimizes the flooding damages to the victim systems serviced by the provider. The system was built over attack-transit routers, which worked together cooperatively. Each ISP domain had a CAT server to aggregate the flooding alerts reported by the routers. CAT domain servers collaborate among themselves to make the final decision. To resolve policy conflicts at different ISP domains, a Secure Infrastructure Protocol (SIP) was developed to establish the mutual trust or consensus. Sixteen network domains were simulated on the DETER testbed. Experimental results showed that 4 network domains were sufficient to yield a 98% detection accuracy with only 1% false-positive alarms. The security coverage was wide enough to safeguard most ISP core networks from real-life DDoS flooding attacks. The work did not address privacy and uncertainty issues.

Ntoukas *et al.* (2011) presented a collaborative network security management platform called Storm to improve security in distributed and complex information Systems with critical data and services. This platform makes use of advanced open source technologies and interactive software tools. The tool was applied to Port Information Systems security and the results show the effectiveness of Collaborative Network Security Management in Distributed System. It however did not address issues such as interoperability, privacy, uncertainty, and quality.

The aim of Chen *et al.* (2013) was to mitigate Botnets, which consisted large number of bots that generate huge volumes of spam or launch Distributed Denial of Service (DDoS) attacks on victim hosts. To address these problems, a practical collaborative network security management system was proposed with an effective collaborative

Unified Threat Management (UTM) and traffic probers. A distributed security overlay network with a centralized security center leverages a peer-to-peer communication protocol used in the UTMs collaborative module and connects them virtually to exchange network events and security rules. Security functions for the UTM were retrofitted to share security rules. In the work, they proposed the design and implementation of a cloud-based security centre for network security forensic analysis. The cloud storage kept collected traffic data and enabled processing of data with cloud computing platforms to find the malicious attacks. The cloud based security centre could instruct each collaborative UTM and prober to collect events and raw traffic, send them back for deep analysis, and generate new security rules. These new security rules were enforced by collaborative UTM and the feedback events of such rules are returned to the security centre. By this type of close-loop control, the collaborative network security management system could identify and address new distributed attacks more quickly and effectively. The Collaborative Network Security Management System did not address uncertainty and trust issues posed by incident sharing and analysis.

The survey of the works is based on the following indices:

a. *Management of Privacy:* This indicated the methods used to manage confidentiality of information shared among multiple network security management domains.

b. *Management of Interoperability:* This indicated the level of unification of operations performed at different network security management domains.

c. *Management of Multidimensionality:* This indicated the mode of integration and scalability of shared information.

d. *Management of Quality:* This indicated the level of relevance of information shared by the events and incidents.

e. *Management of Uncertainty:* This indicated the methods used to manage ignorance and divers beliefs associated with different network security management domains.

f. *Management of Trust:* This indicated the kind of measure put in place to alleviate distrust among unfamiliar Collaborative network security management domains.

Table 2.7 presents the survey of the Collaborative Network Security Management Systems.

Table 2.7: Comparison of Network Threat Management Systems based on Management of Event/Incident Sharing and Analysis Issues

S/N	Tool	Privacy Management	Interoperability	Multidimensionality Reduction	Quality	Uncertainty Elimination	Trust Management
1	**AlienVault** (Alien Vault, http://www.alienvault.com)	✓	✓	X	X	X	✓
2	**CorreLog** (CorreLog, https://correlog.com)	✓	✓	X	X	X	X
3	**IBM Tivoli Security Operation Manager** (IBM, *www.ibm.com*)	✓	✓	X	X	X	X
4	**McAfee IntruShield Security Manager** (McAfee, *www.mcafee.com*)	✓	✓	X	X	X	X
5	**OSSEC** (Trend Micro, Inc, *www.trendmicro.co.uk*)	✓	✓	✓	X	X	X
6	**STORM (**Ntouskas *et al.*, 2011)	✓	✓	✓	✓	X	X
7	**DShield** (Ullrich, 2004)	✓	✓	X	X	X	X
8	**TIAA** (Ning *et al.*, 2003)	✓	✓	✓	✓	X	X
9	**SEQUOIA** (Kang *et al.*, 2004)	✓	✓	X	✓	X	✓
10	**Worminator** (Locasto *et al.*, 2005)	✓	✓	✓	✓	X	✓
11	**DOMINO** **(**Yegneswaran et al., 2004)	✓	✓	✓	X	X	✓
12	**Collaborative Network Security Management for Forensic Analysis** (Chen *et al.*, 2007)	✓	✓	✓	✓	X	X

2.9 Related Works

The review of existing works on Threat Modelling and Mitigation in Network Threat Management is presented as follows:

Caswell and Roesch (1998) developed Snort, which is one of the most popular open source security tools. Snort runs in different modes: Sniffer mode; Packet Logger mode; NIDS mode; and Inline (IPS) mode. Working as an IDS, Snort uses preprocessors and rules. *Snort Preprocessors* allow the functionality of Snort to be extended by allowing users and programmers add modular plug-ins. While Snort does not offer a GUI, there are many complementary open-source tools like Analysis Console for Intrusion Detection (PHP-based), Sguil, or BASE (Basic Analysis and Security Engine) which provide the GUI functionality for Snort to be able to perform Network Threat Management. Snort offers intrusion standard classification scheme for intrusions and prioritise threats using predefined integer values between 1 and 4. With Snort, threats can be identified, prioritised and mitigated. However, the classification and prioritisation mechanism are not suitable for complex and dynamic threats such as Internet-facilitated Organised Crime Threats.

Dondo (2009) presented a fuzzy systems approach for assessing the relative risk associated with computer network assets. He used the approach to rank vulnerabilities so that analysts can prioritise their work based on the potential risk exposures of assets and networks and associated vulnerabilities to individual assets, and therefore networks. Fuzzy models of the vulnerability attributes were developed in which fuzzy rules is used to make an inference on the risk exposure and the likelihood of attack, which allows ranking of the vulnerabilities and shows which ones need more immediate attention. The work did not address threat identification while the Threat Prioritisation used only vulnerability information to rate threats.

According to Mell *et al.* (2009), Common Vulnerability Scoring System is standard approach used to quantitatively analyse vulnerabilities and rank risk between 0 and 10. It can qualitatively described risk as low, medium and high. It based its risk estimation on three factors: base factors, temporal factors and environmental factors.

This approach has the advantage that it takes into consideration vulnerability attributes, and uses them to calculate a score for relative comparison. However, CVSS's rough estimates of the number of assets affected by vulnerability, its course-grained inclusion of asset values and the limited variability of its temporal metrics makes its vulnerability prioritisation less accurate. Also, it is limited by the fact that its risk estimation was based on the presence of availability of Common Vulnerability and Exposure Identification.

Ahmed *et al.* (2010) presented Risk based proactive seCurity cOnfiguration maNAger (ROCONA). They proposed a security metric framework that quantifies objectively the most significant security risk factors, which include existing vulnerabilities, historical trend of vulnerability of the remotely accessible services, prediction of potential vulnerabilities for any general network service and their estimated severity and finally propagation of an attack within the network. The risks were obtained based on the information in National Vulnerability Databases. The approach addressed the dynamism of threats through prediction of vulnerabilities and attack pattern recognition. This improved the accuracy and confidence of threat modelling. The framework focused on modelling and mitigation of Major Threats.

Porras *et al.* (2002) described a mission-impact-based approach for the analysis of security alerts produced by spatially distributed heterogeneous information security (INFOSEC) devices, such as firewalls, intrusion detection systems, authentication services, and antivirus software. The objective of the work was to deliver an automated capability to reduce the time and cost of managing multiple INFOSEC devices through a strategy of topology analysis, alert Prioritisation, and common attribute-based alert aggregation. They developed a prototype system called the Mission Impact Intrusion Report Correlation System, or MCorrelator. M-Correlator was intended to provide analysts (at all experience levels) a powerful capability to automatically fuse together and isolate those INFOSEC alerts that represent the greatest threat to the health and security of their networks. Once translated to an internal incident report format, INFOSEC alerts are augmented, and, where possible, fused together through a chain of processing. A relevance score was produced through a comparison of the alert target's known topology against the vulnerability requirements of the incident type, which was provided to M-Correlator by an Incident Handling Fact Base. Next, a priority calculation was performed per alert to indicate the degree to which the alert was targeted at critical assets and the amount of interest the user had registered for this alert type. Last, an overall incident rank was assigned to each alert, which brings together the priority of the alert with the likelihood of success. Once ranked, the M-Correlator attempted to combine related incident alarms with an attribute-based alert clustering algorithm. The resulting correlated incident stream represents a filtered, lower-volume, content-rich security-incident stream, with an incident-ranking scheme that allows the analyst to identify those incidents that pose the greatest risk to the monitored network. The M-Correlator was able to combine information from different sources but did not address or state how it addressed the issues that affect this kind of framework. Also, no mechanism was develop to address bias modelling and mitigation of Minor Threats.

Yu *et al.* (2004) proposed a general collaborative architecture for multiple IDS products by combining intelligent agents and knowledge-based alert evaluation. They evaluated the alert priority, based on asset characteristics, and they used it as the input to their correlation system. No mechanism was developed to address bias modelling and mitigation of Minor Threats and it did not address or state how it addressed the issues that affect the Collaborative framework.

Årnes *et al.* (2006) proposed a network risk assessment using several strategies including examining the composition of risks to the individual host and applying the Hidden Markov Model (HMM) to represent the likelihood of transitions between security states. The model was static and so could not address the continuously emerging threats.

Alshubi *et al.* (2008) proposed a fuzzy-logic based technique for scoring and prioritizing alerts generated by intrusion detection systems. In addition, they presented an alert rescoring technique that led to further reduction of the number of alerts. The IDS alerts were evaluated based on a number of criteria representing the seriousness of the alerts. A Fuzzy Logic Inference Mechanism was developed to score alerts. The approach was therefore applied to the alerts generated by scanning in DARPA 2000 LLDOS 1.0 dataset which successfully prioritized the most critical alerts along with their preparation steps. They did not addressed how alert priority changes with time, that is action based alerts.

A very popular Threat Model is DREAD (Meier et al., 2007). It makes use of a static Threat Modelling approach.The ratings can fall in the range of 5–15. The risk determination factors are organised into five descriptions. **D**amage potential: How great is the damage if the vulnerability is exploited? **R**eproducibility: How easy is it to reproduce the attack? **E**xploitability: How easy is it to launch an attack? **A**ffected users: As a rough percentage, how many users are affected? **D**iscoverability: How easy is it to find the vulnerability? It usually makes use of STRIDE Threat Identification Model Hernan *et al.* (2006) to identify threats. As such, it is not suitable for modelling complex scenario threats.

Data mining approach was applied in generating attack graphs in Li *et al.* (2007). through Association Rule Mining without training, the algorithm generated multi-step attack patterns from historical intrusion alerts which comprised the attack graphs. The algorithm also calculated the predictability of each attack scenario in the attack graph which represented the probability for the corresponding attack scenario to be the precursor of future attacks. The algorithm predicted most major threats with very high accuracy and confidence; however, minor threats were predicted less accurately with low confidence.

Haslum (2010) developed Distributed Intrusion Prediction and Prevention System. A Probabilistic Hidden Markov Model (HMM) that captures the interaction between the attacker and the network was provided. The interaction between various Distributed IDS and integration of their output were achieved through a HMM. He modelled the interaction between the attackers and the system using a Markov model and assumed the system to be in one of the following states: Normal (N) indicating that there is no on-going suspicious activity, Intrusion Attempt (IA) indicating suspicious activity against the network, Intrusion in Progress (IP) indicating that one or more attacker have started an attack against the system, and Successful Attack (SA) one or more attackers have already broken into the system. By using a Markov model, he assumed that next state transition only depend on current state. The risks of the predicted attacks were estimated based on severity, resistance, frequency, etc using fuzzy logic. The risks determined the response options. The prediction was static while the prioritisation relied on expert knowledge which is scarce in network security domains.

Another data mining technique *to* discover, visualize, and predict behavioural pattern of attackers in a network based system was developed by Katipally *et al.* (2010). They proposed a system that was able to discover temporal pattern of intrusion which revealed behaviours of attackers using alerts generated by Intrusion Detection System (IDS). They used data mining techniques to find the patterns of generated alerts by generating Association rules. Their system was able to stream real-time Snort alerts and predict intrusions based on the learnt rules. The algorithm is not suitable for complex scenario attack and emerging threats.

Jumaat (2012) proposed a framework for modelling risk through incident prioritisation and responding to the intrusion. It prioritised and responded to incident using their urgency and criticality. A Risk Index Model (RIM) was used to estimate the risk while a Response Strategy Model (RSM) dynamically maps incidents into different types of response, with serious incidents being mapped to active responses in order to minimise their impact, while incidents with less impact have passive responses. Through the results gathered, the study demonstrated that that alerts priorities change with time and prioritisation process can feasibly be used to facilitate the response selection process in Intrusion Response Systems. However, the incident prioritisation scheme did not address bias against Minor Threats while the response applied a single sensor.

The survey of works on modelling and mitigation of Threats is based on the following indices:

a. *INFOSEC sources:* This stands for the number of information security devices or vulnerability sources. They were either single or multiple sensors.

b. *Administrator:* Population of administrators that participated in the administration of security. They were either be single or multiple.

c. *Point of Analysis:* It represented the location of threat analysis. They were either central or distributed.

d. *Stage of Threat Analysis:* This referred to the point at which the analysis takes place. They were either by pre-incident or post-incident. The pre-incident analysis is also referred to as Predictive Analysis.

e. *Perspective of Threat Analysis:* This referred to the point of view in which threat analysis were performed. The perspectives were either Attacker or Victim.

f. *Type of Threat Identified:* These were the kinds of threats that were identified. These types included: Minor, Major and All.

g. *Method of Threat Identification:* This is the method that was used to recognize and understand the threat. They were mainly by Single Step, Step-by-Step and Attack Pattern. Some attack patterns were based on predictive analysis.

h. *Type of Threat Prioritised:* These were the kinds of threats that are prioritised. These types included: Minor, Major and All.

i. *Method of Threat Prioritisation:* These are the methods that were used to rate threats. They included Vulnerability-based Threat Prioritisation Severity-based Threat Prioritisation, Likelihood-based Threat Prioritisation, and Risk-based Threat Prioritisation.

j. *Type of Threat Mitigation:* These are the kinds of threats that were mitigated. These types include: Minor, Major and Arbitrary Threat Mitigation.

k. *Method of Threat Mitigation:* These were the method used to select the configuration options. This included Arbitrary, Cost-effective and Cost-benefit.

Table 2.8 presents the survey of existing works on modelling and mitigation of Threats.

Table 2.8: Survey of Existing Network Threat Management

S/N	Tool	INFOSEC Sensor/Sources	Administrator	Point of Analysis	Mode of Analysis	Perspective of Analysis	Stage of Threat Analysis	Type of Threat Identified	Method of Threat Identification	Type of Threat Prioritised	Method of Threat Prioritisation	Type of Threat Mitigation	Focus of Threat Mitigation
1	**DREAD (**Microsoft Inc in Meier et al., 2006**)**	Single	Single	Central	Offline	Victim	Pre-incident	All	Single Step	All	Risk	NA	NA
2	**Collaborative Architecture** (Yu et al., 2004)	Multiple	Single	Central	Online	Victims	Post-incident	All	Attack Pattern (Predictive)	All	Risk	NA	NA
3	**CVSS** (Mell et al., 2009**)**	Multiple	Single	Central	Offline	Victim	Pre-incident	All	Single Step	All	Vulnerability	Arbitrary	Arbitrary
4	**Fuzzy System Approach** (Dondo, 2009)	Multiple	Single	Central	Offline	Victim	Pre-Incident	All	Single Step	All	Vulnerability	Major	Cost-effective
5	**SNORT (**Caswell and Roesch, 1998)	Single	Single	Central	Online	Victim	Pre-incident	All	Single Step	All	Severity	Arbitrary	Arbitrary
6	**Incident Prioritisation for Intrusion Response** (Jumaat, 2012)	Single	Single	Central	Online	Victim	Post-Incident	All	Single Step	All	Risk	Major	Cost-effective
7	**ROCONA** (Ahmed *et al.*, 2010)	Single	Single	Central	Online	Victim	Post-incident	Major	Attack Pattern (Predictive)	Major	Risk	Major	Cost-effective
8	**DIPPS** (Haslum, 2010)	Multiple	Single	Central	Online	Attacker	Post Incident	All	Attack Pattern	All	Risk	Major	Cost-effective
9	**M-Correlator** (Porras et al., 2002)	Multiple	Collaborative	Distributed	Online	Victim	Post-incident	All	Single step	All	Risk	Major	Cost-effective
10	**FuzMet**	Multiple	Single	Central	Online	Victim	Post-incident	All	Step-by-step	All	Risk	Minor	Cost-

	(Alsubhi et al., 2009)											(Scanning)	effective
11	**Network Risk Assessment** Årnes *et al.* (2006)	Single	Single	Central	Online	Victim	Pre-incident	All	Step-by-Step	All	Risk	Major	Cost-effective
12	**Sequential Association Mining without Training** (Li et al., 2007)	Multiple	Single	Central	Offline	Attacker	Post-incident	All	NA	NA	NA	NA	NA
13	**Sequential Association Mining with Training** **(**Katipally *et al.*, 2010).	Multiple	Single	Central	Offline	Attacker	Post-incident	All	NA	NA	NA	NA	NA

2.10 Remarks

The following strengths and shortcomings are observed in the reviewed works.

Sequential Association Mining Algorithms of Li *et al.* (2007) and Katipally *et al.* (2010) were able to predict scenario threats dynamically. Li *et al.* (2007) Sequential Association Mining Algorithm without Training performed better than Katipally *et al.* (2010) Sequential Association Mining Algorithm with Training in predicting threats and recognising attack paths. It performed well with Major Threats in simple attack scenario of LLDOS 1.0 by yielding minimum confidence above 0.5. However, it performed poorly with Minor Threats in the same scenario by yielding maximum confidence of 0.26. Therefore, Sequential Association Mining Algorithm without Training was adopted in this study with modifications to predict actionable Minor Threats from different networks accurately.

All the works reviewed were biased in prioritising Minor Threats leading to inaccurate ratings. Haslum (2010), Dondo (2009) and Alsubhi et al., (2009) however prioritised threats and addressed Information Reconciliation, Fusion and Uncertainty using Fuzzy Logic, which needed expert knowledge, large data or prior information. These requirements are scarce in network security domain. A Belief Function that does not need such requirements and could reconcile, fuse and remove uncertainty was applied on strategic risk-determination factors, which are selected from Hybrid-centric Threat Modelling based on Attacker and Victim Perspectives of Intrusion, to prioritise Minor Threats.

Existing works focusing on Cost-effectiveness mitigated only Major Threats to ensure compliance with the scope of Network Threat Management. None of the reviewed works mitigated harmful Minor Threats. Hence, the standard Risk Mitigation Model of Hillson (1999) was adapted to allow for mitigation of harmful Minor Threats from Internet-facilitated Organised Crime Threats.

All the SIEMs performed well by effectively detecting threats. Chen *et al.* (2007) Collaborative-based change point detection for DDoS, Ntoukas *et al.* (2011) Storm, and Chen *et al.* (2013) Cloud-based Collaborative Network Security Management for Forensic Analysis performed well in effectively managing

Internet-facilitated Organised Crime Threats. However, they were not applied to Threat Modelling involving Minor Threats and would not manage all the Incident Sharing and Analysis Issues such as Privacy, Multidimensionality, Uncertainty, Trust, Interoperability and Quality. Hence, a new Collaborative Network Security Management Framework involving multiple network security managers, multiple sensors and multiple networks that addressed the issues of Incident Sharing and Analysis, accurately modelled and cost-effectively mitigated Minor Threats was developed.

The study therefore bridged the gaps by:

a. Modification of Li *et al.* (2007) Sequential Association Mining Algorithm by adding actionable attributes and setting support to highest possible level in order to improve the accuracy of predicting actionable Minor Threats.

b. Reconciliation of Information derived from McHugh *et al.* (2001) Attacker and Victim Perspective of Intrusion-based Hybrid-centric Threat Model using Dempster-Shafer Theory (Shafer, 1976), and their Fusion using Expectation Theory (Ross, 2007) to improve the accuracy of prioritising Minor Threats.

c. Adaptation of Hilson (1999) Risk Mitigation Model to mitigate harmful Minor Threats from distributed Information Security sensors without affecting the scope of Network Threat Management.

d. Development of a new Collaborative Network Security Management framework with centralised sharing and analysis unit to manage trust, interoperability, privacy, uncertainty, quality and multidimensionality over collaborative network security management domains.

CHAPTER THREE

RESEARCH METHODOLOGY

This chapter presents the Conceptualisation of Collaborative Network Security Management Framework for Event Sharing, Analysis and Security Configuration, Development of Modelling and Mitigation Models for Minor Threat and Experimental Design, which include creation of Internet-facilitated Organised Crime Threats and Collaborative Network Security Management Testbed.

3.1 Development of Collaborative Network Security Management Framework for Event Sharing, Analysis and Security Configuration

The methodology for this work is premised on the fact that the collaboration of local Network Security Management domains will assist in comprehensive Threat Modelling. The Collaborative Framework consists of Event Sharing, Event Analysis and Security Configuration Components. The components are Organised as a Server-Client Architecture consisting of a Central Administrative System, which serves as the Server and Local Network Security Management domains that are the Clients. The Event Sharing Component has Data Collection and Information Sharing Units while the Event Analysis has Threat Prediction Unit and Threat Prioritisation Unit.

3.1.1 Event Sharing Component

The Event Sharing Component has the data collection and the information sharing model.

3.1.1.1 Data Collection Model

Due to the strength of Incident Object Description Exchange Format for Structured Cyber Security Information (IODEF-SCI) *(Takahashi, 2013)* in providing additional information, which is important to our model, we operationalised incident data layout consisting of the following fields for the proposed Event Sharing Model.

The Takahashi (2013) IODEF-SCI consists of the following Event and Incident Class attributes:

- *Incident_ID*
- *Alternative_ID*
- *Related_Activity*
- *Detect_Time*
- *Start_Time*
- *End_Time*
- *Report_Time*
- *Assessment*
- *Method*
- *Event_Data*
- *History*
- *Additional_Data*

The IODEF-SCI data model is customized as Event Fact Base with the following attributes as presented in Table 3.1.

3.1.1.2 Information Sharing Model

In Figure 3.1, the Layout of the Information Sharing Model is presented. The Collaborating Network Security Managers submit incident information to the Central Controller Fact Base in Structured Query Language (SQL) interoperability format such as Comma Separated Value (.csv) and Extensible

Table 3.1: Event Fact Base

S/N	**Category**	**Feature**	**Description**
1	Alert	Event_ID	The serial number of event and incident
		Alternative_ID	Other identification number of incident
		Related Activity	Description of Incident
		Start_Time	The first time the incident was detected
		End_Time	The last time the incident was detected
		Detect_Time	The time the incident was detected by InfoSec
		Report Time	The time the alert was generated
		Source_Port	The traced port where incident originate
		Source_IP	The traced IP where incident originate
		Dest_Port	The expected victim's port
		Dest_IP	The expected victim's IP
2	System Data	Asset	Name of the Platform or Package
		Asset Category	Category of the Asset
		Attack Pattern	Description of Asset Service/ Port that is vulnerable
		InfoSec	Name of Information Security Products
		InfoSec Configuration	Configuration of InfoSec device
3	Internet Sources	Vulnerability	Asset Vulnerability Information
		Weakness	Asset Weakness Information

4	User	Contact	Contact Address of User
		History	The History of the Incident in the domain
		Additional Information	Any other information

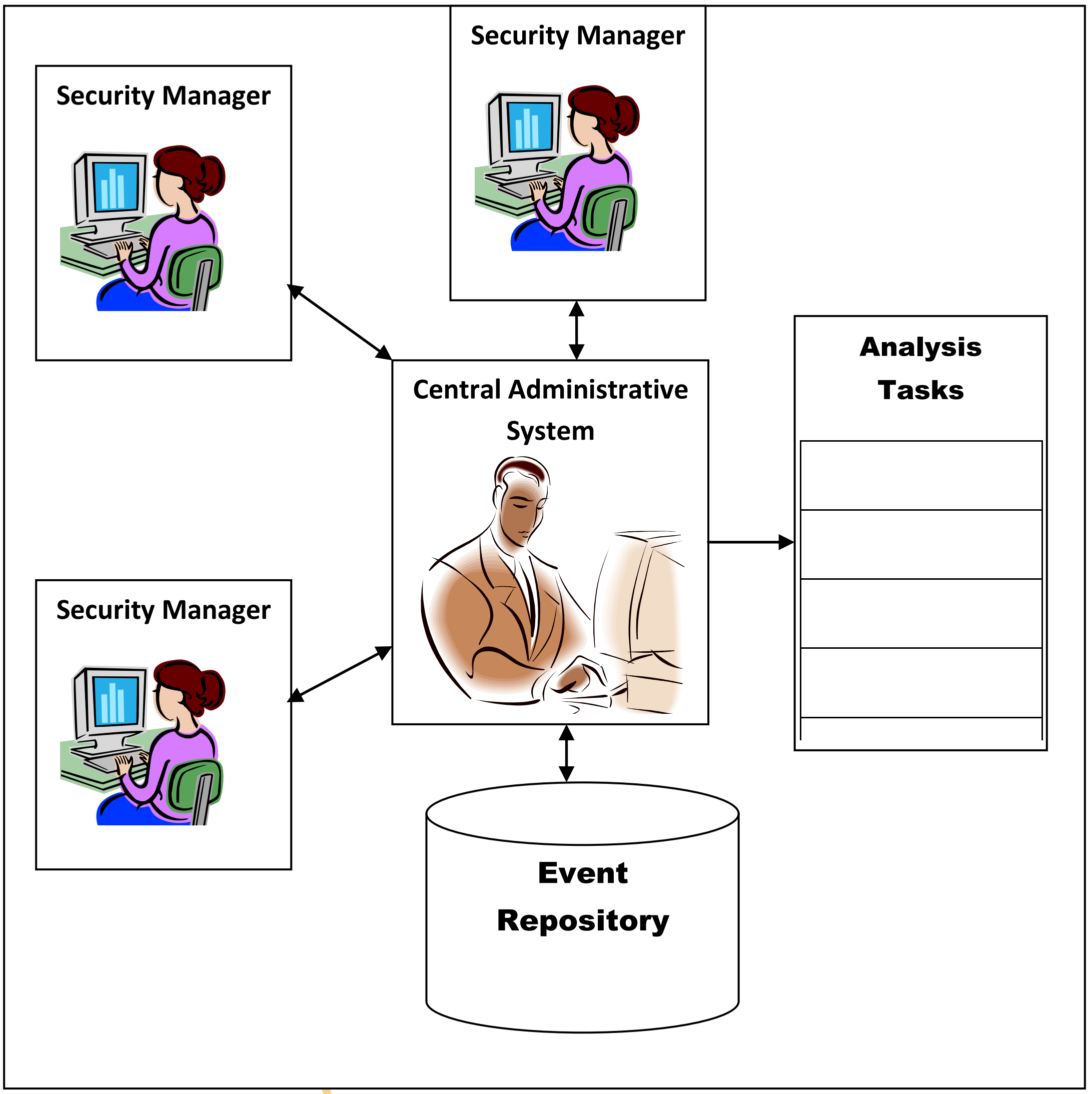


Figure 3.1: Layout of Information Sharing Model

Mark-up Language (.xml). The Central Administrative System filters the information and performs analysis based on the request of the Managers. The outcomes of the analyses are reported by the Central Administrative System to the Security Managers.

3.2 Event Analysis Component

The Event Analysis Component of the Collaboration Framework consists of Threat Prediction and Threat Prioritisation Units.

3.2.1 Threat Prediction

In Figure 3.2, the Layout of Collaborative-based Threat Prediction Unit is presented. The Central Administrative System performs Data Mining activities, which is summarized into Data Pre-processing, Data Mining and Interestingness Analysis. The Local Network Security Management Domains receive the results of the data mining via their contacts and make use of them in managing the Internet-facilitated Organised Crime Threats.

3.2.2 Threat Prioritisation

The Threat Prioritisation unit consists of Attacker and Victim-based Threat Rating, Threat Rating and Ranking components. The layout for Threat Prioritisation is presented in Figure 3.3. The Attacker-based Threat Rating Component consists of Vulnerability Measurement, Vulnerability Reconciliation, Attacker-based Threat Rating units. The Vulnerability Measurement unit uses the Attacker's Perspective of Intrusion Detection to characterise Vulnerability. The Vulnerability Reconciliation unit uses Dempster-Shafer Decision Fusion Technique to map qualitative value of vulnerability criteria to quantitative value. The Threat Rating units rate Threat with respect to the asset criticality using Expectation Theory.

The Victim-based Threat Rating component consists of Event Measurement, Event Reconciliation and Victim-based Threat Rating units. The Event Measurement unit uses the direct Victim's Perspective of Intrusion Detection to

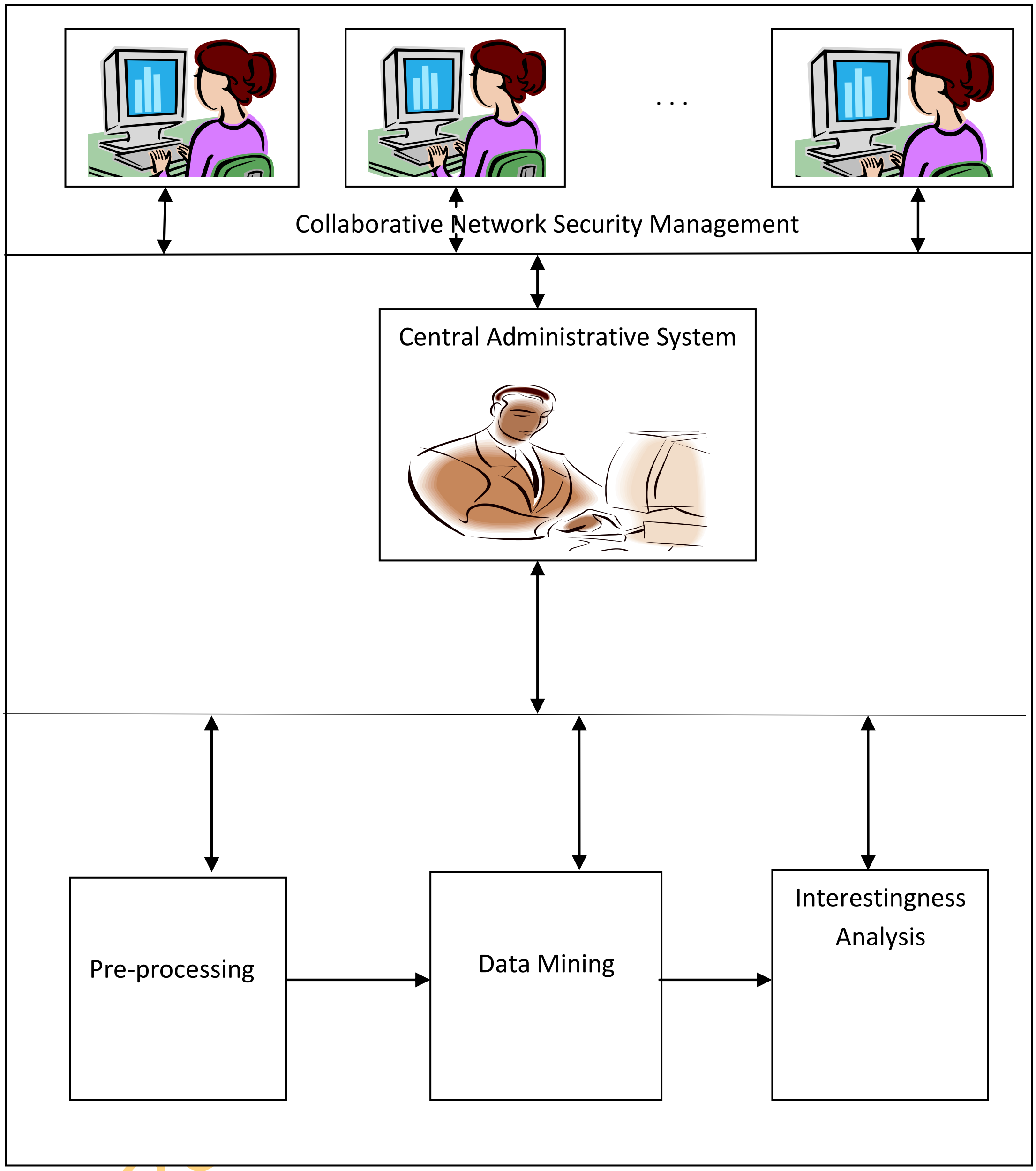

. . .
Collaborative Network Security Management
Central Administrative System
Pre-processing
Data Mining
Interestingness Analysis

Figure 3.2: Layout of Threat Prediction Unit

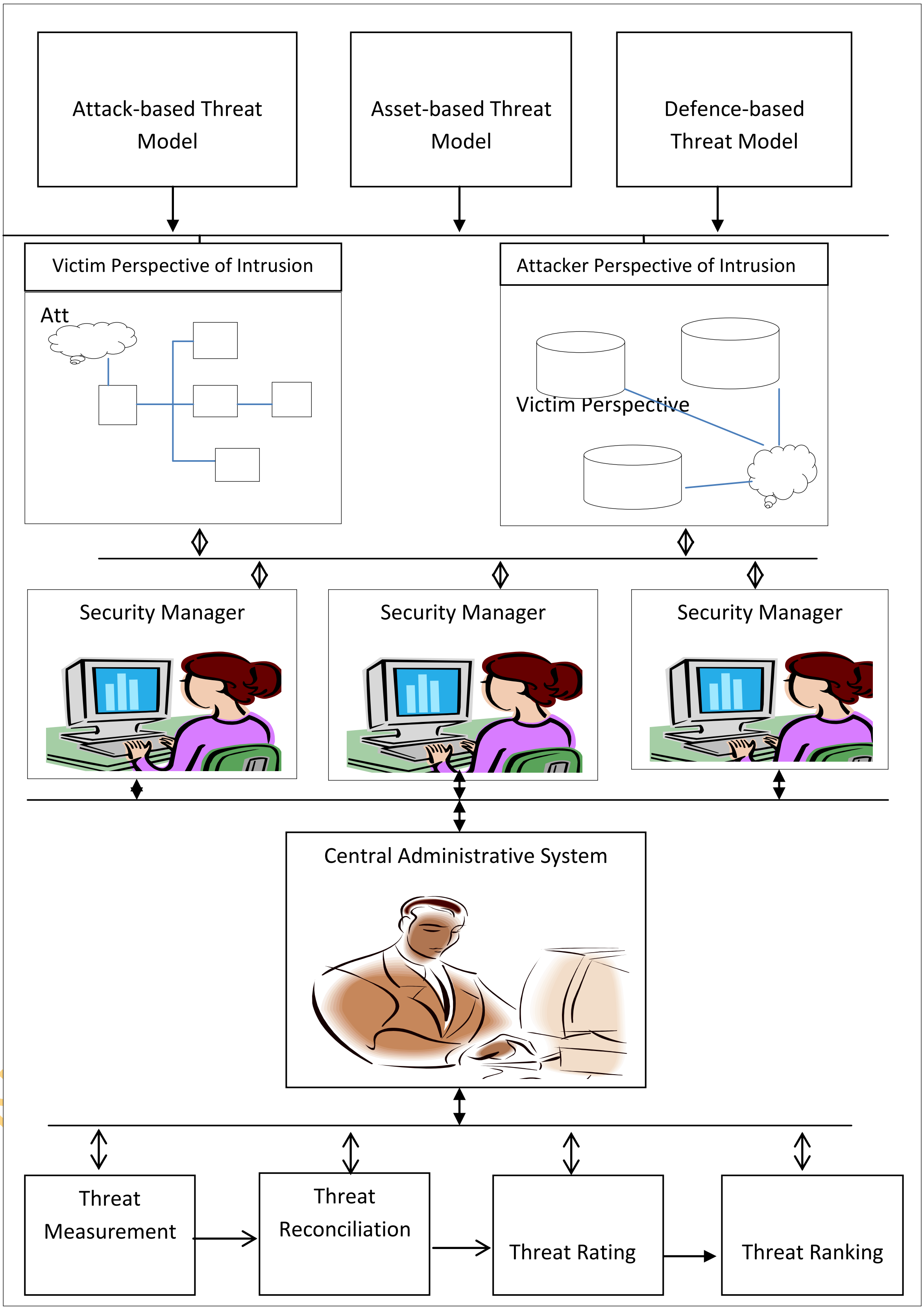


Figure 3.3: The Threat Prioritisation Unit

characterise Threat. The Event Reconciliation unit uses Dempster-Shafer Decision Fusion Technique to map qualitative value to quantitative value. The Threat Rating unit rate Threat with respect to the asset criticality using Expectation Theory. The justifications for the assumptions are that: an attacker does not have privilege in a victim system but uses exploit codes or tools to observe the vulnerability and takes steps to achieve his objectives; he only launches through the perceived vulnerability; his proven profiles are published in vulnerability sites and social networks (direct measurement) while a victim system is made up of assets, InfoSec sensors and security policy (defence); it keeps records of events and incidents in the alert information log which are observed by the InfoSec sensors.

3.2.3 Threat Mitigation

As presented in Figure 3.4, the reputable Risk Control Security Configuration advices are provided by the Central Administrative System on both the Minor Threats and Major Threats. Each Network Security Managers uses the feedback from the Central Administrative System to perform cost-effective Network Threat Management.

In Figure 3.5, the General Architecture of the Collaborative Network Security Management Framework is presented.

a. *Central Administrative System:* This is the most significant element in the architecture. It is responsible for ensuring security of the elements, certainty of measurement, trust among partnering security administrative domains, reconciliation of multiple dimensions and administration of other functions.

b. *Local Console:* This serves as user interface that a Network Security Manager uses to communicate with other functions. The requests of the managers are acted upon by the Central Administrator.

c. *Distributed InfoSec Sensors:* These consist of devices that are used to secure the network and systems in each domain under the purview of each Network Security Manager. These may include Network Intrusion Detection and Prevention Systems.

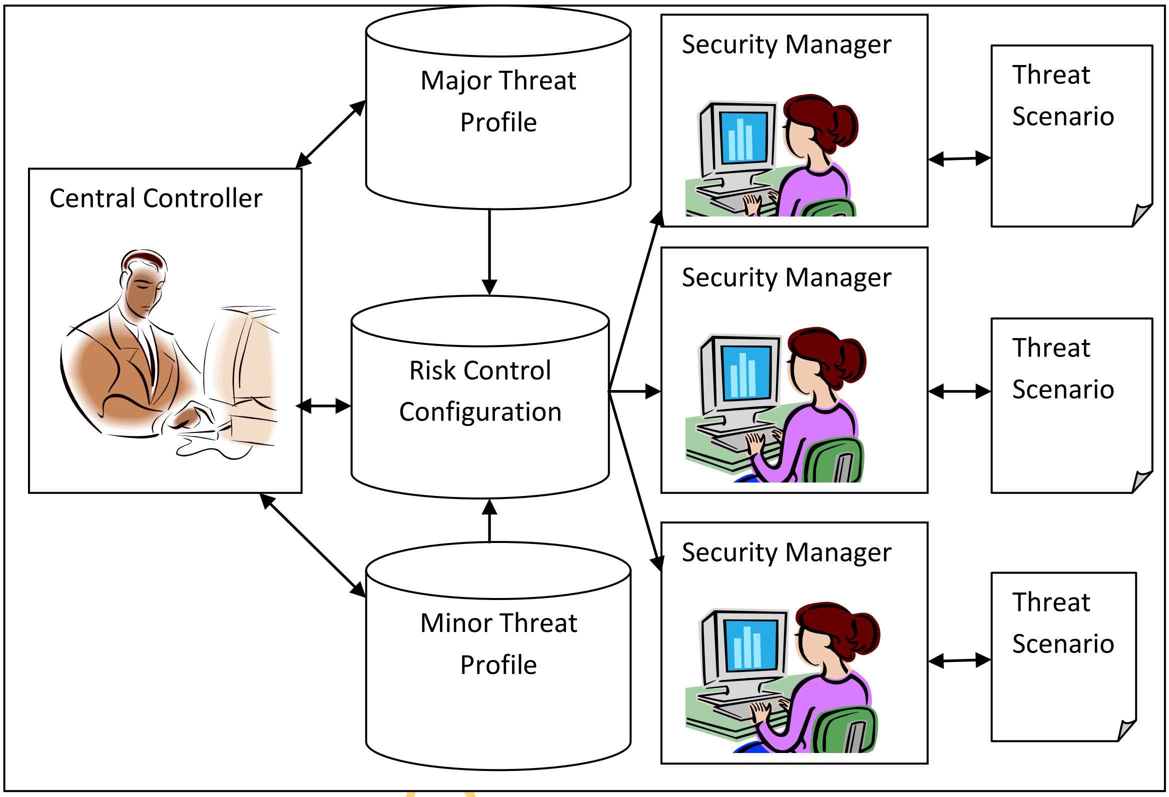


Figure 3.4: Layout of Threat Mitigation Unit

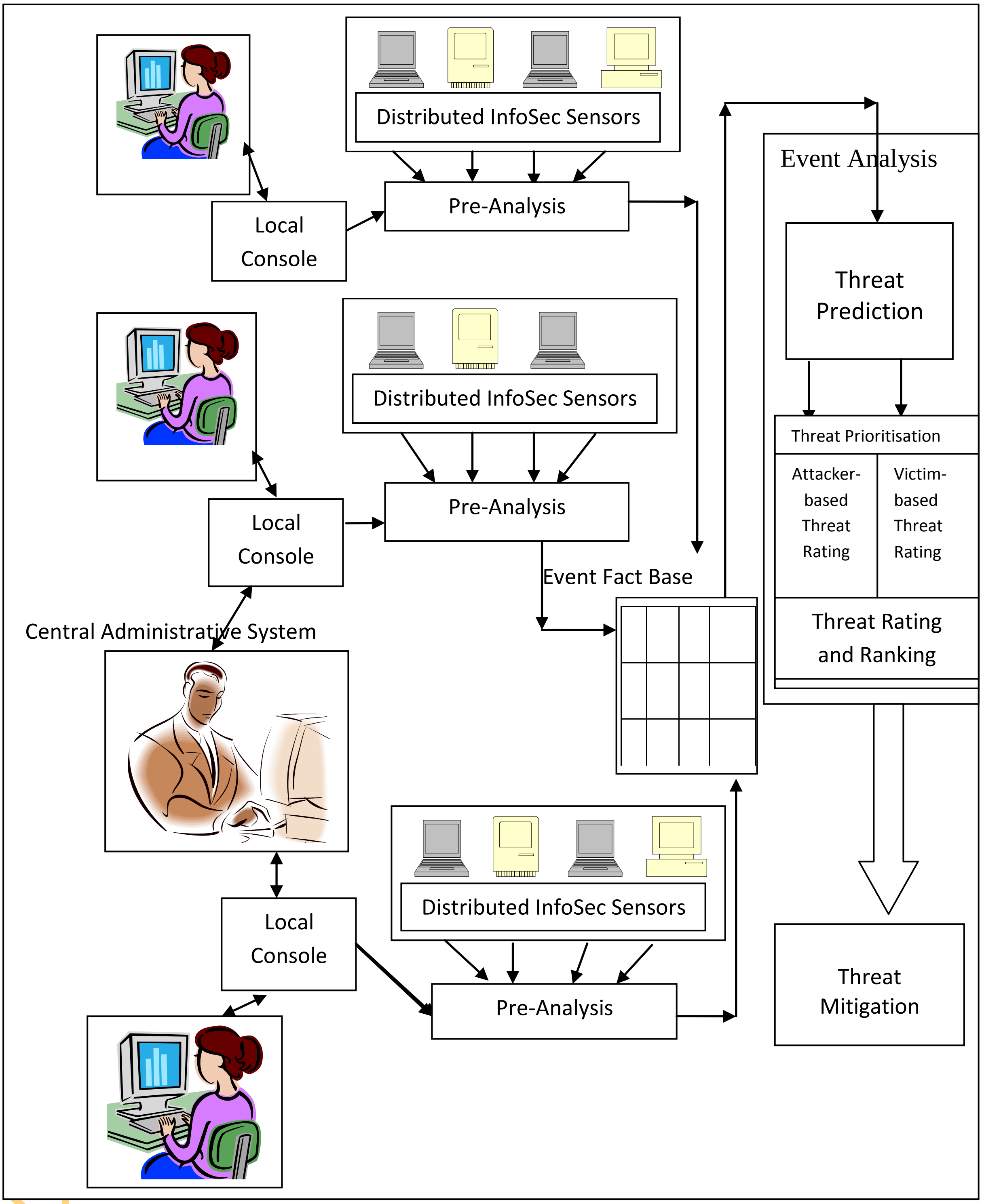

Distributed InfoSec Sensors
Pre-Analysis
Local Console
Distributed InfoSec Sensors
Pre-Analysis
Local Console
Event Fact Base
Central Administrative System
Distributed InfoSec Sensors
Local Console
Pre-Analysis
Event Analysis
Threat Prediction
Threat Prioritisation
Attacker-based Threat Rating
Victim-based Threat Rating
Threat Rating and Ranking
Threat Mitigation

Figure 3.5: Collaborative Network Security Management Framework for Event Sharing, Analysis and Security Configuration

d. *Pre-Analysis:* This is where the Network Security Managers collect incidents from internal sources and search for more information from external sources; it is at the place that the assets are categorized, systems are decomposed and vulnerabilities are identified. The tools that are used by local security managers include Vulnerability Assessment Tools and Incident Analysis Tools.

e. *Event Fact Base*: This is a database of relevant information about events. The database contains different tables, which are related. Such information include time of detection, source port, source IP, destination port, destination IP, incident name, InfoSec sensor name, InfoSec sensor configuration, vulnerability reference, vulnerable service, asset, asset importance, project resources, etc.

f. *Threat Prediction*: The component consists three functions: Event and Incident Pre-processing, Data Mining and Incident Interestingness Analysis. The first function ensures that the data is scaled and presented in the format acceptable for data mining. The data mining is used to extract sequential association rules from the event data. Incident Interestingness Analysis is for the purpose of evaluating the predictability of sequential rules based on their support and confidence.

g. *Threat Prioritisation:* The components consists three functions: Attacker-based Threat Rating, Victim-based Threat Rating, and Threat Rating and Ranking sub-components.

i. *Attacker-based Threat Rating:* The sub-component consists of Vulnerability Measurement, Vulnerability Reconciliation, Attacker-based Threat Rating units. The Vulnerability Measurement unit uses the Attacker's Perspective of Intrusion Detection to characterise Vulnerability. The Vulnerability Reconciliation unit uses Dempster-Shafer Decision Fusion Technique to map qualitative value of vulnerability criteria to quantitative value. The Threat Rating units rate Threat with respect to the asset criticality using Expectation Theory.

ii. *Victim-based Threat Rating*: The component consists of Threat Measurement, Threat Reconciliation, Victim-based Threat Rating units. The Threat Measurement unit uses the direct Victim's Perspective of Intrusion

Detection to characterise Threat. The Threat Reconciliation unit uses Dempster-Shafer Decision Fusion Technique to map qualitative value to quantitative value. The Threat Rating unit rate Threat with respect to the asset criticality using Expectation Theory.

iii. *Threat Rating and Ranking*: It involves the rating of threats by summing Attacker-based Threat rates and Victim-based Threat rates. The scores are ranked based on the Network Threat Management requirements or scope.

h. *Threat Mitigation:* This is an element of the model, which is used to select Risk Control measures. It is based on Avoid-Transfer-Mitigate-Accept Model of Hillson (1999).

3.3 Model Development of the Hybrid-centric Threat Modelling Approach

The Olzak (2006) Threat Modelling and Fayyad *et al.* (1996) Knowledge Discovery in Databases are integrated to formulate the Hybrid-centric Threat Modelling. The framework is presented in Figure 3.6.

3.3.1 Threat Prediction Model

The Threat Prediction Model is designed by modifying Li *et al*. (2007) Sequential Association Mining Algorithm. In addition to timestamp and event name, other actionable attributes such as source IP address and destination IP address are included in creating an event instance. In this section, the Data Mining Model for Threat Prediction is presented.

3.3.1.1 Data Mining Model for Threat Prediction

The data mining technique has three parts:

a. Data Pre-processing

b. Sequential Association Generation

c. Rule Interestingness Estimation

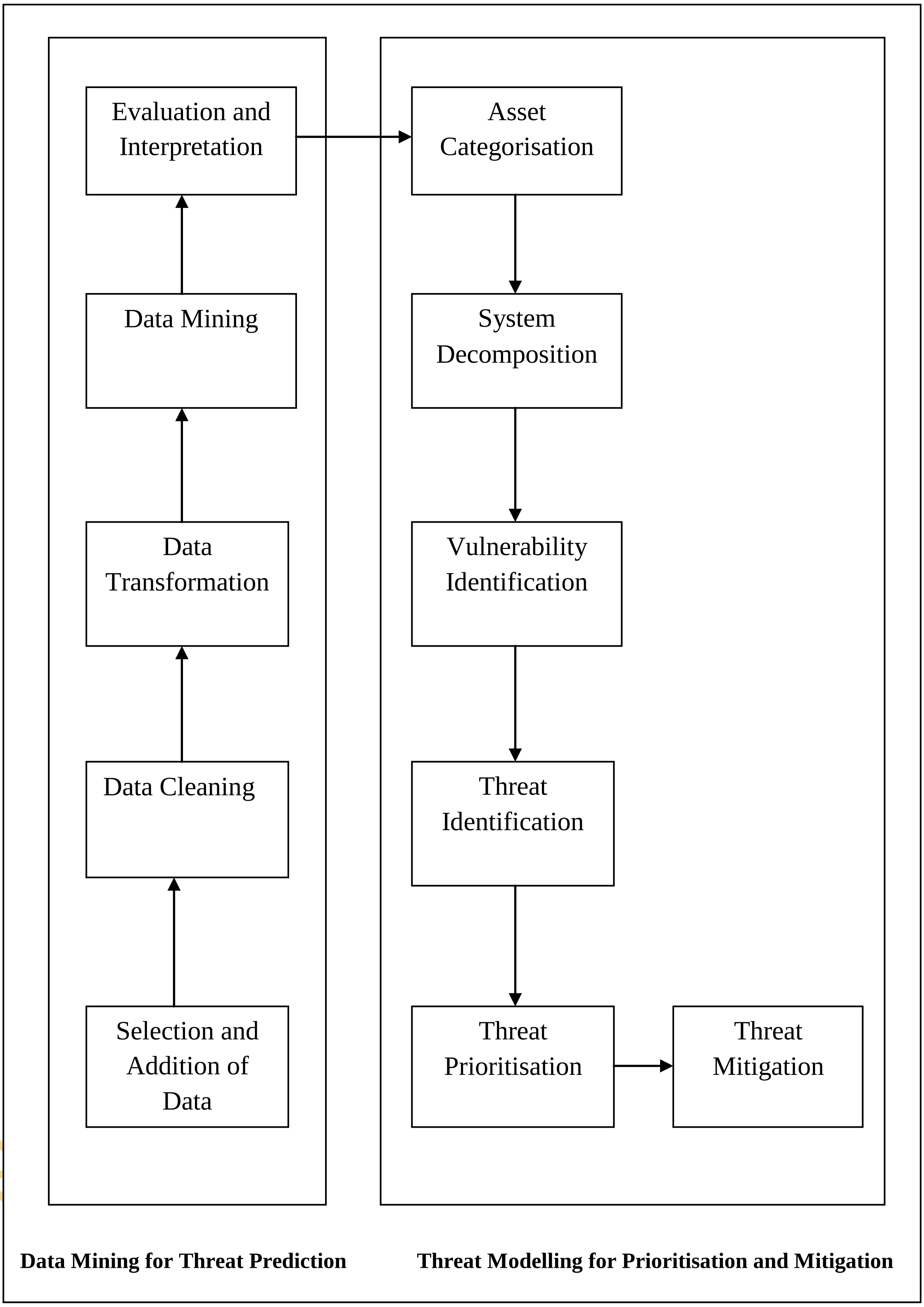

Evaluation and Interpretation
Data Mining
Data Transformation
Data Cleaning
Selection and Addition of Data
Asset Categorisation
System Decomposition
Vulnerability Identification
Threat Identification
Threat Prioritisation
Threat Mitigation
Data Mining for Threat Prediction
Threat Modelling for Prioritisation and Mitigation

Figure 3.6: Framework for the Predictive Hybrid-centric Threat Modelling

a. *Data Pre-processing*

The Central Administrative System scales the alerts. After that operation, it merges all the events from local console and sorts them based on the Time. To perform Association Data Mining, only the actionable features presented in Table 3.2 are used. .

b. *Sequence Association Generation*

The Sequential Association Mining technique is used to generate association sequences. The illustration below describes the method used to generate the sequences. Suppose that x_1, x_2, …, x_n is a stream of events. Using Sliding Window Approach similar to Li *et al.* (2007) and Farhadi *et al.* (2011), once the algorithm is run with a time-based window, the window "slides" Δ alerts in the stream ($1 \leq \Delta \leq L$). That is, if $[\alpha_i, \alpha_{i+1}, \ldots, \alpha_{i+L-1}]$ is a window, the next window will be $[\alpha_{i+\Delta}, \alpha_{i+\Delta+1}, \ldots, \alpha_{i+\Delta+L-1}]$ such that any two adjacent windows share L- Δ alerts. In Figure 3.7, a Typical Window with size W is illustrated.

The following algorithm of Li *et al.* (2007) represented procedurally is used to generate the association sequences:

Step 1*: Set Window size to P, SequenceSize to 1, MaximumSequence Size to L, Sequence to empty*

Step 2*: Sort Incidents based on their timestamps.*

Step 3*: Set the current WindowStep to 1.*

Step 4: *Set Temp to empty.*

Step 5: *Store Incidents according to WindowStep in Temp*

Step 6: *If Sequence Size is L, Continue otherwise Go to Step 9*

Step 7: *Increment WindowStep by 1*

Step 8*: Repeat Step 4 and 6*

Step 9: *Add incident to Temp*

Step 10: *Add Temp to Sequence*

Step 11: Return WindowStep, Sequence

Table 3.2: Sample Event Record for Data Mining

Time	**Source_IP**	**Destination_IP**	**Event_Name**
19:43:45	10:1:0:3	10:1:0:132	ET POLICY SUSPICIOUS INBOUND TO MYSOL PORT 3306
19:43:45	10:1:0:3	10:1:0:133	ET SCAN POTENTIAL SSH SCAN 5900-5920
19:43:45	10:1:0:3	10:1:0:133	ET SCAN POTENTIAL SSH SCAN OUTBOUND
19:43:45	10:1:0:3	10:1:0:133	ET SCAN POTENTIAL SSH SCAN
19:43:45	10:1:0:3	10:1:0:194	ET DOS MICROSOFT REMOTE DESKTOP(RDP) SYN THEN RESET 30 SECOND DOS ATTEMPT
19:43:45	10:1:0:3	10:1:0:228	ET POLICY SUSPICIOUS INBOUND TO ORACLE SQL PORT 1521
19:43:47	10:1:0:3	10:1:0:131	ET SCAN POTENTIAL VNC SCAN 5800-5820
19:43:47	10:1:0:3	10:1:0:131	GPL DNS NAMED VERSION ATTEMPT
19:43:47	10:1:0:3	10:1:0:131	GPL SNMP PUBLIC ACCESS UDP
19:43:47	10:1:0:3	10:1:0:131	GPL RPC PORTMAP LISTING UDP 111
19:43:47	10:1:0:3	10:1:0:131	GPL POLICY PC ANYWHERE SERVER RESPONSE
19:43:47	10:1:0:3	10:1:0:229	ET POLICY SUSPICIOUS INBOUND TO MSSQL PORT 1433

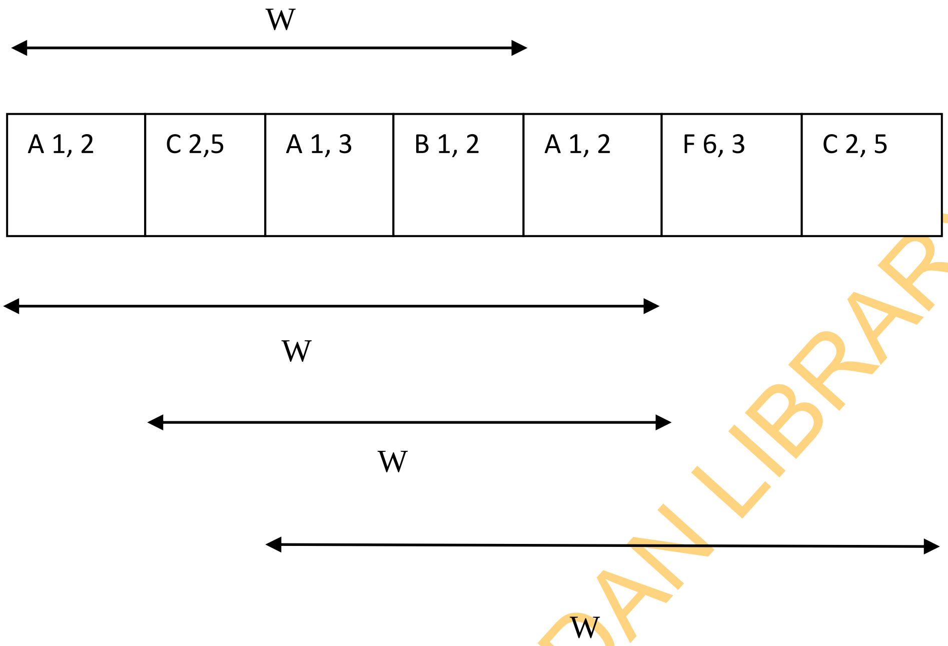


Figure 3.7: Illustration of a Typical Window

c. Rule Interestingness Analysis

The rule Interestingness Analysis is carried out using the support and the confidence of sequence. In this case, a minimum support is set while the confidences of the sequences that meet up with the minimum support produce the interestingness of the sequences.

Given that A → B is an association, A is known as Antecedent and B is known as Consequent. The Support and the Confidence of the Consequent given the Antecedent can be statistically calculated as presented in Equations 3.1 and 3.2.

$$\text{Support (B)} = \frac{n(A \cap B)}{N} \qquad \dots \qquad (3.1)$$

$$\text{Confidence (B)} = \frac{n(A \cap B)}{n(A)} \qquad \dots \qquad (3.2)$$

The Li *et al.* (2007) algorithm is extended this way to generate the association sequence interestingness:

```
Step 1: Assign MinimumSupport to MinSup, WindowStep to Max
Step 2: Set WindowStep to 1
Step 3: Set TempLocation to 0, Temp to empty
Step 4: While WindowStep < Max
Step 5: Increment the WindowStep
Step 6: Add Sequence by WindowStep to Temp
Step 7: If TempLocation != Temp Then Increment the TempLocation
Step 8: Compute the Support of Temp
Step 9: While Support ≥ MinSup, Compute the Confidence
Step 10: Assign Confidence to Interestingness
Step 11: Return WindowStep, Sequence, Interestingness
```

3.3.2 Threat Prioritisation Model

To develop Threat Prioritisation Model, three steps are used.

a. Conceptualisation of Theoretical Framework

The popular McHugh *et al.* (2001) Threat Analysis Theory is adopted for the following reasons:

i. Hybrid-centric Threat Modelling Perspective

ii. Focus of the Threat Modelling on Attacker and Victim, which are associated with Asset, Attack and Defence-centric Threat Modelling Perspectives.

iii. Explicitness and Practicality of the Threat Model

The following presents the theory and its adaptation:

McHugh *et al.* (2001) Attacker Perspective of Intrusion

- *What is my objective?*
- *What vulnerabilities exist in the target system?*
- *What damage or other consequences are likely?*
- *What exploit scripts or other attack tools are available?*
- *What is my risk of exposure?*

McHugh *et al.(2001)* Victim Perspective of Intrusion

- *What happened?*
- *Who is affected and what were the consequences?*
- *Who is the intruder?*
- *Where and when did the intrusion originate?*
- *How and why did the intrusion happen?*

Some of the answers to the victim perspective of intrusion are hidden- they require analysis for their disclosure. Therefore, this stage adopts only the direct perspective that can easily be obtained by the security manager which is:

What happened?

The perspective "what happened?" generates sub-questions such as:

What event is reported, at a particular instance by sensor, S?

How many of such events were reported by sensor, S?

How many alarms escape detection by sensor, S?

What is the severity of the event as reported by sensor, S?

Which host was the target as reported by sensor, S?

Table 3.3 and Table 3.4 present the perspectives and criteria for measurement. The perspectives are drawn directly from the perspectives discussed above while the criteria are drawn from existing Risk Analysis works. The attacker-centric sub-criteria conform to Bhattacharya *et al.* (2008), CVE-MITRE, Bugtraq and OSVDB (Porras *et al.*, 2002) while the victim-centric sub-criteria conform to Haslum *et al.* (2007) and Killourhy *et al.* *(*2004) works.

In Table 3.3 and Table 3.4, there is no score 'unknown' because the data fusion algorithm tolerates all the possible Perception Level which represents the criteria score.

b. Threat Rating

Because multiple security administrative domains collaborate in managing the threat, a policy web of trust is developed to overcome distrust which may result in inaccurate Threat Measurement. Three actors associated with each administrative domain are identified as determinants in this respect: administrator, communication channel and data source. The following factors determine the trust of each actor:

i. *Administrator:* Integrity, Ability, Benevolence and Trust Propensity

ii. *Contact Medium:* Integrity, Confidentiality and Availability

iii. *Sensor/ Data Sources:* Comprehension, Integrity and Reliability

The sum of the scores of all the variables is the mass value of trust for such perception. The value is less than or equal to 1. This scoring of these variables is presented in Table 3.5.

Table 3.3: Description of Attacker-centric Perspective, Criteria and Measurement

Perspective	**Criterion**	**Perception (Level 1)**	**Perception (Level 2)**	**Perception (Level 3)**
Exploitability	Exploit Availability	Unavailable	Scarce	Readily
	Ease of Exploitation	Expert	Trained	Novice
Risk of Exposure	Discoverability	Year	Month	Day
	Remediation	Adequate	Inadequate	Unavailable
Damage	Confidentiality Impact	None	Partial	Fully
	Integrity Impact	None	Partial	Fully
	Availability Impact	None	Partial	Fully

Table 3.4: Description of Victim-centric Perspectives, Criteria and Measurement

Perspective	**Criterion**	**Perception (Level 1)**	**Perception (Level 2)**	**Perception (Level 3)**
Frequency	Sensor Type 1	Less or equal to A	greater than A and less than B	greater or equal to B
	Sensor Type 2	Less or equal to A	greater than A and less than B	greater or equal to B
	. . .	…	…	…
	Sensor Type n	Less or equal to A	greater than A and less than B	greater or equal to B
Resistance (Inverse of Sensitivity)	Sensor Type 1	Less or equal to R	greater than R and less than S	greater or equal to S
	Sensor Type 2	Less or equal to R	greater than R and less than S	greater or equal to S
	. . .	…	…	…
	Sensor Type n	Less or equal to R	greater than R and less than S	greater or equal to S
Severity	Sensor Type 1	Less or equal to X	greater than X and less than Y	greater or equal to Y
	Sensor Type 2	Less or equal to X	greater than X and less than Y	greater or equal to Y
	. . .	…	…	…
	Sensor Type n	Less or equal to X	greater than X and less than Y	greater or equal to Y

Table 3.5: Table showing the Trust Model Actors, Variables and Descriptions

Actor	**Variable**	**Description**
Administrator	Integrity	It is defined as the extent to which a trustee is believed to adhere to ethical principles. It is assigned value between 0.0 and 0.1.
	Ability	It captures the "can-do" component of trustworthiness by describing whether the trustee has the skills needed to act in an appropriate fashion. It is assigned value between 0.0 and 0.1.
	Benevolence	It is the extent to which a trustee is believed to want to do good for the trustor. It is assigned value between 0.0- 0.1
	Trust Propensity	It is the dispositional trust that is associated to what the actor 'will do' instead of 'can do'. It is assigned between 0.0 and 0.1
Contact Medium	Confidentiality	It measure the state of contact medium in ensuring that only those with sufficient privileges and demonstrated need access certain information. It is assigned value between 0.0 and 0.1
	Integrity	It is the state of wholeness of contact medium. It is assigned values between 0.0 and 0.1
	Availability	It measures the state of contact medium in ensuring uninterrupted user access0. It is assigned value between 0.0 and 0.1
Sensor/Data Source	Integrity	This is the belief in the condition of sensor or data source to produce the right output. 0.0- 0.1
	Comprehension	This is the belief in the condition of sensor to produce understandable outputs. It is assigned value between 0.0- 0.1.

Reliability	This is the belief in the condition of sensor/data sources to always produce the right output. It is assigned value between 0.0 and 0.

In order to tackle the uncertainty problems associated with Threat assessment, Dempster-Shafer Belief Theory is used. The Dempster-Shafer Theory is used to reconcile the scores from different sources. In order to fuse data from same source with different significance, we apply the belief as the weight and find the weighted average of the score. This is similar to Expectation Theory which could predict the expected criteria score of threat.

In order to rate threats, both Dempster-Shafer Theory and Expectation Theory are combined. The Dempster-Shafer Theory is applied to eliminate uncertainty and probability while the Expectation Theory is applied to reduce Multidimensionality.

Dempster-Shafer Method is used to obtain degrees of belief of one evidence from subjective probabilities for a data source. The Dempster-Shafer theory of Belief Function according Shafer (I976) is a generalization of the Bayesian theory of subjective probability. The advantage over Bayesian Theory is Bayesian Theory requires probabilities for each question of interest which are not actualisable in network security field, but with Dempster-Shafer belief functions, degrees of belief for one question can be based on the probabilities for a related question. The Dempster-Shafer Theory consists of hypotheses, pieces of evidence

and data sources. The hypotheses represent all the possible states (evidence assignments). It is required that all hypotheses are elements (singletons) of the frame of discernment, which is given by the finite universal set Ω. The set of all subsets of Ω is its power set 2^{Ω}. The pieces of evidence are the qualitative scores or observations may occur within a system. Data sources are InfoSec devices logs, vulnerability databases or any other information sources that provide information.

The expected value (or expectation) refers, intuitively, to the value of a random variable one would "expect" to find if one could repeat the random variable process an infinite number of times and take the average of the values obtained. More formally, the expected value is a weighted average of all possible values (Ross, 2007). In other words, each possible value that the random variable can assume is multiplied by its assigned weight, and the resulting products are then added together to find the expected value. The weights used in computing this average are the probabilities in the case of a discrete random variable (that is, a random variable that can only take on a finite number of values, such as a roll of a pair of dice).

The following steps are taken to rate Minor Threats:

i. *Computation of Belief Value using Dempster-Shafer Function of Rule of Combination* (Shafer, 1976) *expressed as:*

$$M(Z) = \frac{\sum_{A \cap B = Z \neq \emptyset} m(A).m(B)}{\sum_{A \cap B \neq \emptyset} m(A).m(B)} \quad \ldots \quad (3.\ 3)$$

where *A*, *B*, *Z* $\underline{C}$ Z. M are the mass function. In definite term, the numerator represents the accumulated evidence for the sets *A* and *B*, which supports the hypothesis *Z*, and the denominator is the sum of the amount of conflict between the two sets.

ii. Normalization of the Belief Value

The maximum belief values for the criteria are normalized so that the sum is equal to 1.

$$\text{Normalized } (P_i) = p_i \ / \ \Sigma^{n}_{i=1} P_i \quad \ldots \quad (3.4)$$

iii. Calculation of the Expected Value for Risk-determination factors' Fusion using Expectation Theory (Ross, 2007).

The expected value of objective *X* is defined as

$$E(X) = P_1X_1 + P_2X_2 + \ldots + P_kX_k \quad \ldots \quad (3.5)$$

Since all probabilities p_i add up to one ($p_1 + p_2 + \ldots + p_k = 1$), the expected value can be viewed as the weighted average, with p_i's being the weights.

$$E(X) = \frac{P_1X_1 + P_2X_2 + \ldots + P_kX_k}{P_1 + P_2 + \ldots + P_k} \quad \ldots \quad (3.\ 6)$$

iv. *Estimation of Attack and Victim-based Threat Rating*

In order to rate the threats, the rate of sum of the objective scores with asset criticality rank are estimated.

Attacker-based Threat Rating

Rate (R_A) =

$$\frac{\text{Objective Exploitability + Objective Damage + Objective Risk of Exposure}}{\text{Asset Criticality Rank}} \quad \ldots \quad (3.7)$$

Victim-based Threat Rating

Rate (R_V) =

$$\frac{\text{Objective Frequency + Objective Severity + Objective Resistance}}{\text{Asset Category Rank}} \quad \ldots \quad (3.8)$$

v. *Threat Rating:*

The sum of both Attacker-based Threat Rating and Victim-based Threat Rating is the Threat Score.

Threat Rating Score (R_T) =

Attacker-based Threat Rating ($R_{A)}$ + Victim-based Threat Rating (R_V) … (3.9)

c. *Threat Ranking*

The threats are ranked by grading the threat ratings based on the security policy. For instance, the Major Threats can be ranked from 1 to 2 while the Minor Threats can be ranked from 3 to 4. The illustration of the relation between Rating and Ranking is presented in Figure 3.8.

3.3.3 Threat Mitigation Model

The last step of the Minor Threat Modelling is Threat Mitigation. The Threat Mitigation focuses on Collaborative Network Intrusion Detection and Prevention Configuration Management. Hillson (1999) Risk Mitigation Model is adapted to mitigate Minor Threats using distributed and multiple InfoSec sensors. In the next subsection, the Threat Mitigation Model is presented.

3.4.3.1 Formalisation of Network Threat Management

Collaborative Network Intrusion Detection and Prevention Configuration Management is a Six tuple (T, A, B, D, F, S). T is the Threat that will lead to critical damage in A. A is the Network Security Management domain that is been secured while B consists of the partnering Network Security Management domains. The appropriate domain is chosen among the partners during transfer operation. D is the Intrusion Detection Configuration action at a particular instance while F is the Firewall Configuration action at a particular instance. S is the scope of the configuration project, which includes Time to configure the InfoSec, Cost to configure the InfoSec and Quality or Effectiveness of Configuration action at a particular instance. Figure 3.9 presents the Quadrant for the Network Intrusion Detection and Prevention Configuration Management.

The configuration management actions are categorized into the four mitigation segments of Hillson (1999):

a. Avoid: The action taken is: *“enable the signature rules for Network Intrusion Detection System and the firewall filtering filter in Network Security Management domain A.”*

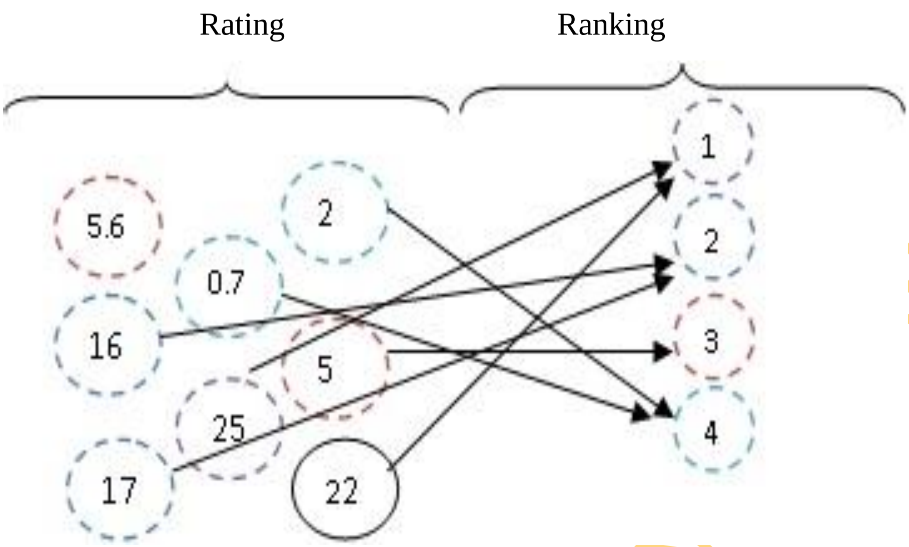


Figure 3.8: Relation between Rating and Ranking

Avoid **High Priority** **1**	Transfer **Moderate Priority** **2**
Mitigate **Low Priority** **3**	Accept **Very Low Priority** **4**

Figure 3.9: Quadrant for Network Intrusion Detection and Prevention Configuration Management

b. Transfer: The action taken is: *"enable the signature rules for Network Intrusion Detection System and the firewall filter in Network Security Management domain B."*

c. Mitigate: The action taken is: *"enable the signature rules for Network Intrusion Detection System and disable the firewall filter in Network Security Management domain A."*

d. Accept: The action taken is: *"disable both the signature rules for Network Intrusion Detection and the firewall filter in Network Security Management domain A."*

The low priority threats among the Minor Threats are mitigated as described in the 3rd Quadrant while the very low Minor Threats are accepted as described in the 4th Quadrant.

3.4 Design and Implementation of Tools for Modelling and Mitigation Minor Threats

Threat Prediction and Threat Prioritisation tools have been fully implemented while external system has been adopted for the Threat Mitigation. The rationales behind adopting existing Threat Mitigation tool, rather than implementing it from scratch are twofold; firstly implementing this would have been out of the scope of the proposed research, and secondly supporting input from existing Threat Mitigation framework serves to provide a more realistic environment and strengthen compatibility with existing solutions. Hence, Security Onion, a popular Unified Threat Management tool with ability to mitigate threats using multiple sensors is used (Burk, 2007). The design and implementation are described under four headings: Unified Modelling Language Designs, Database Models, Program Implementation Procedures and System Implementation Procedures. The Security Onion Architecture and Implementation are also discussed.

3.4.1 Unified Modelling Language Designs

Use case modelling has been widely utilised to graphically portray a functional description of interaction between external entities and systems, as well as their collaborations. They are applied to capture the behaviours of the developed systems, without having to specify how those behaviours are implemented (Booch *et al.*, 2005).

A state diagram describes all the possible states of an object as events occur, and is used to demonstrate the behaviour of an object through many use cases of a system, as well as to emphasise the flow of control from one state to another (Booch *et al.*, 2005). It is also called Activity Diagram.

3.4.1.1 UML for Threat Prediction Tool

Figure 3.10 presents the Use Case model of the roles of the Central Administrative System and the Network Security Managers. Figure 3.11 present the activity of the Central Administrative System and the Network Security Managers.

a. *Use Case Model for Threat Prediction Tool*

The roles of the Central Administrative System in Threat Prediction include:

i. Sign up for Mail Service: The Central Administrator registers and gets an e-mail account of his choice.

ii. Checking, Receipt and Sending of Mails about Events and Incidents: He check his e-mail for receipt of events from different network security managers and send information about the events to the network security manager on regular basis.

iii. Assignment of window size: The administrator sets window sizes in minutes depending on the average period of detection of replayed threats

iv. Generate of Candidate Attack Sequence: He clicks generate sequence button to generate the candidate sequences.

v. Generation of Sequences and Interestingness: He clicks generate support and confidence button to obtain the sequences in steps and their corresponding support and confidence.

vi. Write Result to External Disc: He writes the result to personalised external disc for security purpose.

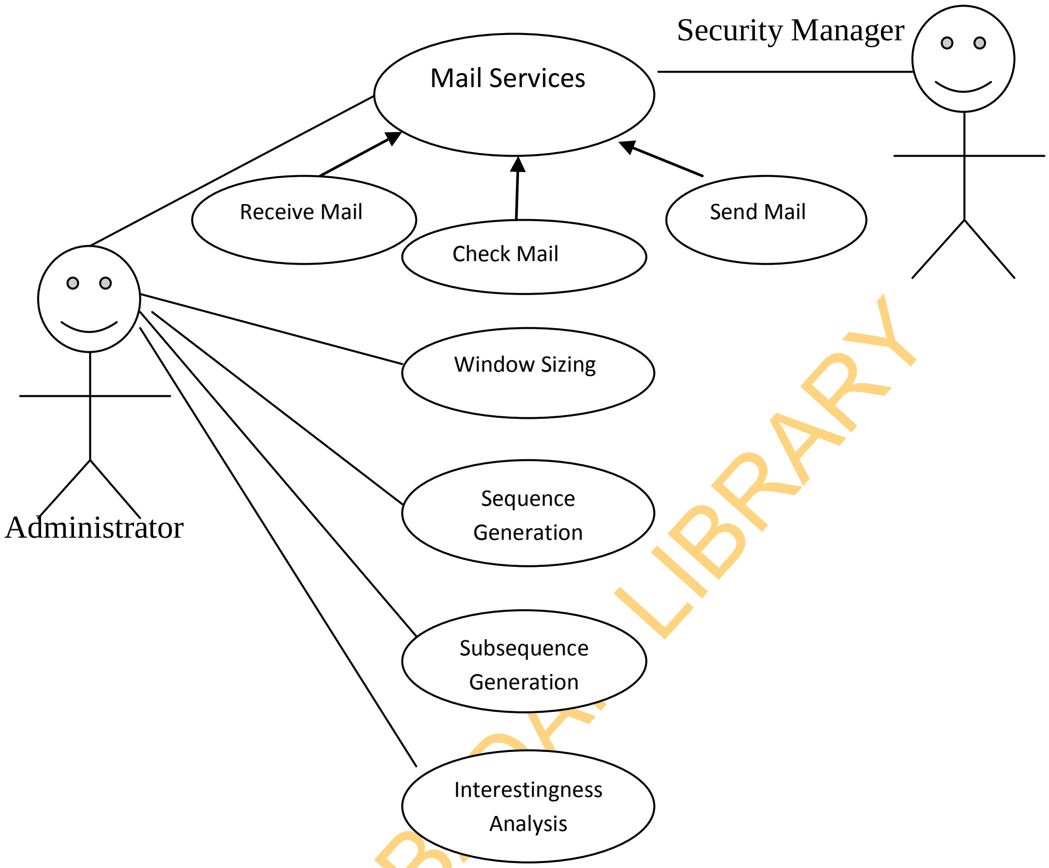


Figure 3.10: Use Case Diagram of Threat Prediction Tool

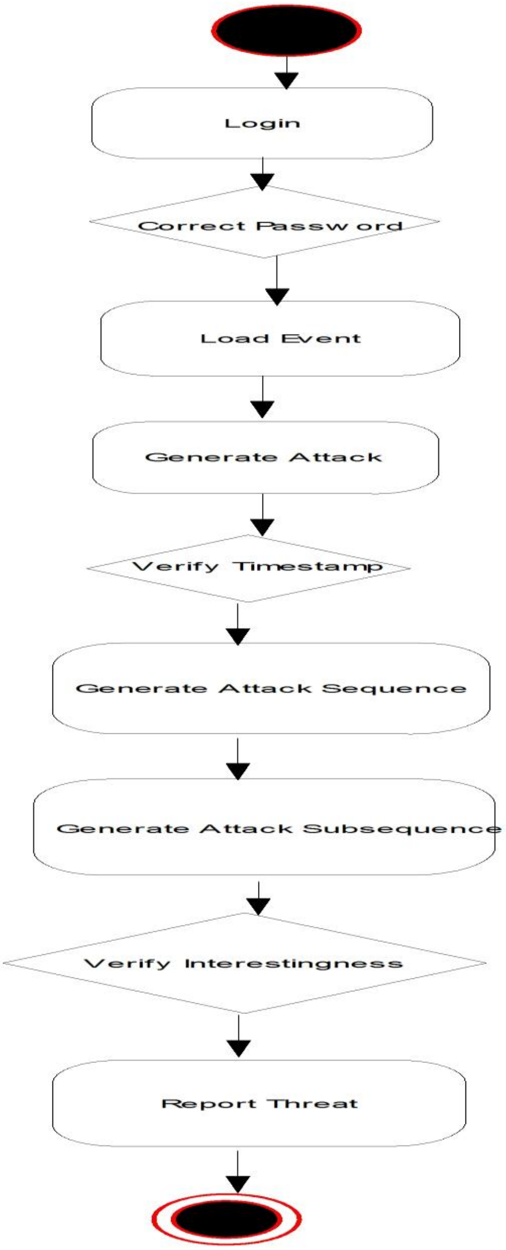


Figure 3.11: Activity Diagram of the Functionalities of Threat Prediction Tool

vii. Analysis of Interestingness: He finally perform interestingness analysis based on highest possible minimum support

The roles of the Network Security Manager in Threat Prioritisation include:

i. Sign up for Mail Service: The Network Security Managers register and get an e-mail account of his choice.

ii. Checking, Receipt and Sending of Mails about Events and Incidents. They check their e-mail for information received from central administrator and send information about the events when necessary.

b. Activity Model for Threat Prediction Tool

Figure 3.11 presents the Activity Diagram of the functionalities that Central Administrative System and Network Security Manager perform. These activities are described below:

i. Login: The administrator login using a password. If the password is correct, he can proceed to generate attack. If the password is wrong, the system remains in the same state.

ii. Generate Attack: Once the administrator login successfully, the event, source address and destination address are used to generate the attack.

iii. Verify Window Size: At the Window Sizing state, the timestamp that fall within the window size is chosen. If the timestamp is greater window size, the window is terminated in the preceding timestamp.

iv. Generate Attack Sequence: The attacks that correspond to the selected timestamp are chosen and Organised in sequence.

v. Generate Attack Subsequence: The attack sequence in each steps are reported with their corresponding support and confidence values.

vi. Verify Subsequence Interestingness: The Administrator the interestingness based on a minimum support value.

3.4.1.2 UML for Threat Prioritisation Tool

Figure 3.12 presents the roles of the Central Administrative System and the Network Security Managers. Figure 3.13 and Figure 3.14 present the activity of the Central Administrative System and the Network Security Managers.

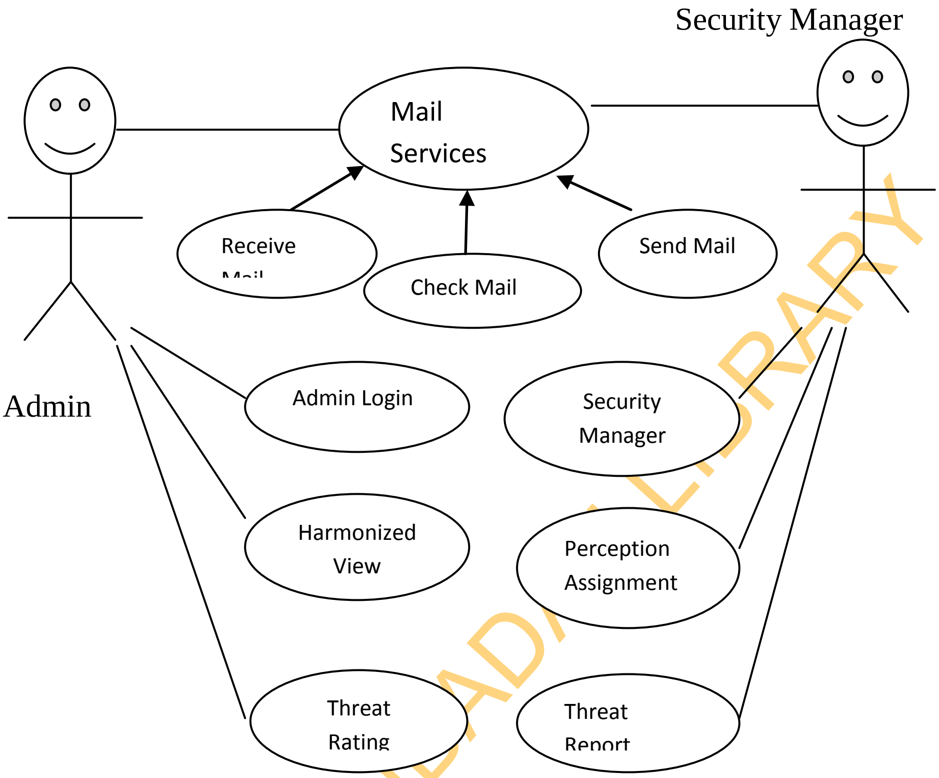


Figure 3.12: Use Case Diagram for Threat Prioritisation Tool

a. Use Case Model for Threat Prioritisation Tool

The roles of the Central Administrative System in Threat Prioritisation include:

i. Sign up for Mail Service: The Central Administrator registers and gets an e-mail account of his choice.

ii. Checking, Receipt and Sending of Mails about Events and Incidents: He check his e-mail for receipt of events from different network security managers and send information about the events to the network security manager on regular basis.

iii. Formatting of Event and Incident, Configuration of Assets, and Scaling InfoSec Configuration.

iv. Harmonization and Fusion of the Security Managers' Perception by assigning trust values to perceptions and clicking harmonise trust.

v. Rating of Threat by clicking Report Icon.

The roles of the Network Security Manager in Threat Prioritisation include:

i. Sign up for Mail Service: The Network Security Managers register and get an e-mail account of his choice.

ii. Checking, Receipt and Sending of Mails about Events and Incidents. They check their e-mail for information received from central administrator and send information about the events when necessary.

iii. Keying of Attacker and Victim-based Perceptions.

iv. Generation of Threat Report by clicking Report Icon.

b. Activity Model for Threat Prioritisation Tool

Although the use case diagram has provided a brief overview of the modules' functionality, it does not clarify how those modules are performed. Hence, this section presents the Activity diagrams for the Threat Prioritisation Model.

Figure 3.13 presents the Activity Diagram of the functionalities that Central Administrative System performs. These activities are described below:

i. Login: The administrator login using a password. If the password is correct, he can proceed to configure the asset, threat and InfoSec. If the password is wrong, the system remains in the same state.

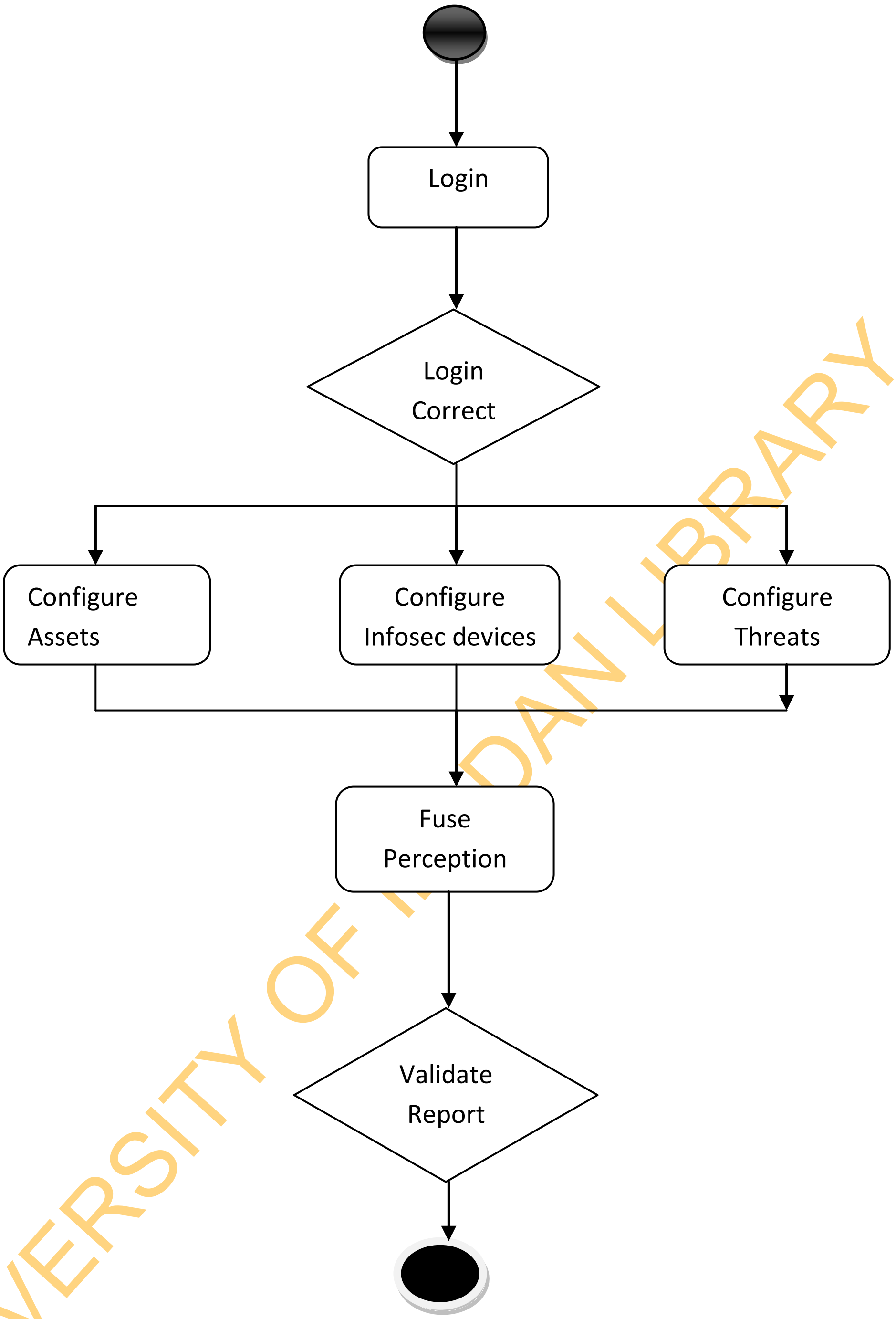


Figure 3.13 Activity Diagram of the Functionalities performed by Central Administrative System for Threat Prioritisation Tool

i. Configure: At the Configure state, the administrator configure the measurement template for the security manager use, which after keying perceptions moves to Harmonize View. Lack of the action means no progress in Threat Prioritisation.

ii. Fuse Perception: At the Harmonize View, the perception receives trust values, which are equated to 1. The administrator finally validates the threat report before it is sent through email to Security Manager.

Figure 3.14 presents the Activity Diagram of the functionalities that Network Security Manager performs. These activities are described below.

i. Login: The security manager registered and login with his password. If the password is correct, he can proceed to configure the asset, threat and InfoSec. If the password is wrong, the system remains in the same state.

ii. Perception Assignment: At this state, the security manager key in the perception into the measurement template, which acted upon by the Administrative System. Lack of the action by the administrator means no progress in Threat Prioritisation.

iii. Verify Fusion: The administrator verifies the fusion before it requests the administrator to generate the threats, which are ranked the security manager.

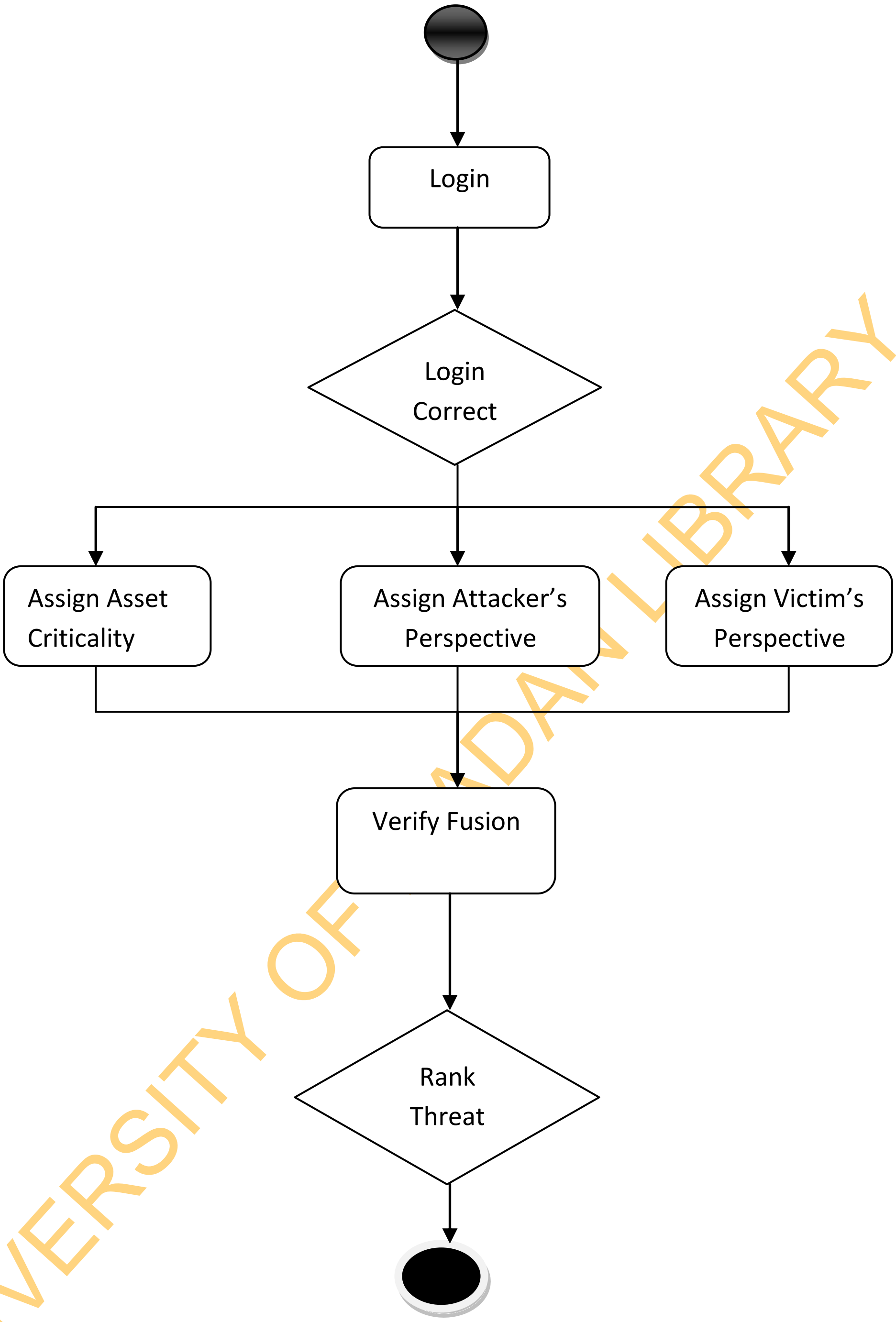


Figure 3.14: Activity Diagram of the Functionalities performed by Network Security Manager for Threat Prioritisation Tool

3.4.2 Relational Database Model

A database is a structured collection of data. It may be anything from a simple shopping list to a picture gallery or the vast amounts of information in a corporate network. To add, access, and process data stored in a computer database, a database management system such as MySQL Server is needed. MySQL is a relational database management system. A relational database stores data in separate tables rather than putting all the data in one big storeroom. This adds speed and flexibility. SQL is the most common standardized language used to access databases.

In Figure 3.15, the relational model for the design of prototype of Threat Prediction Module is presented. The model contains five related tables: Event, Sequence, Ant_Cons, Var_Sequence1, and Var_Sequence2 Tables. In Figure 3.16, the relational model for the design of prototype of Threat Prioritisation Module is presented. The model contains twenty two (22) tables. These include Host, Assets, Clients, Asset_Admin, Asset_Threat_Perception_Certainty, Master, SMP_Perception, SMP_Asset_Threat, Slave, Response, Response_Subcategory, Perspective, Asset_Threat, AssetThreat_Certainty, AssetThreat_Client, Admin_Perspective, Client_Threats, Threats, AssetThreat_Perception, Admin, Threat_Perception, AssetThreat_Objective and ThreatRating

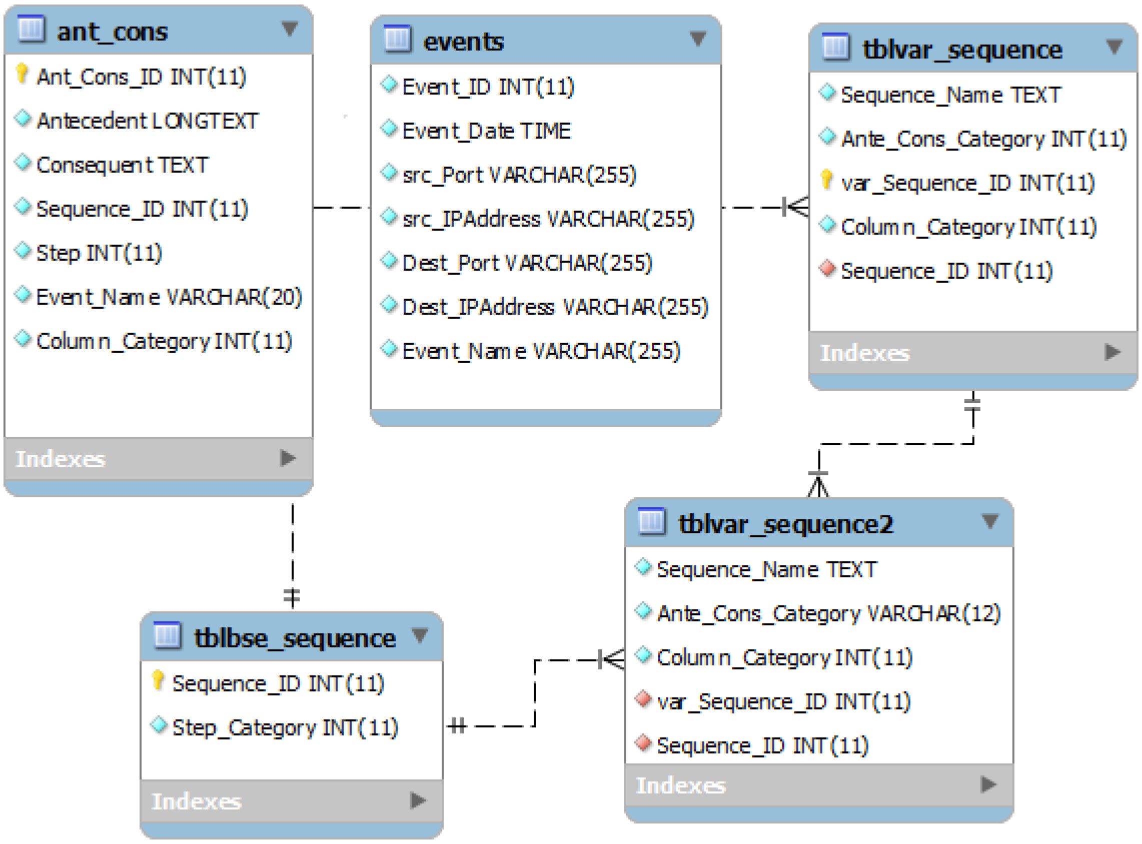


Figure 3.15: Relational Database Model for the design of prototype of Threat Prediction Tool (Threat-Predict)

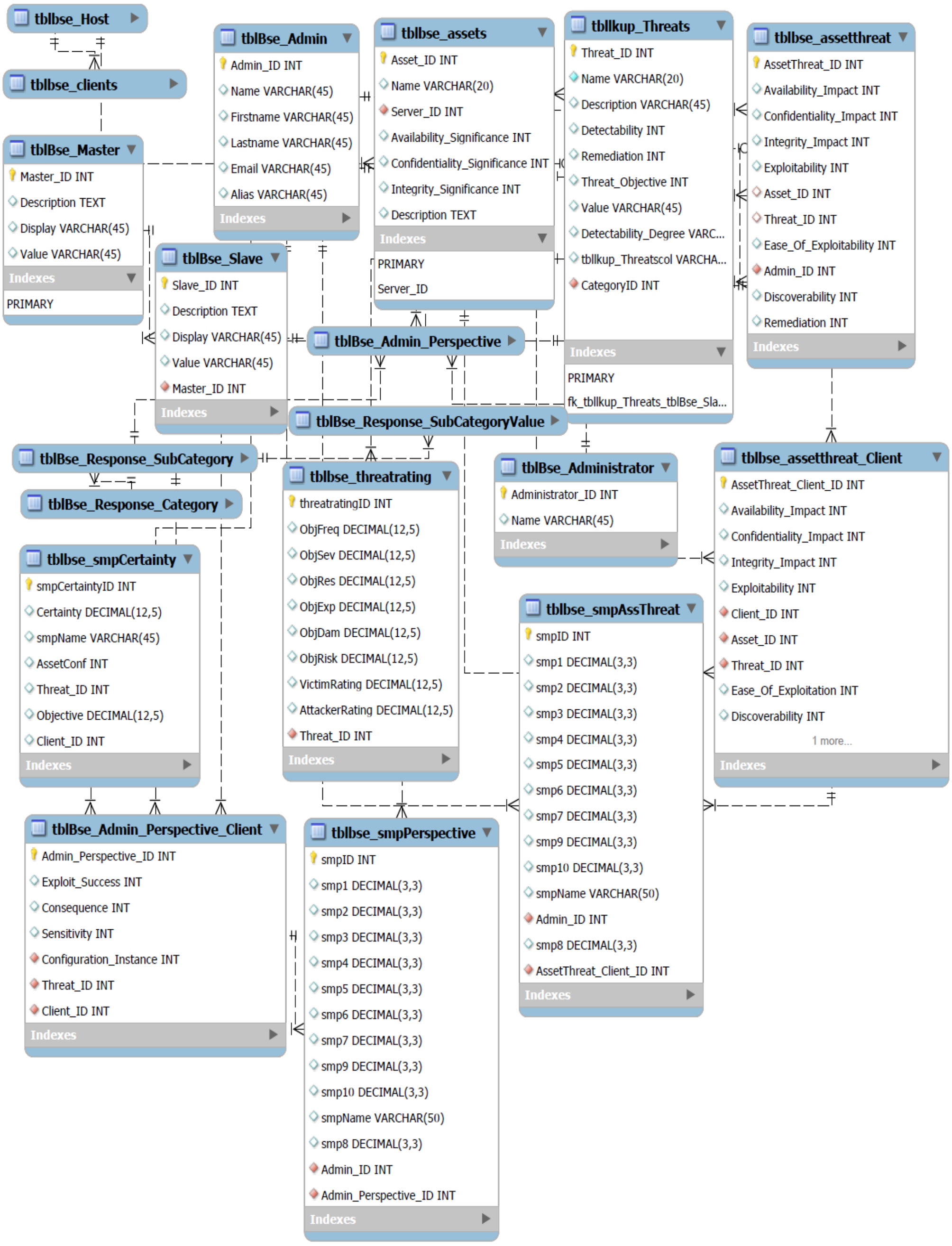


Figure 3.16: Relational Model for the design of prototype of Threat Prioritisation Tool (COSEM-TR)

3.4.3 Implementation

Generally, Java Programming Languages are used to implement the prototypes. The choice of Java is as a result of its general public licence, easy integration with other programming languages, cross platform (platform independence), mobile-enabled characteristic, ability to handle complex task, multithreading and easy extensibility.

3.4.3.1 Implementation of Threat Prediction Tool

a. Program Implementation of Threat Prediction Tool

The Threat Prediction modules are programmed using Java Standard Edition in Netbeans IDE 6.9.1. The database is used to store the inputs and the outputs and it is implemented using a Structured Query Language (SQL)-supported platform known as WAMP v 2.1d. In order to run the application, JDK library must be installed. Appendix 1 presents the codes used for the implementation of the Threat Prediction tool. The system requirements used to run the application were:

i. Intel Processor

ii. 4 GB RAM

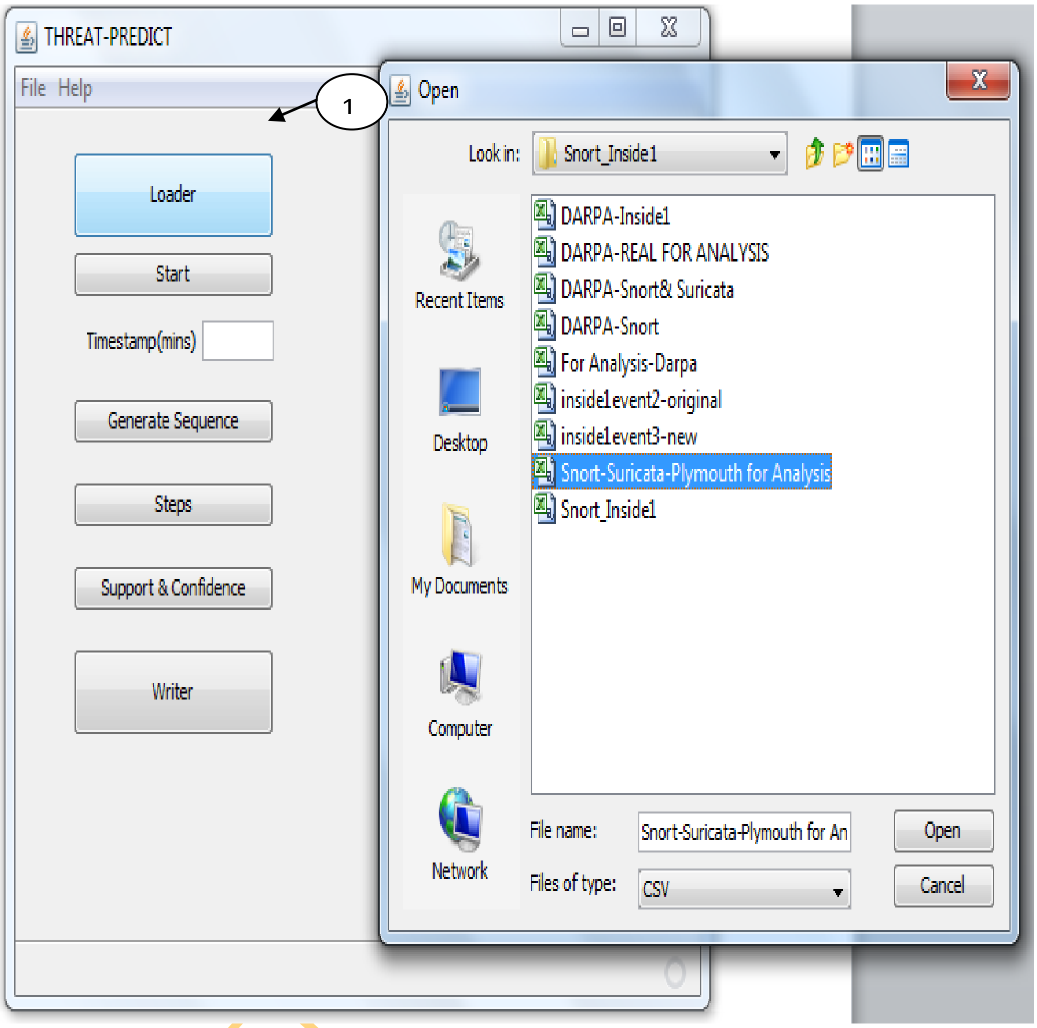
iii. 2 Partitioned Hard disk of 200GB each.

iv. 1.5 GHz CPU

v. Window XP

b. Demonstration of the Prototype of Threat Prediction Tool

The following screenshots demonstrate the usage of the Prototype of Threat Prediction Tool (THREAT-PREDICT). In Figure 3.17, index 1 points to the menu for loading event and incident data into the THREAT-PREDICT. In Figure 3.18, Index 2 presents the screen shots for starting the running of the application while Index 3 points to the textbox used for inputting timestamp for window sizing. The timestamp is in minutes. Index 4, Index 5 and Index 6 are used to perform sequence generation, sequence sorting into steps (subsequence generation) and support and confidence estimation respectively. The Index 7 is used to output the results. Because the size of data may be huge, a separate dedicated storage is designed for the storage of the result, which can be viewed in any word processing tool. Figure 3.19 presents the screenshot for results page written to writex in dedicated storage disk.



Look in: Snort_Inside1

DARPA-Inside1
DARPA-REAL FOR ANALYSIS
DARPA-Snort& Suricata
DARPA-Snort
For Analysis-Darpa
inside1event2-original
inside1event3-new
Snort-Suricata-Plymouth for Analysis
Snort_Inside1

File name: Snort-Suricata-Plymouth for An

Files of type: CSV

Figure 3.17: Screenshot for Event Loader

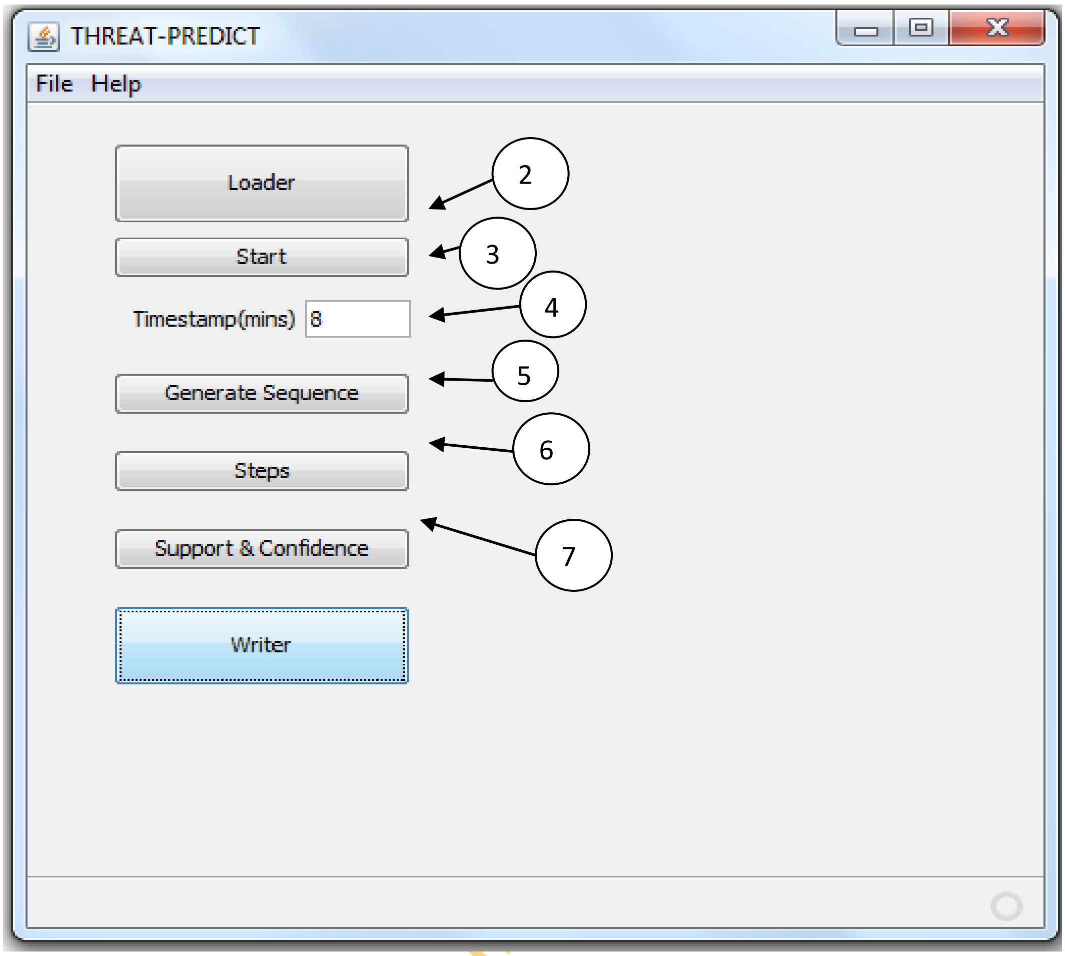
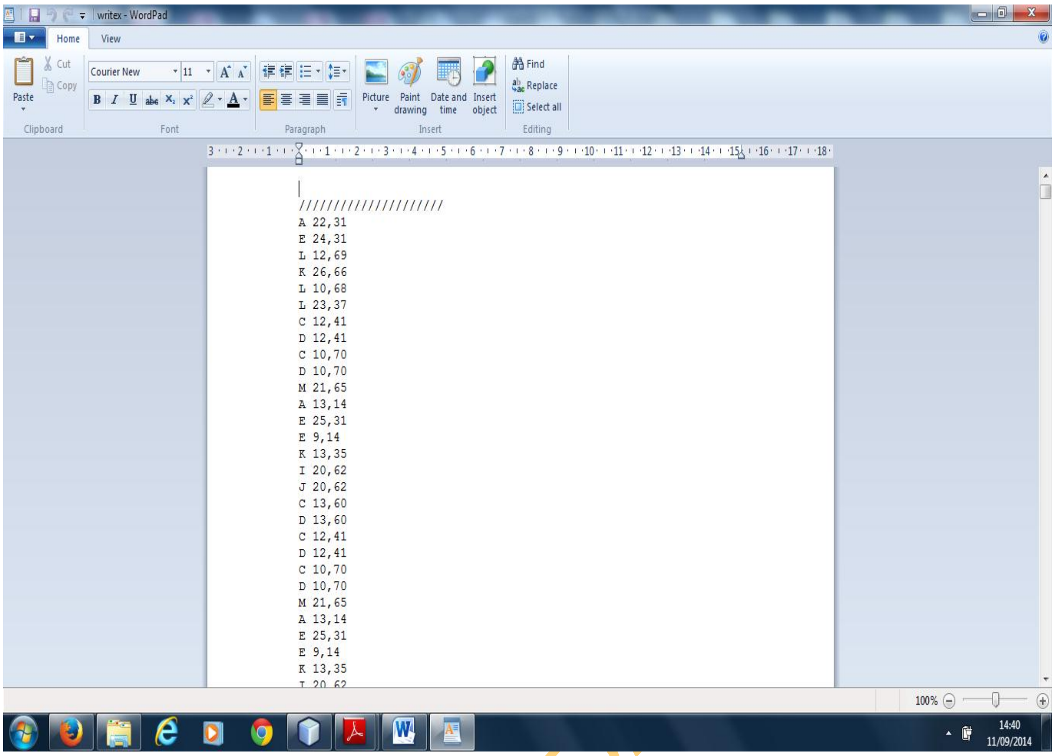


Figure 3.18:Screenshot for executing the program



```
////////////////////
A 22,31
E 24,31
L 12,69
K 26,66
L 10,68
L 23,37
C 12,41
D 12,41
C 10,70
D 10,70
M 21,65
A 13,14
E 25,31
E 9,14
K 13,35
I 20,62
J 20,62
C 13,60
D 13,60
C 12,41
D 12,41
C 10,70
D 10,70
M 21,65
A 13,14
E 25,31
E 9,14
K 13,35
I 20,62
```

Figure 3.19: Screenshot for Results Page written to WriteX in Dedicated Storage Disk

3.4.3.2 Implementation of Threat Prioritisation Tool

a. Program Implementation of Threat Prioritisation Tool

The Threat Prioritisation modules are programmed using Java Server Page (JSP V 1.9) and Java Script in Netbeans IDE 6.9.1. The database is used to store the inputs and the outputs and it is implemented using a Structured Query Language (SQL)-supported platform known as WAMP v 2.1d. Since, the application is a web-based tool, we recommend network connectivity for full operation. Appendix 2 presents the codes for the implementation of the Threat Prioritisation module.

The system requirements used for running the application were:

i. Intel Processor
ii. 4 GB RAM
iii. 200GB Hard disk.
iv. 1.5 GHz CPU
v. Window 7
vi. Ethernet Card
vii. Wired LAN or WLAN
viii. Web Browser
ix. Four Network Security Managers and 1 Central Administrator

b. Demonstration of Prototype of Threat Prioritisation Tool (COSEM-TR)

The web modules provide a graphical user-friendly interface that allows users to view and configure the modules. The web modules provide a web analytics solutions that give rich insights into the Threat Prioritisation process. The simplicity, easy-to-use, customisable and privacy of use allows the central administrator and the local security manager to perform their functions in objective manner. The web modules provide common results related to events. To log into the web modules, the central administrator needs to use a legitimate password; otherwise the web modules will not allow them to visualise the other analytics pages. Label 1 in Figure 3.20 illustrates the login form that needs to be filled by administrators before he can browse the web modules. Once the login process is successful, administrators are redirected to the configuration page of the web modules. Figure 3.21 illustrates the threat configuration board. Figure 3.22 and Figure 3.23 are security managers' registration and account page.

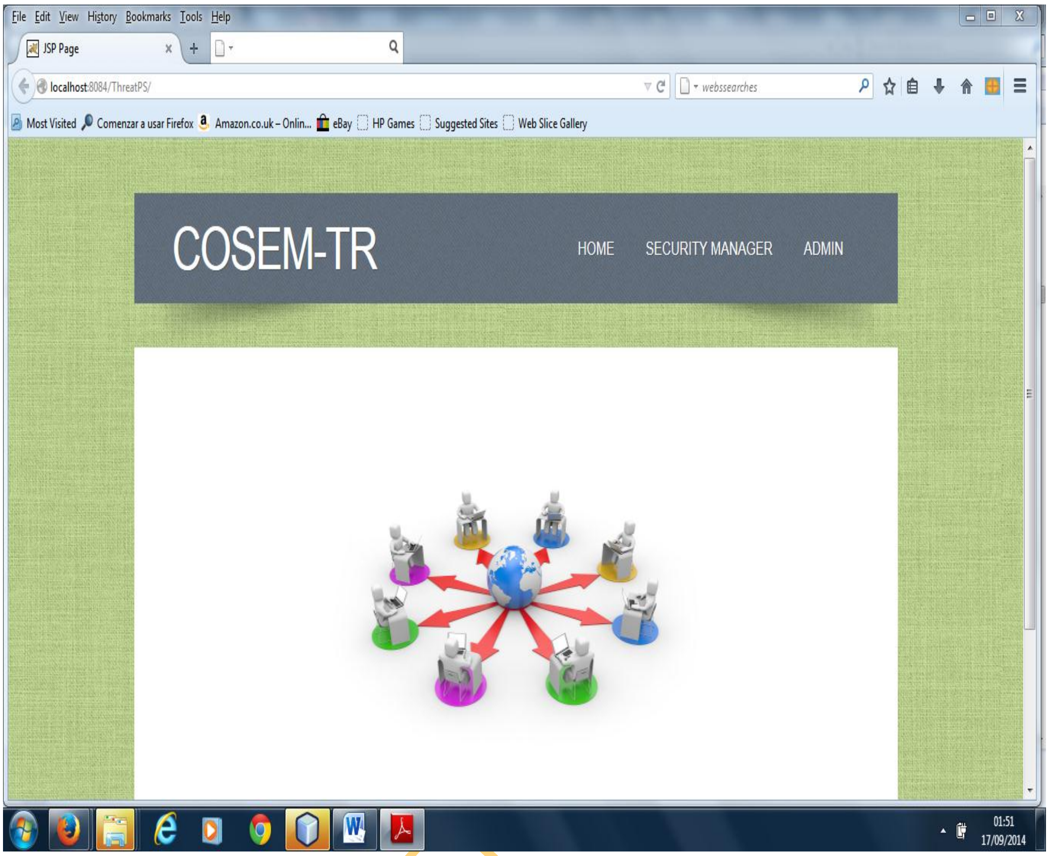


Figure 3.20: Home Screen

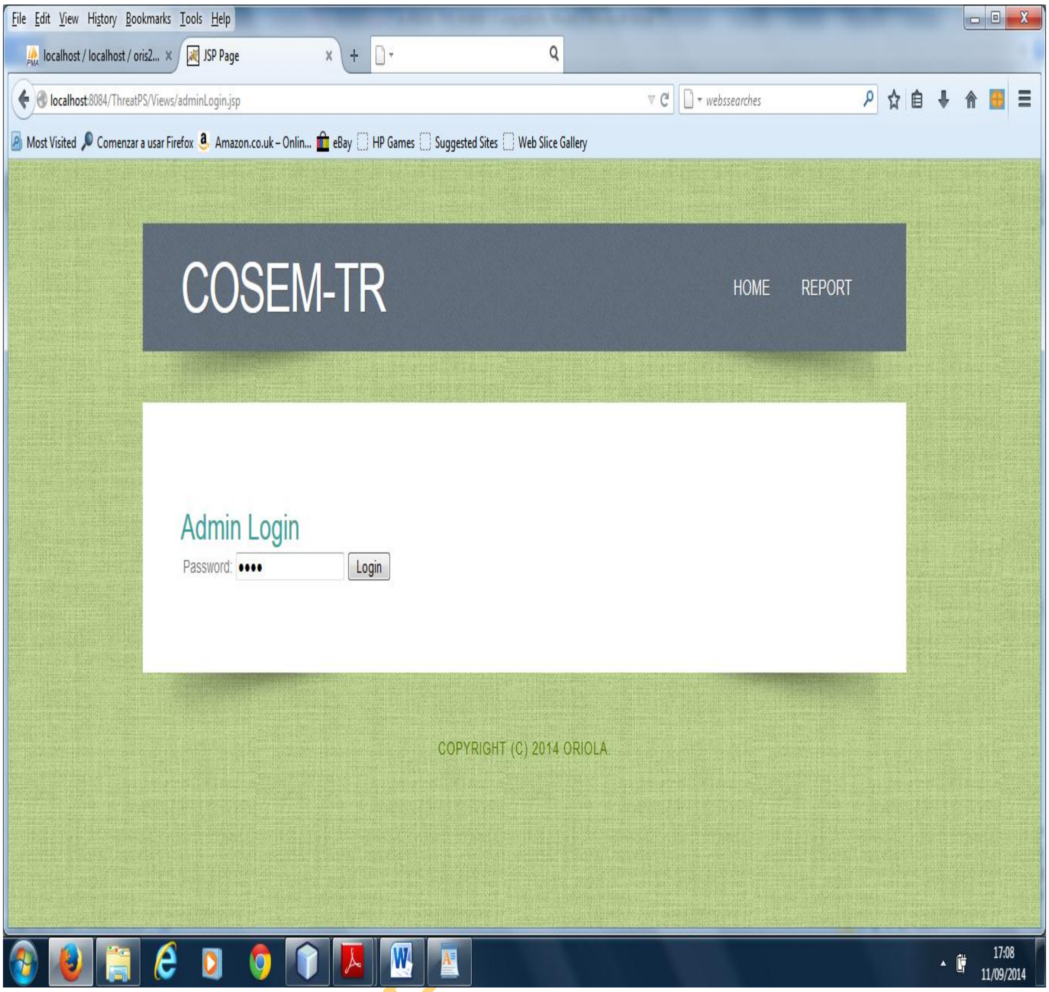
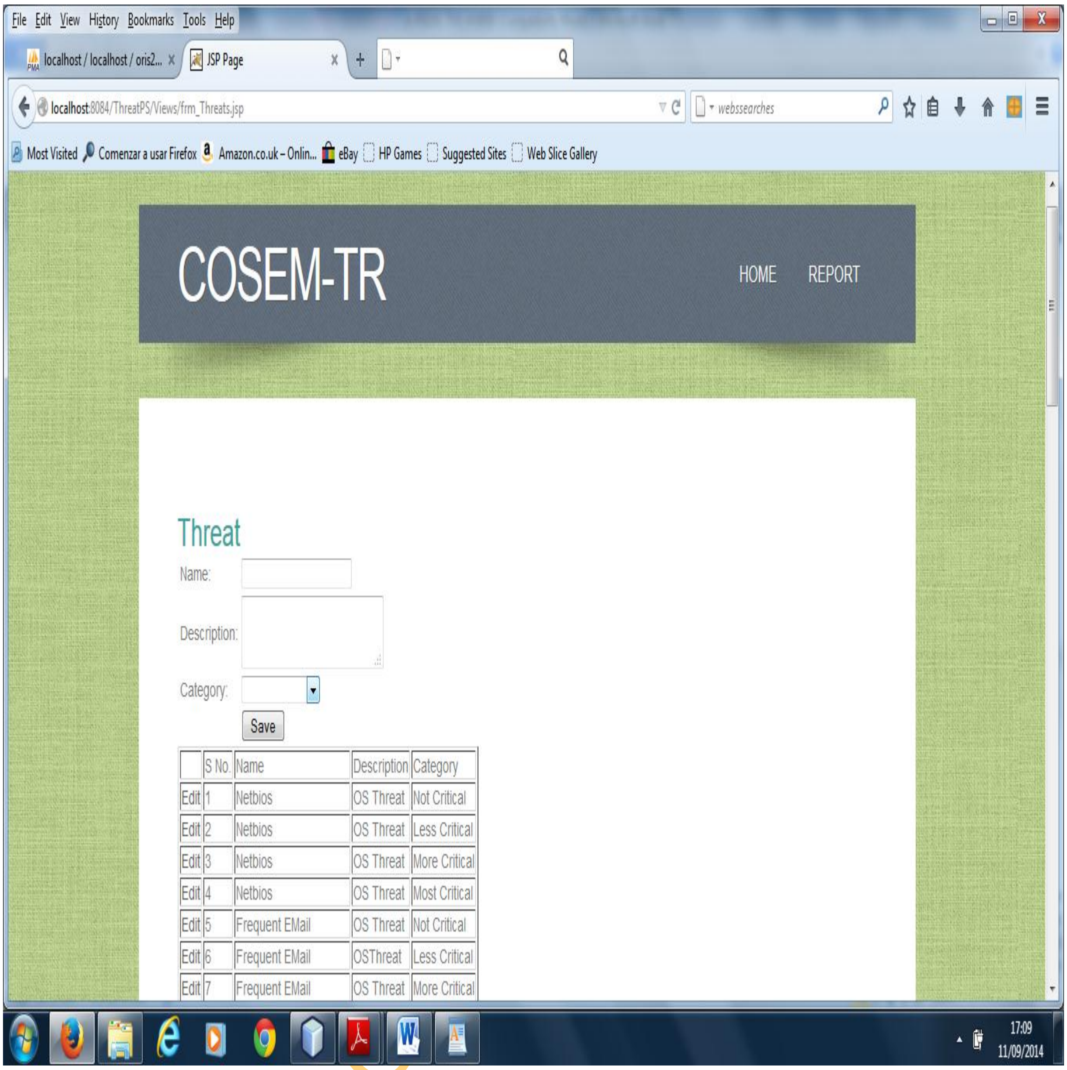


Figure 3.21: Administrator Login Page



Threat

Name:

Description:

Category:

Save

	S No.	Name	Description	Category
Edit	1	Netbios	OS Threat	Not Critical
Edit	2	Netbios	OS Threat	Less Critical
Edit	3	Netbios	OS Threat	More Critical
Edit	4	Netbios	OS Threat	Most Critical
Edit	5	Frequent EMail	OS Threat	Not Critical
Edit	6	Frequent EMail	OSThreat	Less Critical
Edit	7	Frequent EMail	OS Threat	More Critical

Figure 3.22: Threat Loading and Configuration Page

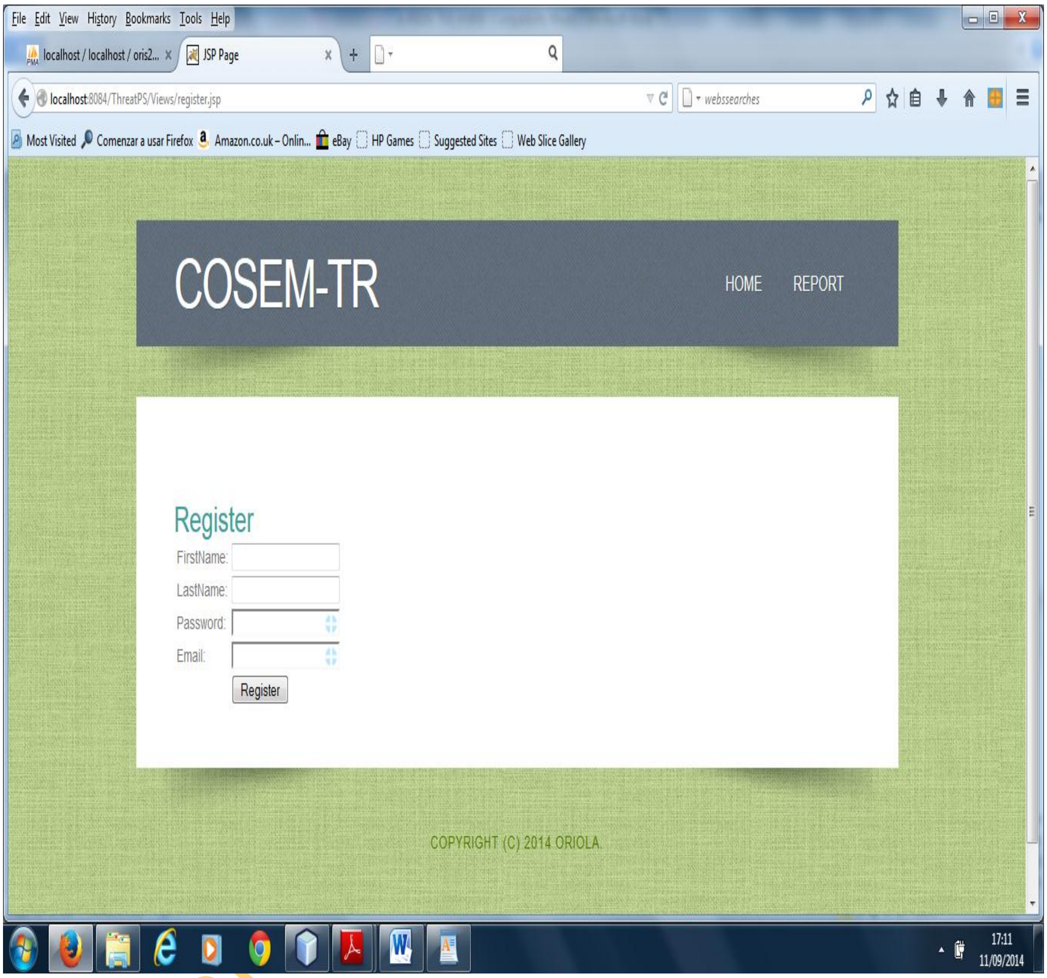

localhost:8084/ThreatPS/Views/register.jsp
COSEM-TR
HOME
REPORT
Register
FirstName:
LastName:
Password:
Email:
Register
COPYRIGHT (C) 2014 ORIOLA.

Figure 3.23: Security Manager Registration Page

Figure 3.24, Figure 3.25 and Figure 3.26 are used by security manager to assign perceptions while Figure 3.27, Figure 3.28 and Figure 3.29 are managed by the central administrator.

3.4.3.3 Implementation of Threat Mitigation Tool

The Threat Mitigation is implemented using Security Onion Tool. Security Onion is a Unified Threat Management System designed by Burk Doug in 2006 (Doug, 2006) to operate in both virtual and physical ubuntu 64bit operating system. It was developed in order to integrate different sensors and threat analysis tools in one single application. It contains OSSEC Host-based Intrusion Detection System, Bro Intrusion Detection System, Ubuntu-based Firewall, Snort and Suricata Network Intrusion Detection System (NIDS). Apart from these, it also has ELSA, Sguil, Squert, Xpico and Snorby.

3.4.4 The Security Onion Architecture for Threat Mitigation

For the purpose of this research, the security onion is operated in an oracle virtualbox, developed by Oracle Inc. We configure only Snort and Suricata NIDS because of their different capabilities, convenience of use, interoperation and open source nature. The Security Onion Framework for Threat Mitigation is presented in Figure 3.30. In order to execute the Security Onion Tool, the following steps are taken:

a. Create VM on VirtualBox: To install SecurityOnion on VirtualBox, first we create a new virtual machine as presented in Figure 3.31.

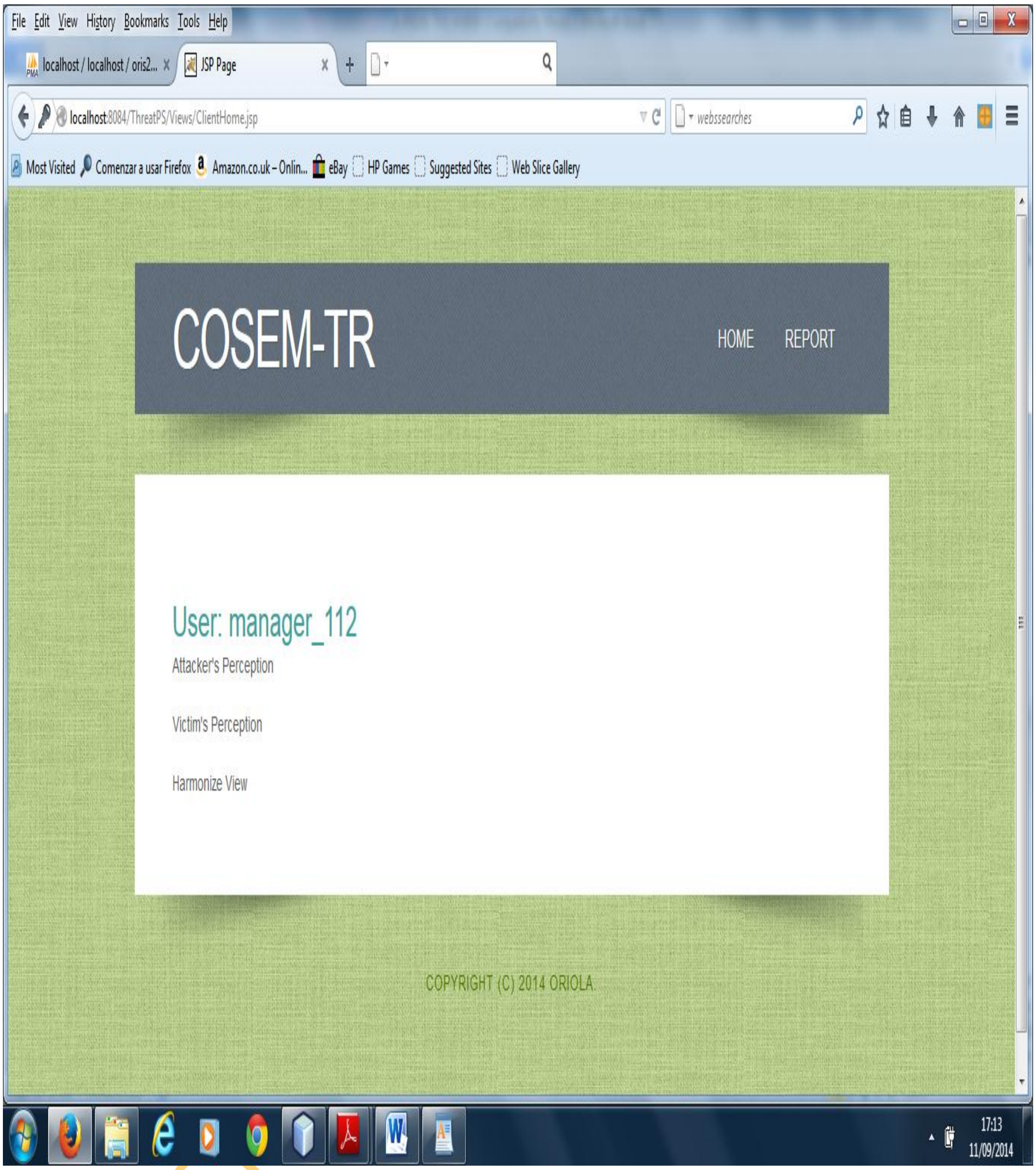
b. Installing Security Onion: The following steps were taken to install the security onion:

i. Double Click On Install Security Onion Icon and Follow the installation menu as presented in Figure 3.32.

ii. Set up Security Onion

iii. After the reboot, we login and complete the Security Onion setup by clicking on set-up as presented in Figure 3.33 and Figure 3.34 respectively.

c. Then select "*Yes, configure /etc/network/interfaces!*" in order to continue as presented in Figure 3.35.



COSEM-TR

User: manager_112

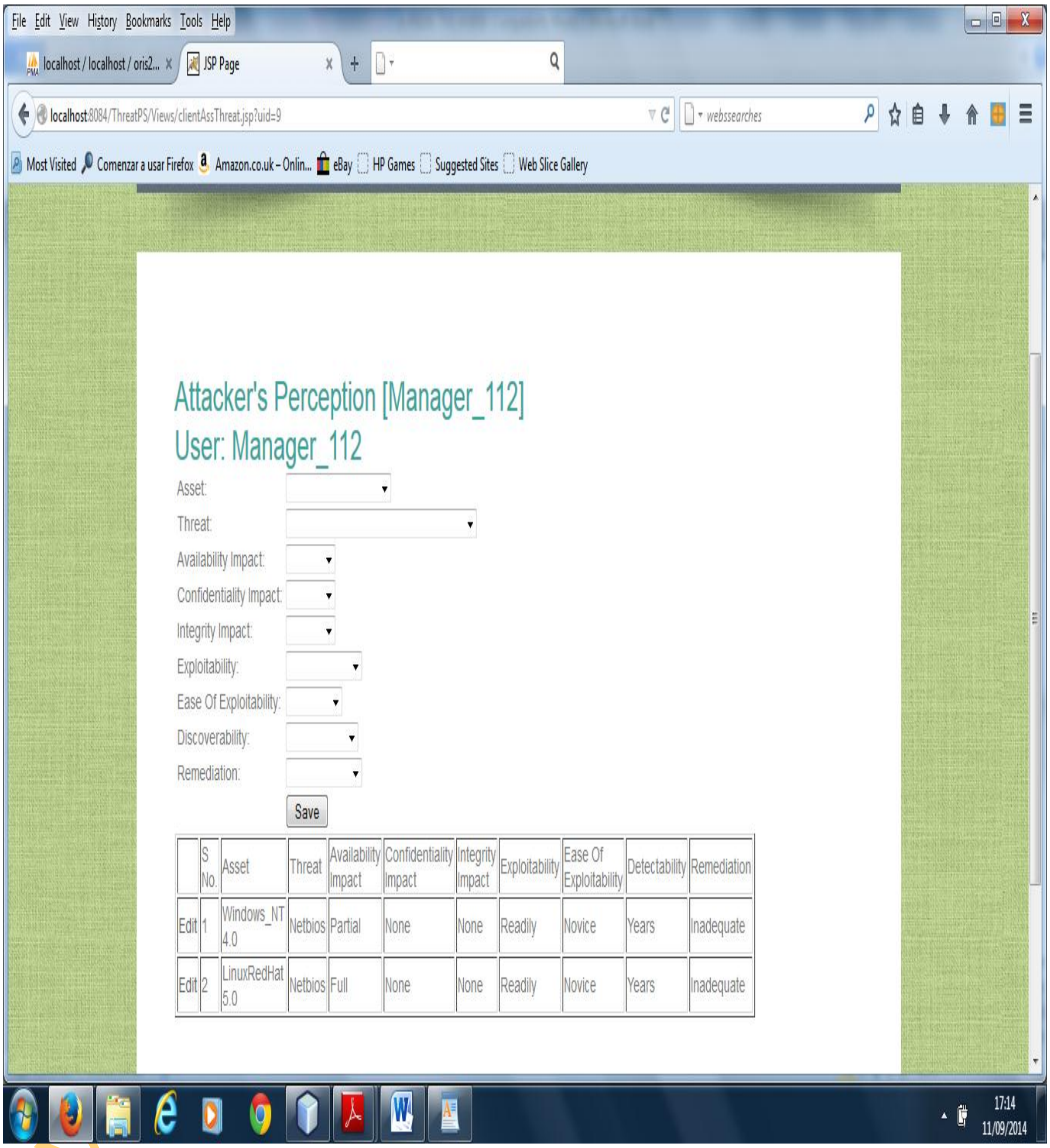
Attacker's Perception

Victim's Perception

Harmonize View

COPYRIGHT (C) 2014 ORIOLA

Figure 3.24: Security Manager Account Page



Attacker's Perception [Manager_112]

User: Manager_112

Asset:

Threat:

Availability Impact:

Confidentiality Impact:

Integrity Impact:

Exploitability:

Ease Of Exploitability:

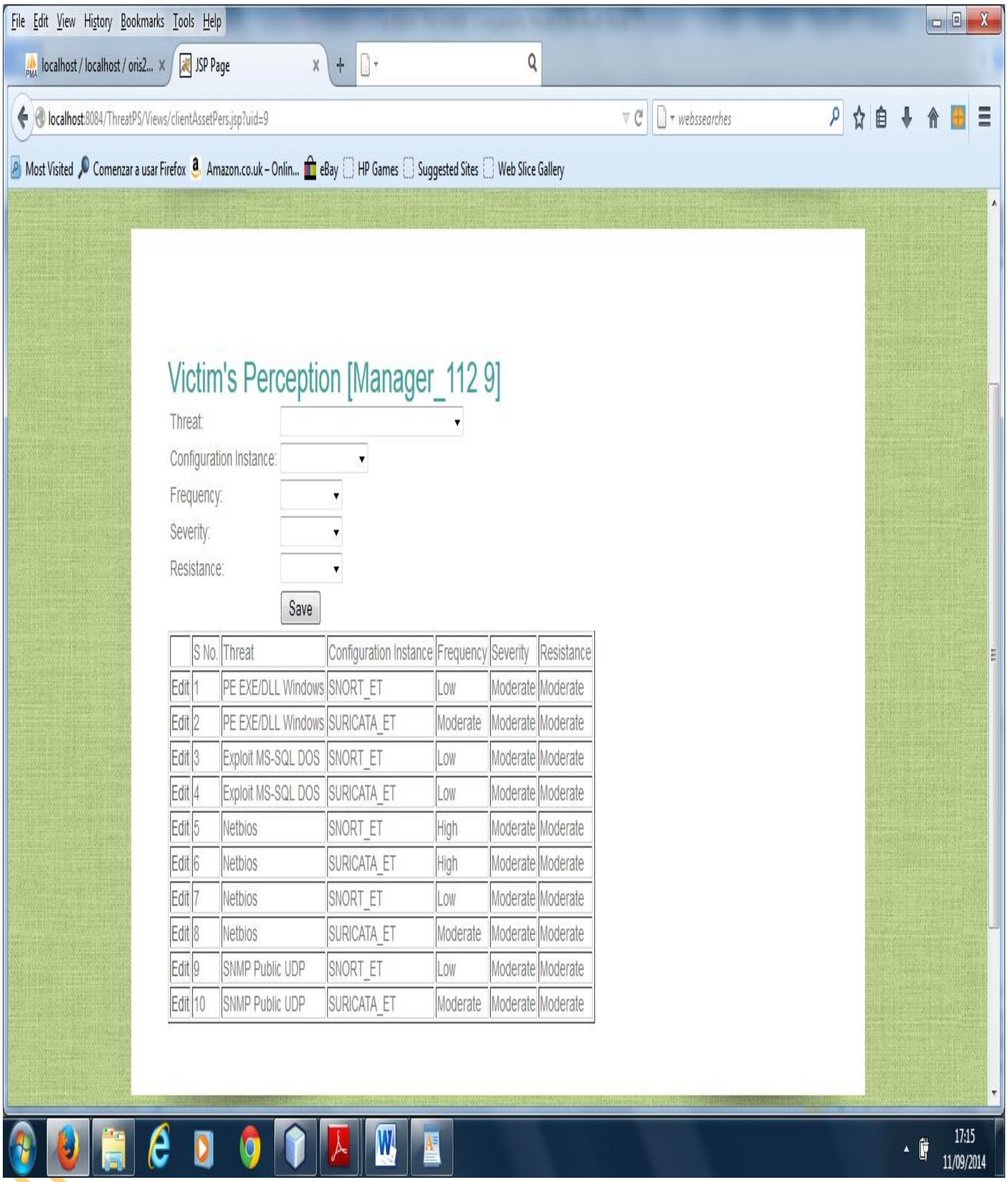
Discoverability:

Remediation:

Save

	S No.	Asset	Threat	Availability Impact	Confidentiality Impact	Integrity Impact	Exploitability	Ease Of Exploitability	Detectability	Remediation
Edit	1	Windows_NT 4.0	Netbios	Partial	None	None	Readily	Novice	Years	Inadequate
Edit	2	LinuxRedHat 5.0	Netbios	Full	None	None	Readily	Novice	Years	Inadequate

Figure 3.25: Security Manager Attacker’s Perspection Input Page



Victim's Perception [Manager_112 9]

Threat:

Configuration Instance:

Frequency:

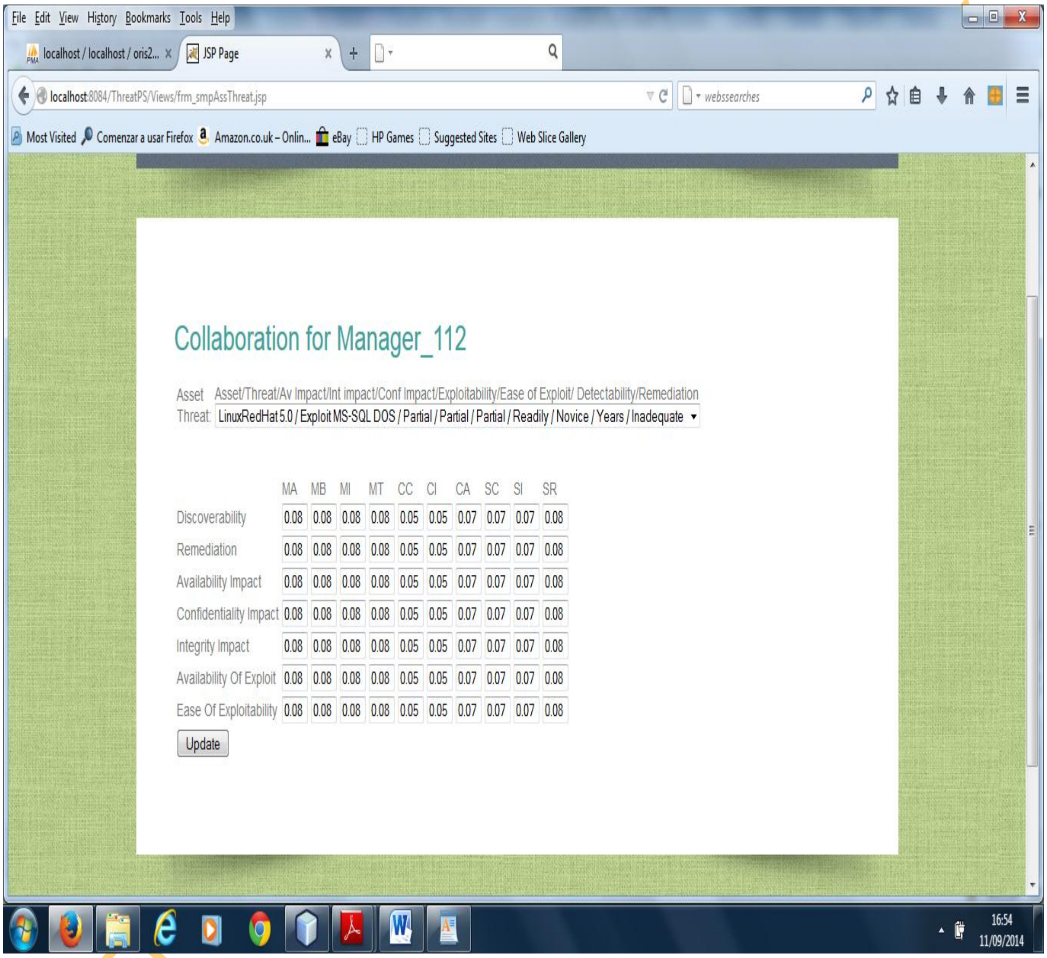
Severity:

Resistance:

Save

	S No.	Threat	Configuration Instance	Frequency	Severity	Resistance
Edit	1	PE EXE/DLL Windows	SNORT_ET	Low	Moderate	Moderate
Edit	2	PE EXE/DLL Windows	SURICATA_ET	Moderate	Moderate	Moderate
Edit	3	Exploit MS-SQL DOS	SNORT_ET	Low	Moderate	Moderate
Edit	4	Exploit MS-SQL DOS	SURICATA_ET	Low	Moderate	Moderate
Edit	5	Netbios	SNORT_ET	High	Moderate	Moderate
Edit	6	Netbios	SURICATA_ET	High	Moderate	Moderate
Edit	7	Netbios	SNORT_ET	Low	Moderate	Moderate
Edit	8	Netbios	SURICATA_ET	Moderate	Moderate	Moderate
Edit	9	SNMP Public UDP	SNORT_ET	Low	Moderate	Moderate
Edit	10	SNMP Public UDP	SURICATA_ET	Moderate	Moderate	Moderate

Figure 3.26: Security Manager Victim's Perception Page



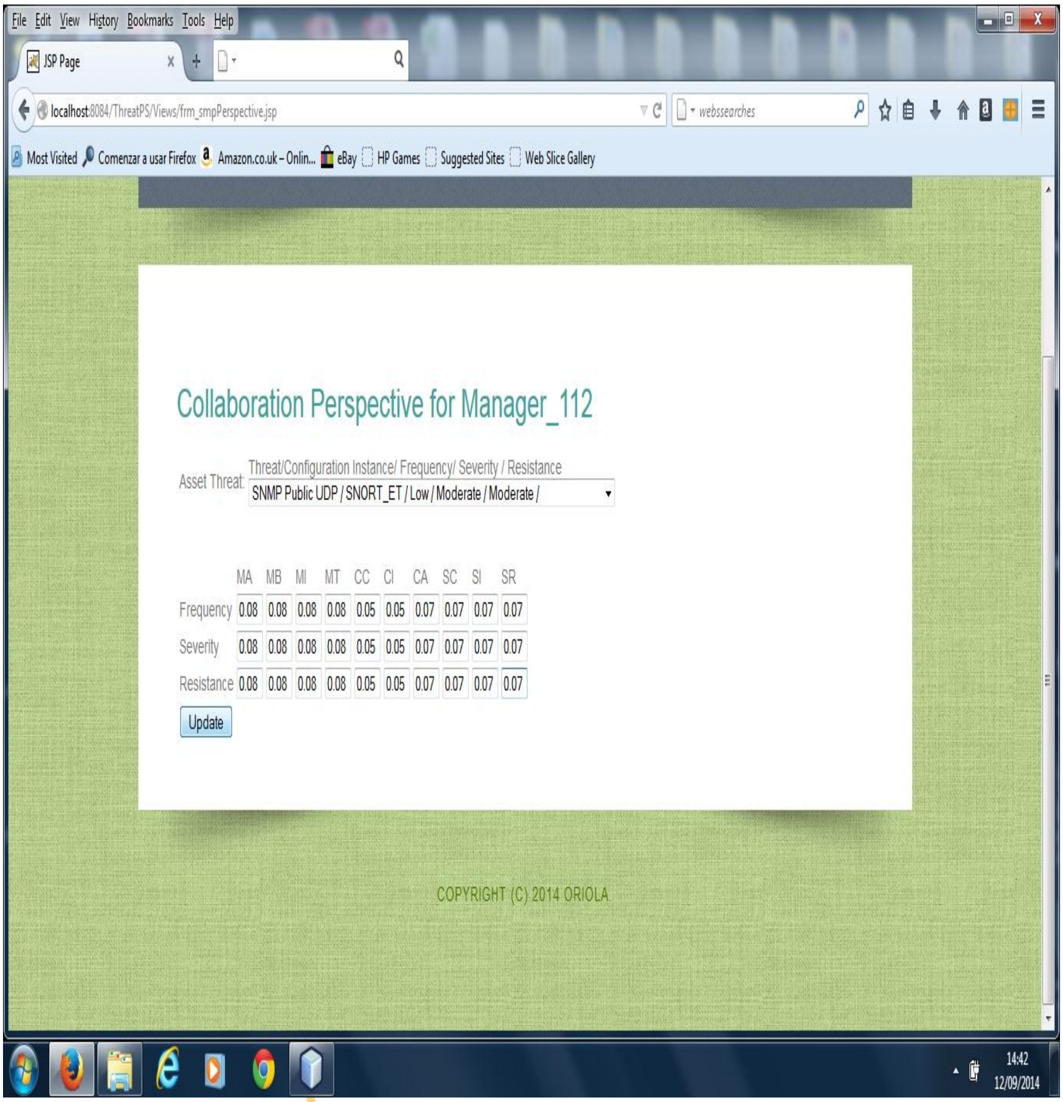
Collaboration for Manager_112

Asset Threat: Asset/Threat/Av Impact/Int impact/Conf Impact/Exploitability/Ease of Exploit/ Detectability/Remediation

LinuxRedHat 5.0 / Exploit MS-SQL DOS / Partial / Partial / Partial / Readily / Novice / Years / Inadequate

	MA	MB	MI	MT	CC	CI	CA	SC	SI	SR
Discoverability	0.08	0.08	0.08	0.08	0.05	0.05	0.07	0.07	0.07	0.08
Remediation	0.08	0.08	0.08	0.08	0.05	0.05	0.07	0.07	0.07	0.08
Availability Impact	0.08	0.08	0.08	0.08	0.05	0.05	0.07	0.07	0.07	0.08
Confidentiality Impact	0.08	0.08	0.08	0.08	0.05	0.05	0.07	0.07	0.07	0.08
Integrity Impact	0.08	0.08	0.08	0.08	0.05	0.05	0.07	0.07	0.07	0.08
Availability Of Exploit	0.08	0.08	0.08	0.08	0.05	0.05	0.07	0.07	0.07	0.08
Ease Of Exploitability	0.08	0.08	0.08	0.08	0.05	0.05	0.07	0.07	0.07	0.08

Figure 3.27: Central Administrator Attacker's PerceptionTrust Management Page

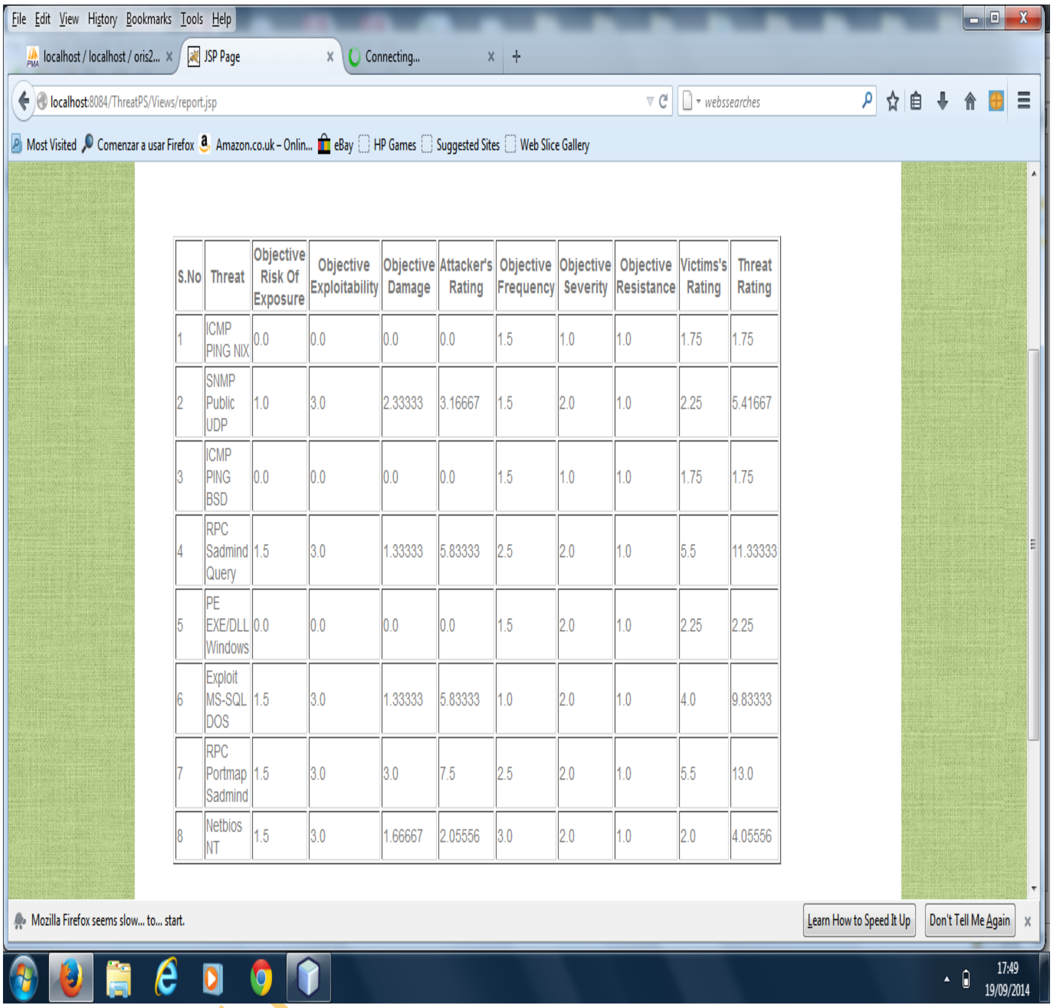


Collaboration Perspective for Manager_112

Asset Threat: Threat/Configuration Instance/ Frequency/ Severity / Resistance
SNMP Public UDP / SNORT_ET / Low / Moderate / Moderate /

	MA	MB	MI	MT	CC	CI	CA	SC	SI	SR
Frequency	0.08	0.08	0.08	0.08	0.05	0.05	0.07	0.07	0.07	0.07
Severity	0.08	0.08	0.08	0.08	0.05	0.05	0.07	0.07	0.07	0.07
Resistance	0.08	0.08	0.08	0.08	0.05	0.05	0.07	0.07	0.07	0.07

Figure 3.28: Central Administrator Victim's PerceptionTrust Management Page

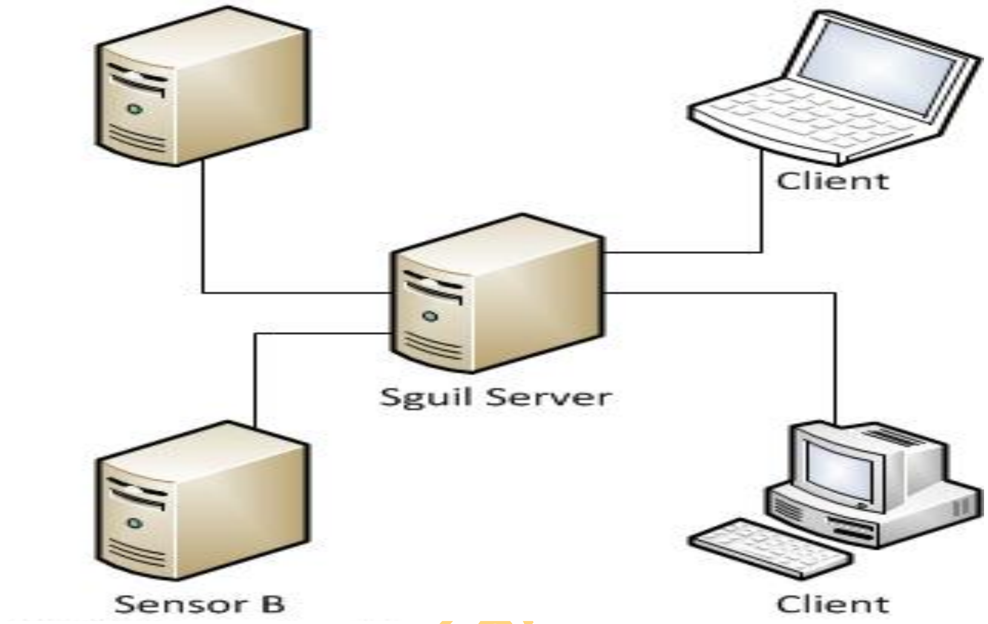


S.No	Threat	Objective Risk Of Exposure	Objective Exploitability	Objective Damage	Attacker's Rating	Objective Frequency	Objective Severity	Objective Resistance	Victims's Rating	Threat Rating
1	ICMP PING NIX	0.0	0.0	0.0	0.0	1.5	1.0	1.0	1.75	1.75
2	SNMP Public UDP	1.0	3.0	2.33333	3.16667	1.5	2.0	1.0	2.25	5.41667
3	ICMP PING BSD	0.0	0.0	0.0	0.0	1.5	1.0	1.0	1.75	1.75
4	RPC Sadmind Query	1.5	3.0	1.33333	5.83333	2.5	2.0	1.0	5.5	11.33333
5	PE EXE/DLL Windows	0.0	0.0	0.0	0.0	1.5	2.0	1.0	2.25	2.25
6	Exploit MS-SQL DOS	1.5	3.0	1.33333	5.83333	1.0	2.0	1.0	4.0	9.83333
7	RPC Portmap Sadmind	1.5	3.0	3.0	7.5	2.5	2.0	1.0	5.5	13.0
8	Netbios NT	1.5	3.0	1.66667	2.05556	3.0	2.0	1.0	2.0	4.05556

Figure 3.29: Threat Rating Report

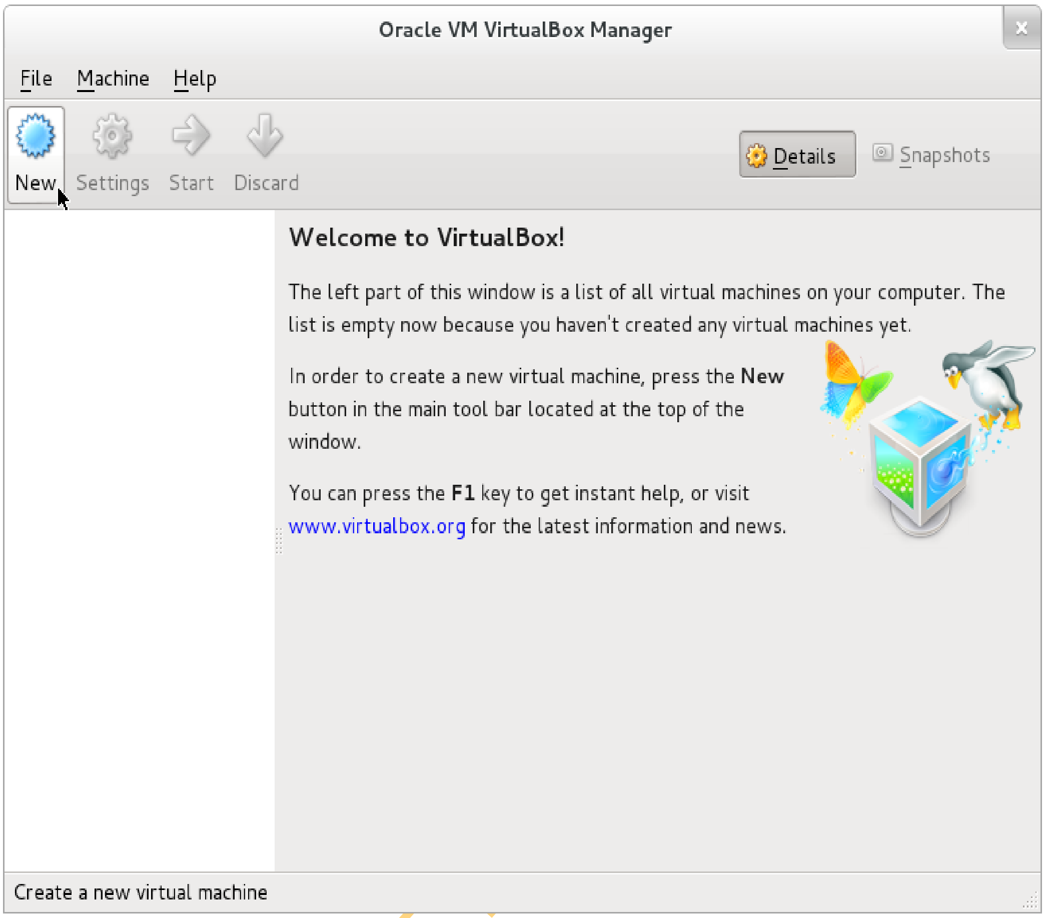
Suricata

Sensor A



Snort

Figure 3.30: Security Onion Framework for Threat Mitigation



Welcome to VirtualBox!

The left part of this window is a list of all virtual machines on your computer. The list is empty now because you haven't created any virtual machines yet.

In order to create a new virtual machine, press the **New** button in the main tool bar located at the top of the window.

You can press the **F1** key to get instant help, or visit www.virtualbox.org for the latest information and news.

Create a new virtual machine

Figure 3.31: Creation of New Virtual Machine

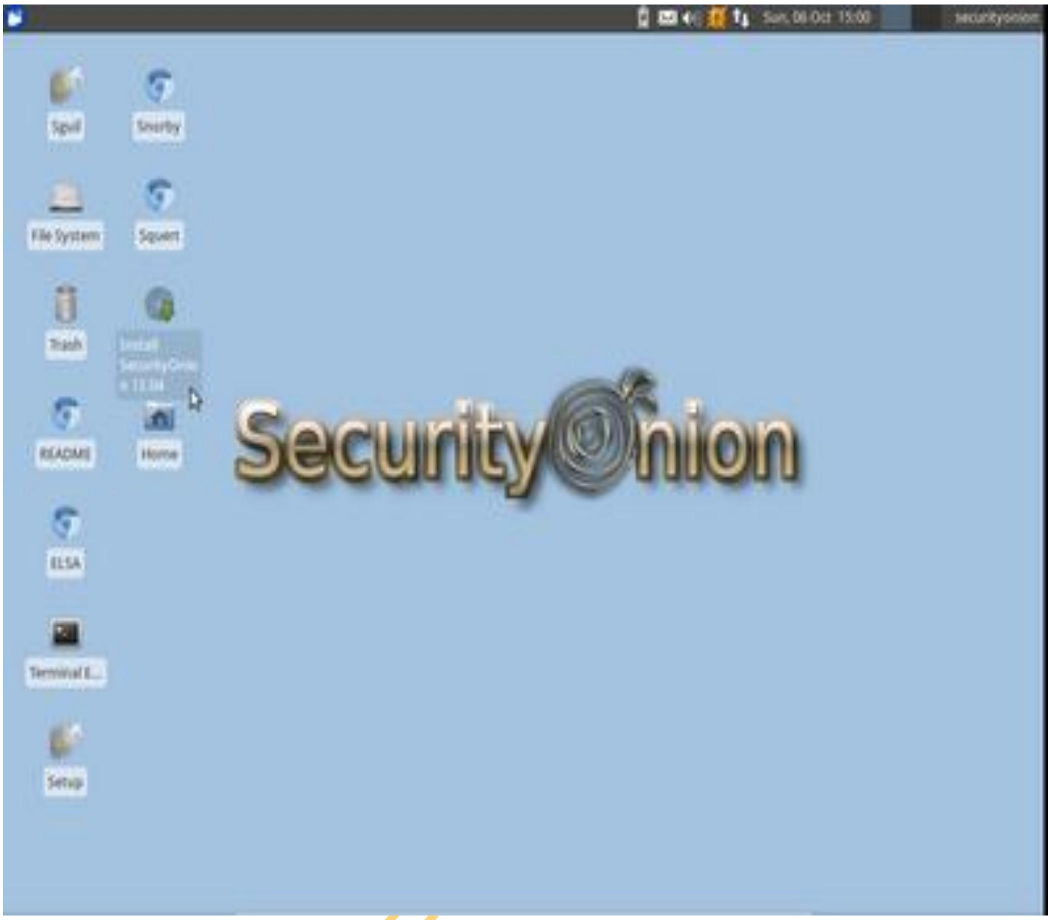
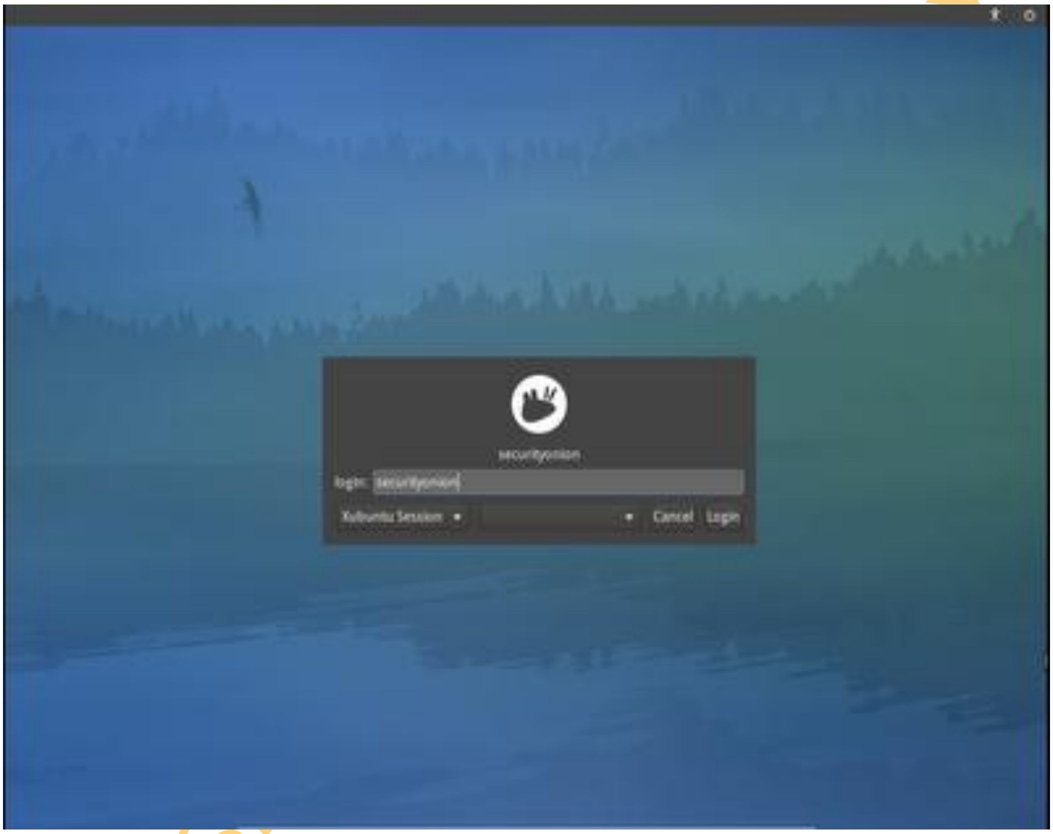


Figure 3.32: Security Onion Home Screen



login: securityonion

Figure 3.33: Security Onion Login Page

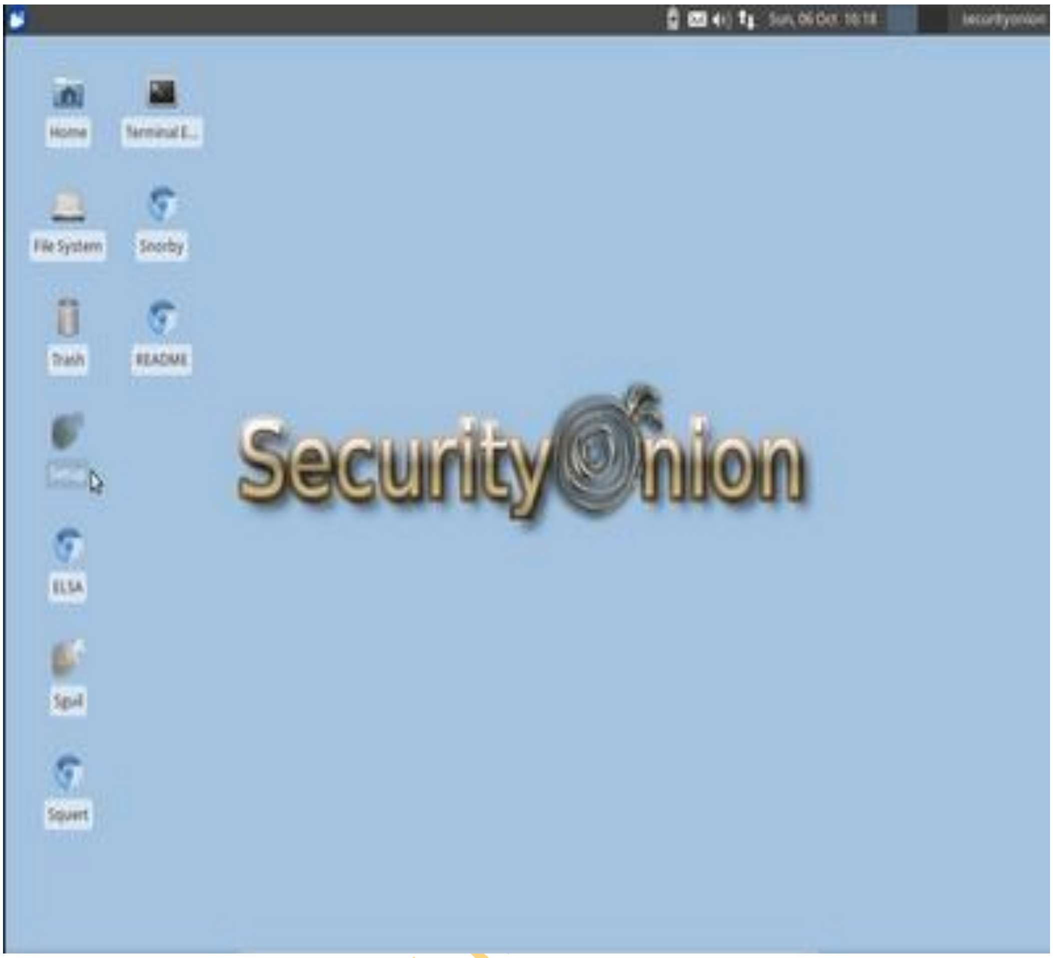
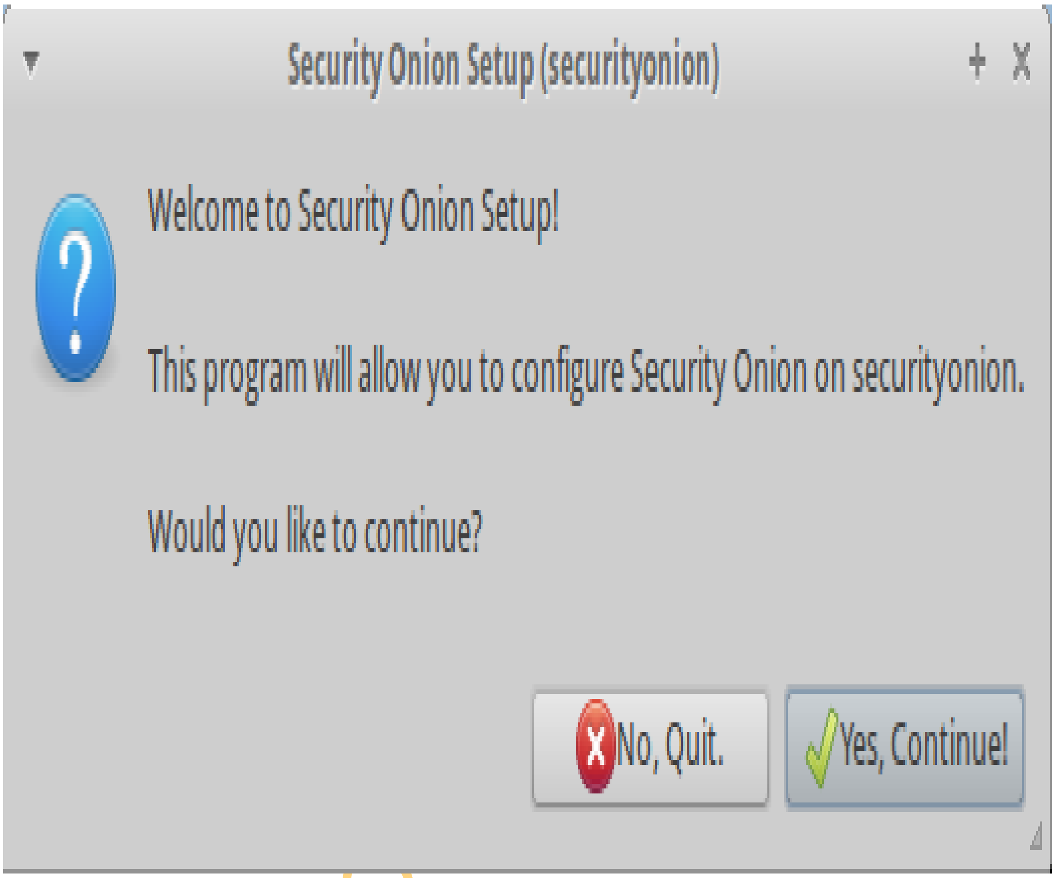


Figure 3.34: Security Onion Set-up Configuration



Welcome to Security Onion Setup!

This program will allow you to configure Security Onion on securityonion.

Would you like to continue?

Figure 3.35: Security Onion Set-up

Select the management interface that will be used to access, administer, and monitor your Security Onion platform. In our particular case we use eth0 for the management interface as presented in Figure 3.36.

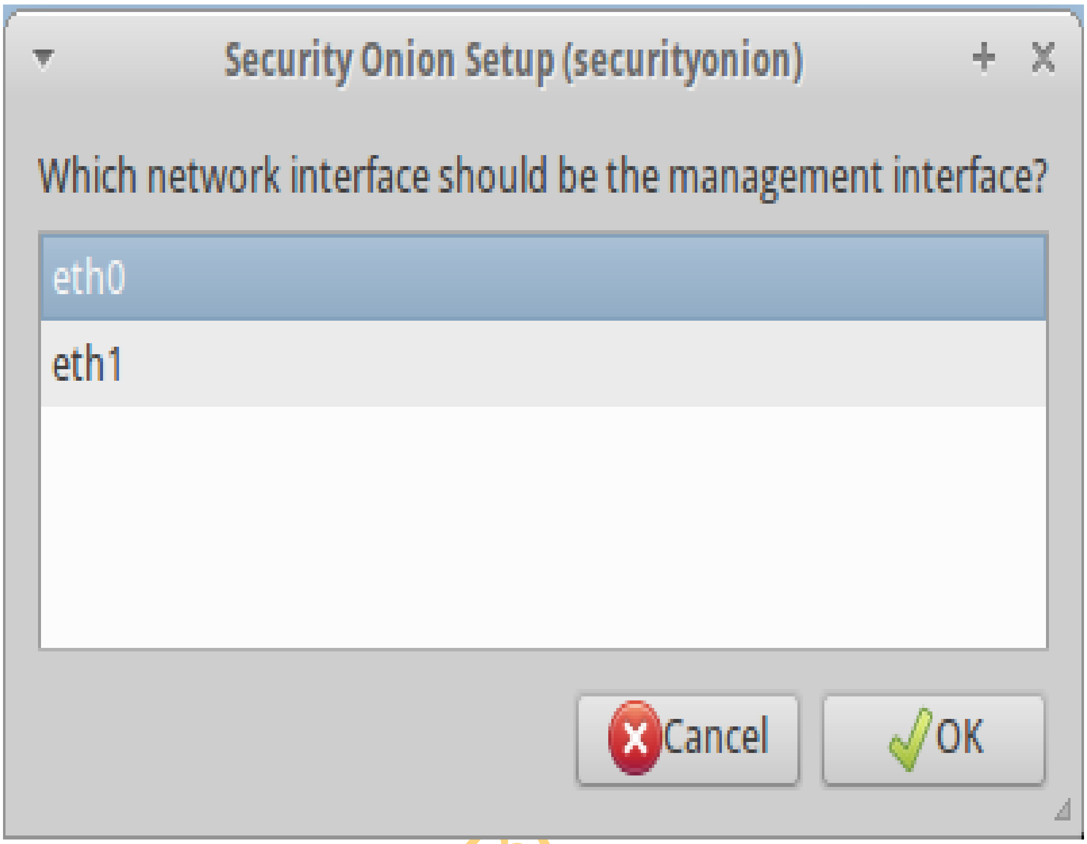
Once the management interface is configured, configure the capture/monitoring interface. To do this press "*Yes, configure monitor interfaces*" in Figure 3.37.

Then, mark checkbox eth1 and then press "*OK*" in Figure 3.38. We then proceed with the Advanced Setup by selecting the "Advanced Setup" radio button In Figure 3.39 and then by pressing "*OK*".

Figure 3.40 specifies which IDS Engine (Snort or Suricata) we would like to use.

Next in Figure 3.41, configure "*Emerging Threats GPL*" ruleset both Snort and Suricata NIDS. We are then required to enable the NIDS we previously selected simply by pressing "*Yes, enable the IDS Engine*!" in IDS Engine Enable Page
In Figure 3.42

The results of the Threat Modelling are used employed to reconfigure the Snort and Suricata Network Intrusion Detection System. Only the low priority threat intrusion rules among the Minor Threat original rules would be configured. After the security onion is set up, the tcpdump .pcap file are replayed against the new. The processing of the rules configuration is presented in Figure 3.43. In other to detect intrusion for the purpose of this research TCPReplay in the Security Onion is employed. The traffic replay terminal is presented in Figure 3.44. The IDS Alert outputs are then observed using Sguil, Elsa and Snorby as demonstrated in Figure 3.45, Figure 3.46 and Figure 3.47 respectively.



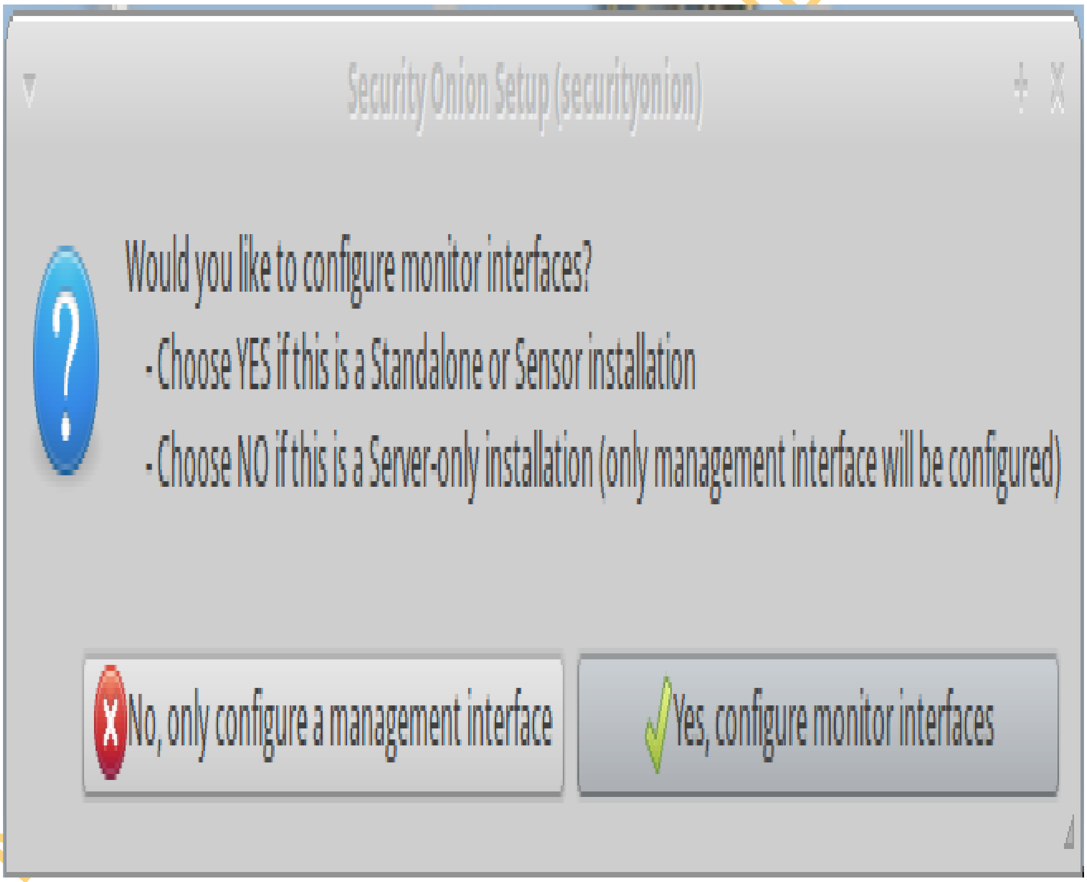
Security Onion Setup (securityonion)

Which network interface should be the management interface?

eth0

eth1

Figure 3.36: Selection of Management Interface



Security Onion Setup (securityonion)

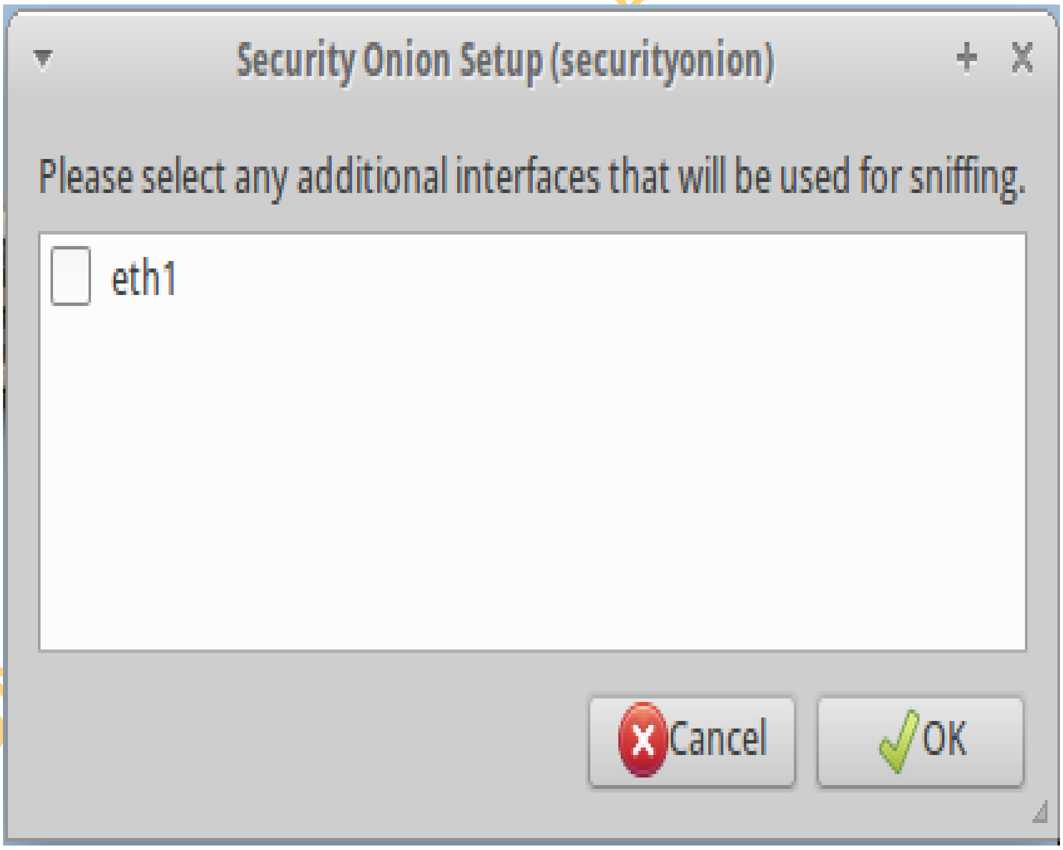
Would you like to configure monitor interfaces?

- Choose YES if this is a Standalone or Sensor installation

- Choose NO if this is a Server-only installation (only management interface will be configured)

No, only configure a management interface
Yes, configure monitor interfaces

Figure 3.37: Configuration of Monitoring Interface

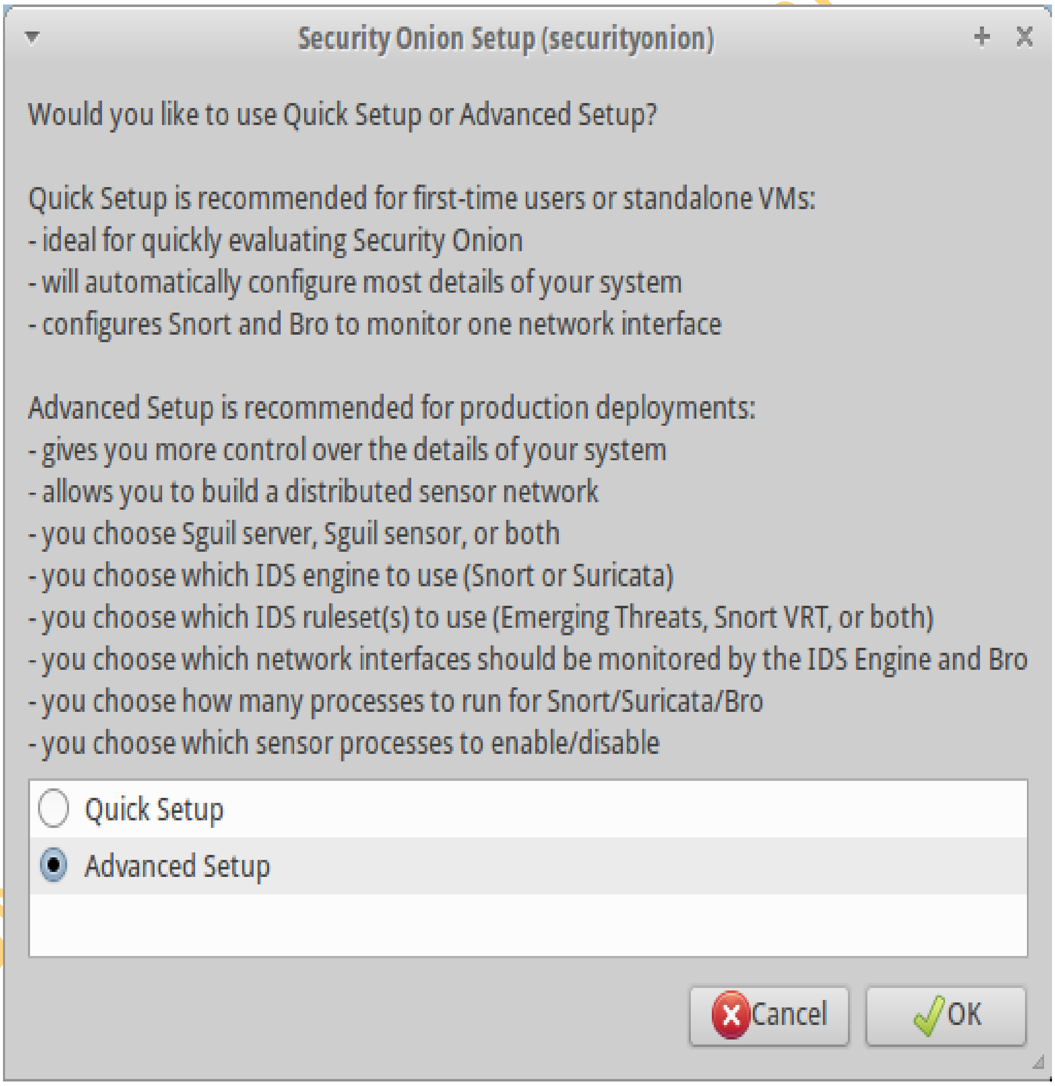


Security Onion Setup (securityonion)

Please select any additional interfaces that will be used for sniffing.

□ eth1

Figure 3.38: Check Monitoring Interface



Would you like to use Quick Setup or Advanced Setup?

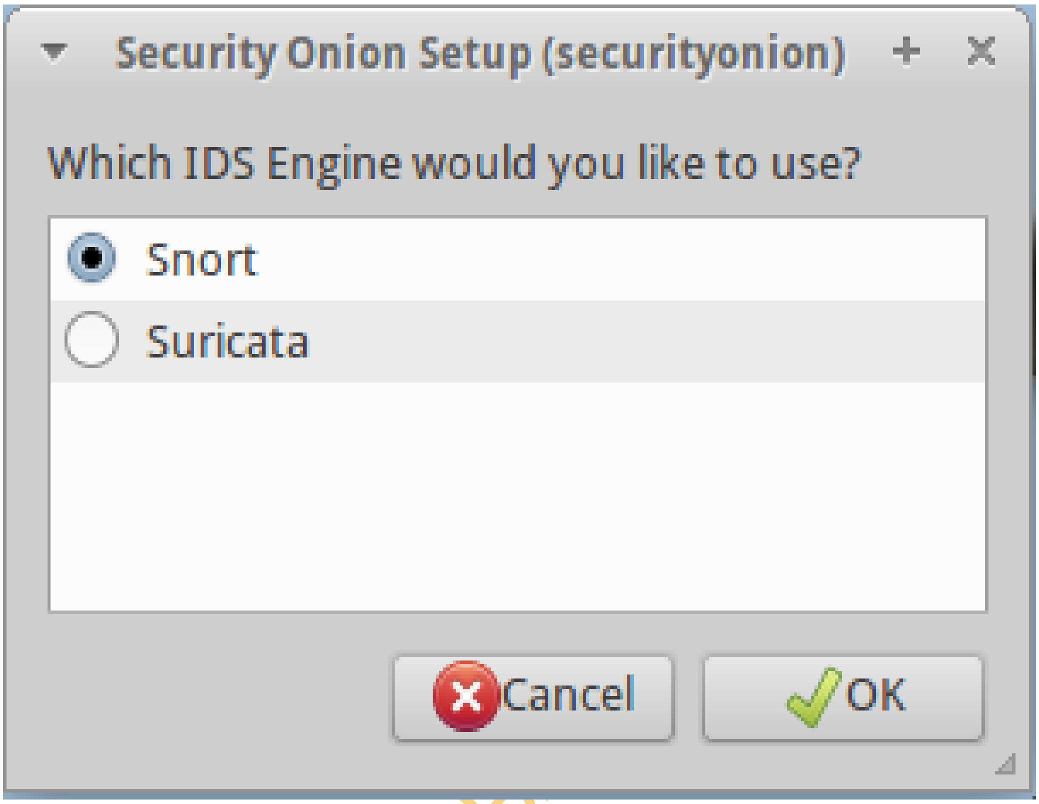
Quick Setup is recommended for first-time users or standalone VMs:
- ideal for quickly evaluating Security Onion
- will automatically configure most details of your system
- configures Snort and Bro to monitor one network interface

Advanced Setup is recommended for production deployments:
- gives you more control over the details of your system
- allows you to build a distributed sensor network
- you choose Sguil server, Sguil sensor, or both
- you choose which IDS engine to use (Snort or Suricata)
- you choose which IDS ruleset(s) to use (Emerging Threats, Snort VRT, or both)
- you choose which network interfaces should be monitored by the IDS Engine and Bro
- you choose how many processes to run for Snort/Suricata/Bro
- you choose which sensor processes to enable/disable

○ Quick Setup

● Advanced Setup

Figure 3.39: Advanced Set-up



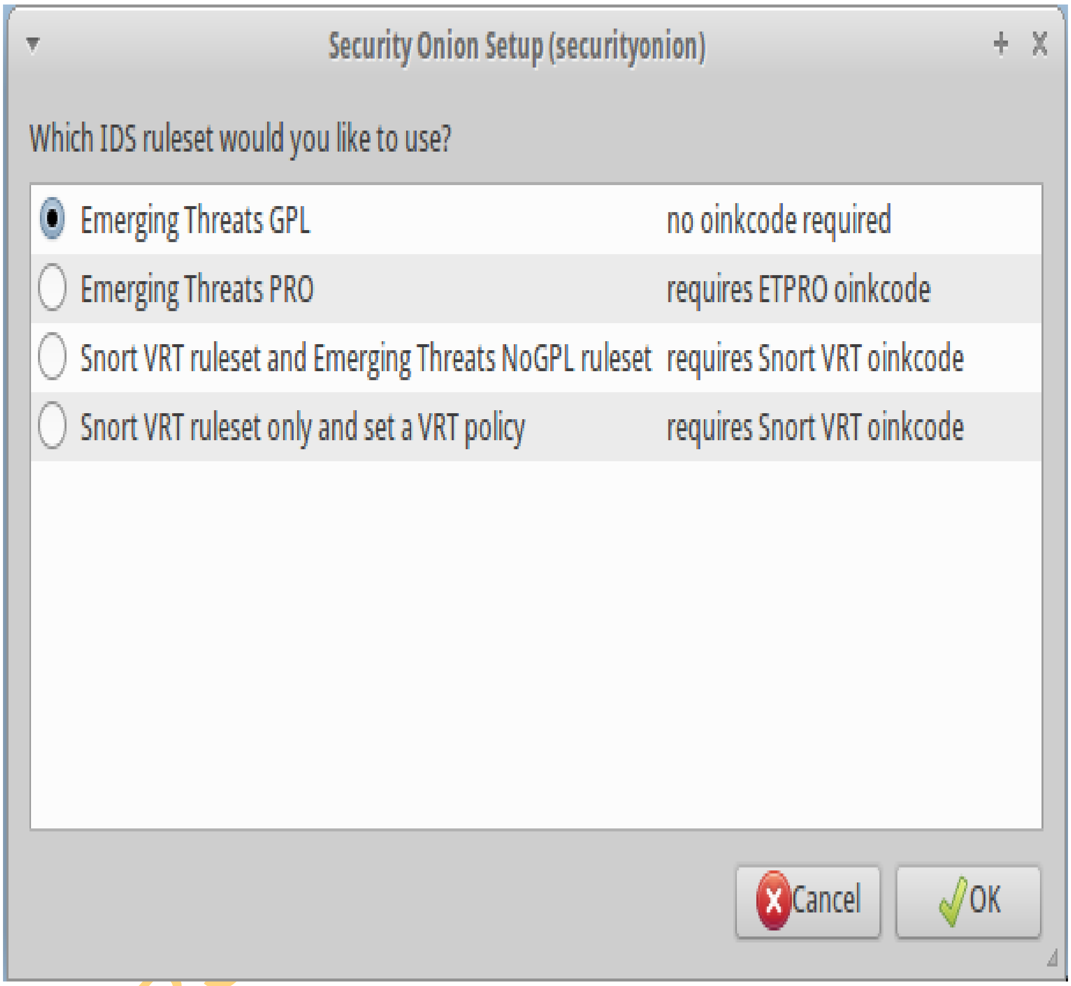
Security Onion Setup (securityonion)

Which IDS Engine would you like to use?

● Snort

○ Suricata

Figure 3.40: IDS Selection Page

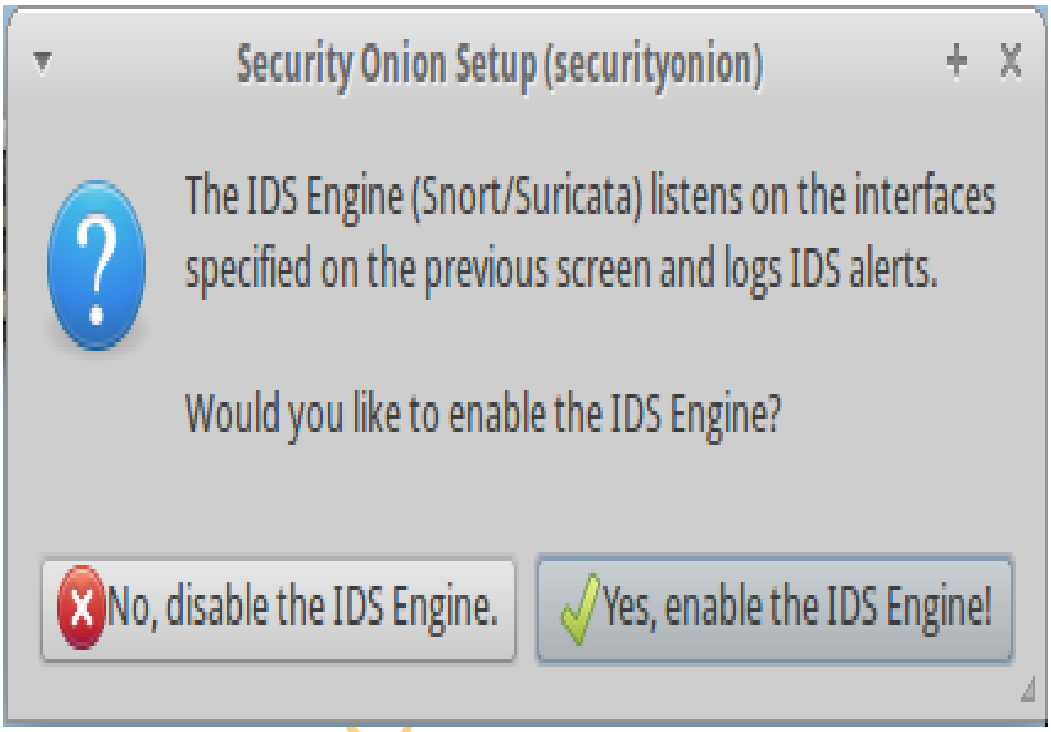


Security Onion Setup (securityonion)

Which IDS ruleset would you like to use?

●	Emerging Threats GPL	no oinkcode required
○	Emerging Threats PRO	requires ETPRO oinkcode
○	Snort VRT ruleset and Emerging Threats NoGPL ruleset	requires Snort VRT oinkcode
○	Snort VRT ruleset only and set a VRT policy	requires Snort VRT oinkcode

Figure 3.41: Security Onion IDS Ruleset Configuration Page

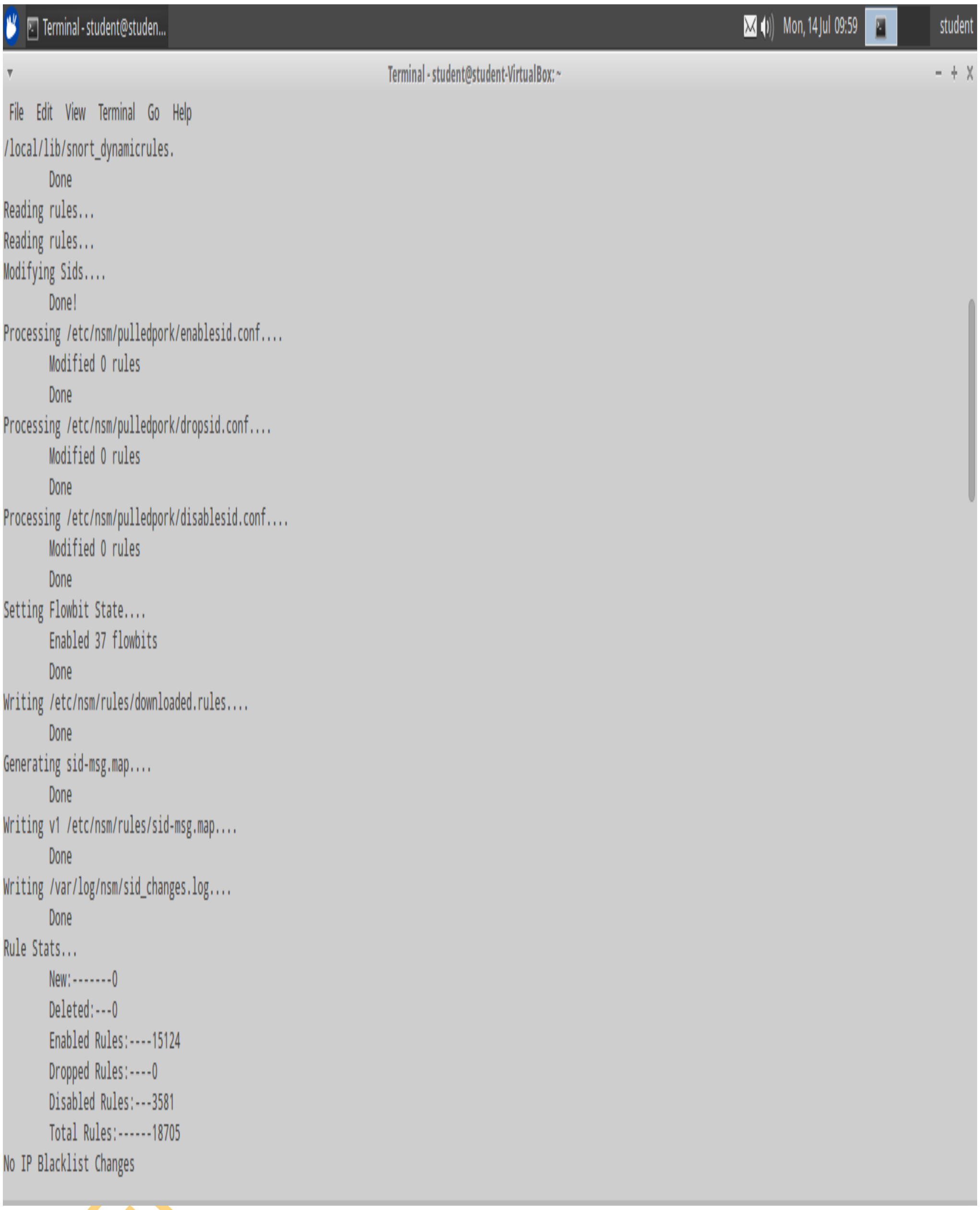


Security Onion Setup (securityonion)

The IDS Engine (Snort/Suricata) listens on the interfaces specified on the previous screen and logs IDS alerts.

Would you like to enable the IDS Engine?

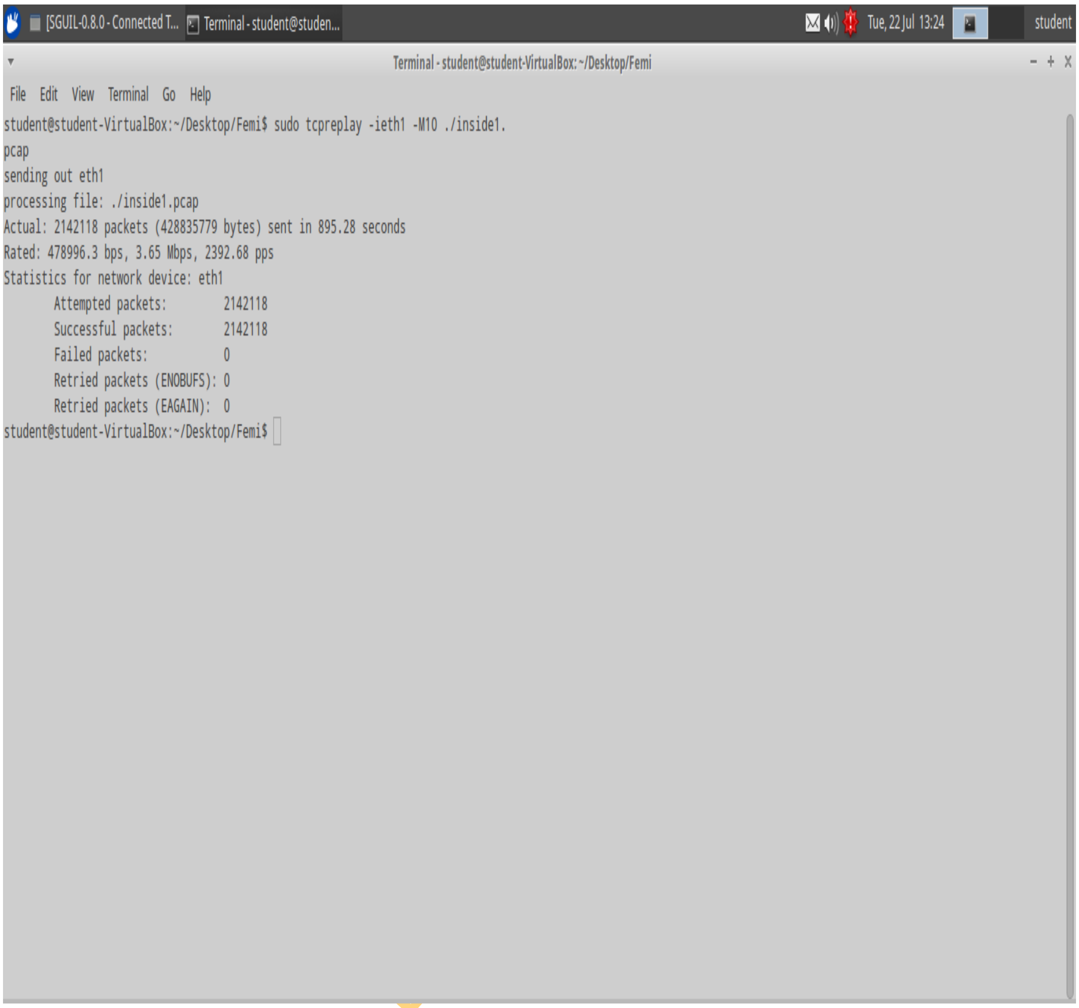
Figure 3.42: IDS Engine Enable Page



Terminal - student@studen...
Mon, 14 Jul 09:59
student
Terminal - student@student-VirtualBox: ~
File Edit View Terminal Go Help

```
/local/lib/snort_dynamicrules.
        Done
Reading rules...
Reading rules...
Modifying Sids....
        Done!
Processing /etc/nsm/pulledpork/enablesid.conf....
        Modified 0 rules
        Done
Processing /etc/nsm/pulledpork/dropsid.conf....
        Modified 0 rules
        Done
Processing /etc/nsm/pulledpork/disablesid.conf....
        Modified 0 rules
        Done
Setting Flowbit State....
        Enabled 37 flowbits
        Done
Writing /etc/nsm/rules/downloaded.rules....
        Done
Generating sid-msg.map....
        Done
Writing v1 /etc/nsm/rules/sid-msg.map....
        Done
Writing /var/log/nsm/sid_changes.log....
        Done
Rule Stats...
        New:-------0
        Deleted:---0
        Enabled Rules:----15124
        Dropped Rules:----0
        Disabled Rules:---3581
        Total Rules:------18705
No IP Blacklist Changes
```

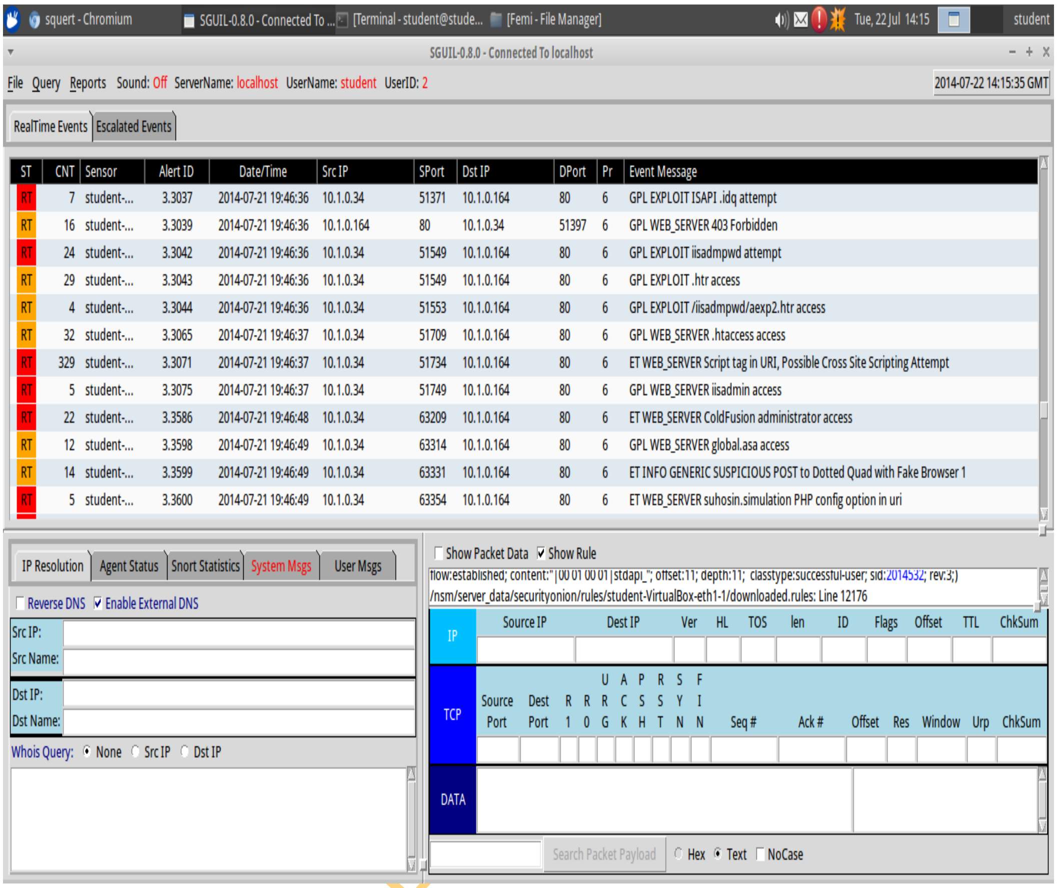
Figure 3.43: Signature Rule Update



[SGUIL-0.8.0 - Connected T...
Terminal - student@studen...
Tue, 22 Jul 13:24
student
Terminal - student@student-VirtualBox: ~/Desktop/Femi
File Edit View Terminal Go Help

```
student@student-VirtualBox:~/Desktop/Femi$ sudo tcpreplay -ieth1 -M10 ./inside1.
pcap
sending out eth1
processing file: ./inside1.pcap
Actual: 2142118 packets (428835779 bytes) sent in 895.28 seconds
Rated: 478996.3 bps, 3.65 Mbps, 2392.68 pps
Statistics for network device: eth1
        Attempted packets:         2142118
        Successful packets:        2142118
        Failed packets:            0
        Retried packets (ENOBUFS): 0
        Retried packets (EAGAIN):  0
student@student-VirtualBox:~/Desktop/Femi$
```

Figure 3.44: Security Onion Traffic Replay Terminal



Sound: Off ServerName: localhost UserName: student UserID: 2

2014-07-22 14:15:35 GMT

ST	CNT	Sensor	Alert ID	Date/Time	Src IP	SPort	Dst IP	DPort	Pr	Event Message
RT	7	student-...	3.3037	2014-07-21 19:46:36	10.1.0.34	51371	10.1.0.164	80	6	GPL EXPLOIT ISAPI .idq attempt
RT	16	student-...	3.3039	2014-07-21 19:46:36	10.1.0.164	80	10.1.0.34	51397	6	GPL WEB_SERVER 403 Forbidden
RT	24	student-...	3.3042	2014-07-21 19:46:36	10.1.0.34	51549	10.1.0.164	80	6	GPL EXPLOIT iisadmpwd attempt
RT	29	student-...	3.3043	2014-07-21 19:46:36	10.1.0.34	51549	10.1.0.164	80	6	GPL EXPLOIT .htr access
RT	4	student-...	3.3044	2014-07-21 19:46:36	10.1.0.34	51553	10.1.0.164	80	6	GPL EXPLOIT /iisadmpwd/aexp2.htr access
RT	32	student-...	3.3065	2014-07-21 19:46:37	10.1.0.34	51709	10.1.0.164	80	6	GPL WEB_SERVER .htaccess access
RT	329	student-...	3.3071	2014-07-21 19:46:37	10.1.0.34	51734	10.1.0.164	80	6	ET WEB_SERVER Script tag in URI, Possible Cross Site Scripting Attempt
RT	5	student-...	3.3075	2014-07-21 19:46:37	10.1.0.34	51749	10.1.0.164	80	6	GPL WEB_SERVER iisadmin access
RT	22	student-...	3.3586	2014-07-21 19:46:48	10.1.0.34	63209	10.1.0.164	80	6	ET WEB_SERVER ColdFusion administrator access
RT	12	student-...	3.3598	2014-07-21 19:46:49	10.1.0.34	63314	10.1.0.164	80	6	GPL WEB_SERVER global.asa access
RT	14	student-...	3.3599	2014-07-21 19:46:49	10.1.0.34	63331	10.1.0.164	80	6	ET INFO GENERIC SUSPICIOUS POST to Dotted Quad with Fake Browser 1
RT	5	student-...	3.3600	2014-07-21 19:46:49	10.1.0.34	63354	10.1.0.164	80	6	ET WEB_SERVER suhosin.simulation PHP config option in uri

□ Reverse DNS ☑ Enable External DNS

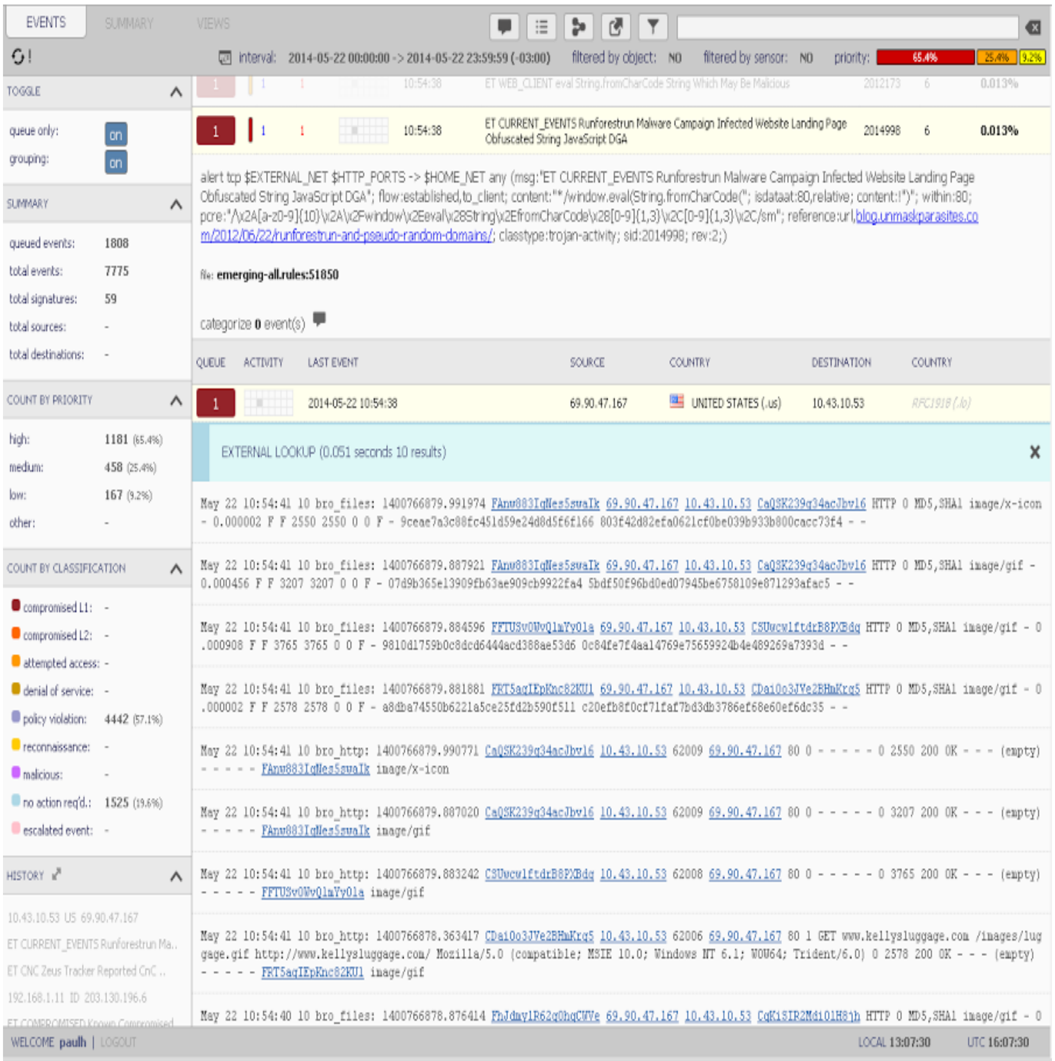
Whois Query: ● None ○ Src IP ○ Dst IP

□ Show Packet Data ☑ Show Rule

```
flow:established; content:"|00 01 00 01|stdapi_"; offset:11; depth:11;  classtype:successful-user; sid:2014532; rev:3;)
/nsm/server_data/securityonion/rules/student-VirtualBox-eth1-1/downloaded.rules: Line 12176
```

○ Hex ● Text □ NoCase

Figure 3.45: Sguil Interface



interval: 2014-05-22 00:00:00 -> 2014-05-22 23:59:59 (-03:00) filtered by object: NO filtered by sensor: NO priority: 65.4% 25.4% 9.2%

TOGGLE

queue only: on
grouping: on

SUMMARY

queued events:	1808
total events:	7775
total signatures:	59
total sources:	-
total destinations:	-

COUNT BY PRIORITY

high:	1181 (65.4%)
medium:	458 (25.4%)
low:	167 (9.2%)
other:	-

COUNT BY CLASSIFICATION

compromised L1:	-
compromised L2:	-
attempted access:	-
denial of service:	-
policy violation:	4442 (57.1%)
reconnaissance:	-
malicious:	-
no action req'd.:	1525 (19.6%)
escalated event:	-

HISTORY

10.43.10.53 US 69.90.47.167
ET CURRENT_EVENTS Runforestrun Ma..
ET CNC Zeus Tracker Reported CnC ..
192.168.1.11 ID 203.130.196.6
ET COMPROMISED Known Compromised

1	1	1	10:54:38	ET WEB_CLIENT eval String.fromCharCode String Which May Be Malicious	2012173	6	0.013%
1	1	1	10:54:38	ET CURRENT_EVENTS Runforestrun Malware Campaign Infected Website Landing Page Obfuscated String JavaScript DGA	2014998	6	0.013%

alert tcp $EXTERNAL_NET $HTTP_PORTS -> $HOME_NET any (msg:"ET CURRENT_EVENTS Runforestrun Malware Campaign Infected Website Landing Page Obfuscated String JavaScript DGA"; flow:established,to_client; content:"*/window.eval(String.fromCharCode("; isdataat:80,relative; content:!")"; within:80; pcre:"/\x2A[a-z0-9]{10}\x2A\x2Fwindow\x2Eeval\x28String\x2EfromCharCode\x28[0-9]{1,3}\x2C[0-9]{1,3}\x2C/sm"; reference:url,blog.unmaskparasites.com/2012/06/22/runforestrun-and-pseudo-random-domains/; classtype:trojan-activity; sid:2014998; rev:2;)

file: **emerging-all.rules:51850**

categorize **0** event(s)

QUEUE	ACTIVITY	LAST EVENT	SOURCE	COUNTRY	DESTINATION	COUNTRY
1		2014-05-22 10:54:38	69.90.47.167	UNITED STATES (.us)	10.43.10.53	RFC1918 (.lo)

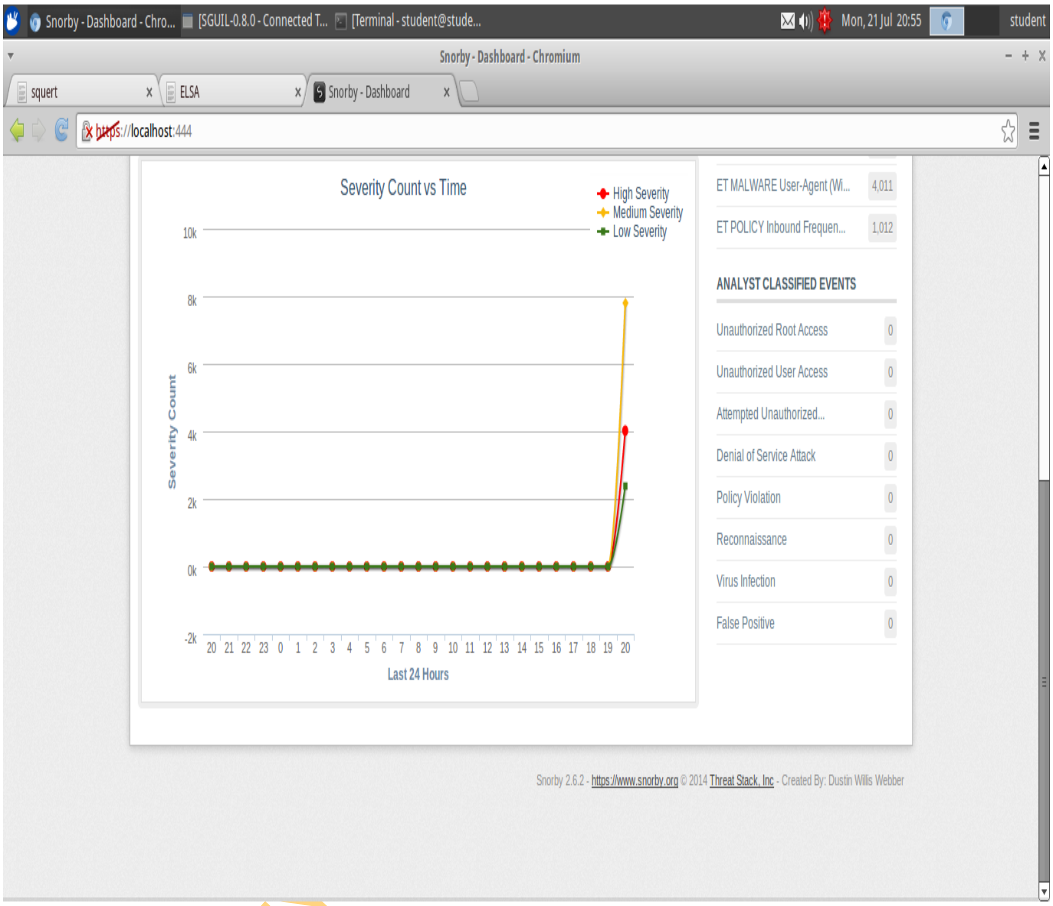
EXTERNAL LOOKUP (0.051 seconds 10 results)

```
May 22 10:54:41 10 bro_files: 1400766879.991974 FAnu883IqNes5swaIk 69.90.47.167 10.43.10.53 CaQSK239q34acJbvl6 HTTP 0 MD5,SHA1 image/x-icon
- 0.000002 F F 2550 2550 0 0 F - 9ceae7a3c88fc451d59e24d8d5f6f166 803f42d82efa0621cf0be039b933b800cacc73f4 - -

May 22 10:54:41 10 bro_files: 1400766879.887921 FAnu883IqNes5swaIk 69.90.47.167 10.43.10.53 CaQSK239q34acJbvl6 HTTP 0 MD5,SHA1 image/gif -
0.000456 F F 3207 3207 0 0 F - 07d9b365e13909fb63ae909cb9922fa4 5bdf50f96bd0ed07945be6758109e871293afac5 - -

May 22 10:54:41 10 bro_files: 1400766879.884596 FFTUSv0NvQ1mYy0la 69.90.47.167 10.43.10.53 CSUwcwlftdrB8PXBdq HTTP 0 MD5,SHA1 image/gif - 0
.000908 F F 3765 3765 0 0 F - 9810d1759b0c8dcd6444acd388ae53d6 0c84fe7f4aa14769e75659924b4e489269a7393d - -

May 22 10:54:41 10 bro_files: 1400766879.881881 FRT5aqIEpKnc82KU1 69.90.47.167 10.43.10.53 CDai0o3JVe2BHmKrq5 HTTP 0 MD5,SHA1 image/gif - 0
.000002 F F 2578 2578 0 0 F - a8dba74550b6221a5ce25fd2b590f511 c20efb8f0cf71faf7bd3db3786ef68e60ef6dc35 - -

May 22 10:54:41 10 bro_http: 1400766879.990771 CaQSK239q34acJbvl6 10.43.10.53 62009 69.90.47.167 80 0 - - - - - 0 2550 200 OK - - - (empty)
- - - - - FAnu883IqNes5swaIk image/x-icon

May 22 10:54:41 10 bro_http: 1400766879.887020 CaQSK239q34acJbvl6 10.43.10.53 62009 69.90.47.167 80 0 - - - - - 0 3207 200 OK - - - (empty)
- - - - - FAnu883IqNes5swaIk image/gif

May 22 10:54:41 10 bro_http: 1400766879.883242 CSUwcwlftdrB8PXBdq 10.43.10.53 62008 69.90.47.167 80 0 - - - - - 0 3765 200 OK - - - (empty)
- - - - - FFTUSv0NvQ1mYy0la image/gif

May 22 10:54:41 10 bro_http: 1400766878.363417 CDai0o3JVe2BHmKrq5 10.43.10.53 62006 69.90.47.167 80 1 GET www.kellysluggage.com /images/lug
gage.gif http://www.kellysluggage.com/ Mozilla/5.0 (compatible; MSIE 10.0; Windows NT 6.1; WOW64; Trident/6.0) 0 2578 200 OK - - - (empty)
- - - - - FRT5aqIEpKnc82KU1 image/gif

May 22 10:54:40 10 bro_files: 1400766878.876414 FhJdmylR62q0hqCWVe 69.90.47.167 10.43.10.53 CqKiSIR2Md101H8jh HTTP 0 MD5,SHA1 image/gif - 0
```

WELCOME **paulh** | LOGOUT

LOCAL **13:07:30** UTC **16:07:30**

Figure 3.46: ELSA Interface



ET MALWARE User-Agent (Wi...	4,011
ET POLICY Inbound Frequen...	1,012

ANALYST CLASSIFIED EVENTS

Unauthorized Root Access	0
Unauthorized User Access	0
Attempted Unauthorized...	0
Denial of Service Attack	0
Policy Violation	0
Reconnaissance	0
Virus Infection	0
False Positive	0

Snorby 2.6.2 - https://www.snorby.org © 2014 Threat Stack, Inc - Created By: Dustin Willis Webber

Figure 3.47: Snorby Interface

A short description of the intrusion detection systems and threat analysis tool is presented below:

a. *Snort:* Snort is currently the de-facto standard for open-source network-based intrusion-detection systems around the world (SourceFire, 2011). It is a light-weight signature based network intrusion detection system. Snort is is multi-threaded.

b. *Suricata**:*** Suricata is a network-based intrusion detection system that is still in early stages of development. It offers speed improvements and capabilities over snort when run on multicore operating system. Apart from the fact that it operate at both the application and packet level, it can also make use of Snort rules.

c. *Sguil* (pronounced sgweel) is built by network security analysts for network security analysts. Sguil's main component is an intuitive GUI that provides access to realtime events, session data, and raw packet captures.

d. *ELSA:* ELSA is a centralized syslog framework built on Syslog-NG, MySQL, and Sphinx full-text search. It provides a fully asynchronous web-based query interface that normalizes logs and makes searching billions of them for arbitrary strings as easy as searching the web. It also includes tools for assigning permissions for viewing the logs as well as email based alerts, scheduled queries, and graphing.

Its features include:

- High-volume receiving/indexing (a single node can receive > 30k logs/sec, sustained)
- Full Active Directory/LDAP integration for authentication, authorization, email settings
- Instant ad-hoc reports/graphs on arbitrary queries even on enormous data sets
- Dashboards using Google Visualizations
- Email alerting, scheduled reports and plugin architecture for web interface
- Distributed architecture for clusters
- Ships with normalization for some Cisco logs, Snort/Suricata, Bro, and Windows via Eventlog-to-Syslog or Snare

e. *Snorby:* Snorby is a web application that is used to query and view event data stored in a Sguil database (typically IDS alert data). Snorby is a visual tool that attempts to provide additional context to events through the use of metadata, time series representations and weighted and logically grouped result sets.

3.5 Experimental Design

Two Internet-facilitated Organised Crime Threats were used in this study. The Plymouth University Advanced Persistent Threats Testbed was developed from scratch and adapted to model and mitigate Minor Threats. In order to benchmark the framework, the popular DARPA 2000 LLDOS 1.0 Inside Threats developed in the year 1999 by MIT Lincoln Laboratory's research team was also adapted for the study. A four-network security management domain to serve as Collaborative Network Security Management System was designed to model and mitigate the Minor Threats.

3.5.1 Development of Plymouth University Advanced Persistent Threats and Collaborative Network Security Management for Modelling and Mitigation of Minor Threats

Four Attacking and four victim domains were involved in the experiment. The activities of the Network Security Manager were guided by a Central Administrative System operated by a Top-level Network Security Administrator over the campus network. The Network Threat Management Systems were evaluated in Xeon 5i Intel with 4Terabyte Hard disk and 8 GB RAM.

The attackers were set up over 10.1.0.0/27, 10.1.0.32/27, 10.1.0.64/27 and 10.1.0.96/27 subnets. The Attackers were given the following tasks to perform:

i. Connect to the Victims

ii. Scan the operating systems for exploitable vulnerability

iii. Attempt to exploit CVE-2012-4681

iv. Exploit CVE-2012-4681

v. Install Backdoors

The attacking experiment took two weeks. The first one week was used to collect the background traffics while the following week was used to conduct the attacking experiment.

The packets with their payloads were collected in each domain. Both the backgrounds and the attack tcpdumps were merged. Table 3.6 presents the descriptions of the Plymouth University Attack Phases. The details show that only tasks (i) to (iii) were successfully performed while tasks (iv) and (v) were not exploited. We therefore assume that the exploited threats were all Minor Threats.

A Collaborative Network Security Management System was set-up in the Networking Laboratory of Plymouth University, United Kingdom to manage threats using Snort and Suricata Emerging Threat Set-up in Security Onion Set. The Private Network had four subnets: 10.1.0.128/27, 10.1.0.160/27, 10.1.0.192/27 and 10.1.0.224/27 each managed by a Network Security Manager.

The next two weeks were used by the Central Administrator to get acquainted to the Security Managers. The two traffics were merged in each subnet. Figure 3.48 presents the diagram showing the set-up of the Plymouth University Experimental Set-up.

The traffics were replayed simultaneously three times each against the emerging Threat Rule Configuration of Snort and Suricata in each subnet. The Threat Analytic tools in Security Onion such as Sguil, Squert and Elsa were used for observing the alert events. Table 3.7 presents the time of each replay with the size of the packets. The replay lasted for average of 4minutes.

Table 3.6: Plymouth University Attack Phases

S/N	Source IP	Start Time	End Time	Attack Description
1	10.1.0.67	2014-07-08 13:23	2014-07-08 13:24	Backtrack NmapPing Scan against 10.1.0.130-254 (unsuccessful)
2	10.1.0.67	2014-07-08 13:24	2014-07-08 13:25	Backtrack Nmap Ping Scan against 10.1.0.0/24
3	10.1.0.3	2014-07-08 13:20	2014-07-08 13:24	Metasploit Scan against 10.1.0.130-10.1.0.254
4	10.1.0.99	2014-07-08 13:26	2014-07-08 13:36	Backtrack Nmap Intense Scan against 10.1.0.0/24
5	10.1.0.34	2014-07-08 13:38	2014-07-08 14:10	Nessus Vulnerability Scan against 10.1.0.130-10.1.0.254
6	10.1.0.3	2014-07-11 13:48	2014-07-11 13:48	Metasploit Exploit (Access the File System) against 10.1.0.135
7	10.1.0.3	2014-07-11 13:48	2014-07-11 13:48	Metasploit Exploit (Command Shell) against 10.1.0.135
8	10.1.0.3	2014-07-11 13:48	2014-07-11 13:50	Metasploit Exploit (Session Killed) against 10.1.0.135
9	10.1.0.3	2014-07-11 13:52	2014-07-11 13:52	Metasploit Exploit (Access the File System) against 10.1.0.166
10	10.1.0.3	2014-07-11 13:52	2014-07-11 13:52	Metasploit Exploit (Command Shell) against 10.1.0.166
11	10.1.0.3	2014-07-11 13:52	2014-07-11 13:53	Metasploit Exploit (Session Killed) against 10.1.0.166
12	10.1.0.3	2014-07-11 13:54	2014-07-11 13:55	Metasploit Exploit (Access the File System) against 10.1.0.197
13	10.1.0.3	2014-07-11 13:55	2014-07-11 13:55	Metasploit Exploit (Command Shell) against 10.1.0.197
14	10.1.0.3	2014-07-11 13:55	2014-07-11 13:56	Metasploit Exploit (Session Killed) against 10.1.0.197
15	10.1.0.3	2014-07-11 13:57	2014-07-11 13:57	Metasploit Exploit (Access Denied) against 10.1.0.194

Table 3.7: Plymouth University Packet Replay

Replay	Size of Packet	Date	Time	
			Snort	Suricata
Replay 1	201,307kb	21/07/2014	19:40:49- 19:43:10	19:40:09- 19:43:43
Replay 2	201, 307kb	21/07/2014	19:43:45- 19:47:10	19:44:29- 19:47:43
Replay 3	201,307kb	21/07/2014	19:47:12- 19:50:10	19:48:50- 19:54:43

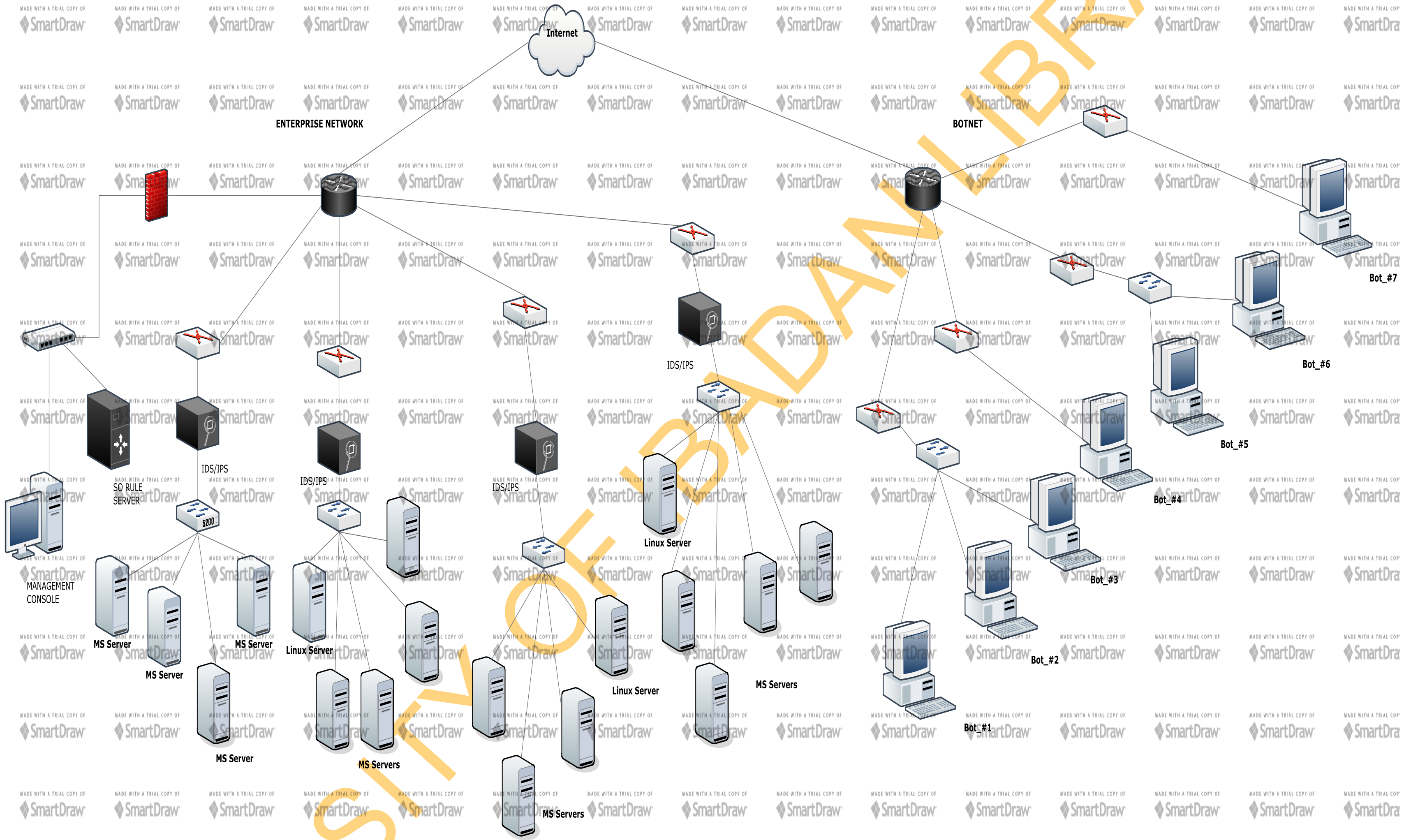


Figure 3.48: Plymouth University Experimental Set-up

3.5.2 Development of MIT Lincoln Lab (DARPA-sponsored) Botnet Threats and Collaborative Network Security Management System

In order to benchmark the model, the Botnet-based LLDOS 1.0 Inside exploits in four critical subnets: 172.16.112.0/24, 172.16.113.0/24, 172.16.114.0/24 and 172.16.115.0/24 created by MIT Lincoln Lab in 2000 (DARPA, 2014) were filtered and merged with DARPA 1999 background data collected on Monday of the first week as reported in same DARPA (2014). The two tcpdump files with their payload were replayed against Suricata and Snort Network Intrusion Detection and Prevention Systems thrice. Emerging Threat Rulesets were configured for both Snort and Suricata NIDS. The same Collaborative Network Security Management System set up for Plymouth University Advanced Persistent Threat was also used. Figure 3.49 presents the diagram showing the original set-up of the MIT Lincoln Lab Experiment.

The premise of the attack is that a relatively novice adversary seeks to show his/her prowess by using a scripted attack to break into a variety of hosts around the internet, install the components necessary to run a Distributed Denial of Service, and then launch a DDOS at a US government site. As a part of the attack the adversary used the Solaris sadmind exploit, a well-known Remote-To-Root attack to successfully gain root access to three Solaris hosts at Eyrie Air Force Base. These attacks succeeded due to the relatively poor security model applied at the AFB, many services, including the dangerous "sunrpc" service, were proxied through the base's firewall from outside to inside. The attacker is using the Mstream DDOS tool, one of the less sophisticated DDOS tools. It did not make use of encryption and does not offer as wide a range of attack options as other tools, such as TribeFloodNetwork or Trinoo. An Mstream "server", the software that actually generates and sends the DDOS attack packets, was installed on each of the three victim hosts, while an Mstream "master", the software that provides a user-interface and controls the "servers" was installed on one of the victims.

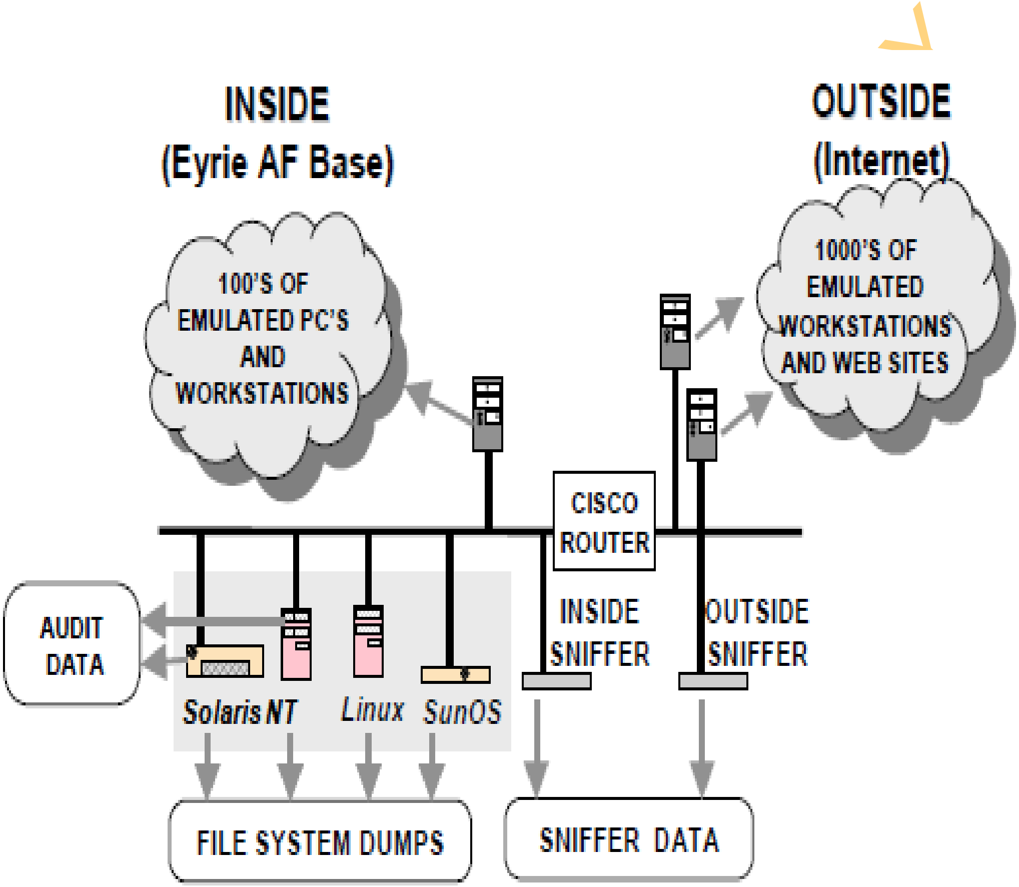


Figure 3.49: MIT Lincoln Lab Experimental Testbed (Haines et al., 2001)

The five phases of the attack scenario are:

i. IPsweep of the AFB from a remote site
ii. Probe of live IP's to look for the sadmind daemon running on Solaris hosts
iii. Breakins via the sadmind vulnerability, both successful and unsuccessful on those hosts
iv. Installation of the trojan mstream DDoS software on three hosts at the AFB
v. Launching the DDoS

Table 3.8 presents the time of each replay with the size of the packets. The replay lasted for average of 4minutes. In the experiment, we focused on the first three phases (i) to (iii), which are largely Minor Threats. In Figure 3.50, the layout of the Collaborative Network Security Management System is presented. The components are discussed thus:

a. Network Domain: Each security manager operates in a network domain that contains networking devices, information and operating system assets and network security devices all identified as servers. The sniffing server, which is tcpdump monitors and collects the packets. It sends the packets with their payloads to the unified threat management system known as Security Onion.
b. Unified Threat Management System: Security Onion is employed to mitigate threat because it is open source, reliable and contains network threat management tools that interoperate together. The security onions virtual machines in the different domains are remotely connected together through bridging.
c. Central Administrative System: This operates as an enterprise server system in the Collaborative Network Security Management System. It performs the modelling for event analysis. The modelling tools are THREAT-PREDICT and COSEM-TR.
d. Mailing System: Different e-mailing system are used to share events and threats among network security managers and central administrator

Table 3.8: DARPA Packet Replay

Replay	**Size of Packet**	**Date**	**Time**	
			Snort	**Suricata**
Replay 1	452,256kb	20/07/2014	14:03:17-14:10:56	13:30:42-13:44:17
Replay 2	452,256kb	20/07/2014	14:10:57-14:18:27	13:44:22-13:54:17
Replay 3	452,256kb	20/07/2014	14:18:28-14:21:13	14:30:42-14:44:17

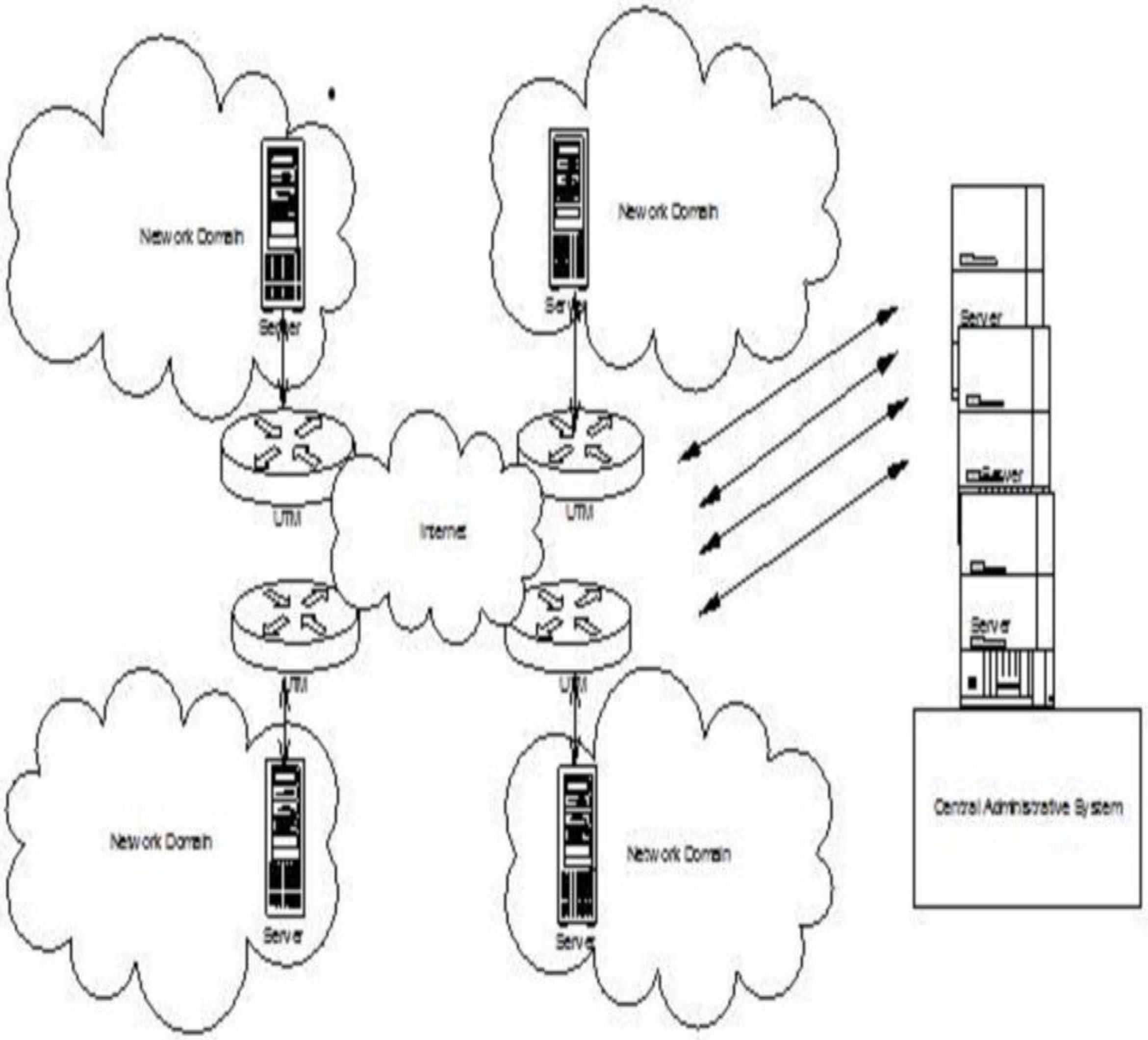


Figure 3.50: Layout of Collaborative Network Security Management System

3.6 Performance Evaluation Metrics

The following performance evaluation metrics are used in comparing the performance of the models:

a. Predictability

This is the probability of having consequence attack given the antecedent attacks. The confidence of the sequential association rules is equated to the predictability in this regards.

Predictability <=> Confidence

$$\text{Confidence} = \frac{\text{Number of Consequence}}{\text{Number of Corresponding Antecedent}} \quad \ldots \quad (3.10)$$

b. Spearman's Rank Correlation Coefficient

Spearman's Rank Correlation Coefficient (r): This is used to evaluate the correlation between the original attack scenario and the priority of the threat.

The Simplified Spearman's Rank Correlation Coefficient formula is given as:

$$r = 1 - \left(6\Sigma d^2 / n(n^2-1)\right) \quad \ldots \quad (3.11)$$

If there were ties in any of previous steps, use the standard Spearman's Rank Correlation Coefficient formular instead:

$$\rho = \frac{\sum_i (x_i - \bar{x})(y_i - \bar{y})}{\sqrt{\sum_i (x_i - \bar{x})^2 \sum_i (y_i - \bar{y})^2}} \quad \ldots \quad (3.12)$$

The interpretation is that it can vary between -1 and 1.

- Close to -1 - Negative correlation.
- Close to 0 - No linear correlation.
- Close to 1 - Positive correlation.

c. *Threat Rating*

This is the estimated value of risk for a specific threat. It is represented quantitatively as integer value and qualitatively represented as either high or low.

d. *False Positive Rate*

This refers to the percentage of number of non-intrusion events detected by NIDS.

$$FPR = (FP/\ FP + TN) \times 100 \qquad \ldots \qquad (3.13)$$

e. *Cost of Detection*

This is expressed as the number of signature rules in the NIDS configuration ruleset.

f. *Time of Detection*

This refers to the time expended in detecting Threats. It is measured in *minutes*.

CHAPTER FOUR

RESULTS AND DISCUSSION

This chapter presents the results of evaluation of Threat Prediction, Threat Prioritisation and Threat Mitigation Models. It also presents the comparison of the models with existing models. This is followed by the discussion of results.

4.1 Results

4.1.1 Results of Threat Prediction

The results of Threat Prediction Model for both Plymouth University Advanced Persistent Threat and DARPA-sponsored MIT Lincoln Lab Network Threat Management are presented.

4.1.1.1 Results of Threat Prediction for Plymouth University Advanced Persistent Threats

The Plymouth University events reported by the Threat Analysis tools presented in Appendix 3 after its pre-processing were processed by the THREAT-PREDICT tool. Based on the assumption that a once successful attack exploit would be exploited by an attacker in the near future than a none successful one; only the attack sequence with full support (sequence that occur three times) are chosen to determine the actionable threat paths. Table 4.1 presents the Actionable Threat Path generated by the Plymouth University Threat Prediction Experiment. Figure 4.1 presents the Plymouth University attack graphs generated by the Threat Prediction Model. However, no attack graph was generated for the Plymouth University Threats based on Li *et al.*(2007) because none of the threats predicted met the minimum support requirements.

4.1.1.2 Results of Threat Prediction for MIT Lincoln Lab (DARPA-sponsored) Air Force-based Botnet LLDOS 1.0 Threats

The MIT Lincoln events reported by the Threat Analysis tools presented in Appendix 4 after its pre-processing were processed by the THREAT-PREDICT tool.

Based on the same assumption that a once successful attack exploit would be exploited by an attacker in the near future than none successful one; only the attack sequence with full support (sequence that occur three times) are chosen to determine the actionable threat paths. Table 4.2 presents the Actionable Threat Path generated by the MIT Threat Prediction Experiment. Figure 4.2 presents the attack graphs for MIT Lincoln LLLDOS 1.0 generated by the Threat Prediction Model. To compare our model performance, we present attack graphs results of Li *et al.* (2007) in Figure 4.3 and Figure 4.4. Figure 4.5 presents the line graph for the comparison. Appendix 5 and Appendix 6 presents the sequences generated in steps for Plymouth University and MIT Lincoln events.

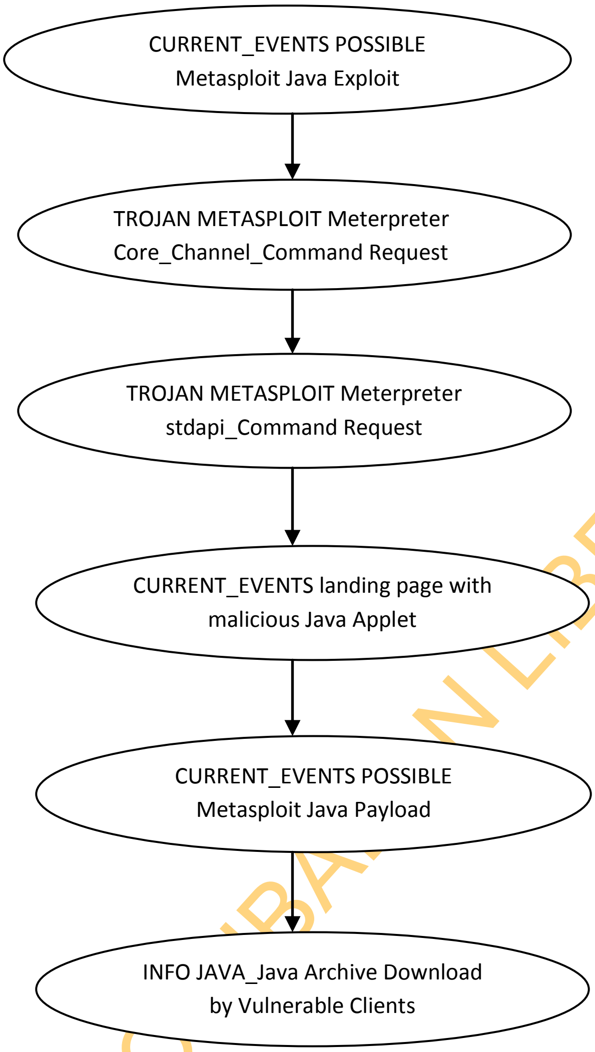

CURRENT_EVENTS POSSIBLE Metasploit Java Exploit
TROJAN METASPLOIT Meterpreter Core_Channel_Command Request
TROJAN METASPLOIT Meterpreter stdapi_Command Request
CURRENT_EVENTS landing page with malicious Java Applet
CURRENT_EVENTS POSSIBLE Metasploit Java Payload
INFO JAVA_Java Archive Download by Vulnerable Clients

Figure 4.1: Plymouth University attack graphs generated by the Threat Prediction Model

Table 4.1: Actionable Threat Paths generated by the Threat Prediction Experiment for Plymouth University Attack Scenario

S/N	Attack Scenario	Exploit	Source	Destination	Frequency/Support	Confidence/ Predictability
1	D2,4	CURRENT_EVENTS Possible Metasploit Java Exploit	10.1.0.3	10.1.0.135	3 times /0.02654867	1 / 100%
2	D2,4=>AN2,11	Trojan Metasploit Meterpreter core_channel Command Request	10.1.0.3	10.1.0.197	3 times /0.02654867	1 / 100%
3	D2,4, AN2,11 => AO2,4	Trojan Metasploit Meterpreter stdapi_Command Request	10.1.0.3	10.1.0.135	3 times /0.02654867	1 / 100%
4	D2,4, AN2,11, AO2,4 => C2,4	CURRENT_EVENTS landing page with malicious Java Applet	10.1.0.3	10.1.0.135	3 times /0.02654867	1 / 100%
5	D2,4, AN2,11, AO2,4, C2,4=> E2,4	CURRENT_EVENTS Possible Metasploit Java Payload	10.1.0.3	10.1.0.135	3 times /0.02654867	1 / 100%
6	D2,4, AN2,11, AO2,4, C2,4, E2,4=> K2,4	INFO JAVA-Java Archive Download by Vulnerable Client	10.1.0.3	10.1.0.135	3 times /0.02654867	1 / 100%

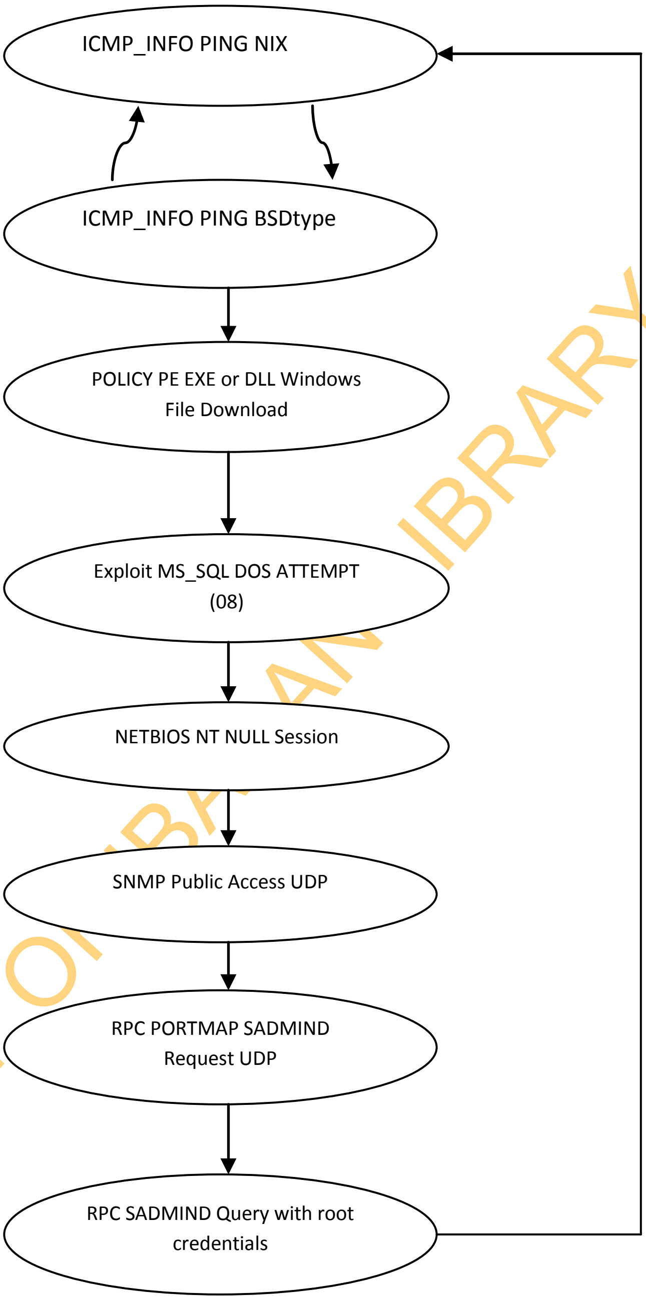

ICMP_INFO PING NIX
ICMP_INFO PING BSDtype
POLICY PE EXE or DLL Windows File Download
Exploit MS_SQL DOS ATTEMPT (08)
NETBIOS NT NULL Session
SNMP Public Access UDP
RPC PORTMAP SADMIND Request UDP
RPC SADMIND Query with root credentials

Figure 4.2: Attack Graphs for the MIT Lincoln LLDOS 1.0 Attack Generated by the Threat Prediction Model

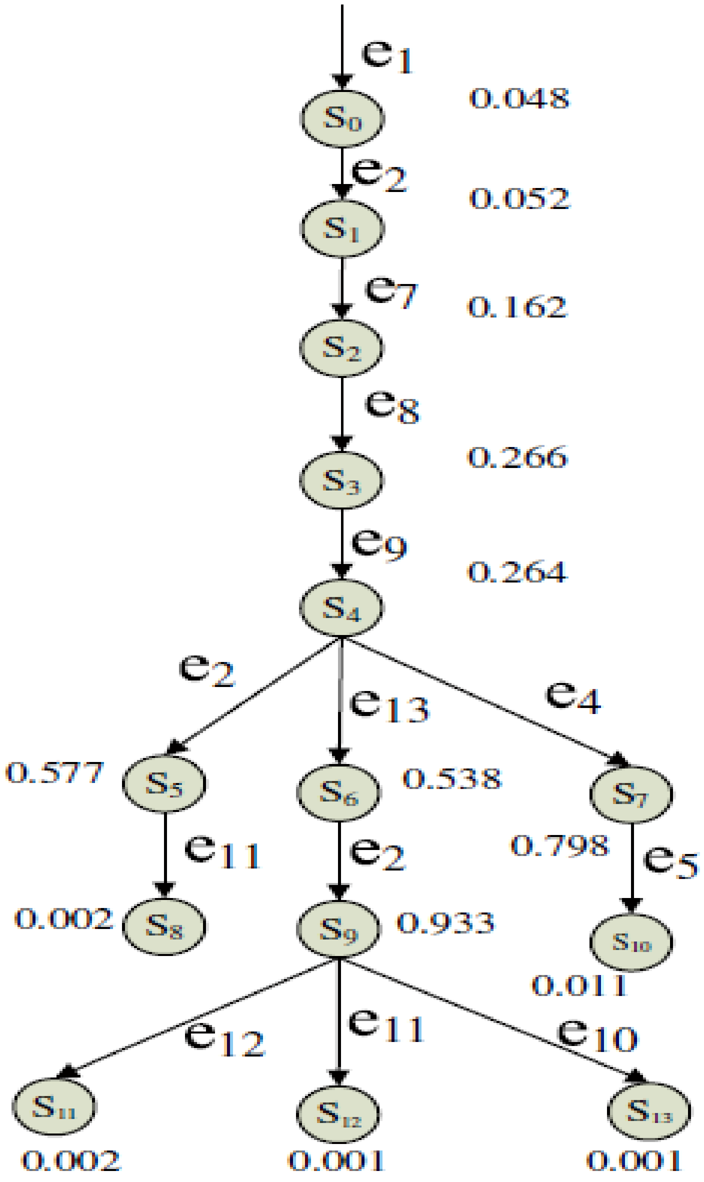


Figure 4.3: Attack Graph with Predictability Values (Li *et al.*, 2007)

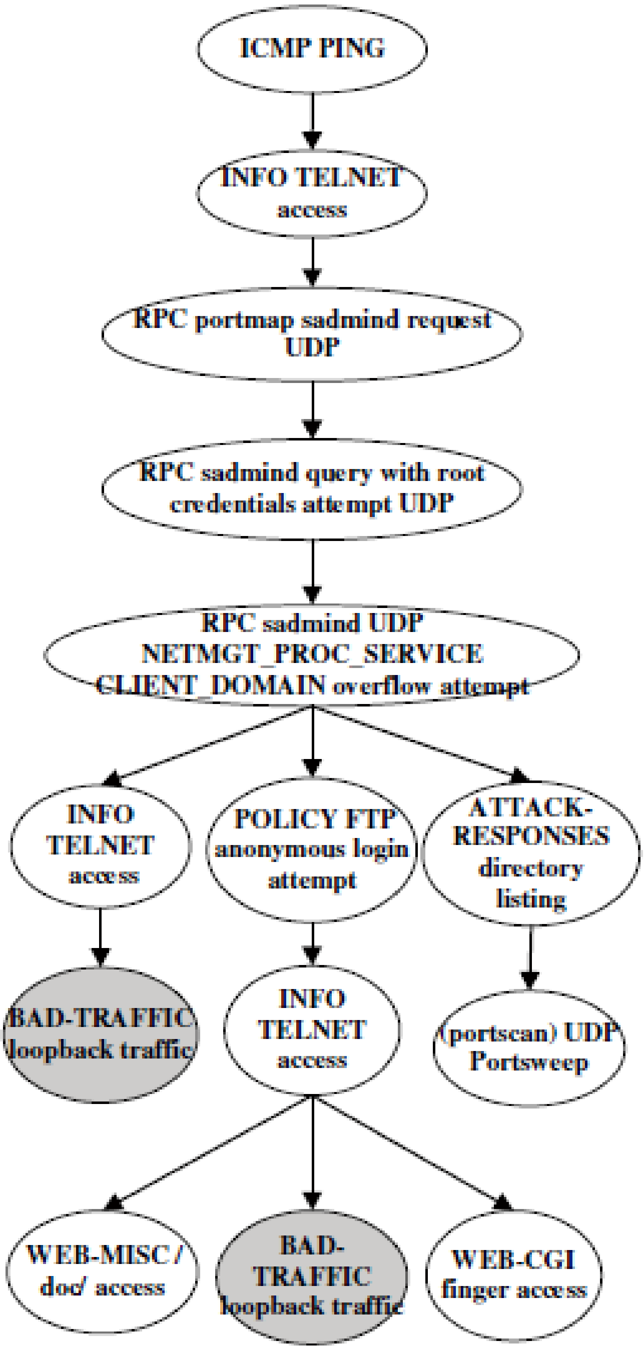

ICMP PING
INFO TELNET access
RPC portmap sadmind request UDP
RPC sadmind query with root credentials attempt UDP
RPC sadmind UDP NETMGT_PROC_SERVICE CLIENT_DOMAIN overflow attempt
INFO TELNET access
POLICY FTP anonymous login attempt
ATTACK-RESPONSES directory listing
BAD-TRAFFIC loopback traffic
INFO TELNET access
(portscan) UDP Portsweep
WEB-MISC / doc/ access
BAD-TRAFFIC loopback traffic
WEB-CGI finger access

Figure 4.4: Exploit Oriented Graph (Li *et al.*, 2007)

Table 4.2: Actionable Threat Paths generated by the Threat Prediction Experiment for MIT Lincoln LLDOS 1.0

S/N	Attack Scenario	Exploit	Source	Destination	Frequency/ Support	Confidence / Predictability
1	C12,41	INFO PING NIX	172.16.113.50	172.16.113.105	3 times /0.021897	1 / 100%
2	C12, 41 =>D12,41	INFO PING BSDtype	172.16.113.50	172.16.113.105	3 times /0.0218979	1 / 100%
3	C12,41, D12,41 => C10,70	INFO PING NIX	172.16.112.50	172.16.114.169	3 times /0.021897	1 / 100%
4	C12,41, D12,41 C10,70 => D10,70	INFO PING BSDtype	172.16.112.50	172.16.114.169	3 times /0.021897	1 / 100%
5	C12,41, D12,41 C10,70, D10,70 => M21,65	POLICY PE EXE/DLL Windows File Download	132.60.168.152	172.16.112.207	3 times /0.021897	1 / 100%
6	C12,41, D12,41 C10,70, D10,70, M21,65 => A13,14	Exploit MS_SQL DOS ATTEMPT(08)	172.16.115.20	172.16.112.20	3 times /0.021897	1 / 100%
7	C12,41, D12,41 C10,70, D10,70, M21,65, A13,14 => F25,31	NETBIOS NT NULL Session	172.16.116.20	172.16.112.100	3 times /0.021897	1 / 100%
8	C12,41, D12,41 C10,70, D10,70, M21,65, A13,14	NETBIOS NT NULL Session	172.16.112.100	172.16.112.100	3 times /0.021897	1 / 100%

	=> F9,14					
9	C12,41, D12,41 C10,70, D10,70, M21,65, A13,14, F9,14 => K13,35	SNMP Public Access UDP	172.16.113.20	172.16.112.105	3 times /0.021897	1 / 100%
10	C12,41, D12,41 C10,70, D10,70, M21,65, A13,14, F9,14, K13,35 => I20,62	RPC PORTMAP SADMIND REQUEST UDP	202.77.162.213	172.16.115.20	3 times /0.021897	1 / 100%
11	C12,41, D12,41 C10,70, D10,70, M21,65, A13,14, F9,14 K13,35, I20,62 => J20,62	RPC Sadmind query with root credentials	202.77.162.213	172.16.115.20	3 times /0.021897	1 / 100%
12	C12,41, D12,41 C10,70, D10,70, M21,65, A13,14, F9,14 K13,35, I20,62, J20,62 => C13,60	ICMP PING NIX	172.16.115.20	172.16.113.204	3 times /0.021897	1 / 100%

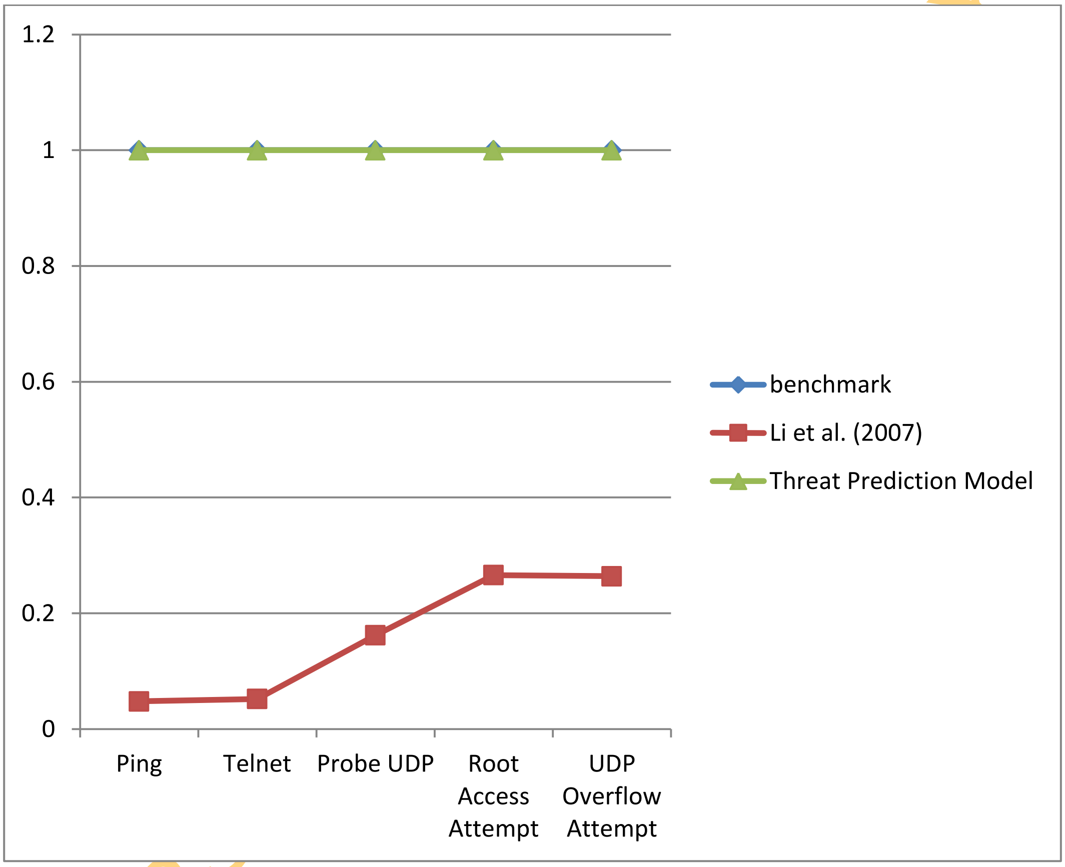


Figure 4.5: Line Graph showing the Performance of Threat Prediction Models based on MIT Lincoln Lab Threats

4.1.2 Results of Threat Prioritisation

Table 4.3 and Table 4.4 present the result of Threat Rating and Threat Ranking for Plymouth University and MIT Lincoln Lab Threat Scenarios, with the assumption that the Threat Rating score that is less than 5 belongs to 'very low' rank while the Threat Rating from 5 and above belong to 'low' rank.

IF Threat Rating>= 5 THEN Threat Rank = Low; ELSE Threat Rank = Very Low

4.1.2.1 Results of Comparison of the Threat Prioritisation Model with CVSS version 2 and Snort Severity

In order to benchmark the Threat Prioritisation Model, the outcome of the model and other models are compared. CVSS v2 and Snort Priority are chosen because of their popularity and standardization. The comparison of their performance in Prioritizing Plymouth University and MIT Threats are presented in Table 4.5 and Table 4.6 respectively. Table 4.7 and Table 4.8 are used to present the Spearman's Correlation for the two threat scenarios.

Table: 4.3: Population of Events, Threat Rating and Threat Ranking for Plymouth University Threats

S/N	Threat	Number of Event detected by Snort	Number of Event detected by Suricata	Threat Rating Score	Threat Ranking value
1	CURRENT_EVENTS Possible Metasploit Java Exploit	96	70	6.5	Low
2	Trojan Metasploit Meterpreter core_channel Command Request	1	1	4.0468	Very Low
3	Trojan Metasploit Meterpreter stdapi_Command Request	64	80	6.0	Low
4	CURRENT_EVENTS landing page with malicious Java Applet	14	14	5.0	Low
5	CURRENT_EVENTS Possible Metasploit Java Payload	90	64	5.5	Low
6	INFO JAVA-Java Archive Download by Vulnerable Client	60	39	5.5	Low

Table: 4.4: Population of Events, Threat Rating and Ranking for MIT Lincoln Lab Threat

S/N	Threat	Number of Events detected by Snort	Number of Events detected by Suricata	Threat Rating Score	Threat Ranking Value
1	ICMP INFO PING NIX	0	3	1.75	Very Low
2	ICMP INFO PING BSDtype	0	3	1.75	Very Low
3	ICMP INFO PING NIX	0	3	1.75	Very Low
4	INFO PING BSDtype	0	3	1.75	Very Low
5	POLICY PE EXE/DLL Windows File Download	0	3	2.25	Very Low
6	Exploit MS_SQL DOS ATTEMPT(08)	1	0	9.8333	Low
7	NETBIOS NT NULL Session	7	5	4.05556	Very Low
8	NETBIOS NT NULL Session	0	3	11.16667	Low
9	SNMP Public Access UDP	0	3	5.41667	Low
10	RPC PORTMAP SADMIND REQUEST UDP	6	3	13.0	Low
11	RPC Sadmind	6	3	11.33333	Low

	query with root credentials				
12	ICMP PING NIX	0	3	3.5	Very Low

Table 4.5: Comparison of Performance of Threat Prioritisation Model, CVSSv2 and Snort for Plymouth University Threats

S/N	Threat	CVE_ID	Threat Rating/Ranking	CVSSV2	Snort Priority
1	CURRENT_EVENTS Possible Metasploit Java Exploit	-	6.5 / Low	-	2
2	Trojan Metasploit Meterpreter core_channel Command Request	-	4.0468 / Very Low	-	2
3	Trojan Metasploit Meterpreter stdapi_Command Request	-	6.0 / Low	-	2
4	CURRENT_EVENTS landing page with malicious Java Applet	-	5.0 / Low	-	2
5	CURRENT_EVENTS Possible Metasploit Java Payload	-	5.5 / Low	-	2
6	INFO JAVA-Java Archive Download by Vulnerable Client		5.5/ Low	-	2

Table 4.6: Comparison of Performance of Threat Prioritisation Model, CVSSv2 and Snort for MIT Lincoln Lab Threats

S/N	Threat	CVE	Threat Rating/Ranking	CVSSV2	Snort Priority
1	INFO PING NIX	-	1.75/ Very Low	-	3
2	INFO PING BSDtype	-	1.75/ Very Low	-	3
3	INFO PING NIX	-	1.75/ Very Low	-	3
4	INFO PING BSDtype	-	1.75/ Very Low	-	3
5	POLICY PE EXE/DLL Windows File Download	-	1.75/ Very Low	-	2
6	Exploit MS_SQL DOS ATTEMPT(08)	CVE:2002-0649	9.8333 / Low	8	1
7	NETBIOS NT	CVE:2000-0347	4.05556 / Very	10	2

	NULL Session		Low		
8	NETBIOS NT NULL Session	CVE:2000-0347	11.16667 / Low	10	2
9	SNMP Public Access UDP	CVE:2002-0013	5.41667 / Low	10	2
10	RPC PORTMAP SADMIND REQUEST UDP	CVE:2003-0722	13.0 / Low	10	2
11	RPC SADMIND Query with root credentials	-	11.33333 / Low	10	2
12	ICMP PING NIX	-	3.5 / Very Low	-	3

Table 4.7: Spearman's Rank Correlation Coefficient and Significance for Plymouth University Threat Prioritisation

Spearman's Correlation Metrics	**Threat Prioritisation Model**
Spearman's Correlation Value	0.6790
Spearman's Correlation Significance	Positive Significance

Table 4.8: Spearman’s Rank Correlation Coefficient and Significance for MIT Lincoln Lab Threat Prioritisation

Spearman’s CorrelationMetrics	**Threat Prioritisation Model**
Spearman’s Correlation Value	0.5857
Spearman’s Correlation Significance	Positive Significance

4.1.3 Results of Threat Mitigation

The Threat Mitigation Model is implemented based on the result of the Threat Prioritisation Model. Specifically, the Threat Ranking values presented in Table 4.5 and Table 4.6 are used for managing the Minor Threats. In this experiment, the Emerging Threat rulesets of low ranked threats are configured instead of the entire rulesets update configured at the initial stage. This decision is taken in order to manage the network security within the constraint of resources available.

Hence, Table 4.9 and Table 4.10 present the resources requirements and the head (time and cost) incurred in managing both the Plymouth University Advanced Persistent Threat and MIT Lincoln Lab LLDOS 1.0 Botnet Threat.

In Appendix 7 and 8, the Security Onion Event Reports based on Security Configurations for Snort and Suricata in the case of Plymouth University Advanced Persistent Threat and MIT Lincoln Lab LLDOS 1.0 Botnet Threats are presented respectively. Based on the reports, the following results in Table 4.11 and Table 4.12 are presented to show the Network Security Management expenses and quality after the application of Threat modelling for both Plymouth University Threat and MIT Lincoln Lab LLDOS 1.0 Minor Threat. Also, Figure 4.6 and Figure 4.7 present the bar charts of the false positive rate for Plymouth University Advanced Persistent Threats and MIT Lincoln Lab LLDOS 1.0.

Table 4.9: Resource Requirements (Scope) for Plymouth University APT Threat Mitigation

Parameters	**Snort**	**Suricata**
Total Number of Signature Rules Required	15	15
Total Detection Time Required	5mins (300s)	5mins (300s)
Number of Signatures Rules that match Major Threats	9	10
Detection Time for Major Threat	4mins (240s)	4mins (240s)

Table 4.10: Resource Requirements (Scope) for MIT Lincoln Lab Threat Mitigation

Parameters	**Snort**	**Suricata**
Total Number of Signature Rules Required	15	15
Total Detection Time Required	5mins (300s)	5mins (300s)
Number of Signatures Rules that match Major Threats	10	10
Detection Time for Major Threat	3.5mins (210s)	3.5mins(210s)

Table 4.11: Expenses and Quality of Mitigation Before and After the Application of Threat Modelling for Plymouth University Threats

Metric	**Experimental Phase**	**Snort**	**Suricata**
Size of Signature Rules	Size of Signature Rules for Minor Threat (Before Threat Modelling and Mitigation)	18701	19082
	Size of Signature Rules for Minor Threat(After Threat Modelling and Mitigation)	5	5
	Total Size of Signature Rules for Network Threat Management	14	15
Detection Time	Detection Time for Minor Threat (Before Threat Modelling and Mitigation)	4mins (240s)	4mins (240s)
	Detection Time for Minor Threat (After Threat Modelling and Mitigation)	0.016669mins (1s)	0.01666mins (1s)
	Total Detection Time Capacity for Network Threat Management	4.01666mins (241s)	4.01666mins (241s)
False Positive Rate	False Positive Rate (Before Threat Modelling and Mitigation)	99.15%	99.81%
	False Positive Rate (After Threat Modelling and Mitigation)	2.78%	1.28%

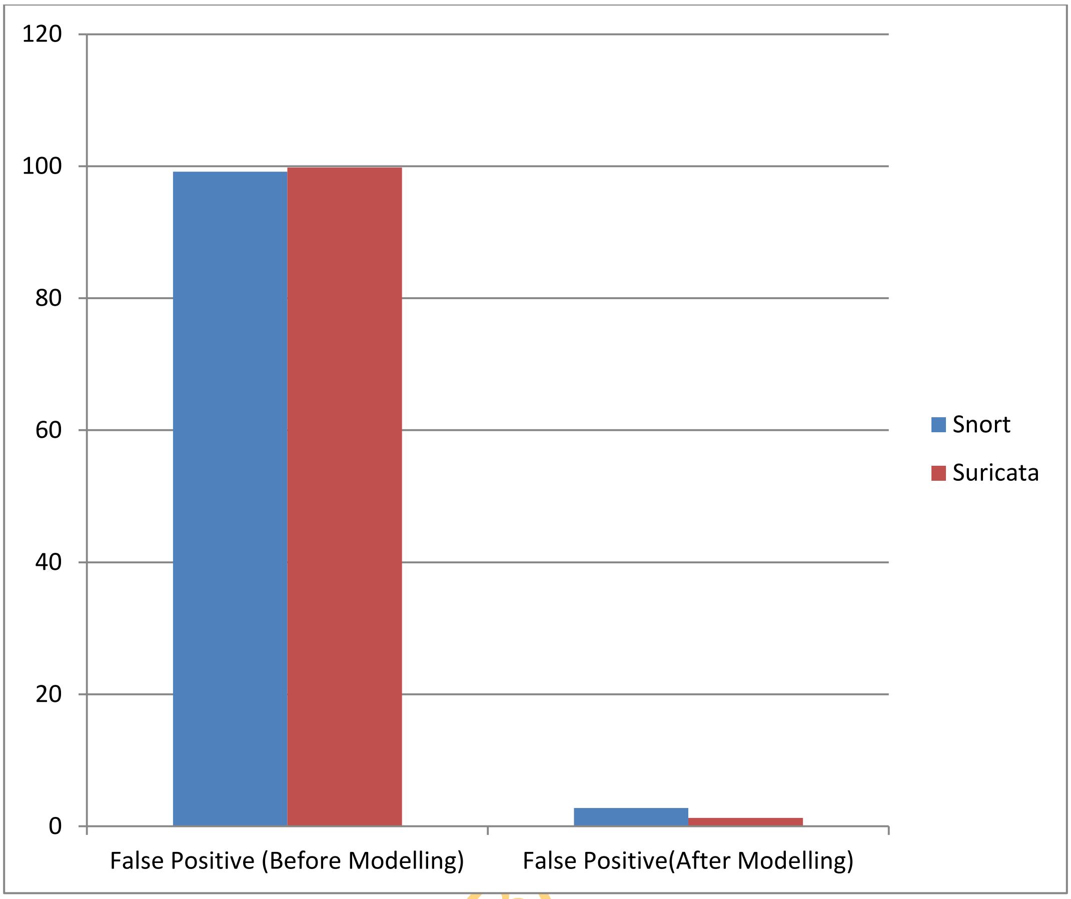


Figure 4.6: Comparison of False Positive Rates Before and After Threat Mitigation for Plymouth University Threats

Table 4.12: Expenses and Quality of Mitigation Before and After the Application of Threat Modelling for MIT Lincoln Lab Threats

Metric	**Experimental Phase**	**Snort**	**Suricata**
Size of Signature Rules	Size of Signature Rules for Minor Threat (Before Threat Modelling and Mitigation)	18701	19, 082
	Size of Signature Rules for Minor Threat(After Threat Modelling and Mitigation)	5	5
	Total Size of Signature Rules for Network Threat Management	15	15
Detection Time	Detection Time for Minor Threat (Before Threat Modelling and Mitigation)	8mins (480s)	8mins (480s)
	Detection Time for Minor Threat (After Threat Modelling and Mitigation)	0.05mins (3s)	1.25mins (75s)
	Total Detection Time Capacity for Network Threat Management	3.55mins (213s)	4.75mins (285s)
False Positive Rate Reduction	False Positive Rate (Before Threat Modelling and Mitigation)	99.9%	99.04%
	False Positive Rate (After Threat Modelling and Mitigation)	21.16%	1.38%

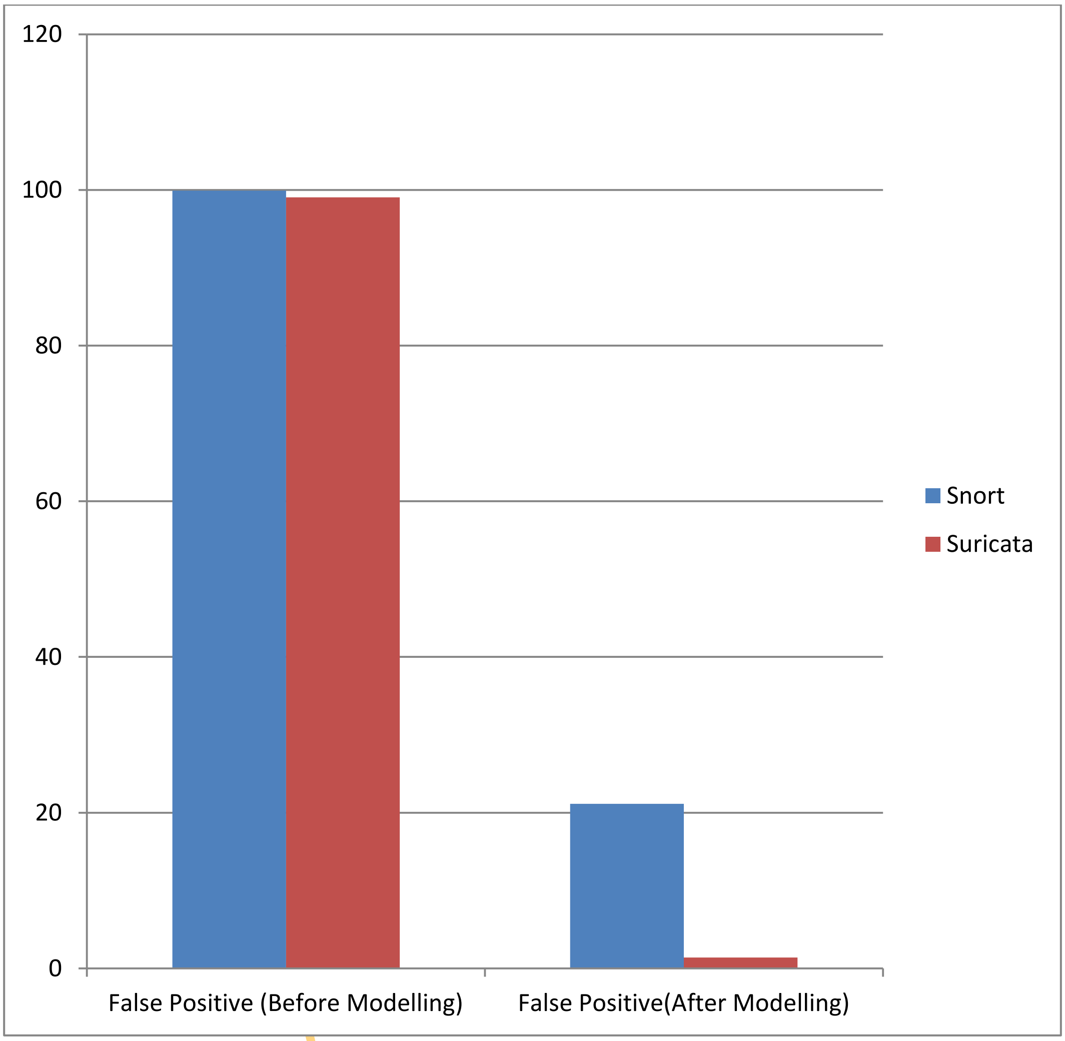


Figure 4.7: Comparison of False Positive Rates Before and After Threat Mitigation for MIT Lincoln Lab Threats

4.2 Discussion of Results

This section discusses the results of Threat prediction and Threat Prioritisation; and compares their performance with the existing tools. It also discusses the Threat Mitigation by comparing the results before the application of Threat Modelling and after the application of Threat Modelling.

4.2.1 Discussion of Threat Prediction

In Table 4.1, five sequences of events of 6 steps with the support of 0.02654867 and confidence of 1 are selected after the interestingness analysis of the Plymouth University Events by the Central Administrator. Each of the sequence steps occurs three times meaning that the attackers prefer to use the exploit because it always lead to success since an attacker will adhere to the strategy that will give him/her maximum benefit. This conforms to the earlier study that a novice attacker exploits easy-to-use kit (Bhattacharya and Ghosh, 2008). Figure 4.1 presents the actionable Threat Path derived from the sequences. In Table 4.2, 11 sequences of 12 steps with the support of 0.021897 and Confidence of 1 are selected after the interestingness analysis of the MIT Lincoln LLDOS 1.0 by the Central Administrator. The Central Administrator selects only the sequences that occur three times and prunes the sequences in order to obtain the longest actionable Threat Paths. This path is presented as Attack Graph in Figure 4.2.

The comparison of the attack graph with the original attack description shows that the Threat Paths reflect to a large extent the attack steps. Different bots were applied at the reconnaissance IPsweep and scanning phases as shown in step 1 and Step 2. The Attack Graph shows that after a successful exploit of sadmind vulnerability in a host 172. 16.115.20 in a particular subnet, the attacker pings host 172.16.113.204 in another subnet. This conforms to the description in DARPA (2014). The comparison of the Threat Prediction result with previous Sequential Association Mining Technique by Li *et al.* (2007) in Figure 4.3 and Figure 4.4 shows that the new approach is better than the latter. In fact, none of the attack steps in Plymouth University APT scenario could be predicted using previous Sequential Association Mining Technique by Li *et al.* (2007). In MIT Lincoln LLDOS 1.0 analysis, Li *et al.* (2007) did not show the loop but rather showed sequential attack paths, which did not reflect well how a hacker works in real settings. Also, the Threat Prediction model recorded a very good performance with the predictability of 1 for all sequence while that of Li *et al*. (2007) recorded the highest of 0.266 as shown in Figure 4.5.

4.2.2 Discussion of Threat Prioritisation

Table 4.3 presents the results of the rating and ranking of the predicted Minor Threats for Plymouth University Attack Scenario. The result shows that the population of event detected is fairly proportional to the Threat Rating score and Threat Ranking values. Table 4.4 also shows that proportionate relationship. This conforms to the general fact in computation that the memory loads affect the performance of instruction processing, hence the higher the population of events reported, the higher the demands of computation and the higher the cost and time of processing. In Table 4.3, five threats have the Threat Rating scores that are greater or equal to 5 while 1 threat has rating that is below 5. In Table 4.4, five threats have the Threat Rating scores that are greater or equal to 5 while 7 threat are below 5. All the 5 threats in the two tables are ranked low while the remaining threats are ranked very low.

Table 4.5 and Table 4.6 present the comparison of the performance of the Threat Prioritisation model, CVSSv2 and Snort for Plymouth University Threats and MIT Lincoln Lab Threats respectively. In Table 4.5, none of the threats has Common Vulnerability and Exposure Identification (CVE_ID). This is the reason none of the threats has CVSSv2 score. However, Snort classifies the Threats into group 2 i.e low priority threat. This Prioritisation by Snort does not reflect the original Attack Scenario presented in Chapter 3. The outcome of the Threat Prioritisation model is correlated with the original scenario using the Spearman's rank correlation coefficient in Table 4.7 to analyse the performance of the model. A correlation coefficient of 0.6790 was estimated showing that the correlation is positively significant.

In MIT Lincoln LLDOS 1.0 Threat Prioritisation comparison presented in Table 4.6, five threats have CVE_ID with CVSS in high rank category; the minimum CVSS score was 8. Snort Priority also grouped the threats into three priority groups: 1, 2, 3. Our Threat Prioritisation Model groups them into two groups with various Threat Rating scores. The observation of the outcome shows that CVSSv2 is not appropriate for prioritizing threats because only five of the threats are prioritized. The Snort Priority scores on the other hand do not represent the attack scenario. In fact, it cannot be applied in the emerging threat world where exploit capability continually changes. The outcome of our Threat Prioritisation model is correlated with the original scenario using the Spearman's rank correlation coefficient in Table 4.8 to prove the reputation of the model. A correlation coefficient of 0.5857 was estimated showing that the correlation is positively significant.

4.2.3 Discussion of Threat Mitigation

In this section, the scope or requirements of Network Threat Management is set in the Collaborative Network Security Management System. The Collaborative Network Security Management requirements for Plymouth University and MIT Lincoln Lab Threat Management are presented in Table 4.9 and Table 4.10 respectively.

From Table 4.9, 15 signature rules are required to be enabled in each of Snort and Suricata NIDS while the detection must not exceed 5 minutes. With the number of signature rules for the Major Threats already 9 and 10 respectively for Snort and Suricata, a maximum of 5 signature rules updates can only be accommodated for the Minor Threat. In the same vein, Table 4.10 shows that 15 signature rules are required to be enabled in each of Snort and Suricata NIDS while the detection time must not exceed 5 minutes. Already, 10 signature rules are enabled for the Major Threats. Hence, only 5 new updates of signature rules can be enabled or accomodated. Since, this work is building on the existing conditions which have necessitated the Major Threat to be detected over an average five minutes, therefore we assume that the detection time of Minor Threat must not exceed 1 minute since Major Threats already requires 4 minutes to be detected.

The comparison of the Minor Threat Mitigation for Plymouth University Threat Management before the application of the Threat Modelling and after the application of the Threat Modelling as presented in Table 4.11 shows that there is a drastic reduction in the number of signature rule updates after the application of the Threat Modelling for minor threat mitigation from 18701 and 19082 to 5 and 5 for Snort and Suricata NIDS respectively. The addition of the five rules meets with the initial scope of Network Security Management. The resulting number of signature rules is 14 and 15 for Snort and Suricata respectively The detection time for the Minor Threat lapsed 0.01666 and 0.01666 minutes in Snort and Suricata respectively. These are negligible and show that the new updates do not have any adverse effect on the Major Threat Mitigation.

Also, the comparison of the Minor Threat Mitigation for MIT Lincoln Lab Threat Management before the application of the Threat Modelling and after the application of the Threat Modelling as presented in Table 4.12 shows that there is a drastic reduction in the number of signature rule updates after the application of the Threat Modelling for minor threat mitigation from 18701 and 19082 to 5 and 5 for Snort and Suricata NIDS respectively. The addition of the five rules meets with the initial scope (15 signature rules, 5 minutes) of Network Security Management. The detection time for the Minor Threat lapsed for 0.05 minutes and 1.25 minutes in Snort and Suricata cases. Since, these are are less than 1.5 minutes, the additional time of detection is negligible; hence, the new updates do not have any adverse effect on the Network Threat Management.

The main problem with intrusion detection is false alarm (false positive). Hence, the performance of the rules updates in Threat Mitigation is examined in the context of both Snort and Suricata NIDS for the Plymouth University and MIT Lincoln Lab Threats' Network Security Management. With the application of the signature rules, the false alarm rate reduced from 99.15% and 99.81% to 2.78% and 1.28% for Snort and Suricata respectively in the Plymouth University Advanced Persistent Threat Management. The chart for the comparison is presented in Figure 4.6. Also, the application of the signature rules to combat MIT

Lincoln Threat makes the false alarm rate to reduce from 99.9% and 99.04% to 21.16% and 1.38% for Snort and Suricata NIDS respectively. The chart for the comparison is presented in Figure 4.7.

CHAPTER FIVE

SUMMARY, CONCLUSION AND RECOMMENDATIONS

This chapter summarises the steps taken in the study; it reviews the achievements of the research and presents the limitations. It also presents the contribution to knowledge and potential of new studies within the context of the research.

5.1 Summary

The study was aimed at modelling Minor Threats for the purpose of mitigating harmful Minor Threats in Network Threat Management, where the threats belongs to Internet-facilitated Organised Crime Threats and the Network Threat Management takes place in Collaborative Network Security Management System without the effect of Privacy, Interoperability, Quality, Trust, Multidimensionality and Uncertainty Issues. Collaborative Network Security Management was adapted as framework for modelling and mitigating Minor Threats due to its effectiveness.

Threat Prediction, Threat Prioritisation and Threat Mitigation Models were integrated and incorporated into the framework. Minor Threats from standard scenario threats such as MIT Lincoln Lab Threats otherwise known as DARPA2000 and Plymouth University Advanced Persistent Threats (PUAPT) were created. Threat Prediction Model was designed to identify Minor Threats using actionable sequential association data mining technique while Threat Prioritisation Model was designed to rate the threats using Dempster-Shafer method and expectation theory. Both models were implemented using java application programming interface. Standard Hillson's risk mitigation model was used as template for Threat Mitigation Model while Snort and Suricata security onion set-up was used to implement it.

The Threat Prediction Model accurately identified each step of MIT Lincoln Lab Threats and Plymouth University Advanced Persistent Threats. In the case of Threat Prioritisation Model, harmful Minor Threats were rated high while the non-harmful were rated low. The correlations of its threat rating scores and original steps in Network Threat Management of both MIT Lincoln Lab Threats and Plymouth University Advanced Persistent Threats were positively significant with Spearman's correlation coefficients above 0.5. The results of the Threat Mitigation Model showed that there were drastic reduction in the cost of detection based on Emerging Threats(ET) rules, time of detection and false positive (alarm) rates for Snort and Suricata in the two evaluations.

By these outcomes, the following assertions are drawn:

a. Centralization of Incident Sharing and Analysis enhances effective control of privacy, trust, quality and interoperability in Collaborative Network Security Management.

b. Multi-sensor and multi-target based actionable attribution improves the performance of Data Mining in Minor Threat Prediction Modelling.

c. Attacker and Victim perspectives of Intrusion Threat Model based on the perspectives of attack, asset and defence improves Minor Threat Prioritisation Modelling.

d. The strategic application of Collaborative Network Security Management, Predictive Analysis and Hybrid-centric Threat Modelling perspectives in Threat Modelling trivializes the cost and risk of Mitigating Minor Threats in cost-effective Network Threat Management.

5.2 Conclusion

The Collaborative Network Security Management Framework has been able to improve the modelling of Minor Threats and trivialise the cost and time incurred in mitigating Minor Threats. It has even improved the quality of mitigation in Network Threat Management by reducing the false alarm rate. The proposed

Threat Prediction Model, Threat Prioritisation Model and Threat Mitigation Model have led to modelling and mitigating Minor Threats accurately without exceeding the scope of Network Threat Management. This has changed the focus of Threat Management from Modelling and Mitigating Major Threats to Modelling and Mitigation of both Minor Threats and Major Threats. It has even shown that the mitigation could still be performed within the scope of existing Network Threat Management even in the complex domain of Multiple Information Security (InfoSec) Sensors.

The Threat Prediction Model of the Network Threat Management outperforms the existing models. The comparison of the Threat Prediction Model with Li *et al.,* (2007) Sequential Association Mining Algorithm showed the ability the new model to predict complex attack scenario with support of 3 and confidence of 1. The results conforms to earlier studies that a novice attacker exploits easy-to-use exploit kits and use same strategy for which he derives the maximum benefits (Bhattacharya and Ghosh, 2008). The use of actionable attributes such as Src IP and Dest IP allowed threats to be traced to their sources.

Moreover, the Threat Prioritisation Model of the Network Threat Management has proven to be better than standard tools such as Snort and Common Vulnerability Scoring System at prioritizing Minor Threats. In fact, it has even shown that threats with no CVE-ID can still be ranked adequately unlike (Jumaat, 2012) which derived some parameters scores from the CVSS.

The Threat Mitigation Model proved that the strategic mitigation of Minor Threat using Hilson (1999) Risk Mitigation Model would not aggravate the cost and time of detection of Network Threat Management. It also showed that the mitigation of Minor Threat would reduce false alarm rate.

Some of the limitations of the work include:

a. *Limitation to Experimental Testbed:* Although, the Plymouth University testbed was designed using the possible Internet-facilitated Organised Crime Threats scenario, however, a real enterprise testbed would have been more appropriate but it was not used because of the difficulty of finding such testbeds. Hence, the research ended with the development of a framework for Management of Internet-facilitated Organised Crime Threats.

b. *Few InfoSec Tools:* Only Snort and Suricata NIDS were used for the experiment because of the ease of interoperation. Also Security Onion Suit was used because it can be operated in collaborative manner; it can easily be managed and there is no other Threat Management suit with NIDS (application, packet, process and system NIDS) and Firewall (packet, application and proxy firewall) except it.

c. *Few Network Security Management Domains:* Due to the difficulty of finding Network Security Managers and the sensitivity of the work, only four Network Security Managers participated in the analysis. This however meets up with minimum requirement for effective Network Threat Management as posited by Chen *et al.* (2007).

5.3 Recommendations

In the aspect of industrial significances, the framework can be applied in providing adequate security in Distributed Systems and Enterprise Networks. It can also be applied in providing security to Cloud Clients, Customers of Internet Service Provider and improving Security Information and Event Management Forum.

Furthermore, this research had been able to open up some key areas for future research. These areas include:

a. Application of the Threat Modelling and Mitigation Approach to Major Threat Prediction, Prioritisation and Mitigation.

b. Automation of the entire Threat Modelling and Mitigation process: In this wise, Software Agent Autonomous Automation System would be a good choice.

c. In the aspect of the Threat Prediction, the application of an intelligent computational technique will enhance the selection of interesting sequence. In this regard, the potential of Evolutionary Techniques and Neural Networks can be explored.

d. In the aspect of the Threat Prioritisation, study on the potential of Fuzzy Logic and Neuro-Fuzzy System will be a good choice to enhance automation and remove imprecision.

e. Also, the aspect of Central Administrative System's Security and Trust will be a novel area of research. In this wise, the potentials of Intrusion Detection System, Firewall, Biometric, and Cryptography could be studied.

f. The existing Network Threat Management System such as Security Onion can be extended by integrating the framework with it at the code level.

5.3 Contribution to Knowledge

The framework improved accuracy of Predicting and Prioritising Minor Threats, reduced to negligible level the cost of detection, time of detection and false alarm rate of Mitigating Minor Threats and addressed the challenges of Privacy, Interoperability, Multidimensionality, Quality, Trust and Uncertainty associated with Incident Sharing and Analysis in Collaborative Network Security Management System. Specifically, the work contributed to knowledge in the following ways:

a. Improvement of the confidence of predicting Minor Threats from the previous maximum confidence of 0.26 to 1.0.

b. Improvement of the prioritisation of non-harmful and harmful Minor Threats with and without Common Vulnerability Exposure Identification as 'very low significant' and 'low significant' Minor Threats respectively from the previous correlation coefficient of 0.0 to 0.68.

c. Mitigation of Harmful Minor Threats based on distributed information Security sensors without affecting the required Cost of Detection, Time of Detection and False Alarm Rate of Network Threat Management.

d. Development of an improved Collaborative Network Security Management framework that manages trust, interoperability, privacy, uncertainty, quality and multidimensionality over collaborative network security management domains by using central administrative system as sharing and analysis server for modelling and mitigating Minor Threats.

REFERENCES

Addison-Wesley. 2005. Unified Modeling Language User Guide, The (2 ed.). p. 496. Retrieved 9th April, 2012 from http://www.informit.com

Agrawal, R. and Srikant, R. 1994. Fast algorithms for mining association rules in large databases. Proceedings of the 20th International Conference on Very Large Data Bases, VLDB, pages 487-499, Santiago, Chile, September 1994.

Ahmadi, M. 2008. The Need for Collaborative Threat Modelling. RFID Journal 2008. Retrieved 13th April, 2014 from . Retrieved 13th April, 2014 from http://www.rfidjournal.com.

Ahmed, M.S., Al-Shaer, E., Taibah, M., Khan, L. 2010. Objective Risk Evaluation for Automated Security Management.

Alberts, C. and Dorofee, A. 2004.Security Incident Response: Rethinking Risk Management", International Congress Series, Vol. 1268, pp. 141-146.

AlienVault. Retrieved April 4, 2014 from http://www.alienvault.com

Al-Shaer, E.S. and Hamed, H. H. 2004. Discovery of Policy Anomalies in Distributed Firewalls. IEEE Conf. Computer Communications, 2004, pp. 2605–2616.

Alsubhi, K., Al-Shaer, E. and Boutaba, R. 2008. Alert Prioritisation in Intrusion Detection Systems, Proceedings of the IEEE Network Operations and Management Symposium, Salvador, Brazil, pp. 33-40.

Ammann, P., Wijesekera, D., and Kaushik, S. 2002. Scalable,Graph-Based Network Vulnerability Analysis, In Proceedings of 9th ACM Conference on Computer and Communications Security, Washington, DC.

Amoroso, E. 2011. Cyber Attacks: Protection National Infrastructure. Burlington, MA: Elsevier.

AS/NZS ISO 31000 .2009. Risk Management Standard. Retrieved 2nd January, 2014 from www.standard.co.nz

Arbor Networks. 2012. Arbor Special Report: Worldwide Infrastructure Security Report 2011. Volume VII. Retrieved 8th January, 2013 from www.arbornetworks.com/report.

Årnes, A., Valeur, F., Vigna, G. and Kemmerer, R. 2006. Using Hidden Markov Models to Evaluate the Risks of Intrusions: System Architecture and Model Validation", Proceedings of the International Symposium on Recent Advances in Intrusion Detection (RAID), Hamburg, Germany, pp. 145–164.

Atkinson, R.1999. Project management: cost, time and quality, two best guesses and a phenomenon, its time to accept other success criteria. International Journal of Project Management Vol. 17, No. 6, pp. 337-342.

Baker, S., Ponniah, D. and Smith, S. 1999. Risk response techniques employed currently for major projects. Construction Management and Economics, Vol. 17, pp. 205-213.

Banday, M.T., Qadri, J.A., Shah, N.A. (2009). "Study of Botnets and Their Threats to Internet Security ," . *Sprouts: Working Papers on Information Systems*, 9(24). Retrieved I9th May, 2012 from http://sprouts.aisnet.org/9-24.

Barber, B. 1983. *The logic and limits of trust.* New Brunswick, NJ: Rutgers University Press.

Bartal, Y., Mayer, A., Nissim, K. and Wool, A. 1999. Firmato: A Novel Firewall Management Toolkit. Proceedings of 1999 IEEE Symposium on Security and Privacy.

Ben-David and Raz, T. 2001. An integrated approach for risk response development in project planning. Journal of the Operational Research Society, Vol. 52, pp. 14-25.

Benferhat, S., Autrel, F. and Cuppens, F. 2003. Enhanced Correlation in an Intrusion Detection Process Second International Workshop Mathematical Methods, Models and Architectures for Computer Networks Security, September 20-24, 2003.

Berberidis, C., Angelis, L. and Vahavas, I. 2004. Inter-transaction association rules mining for rare events prediction. Paper presented at the 3rd Hellenic Conference on Artificial Intellligence (SETN'04).

Bhattacharya, S. and Ghosh, S. K.2008. A Decision Model based Security Risk Management Approach. Proceedings of the International MultiConference of Engineers and Computer Scientists IMECS 2008, Hong Kong, Vol. II, 19-21 March, 2008.

Bolzoni, D., Crispo, B. and Etalle, S. 2007. ATLANTIDES: An Architecture for Alert Verification in Network Intrusion Detection Systems. 21st Large Installation System Administration Conference (LISA '07).

Brown, W., Nasuti, F. 2005. Sarbanes-Oxley and Enterprise Security: IT Governance and What It Takes to Get the Job Done. Information Systems Security 14(5), 15{28 (2005).

Brumley, D. 2012. Firewalls and Intrusion Detection Systems. Retrieved May 12, 2013 from www.users.ece.cmu.edu

Bryant, J. 2013. New Mapp Initiatives. Retrieved 4th April, 2014 from http://blogs.technet.com/

Burns, S.F. 2005. Threat Modelling: A Process To Ensure Application Security, GIAC Security Essentials Certification (GSEC) Practical Assignment Version 1.4c, SANS Institute InfoSec Reading Room.

Butler, J. K., Jr. 1991. Toward understanding and measuring conditions of trust: Evolution of a conditions of trust inventory. *Journal of Management,17,* 643–663.

Butler, S.A.2002. Security attribute evaluation method: a cost-benefit approach. In:ICSE'02: Proc. of the 24rd International Conference on Software Engineering. pp. 232{240. ACM Press.

Bye, R., Camtepe S. A., and Albayrak S., Collaborative intrusion detection framework: Characteristics, adversarial opportunities and countermeasures, in Proc. USENIX Symposium on Networked Systems Design and Implementation, Cambridge, MA, USA, 2007, pp. 1-12.

CAIDA .2003. Slammer Worms. Retrieved 4th May, 2014 from www.caida.org.

Cardenas, A.A, Baras, J.S. and Ramezani, V. 2004. Distributed Change Detection for Worms, DDoS and other Network Attacks.

Carroll, J. M., 1983. Decision Support Risk Analysis, Computer and Security," vol. 2, issue 3, pp. 230 – 236, Nov, 1983

Caswell, B. and Beale, J. (2004), "Snort 2.1 Intrusion Detection", 2nd edition, Syngress.

Caswell, B. and Roesch, M. 1998. Snort: The open source network intrusion detection system. Retrieved 10th April, 2014 from http://www.snort.org .

Cavusoglu, H., Cavusoglu, H., Raghunathan, S. 2004. Economics of IT Security Management: Four Improvements to Current Security Practices. Communications of the Association for Information Systems 14, 65 {75}.

Chen, Y. and Malin, B. 2011. Detection of Anomalous Insiders in Collaborative Environments via Relational Analysis of Access Logs. *CODASPY'1-ACM*, San Antonio, Texas, USA.

Chen, Y., Hwang, K and Ku, W. 2007. Collaborative Detection of DDoS Attacks over Multiple Network Domains. IEEE Transactions on Parallel and Distributed Systems, TPDS-0228-0806.

Chen, X., Mu, B., and Chen, Z. 2011. NetSecu: A collaborative network security platform for in-network security, in Proc. the 3rd International Conference on Communications and Mobile Computing (CMC), Qingdao, China, pp. 59-64.

Chen, Z., Han, F., Cao, J., Jiang, X., and Chen, S. 2013. Cloud Computing-Based Forensic Analysis for Collaborative Network Security Management System. Tsinghua Science and Technology ISSN 1007-0214 05/12 pp40-50 Volume 18, Number 1.

CIF, Project. 2009. Collective-Intelligence-Framework. Retrieved 4th April, 2014 from https://code.google.com/p/collective-intelligence-framework/

Cisco Systems Inc. 2009. Signature Engines, Retrieved 2nd January, 2014 from http://www.cisco.com

Clark, R. 2010. Intelligence analysis: A target-centric approach. (Third ed.). Washington, DC: CQ Press.

Computer Economics. 2009. Security Threats in Employee Misuse of IT Resources Retrieved 5th April, 2014 from http://www.computereconomics.com/article

Connolly, J. 2013. The trusted automated exchange of indicator information (TAXII). Retrieved 4th April, 2014 from http://taxii.mitre.org/about/documents/.

CORAS. 2000. A platform for risk analysis of security critical systems. IST-2000-25031, 2000. Retrieved 8th January, 2013 from http://coras.sourceforge.net.

CorreLog. Retrieved 4th May, 2014 from https://correlog.com

COSO. 2004 .Enterprise Risk Management — Integrated Framework Executive Summary. Retrieved 2nd January 2014 from www.coso.org

Cox, K. J. and Gerg C. 2004. Anatomy of an Attack: The Five P, in Managing Security with Snort & IDS Tools. O'Reilly Media.

Cuppens, F., Autrel, F., Miege, A. and Benferhat S. 2002. Recognizing Malicious Intention in an Intrusion Detection Process. In Soft Computing Systems - Design, Management and Applications, volume 87, 806–817, 2002.

CVSS.2014. Common Vulnerability Scoring System version 2 Retrieved 4th July, 2014 from http://www.first.org/cvss/cvss-guide.html

Daneva, M. 2006. Applying Real Options Thinking to Information Security in Networked Organisations. Tech. Rep. TR-CTIT-06-11, Centre for Telematics and Information Technology, University of Twente, Enschede.

Dantu, R., Kolan, P., Akl, R., Loper, K. 2007. Classification of Attributes and Behaviour in Risk Management using Bayesian Networks.1-4244-1330-3/2007 IEEE.

Danyliw, R.,Meijer, J.and Demchenko, Y. 2007. The Incident Object Description Exchange Format. Network Working Group, RFC 5070. Retrieved 2nd April, 2014 from www.ietf.org

DARPA. 2014. DARPA Intrusion Detection Data Sets. Retrieved 10th April, 2014 from http://www.ll.mit.edu/mission/communications/ist/corpora/ideval/data/index.html

Debar, H., Curry, D. and Feinstein, B. 2007. The Intrusion Detection Message Exchange Format (IDMEF), Network Working Group, RFC 476. http://www.ietf.org/rfc/rfc4765.txt

Shafer, G. 1976. A Mathematical Theory of Evidence. Princeton University Press

Dondo, M. 2009. A Fuzzy Risk Calculations Approach for a Network Vulnerability Ranking System. DRDC Ottawa Defence R&D Canada – Ottawa, Technical Memorandum DRDC Ottawa TM 2007-090.

Duggan, D. P., Thomas, S. R.. C., Veitch, K. K. and Woodard, L. 2007. Categorizing Threat: Building and Using a Generic Threat Matrix. Albuquerque, NM: Sandia National Laboratories.

Dynes, S., Eric, H.B., Johnson, M.E. 2005. Information Security in the Extended Enterprise: Some Initial Results From a Field Study of an Industrial Firm. In: Proc. of Int. Workshop on the Economics of Information Security.

Einarsson, S. and Mavin, R. 1998. An approach to Vulnerability analysis of complex industrial systems. Risk Analysis, 18(5):535-545.

Elahi, G., Yu, E., and Zannone, N. 2011.Security Risk Management by Qualitative Vulnerability Analysis. Retrieved April 4, 2014 from http://security1.win.tue.nl/

EmergingThreats. 2013. Enhance your intrusion detection system with etpro™ ruleset. Retrieved 4th December, 2013 from http://www.emergingthreats.net/solutions

ENISA (2006) "Risk Management: Implementation principles and Inventories for Risk Management/Risk Assessment methods and tools." Retrieved 8th January, 2013 from from www.enisa.europa.eu

Eppstein, D. and Muthukrishnan, S. 2001. Internet Packet Filter Management and Rectangle Geometry. Proceedings of 12th Annual ACM-SIAM Symposium on Discrete Algorithms (SODA).

Europol. 2014. "The Internet Organised Crime Threat Assessment," Retrieved 2nd May, 2014 from *https://www.europol.europa.eu/sites/default*

Farhadi, H., AmirHaeri, M., and Khansari, M. 2011. Alert Correlation and Prediction Using Data Mining andHMM. The ISC Int'l Journal of Information Security. July 2011, Volume 3, Number 2 pp. 77-101 Retrieved 13th March 2012 from http://www.isecure-journal.org

Farnham, G. 2013. Tools and Standards for Cyber Threat Intelligence Projects. GIAC (GCPM) Gold Certification.

Fayyad, U., Shapiro, G. P., and Smyth, P. From data mining to knowledge discovery in databases. AI Magazine, 17(3):37-54, Fall 1996. Retrieved 2nd April, 2014 from http://citeseer.ist.psu.edu/fayyad96from.html

FERMA.2002. A Risk Management Standard by the Federation of European Risk Management Associations. Retrieved 2nd January, 2014 from http://www.ferma.eu/risk-management/standards

Geib, C.W.and Goldman, R.P. 2001. Plan Recognition in Intrusion Detection Systems. In Proceedings of the Second DARPA Information Survivability Conference and Exposition, 2001.

Gomez M.A.N. 2011. Indentifying Phases of a Multistage Attack via Clustering, Philippine Computing Science Congress, 2011.

Georgakopoulos, D., Nodine, M., Baker, D. and Cichocki, A. 2006. Awareness-Enabled Coordination for Large Scale Collaboration Management International Symposium on Collaborative Technologies and Systems CTS 2006.

Gordon, L.A. and Loeb, M.P.2006. Budgeting Process for Information Security Expenditures. Communications of the ACM 49(1), 121-125.

Gordon, L.A., Loeb, M.P. and Lucyshyn, W. 2003. : Information Security Expenditures and Real Options: A Wait-and-See Approach. Computer Security Journal 19(2), 1-7.

Gordon, L.A., Loeb, M.P., Lucyshyn, W., and Rochardson, R. 2005. CSI/FBI computer crime

and security survey, Technical Report, Computer Security Institute, 2005.

Gregg, M. and Kim, D. 2005. Inside Network Security Assessment: Guarding your IT Infrastructure, Sams.

Ha, D., Upadhyaya, S., Ngo, H., Pramanik, S., Chinchani, S. and Mathew, S. 2006. Insider Threat Analysis using Information-centric Modelling. Advances in digital forensics iii.

Haines, J.W., Rossey, L.M., and Lippmann, R.P. 2001. Extending the DARPA Off-Line Intrusion Detection Evaluations. Submitted to DISCEX-II

Han, J. and Kamber, M. 2000. Simon Fraser University. Data Mining: Concepts and Techniques. Simon Fraser University. Morgan Kaufman.

Hari, B., Suri, S. and Parulkar, G. 2000. Detecting and Resolving Packet Filter Conflicts. Proceedings of IEEE INFOCOM'00.

Haslum, K., Abraham, A., and Knapskog, S. 2007. DIPS: A Framework for Distributed Intrusion Prediction and Prevention Using Hidden Markov Models and Online Fuzzy Risk Assessment, In Proceedings of the Third International Symposium on Information Assurance and Security.

Haslum, K. 2010. Real-time network intrusion prevention. Doctoral theses at NTNU, 2010:168.

Hasan, R. and Myagmar, S. 2005. Toward a Threat Model for Storage Systems. ACM 1-59593-223-X/05/0011

Hazelhusrt, S. 1999. Algorithms for Analyzing Firewall and Router Access Lists. Technical Report TR-WitsCS-1999, Department of Computer Science, University of the Witwatersrand, South Africa.

Heyman, T., Win, B. D., Huygens, C., and Joosen, W. 2006. Improving Intrusion Detection through Alert Verification. WOSIS 2006 Retrieved 13th February, 2014 from www.cs.kuleuven.be.

Hillson, D. 1999. Developing Effective Risk Responses, Proceedings of the 30th Annual Project Management Institute 1999 Seminars & Symposium, Philadelphia, Pennsylvania, USA.

Hillson, D. 2002. Extending the risk process to manage opportunities. International Journal of Project Management, Vol. 20 No. 3, pp. 235-240.

Hu, J. and Yang-Li, Z. 2007. Association Rules Mining Using Multi-objective Co-evolutionary Algorithm. International Conference on Computational Intelligence and Security Workshops, 2007. CISW 2007.

IBM, Retrieved May, 2014 from http://ww.ibm.com

Incapsula. 2014. Distributed Denial of Service Attack (DDoS) Definition, DDoS Protection Services. Retrieved 9th September, 2014 from http://www.incapsula.com/ddos/ddos-attacks/

ISACA, Retrieved 7th May, 2014 from http://www.isaca.org/COBIT.

ISO17799:2000 Code of Practice for Information Security Management. Retrieved 8th January, 2013 from http://csrc.nist.gov/publications/secpubs/otherpubs/

ISO-17799. 2005. Code of Practice for Information Security Management. Retrieved 8th January, 2013 from http://iso.org

ISO, Retrieved 7th May, 2014 from http://www.iso27001security.com

ISO/1EC 27005. 2010. Technical: FAIR – ISO/IEC 27005 Cookbook. The Open Group. Document Number: C103.

ISO-21827. 2007. Retrieved 8th January, 2013 from www.iso.org.

ISO-17799. 2005. Retrieved 8th January, 2013 from www.iso.org

ISO-13335-1. 2004. Retrieved 8th January, 2013 from www.iso.org.

ISO-7498-2. 1989. Retrieved 8th January, 2013 from www.iso.org.

ISO-21827.2007. Retrieved 8th January, 2013 from www.iso.org.

Jemili, F., Zaghdoud, M. and Ahmed, M.B. 2009. Hybrid Intrusion Detection and Prediction multiAgent System, HIDPAS. (IJCSIS) International Journal of Computer Science and Information Security, Vol. 5, No.1, 2009.

Jha, S., Sheyner, O., and Wing, J. 2002. Two formal analyses of attack graphs. In proceedings of the 15th computer security foundation workshop.

Johnson, A., Dempsey, K., Ross, R., Gupta, S. and Bailey, D. 2011. Guide for Security-Focused Configuration Management of Information Systems: INFORMATIONSECURITY.NIST Special Publication 800-128.

Jumaat, A. N. B. 2012. Incident Prioritization for Intrusion Response. University of Plymouth, Unpublished Ph.D. Thesis.

Kahn, C., Porras, P. A., Staniford-Chen, S., and Tung, B. 1998. A Common Intrusion Detection Framework. Journal of Computer Security. Retrieved 16th March, 2014 from www.gost.isi.edu/cidf

Kang, X., Zhou, D., Rao, D., Li, J. and Lo, V. 2004. Sequoia – A Robust Communication Architecture for Collaborative Security Monitoring Systems. Retrieved 4th April, 2014 from http://netsec.cs.uoregon.edu/research/sequoia.php

Kaplan, S. and Garrick, B. J. 1981. On The Quantitative Definition of Risk, *Risk Analysis*, Vol. 1 No. 1, pp. 11-27.

Kaspersky Security Bulletin. 2009. Malware Evolution 2009. Retrieved 4th April, 2014 from http://kaspersky.com

Katipally, R., Cui, X. and Yang, L. 2010. Multi stage attack Detection system for Network Administrators using Data Mining.

Kee, H. W., & Knox, R. E. 1970. Conceptual and methodological considerations in the study of trust and suspicion. Journal of Conflict Resolution, 14, 357–366.

Kijewski, P. and Pawliński, P. 2012. Proactive Detection and Automated Exchange of Network Security Incidents. S&T Organisation,STO-MP-IST-111

Killourhy, K.S., Maxion, R.A. and Kymie, M.C.T, 2004. A Defence-Centric Taxonomy based on Attack Manafestation, International Conference on Dependable Systems and Networks.

Koller, G. R. 2000. Risk Assessment and Decision Making in Business and Industry: A practical guide. CRC press, LLC.

Kruegel, C., Valeur F. and Vigna G. 2004. Intrusion Detection and Correlation: Challenges and Solutions, University of California, Santa Barbara: Springer.

Kruegel, C., Robertson, W. and Vigna, G. 2004. Using Alert Verification to Identify Successful Intrusion Attempts. PIK 27 (2004) 4 FEHLT NOCH

Lee, W. and Qin, X. 2003. Statistical causality analysis of INFOSEC alert data. Proceedings of the Recent Advances in Intrusion Detection, Pittsburgh, PA, USA, Vol. 2820/2003, pp. 73-93.

Li, J., Lo, V., Kang, X, Zhou, D. and Rao, D .2004. Resilient and Self-Organizing Overlay of Collaborative Security Monitors, Retrieved 4th April, 2014 from http://netsec.cs.uoregon.edu/research/sequoia.php

Li, W. and Tian, S. 2010. An ontology-based intrusion alerts correlation system. Expert Systems with Applications 37:7138–7146.

Li, Z., Lei, J., Wang, L., and Li D. 2007. A Data Mining Approach to Generating Network Attack Graph for Intrusion Prediction. Computer Communications 29.

Locasto, M.E., Parekh, J.J., Keromytis, A.D., Stolfo, S.J. 2005. Towards Collaborative Security and P2P Intrusion Detection. Proceedings of the 2005 IEEE Workshop on Information Assurance and Security T1B2 1555 United States Military Academy, West Point, NY, 15.

Luo, J. and Bridges, S.M. 2000. Mining Fuzzy Association Rules and Fuzzy Frequency Episodes for Intrusion Detection. International Journal of Intelligent Systems, Volume 15 No 1.

Mao, Z.M., Sekar, V., Spatscheck, O., Merwe, J.V. and Vasudevan R. 2006. Analyzing Large DDoS Attacks Using Multiple Data Sources, Proceeding of the 2006 SIGCOMM workshop on Large Scale attack defence. Pages 161-168, ACM, New York, USA.

Maimon, O. and Rokach, L. 2009. Introduction to Knowledge Discovery in Databases. Retrieved 4th April, 2014 from www.ise.bgu.ac.il

MalwareDomains. 2013. DNS-bh – malware domain blocklist. Retrieved 4th April, 2014 from http://www.malwaredomains.com/

Matousek, P., Rab, J., Rysavy, O. and Sveda, M. 2008. A Formal Model for Network-Wide Security Analysis. Proc. IEEE Int. Conf. Engineering of Computer Based Systems, 2008.

Mayer, A., Wool, A.and Ziskind, E. 2000. Fang: A Firewall Analysis Engine. Proceedings of 2000 IEEE Symposium on Security and Privacy.

Mayer, R. C., Davis J. H., and Schoorman F. D. 1995. An Integrative Model of Organisational Trust. Academy of Management Review, 20, 709–734.

McAfee. Retrieved 4th May from www.mcafee.com

McHugh, J., Christie, A. and Allen, J. 2001. Intrusion Detection1: Implementation and Operational Issues. CROSSTALK- The Journal of Defense Software Engineering, Software Engineering Institute, Computer Emergency Response Team/Coordination Centre.

Mell, P., Scarfone, K. and Romanosky, S .2009. "A Complete Guide to the Common Vulnerability Scoring System Version 2.0", Rerieved 1st May 2014 from http://www.first.org/cvss/cvss-guide.html.

Michalski, J. 2010. Threat Characterization. Cybersecurity for Energy Delivery Systems 2010 Peer Review Alexandria, VA, July 20-22, SNL Department 5621.

Michalski, J., Veitch, C., Trevino, C. Mateski, M, Frye, J., Harris, M. and Maruoka, S.. 2012. Cyber Threat Metrics. SAND2012-2427, Unlimited Release, Printed March 2012.

Meier, J.D., Mackman, A., Dunner, M., Vasireddy, S., Escamilla, R. and Murukan, A. 2003. Improving Web Application Security: Threats and Countermeasures, Threat Modelling, Microsoft Corporation.

Michel, C. and Me, L. 2001. An Attack Distribution Language for Knowledge-based Intrusion Detection, in 16th International Conference on Information Security.

Mike, S. The Common Vulnerability Scoring System (CVSS) (online), Retrieved 2nd March, 2012 from http://www.first.org/cvss/

Moriarty, K. 2012. Real-time Inter-network Defense (RID), RFC 6545. Retrieved 2nd April, 2014 from www.ietf.org

Mu, B., Chen, X., and Chen, Z., 2011. A Collaborative Network Security Management System in Metropolitan Area Network, in Proc. the 3rd International Conference on Communications and Mobile Computing (CMC), Qingdao, China, 2011, pp. 45-50.

Mu, C.P., Li, X.J., Huang, H.K. and Tian, S.F. 2008. Online Risk Assessment of Intrusion Scenarios Using D-S Evidence Theory. Proceedings of the 13th European Symposium on Research in Computer Security, Malaga, Spain, pp. 35-48.

Munteanu. A.B. 2006. Information Security Risk Assessment: The Qualitative Versus Quantitative Dilemma", Proceedings of the 6th International Business Information Management Association (IBIMA) Conference, Bonn, Germany, pp. 227-232.

Myagmar, S., Lee, A. J. and Yurcik, W.2005.Threat Modeling as a Basis for Security Requirements (SREIS). In Symposium on Requirements Engineering for Information Security.

Nanda, S. and Deo, N. 2007. A Highly Scalable Model for Network Attack Identification and Path Prediction.

Naqvi, M.F. 2011. Asset, Vulnerability, Threat, Risk and Control. Retrieved 1st May 2014 from www.slideshare.net/mfnaqvi.

Nath, B., Bhattacharyya, D.K. and Ghosh, A.2010. Discovering Association Rules from Incremental Datasets International Journal of Computer Science & CommunicationVol. 1, No. 2, July-December 2010, pp. 433-441

NCI. 2013. National council of isacs. Retrieved 5th February, 2014 from http://www.isaccouncil.org/home.html

Nedjah, N., Abraham, A. and Mourelle, L.M. 2006. Genetic Systems Programming: Theory and Experience. Springerlink Ed.

Nicolett, M. and Kavanagh, K. 2009. Magic Quadrant for Security Information and Event Management., Gartner RAS Core Research Note G00167782.

Ning, P., Peng, P., Hu, Y., and Xu, D. 2003. TIAA: A Visual Toolkit for Intrusion Alert Analysis. Retrieved 4th April, 2014 from http://www.iss.net.

Ning, P., Reeves, D.S. and Cui, Y. 2001. Correlating Alerts Using Prerequisites of Intrusions", Technical Report TR-2001-13, North Carolina State University, Department of Computer Science.

Ntouskas, T., Pentafronimos, G. and Papastergiou, S. 2011. STORM - Collaborative Security Management Environment. Information Security Theory and Practice. Security and Privacy of Mobile Devices in Wireless CommunicationLecture Notes in Computer Science Volume 6633, 2011, pp 320-335.

Oisen, R.P. 1971. Can project management be defined? Project Management Quarterly, 1971, 2(1), 12-14.

Olzak, T. 2006.A Practical Approach to Threat Modelling. Retrieved 4th April, 2014 from www.adventuresinsecurity.com

Olzak, T., 2008. Five phases of a successful network penetration, Retrieved 4th April, 2014 from http://www.techrepublic.com.

Pandey, N.K., Gupta, S. K. and Leekha, S .2008. Algebra for capability based attack correlation.

Paquet, C. 2013. Network Security Concepts and Policies. . Retrieved 19th January, 2014 from http://www.ciscopress.com/articles/article.asp?p=1998559

PCI, Retrieved 7th May, 2014 from https://www.pcisecuritystandards.org/tech

PFau, R. 2003. The Security Lifecycle. SANS Institute. Retrieved 7th May, 2014 from http://www.giac.org/registration/gsec.

Peng, T., Leckie, C. and Ramamohanarao, K. 2007. Detecting Distributed Denial of Service Attacks by Sharing Distributed Beliefs. Retrieved 7th October, 2014 from http://www.ee.mu.oz.au/cubin

Porras, P.A., Fong, M.W. and Valdes, A. 2002. A mission-impact-based approach to INFOSEC alarm correlation", Proceedings of the 5th International Symposium Recent Advances in Intrusion Detection, Zurich, Switzerland, Vol. 2516, pp. 95-114.

Project Management Institute. 2008. A Guide to the Project Management Body of Knowledge (PMBOK® GUIDE). Retrieved 4th April, 2014 from http://my.safaribooksonline.com.

Qin, X. and Lee, W. 2004, Attack Plan Recognition and Prediction Using Causal Networks, ACSAC-O4, 370-379, 2004.

Reeshi, N. 2013. Different Types of Network Attacks and Security Threats and Countermeasure Retrieved 4th April, 2014 from http://www.ayurveda.hubpages.com.

RFC4949. 2007. Request For Comment. Retrieved 4th April, 2014 from http://rfc.org

Ritchey, R. and Ammann, P. R. 2000. Using Model Checking to Analyze Network Vulnerabilities, IEEE Oakland Symposium on Security and Privacy.

Ross, S.M. 2007. Expectation of a Random Variable. Introduction to Probability Models (9th ed.). Academic Press. p. 38

RVA Program, 2010. Operational Threat Assessment Project Execution Plan for a Single Threat Assessment (DRAFT). Federal Network Security, Compliance & Assurance Program, U.S. Department of Homeland Security.

Saklikar, S. 2013. Sharing Threat Intelligence Analytics. RSA Conference, Asia-Pacific 2013. CLT-05 Intermediate Class.

Scott, S.J. 2002. Threat Management Systems, The State of Intrusion Detection.

Shafer, G. 1976. A Mathematical Theory of Evidence. Princeton University Press.

SensePost. 2011. Sense Modelling Threat Modelling. Retrieved 4th April, 2014 from http://www.slideshare.net/sensepost/corporate-threat-modelling

Sheyner, O., Haines, J., Jha, S., Lippmann, R. and Wing, J. 2002. Automated Generation and Analysis of Attack Graphs," in Proceedings of IEEE Symposium on Security and Privacy, Oakland, California.

Snorby.2011. Snorby - All about Simplicity Retrieved 3rd April, 2014 from http://snorby.org/.

Swiderski, F. and Snyder, W. 2004. Threat Modelling. Microsoft Press.

Tjhai, G.C., Furnell, S.M., Papadaki, M. and Clarke, N.L. 2010. A preliminary two-stage alarm correlation and filtering system using SOM neural network and K-means algorithm, Computers & Security, Vol. 29 No. 6, pp. 712-723.

SANS. 2012. Beyond Continuous Monitoring: Threat Modelling for Real-time Response

A SANS Whitepaper, October 2012.

Schneier, B. 2000. Secrets and Lies: Digital Security in a Networked World John Wiley & Sons.

Schneier, B.1999. Attack Trees. Retrieved 4th April, 2014 from www.counterpane.com.

Sharma, A. and Erramilli, V. (2004) Worms: Attacks, Defense and Models.

Shostack, A.2014. Threat Modelling: Designing for Security. From Shostack Adam, Microsoft Threat Modelling Expert. John Wiley & Sons. Retrieved 13th April, 2014 from http://books.google.co.uk

Staniford, S., Paxson, V., and Weaver, N. 2002. How to 0wn the internet in your spare time. in Proceedings of the 11th USENIX Security Symposium (Security '02), 2002.

Su, X. 2006. An Overview of Economic Approaches to Information Security Management. Technical Report TR-CTIT-06-30, University of Twente, 2006.

Symantec. 2003. Internet Security Threat Report, Volume 17 Retrieved 19th January, 2014 from www.symantec.com/content

Symantec .2012. Internet Security Threat Report, Volume 17 Retrieved 19th January, 2014 from www.symantec.com/content

Symantec .2013. Internet Security Threat Report, Volume 18 Retrieved 19th January, 2014 from www.symantec.com/content

Takahashi, T. 2013. Iodef-extension for structured cybersecurity information. Retrieved 4th April, 2014 from http://tools.ietf.org/html

Trend Micro, Inc, Retrieved 4th May, 2014 from www.trendmicro.co.uk

Ullrich, J. 2004 \Dshield home page." Retrieved 19th January, 2014 from http://www.dshield.org/.

Voorbraak, F. 2007. Dempster-Shafer Theory. Retrieved 19th January, 2014 from www.blutner.de/uncert/DSTh.pdf

Waltz, E. 1998. Information warfare principles and operations. Norwood, MA: Artech House, Inc.

Wang, L., Liu, A. and Jajodia, S. 2006. Using attack graphs for correlating, hypothesizing, and predicting intrusion alerts. Computer Communications 29 (2006) 2917–2933.

Wang, J. and Zhao, L. 2006. Experimental Design for Attack Scenario Traces to validate Intrusion Ddetection Alert Correlation. WSRC Paper 2006/4-1, Whartson-SMU Research Centre.

Websense. 2011. “Advanced Persistent Threats and Other Advanced Attacks: Threat Analysis and Defense Strategies for SMB, Mid-Size, and Enterprise Organisations,” Retrieved 2nd May, 2014 from https://www.websense.com

Weforum. 2012. “Organised Crime Enablers,” Retrieved 2nd May, 2014 *https://www.weforum.org*

Whitman, M. E. and Mattord, H. J. 2004. Management of Information Security. Thompson Course Technology. Retrieved 19th January, 2014 from http://www.thomsonrights.com.

Yegneswaran, V., Barford, P. and Jha, S. 2004. Global Intrusion Detection in the DOMINO Overlay System. In Proceedings of Network and Distributed System Security Symposium (NDSS 2004).

Zaiane, O.R .1999. Principle of Knowledge Discovery in Databases. University of Alberta. Department of Computer Science. CMPUT690.

Zhu, B. and Ghorbani, A.A. 2006. Alert Correlation for Extracting Attack Strategies. International Journal of Network Security, Vol.3, No.3, PP.244–258

APPENDICES

Appendix 1: Java Code for Threat Prediction Tool Implementation

```
package oriolajade;
import org.jdesktop.application.Action;
import org.jdesktop.application.ResourceMap;
import org.jdesktop.application.SingleFrameApplication;
import org.jdesktop.application.FrameView;
import org.jdesktop.application.TaskMonitor;
import java.awt.event.ActionEvent;
import java.awt.event.ActionListener;
import javax.swing.Timer;
import javax.swing.Icon;
import javax.swing.JDialog;
import javax.swing.DefaultListModel;
import javax.swing.ComboBoxModel;
import javax.swing.JOptionPane;
import javax.swing.JFrame;
import java.sql.*;
import java.util.*;
import javax.swing.JFileChooser;
import java.io.File;
import javax.swing.filechooser.FileFilter;
import javax.swing.filechooser.FileNameExtensionFilter;
import javax.swing.JOptionPane;

import java.io.BufferedWriter;
import java.io.FileWriter;
import java.io.File;
import java.io.Writer;
import java.io.FileNotFoundException;
import java.io.IOException;

/**
 * The application's main frame.
 */
public class OriolaJadeView extends FrameView {
    String Output = "";

                String driver = "com.mysql.jdbc.Driver";
    String user = "root";
    String pass = "";
    String y="";

     // Step 1: Load the JDBC driver.

int dTime[];
int dTotalRows;
 ArrayList dTimeL= new ArrayList();
String source[];
 ArrayList sourceL= new ArrayList();
String  dest[];
 ArrayList destL= new ArrayList();
 String sequenceID[];
  ArrayList sequqnceIDL= new ArrayList();
 String eventID[];
  ArrayList eventIDL= new ArrayList();
 String sourceIP[];
  ArrayList sourceIPL= new ArrayList();
 String destIP[];
  ArrayList destIPL= new ArrayList();

 String GeneratedSeq[];
  ArrayList GeneratedSeqL= new ArrayList();
 String ConfiCapture[];
 ResultSet nrows;


           ArrayList retSeqList= new ArrayList();
          ArrayList SupportList= new ArrayList();
          ArrayList SupportValue= new ArrayList();


    public OriolaJadeView(SingleFrameApplication app) {
        super(app);

        initComponents();
        loadEvent();
        loadAsset();
        // status bar initialization - message timeout, idle icon and busy animation, etc
        ResourceMap resourceMap = getResourceMap();
        int messageTimeout = resourceMap.getInteger("StatusBar.messageTimeout");
        messageTimer = new Timer(messageTimeout, new ActionListener() {
            public void actionPerformed(ActionEvent e) {
                statusMessageLabel.setText("");
            }
        });
        messageTimer.setRepeats(false);
        int busyAnimationRate = resourceMap.getInteger("StatusBar.busyAnimationRate");
        for (int i = 0; i < busyIcons.length; i++) {
            busyIcons[i] = resourceMap.getIcon("StatusBar.busyIcons[" + i + "]");
        }
        busyIconTimer = new Timer(busyAnimationRate, new ActionListener() {
            public void actionPerformed(ActionEvent e) {
                busyIconIndex = (busyIconIndex + 1) % busyIcons.length;
                statusAnimationLabel.setIcon(busyIcons[busyIconIndex]);
            }
        });
        idleIcon = resourceMap.getIcon("StatusBar.idleIcon");
        statusAnimationLabel.setIcon(idleIcon);
        progressBar.setVisible(false);

        // connecting action tasks to status bar via TaskMonitor
        TaskMonitor taskMonitor = new TaskMonitor(getApplication().getContext());
        taskMonitor.addPropertyChangeListener(new java.beans.PropertyChangeListener() {
            public void propertyChange(java.beans.PropertyChangeEvent evt) {
                String propertyName = evt.getPropertyName();
                if ("started".equals(propertyName)) {
                    if (!busyIconTimer.isRunning()) {
                        statusAnimationLabel.setIcon(busyIcons[0]);
                        busyIconIndex = 0;
                        busyIconTimer.start();
                    }
                    progressBar.setVisible(true);
                    progressBar.setIndeterminate(true);
```

```
                } else if ("done".equals(propertyName)) {
                    busyIconTimer.stop();
                    statusAnimationLabel.setIcon(idleIcon);
                    progressBar.setVisible(false);
                    progressBar.setValue(0);
                } else if ("message".equals(propertyName)) {
                    String text = (String)(evt.getNewValue());
                    statusMessageLabel.setText((text == null) ? "" : text);
                    messageTimer.restart();
                } else if ("progress".equals(propertyName)) {
                    int value = (Integer)(evt.getNewValue());
                    progressBar.setVisible(true);
                    progressBar.setIndeterminate(false);
                    progressBar.setValue(value);
                }
            }
        });
    }

    @Action
    public void showAboutBox() {
        if (aboutBox == null) {
            JFrame mainFrame = OriolaJadeApp.getApplication().getMainFrame();
            aboutBox = new OriolaJadeAboutBox(mainFrame);
            aboutBox.setLocationRelativeTo(mainFrame);
        }
        OriolaJadeApp.getApplication().show(aboutBox);
    }

    /** This method is called from within the constructor to
     * initialize the form.
     * WARNING: Do NOT modify this code. The content of this method is
     * always regenerated by the Form Editor.
     */
    @SuppressWarnings("unchecked")
    // <editor-fold defaultstate="collapsed" desc="Generated Code">
    private void initComponents() {

        mainPanel = new javax.swing.JPanel();
        jButton3 = new javax.swing.JButton();
        btnGSequence1 = new javax.swing.JButton();
        btnStart = new javax.swing.JButton();
        txtInterval = new javax.swing.JTextField();
        btnGSequence = new javax.swing.JButton();
        btn_Step = new javax.swing.JButton();
        btn_Support = new javax.swing.JButton();
        btnLoadCSV = new javax.swing.JButton();
        btnWriter = new javax.swing.JButton();
        jLabel1 = new javax.swing.JLabel();
        menuBar = new javax.swing.JMenuBar();
        javax.swing.JMenu fileMenu = new javax.swing.JMenu();
        javax.swing.JMenuItem exitMenuItem = new javax.swing.JMenuItem();
        javax.swing.JMenu helpMenu = new javax.swing.JMenu();
        javax.swing.JMenuItem aboutMenuItem = new javax.swing.JMenuItem();
        statusPanel = new javax.swing.JPanel();
        javax.swing.JSeparator statusPanelSeparator = new javax.swing.JSeparator();
        statusMessageLabel = new javax.swing.JLabel();
        statusAnimationLabel = new javax.swing.JLabel();
        progressBar = new javax.swing.JProgressBar();
        fchooser = new javax.swing.JFileChooser();
        jFileChooser1 = new javax.swing.JFileChooser();

        mainPanel.setName("mainPanel"); // NOI18N

        org.jdesktop.application.ResourceMap resourceMap =
    org.jdesktop.application.Application.getInstance(oriolajade.OriolaJadeApp.class).getContext().get
    ResourceMap(OriolaJadeView.class);
        jButton3.setText(resourceMap.getString("jButton3.text")); // NOI18N
        jButton3.setName("jButton3"); // NOI18N
        jButton3.addActionListener(new java.awt.event.ActionListener() {
            public void actionPerformed(java.awt.event.ActionEvent evt) {
              jButton3ActionPerformed(evt);
            }
        });

        btnGSequence1.setText(resourceMap.getString("btnGSequence1.text")); // NOI18N
        btnGSequence1.setName("btnGSequence1"); // NOI18N
        btnGSequence1.addActionListener(new java.awt.event.ActionListener() {
            public void actionPerformed(java.awt.event.ActionEvent evt) {
                btnGSequence1ActionPerformed(evt);
            }
        });

        btnStart.setText(resourceMap.getString("btnStart.text")); // NOI18N
        btnStart.setName("btnStart"); // NOI18N
        btnStart.addActionListener(new java.awt.event.ActionListener() {
            public void actionPerformed(java.awt.event.ActionEvent evt) {
                btnStartActionPerformed(evt);
            }
        });

        txtInterval.setText(resourceMap.getString("txtInterval.text")); // NOI18N
        txtInterval.setName("txtInterval"); // NOI18N
        txtInterval.addActionListener(new java.awt.event.ActionListener() {
            public void actionPerformed(java.awt.event.ActionEvent evt) {
                txtIntervalActionPerformed(evt);
            }
        });

        btnGSequence.setText(resourceMap.getString("btnGSequence.text")); // NOI18N
        btnGSequence.setName("btnGSequence"); // NOI18N
        btnGSequence.addActionListener(new java.awt.event.ActionListener() {
            public void actionPerformed(java.awt.event.ActionEvent evt) {
                btnGSequenceActionPerformed(evt);
            }
        });

        btn_Step.setText(resourceMap.getString("btn_Step.text")); // NOI18N
        btn_Step.setName("btn_Step"); // NOI18N
        btn_Step.addActionListener(new java.awt.event.ActionListener() {
            public void actionPerformed(java.awt.event.ActionEvent evt) {
                btn_StepActionPerformed(evt);
            }
        });

        btn_Support.setText(resourceMap.getString("btn_Support.text")); // NOI18N
        btn_Support.setName("btn_Support"); // NOI18N
        btn_Support.addActionListener(new java.awt.event.ActionListener() {
            public void actionPerformed(java.awt.event.ActionEvent evt) {
                btn_SupportActionPerformed(evt);
            }
        });

        btnLoadCSV.setText(resourceMap.getString("btnLoadCSV.text")); // NOI18N
        btnLoadCSV.setName("btnLoadCSV"); // NOI18N
        btnLoadCSV.addActionListener(new java.awt.event.ActionListener() {
            public void actionPerformed(java.awt.event.ActionEvent evt) {
                btnLoadCSVActionPerformed(evt);
            }
        });

        btnWriter.setText(resourceMap.getString("btnWriter.text")); // NOI18N
        btnWriter.setName("btnWriter"); // NOI18N
        btnWriter.addActionListener(new java.awt.event.ActionListener() {
            public void actionPerformed(java.awt.event.ActionEvent evt) {
                btnWriterActionPerformed(evt);
            }
        });

        jLabel1.setText(resourceMap.getString("jLabel1.text")); // NOI18N
        jLabel1.setName("jLabel1"); // NOI18N

        javax.swing.GroupLayout mainPanelLayout = new javax.swing.GroupLayout(mainPanel);
        mainPanel.setLayout(mainPanelLayout);
        mainPanelLayout.setHorizontalGroup(
            mainPanelLayout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
            .addGroup(mainPanelLayout.createSequentialGroup()

    .addGroup(mainPanelLayout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADIN
    G)
              .addGroup(mainPanelLayout.createSequentialGroup()
                .addGap(1566, 1566, 1566)
                .addComponent(jButton3, javax.swing.GroupLayout.PREFERRED_SIZE, 61,
    javax.swing.GroupLayout.PREFERRED_SIZE)
                .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.UNRELATED)
                .addComponent(btnGSequence1, javax.swing.GroupLayout.PREFERRED_SIZE,
    63, javax.swing.GroupLayout.PREFERRED_SIZE))
              .addGroup(mainPanelLayout.createSequentialGroup()
                .addGap(42, 42, 42)

    .addGroup(mainPanelLayout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADIN
    G, false)
                  .addComponent(btn_Support, javax.swing.GroupLayout.DEFAULT_SIZE,
    javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE)
                  .addComponent(btn_Step, javax.swing.GroupLayout.DEFAULT_SIZE,
    javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE)
                  .addComponent(btnGSequence, javax.swing.GroupLayout.DEFAULT_SIZE,
    140, Short.MAX_VALUE)
                  .addComponent(btnWriter, javax.swing.GroupLayout.DEFAULT_SIZE,
    javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE)
                  .addComponent(btnLoadCSV, javax.swing.GroupLayout.DEFAULT_SIZE, 141,
    Short.MAX_VALUE)
                  .addComponent(btnStart, javax.swing.GroupLayout.DEFAULT_SIZE,
    javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE)
                  .addGroup(javax.swing.GroupLayout.Alignment.TRAILING,
    mainPanelLayout.createSequentialGroup()
                    .addGap(10, 10, 10)
                    .addComponent(jLabel1)
                    .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED,
    javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE)
                    .addComponent(txtInterval, javax.swing.GroupLayout.PREFERRED_SIZE,
    52, javax.swing.GroupLayout.PREFERRED_SIZE)))))
                .addContainerGap(javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE))
        );
        mainPanelLayout.setVerticalGroup(
            mainPanelLayout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
            .addGroup(mainPanelLayout.createSequentialGroup()
                .addGap(20, 20, 20)
                .addComponent(btnLoadCSV, javax.swing.GroupLayout.PREFERRED_SIZE, 43,
    javax.swing.GroupLayout.PREFERRED_SIZE)
                .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)
                .addComponent(btnStart)
                .addGap(11, 11, 11)

    .addGroup(mainPanelLayout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELI
    NE)
              .addComponent(txtInterval, javax.swing.GroupLayout.PREFERRED_SIZE,
    javax.swing.GroupLayout.DEFAULT_SIZE, javax.swing.GroupLayout.PREFERRED_SIZE)
              .addComponent(jLabel1))
                .addGap(18, 18, 18)
                .addComponent(btnGSequence)
                .addGap(18, 18, 18)
                .addComponent(btn_Step)
                .addGap(18, 18, 18)
                .addComponent(btn_Support)
                .addGap(18, 18, 18)
                .addComponent(btnWriter, javax.swing.GroupLayout.PREFERRED_SIZE, 43,
    javax.swing.GroupLayout.PREFERRED_SIZE)
                .addGap(1338, 1338, 1338)

    .addGroup(mainPanelLayout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELI
    NE)
              .addComponent(jButton3)
              .addComponent(btnGSequence1))
                .addContainerGap(javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE))
        );

        menuBar.setName("menuBar"); // NOI18N

        fileMenu.setText(resourceMap.getString("fileMenu.text")); // NOI18N
        fileMenu.setName("fileMenu"); // NOI18N

        javax.swing.ActionMap actionMap =
    org.jdesktop.application.Application.getInstance(oriolajade.OriolaJadeApp.class).getContext().get
    ActionMap(OriolaJadeView.class, this);
        exitMenuItem.setAction(actionMap.get("quit")); // NOI18N
        exitMenuItem.setName("exitMenuItem"); // NOI18N
        fileMenu.add(exitMenuItem);

        menuBar.add(fileMenu);

        helpMenu.setText(resourceMap.getString("helpMenu.text")); // NOI18N
        helpMenu.setName("helpMenu"); // NOI18N

        aboutMenuItem.setAction(actionMap.get("showAboutBox")); // NOI18N
        aboutMenuItem.setName("aboutMenuItem"); // NOI18N
```

```
        helpMenu.add(aboutMenuItem);

        menuBar.add(helpMenu);

        statusPanel.setName("statusPanel"); // NOI18N

        statusPanelSeparator.setName("statusPanelSeparator"); // NOI18N

        statusMessageLabel.setName("statusMessageLabel"); // NOI18N

        statusAnimationLabel.setHorizontalAlignment(javax.swing.SwingConstants.LEFT);
        statusAnimationLabel.setName("statusAnimationLabel"); // NOI18N

        progressBar.setName("progressBar"); // NOI18N

        javax.swing.GroupLayout statusPanelLayout = new javax.swing.GroupLayout(statusPanel);
        statusPanel.setLayout(statusPanelLayout);
        statusPanelLayout.setHorizontalGroup(
            statusPanelLayout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
            .addComponent(statusPanelSeparator, javax.swing.GroupLayout.DEFAULT_SIZE, 1710,
Short.MAX_VALUE)
            .addGroup(statusPanelLayout.createSequentialGroup()
                .addContainerGap()
                .addComponent(statusMessageLabel)
                .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED, 1540,
Short.MAX_VALUE)
                .addComponent(progressBar, javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, javax.swing.GroupLayout.PREFERRED_SIZE)
                .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)
                .addComponent(statusAnimationLabel)
                .addContainerGap())
        );
        statusPanelLayout.setVerticalGroup(
            statusPanelLayout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
            .addGroup(statusPanelLayout.createSequentialGroup()
                .addComponent(statusPanelSeparator, javax.swing.GroupLayout.PREFERRED_SIZE,
2, javax.swing.GroupLayout.PREFERRED_SIZE)
                .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED,
javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE)

.addGroup(statusPanelLayout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELI
NE)
                    .addComponent(statusMessageLabel)
                    .addComponent(statusAnimationLabel)
                    .addComponent(progressBar, javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, javax.swing.GroupLayout.PREFERRED_SIZE))
                .addGap(3, 3, 3))
        );

        fchooser.setName("fchooser"); // NOI18N

        jFileChooser1.setName("jFileChooser1"); // NOI18N

        setComponent(mainPanel);
        setMenuBar(menuBar);
        setStatusBar(statusPanel);
    }// </editor-fold>

   private void loadEvent()
```

```
    {
      try
      {
          Class.forName(driver);
    Connection con = DriverManager.getConnection("jdbc:mysql://localhost:3306/oriolajade",
user, pass);

Statement st = con.createStatement();
ResultSet res = st.executeQuery("SELECT * FROM  inside1event where AssetCost<1 and
Severity=" "
        + " and ReportConfidence=" and Detectability=" and ResponseImpact=" "
        + " order by event_date ");
        ComboBoxModel eventModel;
        DefaultListModel eventModel2 = new DefaultListModel();
//this.ddlEvent.removeAllItems();
while(res.next())
 {
  //  eventModel2.addElement(res.getString("Event_ID"))
 //   this.ddlEvent.addItem("[" + res.getString("Event_ID") + ", " + res.getString("Service") +
   //         ", " + res.getString("Event_Date") +
     //       ", " + res.getString("src_IPAddress") + ", " + res.getString("Dest_IPAddress") +
       //      ", " + res.getString("Event_Name"));
    System.out.println(res.getString("Event_ID"));
 }

//this.ddlEvent.setm(eventModel);

      }
      catch(Exception MyError)
      {

      }
    }
     private void loadAsset()
    {
      try
      {
          Class.forName(driver);
    Connection con = DriverManager.getConnection("jdbc:mysql://localhost:3306/oriolajade",
user, pass);

Statement st = con.createStatement();
ResultSet res = st.executeQuery("SELECT DISTINCT Dest_IPAddress FROM  inside1event" );
      // ComboBoxModel eventModel;
        DefaultListModel AssetModel = new DefaultListModel();

while(res.next())
 {
  //  eventModel2.addElement(res.getString("Event_ID"))
   AssetModel.addElement(res.getString("Dest_IPAddress"));
 }
//this.lstAsset.setModel(AssetModel);
//this.ddlEvent.setm(eventModel);

      }
      catch(Exception MyError)
      {

      }
   }

    private void btnStartActionPerformed(java.awt.event.ActionEvent evt) {
        // TODO add your handling code here:
//        int interval=0;
        try
        {
            //////////////// retrieve interval
  //          interval = 60 * Integer.parseInt(this.txtInterval.getText());
             //////////////////


            int h,m,s, totalTime;

/*                            String driver = "com.mysql.jdbc.Driver";
    String user = "java";
    String pass = "java";
    String y="";

      // Step 1: Load the JDBC driver.
    Class.forName(driver);
    Connection con = DriverManager.getConnection("jdbc:mysql://localhost:3306/oriolajade",
user, pass);
 *
 */
Class.forName(driver);
    Connection con = DriverManager.getConnection("jdbc:mysql://localhost:3306/oris2", user,
pass);

    Statement st = con.createStatement();
    Statement st2 = con.createStatement();

    Statement stS = con.createStatement();
    Statement stSC = con.createStatement();

    Statement stD = con.createStatement();
    Statement stDC = con.createStatement();

    Statement stSeq = con.createStatement();
    Statement stSeqC = con.createStatement();

    //st.executeUpdate("insert into maths(sn,studentid,score) value(1,'ST001',37)");

    //int val=st.executeUpdate("insert into personaldata(matno,name,age, address, lga, sex)
value('tr/006' ,'ine' , 34, 'oluku', 'egor', 'm')");
//int val = st.executeUpdate("INSERT into arm(arm,class) VALUES('D','')");
ResultSet res = st.executeQuery("SELECT * FROM  inside1event4  order by event_date ");
 nrows = st2.executeQuery("SELECT COUNT(*) as count FROM  inside1event4   ");

ResultSet sourceID = stS.executeQuery("SELECT DISTINCT src_IPAddress FROM
inside1event4   order by event_date   ");
ResultSet sourceIDC = stSC.executeQuery("SELECT COUNT(DISTINCT src_IPAddress) as
count FROM  inside1event4  ");

ResultSet destID = stD.executeQuery("SELECT DISTINCT Dest_IPAddress FROM
inside1event4   order by event_date   ");
ResultSet destIDC = stDC.executeQuery("SELECT  COUNT(DISTINCT Dest_IPAddress) as
count FROM  inside1event4  ");
```

```
//ResultSet SeqID = stSeq.executeQuery("SELECT SequenceID FROM  inside1event ");
//ResultSet SeqIDC = stSeqC.executeQuery("SELECT  COUNT(SequenceID) as count FROM 
inside1event ");
nrows.next();
sourceIDC.next();
destIDC.next();
//SeqIDC.next();
//SeqIDC.next();

//System.out.println(nrows.getInt("count"));

//////////////
/*
            for(int k=0; k<m; k++)
            {
                int Step_Category=0;

               addInterval=dTime[k] + Interval;

//System.out.println(addInterval);
retSeq="";
seqComb1=""; /// Sequence Combination
seqComb2=""; /// Sequence Combination

 //System.out.println(dTime[k] + " " + addInterval + " " + eventID[k]);
  //System.out.println(addInterval);
  // System.out.println(eventID[k]);
 g=0;
int n=k;

 int Ante_Check = 0; /// for the Antecedent
  int Ante_Value = 0;

            for(int j=0; j<x; j++)
            {

                int Col_Category = 0; ///  for the Column Category
                Col_Category = j;
//System.out.println(".................");
//System.out.println(x + "....");
//break;

                /////// Determine the antecedent

System.out.println("........" +j + ".........");
                //////
                if(dTime[j + g]>addInterval)
                {
                   // System.out.println("stop");

                 break;
                       //       j=x;
                }
                else
                {
                seqComb1 = eventID[n];
                seqComb2 = eventID[n];
```

```
        retSeq=retSeq + eventID[n] + " ";
       Step_Category = Step_Category + 1;


    //(Seq.get_SequenceID(Seq.lastKey)==Seq.lastKey)
      if ((Seq.get_SequenceID(Seq.lastKey)==Seq.lastKey) && (Seq.lastKey!=0))
        {
         // Seq.Update_Step_Category(Step_Category, eventID[n], Col_Category, Ante_Value
);
        }

      else
        {
         //    Seq.Set_Step_Category(Step_Category, eventID[n], Col_Category, Ante_Value  );
// first tblbse sequence

        }

        }

        retSeqList.add(w);
retSeqList.set(w, retSeq);

//System.out.println("==== " + " - " + w + " - " + retSeqList.get(w));

//System.out.println(w);

        n++;
        g++;
        w++;

      }////second for


  //          this.GeneratedSeq[k]=retSeq;
   //         System.out.println(sn++ + "  " + this.GeneratedSeq[k]);
         //System.out.println(sn++ + "  " + retSeq);
         this.ConfiCapture[n]=retSeq;


         SupportList.add(sn);
         SupportList.set(sn, retSeq);
         //System.out.println(retSeqList.get(sn));
System.out.println("err " + sn + " " +  SupportList.get(sn) );
System.out.println("***********************");

sn++;
x--;

Seq.lastKey=0;

}/// first for

//System.out.print(retSeqList.get(6));


//System.out.println(SupportList.get(4));

String val="";
String val2="";

System.out.println("///////////   Support Start  //////////////");

for(int s=0; s < retSeqList.size(); s++)
  {
  val2 = retSeqList.get(s).toString();

int intIndexD=0;
//System.out.print(retSeqList.get(s));

        for(int t=0; t < SupportList.size(); t++)
        {
        //    System.out.println("////////////////////////////////////////");
          //System.out.println(iter.next().toString());
          //System.out.println(SupportList.get(4).toString());

        val = SupportList.get(t).toString();

         //  intIndex=iter.next().toString().indexOf(SupportList.get(4).toString());
        int intIndex = val2.indexOf(val);
        if(intIndex== -1)
        {
          intIndexD = intIndexD + 1;
         SupportValue.add(s);
         SupportValue.set(s, intIndexD);

        }
        //intIndex= intIndex + 1;

        }
System.out.print(retSeqList.get(s) + " = ");
System.out.print(SupportValue.get(s)  + " / ");
System.out.print(dTime.length  + " : ");
double ans =0;
ans = Double.parseDouble(SupportValue.get(s).toString()) / dTime.length ;

System.out.println(ans);
}
System.out.println("///////////   Support End  //////////////");

System.out.println("///////////   Confidence Start  //////////////");
//System.out.println(SupportList.size());
//System.out.println(retSeqList.size());
for(int s=0; s < SupportList.size(); s++)
  {

  String Confido = SupportList.get(s).toString();
//System.out.println(Confido);
//System.out.println(retSeqList.size());

      for(int t=0; t < retSeqList.size(); t++)
      {
        int count =t;
          String Confido2 = retSeqList.get(t).toString();
         // System.out.println(Confido2);
          if(Confido.equals(Confido2))
          {
            if(count==0)
```

```
                  {

                  }
                  else
                  {
                     count--;
                  }
                  //System.out.println(SupportValue.get(count));
               }
         }
   }
System.out.println("//////////   Confidence End  /////////////");

System.out.println("//////////  Actual Confidence Start  /////////////");
//System.out.print(retSeqList.get(s) + " = ");
//System.out.println(SupportValue.get(s));
for(int s=0; s < retSeqList.size(); s++)
   {
   int count = s;
   if(count==0)
                  {

                  }
                  else
                  {
                     count--;
                  }

   int Confido =Integer.parseInt((SupportValue.get(count).toString()));
System.out.println(retSeqList.get(s) + " = " + SupportValue.get(s) + " / " + Confido);
   }
System.out.println("//////////  Actual Confidence End  /////////////");
//////////////////////////////////
 //System.out.println(dTime[921]);

//System.out.println(SupportList.get(5));
//System.out.println("==== " + retSeqList.get(5));
////////////////......ANTECEDENT AND CONSEQUENT
  Unsorted_Step Step = new Unsorted_Step();
     Step.ant_cons();
**/
      }
      catch(Exception MyError)
      {
         System.out.println(" i have an error which says Sequence");
         System.err.println(MyError.getMessage());
      }


   }

   private void jButton3ActionPerformed(java.awt.event.ActionEvent evt) {
      // TODO add your handling code here:

      try
      {
         //////////////// retrieve interval
 //         interval = 60 * Integer.parseInt(this.txtInterval.getText());
          //////////////////
          int h,m,s, totalTime;

/*                        String driver = "com.mysql.jdbc.Driver";
    String user = "java";
    String pass = "java";
    String y="";

      // Step 1: Load the JDBC driver.
    Class.forName(driver);
    Connection con = DriverManager.getConnection("jdbc:mysql://localhost:3306/oriolajade",
user, pass);
 *
 */
          Class.forName(driver);
    Connection con = DriverManager.getConnection("jdbc:mysql://localhost:3306/oriolajade",
user, pass);

    Statement st = con.createStatement();
    Statement st2 = con.createStatement();

    Statement stS = con.createStatement();
    Statement stSC = con.createStatement();

    Statement stD = con.createStatement();
    Statement stDC = con.createStatement();

    Statement stSeq = con.createStatement();
    Statement stSeqC = con.createStatement();

    //st.executeUpdate("insert into maths(sn,studentid,score) value(1,'ST001',37)");

    //int val=st.executeUpdate("insert into personaldata(matno,name,age, address, lga, sex)
value('tr/006' ,'ine' , 34, 'oluku', 'egor', 'm')");
//int val = st.executeUpdate("INSERT into arm(arm,class) VALUES('D','')");
ResultSet res = st.executeQuery("SELECT * FROM  inside1event order by event_date ");
ResultSet nrows = st2.executeQuery("SELECT COUNT(*) as count FROM  inside1event ");

ResultSet sourceID = stS.executeQuery("SELECT DISTINCT src_IPAddress FROM
inside1event  order by event_date  ");
ResultSet sourceIDC = stSC.executeQuery("SELECT COUNT(DISTINCT src_IPAddress) as
count FROM  inside1event  ");

ResultSet destID = stD.executeQuery("SELECT DISTINCT Dest_IPAddress FROM
inside1event  order by event_date  ");
ResultSet destIDC = stDC.executeQuery("SELECT  COUNT(DISTINCT Dest_IPAddress) as
count FROM  inside1event ");

//ResultSet SeqID = stSeq.executeQuery("SELECT SequenceID FROM  inside1event ");
//ResultSet SeqIDC = stSeqC.executeQuery("SELECT  COUNT(SequenceID) as count FROM
inside1event ");

nrows.next();
sourceIDC.next();
destIDC.next();
//SeqIDC.next();
//SeqIDC.next();

//System.out.println(nrows.getInt("count"));
```

```
dTime= new int[nrows.getInt("count")];
 source=new String[sourceIDC.getInt("count")];
 dest=new String[destIDC.getInt("count")];
 sequenceID=new String[nrows.getInt("count")];
 eventID=new String[nrows.getInt("count")];
 sourceIP=new String[nrows.getInt("count")];
 destIP=new String[nrows.getInt("count")];


//System.out.println(sourceIDC.getInt("count"));
//System.out.println(destIDC.getInt("count"));
for(int counter=0; counter<source.length; counter++)
{
   sourceID.next();
   source[counter]=sourceID.getString("src_IPAddress");
System.out.println(source[counter]);
}
System.out.println("/////////////////////");

for(int counter=0; counter<dest.length; counter++)
{
   destID.next();
  dest[counter]=destID.getString("Dest_IPAddress");
//System.out.println(dest[counter]);
}
for(int counter=0; counter<dTime.length; counter++)
{
res.next();
y=res.getString("Event_Date");
h=Integer.parseInt(y.substring(0, 2)) * 3600;
m=Integer.parseInt(y.substring(3, 5))*60;
s=Integer.parseInt(y.substring(6, 8));
totalTime=h+m+s;

dTime[counter]=totalTime;
//sequenceID[counter]=res.getString("SequenceID");

sequenceID[counter]="[" + (counter + 1)  + "] ";

sourceIP[counter]=res.getString("src_IPAddress");
destIP[counter]=res.getString("Dest_IPAddress");

//System.out.println(dTime[counter]);
//System.out.println(sequenceID[counter]);

}
//////////////// EVENT ID

for(int counter=0; counter<eventID.length; counter++)
{
   int srcN, destN;
   srcN=0;
   destN=0;
      for(int k=0; k<source.length; k++)
      {
         if(sourceIP[counter].equals(source[k]))
         {
            eventID[counter]=sequenceID[counter] + String.valueOf(k) + ",";
            k=source.length;
      }
      else
      {
//          eventID[counter]="no" +String.valueOf(k);
      }
        //eventID[counter]=String.valueOf(counter);
    }

    for(int j=0; j<dest.length; j++)
    {
      if(destIP[counter].equals(dest[j]))
      {
        eventID[counter]=eventID[counter] + String.valueOf(j);
        j=dest.length;
      }
      else
      {
        // j=source.length;
//          eventID[counter]="no" +String.valueOf(k);
      }
        //eventID[counter]=String.valueOf(counter);
    }

System.out.println(eventID[counter]);

}
   //int i = res.getString("Arm");
 //this.jTextField3.setText(res.getString("Sex"));
 //this.jTextField2.setText(res.getString("EmployeeID"));

  /*
  nrows.next();
int sam=nrows.getInt("count");
System.out.println(sam);


//ResultSetMetaData rsM=res.getMetaData();

int dTime[]=new int[sam];
  //System.out.println(res.getTime("Event_Date"));

//while (res.next())
  for(int counter=0; counter < dTime.length; counter++)
  {
    /*res.next();
  /* y=res.getTime("Event_Date").toString();
//for(int counter=0; counter < dTime.length; counter++)
//{

//}
//System.out.println(sam);
//System.out.println(y);
h=Integer.parseInt(y.substring(0, 2)) * 3600;
m=Integer.parseInt(y.substring(3, 5))*60;
s=Integer.parseInt(y.substring(6, 8));
totalTime=h+m+s;
//System.out.println(h);
//System.out.println(m);
//System.out.println(s);
```

```
System.out.println(totalTime);


     //int i = res.getString("Arm");

//this.jTextField3.setText(res.getString("Sex"));
//this.jTextField2.setText(res.getString("EmployeeID"));

    */
       }
       catch(Exception MyError)
       {
          System.out.println("i have an error which says");
          System.err.println(MyError.getMessage());
       }
    }

    private void btnGSequence1ActionPerformed(java.awt.event.ActionEvent evt) {
       // TODO add your handling code here:
    }

    private void btn_StepActionPerformed(java.awt.event.ActionEvent evt) {
       // TODO add your handling code here:
       Unsorted_Step Step = new Unsorted_Step();
     // Step.Display_usorted_Step();
      Step.Display_usorted_Step();
      Step.Display_sorted_Step();


    }

    private void btn_SupportActionPerformed(java.awt.event.ActionEvent evt) {
       // TODO add your handling code here:
       Supp_Conf Support = new Supp_Conf();
      Output += " \n \n" +  Support.Support();
    }

    private void btnLoadCSVActionPerformed(java.awt.event.ActionEvent evt) {
     FileFilter ft = new FileNameExtensionFilter("CSV", "csv");
       fchooser.addChoosableFileFilter( ft );
      int result = fchooser.showOpenDialog( mainPanel);

          if(result == JFileChooser.APPROVE_OPTION) {
         java.io.File file = fchooser.getSelectedFile( );
          String file_name = file.toString( );
        // System.out.println("xx");
          CSVLoader2 obj = new CSVLoader2();
          obj.run(file_name);
        JOptionPane.showMessageDialog(mainPanel, file_name + " was uploaded successfully");
        }          // TODO add your handling code here:

    }

    private void btnWriterActionPerformed(java.awt.event.ActionEvent evt) {
       // TODO add your handling code here:
Writer writer = null;

       try {
```

```
      String text = "";


      File file = new File("D:/writex.txt");
      writer = new BufferedWriter(new FileWriter(file));
      writer.write(text);

      try
  {
    String path = "D:/writex.txt";

    File filex = new File(path);

    FileWriter fileWriter = new FileWriter(filex,true);

    BufferedWriter bufferFileWriter  = new BufferedWriter(fileWriter);
text +=Output;
//text +="- \n \nt3Sample text in the file to Preappend 2";
    fileWriter.append(text);

//fileWriter.append();

    bufferFileWriter.close();

    System.out.println("User Registration Completed");

  }catch(Exception ex)
  {
    System.out.println(ex);
  }

    } catch (FileNotFoundException e) {
      e.printStackTrace();
    } catch (IOException e) {
      e.printStackTrace();
    } finally {
      try {
        if (writer != null) {
          writer.close();
        }
      } catch (IOException e) {
        e.printStackTrace();
      }
    }
  }

  // Variables declaration - do not modify
  private javax.swing.JButton btnGSequence;
  private javax.swing.JButton btnGSequence1;
  private javax.swing.JButton btnLoadCSV;
  private javax.swing.JButton btnStart;
  private javax.swing.JButton btnWriter;
  private javax.swing.JButton btn_Step;
  private javax.swing.JButton btn_Support;
  private javax.swing.JFileChooser fchooser;
  private javax.swing.JButton jButton3;
  private javax.swing.JFileChooser jFileChooser1;
  private javax.swing.JLabel jLabel1;
  private javax.swing.JPanel mainPanel;
```

```
    private javax.swing.JMenuBar menuBar;
    private javax.swing.JProgressBar progressBar;
    private javax.swing.JLabel statusAnimationLabel;
    private javax.swing.JLabel statusMessageLabel;
    private javax.swing.JPanel statusPanel;
    private javax.swing.JTextField txtInterval;
    // End of variables declaration

    private final Timer messageTimer;
    private final Timer busyIconTimer;
    private final Icon idleIcon;
    private final Icon[] busyIcons = new Icon[15];
    private int busyIconIndex = 0;

    private JDialog aboutBox;
}
```

Sequence_Class

```
/*
 * To change this template, choose Tools | Templates
 * and open the template in the editor.
 */

package oriolajade;
import java.sql.*;
import java.util.*;

public class Sequence_Class {
     int lastKey = 0;
       String driver = "com.mysql.jdbc.Driver";

   String user = "java";
    String pass = "java";

    public int Set_Step_Category (int Cat, String event, int Col_Category, int Ante_Value)
    {
        int base = 0;

        try
        {

      Class.forName(driver);
   Connection con = DriverManager.getConnection("jdbc:mysql://localhost:3306/oris2", user,
pass);
    Statement st = con.createStatement();
 int val= st.executeUpdate("insert into tblbse_sequence(step_Category) value(" + Cat + ")",
                          Statement.RETURN_GENERATED_KEYS);
 ResultSet keys=st.getGeneratedKeys();
 while (keys.next())
 {
    lastKey = keys.getInt(1);
 }
 Set_Event(event,  lastKey,  Ante_Value, Col_Category);
        }
        catch(Exception MyError)
        {
            System.out.println("Set_Step_Category error which says");
            System.err.println(MyError.getMessage());
        }
        return base;

    }
public int Del_Step_Category ()
    {
        int base = 0;

        try
        {

      Class.forName(driver);
 Connection con = DriverManager.getConnection("jdbc:mysql://localhost:3306/oris2", user, pass);
 Statement st = con.createStatement();
 //int val= st.executeUpdate("delete from tblbse_sequence");
 int val2= st.executeUpdate("delete from tblvar_sequence");
 int val3= st.executeUpdate("delete from ant_cons");
        }
        catch(Exception MyError)
        {
            System.out.println("Set_Step_Category error which says");
            System.err.println(MyError.getMessage());
        }

        return base;
    }

     public int Update_Step_Category (int Cat, String event, int Col_Category, int Ante_Value)
    {
        int base = 0;

        try
        {
      Class.forName(driver);
   Connection con = DriverManager.getConnection("jdbc:mysql://localhost:3306/oris2", user,
pass);
    Statement st = con.createStatement();
 int val= st.executeUpdate("update tblbse_sequence set step_Category =" + Cat + "
where(Sequence_ID=" + lastKey + ")");
 Set_Event(event,  lastKey, Ante_Value, Col_Category);
        }
        catch(Exception MyError)
        {
            System.out.println("Update_Step_Category  error which says");
            System.err.println(MyError.getMessage());
        }
        return base;

    }
     public int Set_Event (String event,int ID,int Ante_Cons, int Col_Category)
    {
        int base = 0;

        try
        {

      Class.forName(driver);
    Connection con = DriverManager.getConnection("jdbc:mysql://localhost:3306/oris2", user,
pass);
    Statement st = con.createStatement();
 int val= st.executeUpdate("insert into tblvar_sequence(Sequence_Name, Sequence_ID,
Ante_Cons_Category, Column_Category) value('" +
         event + "', " + ID + ", " + Ante_Cons + ", " + Col_Category + ")");
        }
        catch(Exception MyError)
        {
            System.out.println("Set_Event  error which says");
            System.err.println(MyError.getMessage());
        }
        return base;

    }
public int get_SequenceID (int seq_ID)
    {

int ID=0;
        try
        {

  Class.forName(driver);
    Connection con = DriverManager.getConnection("jdbc:mysql://localhost:3306/oris2", user,
pass);
    Statement st = con.createStatement();

  ResultSet res = st.executeQuery("select Sequence_ID  from tblbse_sequence
where(Sequence_ID=" + seq_ID + ")");
  if (res.next())
          {
ID=res.getInt("Sequence_ID");
          }
         else
          {

          }
```

Supp_Conf.java

```
/*
 * To change this template, choose Tools | Templates
 * and open the template in the editor.
 */

package oriolajade;
import java.sql.*;
import java.util.*;
import java.text.DecimalFormat;

public class Supp_Conf {
    String driver = "com.mysql.jdbc.Driver";
   String user = "java";
    String pass = "java";
    String Output = "";

     public String Support ()
    {
        try
        {
        System.out.println("............... Support.............");
     Output += " \n  \n" + "............... Support.............";

    Class.forName(driver);
  Connection con = DriverManager.getConnection("jdbc:mysql://localhost:3306/oris2", user,
pass);
   Statement st2 = con.createStatement();
ResultSet nrows = st2.executeQuery("SELECT COUNT(*) as count FROM  inside1event4 ");

nrows.next();
int No_of_Sequence = nrows.getInt("count");


  //////////// Print the Step Headings
   Statement step_Count = con.createStatement();
ResultSet res_Count = step_Count.executeQuery("SELECT DISTINCT Step  FROM
tblbse_antcons"); /// Select the total steps
    int num_of_places=0;
  while(res_Count.next())
  {

     int step_to_Count = res_Count.getInt("Step");
      if((step_to_Count ) > 1) {
 System.out.println("........." + (step_to_Count - 1 )+ "............");
 Output += " \n \n" + "........." + (step_to_Count  -1)+ "............";
      Statement step = con.createStatement();
      ResultSet res_Step = step.executeQuery("SELECT * from tblbse_antcons " +
   " where  Step=" + step_to_Count );
String myVal="";
            while(res_Step.next())
        {
              try {
            //     System.out.println(res_Step.getString("Ant_Cons_ID"));

                String val_To_Check ="";
                     val_To_Check = res_Step.getString("Antecedent") ;
                     String val_To_Be_Check ="";
                     //myVal  = res_Step.getString("Ante").substring(9,
res_Step.getString("Ante").length());

 //System.out.println( val_To_Check);
                Statement check = con.createStatement();
                ResultSet res_check = check.executeQuery("SELECT * from tblbse_antcons " +
                              " where  Step=" + step_to_Count );
                    int n_Times=0;

                    while(res_check.next())
                      {
                       try{
                       val_To_Be_Check = res_check.getString("Antecedent") ;


                        int intIndex = val_To_Check.indexOf(val_To_Be_Check);
                        // if(intIndex== -1)
                      //  statusMessageLabel.setText((text == null) ? "" : text);
                       if(intIndex== -1)
                         {

                         }
                         else
```

```
                              {
                                 n_Times++;
                              }

                        } catch (Exception MyError) {
                           System.out.println("sorted  error which says x");
                           System.err.println(MyError.getMessage());
                        }

                        }
                     System.out.println("");
//val_To_Check +
                        System.out.println(val_To_Check +  " was found "  + n_Times + " times:
Support is = "  + (double)n_Times/No_of_Sequence);
                        Output += " \n \n" + val_To_Check +  " was found "  + n_Times + " times:
Support is = "  + (double)n_Times/No_of_Sequence;
                        // + (double)n_Times/No_of_Sequence
                     } catch (Exception MyError) {
                           System.out.println("sorted  error which says nex");
                           System.err.println(MyError.getMessage());

                     }
         }
      System.out.println(".....................");
         }
  }
System.out.println("............... Support.............");
Output += " \n \n" + "............... Support.............";

Confidence();

}
      catch(Exception MyError)
      {
         System.out.println("sorted  error which says");
         System.err.println(MyError.getMessage());
      }
return Output;

   }
     public void Confidence ()
   {
      try
      {
      System.out.println("............... Confidence.............");
      Output += " \n \n" + "............... Confidence.............";

   Class.forName(driver);
   Connection con = DriverManager.getConnection("jdbc:mysql://localhost:3306/oris2", user,
pass);
   Statement st2 = con.createStatement();
ResultSet nrows = st2.executeQuery("SELECT COUNT(*) as count FROM  inside1event4 ");

nrows.next();
int No_of_Sequence = nrows.getInt("count");

  //////////// Print the Step Headings
   Statement step_Count = con.createStatement();
```

```
ResultSet res_Count = step_Count.executeQuery("SELECT DISTINCT Step  FROM
tblbse_antcons"); /// Select the total steps
      int num_of_places=0;
   while(res_Count.next())
   {
         int step_to_Count = res_Count.getInt("Step");
        if((step_to_Count ) > 1) {

 System.out.println("........." + (step_to_Count - 1)+ "...........");
 Output += " \n \n" + "........." + (step_to_Count - 1)+ "...........";

         Statement step = con.createStatement();
         ResultSet res_Step = step.executeQuery("SELECT * from tblbse_antcons " +
                                   " where  Step=" + step_to_Count );

                 while(res_Step.next())
            {
                 //    System.out.println("........ Step...........");
                       String val_To_Check ="";
                       String val_To_Check_for_Confidence ="";

                               val_To_Check = res_Step.getString("Consequent") ;
                               val_To_Check_for_Confidence= res_Step.getString("Antecedent");

                               String val_To_Be_Check ="";
                               String val_To_Be_Check_for_Confidence ="";

                       Statement check = con.createStatement();
                       ResultSet res_check = check.executeQuery("SELECT * from tblbse_antcons " +
                                   " where  Step=" + step_to_Count );
                             int n_Times=0;
                             int n_Times_Confidence=0;

                           while(res_check.next())
                                {
                                  try{
                                  val_To_Be_Check = res_check.getString("Consequent") ;
                                  val_To_Be_Check_for_Confidence =res_check.getString("Antecedent") ;

                                   int intIndex = val_To_Check.indexOf(val_To_Be_Check);
                                   int intIndex2 =
val_To_Check_for_Confidence.indexOf(val_To_Be_Check_for_Confidence);
                                   // if(intIndex== -1)
                                 //  statusMessageLabel.setText((text == null) ? "" : text);
                                  if(String.valueOf(intIndex).equals(String.valueOf( -1)))
                                    {

                                    } else {
                                           n_Times++;
                                    }
                                    if(String.valueOf(intIndex2).equals(String.valueOf( -1)))
                                    {

                                    } else {
                                           n_Times_Confidence++;
                                    }

                                } catch (Exception MyError) {
                                    System.out.println("sorted  error which says x");
```

```
                              System.err.println(MyError.getMessage());
                            }
                            }
                          //  System.out.print("Support [----" + res_Step.getString("Ante") + " was found
"  +
                            //        n_Times + " times: Support is = " + (double)n_Times/No_of_Sequence
+ " -----]  ");

 System.out.print("Confidence [-----" + val_To_Check_for_Confidence + " was found "  +
                                    n_Times_Confidence + " times:         while         " + val_To_Check  + "
was found " + n_Times + ": Confidence is = " + (double)n_Times/n_Times_Confidence  + " -----]
");

 Output += " \n \n" + "Confidence [-----" + val_To_Check_for_Confidence + " was found "  +
                                    n_Times_Confidence + " times:         while         " + val_To_Check  + "
was found " + n_Times + ": Confidence is = " + (double)n_Times/n_Times_Confidence  + " -----]
";

 System.out.println();
           }
/////////////  /denomiator //////////////////////


/////////////// DENOMINATOR ///////////////

       System.out.println(".....................");
           }
   }

//Success();
//Exp_Success();
System.out.println("............... Confidence..............");
Output += " \n" + "............... Confidence.............";

}
       catch(Exception MyError)
       {
          System.out.println("sorted  error which says");
          System.err.println(MyError.getMessage());
       }
    }
     //////////////// CONFIDENCE

          public void Success ()
    {
       try
       {
       System.out.println("............... Success..............");

     Class.forName(driver);
   Connection con = DriverManager.getConnection("jdbc:mysql://localhost:3306/oris2", user,
pass);
    Statement st2 = con.createStatement();
ResultSet nrows = st2.executeQuery("SELECT COUNT(*) as count FROM  inside1event4   ");

nrows.next();
int No_of_Sequence = nrows.getInt("count")
   //////////// Print the Step Headings
    Statement step_Count = con.createStatement();
```

```
ResultSet res_Count = step_Count.executeQuery("SELECT DISTINCT Column_Category
FROM  tblvar_sequence"); /// Select the total steps
     int num_of_places=0;
   while(res_Count.next())
   {
       int step_to_Count = res_Count.getInt("Column_Category");
 System.out.println("........." + (step_to_Count + 1)+ "............");
        Statement seq_ID = con.createStatement();
        ResultSet res_seq_ID = seq_ID.executeQuery("SELECT * FROM  tblvar_sequence where
column_category=" + step_to_Count);
                while(res_seq_ID.next())
           {
                    int Step = step_to_Count + 1;
                    int Min =1;
                    int Max =0;
                    int ret_seq_ID = res_seq_ID.getInt("Sequence_ID");
                    String Event = "";
                    Event = res_seq_ID.getString("Sequence_Name");

            Statement seq_Max = con.createStatement();
        ResultSet res_Max = seq_Max.executeQuery("SELECT Max(Column_Category) as
Max_Seq FROM  "
                + "tblvar_sequence where sequence_ID=" + ret_seq_ID);
        res_Max.next();
        Max = res_Max.getInt("Max_Seq") + 1;
double Success =0;
int Numerator=0;
       int Denominator = 0;
       Numerator=Step - Min;
Denominator=Max - Min;

Success =(double) Numerator / Denominator;
System.out.println(Event + " - Step: " + Step + " Min: " + Min + " Max: " + Max + " Normalized
Success: =" +
       Success + " while SUCCESS: " + ((Success * 2) + 1));
          }
////////////  /denomiator /////////////////////


////////////// DENOMINATOR //////////////

       System.out.println(".....................");
   }
System.out.println("............... success..............");
}
       catch(Exception MyError)
       {
           System.out.println("sorted  error which says");
           System.err.println(MyError.getMessage());
       }
    }
```

Appendix 2: Java Server Page (JSP) Code for Threat Prioritisation Tool Implementation

Frm_AssetThreat.jsp

```
<%--
   Document   : frm_AssetThreat.jsp
   Created on : Aug 19, 2013, 3:14:04 PM
   Author     : ORIOLA
--%>

<%@page import="dClasses.Database_Object" contentType="text/html" pageEncoding="UTF-8"%>
<%@page  import="dClasses.Database_Object"%>
<%@page  import="java.util.*" %>
<%@page  import="java.sql.*" %>
<%@include  file="head.jsp" %>

<% Database_Object database = new Database_Object(); %>
<%

double m=0.0, n=0.0;

   /*    jSql =   "select  * FROM tblbse_administrator";
         ResultSet get_Admin = database.Find_Record_tablejoin(jSql);
         while(get_Admin.next()) {
            int AdministratorID = get_Admin.getInt("Administrator_ID");

        String       jSql =   "select  Threat_ID FROM tbllkup_Threats";
         ResultSet get_Threat = database.Find_Record_tablejoin(jSql);
         while(get_Threat.next()) {
            int ThreatID = get_Threat.getInt("Threat_ID");
            ///////
            int counter = 1;
           jSql =   "select  * FROM  tblbse_admin_perspective "
                 + "where  Threat_ID =" + ThreatID ;
               ResultSet get_Perspective = database.Find_Record_tablejoin(jSql);
               while(get_Perspective.next()) {
                  int Exp = get_Perspective.getInt("Administrator_ID");
                        if(m==0) {
                          //m = get_Perspective.getDouble("Exploit_Success_Percent");

                        }
                        if(counter==1) {
                            m = get_Perspective.getDouble("Exploit_Success_Percent");
                        } else {
                            n = get_Perspective.getDouble("Exploit_Success_Percent");
                        }
counter += 1;
                  //out.print(get_Perspective.getDouble("Exploit_Success_Percent") + " - ");


               }

          //     out.println( ThreatID + " " + m + " " + n + "<br/>");
            }
              //out.println(m + "<br/>");
```

```
             //out.print( out.print(m + " " + n +
"<br/>");get_Threat.getInt("Administrator_ID"));

      // }


  **/

%>

<%
/// RISK OF EXPOSURE
/*
String jSql =   "select  * FROM tblBse_threat_objective";


ResultSet get_Threat = database.Find_Record_tablejoin(jSql);
       int g =0;
       while(get_Threat.next()) {
         int Threat = get_Threat.getInt("Threat_ID");
         g++;
          double Av_Objective =  0.0  ;
         double Con_Objective =  0.0  ;
         double Int_Objective =  0.0  ;
         double Exp_Objective = 0.0 ;

         Av_Objective = get_Threat.getDouble("Detectability_Objective") *
             get_Threat.getDouble("Remediation_Objective") ;


        Av_Objective = Math.round(1000  * (double)(Av_Objective)) / 1000d ;
         Con_Objective = Math.round(1000  * (double)(Con_Objective)) / 1000d ;
         Int_Objective = Math.round(1000  * (double)(Int_Objective)) / 1000d ;
         Exp_Objective = Math.round(1000  * (double)(Exp_Objective)) / 1000d ;

         out.print(g + ".  Av Imp =" +
             Av_Objective
             + "<br/>" );

        // double Potential_Damage =  Av_Objective +  Con_Objective +  Int_Objective;
          String sql = "insert into tblbse_riskofexposure ";
           sql += "( Threat_ID, riskofexposure) "
               + "  ";
           sql += " Values(" + Threat + ", "   + Av_Objective + ")";

           int Save_AssetThreat = database.insert(sql);

         }
  * */

%>

<%
/*
/// POTENTIAL DAMAGE

String jSql =   "select  tblbse_asset_objective.*, tblbse_assetthreat.*,  "
    + " tblbse_assetthreat_objective.* from  tblbse_asset_objective inner join tblbse_assetthreat"
    + " on tblbse_asset_objective.Asset_ID = tblbse_assetthreat.Asset_ID"
```

```
        + " inner join tblbse_assetthreat_objective"
        + " on tblbse_assetthreat_objective.AssetThreat_ID = tblbse_assetthreat.AssetThreat_ID"
        + " ";


ResultSet get_Threat = database.Find_Record_tablejoin(jSql);
            int g =0;
            while(get_Threat.next()) {
               int AssetThreat = get_Threat.getInt("AssetThreat_ID");
               g++;
                 double Av_Objective =  0.0  ;
               double Con_Objective =  0.0  ;
               double Int_Objective =  0.0  ;
               double Exp_Objective = 0.0 ;

               Av_Objective = get_Threat.getDouble("Availability_Impact_Objective") *
                      get_Threat.getDouble("Availability_Significance_Objective") ;
               Con_Objective = get_Threat.getDouble("Confidentiality_Impact_Objective") *
                      get_Threat.getDouble("Confidentiality_Significance_Objective");
               Int_Objective = get_Threat.getDouble("Integrity_Impact_Objective") *
                      get_Threat.getDouble("Integrity_Significance_Objective") ;

             Av_Objective = Math.round(1000  * (double)(Av_Objective)) / 1000d ;
               Con_Objective = Math.round(1000  * (double)(Con_Objective)) / 1000d ;
               Int_Objective = Math.round(1000  * (double)(Int_Objective)) / 1000d ;
               Exp_Objective = Math.round(1000  * (double)(Exp_Objective)) / 1000d ;

               out.print(g + ".  Av Imp =" + get_Threat.getString("Availability_Impact_Objective") + "
*  " +
                      get_Threat.getString("Availability_Significance_Objective") + " = " +
                      Av_Objective
                      + "<br/>" );

               double Potential_Damage =  Av_Objective +  Con_Objective +  Int_Objective;
                String sql = "insert into tblbse_Potential_Damage ";
                   sql += "( AssetThreat_ID, Potential_Damage) "
                          + "  ";
                   sql += " Values(" + AssetThreat + ", "   + Potential_Damage + ")";

                  // int Save_AssetThreat = database.insert(sql);

               }
 * */

%>

<%

// OBJECTIVITY SCORE ASSET
/*
//out.println(dSupport);

 // assetthreat certainty
  //database.tbl_Name = "tblbse_assets";
  String jSql = "SELECT tbllkup_threats.*, tblbse_threat_perception_certainty.* " +
               " FROM tblbse_threat_perception_certainty inner join tbllkup_threats" +
                " on tblbse_threat_perception_certainty.threat_ID = tbllkup_threats.threat_ID"
                + " where tblbse_threat_perception_certainty.threat_ID = tbllkup_threats.threat_ID";
            ResultSet get_Threat = database.Find_Record_tablejoin(jSql);
            int g =0;
            while(get_Threat.next()) {
               //out.print(get_Admin.getInt("Admin_ID") + " " + get_Admin.getString("Name") +
"............<br/>");
               int Threat = get_Threat.getInt("Threat_ID");
               double Av_degree =0.0;
               double Con_degree =0.0;
               double Int_degree =0.0;
               double Exp_degree =0.0;

               Av_degree = get_Threat.getDouble("Detectability_Degree");
               Con_degree = get_Threat.getDouble("Remediation_Degree");
               out.print(Av_degree);
              // Int_degree = get_Threat.getDouble("Integrity_Significance_Degree");
               //Exp_degree = get_Threat.getDouble("Exploitability_Degree");

               String Av_Impact = "";
               String Con_Impact = "";
               String Int_Impact = "";
               String Exp_Impact = "";
               String Av_Imp ="";

               Av_Impact =
database.Find_Slave_ReferenceByID("tblBse_Slave","Value","Slave_ID",
get_Threat.getInt("Detectability"));
               Con_Impact =
database.Find_Slave_ReferenceByID("tblBse_Slave","Value","Slave_ID",
get_Threat.getInt("Remediation"));
               //Int_Impact =
database.Find_Slave_ReferenceByID("tblBse_Slave","Value","Slave_ID",
get_Threat.getInt("Integrity_Significance"));
              // Exp_Impact =
database.Find_Slave_ReferenceByID("tblBse_Slave","Value","Slave_ID",
get_Threat.getInt("Exploitability"));

               double Av_Objective=0.0;
               double Con_Objective=0.0;
               double Int_Objective=0.0;
               double Exp_Objective=0.0;

               Av_Objective = (Integer.valueOf(Av_Impact) * Av_degree);
               Con_Objective = (Integer.valueOf(Con_Impact) * Con_degree);
               //Int_Objective = (Integer.valueOf(Int_Impact) * Int_degree);
               //Exp_Objective = (Integer.valueOf(Exp_Impact) * Exp_degree) ;

               Av_Objective = Math.round(1000  * (double)(Av_Objective)) / 1000d ;
               Con_Objective = Math.round(1000  * (double)(Con_Objective)) / 1000d ;
               Int_Objective = Math.round(1000  * (double)(Int_Objective)) / 1000d ;
               Exp_Objective = Math.round(1000  * (double)(Exp_Objective)) / 1000d ;

               //Av_Impact +=
database.Find_Slave_ReferenceByID("tblBse_Slave","Display","Slave_ID",
get_Threat.getInt("Availability_Impact"));
                    out.println(Av_Impact + " " + " * " + Av_degree + "  ="
                        + Av_Objective + "<br/>");
//out.print(Threat);

               ///// Get All Threats
               //database.tbl_Name = "tblbse_Threat_Perception";
          String sql = "insert into tblbse_Threat_Objective ";
           sql += "( Threat_ID, Detectability_Objective, Remediation_Objective "
                + " )"
                + "  ";
           sql += " Values(" + Threat + ", "  + Av_Objective + ", "
                + Con_Objective + ")";

int Save_AssetThreat = database.insert(sql);
/*
 /*
          //out.print(sql);

       //     i


        }
      out.println(database.Connection_Error_Msg);
**/

%>

<%

// OBJECTIVITY SCORE ASSET
/*
//out.println(dSupport);

 // assetthreat certainty
  //database.tbl_Name = "tblbse_assets";
  String jSql = "SELECT tblbse_assets.*, tblbse_asset_perception_certainty.* " +
        " FROM tblbse_asset_perception_certainty inner join tblbse_assets" +
         " on tblbse_asset_perception_certainty.Asset_ID = tblbse_assets.Asset_ID"
         + " where tblbse_asset_perception_certainty.Asset_ID = tblbse_assets.Asset_ID";
      ResultSet get_Threat = database.Find_Record_tablejoin(jSql);
      int g =0;
      while(get_Threat.next()) {
        //out.print(get_Admin.getInt("Admin_ID") + " " + get_Admin.getString("Name") +
"............<br/>");
        int Threat = get_Threat.getInt("Asset_ID");
        double Av_degree =0.0;
        double Con_degree =0.0;
        double Int_degree =0.0;
        double Exp_degree =0.0;

        Av_degree = get_Threat.getDouble("Availability_Significance_Degree");
        Con_degree = get_Threat.getDouble("Confidentiality_Significance_Degree");
        Int_degree = get_Threat.getDouble("Integrity_Significance_Degree");
        //Exp_degree = get_Threat.getDouble("Exploitability_Degree");

        String Av_Impact = "";
        String Con_Impact = "";
        String Int_Impact = "";
        String Exp_Impact = "";
        String Av_Imp ="";
        Av_Impact =
database.Find_Slave_ReferenceByID("tblBse_Slave","Value","Slave_ID",
get_Threat.getInt("Availability_Significance"));
```

```
                Con_Impact =
database.Find_Slave_ReferenceByID("tblBse_Slave","Value","Slave_ID",
get_Threat.getInt("Confidentiality_Significance"));
                Int_Impact =
database.Find_Slave_ReferenceByID("tblBse_Slave","Value","Slave_ID",
get_Threat.getInt("Integrity_Significance"));
               // Exp_Impact =
database.Find_Slave_ReferenceByID("tblBse_Slave","Value","Slave_ID",
get_Threat.getInt("Exploitability"));

                double Av_Objective=0.0;
                double Con_Objective=0.0;
                double Int_Objective=0.0;
                double Exp_Objective=0.0;

                Av_Objective = (Integer.valueOf(Av_Impact) * Av_degree);
                Con_Objective = (Integer.valueOf(Con_Impact) * Con_degree);
                Int_Objective = (Integer.valueOf(Int_Impact) * Int_degree);
                //Exp_Objective = (Integer.valueOf(Exp_Impact) * Exp_degree) ;

                Av_Objective = Math.round(1000  * (double)(Av_Objective)) / 1000d ;
                Con_Objective = Math.round(1000  * (double)(Con_Objective)) / 1000d ;
                Int_Objective = Math.round(1000  * (double)(Int_Objective)) / 1000d ;
                Exp_Objective = Math.round(1000  * (double)(Exp_Objective)) / 1000d ;
                //Av_Impact +=
database.Find_Slave_ReferenceByID("tblBse_Slave","Display","Slave_ID",
get_Threat.getInt("Availability_Impact"));
                                out.println(Con_Impact + " " + " * " + Con_degree + "  ="
                                        + Con_Objective + "<br/>");
//out.print(Threat);
                ///// Get All Threats
                //database.tbl_Name = "tblbse_Threat_Perception";
                   String sql = "insert into tblbse_asset_Objective ";
                    sql += "( Asset_ID, Availability_Significance_Objective,
Confidentiality_Significance_Objective, "
                            + " Integrity_Significance_Objective)"
                            + "  ";
                    sql += " Values(" + Threat + ", "  + Av_Objective + ", "
                            + Con_Objective + ", " + Int_Objective + ")";
int Save_AssetThreat = database.insert(sql);

 /*
                  //out.print(sql);


             //      i



              }
            out.println(database.Connection_Error_Msg);
**/

%>

<%
////Damage

/*

//// Remediation and Detecability
```

```
// OBJECTIVITY SCORE ASSET THREAT

//out.println(dSupport);

 // assetthreat certainty
  database.tbl_Name = "tblbse_assetthreat";
  /*String jSql = "SELECT tblbse_assetthreat.*, tblbse_assetthreat_pereception_certainty.* " +
              " FROM tblbse_assetthreat_pereception_certainty inner join tblbse_assetthreat" +
               " on tblbse_assetthreat_pereception_certainty.AssetThreat_ID =
tblbse_assetthreat.AssetThreat_ID"
               + " where tblbse_assetthreat_pereception_certainty.AssetThreat_ID =
tblbse_assetthreat.AssetThreat_ID";

 String jSql ="SELECT tblbse_asset_objective.Availability_Significance_objective, "
        + "tblbse_asset_objective.Integrity_Significance_objective, "
        + "tblbse_asset_objective.Confidentiality_Significance_objective, "
        + "tblbse_assetthreat_objective.Availability_Impact_Objective, "
        + "tblbse_assetthreat_objective.Confidentiality_Impact_Objective, "
        + "tblbse_assetthreat_objective.Integrity_Impact_Objective, "
        + "tblbse_assets.Asset_ID "
        + "FROM tblbse_asset_objective "
        + "INNER JOIN tblbse_assets "
        + "ON tblbse_asset_objective.Asset_ID = tblbse_assets.Asset_ID "
        + "INNER JOIN tblbse_assetthreat "
        + "ON tblbse_assetthreat.Asset_ID = tblbse_assets.Asset_ID "
        + "INNER JOIN tblbse_assetthreat_objective "
        + "ON tblbse_assetthreat_Objective.AssetThreat_ID = tblbse_assetThreat.AssetThreat_ID
";

           ResultSet get_Threat = database.Find_Record_tablejoin(jSql);
           int g =0;
           while(get_Threat.next()) {
              g+=1;

              //out.print(get_Admin.getInt("Admin_ID") + " " + get_Admin.getString("Name") +
"............<br/>");
              int Asset_ID = get_Threat.getInt("Asset_ID");

             // out.println(Av_degree + "  -  " + Av_Significance + "   =" + Av_Obj_Impact + "   :  " +
Av_Obj_Significance  + "<br/>");
            //   out.println(Av_Impact + "  -  " + Av_Obj_Impact + "   =" + Av_Objective_Impact + "
;   " + Av_Significance_Value + "  -  " + Av_Obj_Significance + "   =" +
Av_Objective_Significance + "<br/>xxxxxxxxxxxx<br/>");
/*
               String  sql ="";
                sql = "insert into tblbse_threat_objective ";
                sql += "( Threat_ID, Remediation_Objective,  "
                      + " Detectability_Objective"     + " ) ";
                sql += " Values(" + Threat_ID + ", "  + Remediation_Objective_Impact + ", "
                       + Detectability_Objective_Impact  +  ")";

//int Save_Asset = database.insert(sql);

    }

**/

////  Damage
```

```
%>

<%
//// Remediation and Detecability

// OBJECTIVITY SCORE ASSET THREAT

//out.println(dSupport);

 // assetthreat certainty
 // database.tbl_Name = "tblbse_assetthreat";
  /*String jSql = "SELECT tblbse_assetthreat.*, tblbse_assetthreat_pereception_certainty.* " +
               " FROM tblbse_assetthreat_pereception_certainty inner join tblbse_assetthreat" +
                " on tblbse_assetthreat_pereception_certainty.AssetThreat_ID =
tblbse_assetthreat.AssetThreat_ID"
                + " where tblbse_assetthreat_pereception_certainty.AssetThreat_ID =
tblbse_assetthreat.AssetThreat_ID";

 String jSql ="SELECT tblbse_threat_perception_certainty.Detectability_Degree, "
        + " tblbse_threat_perception_certainty.Remediation_Degree, "
        + " tbllkup_threats.Remediation, tbllkup_threats.Detectability, tbllkup_threats.Threat_ID  "
        + " from tblbse_threat_perception_certainty inner join tbllkup_threats on "
        + " tblbse_threat_perception_certainty.Threat_ID=tbllkup_threats.Threat_ID ";

            ResultSet get_Threat = database.Find_Record_tablejoin(jSql);
            int g =0;
            while(get_Threat.next()) {
               g+=1;

               //out.print(get_Admin.getInt("Admin_ID") + " " + get_Admin.getString("Name") +
"............<br/>");
               int Threat_ID = get_Threat.getInt("Threat_ID");

               double Detectability_Degree =0.0;
               double Remediation_Degree=0.0;

                 double Detectability_Objective =0.0;
               double Remediation_Objective=0.0;


               Detectability_Degree = get_Threat.getDouble("Detectability_Degree");
               Remediation_Degree = get_Threat.getDouble("Remediation_Degree");

               Detectability_Objective = Detectability_Degree /(Remediation_Degree +
Detectability_Degree);
               Remediation_Objective = Remediation_Degree /(Remediation_Degree +
Detectability_Degree);

               String Remediation_Value = "0";
               String Detectabiility_Value = "0";

            Remediation_Value =
database.Find_Slave_ReferenceByID("tblBse_Slave","Value","Slave_ID",
get_Threat.getInt("Remediation"));
            Detectabiility_Value =
database.Find_Slave_ReferenceByID("tblBse_Slave","Value","Slave_ID",
get_Threat.getInt("Detectability"));
```

```
               double Remediation_Objective_Impact =0.0;
               double Detectability_Objective_Impact =0.0;

               Remediation_Objective_Impact = (Integer.valueOf(Remediation_Value) *
Remediation_Objective);
               Detectability_Objective_Impact = (Integer.valueOf(Detectabiility_Value) *
Detectability_Objective);

                Remediation_Objective_Impact = Math.round(1000  *
(double)(Remediation_Objective_Impact)) / 1000d ;
              Detectability_Objective_Impact = Math.round(1000  *
(double)(Detectability_Objective_Impact)) / 1000d ;

            out.println(g + Detectability_Objective_Impact+"<br/>");


              // out.println(Av_degree + "  -  " + Av_Significance + "   =" + Av_Obj_Impact + "   :  " +
Av_Obj_Significance  + "<br/>");
            //   out.println(Av_Impact + "  -  " + Av_Obj_Impact + "   =" + Av_Objective_Impact + "
;   " + Av_Significance_Value + "  -  " + Av_Obj_Significance + "   =" +
Av_Objective_Significance + "<br/>xxxxxxxxxxx<br/>");
                 String  sql ="";
                   sql = "insert into tblbse_threat_objective ";
                   sql += "( Threat_ID, Remediation_Objective,  "
                          + " Detectability_Objective"     + " ) ";
                   sql += " Values(" + Threat_ID + ", "  + Remediation_Objective_Impact + ", "
                           + Detectability_Objective_Impact  +  ")";

//int Save_Asset = database.insert(sql);

    }
////  Remediation and Detectability

%>

<%
/*
// OBJECTIVITY SCORE ASSET THREAT

//out.println(dSupport);

 // assetthreat certainty
  database.tbl_Name = "tblbse_assetthreat";
  /*String jSql = "SELECT tblbse_assetthreat.*, tblbse_assetthreat_pereception_certainty.* " +
               " FROM tblbse_assetthreat_pereception_certainty inner join tblbse_assetthreat" +
                " on tblbse_assetthreat_pereception_certainty.AssetThreat_ID =
tblbse_assetthreat.AssetThreat_ID"
                + " where tblbse_assetthreat_pereception_certainty.AssetThreat_ID =
tblbse_assetthreat.AssetThreat_ID";
  String jSql ="SELECT tblbse_Assetthreat.Asset_ID as Asset_ID, "
         + "tblbse_Assetthreat.AssetThreat_ID as AssetThreat_ID,  "
         + "tblbse_Assetthreat.Availability_Impact, "
         + "tblbse_Assetthreat.Confidentiality_Impact,  "
         + "tblbse_Assetthreat.Exploitability,  "
         + "tblbse_Assetthreat.Ease_Of_Exploitability,  "
         + "tblbse_Assetthreat.Integrity_Impact,   "
         + "tblbse_Assets.Availability_Significance, "
         + "tblbse_Assets.Confidentiality_Significance,  "
         + "tblbse_Assets.Integrity_Significance,   "
         + "tblbse_Assets.Asset_ID as Asset_ID,   "
         + "tblbse_assetthreat_pereception_certainty.Availability_Impact_Degree as 
Availability_Impact_Degree, "
         + "tblbse_assetthreat_pereception_certainty.Confidentiality_Impact_Degree as 
Confidentiality_Impact_Degree, "
         + "tblbse_assetthreat_pereception_certainty.Integrity_Impact_Degree as 
Integrity_Impact_Degree, "
         + "tblbse_assetthreat_pereception_certainty.Exploitability_Degree, "
         + "tblbse_assetthreat_pereception_certainty.Ease_Of_Exploitability_Degree, "
         + "tblbse_asset_perception_certainty.Availability_Significance_Degree as 
Availability_Significance_Degree, "
         + "tblbse_asset_perception_certainty.Confidentiality_Significance_Degree as 
Confidentiality_Significance_Degree, "
         + "tblbse_asset_perception_certainty.Integrity_Significance_Degree as 
Integrity_Significance_Degree "
         + "FROM tblbse_assetthreat "
         + "inner join tblbse_asset_perception_certainty "
         + "on tblbse_assetthreat.Asset_ID = tblbse_asset_perception_certainty.Asset_ID "
         + "inner join tblbse_assets "
         + "on tblbse_assetthreat.Asset_ID = tblbse_assets.Asset_ID "
         + "inner join tblbse_assetthreat_pereception_certainty "
         + "on tblbse_assetthreat_pereception_certainty.AssetThreat_ID = 
tblbse_assetthreat.AssetThreat_ID";

          ResultSet get_Threat = database.Find_Record_tablejoin(jSql);
          int g =0;
          while(get_Threat.next()) {
            //out.print(get_Admin.getInt("Admin_ID") + " " + get_Admin.getString("Name") + 
"............<br/>");
            int AssetThreat_ID = get_Threat.getInt("AssetThreat_ID");
            int Asset_ID = get_Threat.getInt("Asset_ID");
            double Av_degree =0.0;
            double Con_degree =0.0;
            double Int_degree =0.0;
              double Exp_degree =0.0;
             double EaseOfExp_degree =0.0;

            double Av_Significance =0.0;
            double Con_Significance =0.0;
            double Int_Significance =0.0;

              double Av_Obj_Impact =0.0;
              double Con_Obj_Impact =0.0;
            double Int_Obj_Impact =0.0;
              double Exp_Obj_Impact =0.0;
             double EaseOfExp_Obj_Impact =0.0;

             double Av_Obj_Significance =0.0;
               double Con_Obj_Significance =0.0;
            double Int_Obj_Significance =0.0;

            double Exp_Objective=0.0;
            double Ease_Of_Exp_Objective=0.0;

            Av_degree = get_Threat.getDouble("Availability_Impact_Degree");
            Con_degree = get_Threat.getDouble("Confidentiality_Impact_Degree");
            Int_degree = get_Threat.getDouble("Integrity_Impact_Degree");
            Exp_degree = get_Threat.getDouble("Exploitability_Degree");
            EaseOfExp_degree = get_Threat.getDouble("Ease_Of_Exploitability_Degree");
```

```
                Av_Significance = get_Threat.getDouble("Availability_Significance_Degree");
                Con_Significance = get_Threat.getDouble("Confidentiality_Significance_Degree");
                Int_Significance = get_Threat.getDouble("Integrity_Significance_Degree");

                Av_Obj_Impact = Av_degree /(Av_Significance + Av_degree);
                Con_Obj_Impact = Con_degree /(Con_Significance + Con_degree);
                Int_Obj_Impact = Int_degree /(Int_Significance + Int_degree);
                Exp_Obj_Impact = Exp_degree /(EaseOfExp_degree + Exp_degree);
                EaseOfExp_Obj_Impact = EaseOfExp_degree /(EaseOfExp_degree + Exp_degree);

                Av_Obj_Significance = Av_Significance /(Av_Significance + Av_degree);
                Con_Obj_Significance = Con_Significance /(Con_Significance + Con_degree);
                Int_Obj_Significance = Int_Significance /(Int_Significance + Int_degree);
                //Exp_degree = get_Threat.getDouble("Exploitability_Degree");
                //easeOfExp_degree = get_Threat.getDouble("Ease_Of_Exploitability_Degree");

                String Av_Impact = "0";
                String Con_Impact = "0";
                String Int_Impact = "0";

                String Av_Significance_Value = "0";
                String Con_Significance_Value = "0";
                String Int_Significance_Value = "0";

                String Exp_Impact = "0";
                String Ease_Of_Exp_Impact = "0";

            Av_Impact =   database.Find_Slave_ReferenceByID("tblBse_Slave","Value","Slave_ID",
get_Threat.getInt("Availability_Impact"));
            Con_Impact =   database.Find_Slave_ReferenceByID("tblBse_Slave","Value","Slave_ID",
get_Threat.getInt("Confidentiality_Impact"));
            Int_Impact =   database.Find_Slave_ReferenceByID("tblBse_Slave","Value","Slave_ID",
get_Threat.getInt("Integrity_Impact"));
             Exp_Impact =
database.Find_Slave_ReferenceByID("tblBse_Slave","Value","Slave_ID",
get_Threat.getInt("Exploitability"));
             Ease_Of_Exp_Impact =
database.Find_Slave_ReferenceByID("tblBse_Slave","Value","Slave_ID",
get_Threat.getInt("Ease_Of_Exploitability"));
              Av_Significance_Value =
database.Find_Slave_ReferenceByID("tblBse_Slave","Value","Slave_ID",
get_Threat.getInt("Availability_Significance"));
            Con_Significance_Value =
database.Find_Slave_ReferenceByID("tblBse_Slave","Value","Slave_ID",
get_Threat.getInt("Confidentiality_Significance"));
            Int_Significance_Value =
database.Find_Slave_ReferenceByID("tblBse_Slave","Value","Slave_ID",
get_Threat.getInt("Integrity_Significance"));

                  double Av_Objective_Impact =0.0;
                double Con_Objective_Impact =0.0;
                double Int_Objective_Impact =0.0;
                double Exp_Objective_Impact =0.0;
                double EaseOfExp_Objective_Impact =0.0;

                    double Av_Objective_Significance =0.0;
                double Con_Objective_Significance =0.0;
                double Int_Objective_Significance =0.0;
```

```
                 Av_Objective_Impact = (Integer.valueOf(Av_Impact) * Av_Obj_Impact);
                Con_Objective_Impact = (Integer.valueOf(Con_Impact) * Con_Obj_Impact);
                Int_Objective_Impact = (Integer.valueOf(Int_Impact) * Int_Obj_Impact);

               Exp_Objective_Impact = (Integer.valueOf(Exp_Impact) * Exp_Obj_Impact);
               EaseOfExp_Objective_Impact = (Integer.valueOf(Ease_Of_Exp_Impact) *
EaseOfExp_Obj_Impact);

                 Av_Objective_Impact = Math.round(1000  * (double)(Av_Objective_Impact)) / 1000d ;
                Con_Objective_Impact = Math.round(1000  * (double)(Con_Objective_Impact)) /
1000d ;
                Int_Objective_Impact = Math.round(1000  * (double)(Int_Objective_Impact)) / 1000d ;
                Exp_Objective_Impact = Math.round(1000  * (double)(Exp_Objective_Impact)) / 1000d
;
                EaseOfExp_Objective_Impact = Math.round(1000  *
(double)(EaseOfExp_Objective_Impact)) / 1000d ;

                  Av_Objective_Significance = (Integer.valueOf(Av_Significance_Value) *
Av_Obj_Significance);
                Con_Objective_Significance = (Integer.valueOf(Con_Significance_Value) *
Con_Obj_Significance);
                Int_Objective_Significance = (Integer.valueOf(Int_Significance_Value) *
Int_Obj_Significance);

                 Av_Objective_Significance = Math.round(1000  *
(double)(Av_Objective_Significance)) / 1000d ;
                Con_Objective_Significance = Math.round(1000  *
(double)(Con_Objective_Significance)) / 1000d ;
                Int_Objective_Significance = Math.round(1000  *
(double)(Int_Objective_Significance)) / 1000d ;

                out.println(Av_degree + "  -  " + Av_Significance + "   =" + Av_Obj_Impact + "   :  " +
Av_Obj_Significance  + "<br/>");
                out.println(Av_Impact + "  -   " + Av_Obj_Impact + "   =" + Av_Objective_Impact + "  ;
" + Av_Significance_Value + "  -  " + Av_Obj_Significance + "   =" + Av_Objective_Significance
+ "<br/>xxxxxxxxxxx<br/>");


try{
                    String sql = "insert into tblbse_assetThreat_Objective ";
                    sql += "( AssetThreat_ID, Availability_Impact_Objective,
Confidentiality_Impact_Objective, "
                        + " Integrity_Impact_Objective, Exploitability_Objective,
Ease_Of_Exploitability_Objective)"
                        + "  ";
                    sql += " Values(" + AssetThreat_ID + ", "  + Av_Objective_Impact + ", "
                        + Con_Objective_Impact + ", " + Int_Objective_Impact +  ", "
                        +  Exp_Objective_Impact + ", " + EaseOfExp_Objective_Impact +  ")";

//int Save_AssetThreat = database.insert(sql);

                    sql ="";
                    sql = "insert into tblbse_asset_objective ";
                    sql += "( Asset_ID, Availability_Significance_Objective,
Confidentiality_Significance_Objective, "
                        + " Integrity_Significance_Objective"     + " ) ";
                    sql += " Values(" + Asset_ID + ", "  + Av_Objective_Significance + ", "
                        + Con_Objective_Significance + ", " + Int_Objective_Significance  +  ")";
//int Save_Asset = database.insert(sql);

}catch(Exception ex){
   out.print(ex.getMessage());

   }
            // Exp_Impact =
database.Find_Slave_ReferenceByID("tblBse_Slave","Value","Slave_ID",
get_Threat.getInt("Exploitability"));
            // Ease_Of_Exp_Impact =
database.Find_Slave_ReferenceByID("tblBse_Slave","Value","Slave_ID",
get_Threat.getInt("Ease_Of_Exploitability"));

   }


         /*     Av_Objective = (Integer.valueOf(Av_Impact) * Av_degree);
             Con_Objective = (Integer.valueOf(Con_Impact) * Con_degree);
             Int_Objective = (Integer.valueOf(Int_Impact) * Int_degree);
             Exp_Objective = (Integer.valueOf(Exp_Impact) * Exp_degree) ;
             Ease_Of_Exp_Objective = (Integer.valueOf(Ease_Of_Exp_Impact) * Exp_degree) ;

             Av_Objective = Math.round(1000  * (double)(Av_Objective)) / 1000d ;
             Con_Objective = Math.round(1000  * (double)(Con_Objective)) / 1000d ;
             Int_Objective = Math.round(1000  * (double)(Int_Objective)) / 1000d ;
             Exp_Objective = Math.round(1000  * (double)(Exp_Objective)) / 1000d ;
             Ease_Of_Exp_Objective = Math.round(1000  * (double)(Ease_Of_Exp_Objective)) /
1000d ;



             //Av_Impact +=
database.Find_Slave_ReferenceByID("tblBse_Slave","Display","Slave_ID",
get_Threat.getInt("Availability_Impact"));
                          out.println(Av_Impact + " " + Av_Imp + " * " + Av_degree + "  ="
                                 + Av_Objective + "<br/>");

//out.print(Threat);

             ///// Get All Threats
             //database.tbl_Name = "tblbse_Threat_Perception";

               String sql = "insert into tblbse_assetThreat_Objective ";
                sql += "( AssetThreat_ID, Availability_Impact_Objective,
Confidentiality_Impact_Objective, "
                      + " Integrity_Impact_Objective, Exploitability_Objective,
Ease_Of_Exploitability_Objective)"
                      + "  ";
                sql += " Values(" + Threat + ", "  + Av_Objective_Impact + ", "
                      + Con_Objective_Impact + ", " + Int_Objective_Impact +  ", "
                      +  Exp_Objective_Impact + ", " + EaseOfExp_Objective_Impact +  ")";

int Save_AssetThreat = database.insert(sql);
/*
}
  **/

 /*
               //out.print(sql);
```

```
        //     i


         }
        out.println(database.Connection_Error_Msg);
**/

%>

<%
/*
 * assetthreat perveption data
//double Det_Certainty =1.0;

//out.println(dSupport);

 // assetthreat certainty
database.tbl_Name = "tblbse_admin";
        ResultSet get_admin = database.Find_All_Record();

        while(get_admin.next()) {

// out.print(get_admin.getInt("Admin_ID") + " " + get_admin.getString("Name") +
"............<br/>");
int Admin_ID = get_admin.getInt("Admin_ID");

 database.tbl_Name = "tblbse_assetthreat";
        ResultSet get_Threat = database.Find_All_Record();
        int g =0;
        while(get_Threat.next()) {
          //out.print(get_Admin.getInt("Admin_ID") + " " + get_Admin.getString("Name") +
"............<br/>");
          int Threat = get_Threat.getInt("AssetThreat_ID");
          double Det_Certainty =1.0;

          double Av_Result = 0.0;
         double Av_Certainty = 1.0;

         double Av_Result_Holder = 0.0;

         double Con_Certainty =1.0;
         double Con_Result = 0.0;

         double Int_Certainty =1.0;
         double Int_Result = 0.0;

         double Exp_Certainty =1.0;
         double Exp_Result = 0.0;

         double easeOfExp_Certainty =1.0;
         double easeOfExp_Result = 0.0;
//out.print(Threat);

            double Av = (1 + (int)(Math.random() * (99 - 0) + 1)) /(double) 100;
            double Con = (1 + (int)(Math.random() * (99 - 0) + 1)) /(double) 100;
            double Inte = (1 + (int)(Math.random() * (99 - 0) + 1)) /(double) 100;
            double Exp = (1 + (int)(Math.random() * (99 - 0) + 1)) /(double) 100;
            double easeOfExp = (1 + (int)(Math.random() * (99 - 0) + 1)) /(double) 100;
```

```
                   out.println(Av + "<br/>");
                 ///// Get All Threats
                 //database.tbl_Name = "tblbse_Threat_Perception";
             // ResultSet get_assetThreat =
database.Find_Record_Where("tblbse_assetThreat_pereception", " AssetThreat_ID=" + Threat);
               //   while(get_assetThreat.next()) {
            /*         Av_Result = 1 - get_assetThreat.getDouble("Availability_Impact_Degree");
                   Con_Result = 1 - get_assetThreat.getDouble("Confidentiality_Impact_Degree");
                   Int_Result = 1 - get_assetThreat.getDouble("Integrity_Impact_Degree");
                   Exp_Result = 1 - get_assetThreat.getDouble("Exploitability_Degree");
                  easeOfExp_Result = 1 -
get_assetThreat.getDouble("Ease_Of_Exploitability_Degree");

                   Av_Certainty = Math.round(1000  * (double)(Av_Certainty * Av_Result)) / 1000d ;
                   Con_Certainty = Math.round(1000  * (double)(Con_Certainty * Con_Result)) / 1000d
;
                   Int_Certainty = Math.round(1000  * (double)(Int_Certainty * Int_Result)) / 1000d ;
                   Exp_Certainty = Math.round(1000  * (double)(Exp_Certainty * Exp_Result)) / 1000d
;
                  easeOfExp_Certainty = Math.round(1000  * (double)(easeOfExp_Certainty *
easeOfExp_Result)) / 1000d ;

                   //double dSupport = Math.round(1000  * (double)7.99876) / 1000d;
                    //  g++;
                     // out.print(g + "<br/>");


             //      out.print(get_assetThreat.getInt("Threat_ID") + " " +
get_assetThreat.getInt("Admin_ID") +  " " +
               //            get_assetThreat.getDouble("Remediation_Degree") + "  = " +Rem_Certainty +
"............<br/>");


                  //  out.print(Av + " " + Con + " " + Inte + " " + Exp + "   -<br/>");
           //  }
 *
 * */
/*
                  String sql = "insert into tblbse_assetThreat_pereception ";
                   sql += "( AssetThreat_ID, Admin_ID, Availability_Impact_Degree,
Confidentiality_Impact_Degree, "
                          + " Integrity_Impact_Degree, Exploitability_Degree,
Ease_Of_Exploitability_Degree)"
                          + "  ";
                   sql += " Values(" + Threat + ", " + Admin_ID + ", "  + Av + ", "
                          + Con + ", " + Inte +  ", " +  Exp + ", " + easeOfExp + ")";

                  //out.print(sql);

                 int Save_AssetThreat = database.insert(sql);

    }
}
                   /*Availability_Impact_Degree
                          Confidentiality_Impact_Degree
                          Integrity_Impact_Degree
                          Exploitability_Degree
```

```
          }
         out.println(database.Connection_Error_Msg);
**/

%>
<%
/*
 // assetthreat certainty

// out.print(get_admin.getInt("Admin_ID") + " " + get_admin.getString("Name") +
"............<br/>");

 database.tbl_Name = "tblbse_assetthreat";
         ResultSet get_Threat = database.Find_All_Record();
         int g =0;
         while(get_Threat.next()) {
            //out.print(get_Admin.getInt("Admin_ID") + " " + get_Admin.getString("Name") +
"............<br/>");
            int Threat = get_Threat.getInt("AssetThreat_ID");
            double Det_Certainty =1.0;

            double Av_Result = 0.0;
          double Av_Certainty = 1.0;

          double Av_Result_Holder = 0.0;

          double Con_Certainty =1.0;
          double Con_Result = 0.0;

          double Int_Certainty =1.0;
          double Int_Result = 0.0;

          double Exp_Certainty =1.0;
          double Exp_Result = 0.0;

          double easeOfExp_Certainty =1.0;
          double easeOfExp_Result = 0.0;

out.print(Threat);

            ///// Get All Threats
            //database.tbl_Name = "tblbse_Threat_Perception";
          ResultSet get_assetThreat =
database.Find_Record_Where("tblbse_assetThreat_pereception", " AssetThreat_ID=" + Threat);
             while(get_assetThreat.next()) {
               Av_Result = 1 - get_assetThreat.getDouble("Availability_Impact_Degree");
               Con_Result = 1 - get_assetThreat.getDouble("Confidentiality_Impact_Degree");
               Int_Result = 1 - get_assetThreat.getDouble("Integrity_Impact_Degree");
               Exp_Result = 1 - get_assetThreat.getDouble("Exploitability_Degree");
              easeOfExp_Result = 1 -
get_assetThreat.getDouble("Ease_Of_Exploitability_Degree");

               Av_Certainty = Math.round(1000  * (double)(Av_Certainty * Av_Result)) / 1000d ;
               Con_Certainty = Math.round(1000  * (double)(Con_Certainty * Con_Result)) / 1000d
;
               Int_Certainty = Math.round(1000  * (double)(Int_Certainty * Int_Result)) / 1000d ;
               Exp_Certainty = Math.round(1000  * (double)(Exp_Certainty * Exp_Result)) / 1000d
;
```

```
                    easeOfExp_Certainty = Math.round(1000  * (double)(easeOfExp_Certainty *
easeOfExp_Result)) / 1000d ;
                     //double dSupport = Math.round(1000  * (double)7.99876) / 1000d;
                       //  g++;
                        // out.print(g + "<br/>");

             //      out.print(get_assetThreat.getInt("Threat_ID") + " " +
get_assetThreat.getInt("Admin_ID") +  " " +
               //              get_assetThreat.getDouble("Remediation_Degree") + "  = " +Rem_Certainty +
"............<br/>");
                   //  out.print(Av + " " + Con + " " + Inte + " " + Exp + "   -<br/>");
             }

                    Av_Certainty = Math.round(1000  * (double)(1 - Av_Certainty )) / 1000d ;
                     Con_Certainty = Math.round(1000  * (double)(1 -Con_Certainty)) / 1000d ;
                     Int_Certainty = Math.round(1000  * (double)(1 - Int_Certainty )) / 1000d ;
                     Exp_Certainty = Math.round(1000  * (double)(1 - Exp_Certainty)) / 1000d ;
                    easeOfExp_Certainty = Math.round(1000  * (double)(1 - easeOfExp_Certainty )) /
1000d ;

                    String sql = "insert into tblbse_assetThreat_pereception_certainty ";
                     sql += "( AssetThreat_ID, Availability_Impact_Degree,
Confidentiality_Impact_Degree, "
                             + " Integrity_Impact_Degree, Exploitability_Degree,
Ease_Of_Exploitability_Degree)"
                             + "  ";
                     sql += " Values(" + Threat + ", "  + Av_Certainty + ", "
                             + Con_Certainty + ", " + Int_Certainty +  ", " +  Exp_Certainty + ", " +
easeOfExp_Certainty + ")";

                    out.print(sql + "<br/>");

               // int Save_AssetThreat = database.insert(sql);
    }


                     /*Availability_Impact_Degree
                             Confidentiality_Impact_Degree
                             Integrity_Impact_Degree
                             Exploitability_Degree
               }
             out.println(database.Connection_Error_Msg);
**/
%>

<%
/*
//double Det_Certainty =1.0;

//out.println(dSupport);

    /// asset certanty
  database.tbl_Name = "tblbse_assets";
             ResultSet get_Threat = database.Find_All_Record();
             int g =0;
             while(get_Threat.next()) {
                //out.print(get_Admin.getInt("Admin_ID") + " " + get_Admin.getString("Name") +
"............<br/>");
                int Threat = get_Threat.getInt("Asset_ID");
```

```
          double Det_Certainty =1.0;
        double Av_Result = 0.0;
         double Av_Certainty = 1.0;
         double Av_Result_Holder = 0.0;

         double Con_Certainty =1.0;
         double Con_Result = 0.0;

         double Int_Certainty =1.0;
         double Int_Result = 0.0;

         double Exp_Certainty =1.0;
         double Exp_Result = 0.0;
          ///// Get All Threats
          //database.tbl_Name = "tblbse_Threat_Perception";
          ResultSet get_assetThreat = database.Find_Record_Where("tblbse_asset_perception", "
asset_ID=" + Threat);
          while(get_assetThreat.next()) {
             Av_Result = 1 - get_assetThreat.getDouble("Availability_Significance_Degree");
             Con_Result = 1 - get_assetThreat.getDouble("Confidentiality_Significance_Degree");
             Int_Result = 1 - get_assetThreat.getDouble("Integrity_Significance_Degree");
              out.print(get_assetThreat.getDouble("Availability_Significance_Degree") + "  -   "
+Av_Certainty + "<br/>xxxx");

             Av_Certainty = Math.round(1000  * (double)(Av_Certainty * Av_Result)) / 1000d ;
             Con_Certainty = Math.round(1000  * (double)(Con_Certainty * Con_Result)) / 1000d
;
             Int_Certainty = Math.round(1000  * (double)(Int_Certainty * Int_Result)) / 1000d ;
             out.print(get_assetThreat.getDouble("Availability_Significance_Degree") + "  -   "
+Av_Certainty + "<br/>");

             //double dSupport = Math.round(1000  * (double)7.99876) / 1000d;
              //  g++;
               // out.print(g + "<br/>");


       //      out.print(get_assetThreat.getInt("Threat_ID") + " " +
get_assetThreat.getInt("Admin_ID") +  " " +
        //             get_assetThreat.getDouble("Remediation_Degree") + "  = " +Rem_Certainty +
"...........<br/>");


            //  out.print(Av + " " + Con + " " + Inte + " " + Exp + "   -<br/>");
        }
            Av_Certainty = Math.round(1000  * (double)(1 - Av_Certainty )) / 1000d ;
             Con_Certainty = Math.round(1000  * (double)(1 - Con_Certainty )) / 1000d ;
            Int_Certainty = Math.round(1000  * (double)(1 - Int_Certainty )) / 1000d ;


            String sql = "insert into tblbse_asset_perception_certainty ";
             sql += "( Asset_ID, Availability_Significance_Degree,
Confidentiality_Significance_Degree, "
                  + " Integrity_Significance_Degree)"
                  + "  ";
             sql += " Values(" + Threat + ", "  + Av_Certainty + ", "
                  + Con_Certainty + ", " + Int_Certainty +  ")";

            //out.print(sql);
```

```
                 int Save_AssetThreat = database.insert(sql);

}

                  /*Availability_Impact_Degree
                        Confidentiality_Impact_Degree
                        Integrity_Impact_Degree
                        Exploitability_Degree
               }
             out.println(database.Connection_Error_Msg);
* */

%>

<%
  //double Det_Certainty =1.0;

//out.println(dSupport);

/*
  database.tbl_Name = "tbllkup_threats";
            ResultSet get_Threat = database.Find_All_Record();
            int g =0;
            while(get_Threat.next()) {
               //out.print(get_Admin.getInt("Admin_ID") + " " + get_Admin.getString("Name") +
"............<br/>");
               int Threat = get_Threat.getInt("Threat_ID");
               double Det_Certainty =1.0;
             double Det_Result = 0.0;
              double Det_Result_Holder = 0.0;

              double Rem_Certainty =1.0;
              double Rem_Result = 0.0;
               ///// Get All Threats
               //database.tbl_Name = "tblbse_Threat_Perception";
               ResultSet get_assetThreat = database.Find_Record_Where("tblbse_threat_perception", "
Threat_ID=" + Threat);
               while(get_assetThreat.next()) {
                  Det_Result = 1 - get_assetThreat.getDouble("Detectability_Degree");
                  Rem_Result = 1 - get_assetThreat.getDouble("Remediation_Degree");

                  Det_Certainty = Math.round(1000  * (double)(Det_Certainty * Det_Result)) / 1000d ;
                  Rem_Certainty = Math.round(1000  * (double)(Rem_Certainty * Rem_Result)) /
1000d ;

                  //double dSupport = Math.round(1000  * (double)7.99876) / 1000d;
                   //  g++;
                    // out.print(g + "<br/>");

                 out.print(get_assetThreat.getInt("Threat_ID") + " " +
get_assetThreat.getInt("Admin_ID") +  " " +
                        get_assetThreat.getDouble("Remediation_Degree") + "  = " +Rem_Certainty +
"...........<br/>");
                 //  out.print(Av + " " + Con + " " + Inte + " " + Exp + "   -<br/>");
            }
                 String sql = "insert into tblbse_threat_perception_certainty ";
                  sql += "( Threat_ID, Detectability_Degree, Remediation_Degree )"
                        + "  ";
                  sql += " Values(" + Threat + ", "  + Det_Certainty + ", "
                        + Rem_Certainty +  ")";



                 //out.print(sql);

                 int Save_AssetThreat = database.insert(sql);

                  /*Availability_Impact_Degree
                        Confidentiality_Impact_Degree
                        Integrity_Impact_Degree
                        Exploitability_Degree
              }
            out.println(database.Connection_Error_Msg);
          * */

%>

<%
 /*
  database.tbl_Name = "tblbse_Admin";
            ResultSet get_Admin = database.Find_All_Record();
            int g =0;
            while(get_Admin.next()) {
               out.print(get_Admin.getInt("Admin_ID") + " " + get_Admin.getString("Name") +
".............<br/>");
               int Admin = get_Admin.getInt("Admin_ID");

               ///// Get All Threats
               database.tbl_Name = "tbllkup_Threats";
               ResultSet get_assetThreat = database.Find_All_Record();
               while(get_assetThreat.next()) {
                   //  g++;
                    // out.print(g + "<br/>");


                 //  out.print(get_assetThreat.getInt("AssetThreat_ID") + "............");

                  int AssetTh = get_assetThreat.getInt("Threat_ID");

                  double Av = (1 + (int)(Math.random() * (99 - 0) + 1)) /(double) 100;
                  double Con = (1 + (int)(Math.random() * (99 - 0) + 1)) /(double) 100;
                  double Inte = (1 + (int)(Math.random() * (99 - 0) + 1)) /(double) 100;
                  double Exp = (1 + (int)(Math.random() * (99 - 0) + 1)) /(double) 100;

                  out.print(Av + " " + Con + " " + Inte + " " + Exp + "   -<br/>");

                  String sql = "insert into tblbse_threat_perception ";
                  sql += "(Admin_ID,  Threat_ID, Detectability_Degree, Remediation_Degree )"
                        + "  ";
                  sql += " Values(" + Admin + ", "  + AssetTh + ", "  + Av + ", "
                        + Con +  ")";



                 //out.print(sql);

                  int Save_AssetThreat = database.insert(sql);
```

```
                    /*Availability_Impact_Degree
                            Confidentiality_Impact_Degree
                            Integrity_Impact_Degree
                            Exploitability_Degree

                    }

              }
            out.println(database.Connection_Error_Msg);
   * */

%>

<%

/*

            //// counting Asset
            database.tbl_Name = "tblbse_Assets";
            ResultSet get_Asset = database.Find_All_Record();
            while(get_Asset.next()) {
                out.print(get_Asset.getInt("Asset_ID") + " " + get_Asset.getString("Name") +
"............<br/>");

                ///// Get All Threats
                database.tbl_Name = "tbllkup_Threats";
                ResultSet get_Threat = database.Find_All_Record();
                while(get_Threat.next()) {
                    int Av_Imp = 20 + (int)(Math.random() * (23 - 20) +1);
                    int Conf_Imp = 23 + (int)(Math.random() * (26 - 23) +1);
                    int Int_Imp = 26 + (int)(Math.random() * (29 - 26) +1);
                    int Expl = 29 + (int)(Math.random() * (33 - 29) +1);
                    int easeOfExpl = 36 + (int)(Math.random() * (39 - 36) +1);

                    int Asset_ID = get_Asset.getInt("Asset_ID");
                    int Threat_ID = get_Threat.getInt("Threat_ID");
                    int Availability_Impact = Av_Imp;
                    int Confidentiality_Impact = Conf_Imp;
                    int Integrity_Impact = Int_Imp;
                    int Exploitability = Expl;
                    int easeOfExploitability = easeOfExpl;

                    String sql = "insert into tblBse_AssetThreat ";
                    sql += "(Asset_ID,  Threat_ID, Availability_Impact, Confidentiality_Impact, "
                            + "Integrity_Impact, Exploitability, ease_Of_Exploitability ) ";
                    sql += " Values('" + Asset_ID + "', "  + Threat_ID + ", "  + Availability_Impact + ", "
                            + Confidentiality_Impact + ", " + Integrity_Impact + ", " + Exploitability + ", " +
easeOfExploitability + ")";
                    out.print(sql);

                    int Save_AssetThreat = database.insert(sql);

                    String sAv_Imp
=database.Find_Slave_ReferenceByID("tblBse_Slave","Display","Slave_ID", Av_Imp);
                    String sConf_Imp
=database.Find_Slave_ReferenceByID("tblBse_Slave","Display","Slave_ID", Conf_Imp);
                    String sInt_Imp
=database.Find_Slave_ReferenceByID("tblBse_Slave","Display","Slave_ID", Int_Imp);
```

```
                    String sExpl
=database.Find_Slave_ReferenceByID("tblBse_Slave","Display","Slave_ID", Expl);

                    out.print(get_Threat.getInt("Threat_ID") + "  "
                            + get_Threat.getString("Name") + "  ="
                            + sAv_Imp + " / " + sConf_Imp + " / " + sInt_Imp + " / " + sExpl
                            + "<br/>");
                }

            } ///  End Get Asset

/*for (int j=0; j < 25; j++) {

int len =9;
String Alpha_Num ="1234567890ABCDEFGHIJKLMNOPQRSTUVWXYZ";

StringBuffer sb = new StringBuffer(len);

for (int i=0; i < len; i++) {

    int ndx = (int)(Math.random() * Alpha_Num.length());
    sb.append(Alpha_Num.charAt(ndx));
    }

int threat = 0 + (int)(Math.random() * (5 - 0) +1);
int detect = 5 + (int)(Math.random() * (8 - 5) +1);
int rem = 8 + (int)(Math.random() * (11 - 8) +1);

out.print(sb.toString()  + " " + threat + " " + detect + " " + rem );

String sql = "insert into tblIkup_Threats ";
sql += "(Name, Threat_Objective, Detectability, Remediation) ";
sql += " Values('" + sb.toString()  + "', " + threat + ", "  + detect + ", "
        + rem + ")";
out.print(sql);

int Save_Threat = database.insert(sql);
}
**/
%>

<%

if(request.getParameter("btn_Save")!=null) {

int Asset_ID =0;
int Threat_ID=0;
int Availability_Impact=0;
int Confidentiality_Impact=0;
int Integrity_Impact=0;
int Discoverability=0;
int Remediation=0;
int Exploitability =0;

Asset_ID = Integer.valueOf(request.getParameter("ddl_Asset_ID"));
Threat_ID = Integer.valueOf(request.getParameter("ddl_Threat_ID"));
Availability_Impact = Integer.valueOf(request.getParameter("ddl_Availability_Impact"));
Confidentiality_Impact = Integer.valueOf(request.getParameter("ddl_Confidentiality_Impact"));
Integrity_Impact = Integer.valueOf(request.getParameter("ddl_Integrity_Impact"));
```

```
Discoverability = Integer.valueOf(request.getParameter("ddl_Discoverability"));
Remediation = Integer.valueOf(request.getParameter("ddl_Remediation"));
Exploitability = Integer.valueOf(request.getParameter("ddl_Exploitability"));

String sqli = "insert into tblBse_AssetThreat ";
sqli += "(Asset_ID,  Threat_ID, Availability_Impact, Confidentiality_Impact, "
        + "Integrity_Impact, Discoverability, Remediation, Exploitability ) ";
sqli += " Values('" + Asset_ID + "', "  + Threat_ID + ", "  + Availability_Impact + ", "
        + Confidentiality_Impact + ", " + Integrity_Impact + ", " + Discoverability + ", " +
Remediation + ", " + Exploitability + ")";
out.print(sqli);

int Save_AssetThreati = database.insert(sqli);
out.print(database.Connection_Error_Msg);
//out.print("Saved");
}
else
    {
    }
%>
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN"
   "http://www.w3.org/TR/html4/loose.dtd">

<html>
    <head>
        <meta http-equiv="Content-Type" content="text/html; charset=UTF-8">
        <title>JSP Page</title>
    </head>
    <body>
        <h1>Asset Threat</h1>

        <form name="frm" action="" method="post">
            <table>

               <tr><td>Asset:</td>
                <td>
                <select name="ddl_Asset_ID" >
                         <%   out.print(database.Load_Asset_On_Host("tblBse_Assets", "Name" ,
"Asset_ID")); %>
                </select>
                </td>
                </tr>

                <tr><td>Threat:</td>
                <td>
                <select name="ddl_Threat_ID" >
                         <%   out.print(database.Load_Master_Value("tbllkup_Threats", "Name" ,
"Threat_ID")); %>
                </select>
                </td>
                </tr>
                <tr><td>Availability Impact:</td>
                <td>
                <select name="ddl_Availability_Impact" >
                         <%   out.print(database.Slave_Reference("Availability Impact")); %>
                </select>
                </td>
                </tr>
                <tr><td>Confidentiality Impact:</td>
                <td>
                <select name="ddl_Confidentiality_Impact" >
                         <%   out.print(database.Slave_Reference("Confidentiality Impact")); %>
                </select>
                </td>
                </tr>

                <tr><td>Integrity Impact:</td>
                <td>
                <select name="ddl_Integrity_Impact" >
                         <%   out.print(database.Slave_Reference("Integrity Impact")); %>
                </select>
                </td>
                </tr>

                <tr><td>Discoverability:</td>
                <td>
                <select name="ddl_Discoverability" >
                         <%   out.print(database.Slave_Reference("Discoverability")); %>
                </select>
                </td>
                </tr>

                <tr><td>Remediation:</td>
                <td>
                <select name="ddl_Remediation" >
                         <%   out.print(database.Slave_Reference("Remediation")); %>
                </select>
                </td>
                </tr>

                <tr><td>Exploitability:</td>
                <td>
                <select name="ddl_Exploitability" >
                         <%   out.print(database.Slave_Reference("Exploitability")); %>
                </select>
                </td>
                </tr>

                <tr><td></td><td><input type="Submit" name="btn_Save" value="Save"/></td></tr>
            </table>
            <table border="1">
                <tr>
                    <td>S No.</td>
                    <td>Asset</td>
                    <td>Threat</td>
                    <td>Availability Impact</td>
                    <td>Confidentiality Impact</td>
                    <td>Integrity Impact</td>
                    <td>Discoverability</td>
                    <td>Remediation</td>
                    <td>Exploitability</td>
                     <td>Ease Of Exploitability</td>
                </tr>

                  <%
            String _View="";
            int S_No =0;
```

```
            database.tbl_Name = "tblbse_AssetThreat";
            ResultSet Show_All = database.Find_All_Record();
            while(Show_All.next()) {
              S_No +=1;
               _View += "<tr><td>" + S_No + "</td>";
               _View += "<td>" + database.Find_Slave_ReferenceByID("tblBse_Assets", "Name",
"Asset_ID", Show_All.getInt("Asset_ID")) + "</td>";
               _View += "<td>" + database.Find_Slave_ReferenceByID("tbllkup_Threats", "Name",
"Threat_ID", Show_All.getInt("Threat_ID"))+ "</td>";
               _View += "<td>" +
database.Find_Slave_ReferenceByID("tblBse_Slave","Display","Slave_ID",
Show_All.getInt("Availability_Impact"))+ "</td>";
               _View += "<td>" +
database.Find_Slave_ReferenceByID("tblBse_Slave","Display","Slave_ID",
Show_All.getInt("Confidentiality_Impact"))+ "</td>";
               _View += "<td>" +
database.Find_Slave_ReferenceByID("tblBse_Slave","Display","Slave_ID",
Show_All.getInt("Integrity_Impact"))+ "</td>";
               _View += "<td>" +
database.Find_Slave_ReferenceByID("tblBse_Slave","Display","Slave_ID",
Show_All.getInt("Discoverability"))+ "</td>";
               _View += "<td>" +
database.Find_Slave_ReferenceByID("tblBse_Slave","Display","Slave_ID",
Show_All.getInt("Remediation"))+ "</td>";
               _View += "<td>" +
database.Find_Slave_ReferenceByID("tblBse_Slave","Display","Slave_ID",
Show_All.getInt("Exploitability"))+ "</td>";
              _View += "<td>" +
database.Find_Slave_ReferenceByID("tblBse_Slave","Display","Slave_ID",
Show_All.getInt("Ease_Of_Exploitability"))+ "</td>";
               _View += "</tr>";

            }

          out.print(_View);

            %>
            </table>
        </form>
<%@include  file="foot.jsp" %>
    </body>
</html>
```

LoadHarmoniseView.jsp

```
<%@page import="dClasses.Database_Object" contentType="text/html" pageEncoding="UTF-8"
%>
<%@page   import="dClasses.Database_Object"%>
<%@page  import="java.util.*" %>
<%@page  import="java.sql.*" %>

 <table border="1">
<%
Database_Object database = new Database_Object();
int ClientID = Integer.parseInt(request.getParameter("ClientID"));
String view = request.getParameter("view");
if(view.equals("threat")) {
%>
               <tr>
```

```
                <td>S No.</td>
                <td>Asset</td>
                <td>Threat</td>
                <td>Availability Impact</td>
                <td>Confidentiality Impact</td>
                <td>Integrity Impact</td>
                <td>Exploitability</td>
                <td>Ease Of Exploitability</td>
              <td>Detectability</td>
              <td>Remediation</td>
              </tr>
<%


 String _View="";
      int S_No =0;

      database.tbl_Name = "tblBse_AssetThreat_Client where Client_ID=" + ClientID;
      ResultSet Show_All = database.Find_All_Record();
      if(Show_All != null) {
      while(Show_All.next()) {
        String smps = "";
        smps = "smp1 + smp2 + smp3 + smp4 + smp5 + smp6 + smp7 + smp8 + smp9 +
smp10";
       S_No +=1;
       _View += "<tr><td>" + S_No + "</td>";
        _View += "<td>" + database.Find_Slave_ReferenceByID("tblBse_Assets", "Name",
"Asset_ID", Show_All.getInt("Asset_ID")) + "</td>";
        _View += "<td>" + database.Find_Slave_ReferenceByID("tbllkup_Threats", "Name",
"Threat_ID", Show_All.getInt("Threat_ID"))+ "</td>";
        _View += "<td>" +
database.Find_Slave_ReferenceByID("tblBse_Slave","Display","Slave_ID",
Show_All.getInt("Availability_Impact"))+ " / /";
        _View += "" + database.sumColumn("tblbse_smpassthreat",  "  smpName='Availability
Impact' AND Admin_ID=" + ClientID +
              " AND AssetThreat_Client_ID=" + Show_All.getInt("AssetThreat_Client_ID"),
smps) + "</td>";
        _View += "<td>" +
database.Find_Slave_ReferenceByID("tblBse_Slave","Display","Slave_ID",
Show_All.getInt("Confidentiality_Impact"))+ " / ";
        _View += "" + database.sumColumn("tblbse_smpassthreat",  "
smpName='Confidentiality Impact' AND Admin_ID=" + ClientID +
              " AND AssetThreat_Client_ID=" + Show_All.getInt("AssetThreat_Client_ID"),
smps) + "</td>";

        _View += "<td>" +
database.Find_Slave_ReferenceByID("tblBse_Slave","Display","Slave_ID",
Show_All.getInt("Integrity_Impact"))+ " / ";
        _View += "" + database.sumColumn("tblbse_smpassthreat",  "  smpName='Integrity
Impact' AND Admin_ID=" + ClientID +
              " AND AssetThreat_Client_ID=" + Show_All.getInt("AssetThreat_Client_ID"),
smps) + "</td>";
        _View += "<td>" +
database.Find_Slave_ReferenceByID("tblBse_Slave","Display","Slave_ID",
Show_All.getInt("Exploitability"))+ " / ";
        _View += "" + database.sumColumn("tblbse_smpassthreat",  "
smpName='Exploitability' AND Admin_ID=" + ClientID +
                        " AND AssetThreat_Client_ID=" + Show_All.getInt("AssetThreat_Client_ID"),
smps) + "</td>";
              _View += "<td>" +
database.Find_Slave_ReferenceByID("tblBse_Slave","Display","Slave_ID",
Show_All.getInt("Ease_Of_Exploitation"))+ " / ";
              _View += "" + database.sumColumn("tblbse_smpassthreat",  "  smpName='Ease Of
Exploitability' AND Admin_ID=" + ClientID +
                        " AND AssetThreat_Client_ID=" + Show_All.getInt("AssetThreat_Client_ID"),
smps) + "</td>";
              _View += "<td>" +
database.Find_Slave_ReferenceByID("tblBse_Slave","Display","Slave_ID",
Show_All.getInt("Discoverability"))+ " / ";
              _View += "" + database.sumColumn("tblbse_smpassthreat",  "
smpName='Discoverability' AND Admin_ID=" + ClientID +
                        " AND AssetThreat_Client_ID=" + Show_All.getInt("AssetThreat_Client_ID"),
smps) + "</td>";
              _View += "<td>" +
database.Find_Slave_ReferenceByID("tblBse_Slave","Display","Slave_ID",
Show_All.getInt("Remediation"))+ " / ";
              _View += "" + database.sumColumn("tblbse_smpassthreat",  "
smpName='Remediation' AND Admin_ID=" + ClientID +
                        " AND AssetThreat_Client_ID=" + Show_All.getInt("AssetThreat_Client_ID"),
smps) + "</td>";
              _View += "</tr>";
              _View += "<input id='hfAsset" + Show_All.getString("AssetThreat_Client_ID") + "'
type='hidden' value='" + Show_All.getString("Asset_ID") + "' >";



          }
          }

        out.print(_View);
//out.print("Goo" + AdminID);
%>
<% } else { %>


              <tr>

                  <td>S No.</td>
                 <td>Threat</td>
                 <td>Configuration Instance</td>
                 <td>Exploit Success</td>
                 <td>Consequence</td>
                 <td>Sensitivity</td>
              </tr>
<%

 String _View="";
          int S_No =0;

          database.tbl_Name = "tblbse_Admin_Perspective_Client where Client_ID=" + ClientID;
          ResultSet Show_All = database.Find_All_Record();
          if(Show_All != null) {
          while(Show_All.next()) {

              String smps = "";
```

```
                smps = "smp1 + smp2 + smp3 + smp4 + smp5 + smp6 + smp7 + smp8 + smp9 +
smp10";
                S_No +=1;
             int Threat_Objective =
Integer.valueOf(database.Find_Slave_ReferenceByID("tbllkup_Threats", "Threat_Objective",
"Threat_ID", Show_All.getInt("Threat_ID")));

              _View += "<tr><td>" + S_No + "</td>";
            _View += "<td>" + database.Find_Slave_ReferenceByID("tbllkup_Threats", "Name",
"Threat_ID", Show_All.getInt("Threat_ID"))+ "</td>";

                _View += "<td>" +
database.Find_Slave_ReferenceByID("tblBse_Response_SubCategory","Name","Response_SubC
ategory_ID", Show_All.getInt("Configuration_Instance"))+  "</td>";
             _View += "<td>" +
database.Find_Slave_ReferenceByID("tblBse_Slave","Display","Slave_ID",
Show_All.getInt("Exploit_Success"))+ " / ";
                _View += "" + database.sumColumn("tblbse_smpperspective",  "
smpName='Frequency' AND Admin_ID=" + ClientID +
                            " AND Admin_Perspective_ID=" + Show_All.getInt("Admin_Perspective_ID"),
smps) + "</td>";
                _View += "<td>" +
database.Find_Slave_ReferenceByID("tblBse_Slave","Display","Slave_ID",
Show_All.getInt("Consequence"))+ " / ";
                _View += "" + database.sumColumn("tblbse_smpperspective",  "  smpName='Severity'
AND Admin_ID=" + ClientID +
                            " AND Admin_Perspective_ID=" + Show_All.getInt("Admin_Perspective_ID"),
smps) + "</td>";
                _View += "<td>" +
database.Find_Slave_ReferenceByID("tblBse_Slave","Display","Slave_ID",
Show_All.getInt("Sensitivity"))+ " / ";
                _View += "" + database.sumColumn("tblbse_smpperspective",  "
smpName='Resistance' AND Admin_ID=" + ClientID +
                            " AND Admin_Perspective_ID=" + Show_All.getInt("Admin_Perspective_ID"),
smps) + "</td>";
 //             _View += "<input id='hfAPID" + Show_All.getString("Admin_Perspective_ID") + "'
type='hidden' value='" + Show_All.getString("Admin_Perspective_ID") + "' >";

                _View += "</tr>";

              //  _View += "<input id='hfAsset" + Show_All.getString("AssetThreat_Client_ID") + "'
type='hidden' value='" + Show_All.getString("Asset_ID") + "' >";



             }
            }

          out.print(_View);
//out.print("Goo" + AdminID);
%>

<% } %>


 </table>
```

LoadReport.jsp

```
<%--
   Document   : loadReport
```

```
    Created on : Apr 9, 2014, 5:21:13 AM
    Author     : ORIOLA
--%>

<%@page import="dClasses.Database_Object" contentType="text/html" pageEncoding="UTF-
8"%>
<%@page   import="dClasses.Database_Object"%>
<%@page  import="java.util.*" %>
<%@page  import="java.sql.*" %>
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN"
  "http://www.w3.org/TR/html4/loose.dtd">

<html>
    <head>
        <meta http-equiv="Content-Type" content="text/html; charset=UTF-8">
        <title>JSP Page</title>
    </head>
    <body>
        <div>
            <table border="1">
                <tr>
                    <th>S.No</th>
                    <th>Threat</th>
                    <th>Objective Risk Of Exposure</th>
                    <th>Objective Exploitability</th>
                    <th>Objective Damage</th>
                    <th>Attacker's Rating</th>
                    <th>Objective Frequency</th>
                    <th>Objective Severity</th>
                    <th>Objective Resistance</th>
                    <th>Victims's Rating</th>
                    <th>Threat Rating</th>

                </tr>
<%
Database_Object database = new Database_Object();
  database.tbl_Name = "tblbse_threatrating";

            ResultSet Show_All = database.Find_All_Record();
            if(Show_All != null) {
                int sn=0;

                while(Show_All.next()) {
                    sn++;

                    out.print("<tr>"
                            + "<td>" + sn + "</td>"
                            + "<td>" +
database.Find_Slave_ReferenceByID("tbllkup_threats","Name","Threat_ID",
Show_All.getInt("Threat_ID")) + "</td>"
                            + "<td>" + Show_All.getDouble("ObjRisk") + "</td>"
                            + "<td>" + Show_All.getDouble("ObjExp") + "</td>"
                            + "<td>" + Show_All.getDouble("ObjDam") + "</td>"
                            + "<td>" + Show_All.getDouble("AttackerRating") + "</td>"
                            + "<td>" + Show_All.getDouble("ObjFreq") + "</td>"
                            + "<td>" + Show_All.getDouble("ObjSev") + "</td>"
                            + "<td>" + Show_All.getDouble("ObjRes") + "</td>"
                            + "<td>" + Show_All.getDouble("VictimRating") + "</td>"
```

```
                     + "<td>" + (Show_All.getDouble("VictimRating") +
Show_All.getDouble("AttackerRating")) + "</td>"
                     + "</tr>");
            }

         }

%>

         </table>

      </div>
   </body>
</html>
```

Report.jsp

```
<%--
   Document   : frm_Client.jsp
   Created on : Aug 19, 2013, 3:14:04 PM
   Author     : ORIOLA
--%>

<%@page import="dClasses.Database_Object" contentType="text/html" pageEncoding="UTF-
8"%>
<%@page  import="dClasses.Database_Object"%>
<%@page  import="java.util.*" %>
<%@page  import="java.sql.*" %>
<%@include  file="head.jsp" %>
<script src="smp.js" type="text/javascript" ></script>
<script src="../js/jquery.min.js" ></script>



<%
Database_Object database = new Database_Object();

%>

<%
/// RISK OF EXPOSURE
String AdminID = request.getParameter("client");
String Threat = request.getParameter("Threat");
String dTable = request.getParameter("dbtable");

int S_No=0;
String View = "";
//out.print(AdminID + Threat);

 database.tbl_Name = "tblbse_assetthreat_client";
 ArrayList<String> AssetThreatList = new ArrayList<String>();
 String sql = "select Asset_ID, Threat_ID, AssetThreat_Client_ID from tblbse_assetthreat_client ";
          ResultSet Show_All = database.query(sql);
           View += "<table>";
           int x=0;
          while(Show_All.next()) {
             // View += "<tr><td>" + Show_All.getInt("Asset_ID") + "</td></tr>";

             String value = "";
             value = String.valueOf(Show_All.getInt("Asset_ID"));
             value += " " + String.valueOf(Show_All.getInt("Threat_ID"));
```

```
              int chk = 0;
            //  out.println(AssetThreat.size() + "<br/>");
              int pos = -1;
             // if(AssetThreatList.size() <= 0){
               //      AssetThreatList.add(value);
              //} else {
               //     out.print(Show_All.getInt("Asset_ID") + " " + Show_All.getInt("Threat_ID") +
"<br/>");

                  if(AssetThreatList.indexOf(value) >= 0) {
               //  out.print("vv<br/>");
                  } else {

                    String sql2 = "select AssetThreat_Client_ID "
                            + "from tblbse_assetthreat_client "
                            + " where  Asset_ID=" + Show_All.getInt("Asset_ID")
                            + " and " + " Threat_ID=" + Show_All.getInt("Threat_ID");
                     ResultSet Show_All2 = database.query(sql2);
                     String dWhere = "";
                     int ClientCount =0;
                     while(Show_All2.next()) {
                      /////   out.println(Show_All2.getInt("AssetThreat_Client_ID") + "<br/>");
                        dWhere += " or AssetThreat_Client_ID=" +
Show_All2.getInt("AssetThreat_Client_ID");
                        ClientCount++;

                     }
                     //out.println("---" + ClientCount + "---<br><br>");
                     String smpName="";
                     String smpNameCol="";

                     for(int i=0; i<7; i++) {
                        switch(i){
                           case 0:
                              smpName="Discoverability";
                              smpNameCol="Discoverability";
                              break;
                           case 1:
                              smpName="Remediation";
                              smpNameCol="Remediation";
                              break;
                           case 2:
                              smpName="Availability Impact";
                              smpNameCol="Availability_Impact";
                              break;
                           case 3:
                              smpName="Confidentiality Impact";
                              smpNameCol="Confidentiality_Impact";
                              break;
                           case 4:
                              smpName="Integrity Impact";
                              smpNameCol="Integrity_Impact";
                              break;
                           case 5:
                              smpName="Availability Of Exploit";
                              smpNameCol="Exploitability";
                              break;
                           case 6:
                              smpName="Ease Of Exploitability";
                              smpNameCol="Ease_Of_Exploitation";
                              break;


                        }



                     // out.println(dWhere + " and (smpName='" + smpName + "')" + "<br/>");
                        String sql3 = "select * "
                               + "from tblbse_smpassthreat"
                               + " Where( AssetThreat_Client_ID=0 "
                               + dWhere + ")  and (smpName='" + smpName + "')";
                        //out.println(sql3 + "<br/>");
                        ResultSet Show_All3 = database.query(sql3);
                        //String dWhere = "";
                        int nClient=0;
                        String alpha = "ABCDEFGHIJKLMNOPQRSTUVWXYZ";
                        String dataStringValue="";
                        String dataStringKey="";
                        String dataStringClient="";
                        String dataString="";

                        while(Show_All3.next()) {
                            ///// out.println(Show_All3.getString("smpName") + "<br/>");

                           double smpAdd = 0;
                           for(int m=1; m<=10; m++){
                              smpAdd += Show_All3.getDouble("smp"+m);
                              // out.println("x---" +  Show_All3.getDouble("smp"+m) + "---x<br><br>");
                           }

                     /////        out.println("x---" + smpAdd  + "---x<br><br>");


                           dataStringValue +=  "-" + smpAdd + "";

                           dataStringKey += alpha.charAt(nClient);
                           dataStringClient +=  "-" + Show_All3.getString("Admin_ID");
                           nClient++;

                            }
                        //dataString = "'" + dataStringKey + "', " + dataStringValue ;
                     //out.println(dataString);
                     int AssetID = Show_All.getInt("Asset_ID");
                     int ThreatID = Show_All.getInt("Threat_ID");
//out.print(dataStringValue + " £££" + ThreatID+  " £££" +dataStringClient + "------<br/>");

                     out.println("<script  > report('" + AssetID + "', '" +
                              ThreatID + "', '" +
                              smpName + "', '" +
                              smpNameCol + "', '" +
                        dataStringKey + "', '" + dataStringValue +"', '" + dataStringClient +
"');</script>");

                        /* out.println("<script  > reportObjective('" + AssetID + "', '" +
                              ThreatID + "', '" +
                              smpName + "', '" +
                              smpNameCol  + "');</script>");
```

```
                        // int Client_ID = Integer.parseInt(database.Single_Value("tblbse_smpcertainty",
" AssetConf=" + AssetID +
                         //      " and Threat_ID=" + ThreatID +  " and smpName='" + smpName + "'",
"Client_ID"));

                         int Client_ID = 0;
                        Client_ID = database.Single_intValue("tblbse_smpcertainty",  " AssetConf=" +
AssetID +
                         " and Threat_ID=" + ThreatID +  " and smpName='" + smpName + "'",
"Client_ID");
                        out.println(Client_ID + " client<br/>");

                         int SlaveID = database.Single_intValue("tblbse_assetthreat_client",  "
Asset_ID=" + AssetID +
                               " and Threat_ID=" + ThreatID + " and Client_ID=" + Client_ID,
smpNameCol);

                         out.println(SlaveID + " slaveID" + AssetID + " " + ThreatID + "<br/>");
                         int smpPerception = database.Single_intValue("tblbse_slave",  " Slave_ID=" +
SlaveID, "Value");
                        // smpCertainty = Single_Value(String dTable,  String where_Column, String
ValueToReturn);
                         double smpCertainty = 0;
                         String smpCert =database.Single_Value("tblbse_smpcertainty",  " AssetConf="
+ AssetID +
                              " and Threat_ID=" + ThreatID +  " and smpName='" + smpName + "'",
"Certainty");

                         if(smpCert.equals("")) {
                            smpCertainty = 0;
                         } else {
                            smpCertainty =
Double.parseDouble(database.Single_Value("tblbse_smpcertainty",  " AssetConf=" + AssetID +
                              " and Threat_ID=" + ThreatID +  " and smpName='" + smpName + "'",
"Certainty"));
                         }


                        double Objective = smpCertainty * smpPerception;
                        out.println("<br/>" + smpName + " Value = Certainty * Perception : " +
smpCertainty +
                                      " * " + smpPerception + " = " + Objective + "xxxxx<br/>");
                       /// save Objective
                        int chksmp = database.Single_intValue("tblbse_smpcertainty", " AssetConf=" +
AssetID
                               + " and Threat_ID=" + ThreatID + " and smpName='" + smpName + "'",
"smpCertaintyID");
                         if(chksmp > 0) {

                           String UpdateSql = "Update tblbse_smpcertainty set Objective=" + Objective
                                      + " where smpCertaintyID=" + chksmp;
                           database.update(UpdateSql);

                         }**/
                        /// save Objective

                      } /// end smp name for
```

```
          /////    out.println(dWhere + "<br/>");
           AssetThreatList.add(value);

          }
        //}

      }
      //  out.println( "g<br/>");
//out.println( "gu<br/>");
       // /// Risk Of Exposure

        String sql3 = "select * from tblbse_assetthreat_client";
           ResultSet Show_AssThreat = database.query(sql3);
           while(Show_AssThreat.next()) {
            //   out.print("xx");
             int AssetID =  Show_AssThreat.getInt("Asset_ID");
             int ThreatID =  Show_AssThreat.getInt("Threat_ID");

             sql = "select * from tblbse_smpcertainty where AssetConf=" + AssetID
                 + " and  Threat_ID=" + ThreatID  +
                 "  and  (smpName='Discoverability' "
                 + "or  smpName='Remediation'"
                 + "or  smpName='Availability Impact'"
                 + "or  smpName='Confidentiality Impact'"
                 + "or  smpName='Integrity Impact'"
                 + "or  smpName='Availability Of Exploit'"
                 + "or  smpName='Ease Of Exploitability'"
                 + ") ";

             ResultSet res_smp = database.query(sql);
                  double RiskOfExposure =0;
                  double Exploitability =0;
                  double Damage =0;
                  double sumRCertainty =0;
                  double sumRObjective =0;
                  double sumECertainty =0;
                  double sumEObjective =0;
                  double sumDCertainty =0;
                  double sumDObjective =0;
                  String print = "";

                while(res_smp.next()) {
                  print = "x";
                  if (res_smp.getString("smpName").equals("Discoverability")
                      || res_smp.getString("smpName").equals("Remediation")) {

                   sumRCertainty += res_smp.getDouble("Certainty");
                   sumRObjective += res_smp.getDouble("Objective");

                  }

                  if (res_smp.getString("smpName").equals("Availability Of Exploit")
                      || res_smp.getString("smpName").equals("Ease Of Exploitability")) {

                   sumECertainty += res_smp.getDouble("Certainty");
                   sumEObjective += res_smp.getDouble("Objective");

                  }
```

```
                   if (res_smp.getString("smpName").equals("Availability Impact")
                        || res_smp.getString("smpName").equals("Integrity Impact")
                         || res_smp.getString("smpName").equals("Confidentiality Impact")) {

                     sumDCertainty += res_smp.getDouble("Certainty");
                     sumDObjective += res_smp.getDouble("Objective") ;

                   }

                // out.print(res_smp.getString("smpName") + (sumRObjective/sumRCertainty));
                  }
                    RiskOfExposure = sumRObjective/sumRCertainty;
                    Exploitability = sumEObjective/sumECertainty;
                    Damage = sumDObjective/sumDCertainty;

                  double victimThreatScore = 0;


                  int CatID = 
Integer.parseInt(database.Find_Slave_ReferenceByID("tbllkup_threats", "CategoryID", 
"Threat_ID", ThreatID));
                  int CatValue = 
Integer.parseInt(database.Find_Slave_ReferenceByID("tblbse_slave", "Value", "Slave_ID", 
CatID));
                 // out.println("val--" + CatValue + "--val");
                  victimThreatScore = (RiskOfExposure +  Exploitability + Damage)/ CatValue;

               //    out.print(RiskOfExposure + " " + Exploitability + " " + Damage + "<br/>");
                   out.println("<script  > reportThreatRating('" + ThreatID + "', '" + 
                    RiskOfExposure + "', '" +
                    Exploitability + "', '" +
                    Damage + "', '" +
                    victimThreatScore + "');</script>");
                 if(print.endsWith("x")){
                /////   out.println(AssetID + " " + ThreatID + " " + RiskOfExposure  + " " + 
Exploitability   + " " + Damage + " - " + CatValue + " - " + victimThreatScore + "<br/>");
                 }

             }

        /////


 out.println("<script  > document.write(loadReport());</script>");

         for(int i=0; i<AssetThreatList.size(); i ++){
         ///// out.println(AssetThreatList.get(i));
         }
        View += "</table>";
%>
 %@include  file="foot.jsp" %
```

Appendix 3: Snort and Suricata Event Reports for Plymouth University APT (Before Mitigation)

Table A: Snort Report for Plymouth University APT

DATE/TIME	SRCIP	SPORT	DST IP	D PORT	EVENT MESSAGE
19:43:45	10:1:0:3	44724	10:1:0:132	3306	ET POLICY SUSPICIOUS INBOUND TO MYSOL PORT 3306
19:43:45	10:1:0:3	44724	10:1:0:133	5900	ET SCAN POTENTIAL SSH SCAN 5900-5920
19:43:45	10:1:0:3	44724	10:1:0:133	22	ET SCAN POTENTIAL SSH SCAN OUTBOUND
19:43:45	10:1:0:3	44724	10:1:0:133	22	ET SCAN POTENTIAL SSH SCAN
19:43:45	10:1:0:3	44724	10:1:0:194	3389	ET DOS MICROSOFT REMOTE DESKTOP(RDP) SYN THEN RESET 30 SECOND DOS ATTEMPT
19:43:45	10:1:0:3	44724	10:1:0:228	1521	ET POLICY SUSPICIOUS INBOUND TO ORACLE SQL PORT 1521
19:43:47	10:1:0:3	44724	10:1:0:131	5800	ET SCAN POTENTIAL VNC SCAN 5800-5820

19:43:47	10:1:0:3	54802	10:1:0:131	53	GPL DNS NAMED VERSION ATTEMPT
19:43:47	10:1:0:3	54802	10:1:0:131	161	GPL SNMP PUBLIC ACCESS UDP
19:43:47	10:1:0:3	54802	10:1:0:131	111	GPL RPC PORTMAP LISTING UDP 111
19:43:47	10:1:0:3	16892	10:1:0:131	5632	GPL POLICY PC ANYWHERE SERVER RESPONSE
19:43:47	10:1:0:3	44724	10:1:0:229	1433	ET POLICY SUSPICIOUS INBOUND TO MSSQL PORT 1433
19:43:47	10:1:0:3	44724	10:1:0:229	5432	ET POLICY SUSPICIOUS INBOUND TO POSTRESQL PORT 5432
19:43:47	10:1:0:3	63246	10:1:0:229	33405	GPL SHELLCODE * 86 INC EBX NOOP
19:43:47	10:1:0:3	63246	10:1:0:229	33405	ET SCAN NMAP OS DETECTION PROBE
19:43:48	10:1:0:3	16914	10:1:0:133	139	GPL NETBIOS SMB IPC$ UNICODE SHARE ACCESS
19:43:48	10:1:0:3	16892	10:1:0:194	443	ET POLICY WINDOWS-BASED OPENSSL TUNNEL OUTBOUND
19:43:48	10:1:0:194	17018	10:1:0:194	135	ET SCAN BEHAVIORAL UNUSUAL PORT 135 TRAFFIC,POTENTIAL SCAN OR INFECTION
19:43:48	10:1:0:3	16898	10:1:0:228	445	GPL NETBIOS SMB IPC$ UNICODE SHARE ACCESS
19:43:49	10:1:0:3	63885	10:1:0:165	111	GPL RPC PORTMAP MOUNTD REQUEST UDP
19:43:49	10:1:0:3	17106	10:1:0:195	80	ET INFO GENERIC SUSPICIOUS POST TO DOTTED QUAD WITH FAKE BROWSER 1
19:43:55	10:1:0:99	43733	10:1:0:133	135	ET SCAN BEHAVIORAL UNUSUAL PORT 135 TRAFFIC,POTENTIAL SCAN OR INFECTION
19:43:58	10:1:0:99	43734	10:1:0:227	3389	ET SCAN UNUSUAL FAST TERMINAL SERVER TRAFFIC, POTENTIAL SCAN OR INFECTION
19:44:03	10:1:0:99	43734	10:1:0:227	3389	ET SCAN UNUSUAL FAST TERMINAL SERVER TRAFFIC, POTENTIAL

					SCAN OR INFECTION
19:44:07	10:1:0:99	43896	10:1:0:163	53	GPL DNS NAMED VERSION ATTEMPT
19:44:08	10:1:0:99	45957	10:1:0:131	445	ET SCAN BEHAVIORAL UNUSUAL PORT 445 TRAFFIC,POTENTIAL SCAN OR INFECTION
19:44:08	10:1:0:99	33743	10:1:0:133	445	GPL NETBIOS SMB-DS IPC$ share access
19:44:08	10:1:0:99	35210	10:1:0:195	80	ET SCAN NMAP SCRIPTING ENGINE USER AGENT DETECTED (NMAP SCRIPTING ENGINE)
19:44:09	10.1.0.66	38574	10.1.0.196	3306	ET POLICY SUSPICIOUS INBOUND TO MYSQL PORT 3306
19:44:10	10.1.0.66	38574	10.1.0.133	5800	ET SCAN POTENTIAL VNC SCAN 5800-5820
19:44:11	10.1.0.66	38574	10.1.0.164	1433	ET POLICY SUSPICIOUS INBOUND TO MSSQL PORT 1433
19:44:11	10.1.0.66	38574	10.1.0.195	1521	ET POLICY SUSPICIOUS INBOUND TO ORACLE SQL PORT 1521
19:44:11	10.1.0.66	38574	10.1.0.229	5432	ET POLICY SUSPICIOUS INBOUND TO POSTGRESQL PORT 5432
19:44:12	10.1.0.66	62685	10.1.0.131	161	GPL SNMP PUBLIC ACCESS UDP
19:44:12	10.1.0.66	62685	10.1.0.131	53	GPL DNS NAMED VERSION ATTEMPT
19:44:12	10.1.0.66	62685	10.1.0.163	111	GPL RPC PORTMAP LISTING 111
19:44:12	10.1.0.66	62685	10.1.0.163	5632	GPL POLICY PC ANYWHERE SERVER RESPONSE
19:44:12	10.1.0.66	36466	10.1.0.196	31837	ET SCAN NMAP OS DETECTION PROBE
19:44:13	10.1.0.66	1838	10.1.0.133	139	GPL NETBIOS SMB IPC & SHARE ACCESS
19:44:13	10.1.0.66	1829	10.1.0.227	445	GPL NETBIOS SMB-DS IPC & SHARE ACCESS
19:44:15	10.1.0.66	2039	10.1.0.164	80	ET INFO GENERIC SUSPICIOUS POST TO DOTTED QUAD WITH FAKE BROWSER 1
19:44:22	10.1.0.3	63986	10.1.0.163	445	ET SCAN BEHAVIORAL UNUSUAL PORT 445

					TRAFFIC,POTENTIAL SCAN OR INFECTION
19:44:43	10.1.0.34	29537	10.1.0.131	445	GPL NETBIOS SMB-DS ADMIN & UNICODE SHARE ACCESS
19:44:43	10.1.0.34	29559	10.1.0.131	445	GPL NETBIOS SMB-DS IPC & SHARE ACCESS
19:44:44	10.1.0.34	21546	10.1.0.133	5432	ET POLICY SUSPICIOUS INBOUND TO POSTGRESQL PORT 5432
19:44:45	10.1.0.34	26840	10.1.0.133	1433	ET POLICY SUSPICIOUS INBOUND TO MSSQL PORT 1433
19:44:47	10.1.0.34	53263	10.1.0.133	161	GPL SNMP PUBLIC ACCESS UDP
19:44:48	10.1.0.34	30563	10.1.0.133	445	GPL NETBIOS SMB-DS C & UNICODE SHARE ACCESS
19:44:48	10.1.0.34	30564	10.1.0.133	445	GPL NETBIOS SMB-DS C & UNICODE SHARE ACCESS
19:44:48	10.1.0.34	30564	10.1.0.133	445	GPL NETBIOS SMB-DS SESSION SETUP NTMLSSP UNICODE ASN1 OVERFLOW ATTEMPT
19:44:50	10.1.0.34	56536	10.1.0.133	5768	ET P2P EDONKEY PUBLICIZE FILE
19:44:52	10.1.0.34	51991	10.1.0.133	177	GPL RPC XDMCP INFO QUERY
19:44:52	10.1.0.34	59770	10.1.0.133	69	ET TFTP OUTBOUND TFTP READ REQUEST
19:45:03	10.1.0.34	37958	10.1.0.131	445	ET SCAN BEHAVIORAL UNUSUAL PORT 445 TRAFFIC,POTENTIAL SCAN OR INFECTION
19:45:11	10.1.0.34	23270	10.1.0.132	4333	ET POLICY SUSPICIOUS INBOUND TO MSSQL PORT 4333
19:45:12	10.1.0.34	55496	10.1.0.133	161	GPL SNMP PUBLIC ACCESS UDP
19:45:13	10.1.0.34	43703	10.1.0.132	1521	ET POLICY SUSPICIOUS INBOUND TO ORACLE SQL PORT 1521
19:45:17	10.1.0.34	16414	10.1.0.132	3389	ET DOS MICROSOFT REMOTE DESKTOP RDP SYN THEN RESET 30 SECOND DOS ATTEMPT
19:45:17	10.1.0.34	55504	10.1.0.133	123	ET DOS POSSIBLE NTP

					DDOS INBOUND FREQUENT UN-AUTHED MON LIST REQUEST IMPL 0….
19:45:23	10.1.0.34	18136	10.1.0.163	5803	ET SCAN POTENTIAL VNC SCAN 5800-5820
19:45:27	10.1.0.34	51859	10.1.0.163	3306	ET POLICY SUSPICIOUS INBOUND TO MYSQL PORT 3306
19:45:27	10.1.0.34	36801	10.1.0.163	5903	ET SCAN POTENTIAL VNC SCAN 5900-5920
19:45:36	10.1.0.34	42582	10.1.0.133	445	ET NETBIOS Microsoft SRV2.SYS SMB Negotiate ProcessID Function Table Dereference
19:45:42	10.1.0.34	48940	10.1.0.132	135	ET SCAN BEHAVIORAL UNUSUAL PORT 135 TRAFFIC,POTENTIAL SCAN OR INFECTION
19:46:00	10.1.0.34	44290	10.1.0.165	111	GPL RPC portmap cachefsd request TCP
19:46	10.1.0.34	44294	10.1.0.165	111	GPL RPC portmap listing TCP 111
19:46:01	10.1.0.34	44296	10.1.0.165	111	GPL RPC portmap rusers request TCP
19:46:01	10.1.0.34	44299	10.1.0.165	111	GPL RPC portmap rstatd request TCP
19:46:01	10.1.0.34	44301	10.1.0.165	111	GPL RPC portmap mountd request TCP
19:46:01	10.1.0.34	44310	10.1.0.165	111	GPL RPC portmap bootparam request TCP
19:46:02	10.1.0.34	44347	10.1.0.165	111	GPL RPC portmap ypserv request TCP
19:46:08	10.1.0.34	56943	10.1.0.132	22	ET SCAN POTENTIAL SSH SCAN OUTBOUND
19:46:08	10.1.0.34	56943	10.1.0.132	22	ET SCAN Potential SSH Scan
19:46:17	10.1.0.34	60707	10.1.0.165	161	ET SNMP Samsung Printer SNMP Hardcode RW Community String
19:46:18	10.1.0.34	11604	10.1.0.132	3389	ET SCAN Behavioral Unusually fast Terminal Server Traffic, Potential Scan or Infection (…
19:46:18	10.1.0.34	11604	10.1.0.132	3389	ET SCAN Behavioral Unusually fast Terminal Server Traffic, Potential Scan or infection (I…
19:46:18	10.1.0.34	34376	10.1.0.165	1900	GPL MISC UPnP

					malformed advertisement
19:46:18	10.1.0.34	45519	10.1.0.165	111	GPL RPC portmap ypupdated request TCP
19:46:18	10.1.0.34	45715	10.1.0.165	111	GPL RPC portmap snmpXdmi request TCP
19:46:18	10.1.0.34	45716	10.1.0.165	111	GPL RPC portmap yppasswd request TCP
19:46:20	10.1.0.34	47141	10.1.0.165	111	GPL RPC portmap sadmind request TCP
19:46:21	10.1.0.34	47225	10.1.0.165	111	GPL RPC portmap ttdbserv request TCP
19:46:25	10.1.0.34	48436	10.1.0.164	135	GPL NETBIOS DCERPC Iactivation little endian bind attempt
19:46:25	10.1.0.34	48436	10.1.0.164	135	GPL NETBIOS DCERPC Remote Activation bind attempt
19:46:25	10.1.0.34	53901	10.1.0.164	53	ET POLICY DNS Update From External net
19:46:26	10.1.0.34	48852	10.1.0.131	3389	ET POLICY RDP connection request
19:46:26	10.1.0.34	48939	10.1.0.163	53	GPL DNS named authors attempt
19:46:26	10.1.0.34	48697	10.1.0.164	25	ET EXPLOIT Possible SpamAssassin Milter Plugin Remote Arbitrary Command Injectio..
19:46:26	10.1.0.34	48706	10.1.0.164	25	GPL SMTP expn root
19:46:26	10.1.0.34	48706	10.1.0.164	25	GPL SMTP vrfy root
19:46:26	10.1.0.34	48826	10.1.0.164	8099	ET P2P GNUTella client request
19:46:26	10.1.0.34	48826	10.1.0.164	8099	GPL P2P GNUTella client request
19:46:27	10.1.0.34	49041	10.1.0.164	80	ET SCAN Nessus User Agent
19:46:27	10.1.0.34	57197	10.1.0.164	53	GPL DNS NAMED VERSION ATTEMPT
19:46:27	10.1.0.34	49283	10.1.0.164	21	ET SCAN Nessus FTP Scan detected (ftp_anonymous.nasl)
19:46:28	10.1.0.34	49373	10.1.0.164	80	GPL EXPLOIT ISAPI .ida access
19:46:28	10.1.0.34	49377	10.1.0.164	80	GPL EXPLOIT iissamples access
19:46:28	10.1.0.34	49377	10.1.0.164	80	GPL EXPLOIT ISAPI .idq access
19:46:29	10.1.0.34	49801	10.1.0.164	80	ET WEB_SERVER /system32/ in Uri - Possible Protected

					Directory Access Attempt
19:46:29	10.1.0.34	49801	10.1.0.164	80	ET WEB_SERVER cmd.exe In URI - Possible Command Execution Attempt
19:46:33	10.1.0.34	50755	10.1.0.164	80	ET INFO Executable Download from dotted-quad Host
19:46:36	10.1.0.34	51371	10.1.0.164	80	GPL EXPLOIT ISAPI .idq attempt
19:46:36	10.1.0.34	51549	10.1.0.164	80	GPL EXPLOIT iisadmpwd attempt
19:46:36	10.1.0.34	51549	10.1.0.164	80	GPL EXPLOIT .htr access
19:46:36	10.1.0.34	51553	10.1.0.164	80	GPL EXPLOIT /iisadmpwd/aexp2.htr access
19:46:37	10.1.0.34	51709	10.1.0.164	80	GPL WEB_SERVER .htaccess access
19:46:37	10.1.0.34	51734	10.1.0.164	80	ET WEB_SERVER Script tag in URI, Possible Cross Site Scripting Attempt
19:46:37	10.1.0.34	51749	10.1.0.164	80	GPL WEB_SERVER iisadmin access
19:46:48	10.1.0.34	63209	10.1.0.164	80	ET WEB_SERVER ColdFusion administrator access
19:46:49	10.1.0.34	63314	10.1.0.164	80	GPL WEB_SERVER global.asa access
19:46:49	10.1.0.34	63331	10.1.0.164	80	ET INFO GENERIC SUSPICIOUS POST TO DOTTED QUAD WITH FAKE BROWSER 1
19:46:49	10.1.0.34	63354	10.1.0.164	80	ET WEB_SERVER suhosin.simulation PHP config option in uri
19:46:49	10.1.0.34	63354	10.1.0.164	80	ET WEB SERVER ALLOW URL INCLUDE PHP CONFIG OPTION IN URI
19:46:49	10.1.0.34	63354	10.1.0.164	80	ET WEB SERVER ACCESS TO / PHPPATH/PHP POSSIBLE PLESK 0-DAY EXPLOIT JUNE 05 2013
19:46:49	10.1.0.34	63354	10.1.0.164	80	ET WEB SERVER SAFE MODE PHP CONFIG OPTION IN URI
19:46:49	10.1.0.34	63354	10.1.0.164	80	ET WEB SPECIFIC APPS PHP CGI QUERY STRING PARAMETER

					VULNERABILITY
19:46:49	10.1.0.34	63583	10.1.0.164	80	GPL WEB SERVER VIEWCODE ACCESS
19:46:50	10.1.0.34	63615	10.1.0.164	21	GPL FTP CWD ATTEMPT
19:46:50	10.1.0.34	63619	10.1.0.164	80	GPL EXPLOIT ALTERNATE DATA STREAMS ASP FILE ACCESS ATTEMPT
19:46:50	10.1.0.34	63619	10.1.0.164	80	ET WEB SERVER ALTERNATE DATA STREAM SOURCE VIEW ATTEMPT
19:47:17	10.1.0.34	5726	10.1.0.163	80	ET POLICY PROXY TRACE REQUEST-INBOUND
19:47:17	10.1.0.34	5726	10.1.0.163	80	GPL WEB SERVER TRACE ATTEMPT
19:47:17	10.1.0.34	5764	10.1.0.163	80	ET POLICY INCOMING BASIC AUTH BASE64 HTTP PASSWORD DETECTED UNENCRYPTED
19:47:17	10.1.0.34	5776	10.1.0.163	80	ET WORM THE MOON LINKSYS. ROUTER1
19:47:18	10.1.0.34	5850	10.1.0.163	80	GPL WEB SERVER PRINTENV ACCESS
19:47:18	10.1.0.34	6610	10.1.0.163	80	GPL WEB SERVER PERL COMMAND ATTEMPT
19:47:20	10.1.0.34	8483	10.1.0.163	80	GPL WEB SERVER PERL POST ATTEMPT
19:47:21	10.1.0.34	8554	10.1.0.163	80	GPL EXPLOIT FPCOUNT ACCESS
19:47:22	10.1.0.34	8841	10.1.0.163	80	GPL WEB SERVER MOD GZIP STATUS ACCESS
19:47:23	10.1.0.34	8947	10.1.0.132	3389	ET POLICY WINDOWS-BASED OPENSSL TUNNEL OUTBOUND
19:47:23	10.1.0.34	8922	10.1.0.163	80	GPL WEB SERVER WEB MISC JBOSS WEB CONSOLE ACCESS
19:47:23	10.1.0.34	9186	10.1.0.163	80	ET POLICY OUTGOING BASIC AUTH BASE 64 HTTP PASSWORD DETECTED UNENCRYPTED
19:51:09	10.1.0.134		10.1.0.131		GPL ICMP INFO PING BSDTYPE
19:51:09	10.1.0.134		10.1.0.131		GPL ICMP INFO PING NIS
19:51:10	10.1.0.134	49177	10.1.0.133	445	GPL NETBIOS SMB-DS IPC$ unicode share

					access
19:51:10	10.1.0.3	8080	10.1.0.135	49162	ET CURRENT_EVENTS landing page with malicious Java applet
19:51:10	10.1.0.3	8080	10.1.0.135	49163	ET CURRENT_EVENTS Possible Metasploit Java Payload
19:51:10	10.1.0.3	8080	10.1.0.135	49163	EY CURRENT_EVENTS Possible Metasploit Java Exploit
19:51:10	10.1.0.3	8080	10.1.0.135	49163	ET INFO JAVA - Java Archive Download By Vulnerable Client
19:51:10	10.1.0.3	1024	10.1.0.135	49164	ET TROJAN Metasploit Meterpreter stdapi_*Command Request
19:51:10	**10.1.0.3**	**1024**	**10.1.0.197**	**49174**	**ET TROJAN Metasploit Meterpreter core_channel_* Command Request**
19:51:13	**10.1.0.35**		**10.1.0.133**		**GPL ICMP_INFO PING *NIX**
19:51:13	**10.1.0.134**	**17500**	**10.1.0.225**	**17500**	**ET POLICY Dropbox Client Broadcasting**
19:51:15	**10.1.0.3**	**8080**	**10.1.0.194**	**27497**	**ET INFO JAVA - Java Archive Download**
19:44:09	**10.1.0.99**	**40227**	**10.1.0.165**	**111**	**GPL RPC portmap listing TCP 111**
19:44:09	**10.1.0.66**	**38574**	**10.1.0.165**	**5910**	**ET SCAN POTENTIAL VNC SCAN 5900-5920**

Note: Bolded records of event are False Positive events

Table B: Suricata Report for Plymouth University APT

DATE/TIME	**SRC IP**	**SPORT**	**DST IP**	**DPORT**	**EVENT MESSAGE**
19:40:09	**10.1.0.3**	**44724**	**10.1.0.132**	**3306**	**ET POLICY SUSPICIOUS INBOUND TO MY SQL PORT 3306**
19:40:09	**10.1.0.3**	**44724**	**10.1.0.133**	**5900**	**ET SCAN POTENTIAL VNC SCAN 5900-5920**
19:40:09	**10.1.0.3**	**44724**	**10.1.0.133**	**22**	**ET SCAN POTENTIAL SSH SCAN OUTBOUND**
19:40:09	**10.1.0.3**	**44724**	**10.1.0.133**	**22**	**ET SCAN POTENTIAL SSH SCAN**
19:40:09	**10.1.0.3**	**44724**	**10.1.0.194**	**3389**	**ET DOS MICROSOFT REMOTE DESKTOP (RDP) SYN THEN RESET 30 SECOND DOS SECOND ATTEMPT**

19:40:10	10.1.0.3	44724	10.1.0.228	1521	ET POLICY SUSPICIOUS INBOUND TO ORACLE SQL PORT 1521
19:40:12	10.1.0.3	44724	10.1.0.131	5432	ET POLICY SUSPICIOUS INBOUND TO POSTGRESQL PORT 5432
19:40:12	10.1.0.3	44724	10.1.0.229	1433	ET POLICY SUSPICIOUS INBOUND TO MSSQL PORT 1433
19:40:13	10.1.0.3	54802	10.1.0.131	53	GPL DNS NAMED VERSION ATTEMPT
19:40:13	10.1.0.3	54802	10.1.0.131	161	GPL SNMP PUBLIC ACCESS UDP
19:40:13	10.1.0.3	54802	10.1.0.131	111	GPL RPC PORTMAP LISTING UDP 111
19:40:13	10.1.0.3	54802	10.1.0.131	5632	GPL POLICY PC ANYWHERE SERVER RESPONSE
19:40:13	10.1.0.3	16914	10.1.0.133	139	GPL NETBIOS SMB IPC & UNICODE SHARE ACCESS
19:40:13	10.1.0.3	16892	10.1.0.194	443	ET POLICY WINDOW-BASED OPEN SSL TUNNEL OUTBOUND
19:40:13	10.1.0.3	16898	10.1.0.228	445	GPL NETBIOS SMB-DS SESSION SETUP NTMLSSP ASN1 OVERFLOW ATTEMPT
19:40:13	10.1.0.3	16898	10.1.0.228	445	GPL NETBIOS SMB IPC & UNICODE SHARE ACCESS
19:40:13	10.1.0.3	44724	10.1.0.229	5800	ET SCAN POTENTIAL VNC SCAN 5800-5820
19:40:13	10.1.0.3	63246	10.1.0.229	33405	GPL SHELLCODEx86 inc ebx noop
19:40:13	10.1.0.3	63246	10.1.0.229		ET SCAN NMAP OS DETECTION PROBE
19:40:14	10.1.0.3	17018	10.1.0.194	135	ET SCAN BEHAVIORAL UNUSUAL PORT 135 TRAFFIC, POTENTIAL SCAN OF INFECTION
19:40:21	10.1.0.99	43733	10.1.0.131	135	ET SCAN BEHAVIORAL UNUSUAL PORT 135 TRAFFIC, POTENTIAL SCAN OF INFECTION
19:40:23	10.1.0.99	43734	10.1.0.227	3389	ET SCAN Behavioral Unusally fast Terminal Server Traffic, Potential Scan or Infection (…
19:40:23	10.1.0.99	43734	10.1.0.227	3389	ET SCAN BEHAVIORAL UNUSUALY FAST TERMINAL SERVER TRAFFIC, POTENTIAL SCAN

					OR INFECTION
19:40:28	10.1.0.99	41777	10.1.0.132	53	GPL DNS NAMED VERSION ATTEMPT
19:40:31	10.1.0.99	35210	10.1.0.195	80	ET SCAN NMAP SCRIPTING ENGINE USER-AGENT DETECTED(NMAP SCRIPTING ENGINE)
19:40:32	10.1.0.99	45912	10.1.0.131	445	ET SCAN BEHAVIORAL UNUSUAL PORT 445 TRAFFIC, POTENTIAL SCAN OR INFECTION
19:40:32	10.1.0.99	33743	10.1.0.133	445	GPL NETBIOS SMB-DS IPC $ SHARE ACCESS
19:40:32	10.1.0.99	40227	10.1.0.165	111	GPL RPC PORTMAP LISTING TCP 111
19:40:33	10.1.0.66	38574	10.1.0.132	3389	ET DOS MICROSOFT REMOTE DESKTOP (RDP) SYN THEN RESET 30 SECOND DOS SECOND ATTEMPT
19:40:33	10.1.0.66	38574	10.1.0.195	3306	ET POLICY SUSPICIOUS INBOUND TO MY SQL PORT 3306
19:40:33	10.1.0.66	38574	10.1.0.229	5900	ET SCAN POTENTIAL VNC SCAN 5900-5920
19:40:33	10.1.0.66	38574	10.1.0.229	22	ET SCAN POTENTIAL SSH SCAN OUTBOUND
19:40:33	10.1.0.66	38574	10.1.0.229	22	ET SCAN POTENTIAL SSH SCAN
19:40:34	10.1.0.66	38574	10.1.0.133	5800	ET SCAN POTENTIAL VNC SCAN 5800-5820
19:40:34	10.1.0.66	38574	10.1.0.164	1433	ET POLICY SUSPICIOUS INBOUND TO MSSQL PORT 1433
19:40:35	10.1.0.66	38574	10.1.0.164	5432	ET POLICY SUSPICIOUS INBOUND TO POSTGRESQL PORT 5432
19:40:35	10.1.0.66	38574	10.1.0.195	1521	ET POLICY SUSPICIOUS INBOUND TO ORACLE SQL PORT 1521
19:40:36	10.1.0.66	62685	10.1.0.131	161	GPL SNMP PUBLIC ACCESS UDP
19:40:36	10.1.0.66	62685	10.1.0.131	53	GPL DNS NAMED VERSION ATTEMPT
19:40:36	10.1.0.66	62685	10.1.0.131	111	GPL RPC PORTMAP LISTING UDP 111
19:40:36	10.1.0.66	62685	10.1.0.131	5632	GPL POLICY PC ANYWHERE SERVER RESPONSE
19:40:36	10.1.0.66	1822	10.1.0.131	445	GPL NETBIOS SMB-DS IPC $ SHARE ACCESS

19:40:36	10.1.0.66	1838	10.1.0.133	139	GPL NETBIOS SMB IPC & UNICODE SHARE ACCESS
19:40:36	10.1.0.66	1820	10.1.0.195	445	GPL NETBIOS SMB-DS SESSION SETUP NTMLSSP ASN1 OVERFLOW ATTEMPT
19:40:36	10.1.0.66		10.1.0.229		SURICATA ICMPv4 UNKNOWN CODE
19:40:36	10.1.0.66	36466	10.1.0.229	43071	ET SCAN NMAP OS DETECTION PROBE
19:40:37	10.1.0.66	1908	10.1.0.132	139	ET SCAN BEHAVIORAL UNUSUAL PORT 135 TRAFFIC, POTENTIAL SCAN OF INFECTION
19:40:38	10.1.0.66	2039	10.1.0.164	80	ET INFO GENERIC SUSPICIOUS POST TO DOTTED QUAD WITH FAKE BROWSER 1
19:40:38	10.1.0.66	59329	10.1.0.165	111	GPL RPC PORTMAP MOUNTED REQUEST UDP
19:40:38	10.1.0.66	1996	10.1.0.194	902	SURICATA TLS INVALID RECORD TYPE
19:40:41	10.1.0.3	63976	10.1.0.132	143	ET SCAN Rapid IMAP Connections - Possible Brute Force Attack
19:40:41	10.1.0.3	63976	10.1.0.163	110	ET SCAN Rapid POP3 Connections - Possible Brute Force Attack
19:40:41	10.1.0.3	63975	10.1.0.226	3389	ET SCAN Behavioral Unusally fast Terminal Server Traffic, Potential Scan or Infection (…
19:40:41	10.1.0.3	63975	10.1.0.226	3389	ET SCAN Behavioral Unusally fast Terminal Server Traffic, Potential Scan or infection (I…
19:40:41	10.1.0.3	63976	10.1.0.227	993	ET SCAN Rapid IMAPS Connections - Possible Brute Force Attack
19:40:41	10.1.0.3	63976	10.1.0.227	995	ET SCAN Rapid POP3S Connections - Possible Brute Force Attack
19:40:41	10.1.0.3	63975	10.1.0.229	445	ET SCAN Behavioral Unusual Port 445 traffic, POTENTIAL SCAN OR INFECTION
19:40:44	10.1.0.3	54759	10.1.0.133	139	ET SCAN Behavioral Unusual Port 135 Traffic, Potential Scan or Infection
19:40:51	10.1.0.99	62554	10.1.0.133	110	ET SCAN Rapid POP3

					Connections - Possible Brute Force Attack
19:40:51	10.1.0.99	62554	10.1.0.164	995	ET SCAN Rapid POP3S Connections - Possible Brute Force Attack
19:40:51	10.1.0.99	62554	10.1.0.164	143	ET SCAN Rapid IMAP Connections - Possible Brute Force Attack
19:40:51	10.1.0.99	62554	10.1.0.195	993	ET SCAN Rapid IMAPS Connections - Possible Brute Force Attack
19:40:52	10.1.0.99	62554	10.1.0.133	1433	ET SCAN Behavioral Unusual Port 1433 traffic, Potential Scan or Infection
19:40:57	10.1.0.99	62555	10.1.0.163	1434	et scan behavioral unsual port 1434 traffic potential scan or infection
19:41:00	10.1.0.99	39599	10.1.0.228	19	suricata stream 3way handshake right seg wrong ack evasion
19:41:03	10.1.0.34	60067	10.1.0.131	161	GPL SNMP PUBLIC ACCESS UDP
19:41:03	10.1.0.34	29475	10.1.0.133	445	GPL NETBIOS SMB-DS SESSION SETUP NTMLSSP ASN1 OVERFLOW ATTEMPT
19:41:03	10.1.0.34	29475	10.1.0.133	445	GPL NETBIOS SMB-DS IPC & UNICODE SHARE ACCESS
19:41:03	10.1.0.34	29492	10.1.0.133	445	GPL NETBIOS SMB-DS ADMIN & UNICODE SHARE ACCESS
19:41:06	10.1.0.34	26840	10.1.0.133	1433	ET POLICY SUSPICIOUS INBOUND TO MSSQL PORT 1433
19:41:07	10.1.0.34	47755	10.1.0.133	4333	ET POLICY SUSPICIOUS INBOUND TO MSSQL PORT 4333
19:41:07	10.1.0.34	12233	10.1.0.133	1521	ET POLICY SUSPICIOUS INBOUND TO ORACLE PORT 1521
19:41:08	10.1.0.34	30483	10.1.0.133	445	GPL NETBIOS SMB-DOS & UNICODE SHARE ACCESS
19:41:10	10.1.0.34	30563	10.1.0.133	445	GPL NETBIOS SMB-DOS & UNICODE SHARE ACCESS
19:41:13	10.1.0.34	30574	10.1.0.131	1027	suricata stream shutdown rst invalid ack
19:41:13	10.1.0.34	30574	10.1.0.131	1027	SURICATA STREAM SHUTDOWN PACKET WITH INVALID ACK
19:41:13	10.1.0.34	35123	10.1.0.131	1027	SURICATA STREAM 3WAY

					HANDSHAKE WRONG SEQ WRONG ACK
19:41:13	10.1.0.34	51991	10.1.0.133	177	GPL RPC XDMCP INFO QUERY
19:41:14	10.1.0.34	53940	10.1.0.133	69	ET TFTP Outbound TFTP Read Request
19:41:14	10.1.0.34	52807	10.1.0.133	1900	GPL MISC UPnP malformed advertisement
19:41:14	10.1.0.34	35323	10.1.0.164	445	ET SCAN BEHAVIORAL UNUSUAL PORT 445 TRAFFIC, POTENTIAL SCAN OR INFECTION
19:41:14	10.1.0.34	35332	10.1.0.164	1043	SURICATA STREAM ESTABLISHED invalid ack
19:41:14	10.1.0.34	26727	10.1.0132	3306	ET POLICY Suspicious inbound to mySQL port 3306
19:41:31	10.1.0.34	55496	10.1.0.133	161	GPL SNMP PUBLIC ACCESS UDP
19:41:35	10.1.0.34	16414	10.1.0.132	3389	ET DOS MICROSOFT REMOTE DESKTOP (RDP) SYN THEN RESET 30 SECOND DOS ATTEMPT
19:41:35	10.1.0.34	55504	10.1.0.133	123	ET DOS Possible NTP DDoS Inbound Frequent Un-Authed MON_LIST Requests IMPL 0...
19:41:37	10.1.0.34		10.1.0.133		SURICATA ICMPv4 unknown type
19:41:43	10.1.0.34	25446	10.1.0.163	5432	ET POLICY SUSPICIOUS INBOUND TO POSTGRESQL PORT 5432
19:41:43	10.1.0.34	18136	10.1.0.163	5803	ET SCAN POTENTIAL VNC SCAN 5800-5820
19:41:46	10.1.0.34	36801	10.1.0.163	5903	ET SCAN POTENTIAL VNC SCAN 5900-5920
19:41:57	10.1.0.34	42582	10.1.0.133	445	ET NETBIOS Microsoft SRV2.SYS SMB Negotiate ProcessID Function Table Dereference
19:42:27	10.1.0.34	44327	10.1.0.165	111	GPL RPC portmap bootparam request TCP
19:42:36	10.1.0.34	23362	10.1.0.163	22	ET SCAN POTENTIAL SSH SCAN OUTBOUND
19:42:36	10.1.0.34	23362	10.1.0.163	22	ET SCAN Potential SSH Scan
19:42:36	10.1.0.34	58705	10.1.0.165	5768	ET P2P Edonkey Publicize File
19:42:43	10.1.0.34	60707	10.1.0.165	161	ET SNMP Samsung Printer SNMP Hardcode RW

					Community String
19:42:43	10.1.0.34	45519	10.1.0.165	111	GPL RPC portmap ypupdated request TCP
19:42:43	10.1.0.34	45715	10.1.0.165	111	GPL RPC portmap snmpXdmi request TCP
19:42:43	10.1.0.34	45716	10.1.0.165	111	GPL RPC portmap yppasswd request TCP
19:42:45	10.1.0.34	47141	10.1.0.165	111	GPL RPC portmap sadmind request TCP
19:42:46	10.1.0.34	47225	10.1.0.165	111	GPL RPC portmap ttdbserv request TCP
19:42:46	10.1.0.34	47291	10.1.0.165	111	GPL RPC portmap mounted request TCP
19:42:50	10.1.0.34	48552	10.1.0.131	902	SURICATA TLS INVALID RECORD TYPE
19:42:50	10.1.0.34	48436	10.1.0.164	135	GPL NETBIOS DCERPC Iactivation little endian bind attempt
19:42:50	10.1.0.34	48436	10.1.0.164	135	GPL NETBIOS DCERPC Remote Activation bind attempt
19:42:50	10.1.0.34	53901	10.1.0.164	53	ET POLICY DNS Update From Exernal net
19:42:50	10.1.0.34	48697	10.1.0.164	25	ET EXPLOIT Possible SpamAssassin Milter Plugin Remote Arbitrary Command Injectio…
19:42:50	10.1.0.34	48706	10.1.0.164	25	GPL SMTP vrfy root
19:42:50	10.1.0.34	48706	10.1.0.164	25	GPL SMTP expn root
19:42:51	10.1.0.34	48852	10.1.0.131	3389	ET SCAN Behavioral Unusually fast Terminal Server Traffic, Potential Scan or Infection (…
19:42:51	10.1.0.34	48852	10.1.0.131	3389	ET SCAN Behavioral Unusually fast Terminal Server Traffic, Potential Scan or Infection (I…
19:42:51	10.1.0.34	48852	10.1.0.131	3389	ET POLICY RDP connection request
19:42:51	10.1.0.34	48826	10.1.0.164	8099	GPL P2P GNUTella client request
19:42:51	10.1.0.34	48826	10.1.0.164	8099	ET P2P GnuTella Connect
19:42:55	10.1.0.34	50342	10.1.0.164	80	ET INFO GENERIC SUSPICIOUS POST TO DOTTED QUAD WITH FAKE BROWSER 1
19:42:56	10.1.0.34	50438	10.1.0.164	80	GPL EXPLOIT fpcount access
19:42:57	10.1.0.34	50899	10.1.0.164	80	GPL WEB_SERVER WEB-MISC Jboss web-console

					access
19:43:01	10.1.0.34	51736	10.1.0.164	80	GPL WEB_SERVER .htaccess access
19:43:53	10.1.0.34	10208	10.1.0.163	80	GPL WEB_SERVER iisadmin access
19:44:06	10.1.0.34	11309	10.1.0.132	80	GPL WEB_SERVER mod_gzip_status access
19:44:10	10.1.0.34	11662	10.1.0.132	80	ET POLICY Outgoing Basic Auth Base64 HTTP Password detected unencrypted
19:44:22	10.1.0.34	12675	10.1.0.132	80	GPL EXPLOIT iisadmpwd attempt
19:44:22	10.1.0.34	12675	10.1.0.132	80	GPL EXPLOIT .htr access
19:44:29	10.1.0.34	13217	10.1.0.163	80	ET WEB_SERVER ColdFusion administrator access
19:44:31	10.1.0.34	13325	10.1.0.194	445	SURICATA STREAM ESTABLISHED packet out of window
19:44:32	10.1.0.34	13390	10.1.0.163	80	GPL EXPLOIT Alternate Data streams ASP file access attempt
19:44:32	10.1.0.34	13390	10.1.0.163	80	ET WEB_SERVER Alternate Data Stream source view attempt
19:44:50	10.1.0.34	16457	10.1.0.132	80	ET WEB_SPECIFIC_APPS PHP-CGI query string parameter vulnerability
19:44:50	10.1.0.34	16457	10.1.0.132	80	ET WEB_SERVER Access to/phppath/php Possible Plesk 0-day Exploit June 05 2013
19:44:50	10.1.0.34	16457	10.1.0.132	80	ET WEB_SERVER auto_prepend_file PHP config option in uri
19:44:50	10.1.0.34	16457	10.1.0.132	80	ET WEB_SERVER open_basedir PHP config option in uri
19:44:50	10.1.0.34	16457	10.1.0.132	80	ET WEB_SERVER disable_functions PHP config option in uri
19:44:50	10.1.0.34	16457	10.1.0.132	80	ET WEB_SERVER suhosin.simulation PHP config option in uri
19:44:50	10.1.0.34	16457	10.1.0.132	80	ET WEB_SERVER safe_mode PHP config option in uri
19:44:50	10.1.0.34	16457	10.1.0.132	80	ET WEB_SERVER allow_url_include PHP

					config option in uri
19:44:50	10.1.0.34	16457	10.1.0.132	80	ET WEB_SERVER PHP tags in HTTP POST
19:44:53	10.1.0.34	16890	10.1.0.132	80	GPL WEB_SERVER viewcode access
19:44:58	10.1.0.34	18910	10.1.0.194	3389	ET POLICY Windows-Based OpenSSL Tunnel Outbound
19:45:03	10.1.0.34	20622	10.1.0.194	443	SURICATA TLS invalid handshake message
19:45:34	10.1.0.34	37545	10.1.0.195	21	GPL FTP CWD ~ attempt
19:45:43	10.1.0.34	42211	10.1.0.226	135	ET SCAN Behavioral Unusual Port 135 Traffic, Potential Scan or Infection
19:47:10	10.1.0.34	55274	10.1.0.227	80	GPL EXPLOIT ISAPI .idq attempt
19:47:11	10.1.0.34	55519	10.1.0.227	80	GPL EXPLOIT /iisadmpwd/aexp2.htr access
19:47:24	10.1.0.34	60539	10.1.0.228	8099	GPL WEB_SERVER Tomcat directory traversal attempt
19:47:42	10.1.0.134		10.1.0.131		GPL ICMP_INFO PING BSDtype
19:47:42	10.1.0.134		10.1.0.131		GPL ICMP_INFO PING *NIX
19:47:42	10.1.0.3	8080	10.1.0.135	49162	ET CURRENT_EVENTS landing page with malicious Java applet
19:47:42	10.1.0.3	8080	10.1.0.135	49163	ET INFO JAVA - Java Archive Download By Vulnerable Client
19:47:42	10.1.0.3	8080	10.1.0.135	49163	ET CURRENT_EVENTS Possible Metasploit Java Exploit
19:47:42	10.1.0.3	8080	10.1.0.135	49163	ET CURRENT_EVENTS Possible Metasploit Java Payload
19:47:42	10.1.0.3	1024	10.1.0.135	49164	ET TROJAN Metasploit Meterpreter stdapi_*Command Request
19:47:43	10.1.0.134	49177	10.1.0.133	445	GPL NETBIOS SMB-DS IPC$ unicode share access
19:47:43	10.1.0.3	1024	10.1.0.197	49174	ET TROJAN Metasploit Meterpreter core_channel_*Command Request
19:47:46	10.1.0.134	17500	10.1.0.225	17500	ET POLICY Dropbox Client Broadcasting
19:47:46	10.1.0.35		10.1.0.133		GPL ICMP_INFO PING *NIX
19:47:47	10.1.0.3	8080	10.1.0.194	27505	ET INFO JAVA - Java Archive Download

Note: Bolded records of event are False Positive events

Appendix 4: Snort and Suricata Event Reports for MIT Lincoln Lab LLDOS 1.0 Botnet Threat (Before Mitigation)

Table C: Snort Report for MIT Lincoln Lab LLDOS 1.0

Event_ Date	src_ Port	src_IPAddress	Des_ Port	Des_IPAddress	Event_Name
14:20:58	**3103**	**172.16.116.20**	**139**	**172.16.112.10**	**NETBIOS**

				0	
14:20:59	**1756**	**172.16.112.100**	**139**	**172.16.116.20**	**NETBIOS**
14:21:00	**21**	**172.16.112.100**	**8242**	**172.16.113.204**	**FTP_Login**
14:21:02	**46899**	**196.227.33.189**	**25**	**172.16.112.149**	**Frequent_Emails**
14:21:05	60249	202.77.162.213	111	172.16.115.20	RPC_portmap_sadmind
14:21:05	60253	202.77.162.213	111	172.16.115.20	RPC_portmap_sadmind
14:21:05	60251	202.77.162.213	32773	172.16.115.20	RPC_sadmind_query
14:21:06	60267	202.77.162.213	111	172.16.115.20	RPC_portmap_sadmind
14:21:06	60255	202.77.162.213	32773	172.16.115.20	RPC_sadmind_query
14:21:06	60269	202.77.162.213	32773	172.16.115.20	RPC_sadmind_query
14:21:07	60274	202.77.162.213	111	172.16.115.20	RPC_portmap_sadmind
14:21:07	60287	202.77.162.213	111	172.16.115.20	RPC_portmap_sadmind
14:21:07	60276	202.77.162.213	32773	172.16.115.20	RPC_sadmind_query
14:21:07	60289	202.77.162.213	32773	172.16.115.20	RPC_sadmind_query
14:21:08	**60517**	**202.77.162.213**	**111**	**172.16.112.10**	**RPC_portmap_sadmind**
14:21:08	**60522**	**202.77.162.213**	**111**	**172.16.112.10**	**RPC_portmap_sadmind**
14:21:08	60298	202.77.162.213	111	172.16.115.20	RPC_portmap_sadmind
14:21:08	60300	202.77.162.213	32773	172.16.115.20	RPC_sadmind_query
14:21:08	**60519**	**202.77.162.213**	**32774**	**172.16.112.10**	**RPC_sadmind_query**
14:21:08	**60524**	**202.77.162.213**	**32774**	**172.16.112.10**	**RPC_sadmind_query**
14:21:09	**60540**	**202.77.162.213**	**111**	**172.16.112.10**	**RPC_portmap_sadmind**

14:21:09	60547	202.77.162.213	111	172.16.112.10	RPC_portmap_sadmind
14:21:09	60567	202.77.162.213	111	172.16.112.50	RPC_portmap_sadmind
14:21:09	60542	202.77.162.213	32774	172.16.112.10	RPC_sadmind_query
14:21:09	60549	202.77.162.213	32774	172.16.112.10	RPC_sadmind_query
14:21:10	60576	202.77.162.213	111	172.16.112.50	RPC_portmap_sadmind
14:21:10	60603	202.77.162.213	111	172.16.112.50	RPC_portmap_sadmind
14:21:10	60569	202.77.162.213	32773	172.16.112.50	RPC_sadmind_query
14:21:10	60578	202.77.162.213	32773	172.16.112.50	RPC_sadmind_query
14:21:10	60605	202.77.162.213	32773	172.16.112.50	RPC_sadmind_query
14:21:11	60617	202.77.162.213	111	172.16.112.50	RPC_portmap_sadmind
14:21:11	60619	202.77.162.213	32773	172.16.112.50	RPC_sadmind_query
14:21:13	3104	172.16.116.20	139	172.16.112.100	NETBIOS
14:21:13	1761	172.16.112.100	139	172.16.116.20	NETBIOS
14:28:24	21	172.16.112.100	2067	172.16.113.169	FTP_Login
14:28:27	21	172.16.112.100	2827	172.16.113.84	FTP_Login
14:28:39	48713	197.182.91.233	25	172.16.112.50	Frequent_Emails
14:28:39	48691	194.27.251.21	25	172.16.112.149	Frequent_Emails
14:28:41	48766	196.37.75.158	25	172.16.113.169	Frequent_Emails
14:28:41	21	172.16.112.100	9019	172.16.113.84	FTP_Login
14:28:42	3107	172.16.116.20	139	172.16.112.100	NETBIOS
14:28:44	21	172.16.112.100	9592	172.16.112.149	FTP_Login
14:28:4	4886	195.73.151.50	25	172.16.112.14	Frequent_Emails

5	2			9	
14:28:53	21	172.16.112.100	12415	172.16.112.207	FTP_Login
14:28:55	3108	172.16.116.20	139	172.16.112.100	NETBIOS
14:28:56	1779	172.16.112.100	139	172.16.116.20	NETBIOS
14:28:57	33796	172.16.115.20	161	172.16.112.105	SNMP_public_access_udp
14:29:29		172.16.115.20	204	172.16.113.204	ICMP_BSDtype
14:29:29		172.16.115.20	204	172.16.113.204	ICMP_NIX
14:29:44	3109	172.16.116.20	139	172.16.112.100	NETBIOS
14:29:45	1784	172.16.112.100	139	172.16.116.20	NETBIOS
14:30:00	23	172.16.112.50	28483	172.16.113.204	TELNET_Bad_Login

Note: Bolded records of event are False Positive events

Table D: Suricata Report for MIT Lincoln Lab LLDOS 1.0

DATE/TIME	SRC IP	Sport	DST IP	DPORT	EVENT MESSAGE
13:30:41	135.8.60.182	1170	172.16.114.148	25	ET POLICY Inbound Frequent Emails - Possible

					Spambot Inbound
13:30:42	172.16.112.20	53	172.16.112.100	1434	ET EXPLOIT MS - SQL DOS attempt (08)
13:30:47	194.7.248.153	2120	172.16.112.149	25	ET POLICY Inbound Frequent Emails - Possible Spambot Inbound
13:30:47	135.13.216.191	2139	172.16.113.204	25	ET POLICY Inbound Frequent Emails - Possible Spambot Inbound
13:30:48	195.73.151.50	2539	172.16.113.50	25	ET POLICY Inbound Frequent Emails - Possible Spambot Inbound
13:30:48	197.182.91.233	2299	172.16.114.169	25	ET POLICY Inbound Frequent Emails - Possible Spambot Inbound
13:30:49	172.16.115.234	1061	172.16.112.100	139	GPL NETBIOS NT NULL session
13:30:49	195.115.218.108	2556	172.16.113.84	25	ET POLICY Inbound Frequent Emails - Possible Spambot Inbound
13:30:50	197.218.177.69	2611	172.16.114.168	25	ET POLICY Inbound Frequent Emails - Possible Spambot Inbound
13:30:51	172.16.113.50	23	172.16.114.168	4926	GPL TELNET Bad Login
13:30:52	196.227.33.189	3171	172.16.112.194	25	ET POLICY Inbound Frequent Emails - Possible Spambot

					Inbound
13:30:52	196.37.75.158	3248	172.16.113.105	25	ET POLICY Inbound Frequent Emails - Possible Spambot Inbound
13:30:56	192.168.1.30	32771	172.16.112.5	161	GPL SNMP public access udp
13:30:59	194.27.251.21	4126	172.16.114.207	25	ET POLICY Inbound Frequent Emails - Possible Spambot Inbound
13:31:00	192.168.1.20	6667	172.16.113.84	8253	ET CHAT IRC PING command
13:31:26	205.181.112.65	80	172.16.115.87	5552	SURICATA HTTP response field missing colon
13:31:30	172.16.112.50	23	172.16.114.148	10197	GPL TELNET Bad login
13:31:33	172.16.114.50	23	172.16.112.194	13581	GPL TELNET Bad login
13:32:49	216.32.120.136	80	172.16.115.5	8818	SURICATA HTTP response field missing colon
13:34:05	207.46.185.11	80	172.16.115.5	30534	SURICATA HTTP response field missing colon
13:34:22	172.16.113.50		172.16.113.105		GPL ICMP_INFO PING BSDtype
13:34:22	172.16.113.50		172.16.113.105		GPL ICMP_INFO PING *NIX
13:35:44	172.16.112.50		172.16.114.169		GPL ICMP_INFO PING BSDtype
13:35:44	172.16.112.50		172.16.114.169		GPL ICMP_INFO PING *NIX
13:36:30	132.60.168.152	80	172.16.112.207	2122	ET POLICY PE EXE or DLL Windows file download
13:37:42	208.209.46.36	80	172.16.114.168	27772	SURICATA HTTP unknown error
13:41:37	172.16.116.20	3098	172.16.112.100	139	GPL NETBIOS NT NULL session
13:41:37	172.16.115.20	53	172.16.112.20	1434	ET EXPLOIT MS-SQL DOS ATTEMPT (08)
13:41:58	172.16.112.100	2134	172.16.112.20	139	GPL NETBIOS NT

					NULL session
13:42:07	172.16.115.20	33732	172.16.112.105	161	GPL SNMP public access udp
13:42:19	202.77.162.213	54790	172.16.115.20	111	GPL RPC PORTMAP SADMIND REQUEST UDP
13:42:52	202.77.162.213	60251	172.16.115.20	32773	GPL RPC Sadmind query with root credentials attempt UDP
13:44:17	**172.16.115.20**		**172.16.113.204**		**GPL ICMP_INFO PING BSDtype**
13:44:17	**172.16.115.20**		**172.16.113.204**		**GPL ICMP_INFO PING *NIX**

Note: Bolded records of events are False Positive events

Appendix 5: Threat Prediction Model Output for Plymouth University APT

```
Steps .........1............

AJ 2,3 was found 1 times: Support is = 0.008849557522123894

B 2,3 was found 2 times: Support is = 0.017699115044247787

W 2,2 was found 2 times: Support is = 0.017699115044247787

Y 2,15 was found 2 times: Support is = 0.017699115044247787

S 2,16 was found 2 times: Support is = 0.017699115044247787

AOP 2,3 was found 2 times: Support is = 0.017699115044247787

AOP 2,15 was found 2 times: Support is = 0.017699115044247787

ATU 2,15 was found 1 times: Support is = 0.008849557522123894

I 2,9 was found 1 times: Support is = 0.008849557522123894

AE 6,3 was found 1 times: Support is = 0.008849557522123894

ACDEF 6,14 was found 1 times: Support is = 0.008849557522123894

ABC 6,5 was found 1 times: Support is = 0.008849557522123894

AI 6,9 was found 1 times: Support is = 0.008849557522123894

AQR 6,3 was found 1 times: Support is = 0.008849557522123894

W 5,10 was found 1 times: Support is = 0.008849557522123894

AJ 5,3 was found 1 times: Support is = 0.008849557522123894

S 5,6 was found 1 times: Support is = 0.008849557522123894

Y 5,9 was found 1 times: Support is = 0.008849557522123894

Z 5,16 was found 1 times: Support is = 0.008849557522123894

D 2,4 was found 6 times: Support is = 0.05309734513274336

AN 2,11 was found 5 times: Support is = 0.04424778761061947

AO 2,4 was found 6 times: Support is = 0.05309734513274336

C 2,4 was found 6 times: Support is = 0.05309734513274336

E 2,4 was found 6 times: Support is = 0.05309734513274336
```

```
K 2,4 was found 6 times: Support is = 0.05309734513274336

AFGHIJ 3,5 was found 1 times: Support is = 0.008849557522123894

AO 2,4 was found 6 times: Support is = 0.05309734513274336

C 2,4 was found 6 times: Support is = 0.05309734513274336

D 2,4 was found 6 times: Support is = 0.05309734513274336

E 2,4 was found 6 times: Support is = 0.05309734513274336

K 2,4 was found 6 times: Support is = 0.05309734513274336

AN 2,11 was found 5 times: Support is = 0.04424778761061947

B 2,3 was found 2 times: Support is = 0.017699115044247787

W 2,2 was found 2 times: Support is = 0.017699115044247787

Y 2,15 was found 2 times: Support is = 0.017699115044247787

AH 2,16 was found 1 times: Support is = 0.008849557522123894

AJKL 2,16 was found 1 times: Support is = 0.008849557522123894

S 2,16 was found 2 times: Support is = 0.017699115044247787

Z 2,16 was found 1 times: Support is = 0.008849557522123894

AOP 2,3 was found 2 times: Support is = 0.017699115044247787

AOP 2,15 was found 2 times: Support is = 0.017699115044247787

ACDE 2,7 was found 1 times: Support is = 0.008849557522123894

AT 3,5 was found 1 times: Support is = 0.008849557522123894

AEF 3,5 was found 1 times: Support is = 0.008849557522123894

AR 3,5 was found 1 times: Support is = 0.008849557522123894

AP 3,2 was found 1 times: Support is = 0.008849557522123894

AQ 3,2 was found 2 times: Support is = 0.017699115044247787

AS 3,2 was found 1 times: Support is = 0.008849557522123894
```

```
AU 3,2 was found 1 times: Support is = 0.008849557522123894

AV 3,2 was found 1 times: Support is = 0.008849557522123894

AW 3,2 was found 1 times: Support is = 0.008849557522123894

AX 3,2 was found 1 times: Support is = 0.008849557522123894

AY 3,2 was found 1 times: Support is = 0.008849557522123894

AZ 3,2 was found 1 times: Support is = 0.008849557522123894

ARST 3,2 was found 1 times: Support is = 0.008849557522123894

AIJ 3,9 was found 1 times: Support is = 0.008849557522123894

AHI 3,14 was found 1 times: Support is = 0.008849557522123894

D 2,4 was found 6 times: Support is = 0.05309734513274336

AN 2,11 was found 5 times: Support is = 0.04424778761061947

AO 2,4 was found 6 times: Support is = 0.05309734513274336

C 2,4 was found 6 times: Support is = 0.05309734513274336

E 2,4 was found 6 times: Support is = 0.05309734513274336

K 2,4 was found 6 times: Support is = 0.05309734513274336

ADE 3,14 was found 1 times: Support is = 0.008849557522123894

AH 5,10 was found 1 times: Support is = 0.008849557522123894

AVW 5,5 was found 1 times: Support is = 0.008849557522123894

AXY 5,5 was found 1 times: Support is = 0.008849557522123894

AQR 5,3 was found 1 times: Support is = 0.008849557522123894

AQR 5,14 was found 1 times: Support is = 0.008849557522123894

I 5,6 was found 1 times: Support is = 0.008849557522123894

AF 2,5 was found 1 times: Support is = 0.008849557522123894

AQRS 3,15 was found 1 times: Support is = 0.008849557522123894

AO 2,4 was found 6 times: Support is = 0.05309734513274336
```

```
C 2,4 was found 6 times: Support is = 0.05309734513274336

D 2,4 was found 6 times: Support is = 0.05309734513274336

E 2,4 was found 6 times: Support is = 0.05309734513274336

K 2,4 was found 6 times: Support is = 0.05309734513274336

AN 2,11 was found 5 times: Support is = 0.04424778761061947

Z 3,3 was found 1 times: Support is = 0.008849557522123894

S 3,3 was found 1 times: Support is = 0.008849557522123894

D 3,3 was found 1 times: Support is = 0.008849557522123894

AKLM 3,6 was found 1 times: Support is = 0.008849557522123894

ALMN 3,6 was found 2 times: Support is = 0.017699115044247787

AMN 3,6 was found 2 times: Support is = 0.017699115044247787

ANO 3,6 was found 2 times: Support is = 0.017699115044247787

H 3,6 was found 1 times: Support is = 0.008849557522123894

O 3,6 was found 1 times: Support is = 0.008849557522123894

AUV 3,6 was found 1 times: Support is = 0.008849557522123894

N 3,6 was found 1 times: Support is = 0.008849557522123894

I 3,6 was found 1 times: Support is = 0.008849557522123894

AFG 3,6 was found 1 times: Support is = 0.008849557522123894

ASTU 3,6 was found 1 times: Support is = 0.008849557522123894

ANOP 3,6 was found 1 times: Support is = 0.008849557522123894

ABCDE 3,6 was found 1 times: Support is = 0.008849557522123894

AXYZ 3,6 was found 1 times: Support is = 0.008849557522123894

D 2,4 was found 6 times: Support is = 0.05309734513274336

AN 2,11 was found 5 times: Support is = 0.04424778761061947
```

AO 2,4 was found 6 times: Support is = 0.05309734513274336

C 2,4 was found 6 times: Support is = 0.05309734513274336

E 2,4 was found 6 times: Support is = 0.05309734513274336

K 2,4 was found 6 times: Support is = 0.05309734513274336

AOPQ 3,5 was found 1 times: Support is = 0.008849557522123894

APQR 3,2 was found 1 times: Support is = 0.008849557522123894

Q 3,2 was found 1 times: Support is = 0.008849557522123894

ACD 3,2 was found 1 times: Support is = 0.008849557522123894

AGH 3,2 was found 1 times: Support is = 0.008849557522123894

AO 2,4 was found 6 times: Support is = 0.05309734513274336

C 2,4 was found 6 times: Support is = 0.05309734513274336

D 2,4 was found 6 times: Support is = 0.05309734513274336

E 2,4 was found 6 times: Support is = 0.05309734513274336

K 2,4 was found 6 times: Support is = 0.05309734513274336
.........Step 2............

B 2,3:W 2,2 was found 2 times: Support is = 0.017699115044247787

W 2,2:Y 2,15 was found 2 times: Support is = 0.017699115044247787

K 2,4:AN 2,11 was found 2 times: Support is = 0.017699115044247787

D 2,4:AN 2,11 was found 3 times: Support is = 0.02654867256637168

AN 2,11:AO 2,4 was found 3 times: Support is = 0.02654867256637168

C 2,4:E 2,4 was found 3 times: Support is = 0.02654867256637168

AO 2,4:C 2,4 was found 6 times: Support is = 0.05309734513274336

C 2,4:D 2,4 was found 3 times: Support is = 0.02654867256637168

D 2,4:E 2,4 was found 3 times: Support is = 0.02654867256637168

E 2,4:K 2,4 was found 6 times: Support is = 0.05309734513274336
.........Step 3............

E 2,4:K 2,4:AN 2,11 was found 2 times: Support is = 0.017699115044247787

D 2,4:AN 2,11:AO 2,4 was found 3 times: Support is = 0.02654867256637168

AN 2,11:AO 2,4:C 2,4 was found 3 times: Support is = 0.02654867256637168

AO 2,4:C 2,4:E 2,4 was found 3 times: Support is = 0.02654867256637168

C 2,4:E 2,4:K 2,4 was found 3 times: Support is = 0.02654867256637168

AO 2,4:C 2,4:D 2,4 was found 3 times: Support is = 0.02654867256637168

C 2,4:D 2,4:E 2,4 was found 3 times: Support is = 0.02654867256637168

D 2,4:E 2,4:K 2,4 was found 3 times: Support is = 0.02654867256637168

.........Step 4............

D 2,4:E 2,4:K 2,4:AN 2,11 was found 2 times: Support is = 0.017699115044247787

D 2,4:AN 2,11:AO 2,4:C 2,4 was found 3 times: Support is = 0.02654867256637168

AN 2,11:AO 2,4:C 2,4:E 2,4 was found 3 times: Support is = 0.02654867256637168

AO 2,4:C 2,4:E 2,4:K 2,4 was found 3 times: Support is = 0.02654867256637168

AO 2,4:C 2,4:D 2,4:E 2,4 was found 3 times: Support is = 0.02654867256637168

C 2,4:D 2,4:E 2,4:K 2,4 was found 3 times: Support is = 0.02654867256637168

........Step 5............

C 2,4:D 2,4:E 2,4:K 2,4:AN 2,11 was found 2 times: Support is = 0.017699115044247787

D 2,4:AN 2,11:AO 2,4:C 2,4:E 2,4 was found 3 times: Support is = 0.02654867256637168

AN 2,11:AO 2,4:C 2,4:E 2,4:K 2,4 was found 3 times: Support is 0.02654867256637168

AO 2,4:C 2,4:D 2,4:E 2,4:K 2,4 was found 3 times: Support is =0.02654867256637168

.........Step 6............

D 2,4:AN 2,11:AO 2,4:C 2,4:E 2,4:K 2,4 was found 3 times: Support is = 0.026548672566371

Appendix 6: Threat Prediction Model Output for MIT Lincoln Lab LLDOS 1.0

APPENDIX 6: Threat Prediction Model Output for MIT Lincoln Lab LLDOS 1.0

```
.........Step 1............

A 22,31 was found 2 times: Support is = 0.014598540145985401

F 24,31 was found 2 times: Support is = 0.014598540145985401

L 12,69 was found 2 times: Support is = 0.014598540145985401

K 26,66 was found 2 times: Support is = 0.014598540145985401

L 10,68 was found 2 times: Support is = 0.014598540145985401

L 23,37 was found 2 times: Support is = 0.014598540145985401

C 12,41 was found 3 times: Support is = 0.021897810218978103

D 12,41 was found 3 times: Support is = 0.021897810218978103

C 10,70 was found 3 times: Support is = 0.021897810218978103

D 10,70 was found 3 times: Support is = 0.021897810218978103

M 21,65 was found 3 times: Support is = 0.021897810218978103

A 13,14 was found 3 times: Support is = 0.021897810218978103

F 25,31 was found 3 times: Support is = 0.021897810218978103

F 9,14 was found 3 times: Support is = 0.021897810218978103

K 13,35 was found 6 times: Support is = 0.043795620437956206

I 20,62 was found 3 times: Support is = 0.021897810218978103

J 20,62 was found 3 times: Support is = 0.021897810218978103

C 13,60 was found 3 times: Support is = 0.021897810218978103

D 13,60 was found 2 times: Support is = 0.014598540145985401
```

```
C 12,41 was found 3 times: Support is = 0.021897810218978103

D 12,41 was found 3 times: Support is = 0.021897810218978103

C 10,70 was found 3 times: Support is = 0.021897810218978103

D 10,70 was found 3 times: Support is = 0.021897810218978103

M 21,65 was found 3 times: Support is = 0.021897810218978103

A 13,14 was found 3 times: Support is = 0.021897810218978103

F 25,31 was found 3 times: Support is = 0.021897810218978103

F 9,14 was found 3 times: Support is = 0.021897810218978103

K 13,35 was found 6 times: Support is = 0.043795620437956206

I 20,62 was found 3 times: Support is = 0.021897810218978103

J 20,62 was found 3 times: Support is = 0.021897810218978103

C 13,60 was found 3 times: Support is = 0.021897810218978103

D 13,60 was found 2 times: Support is = 0.014598540145985401

F 5,31 was found 7 times: Support is = 0.051094890510948905

F 9,64 was found 5 times: Support is = 0.0364963503649635

K 13,35 was found 6 times: Support is = 0.043795620437956206

G 10,13 was found 2 times: Support is = 0.014598540145985401

G 9,7 was found 1 times: Support is = 0.0072992700729927005

G 12,8 was found 2 times: Support is = 0.014598540145985401

G 9,9 was found 3 times: Support is = 0.021897810218978103

F 5,31 was found 7 times: Support is = 0.051094890510948905

F 9,64 was found 5 times: Support is = 0.0364963503649635

K 13,35 was found 6 times: Support is = 0.043795620437956206

G 10,13 was found 2 times: Support is = 0.014598540145985401

F 5,31 was found 7 times: Support is = 0.051094890510948905
```

```
F 9,60 was found 1 times: Support is = 0.0072992700729927005

G 12,8 was found 2 times: Support is = 0.014598540145985401

G 9,9 was found 3 times: Support is = 0.021897810218978103

A 13,38 was found 1 times: Support is = 0.0072992700729927005

L 3,37 was found 1 times: Support is = 0.0072992700729927005

G 9,58 was found 2 times: Support is = 0.014598540145985401

F 5,31 was found 7 times: Support is = 0.051094890510948905

G 10,71 was found 1 times: Support is = 0.0072992700729927005

G 9,58 was found 2 times: Support is = 0.014598540145985401

F 5,32 was found 2 times: Support is = 0.014598540145985401

F 5,33 was found 2 times: Support is = 0.014598540145985401

G 9,9 was found 3 times: Support is = 0.021897810218978103

K 13,35 was found 6 times: Support is = 0.043795620437956206

F 5,34 was found 2 times: Support is = 0.014598540145985401

F 9,64 was found 5 times: Support is = 0.0364963503649635

I 7,63 was found 3 times: Support is = 0.021897810218978103

I 20,63 was found 1 times: Support is = 0.0072992700729927005

I 7,63 was found 3 times: Support is = 0.021897810218978103

I 7,61 was found 2 times: Support is = 0.014598540145985401

F 5,31 was found 7 times: Support is = 0.051094890510948905

I 7,41 was found 2 times: Support is = 0.014598540145985401

I 7,42 was found 2 times: Support is = 0.014598540145985401

I 7,43 was found 2 times: Support is = 0.014598540145985401

I 7,44 was found 2 times: Support is = 0.014598540145985401
```

```
I 7,45 was found 2 times: Support is = 0.014598540145985401
I 7,46 was found 2 times: Support is = 0.014598540145985401
F 1,64 was found 1 times: Support is = 0.0072992700729927005
I 7,47 was found 2 times: Support is = 0.014598540145985401
I 7,48 was found 2 times: Support is = 0.014598540145985401
I 7,49 was found 2 times: Support is = 0.014598540145985401
I 7,50 was found 1 times: Support is = 0.0072992700729927005
I 7,51 was found 1 times: Support is = 0.0072992700729927005
I 7,52 was found 1 times: Support is = 0.0072992700729927005
I 7,53 was found 1 times: Support is = 0.0072992700729927005
I 7,54 was found 1 times: Support is = 0.0072992700729927005
I 7,55 was found 1 times: Support is = 0.0072992700729927005
I 7,56 was found 1 times: Support is = 0.0072992700729927005
I 7,57 was found 1 times: Support is = 0.0072992700729927005
I 7,30 was found 4 times: Support is = 0.029197080291970802
I 7,40 was found 5 times: Support is = 0.0364963503649635
I 7,3 was found 1 times: Support is = 0.0072992700729927005
I 7,4 was found 1 times: Support is = 0.0072992700729927005
I 7,6 was found 1 times: Support is = 0.0072992700729927005
F 5,3 was found 1 times: Support is = 0.0072992700729927005
F 9,64 was found 5 times: Support is = 0.0364963503649635
G 9,60 was found 1 times: Support is = 0.0072992700729927005
H 17,36 was found 1 times: Support is = 0.0072992700729927005
I 7,62 was found 5 times: Support is = 0.0364963503649635
J 20,14 was found 4 times: Support is = 0.029197080291970802
```

```
I 7,62 was found 5 times: Support is = 0.0364963503649635

J 20,14 was found 4 times: Support is = 0.029197080291970802

I 7,62 was found 5 times: Support is = 0.0364963503649635

J 20,14 was found 4 times: Support is = 0.029197080291970802

I 7,30 was found 4 times: Support is = 0.029197080291970802

I 7,62 was found 5 times: Support is = 0.0364963503649635

J 20,14 was found 4 times: Support is = 0.029197080291970802

J 20,30 was found 2 times: Support is = 0.014598540145985401

I 7,30 was found 4 times: Support is = 0.029197080291970802

I 7,40 was found 5 times: Support is = 0.0364963503649635

J 20,30 was found 2 times: Support is = 0.014598540145985401

I 7,40 was found 5 times: Support is = 0.0364963503649635

J 20,40 was found 2 times: Support is = 0.014598540145985401

I 7,40 was found 5 times: Support is = 0.0364963503649635

J 20,40 was found 2 times: Support is = 0.014598540145985401

F 5,31 was found 7 times: Support is = 0.051094890510948905

F 9,64 was found 5 times: Support is = 0.0364963503649635

A 22,31 was found 2 times: Support is = 0.014598540145985401

F 24,31 was found 2 times: Support is = 0.014598540145985401

L 12,69 was found 2 times: Support is = 0.014598540145985401

K 26,66 was found 2 times: Support is = 0.014598540145985401

L 10,68 was found 2 times: Support is = 0.014598540145985401

L 23,37 was found 2 times: Support is = 0.014598540145985401

C 12,41 was found 3 times: Support is = 0.021897810218978103
```

D 12,41 was found 3 times: Support is = 0.021897810218978103

C 10,70 was found 3 times: Support is = 0.021897810218978103

D 10,70 was found 3 times: Support is = 0.021897810218978103

M 21,65 was found 3 times: Support is = 0.021897810218978103

A 13,14 was found 3 times: Support is = 0.021897810218978103

F 25,31 was found 3 times: Support is = 0.021897810218978103

F 9,14 was found 3 times: Support is = 0.021897810218978103

K 13,35 was found 6 times: Support is = 0.043795620437956206

I 20,62 was found 3 times: Support is = 0.021897810218978103

J 20,62 was found 3 times: Support is = 0.021897810218978103

C 13,60 was found 3 times: Support is = 0.021897810218978103

.........Step 2............

C 12,41:D 12,41 was found 3 times: Support is = 0.021897810218978103

D 12,41:C 10,70 was found 3 times: Support is = 0.021897810218978103

C 10,70:D 10,70 was found 3 times: Support is = 0.021897810218978103

D 10,70:M 21,65 was found 3 times: Support is = 0.021897810218978103

M 21,65:A 13,14 was found 3 times: Support is = 0.021897810218978103

A 13,14:F 25,31 was found 3 times: Support is = 0.021897810218978103

F 25,31:F 9,14 was found 3 times: Support is = 0.021897810218978103

F 9,14:K 13,35 was found 3 times: Support is = 0.021897810218978103

K 13,35:I 20,62 was found 3 times: Support is = 0.021897810218978103

I 20,62:J 20,62 was found 3 times: Support is = 0.021897810218978103

J 20,62:C 13,60 was found 3 times: Support is = 0.021897810218978103

.........Step 3............

C 12,41:D 12,41:C 10,70 was found 3 times: Support is =

0.021897810218978103

D 12,41:C 10,70:D 10,70 was found 3 times: Support is = 0.021897810218978103

C 10,70:D 10,70:M 21,65 was found 3 times: Support is = 0.021897810218978103

D 10,70:M 21,65:A 13,14 was found 3 times: Support is = 0.021897810218978103

M 21,65:A 13,14:F 25,31 was found 3 times: Support is = 0.021897810218978103

A 13,14:F 25,31:F 9,14 was found 3 times: Support is = 0.021897810218978103

F 25,31:F 9,14:K 13,35 was found 3 times: Support is = 0.021897810218978103

F 9,14:K 13,35:I 20,62 was found 3 times: Support is = 0.021897810218978103

K 13,35:I 20,62:J 20,62 was found 3 times: Support is = 0.021897810218978103

I 20,62:J 20,62:C 13,60 was found 3 times: Support is = 0.021897810218978103

.........Step 4............

C 12,41:D 12,41:C 10,70:D 10,70 was found 3 times: Support is = 0.021897810218978103

D 12,41:C 10,70:D 10,70:M 21,65 was found 3 times: Support is = 0.021897810218978103

C 10,70:D 10,70:M 21,65:A 13,14 was found 3 times: Support is = 0.021897810218978103

D 10,70:M 21,65:A 13,14:F 25,31 was found 3 times: Support is = 0.021897810218978103

M 21,65:A 13,14:F 25,31:F 9,14 was found 3 times: Support is = 0.021897810218978103

A 13,14:F 25,31:F 9,14:K 13,35 was found 3 times: Support is = 0.021897810218978103

F 25,31:F 9,14:K 13,35:I 20,62 was found 3 times: Support is = 0.021897810218978103

F 9,14:K 13,35:I 20,62:J 20,62 was found 3 times: Support is = 0.021897810218978103

K 13,35:I 20,62:J 20,62:C 13,60 was found 3 times: Support is = 0.021897810218978103

.........Step 5............

C 12,41:D 12,41:C 10,70:D 10,70:M 21,65 was found 3 times: Support is = 0.021897810218978103

D 12,41:C 10,70:D 10,70:M 21,65:A 13,14 was found 3 times: Support is = 0.021897810218978103

C 10,70:D 10,70:M 21,65:A 13,14:F 25,31 was found 3 times: Support is = 0.021897810218978103

D 10,70:M 21,65:A 13,14:F 25,31:F 9,14 was found 3 times: Support is = 0.021897810218978103

M 21,65:A 13,14:F 25,31:F 9,14:K 13,35 was found 3 times: Support is = 0.021897810218978103

A 13,14:F 25,31:F 9,14:K 13,35:I 20,62 was found 3 times: Support is = 0.021897810218978103

F 25,31:F 9,14:K 13,35:I 20,62:J 20,62 was found 3 times: Support is = 0.021897810218978103

F 9,14:K 13,35:I 20,62:J 20,62:C 13,60 was found 3 times: Support is = 0.021897810218978103

.........Step 6............

C 12,41:D 12,41:C 10,70:D 10,70:M 21,65:A 13,14 was found 3 times: Support is = 0.021897810218978103

D 12,41:C 10,70:D 10,70:M 21,65:A 13,14:F 25,31 was found 3 times: Support is = 0.021897810218978103

C 10,70:D 10,70:M 21,65:A 13,14:F 25,31:F 9,14 was found 3 times: Support is = 0.021897810218978103

D 10,70:M 21,65:A 13,14:F 25,31:F 9,14:K 13,35 was found 3 times: Support is = 0.021897810218978103

M 21,65:A 13,14:F 25,31:F 9,14:K 13,35:I 20,62 was found 3 times: Support is = 0.021897810218978103

A 13,14:F 25,31:F 9,14:K 13,35:I 20,62:J 20,62 was found 3 times: Support is = 0.021897810218978103

F 25,31:F 9,14:K 13,35:I 20,62:J 20,62:C 13,60 was found 3 times: Support is = 0.021897810218978103

.........Step 7............

C 12,41:D 12,41:C 10,70:D 10,70:M 21,65:A 13,14:F 25,31 was found 3 times: Support is = 0.021897810218978103

D 12,41:C 10,70:D 10,70:M 21,65:A 13,14:F 25,31:F 9,14 was found 3 times: Support is = 0.021897810218978103

C 10,70:D 10,70:M 21,65:A 13,14:F 25,31:F 9,14:K 13,35 was found 3 times: Support is = 0.021897810218978103

D 10,70:M 21,65:A 13,14:F 25,31:F 9,14:K 13,35:I 20,62 was found 3 times: Support is = 0.021897810218978103

M 21,65:A 13,14:F 25,31:F 9,14:K 13,35:I 20,62:J 20,62 was found 3 times: Support is = 0.021897810218978103

A 13,14:F 25,31:F 9,14:K 13,35:I 20,62:J 20,62:C 13,60 was found 3 times: Support is = 0.021897810218978103

.........Step 8............

C 12,41:D 12,41:C 10,70:D 10,70:M 21,65:A 13,14:F 25,31:F 9,14 was found 3 times: Support is = 0.021897810218978103

D 12,41:C 10,70:D 10,70:M 21,65:A 13,14:F 25,31:F 9,14:K 13,35 was found 3 times: Support is = 0.021897810218978103

C 10,70:D 10,70:M 21,65:A 13,14:F 25,31:F 9,14:K 13,35:I 20,62 was found 3 times: Support is = 0.021897810218978103

D 10,70:M 21,65:A 13,14:F 25,31:F 9,14:K 13,35:I 20,62:J 20,62 was found 3 times: Support is = 0.021897810218978103

M 21,65:A 13,14:F 25,31:F 9,14:K 13,35:I 20,62:J 20,62:C 13,60 was found 3 times: Support is = 0.021897810218978103

.........Step 9............

C 12,41:D 12,41:C 10,70:D 10,70:M 21,65:A 13,14:F 25,31:F 9,14:K 13,35 was found 3 times: Support is = 0.021897810218978103

D 12,41:C 10,70:D 10,70:M 21,65:A 13,14:F 25,31:F 9,14:K 13,35:I 20,62 was found 3 times: Support is = 0.021897810218978103

C 10,70:D 10,70:M 21,65:A 13,14:F 25,31:F 9,14:K 13,35:I 20,62:J 20,62 was found 3 times: Support is = 0.021897810218978103

D 10,70:M 21,65:A 13,14:F 25,31:F 9,14:K 13,35:I 20,62:J 20,62:C 13,60 was found 3 times: Support is = 0.021897810218978103

.........Step 10............

C 12,41:D 12,41:C 10,70:D 10,70:M 21,65:A 13,14:F 25,31:F 9,14:K 13,35:I 20,62 was found 3 times: Support is = 0.021897810218978103

D 12,41:C 10,70:D 10,70:M 21,65:A 13,14:F 25,31:F 9,14:K 13,35:I 20,62:J 20,62 was found 3 times: Support is = 0.021897810218978103

C 10,70:D 10,70:M 21,65:A 13,14:F 25,31:F 9,14:K 13,35:I 20,62:J 20,62:C 13,60 was found 3 times: Support is = 0.021897810218978103

.........Step 11............

C 12,41:D 12,41:C 10,70:D 10,70:M 21,65:A 13,14:F 25,31:F 9,14:K 13,35:I 20,62:J 20,62 was found 3 times: Support is = 0.021897810218978103, Confidence is =1

D 12,41:C 10,70:D 10,70:M 21,65:A 13,14:F 25,31:F 9,14:K 13,35:I 20,62:J 20,62:C 13,60 was found 3 times: Support is = 0.021897810218978103

.........Step 12............

C 12,41:D 12,41:C 10,70:D 10,70:M 21,65:A 13,14:F 25,31:F 9,14:K 13,35:I 20,62:J 20,62:C 13,60 was found 3 times: Support is = 0.021897810218978103

Appendix 7: Snort and Suricata Event Reports for Plymouth University APT

Table E: Snort Report for Plymouth University APT (After Mitigation)

DATE/TIME	SRCIP	SPORT	DST IP	D PORT	EVENT MESSAGE
19:51:09	**10.1.0.134**		**10.1.0.131**		**GPL ICMP INFO PING BSDTYPE**
19:51:09	**10.1.0.134**		**10.1.0.131**		**GPL ICMP INFO PING NIS**
19:51:10	10.1.0.3	8080	10.1.0.135	49162	ET CURRENT_EVENTS landing page with malicious Java applet
19:51:10	10.1.0.3	8080	10.1.0.135	49163	ET CURRENT_EVENTS Possible Metasploit Java Payload
19:51:10	10.1.0.3	8080	10.1.0.135	49163	EY CURRENT_EVENTS Possible Metasploit Java Exploit
19:51:10	10.1.0.3	8080	10.1.0.135	49163	ET INFO JAVA - Java Archive Download By Vulnerable Client
19:51:10	10.1.0.3	1024	10.1.0.135	49164	ET TROJAN Metasploit Meterpreter stdapi_*Command Request
19:51:10	**10.1.0.3**	**1024**	**10.1.0.197**	**49174**	**ET TROJAN Metasploit Meterpreter core_channel_* Command Request**
19:51:13	**10.1.0.134**	**17500**	**10.1.0.225**	**17500**	**ET POLICY Dropbox Client Broadcasting**
19:51:15	**10.1.0.3**	**8080**	**10.1.0.194**	**27497**	**ET INFO JAVA - Java Archive Download**

Note: Bolded records of event are False Positive events

Table F: Suricata Report for Plymouth University APT (After Mitigation)

DATE/TIME	SRC IP	SPORT	DST IP	DPORT	EVENT MESSAGE
19:47:42	**10.1.0.134**		**10.1.0.131**		**GPL ICMP_INFO PING BSDtype**
19:47:42	**10.1.0.134**		**10.1.0.131**		**GPL ICMP_INFO PING *NIX**
19:47:42	10.1.0.3	8080	10.1.0.135	49162	ET CURRENT_EVENTS landing page with malicious Java applet
19:47:42	10.1.0.3	8080	10.1.0.135	49163	ET INFO JAVA - Java Archive Download By Vulnerable Client
19:47:42	10.1.0.3	8080	10.1.0.135	49163	ET CURRENT_EVENTS Possible Metasploit Java Exploit
19:47:42	10.1.0.3	8080	10.1.0.135	49163	ET CURRENT_EVENTS Possible Metasploit Java Payload
19:47:42	10.1.0.3	1024	10.1.0.135	49164	ET TROJAN Metasploit Meterpreter stdapi_*Command Request
19:47:43	**10.1.0.134**	**49177**	**10.1.0.133**	**445**	**GPL NETBIOS SMB-DS IPC$ unicode share access**
19:47:46	**10.1.0.134**	**17500**	**10.1.0.225**	**17500**	**ET POLICY Dropbox Client Broadcasting**
19:47:46	**10.1.0.35**		**10.1.0.133**		**GPL ICMP_INFO PING *NIX**

Note: Bolded records of event are False Positive events

Appendix 8: Snort and Suricata Event Reports for MIT Lincoln Lab LLDOS 1.0 (After Mitigation)

Table G: Snort Report for MIT Lincoln Lab LLDOS 1.0

Event_ID	Event_Date	src_Port	src_IPAddress	Dest_Port	Dest_IPAddress
14:21:05	60249	202.77.162.213	111	172.16.115.20	RPC_portmap_sadmind
14:21:05	60253	202.77.162.213	111	172.16.115.20	RPC_portmap_sadmind
14:21:05	60251	202.77.162.213	32773	172.16.115.20	RPC_sadmind_query
14:21:06	60267	202.77.162.213	111	172.16.115.20	RPC_portmap_sadmind
14:21:06	60255	202.77.162.213	32773	172.16.115.20	RPC_sadmind_query
14:21:06	60269	202.77.162.213	32773	172.16.115.20	RPC_sadmind_query
14:21:07	60274	202.77.162.213	111	172.16.115.20	RPC_portmap_sadmind
14:21:07	60287	202.77.162.213	111	172.16.115.20	RPC_portmap_sadmind
14:21:07	60276	202.77.162.213	32773	172.16.115.20	RPC_sadmind_query
14:21:07	60289	202.77.162.213	32773	172.16.115.20	RPC_sadmind_query
14:21:08	**60517**	**202.77.162.213**	**111**	**172.16.112.10**	**RPC_portmap_sadmind**
14:21:08	**60522**		**111**	**172.16.112.10**	**RPC_portm**

		202.77.162.213			ap_sadmind
14:21:08	60298	202.77.162.213	111	172.16.115.20	RPC_portmap_sadmind
14:21:08	60300	202.77.162.213	32773	172.16.115.20	RPC_sadmind_query
14:21:08	60519	202.77.162.213	32774	172.16.112.10	RPC_sadmind_query
14:21:08	60524	202.77.162.213	32774	172.16.112.10	RPC_sadmind_query
14:21:09	60540	202.77.162.213	111	172.16.112.10	RPC_portmap_sadmind
14:21:09	60547	202.77.162.213	111	172.16.112.10	RPC_portmap_sadmind
14:21:09	60567	202.77.162.213	111	172.16.112.50	RPC_portmap_sadmind
14:21:09	60542	202.77.162.213	32774	172.16.112.10	RPC_sadmind_query
14:21:09	60549	202.77.162.213	32774	172.16.112.10	RPC_sadmind_query
14:21:10	60576	202.77.162.213	111	172.16.112.50	RPC_portmap_sadmind
14:21:10	60603	202.77.162.213	111	172.16.112.50	RPC_portmap_sadmind
14:21:10	60569	202.77.162.213	32773	172.16.112.50	RPC_sadmind_query
14:21:10	60578	202.77.162.213	32773	172.16.112.50	RPC_sadmind_query
14:21:10	60605	202.77.162.213	32773	172.16.112.50	RPC_sadmind_query
14:21:11	60617	202.77.162.213	111	172.16.112.50	RPC_portmap_sadmind
14:21:11	60619	202.77.162.213	32773	172.16.112.50	RPC_sadmind_query

Note: Bolded records of event are False Positive events

Table H: Suricata Report for MIT Lincoln Lab LLDOS 1.0

DATE/TIME	SRC IP	Sport	DST IP	DPORT	EVENT MESSAGE
13:31:00	**192.168.1.20**	**6667**	**172.16.113.84**	**8253**	**ET CHAT IRC PING command**
13:31:26	**205.181.112.65**	**80**	**172.16.115.87**	**5552**	**SURICATA HTTP response field missing colon**
13:32:49	**216.32.120.136**	**80**	**172.16.115.5**	**8818**	**SURICATA HTTP response field missing colon**
13:34:05	**207.46.185.11**	**80**	**172.16.115.5**	**30534**	**SURICATA HTTP response field missing colon**
13:41:37	172.16.115.20	53	172.16.112.20	1434	ET EXPLOIT MS-SQL DOS ATTEMPT (08)
13:41:58	172.16.112.100	2134	172.16.112.20	139	GPL NETBIOS NT NULL session
13:42:07	172.16.115.20	33732	172.16.112.105	161	GPL SNMP public access udp
13:42:19	202.77.162.213	54790	172.16.115.20	111	GPL RPC PORTMAP SADMIND REQUEST UDP
13:42:52	202.77.162.213	60251	172.16.115.20	32773	GPL RPC Sadmind query

	with root credentials attempt UDP

Note: Bolded records of event are False Positive events