Evaluating the Availability of Forensic Evidence from Three IDSs: Tool Ability

EMAD ABDULLAH ALSAIARI

A thesis submitted to the Faculty of Design and Creative Technologies
Auckland University of Technology
in partial fulfilment of the
requirements for the degree of
Masters of Forensic Information Technology

School of Engineering, Computer and Mathematical Sciences

Auckland, New Zealand
2016

Declaration

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

Emad Abdullah Alsaiari

Acknowledgement

At the beginning and foremost, the researcher would like to thank almighty Allah. Additionally, I would like to thank everyone who helped me to conduct this thesis starting from my family, supervisor, all relatives and friends.

I would also like to express my thorough appreciation to all the members of Saudi Culture Mission for facilitating the process of studying in a foreign country. I would also like to express my thorough appreciation to all the staff of Saudi Culture Mission for facilitating the process of studying in Auckland University of Technology. Especially, the pervious head principal of the Saudi Culture Mission Dr. Satam Al-Otaibi for all his motivation, advice and support to students from Saudi in New Zealand as well as Saudi Arabia Cultural Attaché Dr. Saud Theyab the head principal of the Saudi Culture Mission.

I would like to thank both of Dr. Alastair Nisbet, thesis supervisor, and Dr. Bobby Mee Loong Yang, my second supervisor for their guidance and advice since the beginning of this thesis. They have been supportive and very helpful during the all stages of the research and without their direction; this thesis would not be accomplished.

Special thanks, I will never ever forget to thank my great father Abdullah, who passed away in 2007. I am sure he will be proud of me to achieve a master's degree in an English speaking country. I am highly indebted to my family to support me as well as their patience through the thesis' phases. This thesis will be never completed without their sacrifices, stimulation and encouragement. In fact, no words can express my deepest appreciation to my beloved wife Nada. She was standing beside all time throughout my study.

Finally, I would like to say thanks to all friends, I would never have completed this thesis without their help and support. I would also like to thank my closest friend Saud Alshaifi for being more than a brother in New Zealand who helped me at all times. I will also never forget my great friends Baraa Bakhsh, Ayman Farid, Ahmed Al Shadidi and all other friends for their help when I asked.

Abstract

There is a risk whenever we use networks, computer systems and Internet technologies that things will go wrong and we need protection in our daily lives. Thus, in many communication networks for a small business or even for home use, people implement Intrusion Detection Systems (IDS). This is to increase the security level of their assets and to detect many malicious activities. An IDS offers significant alerting and logging capabilities that may be useful for forensic purposes. Historically the IDS has been used to detect intrusions and alerts. However, some skilled attacker might be able to erase all logs from the compromised host, which makes it more difficult for the forensic investigator to look for other evidence. The log files generated by the IDSs are essential for identifying the source and the type of the attack, and even the identity of the attacker. However, some LAN network attackers have become very skilled in terms of bypassing some IDSs, which has reduced the capability and efficiency of many signature based security infrastructures. Thus, the aim of this research is to examine three IDSs, and evaluate their capabilities in detecting four different types of network attacks. Additionally, to investigate the IDSs' efficiency in producing admissible forensic evidence. The limitations and shortcomings of each IDS in terms of finding results from each type of attack will also be explored. The challenges and implications encountered while using the three IDSs will be examined, in order to deliver recommendations and suggestions that can assist in developing better system protection.

The objective of the research addresses the implementation of three IDSs (open source) and their abilities for acquiring and preserving digital evidence of LAN networks. This objective will also include a report of the best practice for handling and reporting trials of evidentiary material in the form of digital evidence for four common types of LAN network attacks. The proposed system architecture consists of several devices. These devices are a Firewall, IDSs, namely PADS, OSSEC and Prelude, a Forensic Server and finally end hosts. The selected IDSs will be forensically monitoring the packets traveling from and to the proposed system.

The first stage of this research was to identify and install the proposed system components including their requirements, in order to establish a LAN network experimental environment. All IDSs were running simultaneously on a sole computer to ensure each received the same number of packets and attack types. The reason for this was to ensure the fairness of the evaluation of IDS capabilities to detect and produce digital forensic evidence. Four attack stages were conducted during the research: Reconnaissance, DDoS, Dictionary, and Packet Sniffing attack.

The results illustrate that the selected IDSs can be used as a source of digital evidence as well as the ability to detect, strengths, and weaknesses of each IDSs. These results could assist the LAN networks forensic investigators, law enforcement and other agencies when they are conducting an investigation on similar cases. Some of the IDS fail to detect some well-known LAN network attacks. This failure is related to detection signature databases and the interception functionality. This research will show how each of the selected IDS can be improved, in order to extract admissible digital forensic evidence. Additionally, the opportunities for improvement, development and further research in the LAN network forensic investigation area are also provided.

Table of Contents

CHAPTER 1

INTRODUCTION

CHAPTER 2

LITERATURE REVIEW

CHAPTER 3

RESEARCH METHODOLOGY

CHAPTER 4
RESEARCH FINDINGS

CHAPTER 5
DISCUSSION

CHAPTER 6

CONCLUSION

List of Tables

List of Figures

Glossary of Terms

Elsa	Free software for Network Security Monitoring.
GB	Gigabyte is a multiple of the unit byte for digital information (1024 MB).
HTTP	Hypertext Transfer Protocol.
IP	Internet Protocol.
LAN	Local Area Network.
MySQL dump	Client utility performs logical backups.
NFATs	Network Forensics Analysis Tools.
Ntopng	is a passive network monitoring tool.
Pcap	Packet capture.
RAM	Random Access Memory.
Sguil	Collection of Free software components for Network Security Monitoring.
SNMP	Simple network monitoring protocol.
Snort	An open source network intrusion prevention and detection system.
Squert	Free software component for Network Security Monitoring.
Suricata	is a high performance network intrusion detection system, intrusion prevention system and Network Security Monitoring engine.
SYN	Synchronize packet in transmission control protocol.
TB	Terabyte is a multiple of the unit byte for digital information (1024 GB).
TCP	Transmission Control Protocol.
UDP	User Datagram Protocol.
SQL	Structured Query Language.

Chapter 1

Introduction

1.0 BACKGROUND

Network and computer systems have become very useful to business and are being used in governments and many other organizations to do their daily work. However, these systems are suffering from many issues including network attacks. Srivastava, Gupta, Tyagi, Sharma, and Mishra (2011) highlight that the number of DDoS attacks that target the devices in LAN networks is reaching up to 50000 attacks per week. As a result, the system and security administrators need to apply security measures in order to protect information and enhance their level of security.

The essential processes and principles of network security and forensics are crucial for monitoring, preventing, and recording the activities on various types of networks, regardless of whether these activities are malicious or not. Many tools of the network security can perform network monitoring. For example, a firewall, an intrusion detection system and an intrusion prevention system. One of the main reasons to utilise an intrusion detection system is to detect attacks that can be transmitted through computer networks.

Network security administrators can implement and install many network security devices and use different techniques in order to protect their assets from being targeted by attackers. Network devices can also be deployed in different places in LAN networks to prevent the computers from being compromised. Thus, these devices can monitor and record LAN network traffic, and malicious activities. Therefore, extracting digital evidence from network security systems is possible. Pilli, Joshi, and Niyogi (2010) discussed an advanced model of the network forensics that can acquire and preserve evidence from LAN network devices. This model was adopted for research in order to conduct the experimental phase of the research in a forensically sound manner.

The digital forensic investigation is involving the processes of identification, collection, acquisition, and presentation of digital evidence (ISO, 2012). Sibiya, Venter, Ngobeni, and Fogwill (2012) stated that:

> "The ability of digital forensic investigators to perform their function is heavily reliant on their ability to acquire digital evidence from computer systems and network devices" (p.1).

Often the concepts and principles of forensic analysis and intrusion system detection (IDS) are not considered with each other. While the IDS seems that one of the candidates for collecting information from the traffic of LAN networks. This information may lead to trace and analyse a security incident that occurs in a network-based computer system in order to know who and how the intruder committed his attack. This means the log metadata that is generated by firewalls and IDSs, and stored in databases is the main file that is required by digital forensic investigators to be collected and examined.

The following section will discuss and define some of existing issues in LAN network investigation. Then Section 1.2 will describe the motivation of the research. The research structure will be presented in Section 1.3.

1.1 PROBLEM AREAS

There are tools and techniques that can monitor all packets transferring through the LAN networks. The intrusion detection system is one of these tools that is considered crucial. Some IDSs have the ability not only to detect, but also to generate alerts when malicious activity is detected, and can store all alerts in databases. These alerts could lead to finding potential evidence that may lead to the source of an attack. However, some of IDSs might not be able to detect different types of network attacks, which could cause a failure of the main objective of using IDS for detection purposes. There are many IDSs available as open source opportunities, and some of them have not been examined, and compared with other IDSs. Therefore, there is a need to investigate and examine the ability of different open source IDSs to acquire forensic evidence when different types of network attacks are launched against the network devices. The investigation of IDSs will determine their positive and negative results that can be discovered after the occurrence of the attacks, which will determine the effectiveness of each of the selected IDS in detection.

1.2 MOTIVATION FOR RESEARCH

The prime motivation to conduct this research is that there are many IDSs that are widely used around the world. IDSs have different capabilities as well as different features. These abilities include issuing alerts for network attacks, and some IDSs may not be successful at detecting certain types of attacks, as each IDS has its own signature-based dataset for detecting malicious events. This motivation has led to selecting three different open sources IDSs, and running four different attacks against the system devices in order to evaluate their capabilities in recovering forensic evidence. Therefore, this security software required further study in order to produce evaluate their ability to recover potential evidence that could be used in a courtroom. Three IDSs (PADS, OSSEC, and Prelude) use different techniques and have different abilities to detect malicious events in LAN networks. These devices can collect potential evidence from packets that are traveling in and out of the LAN networks and can be used as a source of digital evidence.

The second motivation for this research is the rising trend of malicious incidents in LAN networks that have happened recently in many places including Saudi Arabia. Some of these malicious incidents have targeted the computers and devices of LAN networks of many ministries and companies of government. For example, one of the world’s largest companies (Saudi Aramco: Oil Industry) has been attacked by a virus via its network devices. These attacking incidents require more knowledge, people, and tools to forensically investigate in the LAN network.

The network attacks have developed as substantial threats for the infrastructure of the LAN network devices and their services. There are many processes and methods that have been made to avert such intrusion events, but these attacks still exist and are increasing steadily ("NCSC Incident Response | NCSC", 2016). The main motivation for conducting this research is to identify and evaluate the ability of three IDSs in terms of seizing forensic evidence from LAN networks. It is also to discover the scope of the availability of digital evidence from the selected IDSs and present the detail of each attack that occurs in the LAN networks. Further research is needed to recognize the ability of the selected IDSs in terms of finding evidence based on their alert systems. This leads to the proposed research question which is:

Do the three selected IDSs working together detect more attacks than any one single IDS?

1.3 STRUCTURE OF THESIS

Chapter 2 will review several related studies and the current state of the area that discusses the LAN network security as well as the objectives for various types of networks. These studies will cover the principles of digital forensics investigation in LAN networks. This review will also identify common attacks and potential threats as well as to identify issues in detecting these attacks. By launching different types of attacks, the impact will be evaluated and preparation for collecting digital forensic evidence from the LAN networks will be discussed. The network investigation life cycle will be taken into consideration starting from the acquisition, preservation, analysis, and reporting of collected digital forensic evidence for examination in the network analysis.

The structure of Chapter 2 consists of twelve sections. An overview of the network security and its objectives will be introduced in section one and two respectively. The third section will discuss the vulnerabilities and threats of the network while the description of the network attacks will provide detailed information about their negative effects in the network. The common attacks in the network environment will be highlighted in detail relating to four selected types of attack in section five. The sixth section will discuss digital evidence as well as digital forensics as a science. The area of network forensics will be discussed and the research approach will be discussed in section seven. The eighth section will describe the steps of the life cycle of network investigation for all types of network. Section nine will discuss the well-known digital forensic tools and their application for usage on computers, memory, networks, mobile platforms, and databases. The tenth section will discuss the intrusion detection system principles and types. The current challenges will discuss in section eleven. Finally, Chapter 2 concludes with summarizing the main issues and problems that can be faced when running a network forensic investigation.

Based on the problems, and issues described in Chapter 2, Chapter 3 will discuss the proposed research methodology that is used in this research. The research question and sub-questions will be developed. This chapter consists of seven sections. Section one will discuss the main research question and sub-questions. The second section will highlight the research design. The proposed system architecture of the research will be outlined in section three. The fourth section will describe the design system devices and additional software needed for the experiment. Section five will discuss the data requirements, which includes preparation of the network design, and how to implement the network architecture in a way to be as similar as possible to a real life scenario. The data requirements also will include data generation, collection, analysis, presentation and data reporting. The sixth section discusses the limitations of the research. Lastly, the conclusion of the chapter will be given in section seven.

In Chapter 4, the installation of the experiment components will be discussed in detail with requested software in section one and two. In Section three, the generated data is used and the results will be collected. Additionally, this chapter will present the results of the experiments in a visual manner and group them based on the IDS. Based on each independent experiment, the comparative analysis in section four will present the results in a table form that will assist the researcher to discuss these findings. However, some alterations were made in the experiment practice phase in order to have fairness in the results. These changes are discussed in Chapter 3.

Chapter 5 will discuss the findings presented in the previous chapter. The findings will help the researcher to answer the main question as well as sub-questions of the research in section one. The answer of the sub-questions will be presented in table forms for each question including the summary of the answer. These answers will help the researcher to discuss the findings of the research in section two and to evaluate the abilities of each IDS used in the research. The answer and discussions together will produce the recommendations in section three for LAN network investigators.

Chapter 6 will summarize the results of the research and describe the limitations of the research in order to improve additional studies in the future. The recommendations for future research will be discussed as well.

Chapter 2
Literature Review

2.0 INTRODUCTION

The aim of this Chapter is to review current studies and work that discusses the principles of computer and network security as they relate to forensic models and tools. These tools are used to monitor the network activities as well as extraction tools for digital evidence from the network. This review is to identify potential attacks and threats to wired networks and to identify issues for more research and investigations. By identifying various attacks that exist in the network environment, this can lead to evaluation of the readiness of LAN networks and impacts upon these networks. The studies within this research will include the forensic investigation of LANs and discussions of the digital forensic life cycle. The digital forensics life cycle beings with acquisition and preservation, and ends with reporting.

Chapter 2 has been divided into ten sections. The first and second section will discuss an overview and objectives of the network security. The third section has a comprehensive review of the serious vulnerabilities and threats that target the LAN networks. Then common network attacks will be discussed in sections four and five, respectively. The next section will show the evidence in the digital forensics life cycle. The seventh section will illustrate the forensic evidence in the network whereas, the network investigation life cycle will be explained in the eighth section. Usage and the structure of an Intrusion Detection System and network forensic tools will be detailed in depth in section nine. Finally, the current challenges for future research will conclude the chapter.

2.1 NETWORK SECURITY OVERVIEW

One of the most powerful sources of information is the data stored within computer systems, as it may contain valuable information, such as user credentials, organisation financial transactions, payroll, and other important information. Thus, it is crucial to apply security on organization LANs, WANs, and WIFI devices including host devices in order to protect information from malicious activities. According to Convery (2004) the definition of network security is “a collection of network-connected devices, technologies, and best practices that work in complementary ways to provide security to information assets" (p.6). However, securing a system and the host devices is not enough, as the infrastructure of networks that link the internal system, servers and its end hosts with the Internet must also be secured (Bishop, 2003). The security of wired network systems is divided into two sections (Mana & Munoz, 2006). The first section is securing of end hosts and the second section is securing the infrastructure. One of the main requirements to secure the end hosts is to ensure that the trusted computers communicate with each other securely, by identifying legitimate users and hosts (Mana & Munoz, 2006). Network administrators must show network threats and vulnerabilities in a comprehensive way that allows prediction, analysis and correlation for different types of attacks that may cause significant damage (Daley, Larson, & Dawkins, 2002). Thus, the network and system administrators must focus and balance securing both sections at the same time. Any failure in one of these sections will increase of the possibilities of negative impacts on the other.

Network security may also be achieved by ensuring the following six requirements are implemented and deployed in the network correctly (Sudin, Tretiakov, Ali, & Rusli, 2008). (1) Confidentiality. The implementation of the encryption method or approach will keep the information unreadable to users or hosts that are unauthorized in the network. (2) Authentication. Identify a legitimate user or host and deny unauthorized hosts and/or users from accessing the system or information. (3) Integrity. Protection of the data from being altered or deleted during transmission in the network by attackers. (4) Non-repudiation. Ensuring that users that send information cannot deny the data was sent from their account. (5) Availability. Maintain of network resources and services available to genuine users or customers, allowing the services to be protected against attackers' incidents. (6) Access control. Denying untrusted users and/or hosts from obtaining the information.

The wired network system consists of different hardware, software, user privileges, and features. All these need to be secured in order to have secure network systems. The effect of insufficient protection in a network, particularly for products or services can seriously damage the corporate reputation. For example, the loss of money, and potentially customer retention and relationships. An insecure network will be more attractive to attackers, and this can lead to the increase of the number of cybercrime attempts against these networks.

Securing of the networks, systems and its devices plays an important part in security. However, the networks and system security is more extensive and involves written procedures and policies that impact the infrastructure of the network.

2.2 OBJECTIVE OF NETWORK SECURITY

In order to protect network resources and transmission of information over a wired network, security services should be applied in and/or at all layers. There are two services that should be implemented properly; the protective measure and defence mechanism. Firstly, the protective measure may be considered as a proper level of awareness and a prerequisite for adequate sufficient network protection (Goodhue & Straub, 1991). Secondly, a defence mechanism is implemented to protect data and systems against attackers. Similarly, Bishop (2003) believed that the security mechanisms could be operational or technical, and the aim is to ensure that a prohibited situation never happens to the system. Deploying security devices, such as Firewalls, Intrusion Detection System (IDS), and Intrusion Prevention System (IPS) will support the robustness of the system, by giving the administrator more tools to control the system as a whole.

In general, there are three fundamental objectives in computer network information systems that can determine the performance of the network security: Confidentiality, Integrity, and Availability (CIA). Confidentiality defined as the device and transmission of network data to ensure the access by authorized users or nodes only. Maintaining confidentiality in the network devices can reduce and/or prevent unauthorized users from accessing and analysing the information. Integrity refers to illegitimate users that cannot add, delete or alter the network information. Deploying integrity will help to guarantee that the information passes through the network devices and at is kept as it is sent. This also prevents the attackers from modifying data. Availability means that the services, applications and information are available to authorized users and nodes at all times, without denying legitimate nodes and users. Making resources available to legitimate users insures that there is no such attack being made to the services, such as DoS attacks. Maintaining availability can be achieved in various ways, for example, providing Firewalls and proxy servers, performing simultaneous backups, and access control lists (Zhiyong & Yunyan, 2014).

If there is any lack in these CIA factors, the attacker will be able to compromise a host or system, and retrieve sensitive data or even destroy the system as a whole.

2.3 NETWORK VULNERABILITIES AND THREATS

With advanced defence tools and methods available on the Internet, and articles and books, the network analyst is able to identify levels of network security comprehensively and relatively. This assists network analysts to determine whether their network level is safe or unsafe, and also to choose the best time and date to add patches in order to protect the organizations' networks. Chasaki and Wolf (2012) state that most of the security problems have been concentrated on end hosts and servers, for example, routers and firewalls, more than network infrastructure.

These problems may cost a lot of money for not only organizations or individuals, but also losses in the global economy. Lesk (2011) reported that in 2010, 65 percent of individual users' lost approximately US$300, and the total personal loss reached upwards of US$560 million. Yet, the global cost was around $1 trillion. The cost of loss in the United States (US) is approximately $105 billion annually, whilst the security defence in the US alone, costs around $67 billion per year. In the United Kingdom (UK), security cost around £650 million and losses would be as much as £210 billion per year.

Other studies made by McAfee (2015) reported that the cost of profits, criminals, defence and recovery for companies is higher than US$445 billion. McAfee estimated that the global losses would be between US$375 and US$575 billion per year. Within the individual sections reported, McAfee reported that there are approximately 800 million people losing personal information globally. These losses could cost more than $160 billion annually. In the companies' section reported, the cost of these incidents is greater. For instance, a company not specifically mentioned but within the UK says that the cost of loss was around US$1.3 billion. The economies of nations also suffered from losses. Countries, such as Japan, US, Germany and China had total losses close to US$200 billion. In addition, the cleaning up of these threats and vulnerabilities was more expensive than their impact or damage on the network systems of the organizations. For example, in Italy the cost of losses from cybercrime is $875 million yearly; whilst the recovery and opportunity costs was approximately ten times more than the actual loss. This means that the lack of sufficient protection of the network security will increase the cost of the recovery as well as the damage. However, there has been a dispute over the measuring the actual cost from malware and/or viruses, denial of service attacks, misuse of access privileges, and equipment damages. Therefore, it is hard to accurately measure the cost, either financially or in reputation damages (Mercuri, 2003).

2.3.1 Network Vulnerabilities

Network vulnerabilities can be found in many sources, including software that are implemented via the network, hardware, human actions, procedures, and policies. The definition of vulnerabilities of the system is found in many studies, such as Kizza and Migga (2009, p.87) who stated, "System vulnerabilities are weaknesses in the software or hardware on a server or a client that can be exploited by a determined intruder to gain access to or shut down a network". Since vulnerabilities can be located in various areas in the system of the network, the exploitation of these vulnerabilities by an attacker can cause a potential threat to the network (Maggi, Pozza, & Sisto, 2008). The possible common sources of these vulnerabilities in networks are as follows:

Firstly, technology flaws. The operating system of the computer, software, protocol and hardware, quite often have security weakness that can be used to obtain access into a system. Keramati and Keramati (2014, p.883) stated, "Vulnerability is a mistake in the software that can be directly used by a hacker to gain access to a system". There are 25 out of 100 security weaknesses in the operating system and 50 out of 100 are known vulnerabilities in the network environment especially in TCP/IP. Another point recalls that TCP/IP is considered a critical vulnerability (Chang, Jain, Slade, & Tsao, 1999).

Secondly, misconfiguration flaws. The network may consist of a large number of devices that communicate with one another. Misconfigurations in any network devices may cause damage to network security (Ritchey & Ammann, 2000).

Thirdly, social engineering is another source of vulnerability. Social engineering can be defined as the use of psychological tricks on legitimate computer and network users in order to obtain credentials and other information from them. Such as usernames, passwords, and credit card information. It can be performed in various ways, such as using website cloning, impersonation of an individual, and the redirection of emails (Hunt & Slay, 2011). This happens as more users have access to computers daily, which are attached to the system of networks.

Next, human flaws. Human error is one of the major keys in computer security. Errors being made by humans can pose serious vulnerabilities to the networks. For example, using a simple password or sharing confidential information with unauthorized users.

Fifthly, poor network monitoring. Network security devices are installed in order to protect and monitor network activities. Insufficient monitoring could be used by an intruder in order to access the system and gain critical information (Wolf, Chandrikakutty, Hu, Unnikrishnan, & Tessier, 2014).

Therefore, discovering and finding solutions to these vulnerabilities in all network system devices is one of the most effective and direct ways to safeguard the system and network (Yonglin, Yongjun, Xin, Zhanrui, & Jie, 2011).

2.3.2 Network Threats

Various types of threats from different sources may face any computer or network system that is connected to the Internet (Geer, 2005). Threats in computer network environments have different formats and types. One of the most significant and comprehensive definitions of the network threat was presented by Russell and Gangemi (1991). They stated the network threat "is a possible danger to the system; the danger might be a person (a system cracker or a spy), a thing (a faulty piece of equipment), or an event (a fire or a flood) that might exploit a vulnerability of the system" (p.11). Furthermore, more known vulnerabilities are available in the computer network, which may lead to increasing the possibilities of threats that occur in the system and cause serious damage. Therefore, the network requires a protection approach to protect the system, resources and its data. There are many common sources for threats, and those sources can be categorized into the two types, intentional threats and unintentional threats.

2.3.2.1 Intentional threats

Intentional threats mean that the human actions or activities are involved to deliberately target a specific network, node or system, for example, utilizing an attack on a target network, creating malware or a virus to attack a network or system; and attempting to cause harm and damage (Xiaofeng & Shifei, 2012). One of the most obvious findings from the research is the number of consumers who suffered from security breaches in the USA. The number of victims exceeded 110 million in one month; during December 2013, the information was obtained from Target Corporation, the enormous US retailer (Wu, Feng, Wang, & Liang, 2015). The common types of intentional network threats are external and internal as follows (Hori, Nishide, & Sakurai, 2011):

The first type of network threat is the external threat, which threatens a network or system from the outer boundary of the network. In 2005, the total number of attacks was around 310 million when the first analysis was made. The majority of these attacks came from China, at 44 percent, and the rest of the attacks were from other countries, such as the USA, South Korea and Russia (SecurityWeek, 2015). There has also been analysis by the National Institute of Information and Communications about the number of external attacks on the Japanese network in 2014. The Institute described the number of attacks, which reached 25.66 billion during 2014, and targeted the network system of Japanese Government offices and other entities only (SecurityWeek, 2015). These results indicate that the number of attacks been increased rapidly.

The second type of network threat is internal threats, which occur within internal organizations or system networks. Most researchers and organizations focus on the external network security threat and do not place specific emphasis on internal network threats. Pfleeger and Stolfo (2009) detailed that in 2007, 59 percent of participants have threats from within the computer network, usually by misuse of the resources of the network. Additionally, a quarter of respondents highlighted that incident from inside cost 40 percent of their losses financially. The protection of the network from external threats may be considered as a difficult task for network administrators, while the internal threats are more dangerous (Alkhishali, Abou El Farag, & Mohamed, 2010). Therefore, network administrators must deploy and install an internal network security device that firstly alerts to an attack and secondly protects the network system from illegitimate internal abuse by legitimate users.

Section 2.4 will discuss and address the type of intentional network attack more comprehensively.

2.3.2.2 Unintentional threats

Other types of threats that can be classified as unintentional threats that include natural catastrophes. This means that threats may occur naturally with no interaction from any users. Unintentional threats also include threats to the network system that take place without participation of user actions directly or intentionally. These threats may occur outside or inside the range of the network system. According to Onwubiko and Lenaghan (2007) the main type of unintentional threats are as follows:

Firstly, natural disaster threats. These threats are considered as an outside threat. This threat can be defined as an environmental condition that occurs naturally and human activities are not involved. Examples of this are the likes of floods, earthquakes and wildfire. In addition, this threat also could damage the whole or part of the network system, and may have a serious impact, such as loss of information and availability.

Secondly, software and hardware malfunction. These threats are inside threats. The network devices and software may fail, or cause errors in the system at any time with or without reason (Klevinsky, Laliberte, & Gupta, 2002).

Thirdly, faults of development. These threats could occur by indirect user actions including: the development or upgrading of a systems software or misconfiguration of a device on the network system. Both misconfiguration and upgrading may create security breaches, such as bugs, openness and open unprotected or unused ports in the system, which can be used by an attacker to access the systems' information and resources.

Fourthly, human errors. The final source of unintentional network threats is users' activity. This is an unintentional threat that creates and occurs in the boundary of the network range by human activities, which already have privileges to access the system, either with or without awareness of an actor's actions. For instance, an employee plugs an unsafe or unknown USB device into the computer, and downloads and opens a document file, software or picture from the Internet. This in turn opens a malicious email or website (Mendyk-Krajewska & Mazur, 2010).

2.4 NETWORK ATTACKS

Network attacks mean access to networks, computer systems, information, and programmes by illegitimate users, whether the access is virtual or physical (Hopper, Hopper, & Womble, 2009). However, the definition of this attack implies network destruction or the attempt to destroy the network system. In fact, the main two reasons in general are intelligence gathering and financial gain (Whitaker & Valentine, 2008). However, Akhgar et al. (2015) mentioned that there are several reasons more than the intelligence gathering and financial gain that could encourage the attacker, such as political and personal satisfaction. There are many inspirational reasons behind the personal motivation of an attacker to launch an attack. There are some attackers that may enjoy the satisfaction of successfully accessing others’ networks. Furthermore, there are many documented cases of hackers who purposely attacked organization or government networks to show that it is not protected, or just for fun (Sallhammar, Helvik, & Knapskog, 2005). In addition, the network attacks can be performed by using sophisticated tools, or operating system techniques, such as Linux, that can protect and safeguard the attackers and allow them to accomplish their attack(s) successfully. Hoque, Bhuyan, Baishya, Bhattacharyya, and Kalita (2014) stated the network attack is “an attempt to bypass security mechanisms of a network by exploiting the vulnerabilities of the target network” (p.308). These attacks have forced some organizations to cooperate with universities in order to monitor and detect the network attacks activities more quickly. For example, New Zealand has stepped forward in looking to prevent cyber-attacks. There is an approach of cyber security that was designed and launched on 25th March 2015, and this new system will help to grow the anti-malware armoury. Additionally, this system will detect the cyber-attacks and alert organizations and companies about when it happens. Japan’s National Institute of Information and Communications Technology (NICT) and Auckland’s Unitec Institute of Technology created this system (Unitec.ac.nz, 2015).

The following subsection will discuss the classification of network attacks.

2.4.1 Physical Attack

A Physical attack refers to an attack that targets physical network devices and causes damage to a system, or network. This attack can be a hacker, software, malware, spyware, rootkits or virus that are considered as a source of the harm (Zhioua, 2013). One of the most well known virus attacks that targeted the network infrastructure devices physically that occurred fairly recently is Stuxnet. Stuxnet influenced the physical network infrastructures that are managed and controlled by software (Chen & Abu-Nimeh, 2011). These types of attacks are raising researchers' focus on the malware field or technology due to two reasons. Firstly, the targeted machines in this case were crucial for nuclear developments or a specific network system. Secondly, Stuxnet has shown a very high level of sophistication to bypass the security defences. Thus, Stuxnet has raised new implications in the future for malware functions (Chen, 2010).

2.4.2 Virtual Attack

Virtual attacks refer to an attack by unauthorized users that targets information of a system, network or users via the Internet (Hopper et al. 2009). The aim and the hope of the attacker in general is gaining access into the system and then obtaining sensitive information from network devices. Additionally, each computer network may have had vulnerabilities that might be used by an attacker to gather specific data (Yonglin et al. 2011). This information about devices may include: servers and their services, the version of applications, and type and/or version of Operating System. This information could be of value for the attacker, as a type of information gathering, in order to identify possible vulnerabilities within the system and evaluate this information to launch an attack or not. There are two types of attacks that can discover hosts and services on the network. Firstly, in Host Sweep attacks, where the attackers can learn about the hosts on the network. Secondly, in Port Scan attacks, the attackers can detect the services that run in the network (Selamat et al. 2011). Additionally, there are two techniques that may be used by an attacker in order to collect general information about the targets' system or network: active and passive scan. Active scan is faster and provides deep information, whereas passive scan can protect the attacker from a detection system that may generate alerts to network administrators, such as firewalls (Wilhelm, 2009).

Generally, there are four important steps or stages to launch an information attack against a network system that all attackers usually follow (Hoque et al., 2014):

Firstly, information gathering. This step for the attacker is to try to gather information about all vulnerabilities in the network system, such as operating system, open ports, Internet Protocol and version and type of the server (Shaikh, Chivers, Nobles, Clark, & Chen, 2008). This is done in order to collect as much useful information for the attacker that might be used later during the attack stage.

Secondly, assessing vulnerability. According to the amount of information on vulnerabilities that is collected in the first step, the attacker exploits a malicious code in order to compromise some machines in the network. This stage is considered a preparation step for the attacker to launch the attack.

Thirdly, launching the information attack. After the second step, the attacker is able to launch the attack by using those machines on the target network entries.

Finally, cleaning up the information attack. This is the final stage of the information attack, when the attacker will try to remove or delete all information and history that relates to the attack from the victims' host, such as log files or the registry.

2.5 THREE COMMON ATTACKS IN THE NETWORK ENVIRONMENT

In recent years, with advanced hacking tools and the reliance on the Internet by people, corporate's, and governments, there are many incidents and attacks that target networks and computers. Such as distributed denial of service (DDoS) attacks and other malicious accidents (Chia-Mei, Han-Wei, Peng-Yu, & Ya-Hui, 2013). Furthermore, generally an attack that targets a network system is more complicated than others are because network mechanisms and features that exist in a system may be very different to one another. Thus, there is no uniform term that can be used to describe all attacks that exist in the network (Shi, 2011). Furthermore, the attacks in a network environment could happen in a specific or non-specific attack.

Firstly, a virus, or a group of attackers, that work together against a specific target, can do the specific network attacks. Additionally, an attacker, who has the ability to code a programme of hacker scripts, and the technical skills and experience in using sophisticated tools, may launch this network attack alone. The attacker does so in order to discover and use the vulnerabilities of specific systems or networks and to be able to compromise that system (Stallings, 2011).

Secondly, non-specific attacks are the action of an attacker searching for any network and/or system vulnerabilities who launches the attack without awareness of the consequences that will follow the attack. For example, the attacker may spread a worm on the Internet, or launch an attack on any network. This attack could cause a more serious impact than an attack through the network.

Generally, the common attacks, both specific and non-specific that exist today in network environments can be classified in the following subsections.

2.5.1 Access Attacks

An access attack means an attacker obtains access into a network or account owned by users by using improper means (Heberlein et al., 1990). However, the network and system administrators must ensure that the legitimate users only have the rights to access the account in the system and/or network devices. This will include the following:

Man-in-the-middle attacks (MITM). Each peer-to-peer communication in the network occurs between two devices that talk amongst each other without interception by a third party. MITM attackers initiate independent connections with a connection to the two victims and receive the messages that are sent between them. Furthermore, both victims do not realize that they have lost the integrity of the message due to the attacker being able to read and control messages (Altaher, Ramadass, & Ali, 2011).

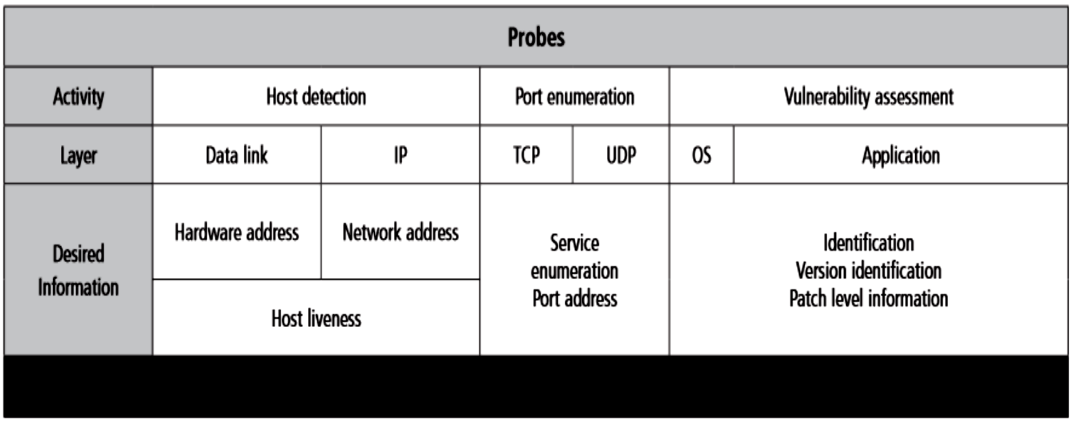
Secondly, password attacks. Most users need a password to authenticate their login into a system, website or application. Many users use simple passwords because it is easy to remember them or use one password for their multiple logins credentials. However, there are two techniques that can be used to obtain their password; brute-force and Dictionary attacks. These two techniques cause many security breaches in network and system security defence mechanisms (Kassim & Sujitha, 2013).

Thirdly, port redirection, which is also known as trust exploitation form, which will be discussed in the next paragraph. Access to internal networks must be by a machine that has the right to access and pass the traffic. This traffic is passed via a port on an access control list (ACL) or firewall. Usually the unauthorised traffic is denied by the port or by compromising a trust host on the network, allowing an attacker to use a redirection port in order to create a tunnel for communication and bypass the security measures of the network (Dattani, Thanthry, Best, Bhagavathula, & Pendse, 2004)

Lastly, an attack based on trust exploitation occurs when an attacker acts as a trusted party or host in order to communicate with the victims' nodes. For instance, if the attacker has compromised a server that exists on a demilitarized zone (DMZ), the clients' nodes are already configured to trust the communication with all servers that exist on the DMZ. The attacker can then initiate a connection to all clients that trust that server, and obtain data or create another attack (Yongle & JunZhang, 2013).

2.5.2 Reconnaissance Attacks

Reconnaissance attacks permit attackers to explore or recognise network or systems vulnerabilities before they launch an attack (Shaikh et al., 2008). For example, an attacker may use a tool in order to gather information about hosts on the system after identifying whether the system is connected or not. Then an attacker will try to obtain information from the system that is live, such as version and identity of the operating system, which system ports are closed and/or open and which applications are used in the network system. With gathering or gaining more information about the system, the attacker can launch an attack that becomes more effective (Morris, Vaughn, & Dandass, 2012). Furthermore, the main information that is most valuable to a hacker includes: the identification of the active networks and hosts that are connected to the Internet through an accessible or a public medium, the vulnerabilities of services and applications that are running in the system and which application could be exploited, as shown in Figure 2.1 (Shaikh et al., 2008). Gathering information about a system will usually include the following methods:



Probes						
Activity	Host detection		Port enumeration		Vulnerability assessment	
Layer	Data link	IP	TCP	UDP	OS	Application
Desired Information	Hardware address	Network address	Service enumeration Port address		Identification Version identification Patch level information	
	Host liveness					

Figure 2.1: Network Reconnaissance (Shakh et al., 2008, p.13)

Gathering information by queries. One of the queries that is commonly used to gather information about hostnames from Internet Protocol addresses or vice versa is an nslookup query (Shaikh et al., 2008). This query is available in most operating systems such as, Windows and Linux systems (Robledo, 2008).

Packet Sniffing. An attacker can only sniff the packets by using software that is running on a device that is attached to the network devices physically. This device receives frames or packets that are passing through the adapter of a network device. This method is also known as Ethernet Sniffing and Protocol or Network Analysing (Qadeer, Zahid, Iqbal, & Siddiqui, 2010). There are many sniffer tools that are available on the Internet such as Tcpdump, Wireshark, and Kismet, some of which are open source. Furthermore, some of them are commonly used in organizations' networks to intercept and log protocol information. For example, segments, packets, and frames. Implementing a packet sniffer in the network makes the monitoring of exchange sequence and contents simpler. However, the overall view of behaviours of the protocol is not provided by sniffers tools (Ming-Hung, Chia-Ming, Chia-Liang, Chien-Chao, & Li-Hsing, 2014).

Port scans. Dabbagh, Ghandour, Fawaz, Hajj, and Hajj (2011) state that there “are many network intrusion methods that can be considered as dangerous”. One of these is the port scan that can be used to discover exploitable communication channels. Port scan can be one of most popular techniques that is used by attackers in Reconnaissance techniques to learn about which services attackers can break into. Port Scanning is mainly the search for open ports in a network device. The port scan is a form of gathering information that aims to discover information about services that are running on the target. There are many techniques that are used today in port scan probes, such as NULL scanning, TCP half-connect scanning, Xmas Tree scanning and TCP connect scanning. Generally, the port scan consists mainly of two parts: Vertical and horizontal scanning. When an attacker is gathering data about a particular host on the network and then launches an attack, the vertical scan is commonly used because it scans the host and provides the information about multiple ports that exist within that host. The vertical scan scans multiple hosts at one time. In contrast, horizontal scans use a different methodology of scanning hosts, as they scan one port over multiple hosts. This means that the attacker can scan a certain port in multiple devices that are connected to that port, which is very useful for the attacker in order to take control over the victim(s) devices by discovering the port services that are running and exploiting the vulnerabilities on that port (Allen, Marin, & Rivera, 2005).

Finally, ping sweeps. Hackers use the ping sweep in order to validate any IP address that is available on the network. The ping tool works by sending an echo request to a target IP address and that IP will accept that echo request and reply with an echo. Additionally, the tool of the ping sweep has the ability to send at the same time, an echo request to many IP addresses in order to discover which machine(s) will accept the echo and send back an echo reply (Shun & Malki, 2008).

2.5.3 Denial of Service (DoS) Attacks

The denial of services attack is one of the most well known attacks to have occurred recently. The DoS attack is usually created by an attacker to deny a legitimate user from using services that are generally available to an organization or a user. When a successful DoS attack is launched against an organization network, users may have no access to their resources. In addition, an attacker can use IP spoofing or man-in-the-middle during DoS attacks. This will allow the attacker to gain the advantage of using a machine that has the right to bypass the security measures. The ‘ping of death’ is another name for DoS attacks because they send large amounts of packets that contain more than 65,535 bytes of ICMP echo request in order to cause an overflow, which may collapse or even crash the target system. A teardrop is an example of a DoS attack that can crash the system by running the CPU to reach its peak, at 100 %. The teardrop will send small fragments in thousands plus overlapping offsets (Kiuchi, Hori, & Sakurai, 2010). Denial of services attacks include a smurf attack, distributed denial of services attack (DDoS) attack or TCP SYN attack. The description of each one' follows in the following subsections.

2.5.3.1 Distributed denial of service (DDoS) attack

Wang and Wang (2008) claim that the developing attack behaviour of Denial of Service (DoS) leads to creating a Distributed Denial of Service (DDoS). The main difference between DoS and DDoS is that the DDoS can be automated. Furthermore, there may be the ability of DDoS to be able to handle or coordinate numerous processes of computers to launch the attack. DDoS attacks can overload the network resources and cause the collapse of a target network.

2.5.3.2 TCP SYN attack

The functionality of TCP/IP that is implemented in the network environment gives an attacker the use of the TCP/IP functions to launch the attack. The way that TCP/IP works, is when a client sends a request by sending a SYN message to a specific system, the system will return a SYN_ACK to the sender. The last step is for the sender to reply with an ACK packet, which will close the connection. This method is called three-way the handshake. In this attack, an attacker will send many requests with spoofed IP addresses to a target a system. The reply will send back to the attacker's system and wait for their ACK to close the connection. The ACK will not send because the IP that sends the request is coming from an incorrect IP address. Thus, the connection will stay active for a length of time and cause the victim to run out of resources for the system. Furthermore, the system will drop the request because of the period on time-out. As a result, because of the large number of requests that are sent by the attacker, the system will not be available to legitimate users (Guangzhi, Hariri, & Yousif, 2005).

2.5.3.3 Smurf attack.

This attack may rely on misconfigured devices of the network as discussed in Section 2.2.2.2. It allows the request to be sent into multiple hosts rather than a particular host. In this attack specifically, the attacker will broadcast multiple ping requests, which are sent to a particular target in order to overwhelm the target's resources as shown in Figure 2.2. In addition, the devices on the network can then assist and work as a smurf amplifier. However, the network administrators may add a command of no IP directed-broadcast to a router, which will help to mitigate this type of attack and its potential risk (Verma, Hasbullah, & Kumar, 2013).

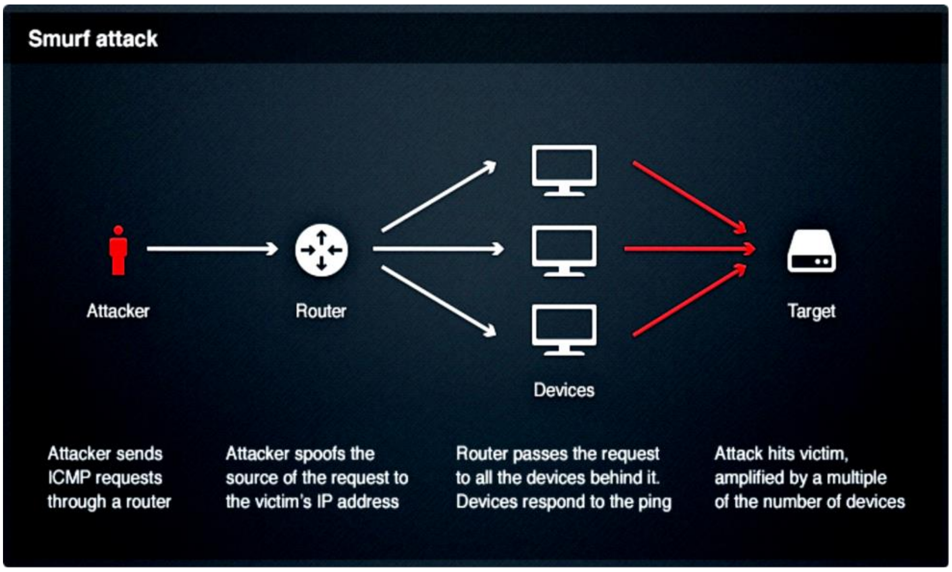

Smurf attack
Attacker
Router
Devices
Target
Attacker sends ICMP requests through a router
Attacker spoofs the source of the request to the victim's IP address
Router passes the request to all the devices behind it. Devices respond to the ping
Attack hits victim, amplified by a multiple of the number of devices

Figure 2.2: Smurf attack (Cloudflare.com, 2015)

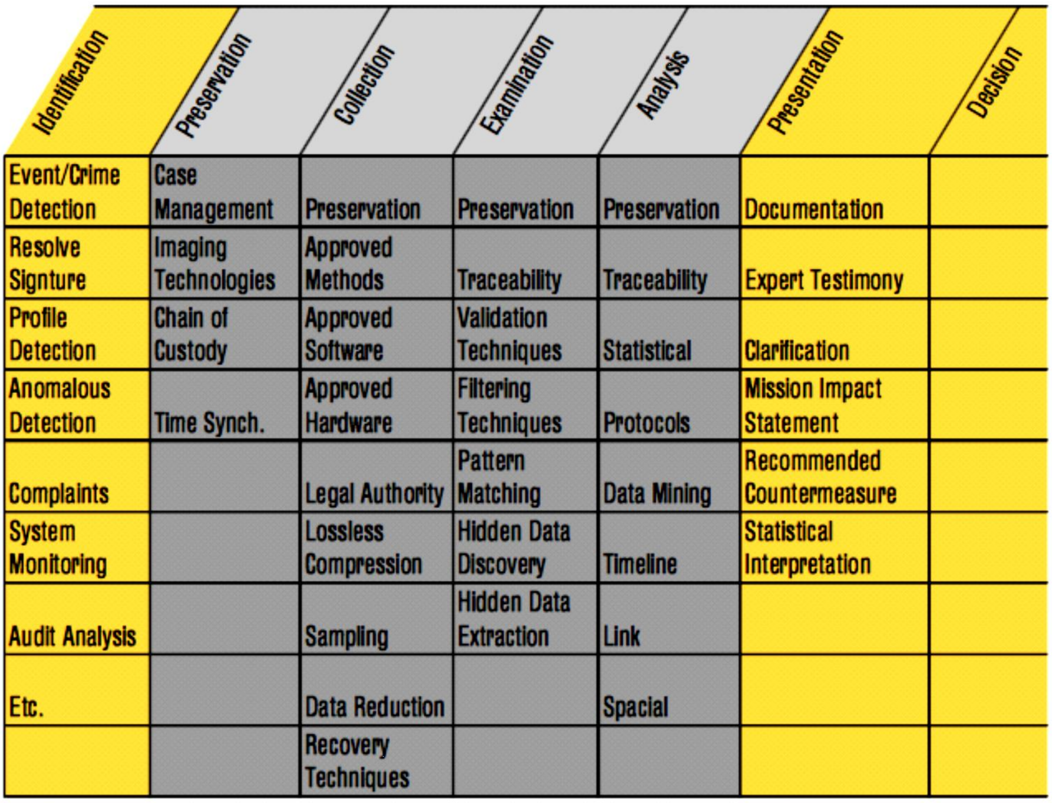
2.6 DIGITAL FORENSICS

Recently, many incidents and crimes have occurred that involve network and computer technology, such as Internet fraud. The number of these incidents has grown as discussed in Section 2.3. Computers have many vulnerabilities as discussed, in Section 2.3 that can be used by an attacker. Therefore, there is a need for a product that can help law enforcement and others to collect evidence in the form of computer–based information in order to prosecute criminals (Rekhis & Boudriga, 2012). The initial development of the computer forensic programmes was specific computer forensic tools used by the FBI laboratory and other agencies of law enforcement in early 1984, for examining digital evidence (Yong-Dal, 2008). Network and computer forensics ensure that evidence of the crime related to digital information is acceptable to a court.

Selamat et al. (2011) state that digital forensics is a “subset of forensic science that indirectly covers computer technology crimes”. While digital forensic as a science had been stated by Palmer (2001) as:

> “The use of scientifically derived and proven methods toward the preservation, validation, identification, analysis, interpretation, documentation and presentation of digital evidence derived from digital sources for the purpose of facilitating or furthering the reconstruction of events found to be criminal, or helping to anticipate unauthorized actions shown to be disruptive to planned operations” (p.16).

Palmer noted that digital forensic science investigates different network devices and consists of several steps that start from identification and preservation and through to decision making as shown in Figure 2.3. Digital Forensic Research Workshop (DFRWS) published the model in 2001 as a baseline for most digital investigators and other researchers.



Identification	Preservation	Collection	Examination	Analysis	Presentation	Decision
Event/Crime Detection	Case Management	Preservation	Preservation	Preservation	Documentation	
Resolve Signture	Imaging Technologies	Approved Methods	Traceability	Traceability	Expert Testimony	
Profile Detection	Chain of Custody	Approved Software	Validation Techniques	Statistical	Clarification	
Anomalous Detection	Time Synch.	Approved Hardware	Filtering Techniques	Protocols	Mission Impact Statement	
Complaints		Legal Authority	Pattern Matching	Data Mining	Recommended Countermeasure	
System Monitoring		Lossless Compression	Hidden Data Discovery	Timeline	Statistical Interpretation	
Audit Analysis		Sampling	Hidden Data Extraction	Link		
Etc.		Data Reduction		Spacial		
		Recovery Techniques				

Figure 2.3: Digital forensic processes (Palmer, 2001, p.16)

Furthermore, digital forensics investigates the evidence relating to a computer that should be reliable, sufficient and convincing to be effective and acceptable to a court

(Fink, North, Endert, & Rose, 2009). In addition, digital evidence is defined as "any data stored or transmitted using a computer that support or refute a theory of how an offense occurred or that addresses critical elements of the offense, such as intent or alibi" (Casey, 2011, p.7). Casey mentioned that the devices, which are used as a source of evidence, could be divided into three sections. The first section, open computer systems, is what most people identify with such as traditional cell phone systems, Internet, network, and wireless communication systems. For example, a message that is sent by phone can be used as evidence. This message will show the time of sending, who sent it and the integrity of the message, and all of this data can be important in the investigation. The navigation system device can be included in section two. The final section consists of embedded computer systems. These devices embed the computer system in the operating system, for instance, digital cameras, smart phones, iPods and other smart devices. For example, the smart phone may contain digital evidence in a form of hidden evidence in a picture (Steganography) or communication and Internet history that might be useful in the investigation processes. Sivaprasad and Jangale (2012) described the sources of digital evidence, and added more sources than these three sections, such as chat transcripts, password protected files, deleted files, emails, and encrypted files. In their research, they also mentioned that online frauds for example, banking, credit card, tax evasion and share trading could be a source of digital evidence. Furthermore, they reported that other sources of digital evidence could be found in hacking activities, such as email hijacking, cyber-attacks, cyber sabotage, denial of services (DoS) attacks and other similar activities. Sivaprasad and Jangale (2012) highlighted that when computer security breaches have occurred, digital forensics will play an important role in discovering the digital evidence. To sum up, the main function for the analysis of data created with and contained within computing devices and computer systems; is specifically in the interest of discovering what occurred, when it occurred, how and when it occurred, and who had participated. Digital forensics is a method used to determine an occurrence within a computer system and the reasoning for the incident.

2.7 NETWORK FORENSICS

Network forensics is related to the analysis of accidents or hacking activities that occurred in the network environment. It includes collecting information from all active network intrusion detection devices by network forensic investigators that have been used to protect the network system. This data will be analysed using network investigation tools in order to find any evidence that can be used in the courtroom. These tools will be discussed in detail in Section 2.9. However, some information that has been captured will be useful for examination whilst some will not.

According to Mukkamala and Sung (2003) the definition of network forensics is "the act of capturing, recording, and analysing network audit trails in order to discover the source of security breaches or other information assurance problems" (p.2). Palmer (2001) has a different definition of network forensics, which could be more comprehensive:

> The use of scientifically proven techniques to collect, fuse, identify, examine, correlate, analyse, and document digital evidence from multiple, actively processing and transmitting digital sources for the purpose of uncovering facts related to the planned intent, or measured success of unauthorized activities meant to disrupt, corrupt, and or compromise system components as well as providing information to assist in response to or recovery from these activities. (p.27).

Garfinkel (2002) states that there are two main approaches to network forensics: "Catch it as you can" and "Stop, look and listen". In his research, he also detailed that the base of the most common approach to network forensics is on monitoring traces.

There are many challenges that may face the network investigators. According to Petersen and Wiil (2011). One of these challenges is how to transform the raw information into reliable data that can be used in a court. They recommend involving other expert people and to check related work and to examine methods that are used by others, such as police cyber-crime investigations guidelines. A live system may provide different evidence that cannot be stored in the system or in computer hard drives or other devices. Adelstein (2006) mentioned that live forensic investigations operate within multiple constraints, putting more effort into finding evidence that cannot be found if the system is disconnected from the network. Thus, finding evidence from a live network can be more effective. Almulhem (2009) reported that another challenge is a lack of details, which means that the information that is collected from protection devices, such as IDSs, Firewalls or other devices is not enough to discover or find the necessary evidence in some cases. Some of these devices cannot store all traffic information that passes through the network because of their inabilities to store a huge amount of information or their lack of capacity to store all information. In addition, Mohay (2005) added more to forensic challenges that include an affordable price of the forensic tools that can achieve investigators' goals and provide evidence acceptable in law enforcement. In addition, the monitoring of networks to develop methods, which, will enable networks and hosts to discover hacking activities in order to reduce damage and prevent the diffusion attacks. The analysing of large amounts of information that can create a number of challenges. Petroni, Walters, Fraser, and Arbaugh (2006) detailed that collecting evidence from a volatile system memory can create other challenges for network forensic investigators for two reasons. The first reason is the nature of transferring data without changing the data over high-speed networks, and the second reason is the complexity of gathering data from different devices.

2.8 NETWORK INVESTIGATION LIFE CYCLE

Network security devices are deployed to protect the network resources from attack activities whilst the network investigation will try to discover who performed the attack and how it was done. Every investigation of cyber-crime has a life cycle with the same principles, starting from identification working through to the reporting phase. In each phase, the investigation involves many tasks in order to obtain reliable evidence. These tasks can be achieved by either human knowledge/experience and by using available tools/software (Wiil, 2013). There are several model of the LAN network investigation processes that have been produced by investigators and researchers. Spafford and Carrier (2003) proposed a model of the digital investigation method based on physical investigations techniques. Their model includes different phases, such as readiness, that covers the operation infrastructure, survey and the documentation phase that collates all evidence. In contrast, Pilli et al. (2010) proposed a model of network forensics containing nine steps as shown in Figure 2.4. The nine steps of this model are:

Preparation: The environments are deployed and network security devices are installed that have sensors, such as the intrusion detection system, firewalls, and other security devices and software at various locations on the network. However, there may be a need in this step for legal warrants and authorizations in order to avoid violating the rights of individuals.

Detection: There are various security tools that generate alarms for many malicious activities that indicate there is a policy violation or security breach occurring in the network. These activities will then be analysed. Different parameters will be determined by the nature and presence of the attack. The evaluation of the possible attack will be confirmed by a validation check. The decision well then be made based on these alarms as to whether to continue the investigation if the alarm is true, or ignore them if the alarm is false. Caution should be taken in order to maintain the nature of evidence from being changed in the process. There are many tools assist in this step, such as, Wireshark, Sebek, Ntop and others. This stage consists of two steps: incident response and collection.

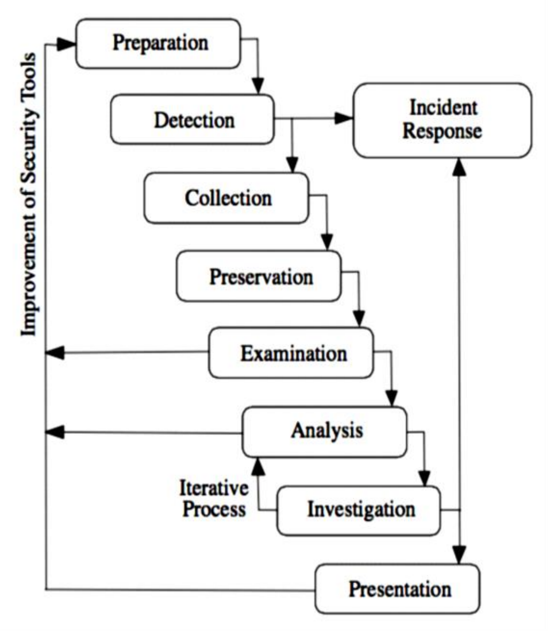


Figure 2.4: Network investigation model (Pilli et al. 2010, p.20)

Incident response: The response to intrusion or a crime detected is initiated based on the data collected to validate and estimate the incident. The response depends on the identifying the attack type and is guided by business constraints, legal constraint and the policy of organization. Two plans are initiated: recovery from damage that occurred and defence from the possible attacks in the future. At the same time, the decision is taken whether to collect more data and continue the investigation or not. After the investigation step (it will be discussed below) a similar response is to being where the data obtained may require specific action to mitigate and control the attack.

Collection: Information is found through sensors that are used to gather traffic data. An outlined process using a reliable range of tools from software and hardware is begun. These must be in place to effectively find information whilst causing minimal disruption to the target network. These must not only have maximum efficiency but must also work quickly as it is very hard to recreate the same trace later on. A large amount of information will be logged. This requires large memory space and systems to guide the different logs appropriately. There are many available tools, such as MySQLDump, TCPDump, Wireshark, TCPFlow, NfDump, Sebek, SiLK, TCPReplay, Snort, and Bro.

Preservation: The information found from tracing and the available logs needs to be stored on backup devices that have read-only limitation. Tracing data needs to be saved and hashed. All information and data found will remain in the same state and verified copy can be used for analysis. This process follows legal requirements so that when the investigator undertakes the process, it is recreated with verified data. Some of common tools available for this are: TCPDump, Wireshark, TCPFlow, NfDump, PADS, Sebek, SiLK, TCPReplay, Bro, and Snort.

Examination: The information from different sources is integrated and used to create a single data set on which analysis can be executed. Issues like overlapping of data due to time zones needs to be analysed and verified. Information gathered can also show-contradicting alerts and this also needs to be checked. A careful procedure must be undertaken, as the investigator must be cautions not to lose information. Evidence found is analysed methodically to find data, which may indicate misconduct or a crime. An attempt should be made to find the most effective evidence within a small data set. Recommendations need to be made in order to improve the security tools and minimize future threats. The tools available to help in this phase are TCPDump, Wireshark, TCPFlow, Flow-tools, NfDump, Argus, Nessus, Sebek, TCPTrace, Ntop, TCPStat, NetFlow, TCPDstat, Ngrep, TCPXtract, SiLK, TCPReplay, P0f, Nmap, Bro, and Snort.

Analysis: The datasets found are grouped and connected to extract the required information by basing this on the availability of attack patterns. Searching data to match attack patterns requires data extraction approaches based on soft computing and statistical analysis. Important considerations are based on network connection establishment, protocol, packet fragmentation, DNS queries, and operating system fingerprinting. The information on the attack patterns is deduced, recreated and rebuilt in order to understand the cause and methods taken by the attacker. A recommendation is provided again to improve the available security tools. Many of the tools that are found to help with this phase are TCPDump, Wireshark, TCPFlow, Flow-tools, NfDump, Argus, Nessus, Sebek, TCPTrace, Ntop, TCPStat, NetFlow, TCPDstat, Ngrep, TCPXtract, SiLK, TCPRe- play, P0f, Nmap, Bro, and Snort.

Investigation: The aim is to find the path from the victims' network through any intermediate network traces to the attackers' location. The information found and statistics collected are then analysed. This phase may require additional processes from the analysis phase; therefore, both phases are used in iterations to reach the result. Establishing whom the attacker or source of attack is may be the hardest step in the network forensic process. The two main methods used by the attacker to disguise themselves are often to spoof the IP used and to utilise a stepping-stone attack. This phase gives information regarding the attack and may lead to a prosecution.

Presentation: The observations must be demonstrated in an intelligible language for lawful personnel while offering an explanation of the different procedures used to reach the result. The formal documentation is included to cover the legal requirements. The findings should be shown in a visual manner so that they are easily understood. This phase finalizes the process of network forensic analysis as the data shown may be used in the prosecution of the attacker. The investigation results may be used in the future to give recommendations and to create guidelines for setting up and improving security products.

2.9 FORENSIC ANALYSIS TOOLS

Computer and network crime has been a challenge since the introduction of the Internet. In many cases, criminals are successful in breaking security mechanisms, and it is impossible to secure a system 100 % at all times. Most systems are prone to a variety of security breaches, no matter how sufficient and strong the security techniques appear on to be site. The digital forensic investigating services are required whenever such incidents happen. The ability of digital forensic investigators to accomplish their function is deeply dependent on their ability to acquire digital evidence from network devices and computer systems. However, technologies and computer systems are changing rapidly. This has a direct influence on the ability of forensic investigators to acquire and identify digital evidence (Sibiya et al., 2012). Both hackers and network forensic investigators use similar sets of tools and may have a similar skill level, but in the functionalities of the tools, they are working against each other (Berghel, 2003). The tools that are used in the extraction phase, when digital evidence is collected from networks or system devices, and in the analysis phase, as discussed in Section 2.8, must be tested forensic investigation tools. Thus, these tools can be accepted by law enforcement and other agencies in order to prosecute the criminals in courtrooms (Yinghua & Slay, 2010). Wazid, Katal, Goudar, and Rao (2013) divided these forensic tools into groups, such as computer, network, and other forensic tools, as shown in Table 2.1.

Table 2.1: Digital forensic tools (Wazid et al., 2013, p.140)

Computer Forensics Tools	Commercial	FTK Imager, EnCase, X-Way Forensics, The Coroner's Toolkit, PTK Forensics, OSForensic, Internet Evidence Finder (IEF), Intella
	Free	Helix, Live View, The Sleuth Kit, Open Computer Forensics Architecture, Digital Forensics Framework
Memory Forensics Tools	Commercial	Memoryze, Second Look, WindowsSCOPE
	Free	CMAT, Volafox, Volatility
Network Forensics Tools	Commercial	NetworkMiner, DeepNines, Omnipeek, PyFlag, Dragon IDS, nGenius Infinistream, RSA EnVision, NetDetector, Solera DS, E-Detective, IPFIX, netflow, NetVCR, NetOmni, NIKSUN Puma Portable, Rootkit Hunter, ssldump, tcpxtract
	Free	TCPDump, Windump, Ngrep, Wireshark, Driftnet, Airmon ng, Airodump ng, Aireplay ng, Aircrack ng, Kismet, Xplico, Argus, Fenris, Flow Tools, Honeyd, SNORT, Sguil, tcpflow
Mobile Phone Forensics Tools	Commercial	Radio Tactics Aceso, Paraben Device Seizure, MicroSystemation XRY/XACT, Oxygen Forensic Suite, MOBILedit Forensic, Cellebrite Mobile Forensics, eDEC's Tarantula, ABC Amber Blackberry Converter
	Free	NetSleuth, DECAF, Bitpim
Database Forensics Tools	Commercial	IDEA, Arbutus, ACL AuditExchange, SQLite Forensic Reporter

Additionally, Yong-Dal (2008) mention that the network forensic tools are sometimes called Network Forensic Analysis Tools (NFATs). The network forensics products may include: nstreams, slogdump, tcpflow, chaosreader, dhcpdump, and other software. However, Pilli et al. (2010) mentioned that monitoring network traffic, also known as network security and monitoring (NSM), should be involved in network forensics in order to determine if there is malicious activity or an abnormality in the traffic and assist in deciding whether it points to an attack. This system can determine the nature of an attack even if the attack is erased. Pilli et al. (2010) also reported that network forensic methods allow investigators to route back to the attackers. In addition, NFATs permit administrators to collect all data about abnormal traffic, monitor networks, help in investigation of network crime and assist in producing an appropriate incident response. Other benefits that can be gained from NFATs, are helping in analysing the misuse of internal resources and identifying insider hacking or theft, estimating network performance, predicting attack goals in future, performing risk assessment and assisting with guarding intellectual property. When network traffic is captured by NFATS, administrators can find out about significant structures in the traffic by analysing this traffic according to their needs. NFATs enable forensic investigators to analyse the network traffic, which can be viewed as a distinct connection of the transport layer between the hosts. Thus, forensic investigators are able to analyse the content of the packet, protocol layers, the data, which has been transmitted, and extract patterns of activity between hosts and networks, in order to find and identify the attacker, the method used, and provide admissible evidence that could be presented in a court.

Some commercial NFATs available offer extensive analysis capabilities and reliable information acquisition. In addition, there are various other free source monitoring tools and network security (which will be discussed in Section 2.10). They were designed with evidence processing and collecting in mind but they have no forensic standing. Otherwise, in particular forensic analysis activities, they can provide assistance. The various tools classifications are shown in Figure 2.5.

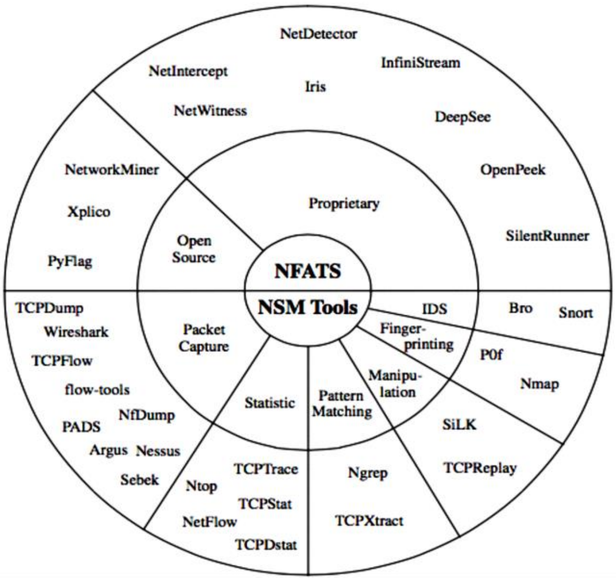


Figure 2.5: Network Forensic Analysis Tools (Pilli et al., 2010, p.18)

2.10 INTRUSION DETECTION SYSTEM OVERVIEW

Tools to provide security of networks, systems and host devices, such as firewalls, Virtual Private Network and mechanisms of authentication, are not sufficient to ensure that the infrastructure resources and system information on the network are secured. This leads the security researchers, vendors or developers to create a system that can monitor the systems, network and application, and identify an attack. This system is called the Intrusion Detection System (IDS) and it is complementary to routine security systems. There are some devices in the network that could protect and store different information that could be accepted and used in a court of law as evidence. For instance, IDS is used to increase the protection of the network and detect any malicious activities in the network. It is one of the most important devices in the network for protecting the network from attacks (Al-Jarrah & Arafat, 2014). The IDS can secure the network by detecting any attack behaviour or unauthorized access to the network. According to Lazarevic, Kumar, & Srivastava (2005) the intrusion detection systems is a hardware component and/or software that is used to monitor network activities and analyse events happening in the network for signs of intrusions. They stated that the IDS is “a combination of software and/or hardware components that monitors computer systems and raises an alarm when an intrusion happens” (p.21). Similarly, the National Institute of Standard and Technology (NIST) defined the IDS as:

> “The process of monitoring the events occurring in a computer system or network and analysing them for signs of intrusions, defined as attempts to compromise the confidentiality, integrity, availability or to bypass the security mechanisms of a computer or network” (Bace & Mell, 2001, p.5).

There are many types of IDS existing today, characterized by different analysis and monitoring approaches as shown in Figure 2.6. Approaches can be described in the following functional elements:

- Information Sources: the various sources of incident data used to determine if an intrusion was occurred. The source of intrusion detection can be classified into three main common categories: Network Intrusion Detection System (NIDS) will monitor the network in real time and sniff network packets; Host Intrusion Detection System (HIDS) will analyse user behaviours and activities on a given device; and, Application Intrusion Detection System (APIDS) will monitor particular applications only and other intrusion detection system (Bace & Mell, 2001). All these devices are very important for network investigators because they can collect and store relevant information on network and host attacks.

- Analysis strategy: there are two common approaches to analysis and detection of an attack: anomaly detection and misuse detection (Mukkamala, Janoski, & Sung, 2002). Anomaly detection monitors the system and identifies the intrusion based on unusual behaviour that occurs in the system, host, or network. While misuse detection will analyse system activities and identify an attack based on a matching predefined pattern of incidents that define a known attack. This detection is known as signature-based detection (Bace & Mell, 2001).

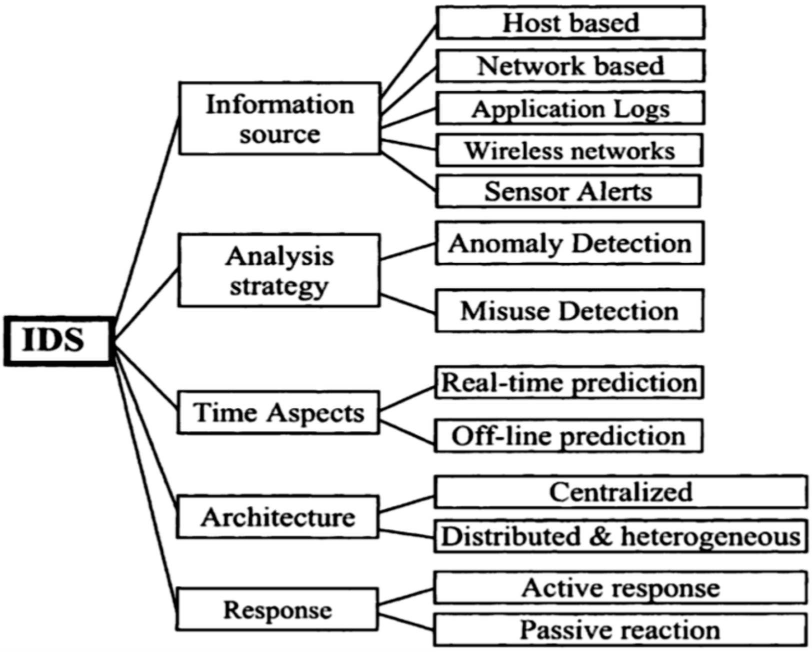


Figure 2.6: Intrusion Detection System criteria (Lazarevic et al., 2005, p.34)

- Time Aspects there are two primary groups: Off-line IDS and real-time (on-line). Off-line IDSs conduct post-analysis of audit information. The common use of this method is when administrators' analysts sometimes examine the behaviour of the network, as well as various attackers' behaviour. Real-time IDSs is a real-time or close real-time attempt to discover intrusions. This method operates when the sessions are in use and streams of continuous data are available from the data source (Lazarevic et al., 2005).

- Architecture of Intrusion Detection Systems varies in terms of the techniques and methods used for collection and analysis of the information. However, most systems are based on architectural framework. Figure 2.7 illustrates the main components of architectural framework for IDS (Lazarevic et al., 2005):
- Sensor (data collecting device) is responsible for gathering information from the information source - monitored system.
- Detector (Intrusion Detection analysis engine) finding intrusive events by processing the data that are gathered from the sensors.
- The knowledge base or the database that contains the pre-processed data format, which are gathered by the sensor for example: signature attack, data profiles, and the filtered data. Network and security experts usually provide this information.
- The current status of IDS is provided by the configuration device.
- Actions are made by the response component, this is initiated when the intrusion is being detected, and provides two responding mechanisms: Active, and passive.

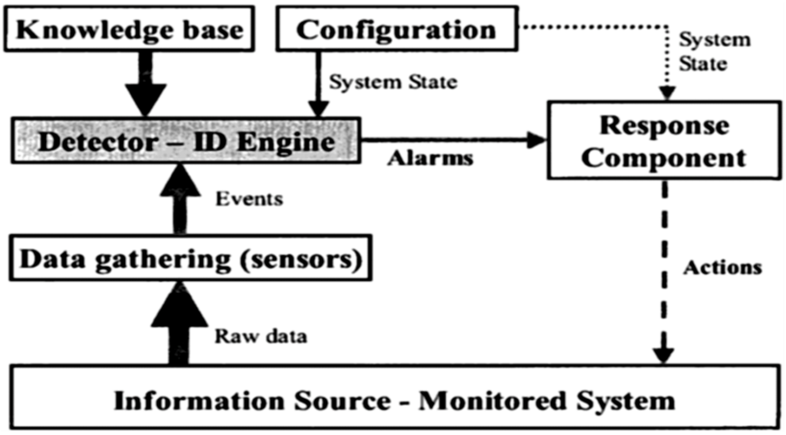


Figure 2.7: Architecture of Intrusion Detection System (Lazarevic et al., 2005, p.31)

- Response*:* Once the system detects intrusions in the system, a set of actions is undertaken. These actions are grouped into passive and active measures. Passive measures will report the finding of IDS to system security to take action as necessary based on the report. These reports are very important to network forensic investigators to investigate and find useful information about what occurred in the network. Whereas, active measures require some automated intervention on a portion the system (Bace & Mell, 2001).

2.11 CURRENT CHALLENGES

This literature review has discussed a number of issues regarding security and the specific issues with wired network security, including several types of attacks. There are many security points that have been discussed in this chapter that need more research in the future. These points are classified in the following subsections:

2.11.1 Lack of Standardization of the Digital Evidence Process

Many researchers and investigators have developed and provided guidelines for digital evidence investigation process that covers events relating to digital crime. Valjarevic and Venter (2012) and also Valjarevic, Venter, and Ingles (2014) believe that undertaking a digital evidence investigation process that can be accepted by law enforcement and court rooms, must follow a standardized and formalized process.

2.11.2 Technical Difficulties

One of the main problems with wired networks was discussed in Section 2.3.2 and was the network threats and their vulnerabilities. These threats cause harm to network security and create many security breaches that can be used by an attacker. Keramati and Keramati (2014) believe that there is a simple solution to solve these issues by removing all vulnerabilities in the network and system. However, whilst this solution is a nice solution in reality it is impossible. Since, vulnerabilities are constantly being discovered. The average rate of patch releasing for those vulnerabilities is not constant and does not keep up with this rapid increase in vulnerability. These patches may provide more bugs and cause more instability into the system and/or network. In addition, some of these patches require rebooting the system, especially for Windows Operating systems and most of the large organizations using this system will be impacted by availability when rebooting the system and network.

2.11.3 IDS Problems

Section 2.10 discussed one of most common security devices that is used in many wired networks, which is the IDS device. However, there are some issues with IDS devices. The main problem is overhead that occurs in IDSs, which can be unsatisfactorily high. The IDS will analyse logs of the system and this requires the system to hold data that relates to all operations and actions performed. This is likely to result in large amounts of information requiring more CPU resources and larger disk space. Next, the system logs must be converted into a format that can be managed by processing in order to compare with the set of attack patterns to determine and identify the possibility of security breaches and violations. In addition, human expertise must be involved to keep continually updating these stored patterns. The main goal is to use tools that are intelligent, adaptable and cost-effective with the capability of real time intrusion detection (Mukkamala et al., 2002).

2.11.4 Lack of Jurisdiction

In the information era, the attack activities have no borders. As a result, in some cases it is hard to bring an intruder to court because of jurisdiction legal issues. Therefore, the lack of jurisdiction between countries has increased the number of international attacks. Rahman (2012) noted that there has been a debate about cyber-crime processes from different countries that are reluctant to approve the Convention on Cybercrime Twelfth United Nations Congress on Crime prevention and Criminal Justice that was held during April 2010. In order to prevent multiple laws on computer related crimes, there has to be clear legislation that should be applied for all countries. This issue is yet to be resolved, so it is clear that jurisdictional issues is one of the challenges that digital forensic investigators are facing.

The other challenges that may occur in cyber-crime is a conflict of national jurisdictions. For example, the attacker causes harm to the organizations' network in the same country. Bryant and Bryant (2014) mentioned that in certain digital crimes there is problems concerning both material jurisdiction and procedural jurisdiction. In material jurisdiction, an attacker will be prosecuted "under which jurisdiction was the offence committed?". However, procedural jurisdiction means an attacker will be prosecuting "the procedural rules of which jurisdiction govern the investigation and evidence".

2.12 CONCLUSION

Overall, the literature review shows that there are many issues within Network Intrusion Detection Systems. It can be summarized in three main sections. These are the security, the hacking activities and the network forensics. The first section discussed and reviewed the network security in general and its vulnerabilities and threats. The second section discussed different types of network attacks and the most common attacks that target wired networks. While the digital forensic section has reviewed, the investigation life cycle and forensic tools. Finally, Intrusion Detection Systems and issues for future research were discussed in sections four and five respectively.

The following chapter will discuss the main research question, sub-questions and the proposed system design of the research.

Chapter 3
Research Methodology

3.0 INTRODUCTION

Chapter 2 reviewed a wide range of literature in network security, network vulnerabilities, threats, and common attacks in a network environment. The literature reviewed in Chapter 2 also includes network forensics, network investigation lifecycle, and Intrusion Detection Systems (IDS). Many current challenges encountered in the field of network forensic investigation have been discussed in order to derive a research question. In Chapter 3, the main research question and sub-questions are presented. The objective of this Chapter is to develop a methodology that is suitable for this research. The research methodology is selected based on the issues that have been previously reviewed. Research in digital forensics can be conducted using different methods, such as a controlled lab environment, simulation of data using virtual machines, and public network environments. The research reported in this project has been conducted using an experimental approach, and using a controlled lab environment, which is vital in order to ensure the validity, integrity, and fairness of the research results, and to preserve admissibility of the collected evidence. Kothari (2004) stated that success in the experimental research approach is achieved when the researcher creates "an experimental design that will manipulate the materials concerned in order to bring forth the desired information" (p.4). This approach assists with making deductions based on observation and experiment.

The structure of Chapter 3 consists of five Sections. Section 3.1 presents the research questions, and sub-questions. The research design is outlined in Section 3.2, and the system design, architecture, and devices are outlined. The section also presents the software and hardware specifications. Section 3.3 discusses data requirements, which consist of data preparation, generation, collection, analysis, presentation and data reporting. The limitations of the proposed research are outlined in Section 3.4, and then Section 3.5 will conclude the chapter.

3.1 THE RESEARCH QUESTION

A review of literature in Chapter 2 discussed the use of IDS devices in LAN networks, and the issues that can be encountered when using IDS as a reliable source of collecting forensic evidence. Section 2.11 highlighted some issues that correlate with the acquisition and preservation of LAN network traffic that can be used as a reliable source of forensic evidence. The main research question is developed based on the reviewed literature in the previous chapter, along with the highlighted issues in Section 2.11. The main research question that is derived from this review is:

> ***Do the three selected IDSs working together detect more attacks than any one single IDS?***

In order to support the main question, a set of sub-questions is developed, which will be answered in Chapter 5.

Sub-question 1*:* Of the three tested IDSs, which one performs most effectively at detecting network intrusions?

Sub-question 2: What percentage of the network attacks are detected?

Sub-question 3: Can the selected IDSs identify a repeated attack as the same attack?

Sub-question 4*:* Which of the three IDSs results in the most forensically sound evidence from a given attack?

Sub-question 5: Which of the three IDSs working alone performs best in identifying an attack?

Sub-question 6: If extra functionality from a third party vender is added to the IDSs, does this change the recommendation?

In order to answer these questions, a research design is developed and discussed in Section 3.2. Section 3.3 outlines the system architecture and components.

3.2 THE RESEARCH DESIGN

In order to answer the research questions, an experimental research design is developed to collect evidence from IDSs on LAN networks in a forensically sound manner. This research was based on gathering evidence from the LAN network traffic by acquisition and preservation of digital evidence. In order to conduct this experiment, the researcher created a network design to address the design requirements that are within the LAN network.

The research design has been developed based on a model developed by Pilli et al. (2010). This model was discussed in Chapter 2, in Section 2.8. The purpose of adopting this model was to ensure that the experiment is conducted in a forensic manner, which ensures the validity of the research examination and results. This assisted the researcher to manage the process of collection, analysis and preservation of the traffic from the LAN networks in an efficient and sound manner. The adoption of this forensic model presented an entire image of the LAN network and guided the researcher to gather the network traffic from the devices of the network.

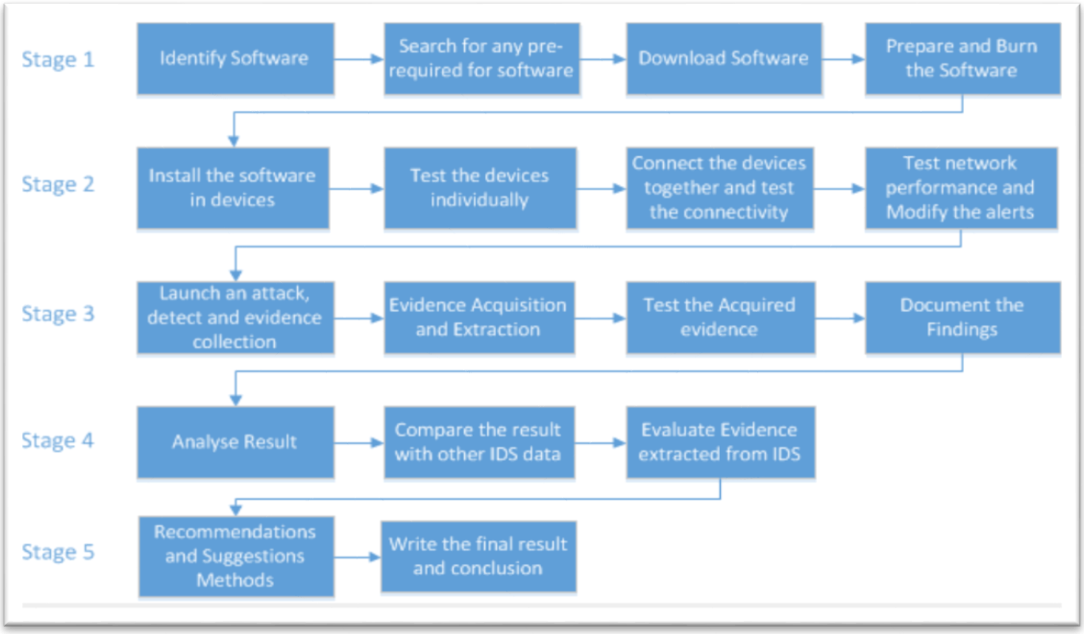

Stage 1
Identify Software
Search for any pre-required for software
Download Software
Prepare and Burn the Software
Stage 2
Install the software in devices
Test the devices individually
Connect the devices together and test the connectivity
Test network performance and Modify the alerts
Stage 3
Launch an attack, detect and evidence collection
Evidence Acquisition and Extraction
Test the Acquired evidence
Document the Findings
Stage 4
Analyse Result
Compare the result with other IDS data
Evaluate Evidence extracted from IDS
Stage 5
Recommendations and Suggestions Methods
Write the final result and conclusion

Figure 3.1: Stages of Experimental Design (adopted from Pilli et al. (2010))

This research contains five stages as shown in Figure 3.1. In stage one, there are several steps that have taken place in order to start the experiment. This is preparation stage. There were several devices, different tools or software/packages, and different operating systems used in this research. Therefore, the first step in stage one was to identify both the software and the operating system that were used in each network and system device. This included tasks such as identifying the operating system of the server and client nodes as well as Firewall, including any required software that assists those devices/software to run smoothly during the experiment. The type of IDSs used in this research determined the packet analysers and other security tools. Some of these operating systems and software were downloaded from the Internet as open source; and others were sourced locally, such as the hardware components and the operating systems.

The second stage began by installing the operating system of the network devices and other security tools. The operating system of IDSs, security tools, and other operating systems that had been identified in a previous stage were installed on an individual computer. Those devices were placed at different locations in the network based on the research design of the LAN. Each device on the network was tested individually. The following step was to connect all network devices and the connectivity was checked in order to be sure each packet could move from a device to another device as expected. To check the connectivity between the network devices ping commands were used

In this stage, Simple Network Monitoring Protocol (SNMP) was needed to test the network performance. One of the main functions of SNMP is that it enables the network administrators to manage and control the various network components efficiently (Xiangyu, 2011). Consequently, the traffic acquisition of the LAN network can be assured by insuring that the functionality of each device in LAN network is functioning and communicating correctly.

At this stage, the firewall and IDSs were tested. All alerts were validated in order to reduce, identify and categorise the false alerts. This technique assisted the researcher, to focus on the genuine alerts in order to discover any attacks, threats or any other network security breaches.

The third stage was begun by launching the attack against the target machines. In this step, the detection of attacks and collection of evidence of the attack were included. However, in the detection step the firewall and IDSs generate alerts about any security breaks into the researcher in this research. This step according to Pilli et al. (2010) model that was adopted is divided into two steps: collection and incident response.

In the collection step, all the network traffic data was collected from the chosen IDSs and their agents. The incident response to a breach that has been detected in the system is established by the data gathered and would be used to authenticate the system and measure the event.

Casey (2004) highlighted that the process of collecting evidence must be done by using a proper software as well as hardware. Casey and Stanley (2004) discussed the impact that might occur when extracting evidence with improper digital tools. The necessity to verify and validate the digital tools has been discussed in the study of Guo, Slay, & Beckett (2009). A discussion has been made of a number of tools including free and proprietary tools (Rod & KPMG, 2002)

To complete the collection step, minor modifications were conducted in this phase by installing and integrating a new sole computer. This machine will store all alerts of all IDSs in different database tables, which was called a forensic server as shown in Figure 3.2. This forensic node was used as a storage centre for all information of traffic and alerts that might be needed later and for further investigation. This forensic server contained all software and hardware that were required to collect all IDSs alerts and traffic captured information. Most of the traffic data was saved as a Pcap format, which is mostly used in network analyser tools, such as Wireshark. This research had many IDSs and most of these IDSs used different approaches and algorithms for capturing the malicious activities and traffic. Thus, these differences between IDSs made capturing the required data complex. The data stored in a forensic server is compared by using the technique of data mining. This technique assisted in the evaluation of the accuracy and capabilities of the captured data.

Once the evidence was collected, then the preservation of evidence proceeded. This step is done by handling carefully all the data gathered from all IDSs. In addition, a hash of evidence data is performed to ensure integrity of the data. This phase had an important step, which was copying the original files and then analysing the copy of the evidence rather than analysed the original. This action has achieved compliance with the standard of legal investigation processes. Copying evidence will avoid any problems that might affect the integrity of the original evidence.

Casey and Stanley (2004) mention that accessing network devices to collect evidence with network forensic tools can create a risk of losing or destroying the evidence by changing or modifying the integrity of the evidence. Therefore, the investigators should have the time and skills to evaluate these tools before use. This reduces the risk. Rod and KPMG (2002) conducted their study and showed that these network forensic tools are designed mainly to perform specific functions whereas others can be multi-functional. Open source and commercial software are referenced to determine which of these tools can be used for a digital forensic investigation. While, Guo et al. (2009) reported that the law enforcement and other agencies require the investigators to verify and validate the tools in order to ensure that the consistency and the integrity of the digital evidence is not altered. They identified the validation as “the confirmation by examination and the provision of objective evidence that a tool, technique or procedure functions correctly and as intended”. Whilst verification means “the confirmation of a validation with laboratory tools, techniques and procedures” (p.S13). This takes into consideration the need of standardization and the requirements to be achieved in order to be accepted in a court of law.

The collected data was investigated forensically by processing gathered information utilizing a combination of both manual and automated techniques. This included extracting and evaluating information of specific interest while protecting the integrity of the information. This will assist to discover the shortcomings in the devices/sensor and demonstrate how they can be enhanced to correct any identified weaknesses in the future.

The study by Casey (2004) mentioned that many organizations underestimate the need for processing and documenting digital evidence. They also are not aware that data can be used as a foundation that can be provided for judgments and drawing conclusions in a certain case. Furthermore, the digital evidence must be protected from illegitimate access and malicious interference.

The fourth stage was begun by analysing the results that gathered from examination step by correlated and classified in order conclude the observation by a rebuild of the present attack patterns that were found in previous step. There were many techniques to match the pattern of the attack, such as soft computing and data mining. Reconstruction of evidence and analysis made for the collected of evidence in order to draw a conclusion about how suspicious activities were committed. Hence, the forensic examination took place by considering the integrity of the acquired evidence. The analysed data were crucial to determine the most efficient IDS in terms of finding the suspicious activities. In addition, they were crucial to draw a conclusion on what IDS should be improved to detect these suspicious activities in the future. Furthermore, this technique assisted the forensic investigators to know how the attacker commits the attack.

The study of Jiang, Tian, and Zhu (2012) emphasised the network forensic processes that the digital evidence can be obtained. This includes the processes of recording and analysing the network events. They believe that the comparison between computer forensics and network forensics is that network forensics is more concentrated on attack analysis and network intrusion. However, the study of Casey (2004) emphasised that different methods of analysing and acquiring of packets that travel via a wired network is used to obtain reliable evidence by capturing network traffic. In addition, the network traffic has many challenges that may affect the integrity of the evidence as it is discussed in the research of Jiang et al., (2012).

In order to complete this step, the forensic server is used to ensure the data gathered from all IDSs were reliable. This means each IDS alerts was compared with the alerts of the other IDSs.

The investigation step is important in order to determine the path of communication between the attacker and the victim. The packets that were already captured by the IDSs are used for attribution tracking of the attack. The investigation phase gained some advantages from several features of the previous phase of the analysis. Thus, the two phases of investigation and the previous phase of analysis determine how the attack was delivered. Recorded packets are utilized for attribution of action of the attack. The hardest portion is to identify the attacker's identity, especially when investigating in network forensics, but the attribution assisted the investigators in this part. This stage provided some comprehensive and valuable information for the prosecution process of the attacker and incident response based on model of Pilli et al. (2010).

The final stage is to present the results based on the observations from experiment in a shape of an understanding or accepting language for law enforcement, other agencies, and courtrooms. In addition, the conclusion will summarize and explain the results from a forensic perspective. These results can be used for legal proceedings against the intruder. In the final stage, the recommendations, and documentations will be written.

3.3 PROPOSED SYSTEM ARCHITECTURE

The modifications in the forensic model in Section 3.2 assisted the researcher to create an initial the experimental LAN network system. This network system was installed and configured in a laboratory environment. The experiment contained several devices as shown in Figure 3.2. The network machines were a firewall, router, IDSs, forensic server, switch and computers with different operating systems. The router was located on the backbone of the network as a gate between the experimental network and the Internet. The firewall was located between the router and the network span port switch. Then, the switch forwarded all packets into the IDS system to sniff and record the traffic that passes through the whole network as well as generating alerts if needed. In addition, the forensic server used to store all alerts that were coming from the IDS system and monitored all computers in the experimental network as well.

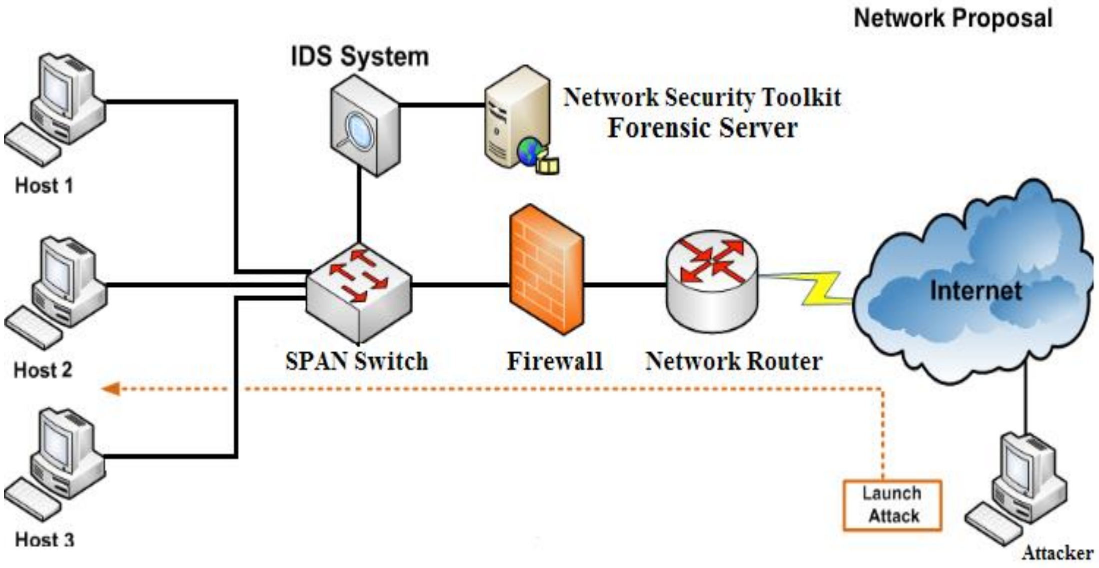


Figure 3.2: The proposed LAN network

The main reason for this outline is to judge the ability of the IDSs to discover or detect simulated attack implemented in this research. The attacks that were used in this experiment were distributed denial of service (DDoS), Reconnaissance, Password crack (Dictionary) and Packet Sniffing. As discussed in Section 2.5, these types of attack were one of the most common recent attacks. Therefore, it is appropriate to test them in the experimental LAN network environment.

The architecture of the proposed system was deployed into a lab environment and included three technical entities as shown in Figure 3.2. The initial components of the infrastructure of the LAN included the forensic server. The second entity was the attacker computer p and any required tools in order to perform the four chosen types of attack. The third component was the main testing element and IDSs system that contains all the log files. After conducting a test attack, a comparative analysis was made on the three IDSs' findings that had captured packets or alerts. These will be discussed in Chapter 4. The following section will identify and explain the required configuration, software specification and hardware required in order to provide the expected packets and log files as well as a robust system and network infrastructure.

3.4 SYSTEM COMPONENTS

The infrastructure of the system was managed to deploy three elements: LAN network configuration, the requirements of IDSs, and forensic server model configuration and the configuration of attacker tools. Most of these components required special software configuration and specific hardware. Thus, the following section will explain each component individually and how to configure them correctly in order to eliminate any misconfiguration.

3.4.1 LAN Network configuration

The design LAN network was configured with software and hardware needed for the experiment. The hardware configuration contained several devices, such as firewall, IDSs, router, forensic servers and a switch. The forensic server contained all the software that was required to accomplish the investigation. The router and firewall were configured with a high level of security settings in order to deny any vulnerabilities or malicious packets that may pass through into the LAN network. The cabling and connection between these devices as discussed in Section 3.2 was the preparation stage. It was very important to configure the routing with the correct ports. The other software, called Salt, has been configured in order to manage all devices in the LAN network. Salt is a configuration management system. The main function of Salt is the ability of maintaining remote devices in well-defined states. Salt is a distributed remote execution system used to perform query information and commands on remote devices, either alone or by arbitrary choice criteria. Salt achieves this through its capability to handle huge loads of data, with hundreds and even thousands of separate servers quickly through a simple and manageable interface (Hosmer, 2012). The next section will explain the configurations of the IDSs that will be used as well as the software and hardware configuration of the forensic server.

3.4.2 IDS and forensic-server model configuration

The tools of the network forensic investigation, security and hardware have been discussed in detail within Chapter 2. Additionally, Chapter 2 and the research design in Section 3.2 proposed and determined the best open source tools and software including: investigation tools, security monitors and IDS that must be installed together in order to gain reliable evidence. At the beginning, each IDS was deployed, tested, and measured in order to evaluate the reliability and accuracy of the evidence. This then leads into each IDS tested by running a pilot test several times, followed by connection into the entire network individually.

This IDS system contained different IDSs and investigation software that assisted the researcher to monitor the system and ensure the network and traffic were running efficiently. The researcher was using several external third party vendor software that installed in a forensic sever in order to assist in finding reliable digital evidence from a compromised network. In addition, this server contained several tools that monitor and detect the LAN network activities. The IDSs software will be introduced as follows:

The first IDS used was OSSEC 2.8.1. It can perform many functions of security such as, checking of integrity, monitor registry of Windows, active response and alerting in real-time. One of the most significant advantages is that OSSEC can run on most common operating systems, for example Linux, Mac OS X, and Windows. In addition, to network intrusion-detection, the OSSEC client has the ability to perform file integrity monitoring and rootkit detection with real-time alerts. All of these are centrally managed with the ability to create different policies, depending on a company's needs. The OSSEC clients run locally on many operating systems, as shown in Figure 3.3. The vendor of OSSEC can also offer commercial support through the Support Team of Trend Micro's Global (Ossec.net, 2015).

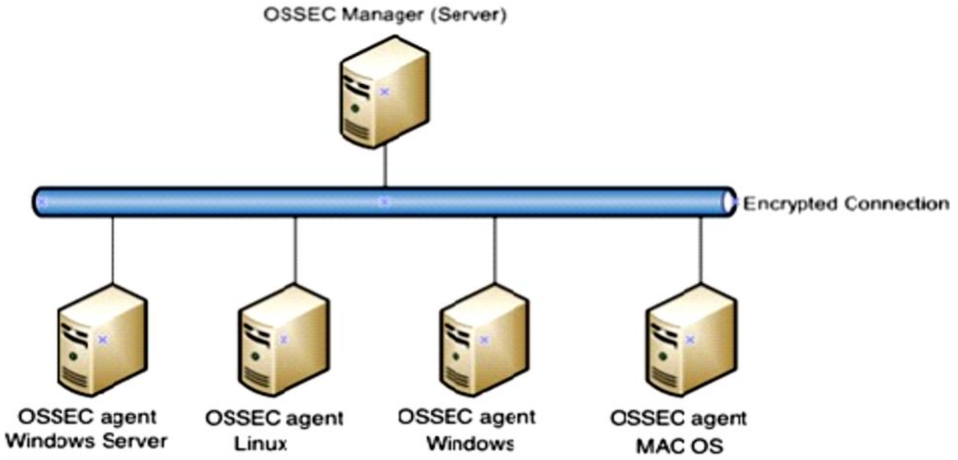


Figure 3.3: OSSEC components

Furthermore, one of the best advantages of OSSEC is that the communication between OSSEC servers and agents is encrypted as shown in Figure 3.3.

Passive Asset Detection System (PADS). PADS was the second IDS installed, deployed and used in this experiment. PADS attempts to identify running programs and applications and monitor the network traffic on the network. The IDS of PADS is a signature-based engine that can discover passively network resources and use IDS techniques to generate alerts. The PADS will never send a packet into the network traffic because its application operates invisibly. Furthermore, PADS has several goals:

- Passive: PADS will listen to the network and record all traffic that passes through the entire network. Thus, there is no packet that will be sent from the application of PADS.
- Portable: PADS has the capability of installing on a remote system easily without requiring any additional libraries except the external libraries associated with libpcap.
- Lightweight: all logging is sent to a CSV file. PADS does not require a data repository or need other databases to be installed on the device. All correlations will be done outside the programming of PADS.

PADS will create and pass through into a FIFO file and the pads-archiver will read from it. Then pads-archiver writes it to one or more information destinations. In addition, version 1.3 of PADS was installed in this research and this version allows MAC addresses to be resolved into hardware seller names (PADS, 2015).

The third IDS installed was Prelude-IDS, which is an open source "Security Information & Event Management" (SIEM) system. Prelude is a global, agentless, and hybrid system, which means Prelude, can be a NIDS, HIDS or both at the same time (Prelude, 2015).

The configuration of hardware was carefully chosen in conformity with the IDS requirements. Every IDS has a structured manual that is available on the Internet that assists to configure them properly. Each IDS configuration file can be seen in the Appendices.

Each IDS system was installed in a Linux distributed system. The fundamental necessities for installing all software of the IDSs was 12GB RAM for the main operating system and 12GB RAM for the main server elements (packages). Furthermore, the capacity of the disk storage was suitable to store all the captured traffic, and this will fill the space of the disk quickly. Thus, to avoid any collision, 1TB capacity of hard disk was utilised.

The configuration of software and packages was based on the IDS requests. All the IDSs were deployed in a single server. Therefore, all of IDSs can be configured, installed and deployed in a Linux operating system as requested by the vendor. The goal for sending all sniffing log files of IDSs and databases that contain all alerts was to have a valid comparative analysis with each other in order to have an effective and accurate evaluation. Consequently, the IDS system device will transmit all databases of all IDSs and the logs file and into forensic server for investigation.

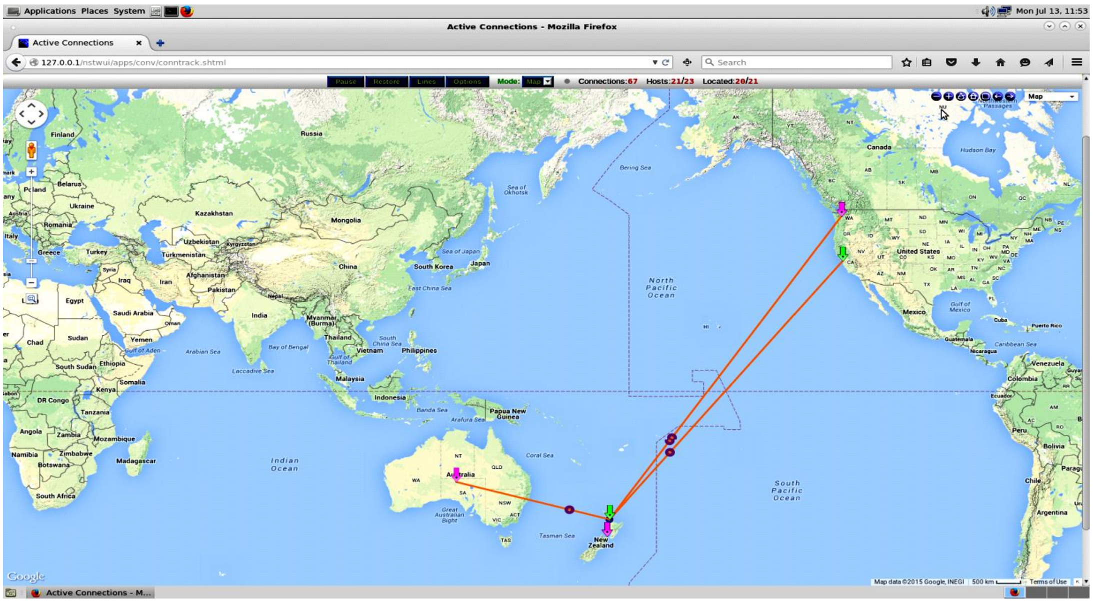

Applications Places System
Mon Jul 13, 11:53
Active Connections - Mozilla Firefox
Active Connections
127.0.0.1/nstwui/apps/conv/conntrack.shtml
Search
Pause
Restore
Lines
Options
Mode: Map
Connections: 67
Hosts: 21/23
Located: 20/21
Map
Google
Map data ©2015 Google, INEGI
500 km
Terms of Use
Active Connections - M...

Figure 3.4: Active Connections

The operating system of the forensic server was Network Security Toolkit (NST). This NST is based on a Linux distribution system. The operating system will assist the network/system administrator to monitor all connections on the entire LAN network. In addition, this system has many features, such as an advanced Web User Interface (WUI) that will help the network/system administrators to monitor the LAN network, host geolocation, automation, navigation, network analysis and configurations of various security and network applications available within the distribution of the NST (NST, 2015). Representing information in this layout allows the operator to consider whether there is an irregular mount of information coming from an unfamiliar destination, which could point to an attack as shown in the Figure 3.4. Without such visualisation being used, the operator is offered with a set of hostnames or IP addresses, which cannot at a look, be connected with a select country. Even though, select IP ranges are assigned to select ISPs in select countries, it is quiet highly likely that the end-user will have learnt these ranges (Hales, Ferguson, & Archibald, 2013).

3.4.3 Attacking tools configuration

The configuration of the attacking tools was the final task in the design of the system. The attacker’s actions will launch a Reconnaissance attack that searches for the open port by open source tools and then sends the malicious code to the victim in order to then get access into a victim’s machine followed by DDoS, password crack, and finally Packet Sniffing attacks. There were many tools available on the Internet that can achieve the chosen attack successfully, but the best open source tools were considered to be the software that was installed in the Linux system. Furthermore, this operating system contains proper tools that can help to accomplish the attacks that were required in this research. It was only needed to know the IP address of the victims' computer to launch these attacks.

3.5 DATA REQUIREMENT

There was a requirement to recognize where information was produced between nodes in the proposed LAN network in order to evaluate the proposed forensic model. All information was gathered then analysed, and will be discussed in Chapter 4. This part was classified into mainly four subsections: preparation, data generation, data collection and data analysis and documentation. Each subsection will be discussed in order to obtain a comprehensive understanding of the design and results.

3.5.1 Preparation

The digital evidence is hidden, concealed, and easily destroyed. One of the most important steps in the evidence information was the preparation. In order to evaluate the selected IDSs, it was critical to prepare what was required before gathering evidence from all IDSs. Original preparation includes preservation of the evidence. It was essential to:

- Have a clear image of the wanted data and intended analysis,
- Include required analysis of validity and reliability of evidence.
- Generate predictable evidence data in the LAN network
- Define the quantity and type of information preparation required before the phase of evidence collection.
- The computer must contain gatherable information for each of the three main IDSs for the purpose of this research.
- Protections are to be used to avoid modification of the data.

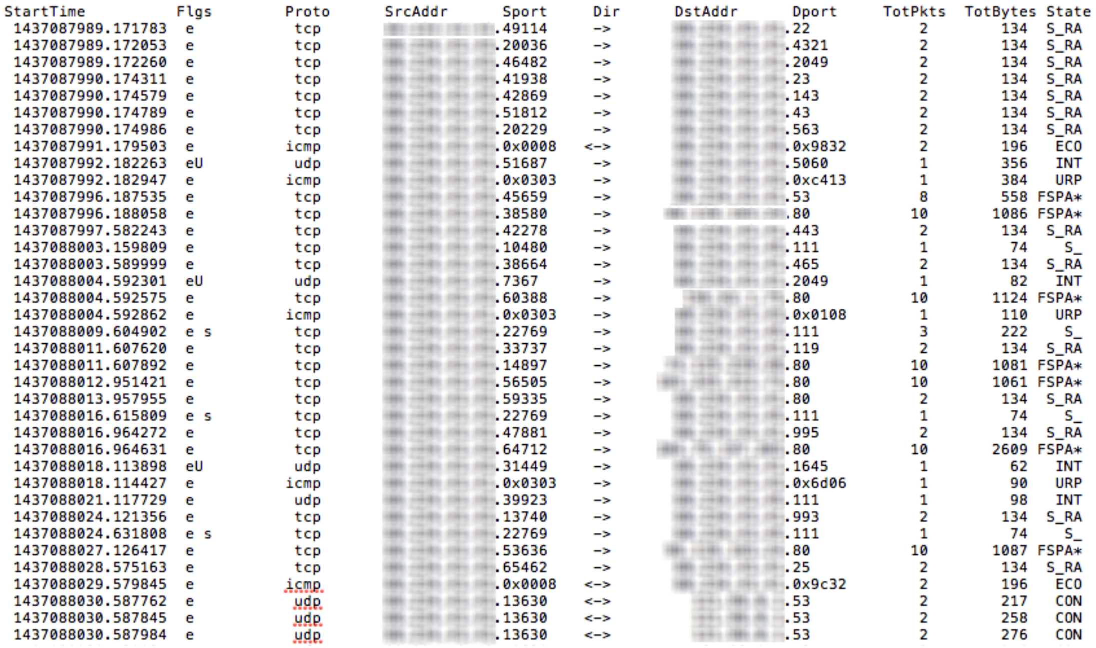
Associated data were considered in all events where disclosure may be needed or additional investigation required. Capturing all LAN traffic data using forensically sound techniques was necessary for the integrity of any investigation. It was necessary to prepare tools or software, such as Wireshark for the analysis and the capture information if needed. In addition, organizing tools for gathering data from digital information, hardware, and software for the examination procedure were prepared. The following software and hardware was prepared for the data generation, data collecting, data examination, and documenting of the digital evidence. All of the above arranged hardware and software was adjusted in accordance with the credentials and recommendations of the creator and the forensically best practices, standards, and procedures.

3.5.2 Data Generation

Data generation is the most crucial phase that will assist and support the research experimental system design. There were several methods for data generation, and in order to gain expectable results, it was carefully structured. The means of generating data in this research was capturing all traffic that moves between all LAN nodes. However, this traffic had different types of formats that communicate with the LAN nodes, such as router, firewall, the forensic server and users. This traffic from the attacker nodes can be malicious code.

The technique of generating data requires the researcher to monitor all traffic in order to confirm that the network devices were efficient and working appropriately. Additionally, SNMP was deployed and installed in order to measure the abilities of the LAN network to communicate between the experimental nodes in section one.

There are many software types available on the Internet as open source for monitoring the LAN network, and one of this software is Argus. It is an observer for flow in real time, utilization system and network Audit Record Generation that is considered to achieve whole data network traffic inspecting as shown in Figure 3.5. One of the most useful things in Argus is that a bi-directional stream modeller and network monitor replies to any network traffic that is sent. Additionally, Argus has several identifiers/descriptors in various flow copies of the OSI starting from Layers 2 through to 5. Argus is presently available in version 3.0.8.1 at the time of this research, and updated frequently since its beginning and supported with a robust supportive community. Argus provides inclusive network forensic inspections, which comprises events of network nonrepudiation. According to Bullard, Argus provides a comprehensive security assurance as it “enables the establishment of a comprehensive audit trail of all network activity, either for a single network element or for an entire network segment” (McRee, 2008, p.7). The main reason to use Argus was to record all traffic that passes through the whole LAN and make sure all IDSs' were facing the same attack techniques. To evaluate all IDSs' the comparison was made between IDSs' results and Argus data. For example, the attacker launches the attack against the network/devices, and one or all of the IDSs failed to show that there is an attack. The Argus data could confirm whether the attack was committed or not.



```
StartTime           Flgs    Proto   SrcAddr           Sport    Dir   DstAddr           Dport     TotPkts  TotBytes State
 1437087989.171783  e        tcp    [redacted].49114    ->   [redacted].22          2       134  S_RA
 1437087989.172053  e        tcp    [redacted].20036    ->   [redacted].4321        2       134  S_RA
 1437087989.172260  e        tcp    [redacted].46482    ->   [redacted].2049        2       134  S_RA
 1437087990.174311  e        tcp    [redacted].41938    ->   [redacted].23          2       134  S_RA
 1437087990.174579  e        tcp    [redacted].42869    ->   [redacted].143         2       134  S_RA
 1437087990.174789  e        tcp    [redacted].51812    ->   [redacted].43          2       134  S_RA
 1437087990.174986  e        tcp    [redacted].20229    ->   [redacted].563         2       134  S_RA
 1437087991.179503  e        icmp   [redacted].0x0008  <->   [redacted].0x9832      2       196   ECO
 1437087992.182263  eU       udp    [redacted].51687    ->   [redacted].5060        1       356   INT
 1437087992.182947  e        icmp   [redacted].0x0303   ->   [redacted].0xc413      1       384   URP
 1437087996.187535  e        tcp    [redacted].45659    ->   [redacted].53          8       558 FSPA*
 1437087996.188058  e        tcp    [redacted].38580    ->   [redacted].80         10      1086 FSPA*
 1437087997.582243  e        tcp    [redacted].42278    ->   [redacted].443         2       134  S_RA
 1437088003.159809  e        tcp    [redacted].10480    ->   [redacted].111         1        74    S_
 1437088003.589999  e        tcp    [redacted].38664    ->   [redacted].465         2       134  S_RA
 1437088004.592301  eU       udp    [redacted].7367     ->   [redacted].2049        1        82   INT
 1437088004.592575  e        tcp    [redacted].60388    ->   [redacted].80         10      1124 FSPA*
 1437088004.592862  e        icmp   [redacted].0x0303   ->   [redacted].0x0108      1       110   URP
 1437088009.604902  e s      tcp    [redacted].22769    ->   [redacted].111         3       222    S_
 1437088011.607620  e        tcp    [redacted].33737    ->   [redacted].119         2       134  S_RA
 1437088011.607892  e        tcp    [redacted].14897    ->   [redacted].80         10      1081 FSPA*
 1437088012.951421  e        tcp    [redacted].56505    ->   [redacted].80         10      1061 FSPA*
 1437088013.957955  e        tcp    [redacted].59335    ->   [redacted].80          2       134  S_RA
 1437088016.615809  e s      tcp    [redacted].22769    ->   [redacted].111         1        74    S_
 1437088016.964272  e        tcp    [redacted].47881    ->   [redacted].995         2       134  S_RA
 1437088016.964631  e        tcp    [redacted].64712    ->   [redacted].80         10      2609 FSPA*
 1437088018.113898  eU       udp    [redacted].31449    ->   [redacted].1645        1        62   INT
 1437088018.114427  e        icmp   [redacted].0x0303   ->   [redacted].0x6d06      1        90   URP
 1437088021.117729  e        udp    [redacted].39923    ->   [redacted].111         1        98   INT
 1437088024.121356  e        tcp    [redacted].13740    ->   [redacted].993         2       134  S_RA
 1437088024.631808  e s      tcp    [redacted].22769    ->   [redacted].111         1        74    S_
 1437088027.126417  e        tcp    [redacted].53636    ->   [redacted].80         10      1087 FSPA*
 1437088028.575163  e        tcp    [redacted].65462    ->   [redacted].25          2       134  S_RA
 1437088029.579845  e        icmp   [redacted].0x0008  <->   [redacted].0x9c32      2       196   ECO
 1437088030.587762  e        udp    [redacted].13630   <->   [redacted].53          2       217   CON
 1437088030.587845  e        udp    [redacted].13630   <->   [redacted].53          2       258   CON
 1437088030.587984  e        udp    [redacted].13630   <->   [redacted].53          2       276   CON
```

Figure 3.5: Capabilities of Argus to monitor the LAN network

In stage two, there was other software installed, configured and deployed in the LAN network that can monitor the network devices. This software was Ntopng. It was the basic or baseline for delivering the main evaluation of the network performance. The installation is shown in Appendix 7.

Through the third stage, attacker actions were conducted against the victims' machine in the experimental design to evaluate the ability of the forensic model in order to gather evidence. The attack tools that were discussed in depth earlier in Section 3.2.2.3 were used as the hacking tools. Nevertheless, to correctly evaluate the ability of each IDS, these attacker tools were used against the three IDSs at the same time, to be fair when evaluating all IDSs. By these methods the abilities of each IDS is evaluated.

3.5.3 Data Collection

One of the most important parts was data collection because it was the starting point that determines the ideal ability of the examined IDSs. The various log files are collected from traffic that has been generated from different tools discussed and described in the previous section 3.3.1. These gathered log files are maintained and supported by the forensic model, which acquires and preserves the traffic of the LAN network transferred between all nodes on the network. In addition, the main collection of data was from a forensic server machine. The goal of gathering data traffic from another machine was to accomplish and triangulate information from the source and for evaluation.

The second stage depended on two means of data collection; multiple IDSs log and Argus record files. The first core means of gathering the data is the IDS alerts database and logs in order to approve the traffic of the network that is generated. While the second method of data gathering is the multiple files of records generated by the proposal forensics model in the forensic server that hosts all IDS'. In addition, different databases will be installed in the forensic server device. This is to ensure that all log files are centrally acquired within the database and prevent data loss. Each IDS is connected to a different database table, which will act as forensic evidence storage, and provide comprehensive data about any security breaches which will be typically alerts. These databases are the core sources of information gathered through the examining stage. Consequently, the core idea of this stage is to gather log data and databases from the examined IDSs and then send all of these files to the forensic server where the analysis tools start by examining and presenting the findings about these databases and logs. This will assist in having the correct IDSs log files, which will be matched to the network traffic captured by Argus. This is to approve the traffic of the whole network that is acquired.

The third stage depends on the actions of attacker that were performed against the victims' machine. This was done by installing several open source tools that were discussed in Chapter 2 section 2.2.2.3. The attacker actions consist of four ways of attacking the network. The first attack will be a Reconnaissance attack then DDoS followed by the Dictionary attack and the last attack is Packet Sniffing. All attack attempts will be collected by different IDSs. The results of the attacker actions of all attack methods will be gathered and then compared with captured packets of other IDS. This means all captured packets of IDSs will be compared simultaneously in order to evaluate the acquisitioning abilities of each IDS. Finally, this technique will guarantee that each IDS has the same test as the other IDS. In addition, this technique is effective as well as time consuming.

3.5.4 Data Analysis, Presentation and Reporting

The data that has been captured and collected from the proposed forensic models will be analysed forensically including all Pcap files that contain all logs from each IDS. There are many methods detailed and discussed previously in Section 3.2 in analysing such information as well as the forensic analysis tools in Section 2.9 of Chapter 2. There are several tools that will be used in this experiment in order to assist the researcher to analyse the results. These analysis tools will be installed in a forensic server as follows:

Firstly, Enterprise Log Search and Archive (ELSA) will be installed and deployed in the forensic server. ELSA is a centralized system of a logs framework constructed on several software and tools including: MySQL, Syslog-NG, and Sphinx with full text search capability. With the support of these open source tools, ELSA can present evidence or data in user-friendly interfaces. ELSA has the ability to normalize logs, which makes it easier to search within huge amounts of strings (Holste, 2015). ELSA can be used in different security operating systems, such as Security Onion. Furthermore, the Enterprise Log Archive and interface is a solution to achieve the following:

- Store, normalize, and index logs at infinite rates and volumes.
- Offer a simple search API and interface.
- Offer an infrastructure for reporting, sharing and alerting logs
- Available as an open-source and totally free project
- The system plugin is available to take actions with logs

Secondly, the application of Squert was installed and deployed in the forensic server. Squert is a web application that is used mainly to view and query event information that has been stored in Sguil database, which is specifically IDS alert data. One of the most useful features in Squert is that it maximises to deliver additional context to events data via the use of metadata, time sequences weighted and representations and logically assembled result sets in a visual view (Squert, 2015). The installation is attached in the Appendix 9 Squert Installation.

The third tool used to analyse and present the OSSEC IDS alerts was Splunk. It is a platform for analysing machine statistics and information that machines emit in powerful volumes, but which is infrequently used effectively. It has an ability to export the alerts in different formats and present OSSEC alerts in different shapes or styles (Hanley & Montelibano, 2011).

All the data (log files and records of Argus) that has been examined will be reported and presented in order to analyse the collected findings from all selected IDS. The result will be presented in a visual manner by using different graphing mechanisms for a clear presentation.

3.6 LIMITATION OF RESEARCH DESIGN

The research design and methodology has several limitations that are identified with the purpose of avoiding any impediments that may lead to a failure in the experiment. Likewise, it was critical to identify all restrictions to appropriately evaluating the results obtained and to describe whether there is a necessity to qualify the findings.

Firstly, the location of some devices in LAN networks, such as the router and firewall in the proposed network design. There have been debates about the location of IDS in the network by some researchers and security administrators (Fosic & Zagar, 2011). If the IDS is located at the edge of the network and before the firewall, then this means that the IDS will monitor all packets that will include the firewall traffic and there are possibilities of stopping the firewall traffic. As a result, this will increase the number of alerts, which may confuse the network administrators by not realizing there are serious threats, and being confused with the minor threats. However, locating the IDS after the firewall will decrease the number of alerts and the IDS will be able to monitor the traffic that passes the firewalls. Thus, the traffic that attempts to attack the firewall will not be captured and these packets cannot be analysed.

The second limitation was that there are many types of IDSs are used to protect the network, such as NIDS and HIDS as discussed in Chapter 2, Section 2.10. The IDS depends on various domains of network measures. IDSs will inspect all network traffic by using two general measures of low granularity and high granularity (Eiland et al., 2008). This experiment has a limited budget and will examine only open source IDS. Thus, this limitation makes the determination of IDS restricted to open source IDSs.

According to Jiang et al. (2012), there are some challenges that might face the network investigators. One of these challenges is the amounts of data that must be collected and analysed and be ensure there is no a lack in process of the evidence that is collected from networks, and others are minor difficulties. However, Casey (2004) believes that there is another main challenge in network forensics that can be faced with the network investigators, which is that unreliable digital evidence being collected can be misleading and may lead to poor choices and incorrect conclusions that can be a reason for more harm than the incident itself. This leads to the installation of some external software in order to increase the concentration on the reliable evidence.

The fourth limitation is that there were many ways to design and configure the architecture of the network system. In addition, these alterations depend on the employment, capabilities, and the operations required for a network. While, the practical alterations make standardized collection of data difficult. These applied variances have a means of producing communication between compatible nodes, but they make it more arduous to standardise the collection of data.

Another limitation was the proposed methodology for testing in the management of computers when deploying IDS's. This means each IDS was tested many times in order to confirm this IDS is operating, and interacting properly. When the installation was finished and IDS tested, the backup could be the best way in order to reduce the time of installation of the IDS another time, if necessary.

The final limitation was the difficulties of applying all types of attacks that had been chosen in this research to be targeted or launched against the victim's machine. For example, there are many methods to accomplish the password attacks. There are several attacking tools and methods have the capability to accomplish their job such attacks, with the aim of compromising computers or networks. Thus, according to the time constraints and research required, this research was restricted to chosen types of attacks, tools, and methods.

3.7 CONCLUSION

This chapter has discussed the methodology of the research and design. Each feature of the selected area is reviewed for the research starting from gathering, acquiring and preserving evidence from all selected IDSs in LAN networks. The literature review in Chapter 2 and Section 3.1 in combination, has led to constructing the main question of the research. Additionally, Chapter 3 highlights the research question and the structure of the system to cover the proposed architecture and its components. A descriptive technique was implemented to move the design of the research into existence and determine the main software and hardware.

In the following chapter, the results of the experiments and findings will be presented in a visual manner as well as a comparative analysis.

Chapter 4

Research Findings

4.0 INTRODUCTION

The primary objective of Chapter 4 is to report the research findings obtained during the experiment stage. The findings presented in this chapter are based on the selected network attacks discussed in Chapter 3. The outcomes produced from different attack experiments are independent from one another. In addition, each experiment used different tools and methods. The results are grouped based on the IDS into the presentation, analysis, and reporting. This approach assists in evaluating the design of the system and concludes with the abilities of each selected IDS in collecting digital evidence. The results from Chapter 4 produce insights into the capabilities of the selected IDSs in producing digital evidence in a forensic manner.

However, it is also necessary to highlight the main steps of installation for all IDSs and analysis tools that have been used in this research. These software and tools have different approaches for installation and would often lead to different results.

Chapter 4 is divided into five sections. The first section will discuss the configurations and testing of each device used in the design system in order to ensure that all devices are working as expected. Then, this section will include some of the required configurations of the attack device that was selected. The following section will describe the main steps of installation of each chosen IDS and additional tools that are needed for analysis. Section 4.3 will discuss the attacks performed against the system. Finally, Section 4.4 will present a comparative analysis and evaluation of the results and this is followed by a brief conclusion

4.1 PREPARATION OF LAN COMPONENTS

As shown in Chapter 3, Figure 3.2, the system design contained several devices connected to a LAN. All devices in the design were configured with a static IP in order to target the specific machine when launching the attack phase. Additionally, the design had different devices: two servers, router, switch, firewall, IDS device, and end hosts as described in Section 3.3. One of these servers was Windows Server 2012 and the other was Fedora version 23. Fedora was used to back up the IDSs databases and monitor all network devices, which were running and connected to the LAN. The monitoring software used was Ntopng version 2.3.151213. Ntopng required other software to be installed and worked as a database in the back-end. Redis Server software was installed, in order to store all data. The switch was configured to work as a span port switch, in order to enable it to send a copy of all network packets seen on the network into a specific port, thus enabling all the selected IDS to analyse all the incoming packets. The IDS device contained all selected IDSs in this research and their databases. Therefore, each IDS had a separate database with different tables but shared the main software of the database. The operating system of the IDS device was Red hat version 6.7. Windows 7 and Ubuntu were configured on an end host.

4.2 EXPERIMENTAL SETUP

In order to collect data from all IDSs, there were several steps required to be performed first to ensure all components of the system would perform as expected. All IDSs in this research were downloaded and installed in a computer (IDS device) that had Red hat installed as the operating system. The reason for choosing Red hat is that Prelude has been implemented and examined in different Linux and other operating systems and encountered some technical issues that affect its performance and detecting function. Enquiries with the Prelude developers found that Prelude is running and has full support in Red hat environment only, when I emailed them (Prelude Team, personal communication, November 16, 2015). Thus, based on their comments, Prelude was installed in Red Hat, and these issues were resolved and the function of detecting attack activities improved.

4.2.1 Installation and Configuration of IDS

The installation of three chosen IDSs and other third party tools used in this research are highlighted in this section. Only the main points of how to complete the installation of all IDSs, packages and analysis tools are discussed in this section. These installations are based on the requirements of this research to conduct the experiment. All installation steps of IDSs and all processes of installations of third party tools that were used for analysis are attached in the Appendices. The reason for highlighting these installations is that different packages, software, and tools might lead to different results.

The first IDS installed in the IDS device was OSSEC. In the installation of OSSEC, there were many additional choices of installations after downloading the file from the Internet, whether choosing it to be an agent, a server, or a hybrid (which means the machine will be performing as the server as well as the agent at the same time). In the IDS device, OSSEC was a hybrid while in other devices of the design system OSSEC was an agent. The main reason for choosing the hybrid installation in the IDS device was to receive all alerts from other devices and itself; and then store them in a database.

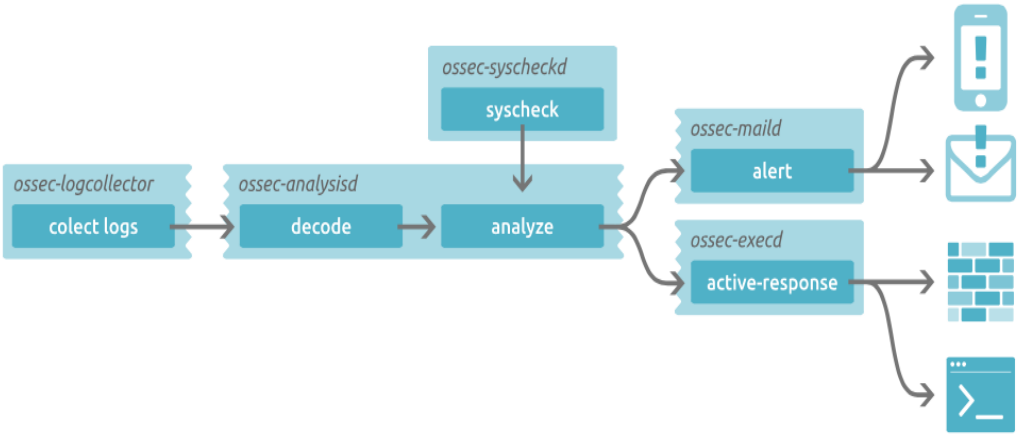
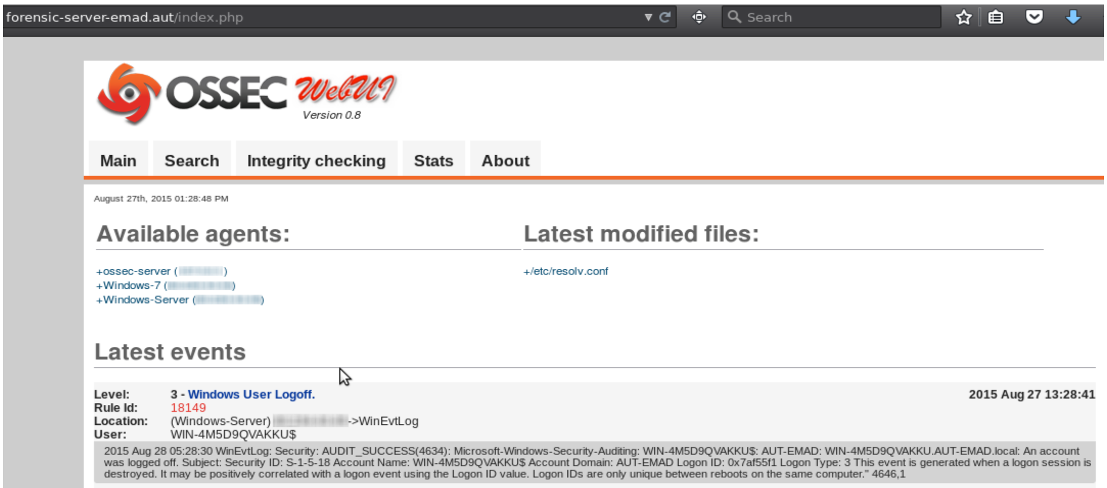


Figure 4.1: OSSEC fundamentals

Figure 4.1 illustrates how OSSEC analyses the incoming packets with any incoming attacks after installation as well as the ways to notify the network or system administrators. In this research, the researcher used two notifications methods: email and user interface notification. The web user interface (UI) of OSSEC can present all alerts, including which agent is active or inactive, and has many other advantages as shown in Figure 4.2. This UI is easy to install and available on the OSSEC website and the steps of installation are attached in Appendix 1. There are many steps that must be done in the configuration file before launching the selected attacks such as, setting the period of alert time (the default time was 79200 seconds, which is an unacceptable time to detect and alert an attack into network or system administrations). This setting will also delay the notifications and emails about any network breaches.



August 27th, 2015 01:28:48 PM

Available agents:

+ossec-server ([redacted])
+Windows-7 ([redacted])
+Windows-Server ([redacted])

Latest modified files:

+/etc/resolv.conf

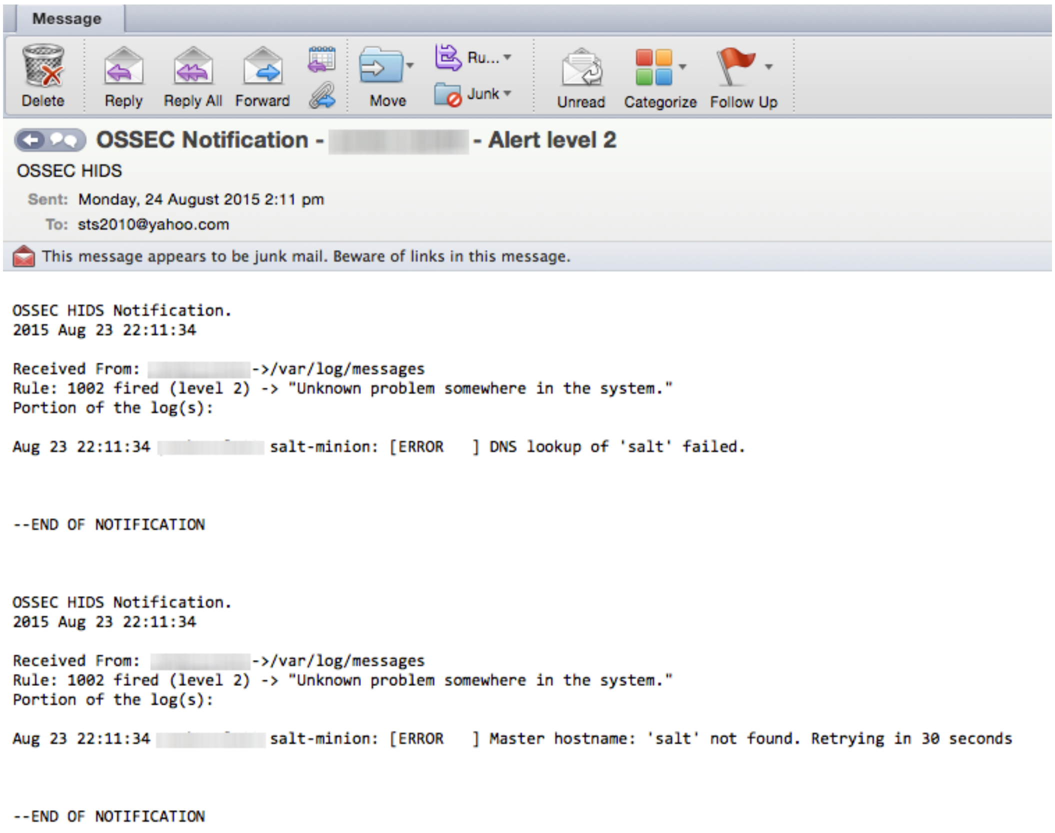
Latest events

Level: 3 - Windows User Logoff. 2015 Aug 27 13:28:41
Rule Id: 18149
Location: (Windows-Server) [redacted]->WinEvtLog
User: WIN-4M5D9QVAKKU$

```
2015 Aug 28 05:28:30 WinEvtLog: Security: AUDIT_SUCCESS(4634): Microsoft-Windows-Security-Auditing: WIN-4M5D9QVAKKU$: AUT-EMAD: WIN-4M5D9QVAKKU.AUT-EMAD.local: An account was logged off. Subject: Security ID: S-1-5-18 Account Name: WIN-4M5D9QVAKKU$ Account Domain: AUT-EMAD Logon ID: 0x7af55f1 Logon Type: 3 This event is generated when a logon session is destroyed. It may be positively correlated with a logon event using the Logon ID value. Logon IDs are only unique between reboots on the same computer." 4646,1
```

Figure 4.2: OSSEC GUI

Another configuration is the mailing settings, so the network/system administration(s) can add as many emails as needed. This feature allows the OSSEC to send an email to whoever is concerned about the security of the LAN network based on the level of the activities as shown in Figure 4.3. These levels are available in the OSSEC website for more information.



OSSEC Notification - [redacted] - Alert level 2

OSSEC HIDS

Sent: Monday, 24 August 2015 2:11 pm

To: sts2010@yahoo.com

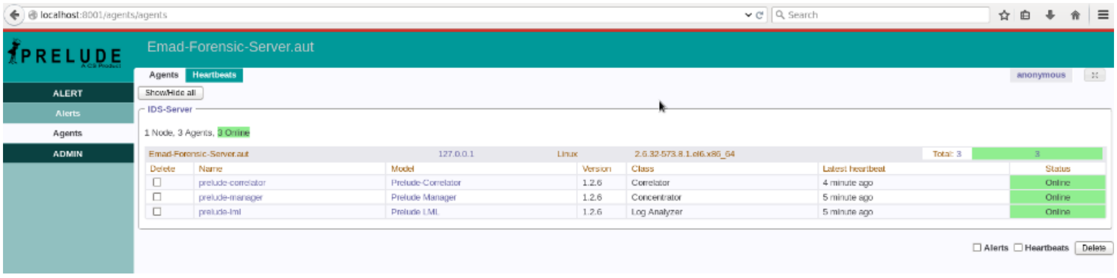
This message appears to be junk mail. Beware of links in this message.

```
OSSEC HIDS Notification.
2015 Aug 23 22:11:34

Received From: [redacted]->/var/log/messages
Rule: 1002 fired (level 2) -> "Unknown problem somewhere in the system."
Portion of the log(s):

Aug 23 22:11:34 [redacted] salt-minion: [ERROR   ] DNS lookup of 'salt' failed.


--END OF NOTIFICATION


OSSEC HIDS Notification.
2015 Aug 23 22:11:34

Received From: [redacted]->/var/log/messages
Rule: 1002 fired (level 2) -> "Unknown problem somewhere in the system."
Portion of the log(s):

Aug 23 22:11:34 [redacted] salt-minion: [ERROR   ] Master hostname: 'salt' not found. Retrying in 30 seconds


--END OF NOTIFICATION
```

Figure 4.3: OSSEC emails

All steps of the OSSEC installation as a hybrid (server and agent) are attached in Appendix 1. While the installations of OSSEC agents in Windows (Windows 7 and Server) are available in Appendix 2.

The second installation of an IDS was Prelude. Three agents are available with the installation as a default with the Prelude installation. These agents are Prelude-manager, Prelude-lml, and Prelude-correlate as shown in Figure 4.4. There are not many configurations in Prelude, except some TLS options in the configuration file of both Prelude-manager and Prelude-lml. The installation steps and required packages are presented in Appendix 3. In addition, Prelude can accept as many agents as possible to be registered. Therefore, different agents might lead to different results. These registration and installations steps are shown in Appendix 3.



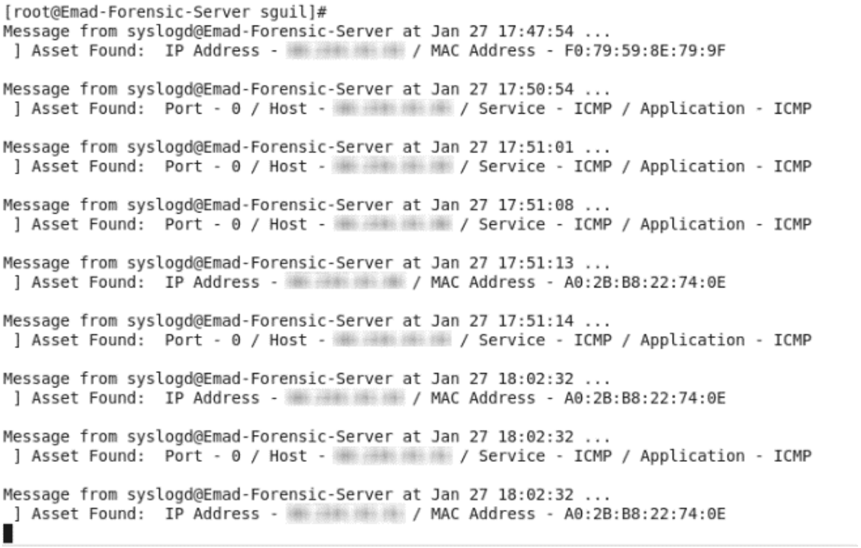
1 Node, 3 Agents, 3 Online

Emad-Forensic-Server.aut 127.0.0.1 Linux 2.6.32-573.8.1.el6.x86_64 Total: 3 3

Delete	Name	Model	Version	Class	Latest heartbeat	Status
□	prelude-correlator	Prelude-Correlator	1.2.6	Correlator	4 minute ago	Online
□	prelude-manager	Prelude Manager	1.2.6	Concentrator	5 minute ago	Online
□	prelude-lml	Prelude LML	1.2.6	Log Analyzer	5 minute ago	Online

Figure 4.4: Prelude agents

The third IDS was PADS. Chapter 3 discussed features and advantages of PADS as shown in Figure 4.5. The installation of PADS requires some software and packages in order to perform and obtain all the accepted forensic results. This software, with packages and installations, are shown in Appendix 4.



```
[root@Emad-Forensic-Server sguil]#
Message from syslogd@Emad-Forensic-Server at Jan 27 17:47:54 ...
 ] Asset Found:  IP Address - [redacted] / MAC Address - F0:79:59:8E:79:9F

Message from syslogd@Emad-Forensic-Server at Jan 27 17:50:54 ...
 ] Asset Found:  Port - 0 / Host - [redacted] / Service - ICMP / Application - ICMP

Message from syslogd@Emad-Forensic-Server at Jan 27 17:51:01 ...
 ] Asset Found:  Port - 0 / Host - [redacted] / Service - ICMP / Application - ICMP

Message from syslogd@Emad-Forensic-Server at Jan 27 17:51:08 ...
 ] Asset Found:  Port - 0 / Host - [redacted] / Service - ICMP / Application - ICMP

Message from syslogd@Emad-Forensic-Server at Jan 27 17:51:13 ...
 ] Asset Found:  IP Address - [redacted] / MAC Address - A0:2B:B8:22:74:0E

Message from syslogd@Emad-Forensic-Server at Jan 27 17:51:14 ...
 ] Asset Found:  Port - 0 / Host - [redacted] / Service - ICMP / Application - ICMP

Message from syslogd@Emad-Forensic-Server at Jan 27 18:02:32 ...
 ] Asset Found:  IP Address - [redacted] / MAC Address - A0:2B:B8:22:74:0E

Message from syslogd@Emad-Forensic-Server at Jan 27 18:02:32 ...
 ] Asset Found:  Port - 0 / Host - [redacted] / Service - ICMP / Application - ICMP

Message from syslogd@Emad-Forensic-Server at Jan 27 18:02:32 ...
 ] Asset Found:  IP Address - [redacted] / MAC Address - A0:2B:B8:22:74:0E
█
```

Figure 4.5: PADS Alerts

4.2.2 Analysis Software

In order to investigate IDSs alerts, it is necessary to install various software and packages to run on the forensic server. Some of these software and packages will be linked to all of the selected IDSs databases in order to present the data/evidence in a way that can be analysed with the findings and compare them with the other data or evidence that is generated from the other IDSs. The additional software and packages include Splunk, Prewikka, Sguil, and Squert.

4.2.3 Configuration of the Attack Device

Before starting the attack phase that was selected and discussed in Chapter 3, the attacker device should be prepared with suitable software and tools. This preparation will assist the researcher to launch the four chosen attacks. One of these has the ability to create username and password lists. These two lists should have predictable usernames and passwords. For example, in the username list, it should have usernames such as admin or administrator and so on.

During the Reconnaissance attack, some software saves the results into the database, whether it is a MySQL database or other open source databases based on the configuration of the software. Thus, the databases should be configured, updated and running well in order to reduce failure that might occur during the attack phase or deny the software from working.

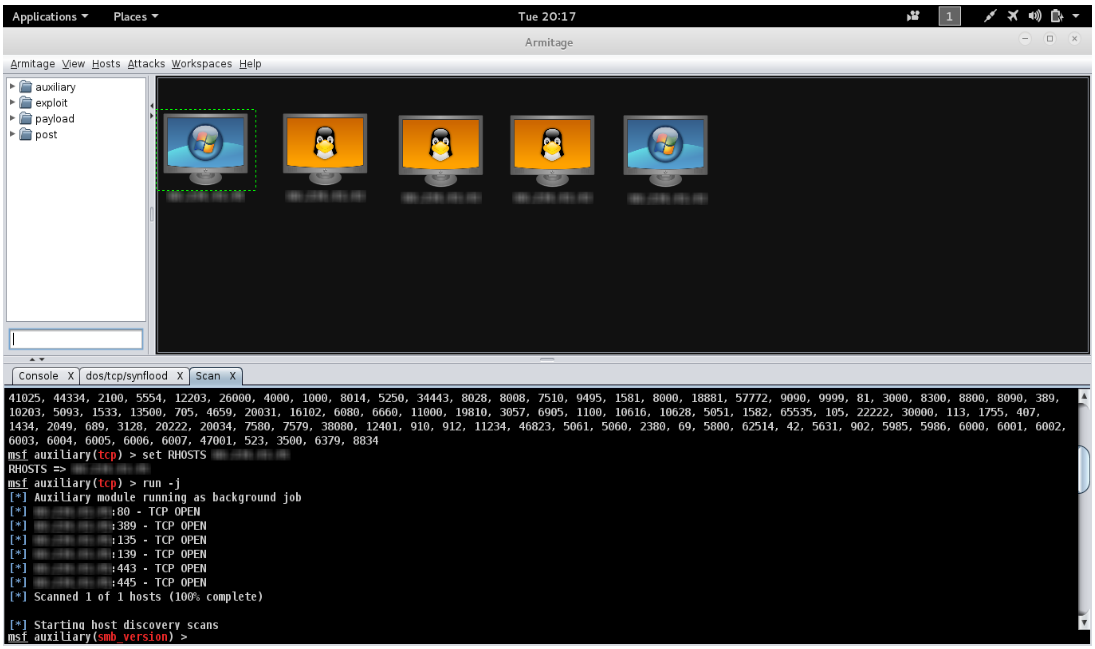
4.3 DATA GENERATION AND COLLECTION

After the components of the experimental system design were connected, the three chosen IDSs were deployed and additional software was also installed and analysing tools were inaugurated becoming ready. The method of generating data in this research is to perform all the attacks at once. This method will ensure that all the selected IDSs are facing the same attack approaches, as well as the same number of packets received. As a result, the evaluation of these IDSs will be more accurate and efficient.

Data generation and collection were triggered, the generation of data in this research was based on four selected attacks that were discussed in Chapter 3. These attacks were Reconnaissance, DDoS followed by Dictionary then lastly the Packet sniffer attack. After that, the collection of the evidence is conducted based on the observation and then the analysing of all alerts from IDSs in order to discover who did these attacks and how, which will be considered as network forensic evidence. These examinations and analysis will assist in the final evaluation of the abilities of each IDS that has in detecting these attacks.

4.3.1 Performing the Attacks

The objective of performing the attacks at this stage is to ensure that the design system has been compromised in order to proceed with the experimental investigation. Once the attacks are successful, the examination of IDSs ability to detect the malicious activities will be conducted. As mentioned previously in Chapter 2, the first attack was a Reconnaissance attack (scan port) in order to discover the information about the design system devices. In this research, the Linux tool was used to launch this attack as shown in Figure 4.6. The information from first attack was used in this research as basic knowledge to perform other selected attacks.

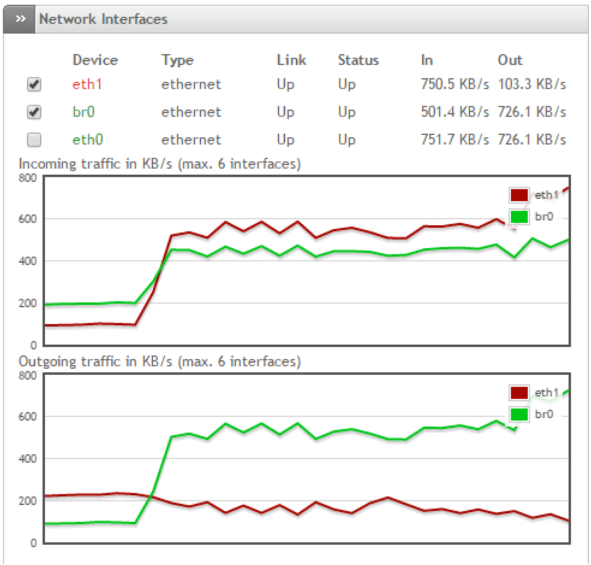


```
41025, 44334, 2100, 5554, 12203, 26000, 4000, 1000, 8014, 5250, 34443, 8028, 8008, 7510, 9495, 1581, 8000, 18881, 57772, 9090, 9999, 81, 3000, 8300, 8800, 8090, 389,
10203, 5093, 1533, 13500, 705, 4659, 20031, 16102, 6080, 6660, 11000, 19810, 3057, 6905, 1100, 10616, 10628, 5051, 1582, 65535, 105, 22222, 30000, 113, 1755, 407,
1434, 2049, 689, 3128, 20222, 20034, 7580, 7579, 38080, 12401, 910, 912, 11234, 46823, 5061, 5060, 2380, 69, 5800, 62514, 42, 5631, 902, 5985, 5986, 6000, 6001, 6002,
6003, 6004, 6005, 6006, 6007, 47001, 523, 3500, 6379, 8834
msf auxiliary(tcp) > set RHOSTS [redacted]
RHOSTS => [redacted]
msf auxiliary(tcp) > run -j
[*] Auxiliary module running as background job
[*] [redacted]:80 - TCP OPEN
[*] [redacted]:389 - TCP OPEN
[*] [redacted]:135 - TCP OPEN
[*] [redacted]:139 - TCP OPEN
[*] [redacted]:443 - TCP OPEN
[*] [redacted]:445 - TCP OPEN
[*] Scanned 1 of 1 hosts (100% complete)

[*] Starting host discovery scans
msf auxiliary(smb_version) >
```

Figure 4.6: Reconnaissance results

After gathering information about the target network devices, operating systems, and active ports in the first attack, the second launched attack was DDoS. The previous gathered information, from the victim' network will assist the attacker to determine which device will be the target machine by using the IP and the active port on that machine. However, in Chapter 2, the TCP SYN attack was described in detail. This type of attack will be used as DDoS attacks in this research.

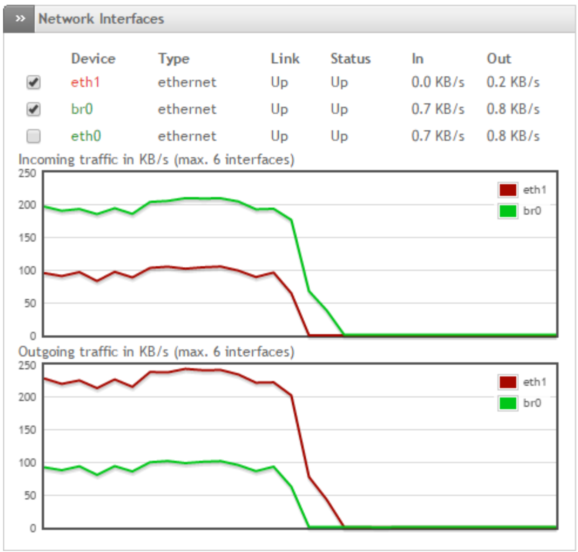
The aim of the attacker in this research from this attack in general is an attempt to make a network or machine resource unavailable to legitimate users. In this research, the commands baseline tool has been used to perform the DDoS attack. In order to ensure the attack was launched successfully, Figure 4.7 shows the status of the machine of the victim before and when the attacker performed the DDoS attack and the impact on the machine. Correspondingly, the effect of the attack on the device was clear and ran out the resources of the machine of the victim by sending a large number of incoming packets (these packets are TCP SYN, sent from the device of the attacker) into the victims' machine. Figure 4.8 shows the incoming packets during the attack and the impact when the attack was stopped and how the attack affected the machine from the response.


Network Interfaces

	Device	Type	Link	Status	In	Out
☑	eth1	ethernet	Up	Up	750.5 KB/s	103.3 KB/s
☑	br0	ethernet	Up	Up	501.4 KB/s	726.1 KB/s
☐	eth0	ethernet	Up	Up	751.7 KB/s	726.1 KB/s

Incoming traffic in KB/s (max. 6 interfaces)
800
600
400
200
0
🟥 eth1
🟩 br0
Outgoing traffic in KB/s (max. 6 interfaces)
800
600
400
200
0
🟥 eth1
🟩 br0

Figure 4.7: DDoS Attacks Before

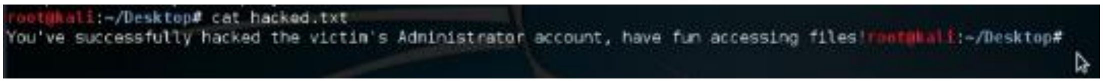

Network Interfaces

	Device	Type	Link	Status	In	Out
☑	eth1	ethernet	Up	Up	0.0 KB/s	0.2 KB/s
☑	br0	ethernet	Up	Up	0.7 KB/s	0.8 KB/s
☐	eth0	ethernet	Up	Up	0.7 KB/s	0.8 KB/s

Incoming traffic in KB/s (max. 6 interfaces)
250
200
150
100
50
0
🟥 eth1
🟩 br0
Outgoing traffic in KB/s (max. 6 interfaces)
250
200
150
100
50
0
🟥 eth1
🟩 br0

Figure 4.8: DDoS Attacks After

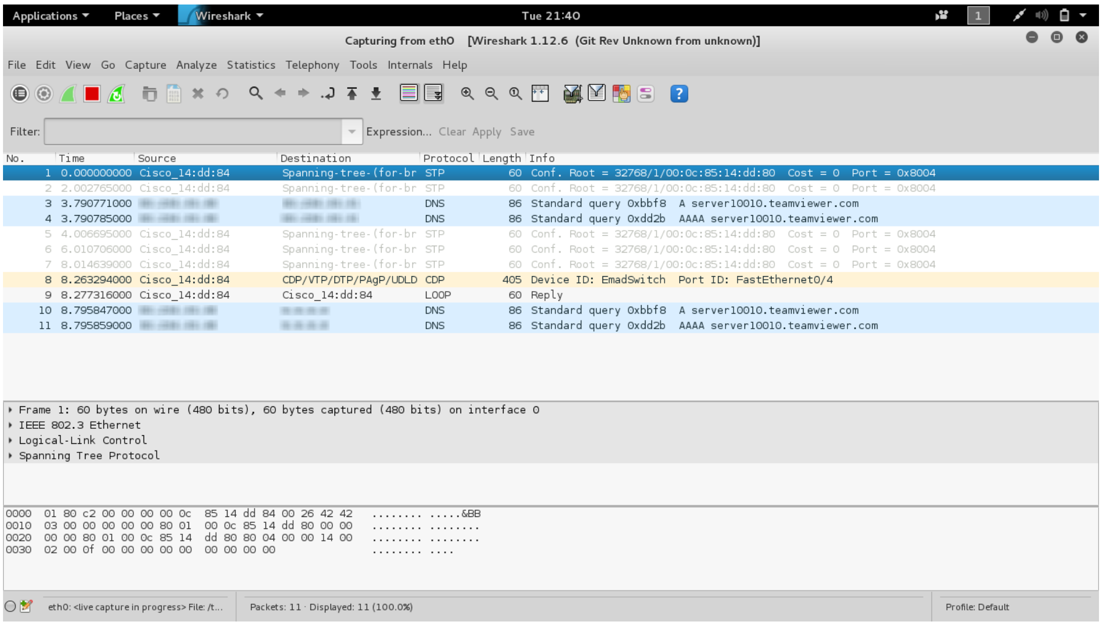
By creating lists of usernames and passwords and information gathered in the first attack, the Dictionary attack was ready to launch. By using these two lists and the IP of the victim and known port that gathered from first attack, the attacker could successfully obtain access to the victims' machine. Figure 4.9 shows the successful attack and the administrator account was compromised by the attacker.



```
root@kali:~/Desktop# cat hacked.txt
You've successfully hacked the victim's Administrator account, have fun accessing files!root@kali:~/Desktop#
```

Figure 4.9: Results of Dictionary attacks

The last selected attack in this research was Packet Sniffing. This attack was discussed in detail in Chapter 2, Section 2.5. Wireshark was installed and used in the attackers' device in order to ensure a successful attack. Once the attack was performed against the victims' device, the Wireshark started monitoring and recording all packets that came in and out from the victims' machine. The capability of Wireshark to record all traffic gave the attacker the ability to analyse all information in order to discover relevant information. Figure 4.10 shows the successful attack on the victims' machine.



Applications
Places
Wireshark
Tue 21:40
File Edit View Go Capture Analyze Statistics Telephony Tools Internals Help
Filter:
Expression... Clear Apply Save

Capturing from eth0 [Wireshark 1.12.6 (Git Rev Unknown from unknown)]

```
No.  Time         Source          Destination           Protocol Length Info
  1  0.000000000  Cisco_14:dd:84  Spanning-tree-(for-br STP       60  Conf. Root = 32768/1/00:0c:85:14:dd:80  Cost = 0  Port = 0x8004
  2  2.002765000  Cisco_14:dd:84  Spanning-tree-(for-br STP       60  Conf. Root = 32768/1/00:0c:85:14:dd:80  Cost = 0  Port = 0x8004
  3  3.790771000  [redacted]      [redacted]            DNS       86  Standard query 0xbbf8  A server10010.teamviewer.com
  4  3.790785000  [redacted]      [redacted]            DNS       86  Standard query 0xdd2b  AAAA server10010.teamviewer.com
  5  4.006695000  Cisco_14:dd:84  Spanning-tree-(for-br STP       60  Conf. Root = 32768/1/00:0c:85:14:dd:80  Cost = 0  Port = 0x8004
  6  6.010706000  Cisco_14:dd:84  Spanning-tree-(for-br STP       60  Conf. Root = 32768/1/00:0c:85:14:dd:80  Cost = 0  Port = 0x8004
  7  8.014639000  Cisco_14:dd:84  Spanning-tree-(for-br STP       60  Conf. Root = 32768/1/00:0c:85:14:dd:80  Cost = 0  Port = 0x8004
  8  8.263294000  Cisco_14:dd:84  CDP/VTP/DTP/PAgP/UDLD CDP      405  Device ID: EmadSwitch  Port ID: FastEthernet0/4
  9  8.277316000  Cisco_14:dd:84  Cisco_14:dd:84        LOOP      60  Reply
 10  8.795847000  [redacted]      [redacted]            DNS       86  Standard query 0xbbf8  A server10010.teamviewer.com
 11  8.795859000  [redacted]      [redacted]            DNS       86  Standard query 0xdd2b  AAAA server10010.teamviewer.com

▸ Frame 1: 60 bytes on wire (480 bits), 60 bytes captured (480 bits) on interface 0
▸ IEEE 802.3 Ethernet
▸ Logical-Link Control
▸ Spanning Tree Protocol

0000  01 80 c2 00 00 00 00 0c  85 14 dd 84 00 26 42 42   ........ .....&BB
0010  03 00 00 00 00 00 80 01  00 0c 85 14 dd 80 00 00   ........ ........
0020  00 00 80 01 00 0c 85 14  dd 80 80 04 00 00 14 00   ........ ........
0030  02 00 0f 00 00 00 00 00  00 00 00 00               ........ ....

eth0: <live capture in progress> File: /t...   Packets: 11 · Displayed: 11 (100.0%)   Profile: Default
```

Figure 4.10: Packet Sniffing attacks

4.3.2 Evidence Collection

In order to collect the results, all IDSs monitored and Argus recorded all the packets that passed the experimental devices during the research period, specifically during the stage of performing all attacks. This will assist the researcher to focus on the specific time of the attack. When an attack is performed, firstly, the researcher checked the hacker device in order to confirm that the attack was finished and had compromised the design system' device. Secondly, the researcher checked the alerts of all IDSs. Thirdly, the researcher examined the recording of Argus in order to ensure the connection between the attacker and the victims' machine is setup correctly. For example, the researcher performed the Dictionary attack, then checked all IDSs alerts one by one in order to collect the results and then inspected the recording of Argus. If the IDS did not generate alerts and the attacker device and Argus showed there were connections in this case, this means the IDS failed in the detection of this attack. If the IDS detected the attack and generated the alert about the incident, then in this case, checking the recording of Argus is used to reconstruct the connection and support the IDS' findings.

All alerts of the selected IDSs were stored in a secure database (in the form of tables) that were installed. These databases were extracted in order to do more investigating about the evidence. After the data generation had been finished and all selected attacks were performed, examination and analysis of which IDSs detect all or some attacks was made based on generated alerts about these incidents. These databases and log files are considered the main source of forensic digital evidence against the origin of the attacks in this research. The methodology of Pilli et al. (2010) mentioned that in evidence collection, this evidence should be handled carefully (see Chapter 2, Section 2.8). The acquisition of databases and log files that contain all alerts of IDSs consisted of two steps. The first step was exporting databases and log files. While the second step was to ensure that there was no alteration in all databases after transferring.

As mentioned previously in Section 3.5.4 and in Section 4.2.1 these databases of all IDSs was MySQL. The way to collect the evidence from all IDSs, databases and log files, consists of three steps. The first step was to export databases by using MySQL dump commands for all databases and zip for log files. To ensure integrity, before exporting databases and log files, the hash was created to be used later. The second step was to export/transfer all databases and log files into a forensic server. The third step was to ensure the integrity of evidence after transferring them into the forensic server through the network. Calculation of a cryptographic hash was used and compared with the previous value of the hash. Since there is no difference between the two hash values this means the transferring of the data was successful as shown in

Table 4.1. The step of checking and investigation of this information will lead to the next step.

Table 4.1: Hash values for all files

File Name	Before transferring	After transferring
Argus.zip	e754d9e299a9a6077d7d1eeff750f b82	e754d9e299a9a6077d7d1eeff750f b82
elsa_web.sql	d8033ed631d6ec8bde16468607e8afbd	d8033ed631d6ec8bde16468607e8afbd
prelude.sql	c850f3a0c167d1e86dd77c4389eb0407	c850f3a0c167d1e86dd77c4389eb0407
sguildb.sql	fabc299a9a8e16d7a401458d2ae936bf	fabc299a9a8e16d7a401458d2ae936bf
syslog_data.sql	c576d5475bd154d319e364bb392f83a0	c576d5475bd154d319e364bb392f83a0
ossec.sql	b76d45f1335d6df38502253b82698f33	b76d45f1335d6df38502253b82698f33
prewikka.sql	aaca7430a0d6955338181c508937dbb5	aaca7430a0d6955338181c508937dbb5
sguil_error.log	ab08a2294ca09ca7eff7f38db56d2b38	ab08a2294ca09ca7eff7f38db56d2b38
syslog.sql	e0a40d9476723e3864d9e97a6f61c078	e0a40d9476723e3864d9e97a6f61c078

In order to collect all evidence from all attacks, four attacks were part of this experiment. Once the first attack was established, the following step is to examine the evidence, which will be managed for all alarms that came from all IDSs in order to obtain and present the results. These steps are repeated for the other remaining attacks and evidence testing. An in-depth discussion of the findings of the four attacks is in Chapter 5 in order to identify all generated alerts from IDSs associated with all the attacks conducted.

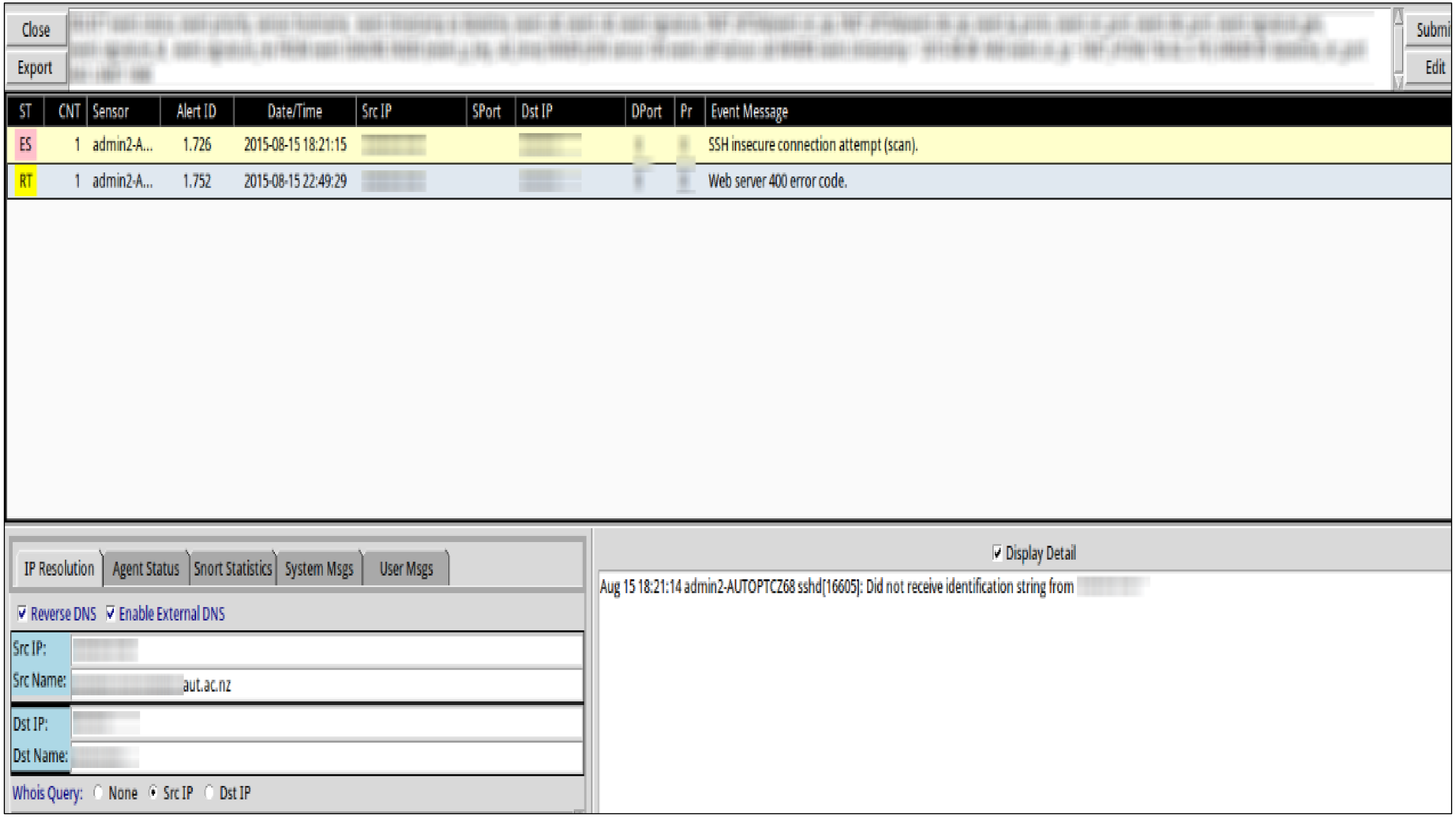
4.3.3 Evidence Examination and Analysis

The examinations and investigations from the experiment's results is grouped based on the name of the IDSs with four attacks. Therefore, the first investigation will discuss the results of four attacks with only one IDS. After that, the discussion will move to a second IDS and so on. The summary of the findings will be based on a similar approach.

4.3.3.1 Results of PADS

The first investigation and findings will discuss the results of PADS with all four selected network attacks in order to find a connection between the attacker and the victim as discussed in the methodology of Pilli et al. (2010). The first scenario of data generation of the design system was to launch the four attacks respectively and check the capability of PADS to detect and generate alerts about these attacks. The sequences of attacks were discussed in Section 4.3.3. However, PADS cannot export reports about these alerts and does not have a user interface as shown in Figure 4.5. Thus, there is no way to investigate these alerts, and present more information about the attacker. Therefore, Sguil was the first extra software used in this research for examination of the alerts of PADS. When Sguil was installed and linked with the database of PADS. This can pass these limitations. The use of Sguil in this research was to present the PADS alerts in a flexible user interface and provide information about the malicious incidents. The user interface of Sguil makes PADS' alerts more readable, visual and groups these alerts. In addition, Sguil has the ability to present more information about the source and destination IP (SRC IP and DES IP). This information will be valuable in order to investigate the attacker' IP location in order to understand more information about the attacker. The installation of PADS and other required packages and software are presented in Appendix 4. Launching the first attack against the LAN of the design system was successful as shown in Figure 4.6. PADS was observed and generated alerts immediately about the attack incidents. These produced alarms showing that there was an attempted attack on the network including the detail as displayed in Figure 4.11. These alerts contain information about the signature of the attack, source plus destination IPs and the port number for both devices.

In order to investigate PADS' alerts, Squert was the second tool used for examination and analysing the alerts of PADS in this research. Squert is written in Hypertext Pre-processor (PHP) language. This tool is linked with PADS databases only in this research. The main reason for installing Squert in this research with Sguil is getting more information on the investigation in PADS' alerts and the way of interacting with Sguil. Both tools work in conjunction and can provide a reliable digital evidence based on PADS' alerts. It has several significant advantages in order to produce reliable evidence such as, visualizing PADS' alerts that are shown in Sguil. The ability to locate the attackers' location based by IP by showing the country name with the flag, showing the signature of the attack, grouping these alerts in some common group and many other advantages as shown in Figure 4.12. One of most important aspects aside from these features is the simplicity of the installation. There are several steps that must be followed as shown in Appendix 9. Another advantage was that, the investigator could determine the time and Squert will show the events that occurred in that period and whether there was an attack attempt or not. This feature does not exist in Sguil.



[redacted]

ST	CNT	Sensor	Alert ID	Date/Time	Src IP	SPort	Dst IP	DPort	Pr	Event Message
ES	1	admin2-A...	1.726	2015-08-15 18:21:15	[redacted]		[redacted]	[redacted]	[redacted]	SSH insecure connection attempt (scan).
RT	1	admin2-A...	1.752	2015-08-15 22:49:29	[redacted]		[redacted]	[redacted]	[redacted]	Web server 400 error code.

☑ Reverse DNS ☑ Enable External DNS

Src IP: [redacted]

Src Name: [redacted]aut.ac.nz

Dst IP: [redacted]

Dst Name: [redacted]

Whois Query: ○ None ● Src IP ○ Dst IP

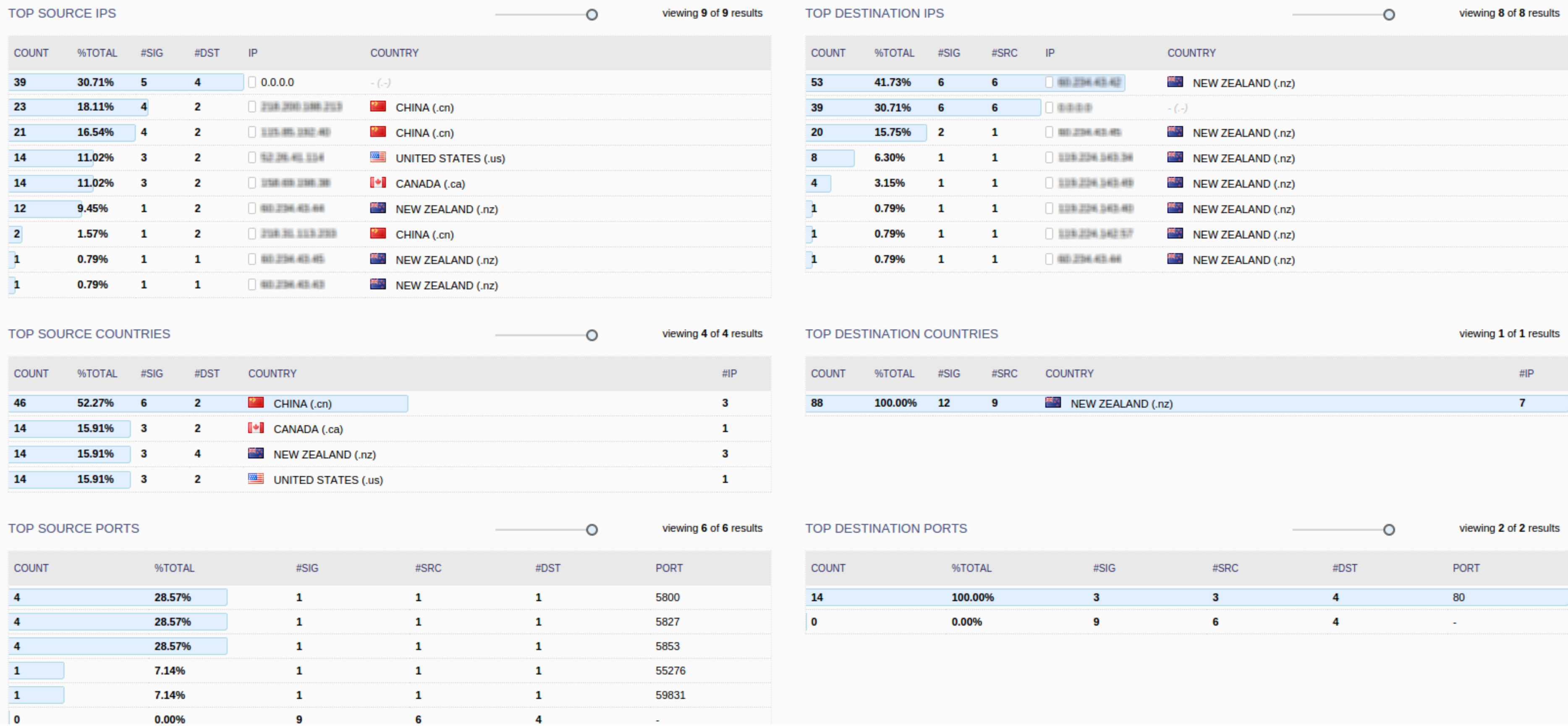
☑ Display Detail

```
Aug 15 18:21:14 admin2-AUTOPTCZ68 sshd[16605]: Did not receive identification string from [redacted]
```

Figure 4.11: PADS detecting Reconnaissance attack

If there was an attempt, Squert has the ability to show more information about the attack and determine the attacker location based on the IP in the summary tab as shown in Figure 4.13. This tool also provides more useful information about alerts such as, the time stamp of the incident, the number of attack attempts, event numbers and the port number that has been used in the attack. Squert shows more information about the attack than PADS and Sguil. Thus, the result of PADS was successful in detecting and generating alerts about the Reconnaissance attack immediately.

The second attack was then performed, followed by the third and fourth attacks as shown in Figures 4.12, to 4.15. However, PADS failed to detect them.



TOP SOURCE IPS

viewing 9 of 9 results

COUNT	%TOTAL	#SIG	#DST	IP	COUNTRY
39	30.71%	5	4	0.0.0.0	- (-)
23	18.11%	4	2	[redacted]	CHINA (.cn)
21	16.54%	4	2	[redacted]	CHINA (.cn)
14	11.02%	3	2	[redacted]	UNITED STATES (.us)
14	11.02%	3	2	[redacted]	CANADA (.ca)
12	9.45%	1	2	[redacted]	NEW ZEALAND (.nz)
2	1.57%	1	2	[redacted]	CHINA (.cn)
1	0.79%	1	1	[redacted]	NEW ZEALAND (.nz)
1	0.79%	1	1	[redacted]	NEW ZEALAND (.nz)

TOP DESTINATION IPS

viewing 8 of 8 results

COUNT	%TOTAL	#SIG	#SRC	IP	COUNTRY
53	41.73%	6	6	[redacted]	NEW ZEALAND (.nz)
39	30.71%	6	6	[redacted]	- (-)
20	15.75%	2	1	[redacted]	NEW ZEALAND (.nz)
8	6.30%	1	1	[redacted]	NEW ZEALAND (.nz)
4	3.15%	1	1	[redacted]	NEW ZEALAND (.nz)
1	0.79%	1	1	[redacted]	NEW ZEALAND (.nz)
1	0.79%	1	1	[redacted]	NEW ZEALAND (.nz)
1	0.79%	1	1	[redacted]	NEW ZEALAND (.nz)

TOP SOURCE COUNTRIES

viewing 4 of 4 results

COUNT	%TOTAL	#SIG	#DST	COUNTRY	#IP
46	52.27%	6	2	CHINA (.cn)	3
14	15.91%	3	2	CANADA (.ca)	1
14	15.91%	3	4	NEW ZEALAND (.nz)	3
14	15.91%	3	2	UNITED STATES (.us)	1

TOP DESTINATION COUNTRIES

viewing 1 of 1 results

COUNT	%TOTAL	#SIG	#SRC	COUNTRY	#IP
88	100.00%	12	9	NEW ZEALAND (.nz)	7

TOP SOURCE PORTS

viewing 6 of 6 results

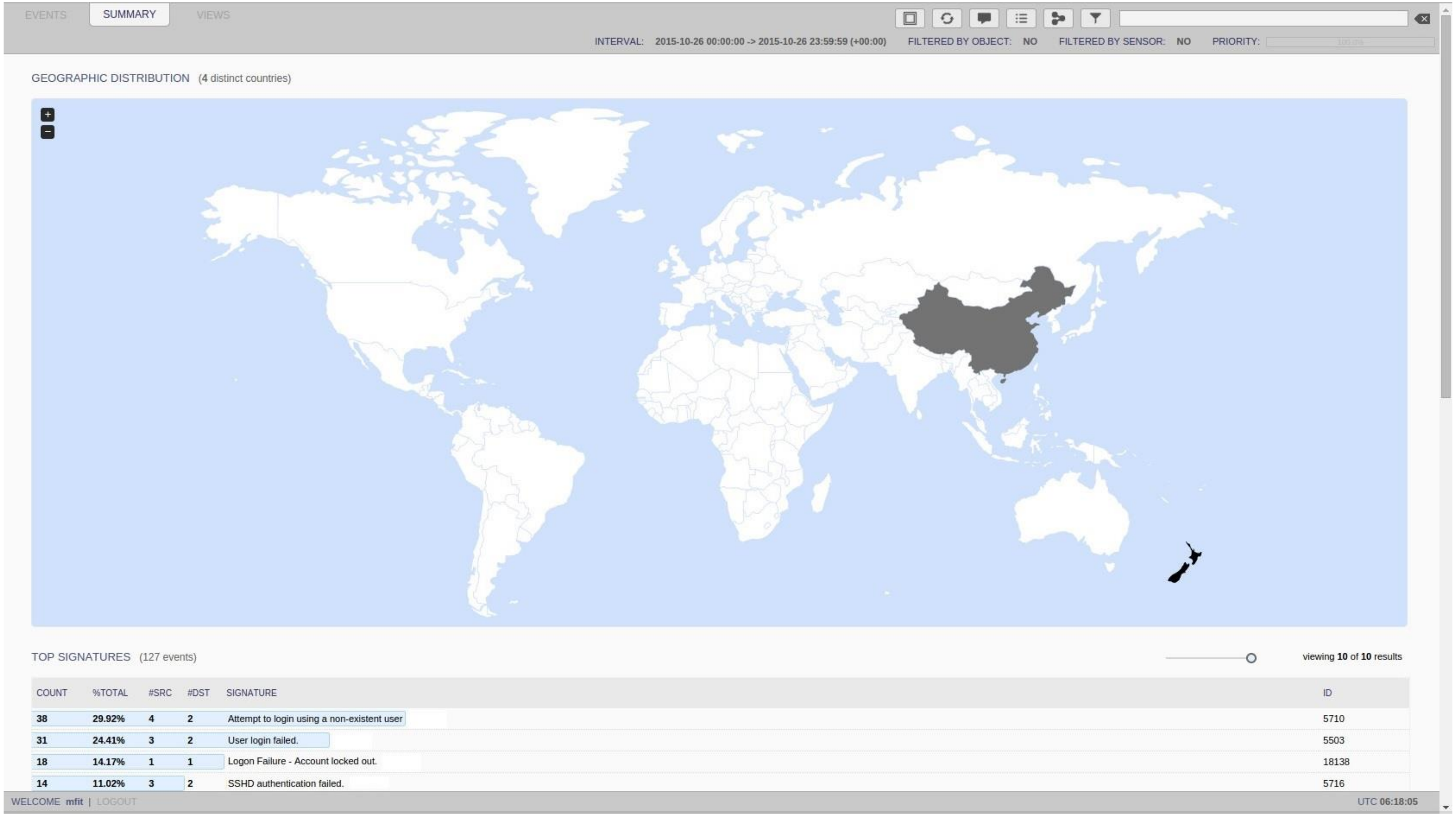
COUNT	%TOTAL	#SIG	#SRC	#DST	PORT
4	28.57%	1	1	1	5800
4	28.57%	1	1	1	5827
4	28.57%	1	1	1	5853
1	7.14%	1	1	1	55276
1	7.14%	1	1	1	59831
0	0.00%	9	6	4	-

TOP DESTINATION PORTS

viewing 2 of 2 results

COUNT	%TOTAL	#SIG	#SRC	#DST	PORT
14	100.00%	3	3	4	80
0	0.00%	9	6	4	-

Figure 4.12: Squert groups



INTERVAL: 2015-10-26 00:00:00 -> 2015-10-26 23:59:59 (+00:00) FILTERED BY OBJECT: NO FILTERED BY SENSOR: NO PRIORITY:

GEOGRAPHIC DISTRIBUTION (4 distinct countries)

TOP SIGNATURES (127 events)

viewing 10 of 10 results

COUNT	%TOTAL	#SRC	#DST	SIGNATURE	ID
38	29.92%	4	2	Attempt to login using a non-existent user	5710
31	24.41%	3	2	User login failed.	5503
18	14.17%	1	1	Logon Failure - Account locked out.	18138
14	11.02%	3	2	SSHD authentication failed.	5716

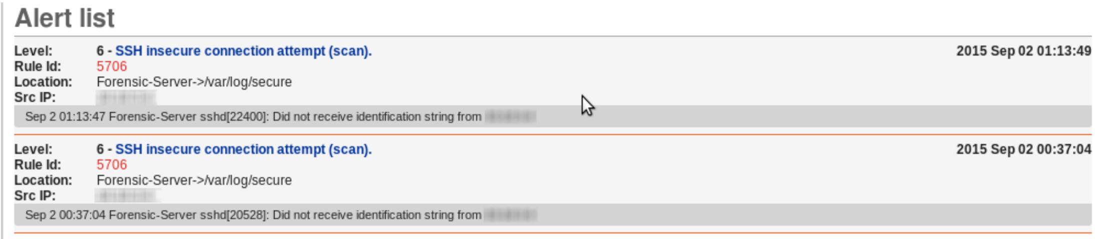
WELCOME mfit | LOGOUT

UTC 06:18:05

Figure 4.13: Location of the attacker and the victim

4.3.3.2 Results of OSSEC

The second tested IDS was OSSEC. At the same time of performing the Port Scanning attack, the researcher was observing OSSEC UI for detecting and alerting the attack incident. Once the first attack was launched, OSSEC generated alerts about the incidents immediately including the IP of the attacker, date, time and the signature of the attack as shown in Figure 4.14. In order to investigate more of the incident, Splunk was used based on the attacker IP that showed in the alerts of OSSEC UI. Splunk has many advantages and plays an important role in this research specifically when it is installed and linked with OSSECs' database. The OSSECs' GUI has some limitations. For example, it does not export incidents reports, has no search by specific words, and it does not allow adding or removing fields that were presented in the search results. These features are available by using of Splunk.

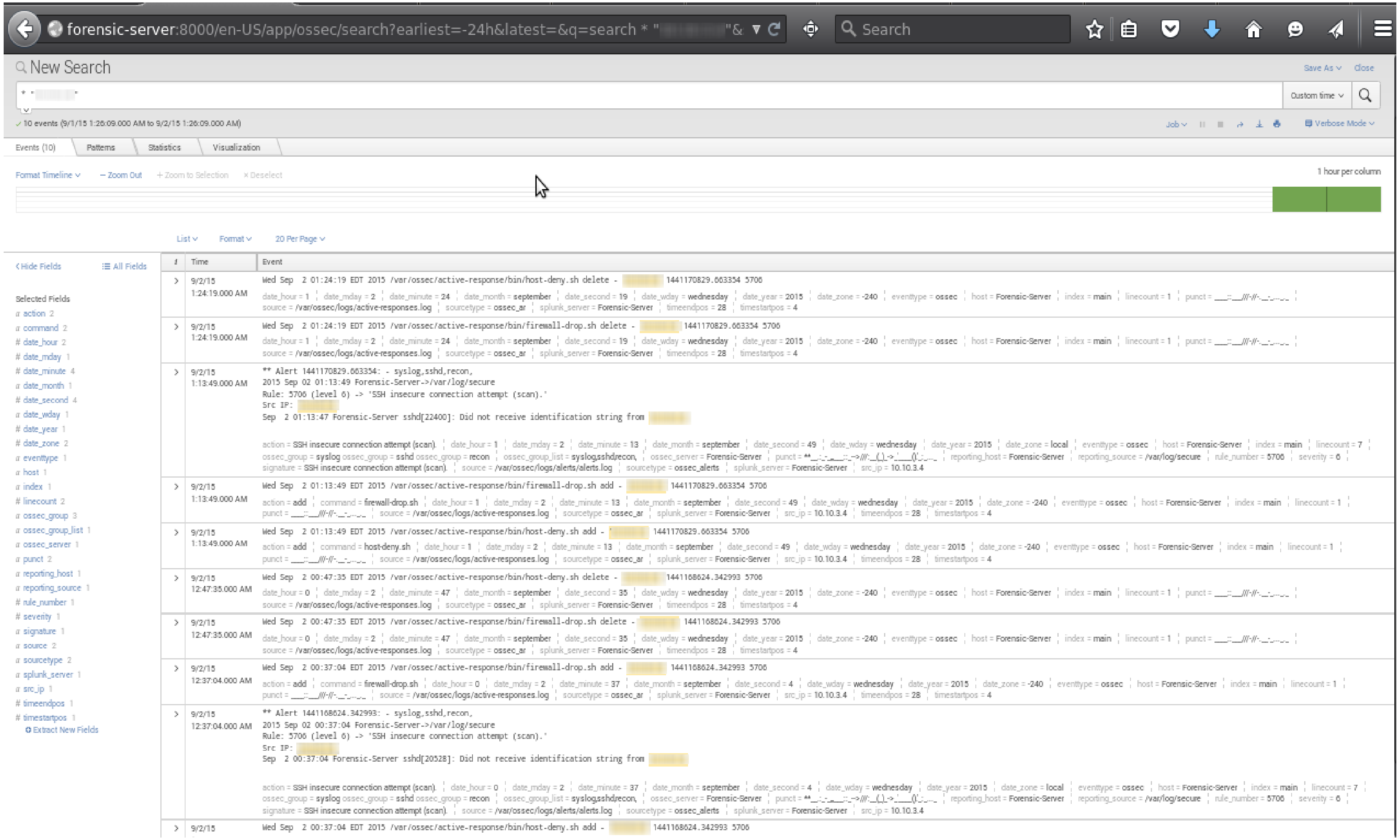


Alert list

```
Level:      6 - SSH insecure connection attempt (scan).                2015 Sep 02 01:13:49
Rule Id:    5706
Location:   Forensic-Server->/var/log/secure
Src IP:     [redacted]
Sep 2 01:13:47 Forensic-Server sshd[22400]: Did not receive identification string from [redacted]

Level:      6 - SSH insecure connection attempt (scan).                2015 Sep 02 00:37:04
Rule Id:    5706
Location:   Forensic-Server->/var/log/secure
Src IP:     [redacted]
Sep 2 00:37:04 Forensic-Server sshd[20528]: Did not receive identification string from [redacted]
```

Figure 4.14: OSSSEC alerts of Port Scanning attack

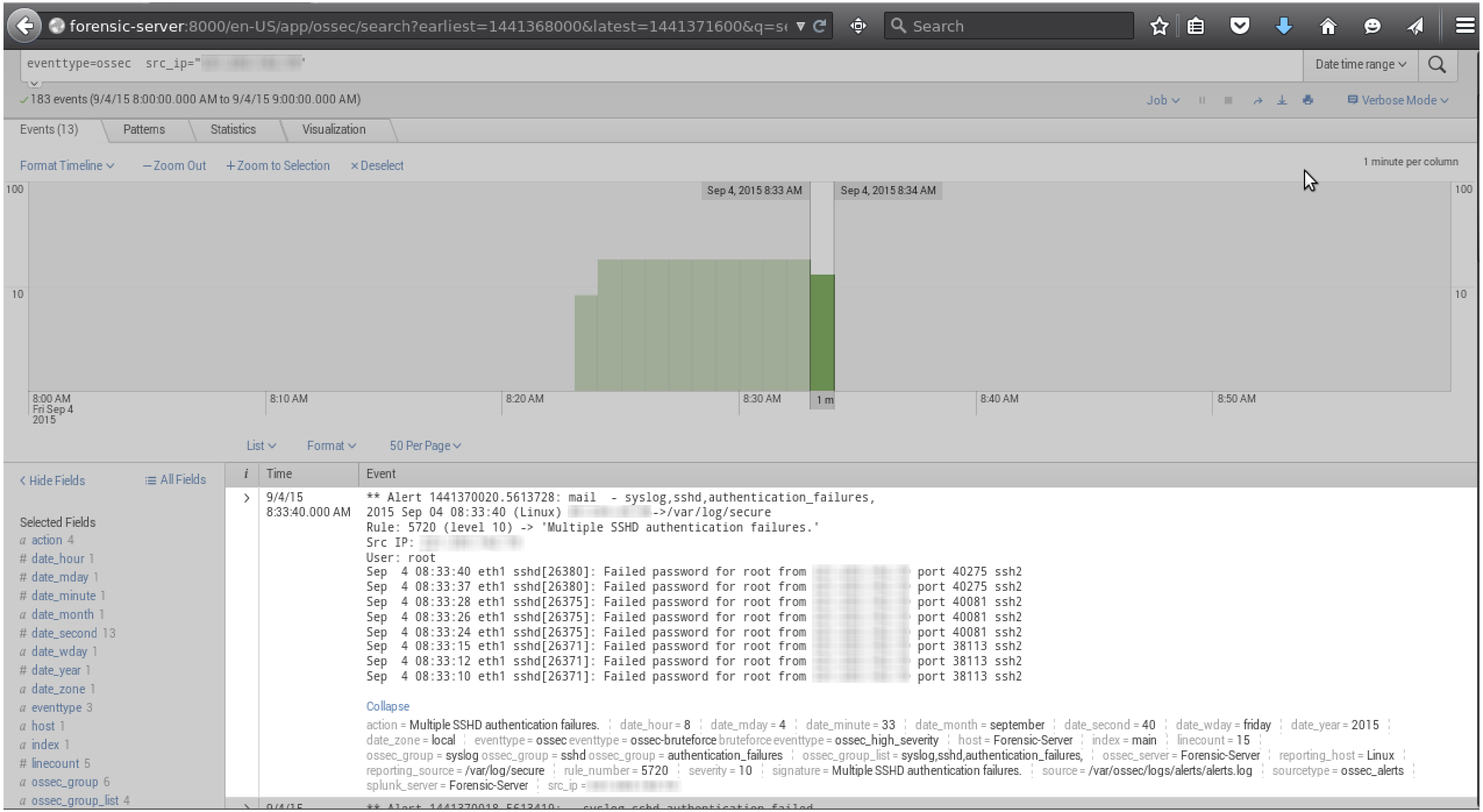
OSSEC GUI limitations make Splunk installation crucial in order to investigate the exact event(s) and extract reliable evidence from the incidents as shown in Figure 4.15. The installation of Splunk, required software, and files to connected with OSSECs databases and is attached in Appendix 5. Splunk confirmed the result and the attack that occurred in the system; and provided more information about this event. The search feature can present how many times this event occurred in the system by the same attacker IP. The second performed attack was not detected by OSSEC. While the recording of Argus and the computer used by the attacker confirmed the attack was committed. The third attack was detected and alerted immediately, once it was performed. In this detection, the researcher did all previous steps as in the first detection. Figure 4.16 shows the attacker was trying to obtain access to the victims' machine by using different passwords and ports for the root user account. OSSEC was not able to detect the Packet Sniffing attack.



✓ 10 events (9/1/15 1:26:09.000 AM to 9/2/15 1:26:09.000 AM)

```
  i  Time                    Event
  >  9/2/15                  Wed Sep  2 01:24:19 EDT 2015 /var/ossec/active-response/bin/host-deny.sh delete - [redacted] 1441170829.663354 5706
     1:24:19.000 AM          date_hour = 1 | date_mday = 2 | date_minute = 24 | date_month = september | date_second = 19 | date_wday = wednesday | date_year = 2015 | date_zone = -240 | eventtype = ossec | host = Forensic-Server | index = main | linecount = 1 | punct = ___::___///-//-.__-_...-._ |
                             source = /var/ossec/logs/active-responses.log | sourcetype = ossec_ar | splunk_server = Forensic-Server | timeendpos = 28 | timestartpos = 4
  >  9/2/15                  Wed Sep  2 01:24:19 EDT 2015 /var/ossec/active-response/bin/firewall-drop.sh delete - [redacted] 1441170829.663354 5706
     1:24:19.000 AM          date_hour = 1 | date_mday = 2 | date_minute = 24 | date_month = september | date_second = 19 | date_wday = wednesday | date_year = 2015 | date_zone = -240 | eventtype = ossec | host = Forensic-Server | index = main | linecount = 1 | punct = ___::___///-//-.__-_...-._ |
                             source = /var/ossec/logs/active-responses.log | sourcetype = ossec_ar | splunk_server = Forensic-Server | timeendpos = 28 | timestartpos = 4
  >  9/2/15                  ** Alert 1441170829.663354: - syslog,sshd,recon,
     1:13:49.000 AM          2015 Sep 02 01:13:49 Forensic-Server->/var/log/secure
                             Rule: 5706 (level 6) -> 'SSH insecure connection attempt (scan).'
                             Src IP: [redacted]
                             Sep  2 01:13:47 Forensic-Server sshd[22400]: Did not receive identification string from [redacted]

                             action = SSH insecure connection attempt (scan). | date_hour = 1 | date_mday = 2 | date_minute = 13 | date_month = september | date_second = 49 | date_wday = wednesday | date_year = 2015 | date_zone = local | eventtype = ossec | host = Forensic-Server | index = main | linecount = 7 |
                             ossec_group = syslog ossec_group = sshd ossec_group = recon | ossec_group_list = syslog,sshd,recon, | ossec_server = Forensic-Server | punct = **__:_-_.___::_->///:__(_)_->_'____()'_:_..._ | reporting_host = Forensic-Server | reporting_source = /var/log/secure | rule_number = 5706 | severity = 6 |
                             signature = SSH insecure connection attempt (scan). | source = /var/ossec/logs/alerts/alerts.log | sourcetype = ossec_alerts | splunk_server = Forensic-Server | src_ip = 10.10.3.4
  >  9/2/15                  Wed Sep  2 01:13:49 EDT 2015 /var/ossec/active-response/bin/firewall-drop.sh add - [redacted] 1441170829.663354 5706
     1:13:49.000 AM          action = add | command = firewall-drop.sh | date_hour = 1 | date_mday = 2 | date_minute = 13 | date_month = september | date_second = 49 | date_wday = wednesday | date_year = 2015 | date_zone = -240 | eventtype = ossec | host = Forensic-Server | index = main | linecount = 1 |
                             punct = ___::___///-//-.__-_...-._ | source = /var/ossec/logs/active-responses.log | sourcetype = ossec_ar | splunk_server = Forensic-Server | src_ip = 10.10.3.4 | timeendpos = 28 | timestartpos = 4
  >  9/2/15                  Wed Sep  2 01:13:49 EDT 2015 /var/ossec/active-response/bin/host-deny.sh add - [redacted] 1441170829.663354 5706
     1:13:49.000 AM          action = add | command = host-deny.sh | date_hour = 1 | date_mday = 2 | date_minute = 13 | date_month = september | date_second = 49 | date_wday = wednesday | date_year = 2015 | date_zone = -240 | eventtype = ossec | host = Forensic-Server | index = main | linecount = 1 |
                             punct = ___::___///-//-.__-_...-._ | source = /var/ossec/logs/active-responses.log | sourcetype = ossec_ar | splunk_server = Forensic-Server | src_ip = 10.10.3.4 | timeendpos = 28 | timestartpos = 4
  >  9/2/15                  Wed Sep  2 00:47:35 EDT 2015 /var/ossec/active-response/bin/host-deny.sh delete - [redacted] 1441168624.342993 5706
     12:47:35.000 AM         date_hour = 0 | date_mday = 2 | date_minute = 47 | date_month = september | date_second = 35 | date_wday = wednesday | date_year = 2015 | date_zone = -240 | eventtype = ossec | host = Forensic-Server | index = main | linecount = 1 | punct = ___::___///-//-.__-_...-._ |
                             source = /var/ossec/logs/active-responses.log | sourcetype = ossec_ar | splunk_server = Forensic-Server | timeendpos = 28 | timestartpos = 4
  >  9/2/15                  Wed Sep  2 00:47:35 EDT 2015 /var/ossec/active-response/bin/firewall-drop.sh delete - [redacted] 1441168624.342993 5706
     12:47:35.000 AM         date_hour = 0 | date_mday = 2 | date_minute = 47 | date_month = september | date_second = 35 | date_wday = wednesday | date_year = 2015 | date_zone = -240 | eventtype = ossec | host = Forensic-Server | index = main | linecount = 1 | punct = ___::___///-//-.__-_...-._ |
                             source = /var/ossec/logs/active-responses.log | sourcetype = ossec_ar | splunk_server = Forensic-Server | timeendpos = 28 | timestartpos = 4
  >  9/2/15                  Wed Sep  2 00:37:04 EDT 2015 /var/ossec/active-response/bin/firewall-drop.sh add - [redacted] 1441168624.342993 5706
     12:37:04.000 AM         action = add | command = firewall-drop.sh | date_hour = 0 | date_mday = 2 | date_minute = 37 | date_month = september | date_second = 4 | date_wday = wednesday | date_year = 2015 | date_zone = -240 | eventtype = ossec | host = Forensic-Server | index = main | linecount = 1 |
                             punct = ___::___///-//-.__-_...-._ | source = /var/ossec/logs/active-responses.log | sourcetype = ossec_ar | splunk_server = Forensic-Server | src_ip = 10.10.3.4 | timeendpos = 28 | timestartpos = 4
  >  9/2/15                  ** Alert 1441168624.342993: - syslog,sshd,recon,
     12:37:04.000 AM         2015 Sep 02 00:37:04 Forensic-Server->/var/log/secure
                             Rule: 5706 (level 6) -> 'SSH insecure connection attempt (scan).'
                             Src IP: [redacted]
                             Sep  2 00:37:04 Forensic-Server sshd[20528]: Did not receive identification string from [redacted]

                             action = SSH insecure connection attempt (scan). | date_hour = 0 | date_mday = 2 | date_minute = 37 | date_month = september | date_second = 4 | date_wday = wednesday | date_year = 2015 | date_zone = local | eventtype = ossec | host = Forensic-Server | index = main | linecount = 7 |
                             ossec_group = syslog ossec_group = sshd ossec_group = recon | ossec_group_list = syslog,sshd,recon, | ossec_server = Forensic-Server | punct = **__:_-_.___::_->///:__(_)_->_'____()'_:_..._ | reporting_host = Forensic-Server | reporting_source = /var/log/secure | rule_number = 5706 | severity = 6 |
                             signature = SSH insecure connection attempt (scan). | source = /var/ossec/logs/alerts/alerts.log | sourcetype = ossec_alerts | splunk_server = Forensic-Server | src_ip = 10.10.3.4
  >  9/2/15                  Wed Sep  2 00:37:04 EDT 2015 /var/ossec/active-response/bin/host-deny.sh add - [redacted] 1441168624.342993 5706
```

Figure 4.15: Investigation of Port Scanning attack



```
eventtype=ossec  src_ip="[redacted]"
```

✓183 events (9/4/15 8:00:00.000 AM to 9/4/15 9:00:00.000 AM)

9/4/15
8:33:40.000 AM

```
** Alert 1441370020.5613728: mail  - syslog,sshd,authentication_failures,
2015 Sep 04 08:33:40 (Linux) [redacted]->/var/log/secure
Rule: 5720 (level 10) -> 'Multiple SSHD authentication failures.'
Src IP: [redacted]
User: root
Sep  4 08:33:40 eth1 sshd[26380]: Failed password for root from [redacted] port 40275 ssh2
Sep  4 08:33:37 eth1 sshd[26380]: Failed password for root from [redacted] port 40275 ssh2
Sep  4 08:33:28 eth1 sshd[26375]: Failed password for root from [redacted] port 40081 ssh2
Sep  4 08:33:26 eth1 sshd[26375]: Failed password for root from [redacted] port 40081 ssh2
Sep  4 08:33:24 eth1 sshd[26375]: Failed password for root from [redacted] port 40081 ssh2
Sep  4 08:33:15 eth1 sshd[26371]: Failed password for root from [redacted] port 38113 ssh2
Sep  4 08:33:12 eth1 sshd[26371]: Failed password for root from [redacted] port 38113 ssh2
Sep  4 08:33:10 eth1 sshd[26371]: Failed password for root from [redacted] port 38113 ssh2
```

action = Multiple SSHD authentication failures. | date_hour = 8 | date_mday = 4 | date_minute = 33 | date_month = september | date_second = 40 | date_wday = friday | date_year = 2015 |
date_zone = local | eventtype = ossec eventtype = ossec-bruteforce bruteforce eventtype = ossec_high_severity | host = Forensic-Server | index = main | linecount = 15 |
ossec_group = syslog ossec_group = sshd ossec_group = authentication_failures | ossec_group_list = syslog,sshd,authentication_failures, | ossec_server = Forensic-Server | reporting_host = Linux |
reporting_source = /var/log/secure | rule_number = 5720 | severity = 10 | signature = Multiple SSHD authentication failures. | source = /var/ossec/logs/alerts/alerts.log | sourcetype = ossec_alerts |
splunk_server = Forensic-Server | src_ip = [redacted]

```
9/4/15  ** Alert 1441370018.5613410:  - syslog,sshd,authentication_failed
```

Figure 4.16: Dictionary Attack investigation

There are many features that exist in Splunk that do not exist in the OSSEC UI such as displaying the results in a graphic view as displayed in Figure 4.17 and exporting incident reports as shown in Figure 4.18.

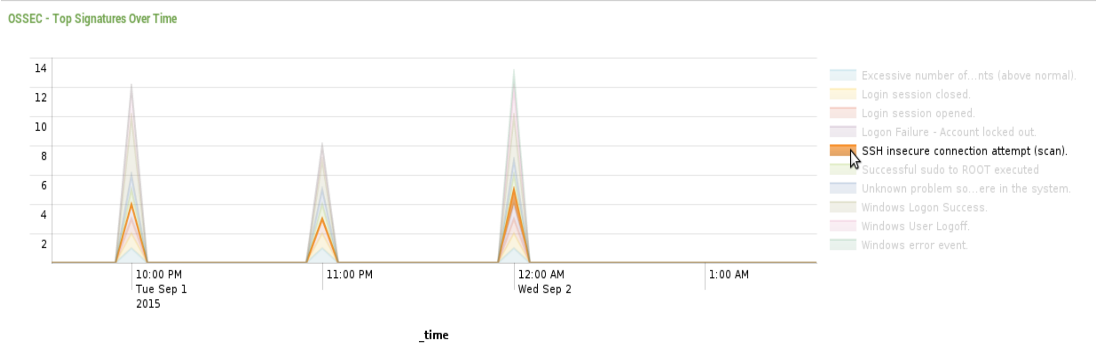
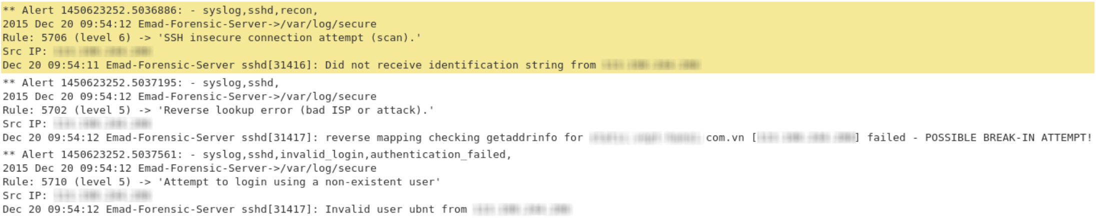


Figure 4.17: Reconnaissance attack result

It was found that OSSEC only detected the Port Scanning and Dictionary attacks. In both attacks, OSSEC generated alerts about the attack including the IP of the attacker, date and time of the attack, and the signature of the attack as shown in Figure 4.14. The experiment results include the two detected attacks generated by Splunk as displayed in Figure 4.18.

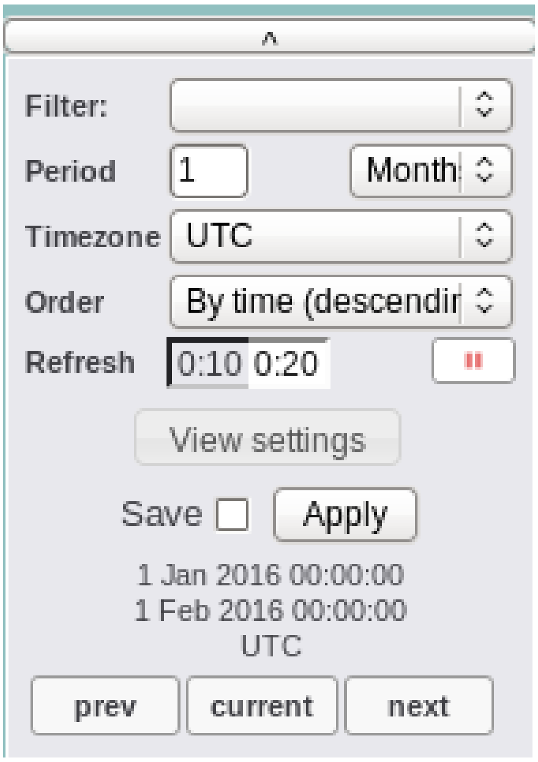


```
** Alert 1450623252.5036886: - syslog,sshd,recon,
2015 Dec 20 09:54:12 Emad-Forensic-Server->/var/log/secure
Rule: 5706 (level 6) -> 'SSH insecure connection attempt (scan).'
Src IP: [redacted]
Dec 20 09:54:11 Emad-Forensic-Server sshd[31416]: Did not receive identification string from [redacted]
** Alert 1450623252.5037195: - syslog,sshd,
2015 Dec 20 09:54:12 Emad-Forensic-Server->/var/log/secure
Rule: 5702 (level 5) -> 'Reverse lookup error (bad ISP or attack).'
Src IP: [redacted]
Dec 20 09:54:12 Emad-Forensic-Server sshd[31417]: reverse mapping checking getaddrinfo for [redacted] com.vn [[redacted]] failed - POSSIBLE BREAK-IN ATTEMPT!
** Alert 1450623252.5037561: - syslog,sshd,invalid_login,authentication_failed,
2015 Dec 20 09:54:12 Emad-Forensic-Server->/var/log/secure
Rule: 5710 (level 5) -> 'Attempt to login using a non-existent user'
Src IP: [redacted]
Dec 20 09:54:12 Emad-Forensic-Server sshd[31417]: Invalid user ubnt from [redacted]
```

Figure 4.18: Incidents' Report

4.3.3.3 Results of Prelude

The final IDS examined was Prelude. Prewikka is a graphical user interface that has the ability to present the alerts of Prelude. This presenting of all Preludes’ alerts can be configured by changing the setting of Prewikka as shown in Figure 4.19. One of the main points in this setting is the refreshing time. In this setting panel, the investigator/administrator sets the time of refreshment of the page as the minimum in seconds to refresh the web page, in order to monitor real-time packets traveling in and out of the LAN network and present the alerts in a short period of time. The web page timeframe can be a vulnerability if the network administrator who is responsible for monitoring the network is to set the refreshing time to a minute or more. This may be sufficient time for an attacker to perform, paralyze, and access the network before the web page is refreshed and present the attack alerts. Thus, in this research, the refreshing time was set to 20 seconds in order to keep up with live packet inspections.



Filter:

Period 1 Month

Timezone UTC

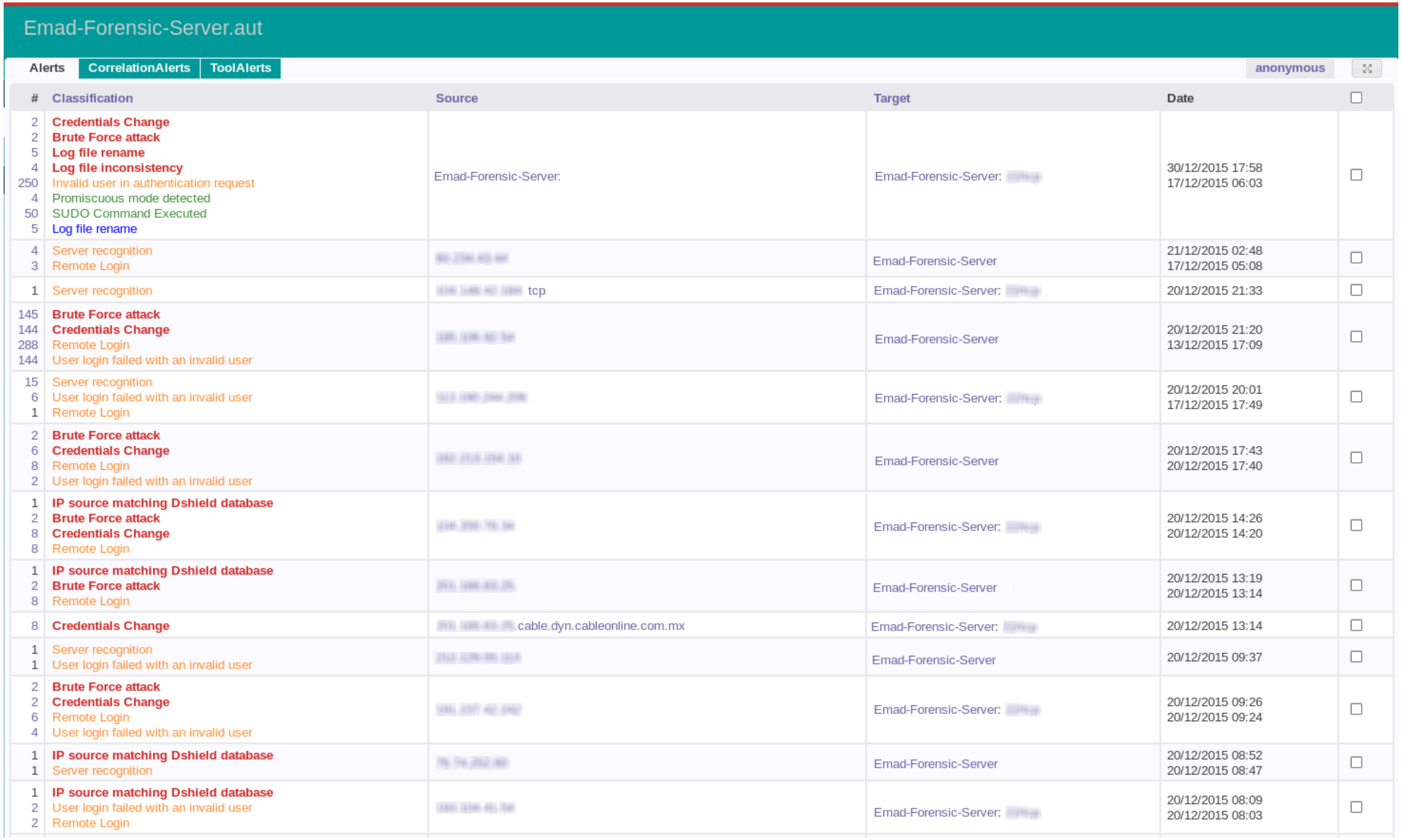
Order By time (descendir

Refresh 0:10 0:20

1 Jan 2016 00:00:00
1 Feb 2016 00:00:00
UTC

Figure 4.19: Prewikka Setting

Prelude detected the Port Scanning and generated alerts about this attack (under the name of Server recognition) as presented in Figure 4.20. In order to investigate the alert as well as obtain more information about the incident, the investigator can select these alerts and see the details.

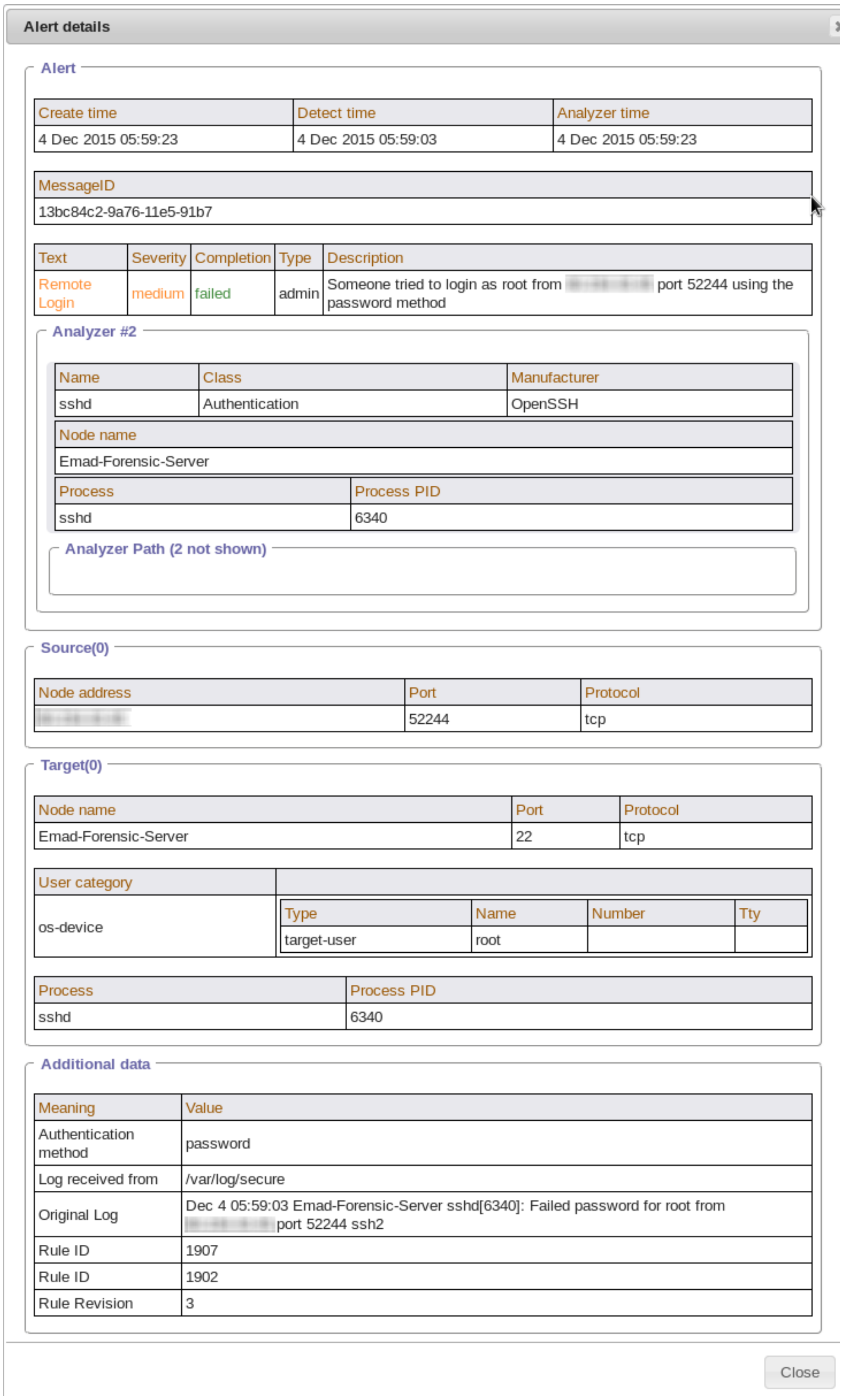


Emad-Forensic-Server.aut

#	Classification	Source	Target	Date	□
2 2 5 4 250 4 50 5	Credentials Change Brute Force attack Log file rename Log file inconsistency Invalid user in authentication request Promiscuous mode detected SUDO Command Executed Log file rename	Emad-Forensic-Server:	Emad-Forensic-Server: [redacted]	30/12/2015 17:58 17/12/2015 06:03	□
4 3	Server recognition Remote Login	[redacted]	Emad-Forensic-Server	21/12/2015 02:48 17/12/2015 05:08	□
1	Server recognition	[redacted] tcp	Emad-Forensic-Server: [redacted]	20/12/2015 21:33	□
145 144 288 144	Brute Force attack Credentials Change Remote Login User login failed with an invalid user	[redacted]	Emad-Forensic-Server	20/12/2015 21:20 13/12/2015 17:09	□
15 6 1	Server recognition User login failed with an invalid user Remote Login	[redacted]	Emad-Forensic-Server: [redacted]	20/12/2015 20:01 17/12/2015 17:49	□
2 6 8 2	Brute Force attack Credentials Change Remote Login User login failed with an invalid user	[redacted]	Emad-Forensic-Server	20/12/2015 17:43 20/12/2015 17:40	□
1 2 8 8	IP source matching Dshield database Brute Force attack Credentials Change Remote Login	[redacted]	Emad-Forensic-Server: [redacted]	20/12/2015 14:26 20/12/2015 14:20	□
1 2 8	IP source matching Dshield database Brute Force attack Remote Login	[redacted]	Emad-Forensic-Server	20/12/2015 13:19 20/12/2015 13:14	□
8	Credentials Change	[redacted].cable.dyn.cableonline.com.mx	Emad-Forensic-Server: [redacted]	20/12/2015 13:14	□
1 1	Server recognition User login failed with an invalid user	[redacted]	Emad-Forensic-Server	20/12/2015 09:37	□
2 2 6 4	Brute Force attack Credentials Change Remote Login User login failed with an invalid user	[redacted]	Emad-Forensic-Server: [redacted]	20/12/2015 09:26 20/12/2015 09:24	□
1 1	IP source matching Dshield database Server recognition	[redacted]	Emad-Forensic-Server	20/12/2015 08:52 20/12/2015 08:47	□
1 2 2	IP source matching Dshield database User login failed with an invalid user Remote Login	[redacted]	Emad-Forensic-Server: [redacted]	20/12/2015 08:09 20/12/2015 08:03	□

Figure 4.20: Prelude Alerts

For example, in Figure 4.21, Prelude has generated alarms for the Dictionary attack detection. By investigating these alarms status of the attack (success or fail), can be seen as well as the target machine in the system. Furthermore, the IP address of the attacker, the port number that has been used in the attack and other useful information about the attacker and the attack is presented.



Alert details

Alert

Create time	Detect time	Analyzer time
4 Dec 2015 05:59:23	4 Dec 2015 05:59:03	4 Dec 2015 05:59:23

MessageID
13bc84c2-9a76-11e5-91b7

Text	Severity	Completion	Type	Description
Remote Login	medium	failed	admin	Someone tried to login as root from [redacted] port 52244 using the password method

Analyzer #2

Name	Class	Manufacturer
sshd	Authentication	OpenSSH

Node name
Emad-Forensic-Server

Process	Process PID
sshd	6340

Analyzer Path (2 not shown)

Source(0)

Node address	Port	Protocol
[redacted]	52244	tcp

Target(0)

Node name	Port	Protocol
Emad-Forensic-Server	22	tcp

User category	Type	Name	Number	Tty
os-device	target-user	root		

Process	Process PID
sshd	6340

Additional data

Meaning	Value
Authentication method	password
Log received from	/var/log/secure
Original Log	Dec 4 05:59:03 Emad-Forensic-Server sshd[6340]: Failed password for root from [redacted] port 52244 ssh2
Rule ID	1907
Rule ID	1902
Rule Revision	3

Figure 4.21: Dictionary Attack (Prelude)

The second attack, Prelude fails in detecting the DDoS attack that targeted the machine in the test system. However, the third attack (Dictionary attack) and fourth attack (Packet Sniffing attack) were detected and the alerts were generated. These attacks were investigated by following the same investigation steps as in the previous attack.

The following section will summarize and present these findings in order to answer the research question.

4.3.4 Presentation of the Findings

In this section, findings recovered from the investigation and examination of evidence are presented. This presentation of findings will assist in answering the research question and sub-questions outlined in Chapter 3. The findings will be presented based on the evidence found, regardless of some evidence being partially recovered. If the IDS has detected an attack type, and the evidence outlined in the IDS is partial, which means some missing information was not recovered, then the IDS has succeeded. The results are generally presented visually in order to identify the specific attack easily. Chapter 5 will discuss this information in depth. These results will be presented in a visual way in order to be intelligible, clear and efficient as mentioned in the presentation of the evidence by Pilli et al. (2010).

The first presentation of the findings was all alerts for PADS. The data generation was executed based on the discussion in Chapter 3, Section 3.3.2, and Section 4.2.1. The first figure presents the result of four attacks utilising PADS. The order of all attacks is as follows: firstly, Port Scanning, then DDoS followed by the third attack, which was Dictionary, and the last attack, which is Packet Sniffing. This technique used for testing all IDSs one followed by the others. The findings of PADS from all attacks is shown in the Figure 4.22.

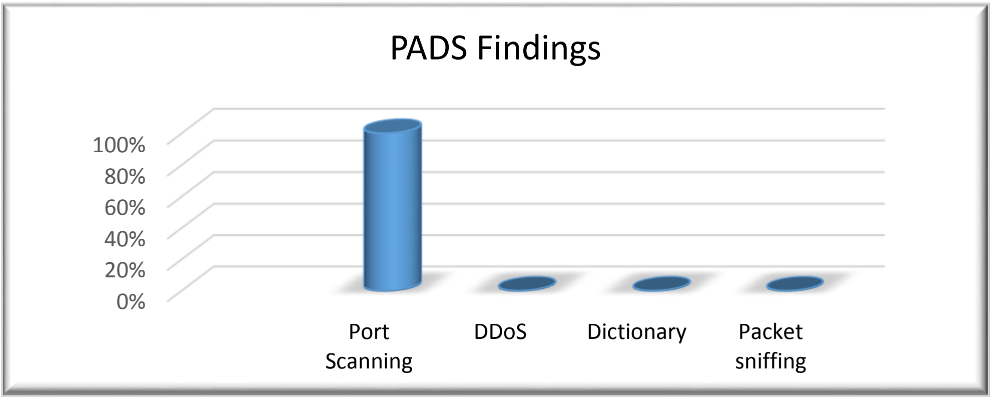


Figure 4.22: PADS results

The second demonstration of alerts is for OSSEC IDS. By checking the alerts, and analysing the alerts, investigation of all alerts information that was generated by OSSEC, and the results are presented in Figure 4.23.

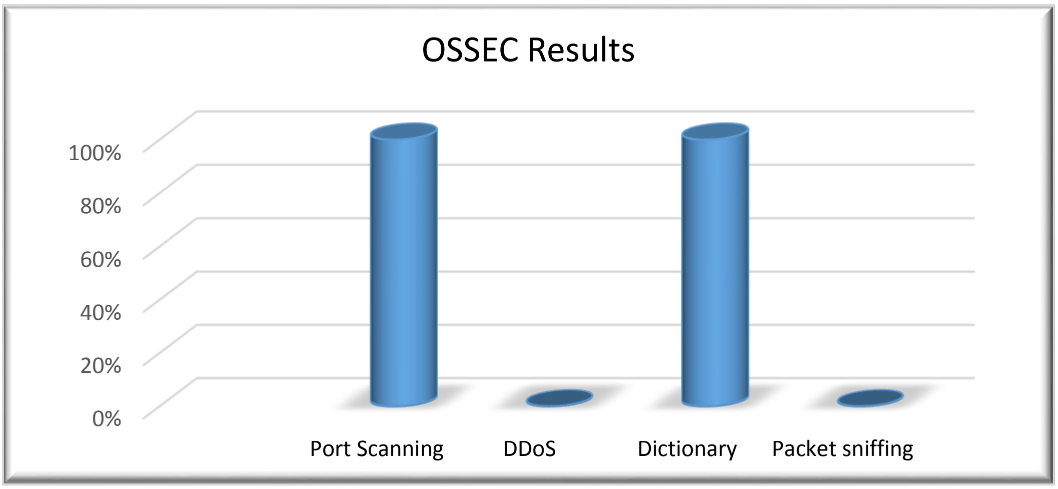


Figure 4.23: Findings of OSSEC

Prelude IDS was the last IDS examined in this research. The results of all attacks that occurred in the designed system for Prelude are displayed in Figure 4.24.

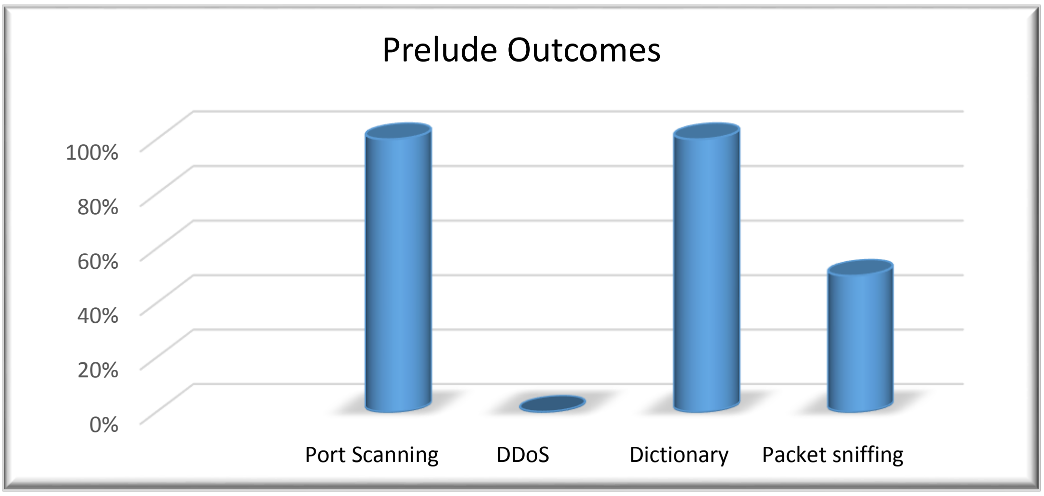


Figure 4.24: The result of Prelude alters

4.4 COMPARATIVE ANALYSIS AND EVALUATION

The main aim of doing comparative analysis is to summarize the findings of all the IDSs and to compare the events of the attacks that are discussed in Section 4.3.3 and Section 4.3.4. The comparative analysis table is essential to answer the main research question and the sub-questions as shown in Table 4.2.

Table 4.2: Comparative Analysis

Type of Attacks	**Intrusion Detection System**		
	PADS	**OSSEC**	**Prelude**
TCP SYN flood	Not Found	Not Found	Not Found
Port Scan	Found	Found	Found
Dictionary	Not Found	Found	Found
Packet Sniffing	Not Found	Not Found	Partially Found

This table is the basis of the discussion in Chapter 5 in order to evaluate these IDSs and discuss the ability of each IDS and their results. Additionally, the main question and sub-questions will be based on the results from this table.

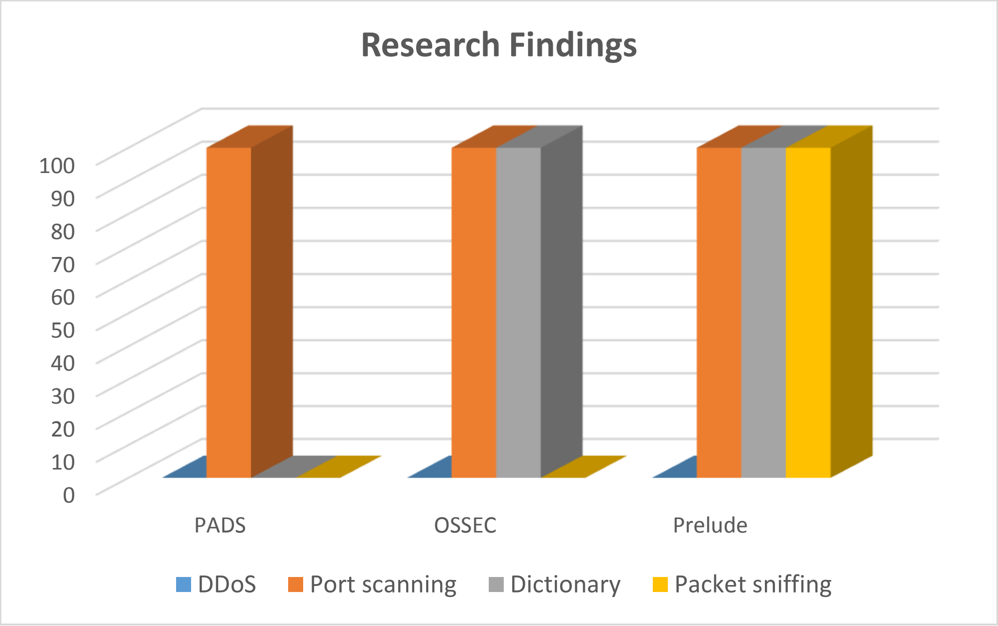


Figure 4.25: The capabilities of all IDSs

Figure 4.25 shows the comparison between all IDSs in detecting the four selected attacks in this research. Each IDS has different abilities than the other. For example, all IDSs failed in detecting DDoS but success in detecting Port Scanning.

4.5 CONCLUSION

The findings of the research were collected, analysed, reported and presented in the visual method based on the methodology of (Pilli et al., 2010). The attack phase was launched by performing four attacks. The first attack was Port Scanning then the second one was DDoS followed by the third 'Dictionary' and the last attack was Packet Sniffing. The IDSs were working simultaneously in the IDS machine and ready to detect and alert for all attacks that targeted the system machines. Databases of all IDSs and log files were collected for analysis. The integrity of the databases was ensured during the transmission by calculation of the two hash values, which were compared in order to ensure the evidence maintained its integrity.

The findings of this experiment show that all IDSs have some advantages in detection and alerting the selected LAN network attacks over others. The alerts information including important data, such as the signature of the attack, timestamp of incidents and the IP of the attacker were collected. Furthermore, the design of the system, the tools used and the software demonstrated the capture of the live packets in a LAN network can be accomplished and forensic evidence can be preserved.

Chapter 5

Discussion of Findings

5.0 INTRODUCTION

Based on the methodology discussed in Chapter 3, the findings of the experiment were analysed and described in Chapter 4. The main purposed of this study is to identify the value and to examine the abilities of the three IDSs. These IDSs were installed and deployed in a LAN network in order to produce forensic evidence for investigators. The significant and major findings in the experiment will be discussed in Chapter 5. The results will be evaluated in order to provide recommendations and suggestions for improved practices for digital forensics investigators in LAN networks utilising these IDSs software.

The structure of Chapter 5 will comprise the developed main question of the research that is discussed in Section 3.1 and this will be answered in Section 5., including answers to the sub-questions. The answer of the sub-questions will be grouped and displayed in tabulated forms in order to analyse the answers of these sub-questions and to make a summary of the answers. While, in Section 5.2, the results of the research and evaluation of the extra third party tools will be described. The recommendations and suggestions of all IDSs will be addressed in Section 5.3. The conclusion and summary of the discussion of the chapter is in Section 5.4

5.1 RESEARCH QUESTIONS

After the experiment was completed, the main question is ready to be answered. In the following subsections, the sub-question and the main question answers will be discussed and evaluated based on the findings that are presented in Chapter 4.

5.1.1 Sub-questions

Six sub-questions were formulated in Chapter 3 in order to support the main question. In this section, each sub-question is formatted into a table form providing the question, the answer to the question and a summary that provides the main facts observed from the experiment results.

Table 5.1: Sub-Question 1 and Answer

***Sub-question 1**: Of the three tested IDSs, which one performs most effectively at detecting network intrusions?*
The answer: The best IDS in detecting the network intrusions is Prelude
Summary: Although all the selected IDSs failed to detect DDoS attack, the IDSs have succeeded in detecting a Port Scanning attack. The Dictionary attack was detected by OSSEC and Prelude, and Packet Sniffing was only detected by Prelude. Thus, Prelude was the best at detecting network intrusion attempts.

Table 5.2: Sub-Question 2 and Answer

***Sub-question 2**:* What percentage of the network attacks are detected?
The answer: For PADS IDS: • The percentage of attacks recovered from Port Scanning attacks is 100% • The percentage of attacks recovered from DDoS, Dictionary, and Packet Sniffing attacks is 0%. FOR OSSEC IDS: • The percentage of attacks recovered from Port Scanning, and Dictionary attacks is 100%. • The percentage of attacks recovered from DDoS, and Packet Sniffing attacks is 0%. FOR Prelude IDS: • The percentage of attacks recovered from Port Scanning, and Dictionary attacks is 100%. • The percentage of attacks recovered from DDoS attack is 0%. • The percentage of attacks recovered from Packet Sniffing attack is 50%.

Table 5.3: Sub-Question 3 and Answer

Sub-question 3: Can the selected IDSs identify a repeated attack as the same attack?
The answer: YES for Prelude and PADS
Summary: Only Prelude and PADS can identify a repeated attack as the same attack. When a certain attack is performed many times against the network, in OSSEC IDS, the detection alert is repeated separately which makes it hard for the

investigator to follow up and count how many times a certain attack was detected. Therefore, OSSEC does not classify or categorize the events based on the type of attack. PADS and Prelude classify the detection alerts. For instance, Prelude can categorize these based on the source of the attack (IP address), and if the same attack is performed by the same IP address, then Prelude categorizes this attack as one event but with different attempt numbers. However, one drawback is that the investigator has to click on this event to view the number of attempts detected.

Table 5.4: Sub-Question 4 and Answer

***Sub-question 4**:* Which of the three IDSs results in the most forensically sound evidence from a given attack?
The answer: All of them provide forensically sound evidence from a given attack.
Summary: Although all the IDSs succeeded in identifying source and destination IP, port numbers, events detail such as type of attack and the number of events, and signature of attack. Nevertheless, OSSEC with Splunk only presents the status of each event clearly. In graphical user interface of Splunk, each event status is presented to the investigator whether the attack event failed, or succeeded which assist investigator(s) to focus on succeeded attack. In the other IDSs, the investigator has to click on each event in order to distinguish between the successful and the failed attacks.

Table 5.5: Sub-Question 5 and Answer

Sub-question 5: Which of the three IDSs working alone performs best in identifying an attack?
The answer: Prelude
Summary: Prelude identifying more attacks than other two IDSs and has an advantage over the other IDSs in terms of partially finding Packet Sniffing attacks and presenting the alerts in visual and clear groups while other failed to detect this attack.

Table 5.6: Sub-Question 6 and Answer

Sub-question 6: If extra functionality from a third party vendor is added to the IDSs, does this make the IDSs better in providing more information?
The answer: YES
Summary: PADS has succeeded in detecting reliable evidence from Port Scanning. Furthermore, if PADS connects with Sguil and Squert, the revealed evidence is displayed more efficiently with more information about the event(s). This can assist the network investigator to discover more information about intruder such as the signature of the attacks and other such useful information. OSSEC has succeeded in terms of finding sufficient evidence in Port Scanning, and Dictionary attack, the recovered evidence is presented more efficiently than the other three IDSs. Additionally, if OSSEC connects with Splunk, the investigator can run a custom search command for evidence easier, check integrity, and generate and export reports easily in different formats. Prelude has discovered more attacks than other two IDSs but failed to detect DDoS. Additionally, if Prelude connects with third party software such as Prewikka, the investigator can show information about the attack in a visual manner. However, if Prelude connects with Suricata, the investigator can gather more information about DDoS attacks.

5.1.2 The research question

The main research question given in Chapter 3, has directed the phases of the research. The question of this research is:

Do the three selected IDSs working together detect more attacks than any one single IDS?

This question led the researcher to create a system design with different machines and operating systems in order to answer it. The objective of this research was to test the ability of LAN network IDSs in order to produce forensic evidence from four selected network attacking activities as well as to evaluate the evidence that can be gathered from these selected IDSs in a forensic investigation. This led the researcher to divide the experiment into five sections, as discussed in Chapter 3.

The most important steps in the stages of the experimental design were step three and four. These steps include the main activities of the research and major events of the research, such as launching the attacks, acquisition and presentation of the evidence. This evidence was analysed based on the alerts of IDSs, and the result of captured LAN packets was also examined by external software, such as Argus in order to confirm the abilities of these IDSs to discover the attacking activities. The collected evidence was transformed into evidence result tables in order to show the abilities of these IDSs.

In order to obtain accepted findings for this research, the researcher installed and used additional software, packages and tools that can assist him to manage all network devices correctly and ensure the network connectivity is configured accurately so that each packet reaches its destination properly. For example, the installed and deployed Salt that has ability to manage remotely all machines of the LAN networks systems.

In order to acquire forensic evidence from the LAN network, extra tools were installed alongside the IDSs. These tools were examined and recommended by many other researchers, such as Pilli et al. (2010). Furthermore, these tools were used to analyse the generated alerts from all IDSs in relation to the incidents that occurred in the design of the experimental system.

The four attacks were performed one following another, and the results were collected from all machines of the experimental system. These findings were summarized into separate graphs based on the name of each IDS used, and this was followed by a table of comparative analysis. This illustrated all the IDSs' findings with performed attacks, regardless of the IDSs' findings and whether they detected the attacker IP, or not. This comparative analysis table will assist the researcher in answering the main question and sub-questions, and allow the discussion of the findings based on the abilities of all selected IDSs.

The answer of the main research question, based on the research findings is as follows:

> Prelude detected three of the selected LAN network attacks, which are Port Scanning, Dictionary, and Packet Sniffing attack, while OSSEC detected Port Scanning and Dictionary attack, and PADS only detected Port

Scanning attack. Additionally, all of them failed to detect DDoS attack. Thus, the three selected IDSs working together do not detect more attacks than a single IDS working individually. Prelude detected all of the attacks that have been detected by the other two IDSs, which leads to conclusion that Prelude has better capability in detecting the attacks than the two other IDSs.

5.2 DISCUSSION

An initial objective of the project was to identify the capability of selected IDSs. The researcher examined and evaluated three IDSs and identified significant results. The experimental procedure started by setting up the testing environment for a case scenario, and then extracting evidence from each IDS. The strengths and weaknesses of each IDS were identified. In this section, the research findings will be discussed in Section 5.2.1, and a discussion on the conducted experiment and findings of the case scenario is made in Section 5.2.2. Then the results of three select IDSs will be discussed in Section 5.2.3.

5.2.1 Discussion of Findings

There are several definitions of digital forensics on the Internet, books, articles and other resources but Palmer (2001) defines digital forensics as:

> The use of scientifically derived and proven methods toward the preservation, validation, identification, analysis, interpretation, documentation and presentation of digital evidence derived from digital sources for the purpose of facilitating or furthering the reconstruction of events found to be criminal, or helping to anticipate unauthorized actions shown to be disruptive to planned operations. (p.16).

Furthermore, digital evidence is defined as “any data stored or transmitted using a computer that supports or refutes a theory of how an offense occurred or that address critical elements of the offense, such as intent or alibi” (Casey, 2011, p.7). These two definitions were the base of the experimental tests of this research and presented in Chapter 2. Therefore, the digital investigation was designed as a network intrusion incident based on IDS alarms.

In this section, a discussion is made on the case scenarios that were used in this research to conduct the experiment and then collect the research findings. The discussion of the research findings will then follow.

5.2.1.1 Discussion of the Case Scenarios

In this research, the researcher developed the case scenarios to be as close to real scenarios as possible. These scenarios were performing Port Scanning then DDoS followed by Dictionary and lastly Packet Sniffing attacks. Before collecting the results from the experiment, there were several stages that were required. The first stage was to install a small LAN network and validate the requirements to run it correctly. The second stage was to study the possible data that can be extracted from all selected IDSs when performing the four selected attacks, which then can be used as forensic evidence. Therefore, different operating systems were installed in several devices. The third stage was to identify other studies that discussed and investigated in the abilities of IDSs to detect the attacks in LAN networks. The main reason for this was to avoid any issue that might occur and know how other researchers conducted their study and their way of presenting these detections to use as a guideline. All attacks were performed on system machines and each attack was recorded along with the precise date and time. Although these stages were not part of answering the primary research question and sub-questions, it is essential to have an appropriate environment setup. It was also crucial to know what precisely has occurred when the attack was performed and which device of the design system was affected. Therefore, this requires a LAN network investigator to be aware of what can be predictably detected by the three selected IDSs.

The literature review in chapter 2 discussed the aim of the attacker when performing the port scan attack. Morris et al. (2012) highlighted the aim of the port scan attack in general, is to gather useful information about LAN networks devices such as which devices are working and what ports are open. In Section 2.5.2, Allen et al. (2005) discussed different types of port scan attack that can be performed in LAN networks. Based on the investigation conducted about the port scan, this information was shown to be applicable in this research and the attacker acquired all information about the system design when performing the attack.

In Section 2.1, Sudin et al. (2008) discussed the six elements that should be implemented in LAN networks in order to be more secure and one of these six elements is availability. This availability was the target of DDoS attacker in this research. In general, the aim of DDoS attacker when attacking a website is to crash the website and deny the user from browsing that website. However, the aim of attacking a LAN network is to deny legitimate users from accessing the system. Kiuchi et al. (2010) discussed the differentiations between DoS and DDoS attack in Section 2.5.3. However, in Chapter 2, Guangzhi et al. (2005) discussed the different types of DDoS attack in the LAN network. These two studies were the basis of performing the DDoS attack in this research. This research has been able to demonstrate that the information in both studies was shown workable, as the attacker succeeded in denying all users from accessing the system design and overextended the resources of the target machine.

In the literature review, Kassim and Sujitha (2013) mentioned that there are two methods that can be used to obtain users password: brute-force and Dictionary attacks. These two methods cause many security breaches in LAN networks and system security defence mechanisms. A Dictionary attack was used in this research in order to obtain the user's password and breach the security of the design system. This type of LAN network attack and the tool used were sufficient to obtain the password by compromising the target machine.

Based on the study of Qadeer et al. (2010) discussed in the literature review, the attacker can sniff the network packets by implementing tools that have the ability to record and analyse all packets traveling through the network. They highlighted that in a LAN network the attacker devices must be physically connected with the network of the victim, which is correct based on attempts that are used in this research. Deprived of physical connection, the tool used cannot listen to any packets that come in and out of the system. Since the connection was made, the attacker was able to listen for all communications between all devices in the system and others by using one of the listed tools in order to obtain information such as a password. This finding of the current research supports the previous study.

All scenarios, methods, and tools that were used in this research by the attacker were effective to compromise their target. The discussion of the results for all IDSs will be in the following sub-sections

5.2.1.2 Discussion of PADS results

Based on the comparative analysis in Chapter 4, PADS produced the lowest number of IDS generated alerts about intrusions that occurred in the network. What is surprising is that PADS did not discover and detect the most well known LAN network attacks at the time of this research. For example, the DDoS and Dictionary attacks are the most well known attacks, as discussed in Chapter 2, and PADS was unable to detect them and generate any alerts about them when they were performed in the designed system. PADS, as mentioned in Chapter 2, is a signature-based detection IDS. A possible explanation for misdetections may be that the database of PADS does not contain the pattern of these attacks. Misdetections of DDoS, Dictionary, and Packet Sniffing attack are unsatisfactory for effective security or a forensics capable system.

However, the alert generated for a port scan attack was sufficient and contained full information about the incidents. Especially, when PADS is linked with Sguil, this change the alert to become investigable and reliable and can be extracted. This extracted information about the attack attempts may be used as an evidence about the attacker. The ability of PADS to detect Reconnaissance attack was significant. It can thus be suggested that PADS can be used only to detect this type of attack.

5.2.1.3 Discussion of OSSEC Results

The comparative analysis in Chapter 4 shows the IDS of OSSEC has better results than PADS. OSSEC detected the Port Scanning and Dictionary attacks. However, OSSEC failed in detecting the DDoS and Packet Sniffing attack. These results were not very encouraging. This experiment did not detect any evidence for the capability of OSSEC to detect these two types of attack. A potential explanation for these results may be the lack of OSSEC's detection database to identify these two attacks. Misdetection of DDoS and Packet Sniffing is unacceptable for effective security requirements.

Nevertheless, the alerts generated about incidents that occurred in the system were straightforward and OSSEC sent emails about these incidents effectively. These two alert methods were used in this research as mentioned in Chapter 4. OSSEC alerts provide useful information about these two attacks including the source and distention IP, port number, the signature of the attack, level of attack and the type of attack. It could be argued that the positive results were due to the capability of OSSEC to identify these attacks immediately. The printed reports that were generated from Splunk when linked with OSSEC could be used as evidence to prosecute the attacker. These findings suggest that, in order to investigate OSSEC alerts, it is recommended to install Splunk and link it with the database of OSSEC. Another point is that Splunk in this research added a significant value to generated alerts of OSSEC and decreased the missing feature of OSSEC.

5.2.1.4 Discussion of Prelude results

As can been seen from the chart of the abilities of all IDSs in Figure 4.28 in Chapter 4, and sub-questions in Section 5.1, Prelude detected more attacks than PADS and OSSEC. These results are presented in Table 4.2 of comparative analysis in Chapter 4. However, Prelude did not detect the DDoS attack. A likely reason for this is that the detection mechanisms of Prelude is not capable of recognising this type of attack by itself.

One of the interesting findings is that the attack of Packet Sniffing was only detected by Prelude. Nevertheless, the information on this attack was incomplete for forensic investigations. For example, in a Port Scanning attack, Prelude provided full information about the attack including the IP for the attacker and the destination as well as the port number that was used in the attack plus the status of the attack and the number of times the attack had been attempted. Nonetheless, in Packet Sniffing the IP of the attacker was blank. Thus, the researcher presented this missing information in Table 4.2 of the comparative analysis as partially found. The results of this study show that Prelude detected and generated a greater number of alerts about LAN attacks than the other IDSs. Additionally, Prelude grouped these alerts based on the IP of attackers which is useful to identify the security breaches faster.

5.2.1.5 Summary

The final result from this experiment is that if the misdetection of DDoS is solved by Prelude developer teams, Prelude could then be recommended as the main IDSs to be used in LAN networks for a security system. This is because it has capability to detect all selected attacks. In addition, all alerts of Prelude were straightforward and contained all information that is required for a forensics investigator. Another point is that Prelude does not extra require software to be installed with it except Prewikka. Since, no previous study has conducted similar tests on PADS, OSSEC and Prelude, this is an important outcome for forensic network professionals.

These findings from all four selected LAN network attacks raise intriguing questions regarding the nature and extent of detecting others types of LAN network attacks by these three selected IDSs.

5.3 EVALUATIONS

It is important for any researcher or professional to reflect on their practice. Therefore, in the following sub-sections, the researcher will evaluate the research design that has been used in this experiment in Section 5.3.1. Next, the system design for the research will be evaluated in Section 5.3.2. Then, in Section 5.3.3, the evaluation of all IDSs tested will be discussed. Finally, in Section 5.3.4, the analysis tools that have been used in order to collect the findings and conduct the research will be evaluated.

5.3.1 Evaluation of Research Design

The research design contains five stages. This design was based on the methodology of the research that was adopted for investigation in LAN network incidents as discussed in Chapter 2. This methodology consists of nine steps starting from the preparation step to presentation. According to Pilli et al. (2010), the preparation step is the installed sensors in the network. The researcher changed this step into installations and a deploying step. The researcher has divided the preparation step into 5 steps in order to cover all experimental requirements and achieve the results. However, the installation of the software and IDSs sensors was moved into stage two. The experiment has been done in a lab computer environment. This situation led the researcher to create stage two in research design in order to be ready for the detection step.

The design system consists of different operating systems, software, packages and tools. Therefore, all these installations contributed making the first phase the longest. Accordingly, most of the software that was used in this experiment led to complications and created some minor technical challenges. These challenges forced the researcher to divide the preparation phase into five steps.

The system design consists of different operating systems in several standalone devices, thus the connectivity and management of these devices is essential. The modifications are not found in the adopted methodology. The researcher created phase two in order to ensure all packets can reach their destinations correctly. This led to creating five steps in phase two.

The phase three was starting with launching attacks in order to be ready to detect any malicious activity. The remaining phases were based on the adoption phase with a different name in order to ensure accuracy with the experimental activity.

Although this research involves the investigation of the ability of three selected IDS to detect four chosen network attacks the method was subject to continuous improvement as the testing proceeded. Thus, the research design presented in Chapter 3 was steadily improved. In addition, it assisted the researcher to accomplish the research for investigation in the test network for attackers and incidents based on the alarm of IDSs

5.3.2 System Design Evaluation

One of the most challenging phases in this study was the preparation phase. This was because there were many operating systems running and different IDSs were examined. It led to taking a longer time in order to determine suitable software, packages, and hardware that can help the researcher to conduct the acceptable and expected results without having any issues.

However, the most important step for the research was performing the four selected attacks one by one and observing the findings by seeing the IDSs alerts. This step needed more attention to these alerts because some IDSs failed to detect the attack of the LAN network. It had to be investigated more by analysing the recording of Argus to ensure that the connection was established and the attack was committed. Therefore, the third party analysis tools played an important role in this research in order to investigate the link between the attacker and the victim’s device. Other tools were also used to investigate the IDS alerts and collect the evidence.

Performing four selected attacks was the second important step. In this step, the researcher should select the proper tools that can compromise the target device. There are many tools available in the attacker’s computer and available on the Internet. Some of these tools are easy to use and others are not. Choosing the right tools might reduce the technical issues and assist in achieving the research objective.

5.3.3 Evaluation all IDSs

This section discusses all IDSs and highlights the most important advantages and disadvantages of each IDS based on the experiment results and use of these IDSs through conducting the research. These evaluations will produce some recommendations.

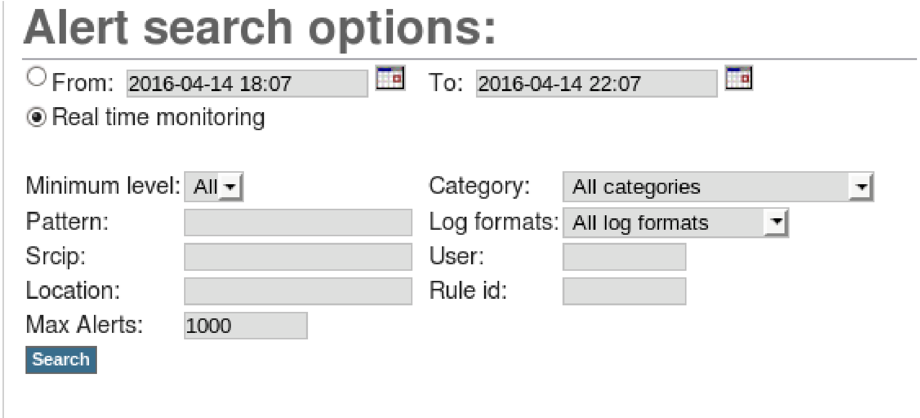
5.3.3.1 PADS

The installation of PADS is straightforward. It needs some minor modifications in the configuration of the PADS file in order to get it to work smoothly with the network layout and avoid any technical issue(s) that might impact the results. In order to investigate all the packets that are transferring in the LAN network, PADS cannot provide any details about the packets in addition to that shown in the alert. This is one of the disadvantages in using PADS as an IDS in a LAN network, since PADS generates alarms in the command base line, which is a very basic way to present them and not ideal to investigate these alerts forensically. It was very difficult to read these alerts one by one because some of them were repeated and it was not straightforward to investigate each incident.

These disadvantages led the researcher to install extra third party software, which was Sguil for presenting alarms in a graphical user interface (GUI) and grouping these alarms rather than to read repeated alarms. The PADS developer teams must work hard to solve these limitations and add more attack signatures. Additionally, they need to implement many things in order to provide more detail about the packets, and how to work without a GUI and presenting information of the alerts that are generated. PADS has advantages and disadvantages but the disadvantages are more significant than the advantages. Therefore, the evaluation of PADS is low, and not recommended for use by a system administrator in a LAN network, as it detected only one attack out of four attacks. The main disadvantage of PADS is it generated the alerts by itself in command line only.

5.3.3.2 OSSEC

The OSSEC installation is straightforward but it was the longest installation of an IDS in this research. This installation was divided into two steps. The first step is to install the IDS and the second one is to install the GUI. As aforementioned in Chapter 3, the installer must choose the installation type of OSSEC such as hybrid (server and agent at the same time), local, server or agent. If the installer chooses hybrid installation, as the researcher did in this research, the installation will repeat itself in order to install the IDS firstly to work as a server and then as an agent working simultaneously. This step must be combined in one step rather than repeated as the same steps. The step of entering the IP of the server should be a single entry. In fact, the installation of the GUI is more complex than the IDS itself. However, the installation of OSSEC IDS needs some modifications in the configure file. One of the most important things to change is the time of the attack detection. The OSSEC developers' team must either give the installer privileges/right to choose the time or make it shorter than 79200 seconds, which is 22 hours. This is a long time and not an acceptable time to monitor/investigate network activities. The advanced attackers in general need a few minutes in order to launch their attack and compromise the system but not hours. The GUI does not provide the export report function about the incidents that have been detected. Another point concerns the research panel in the GUI presentation for use. This feature is limited in the GUI because the researcher has to choose only from a drop menu the topic or the field that he/she wants to search, as shown in Figure 5.1. This limitation is not acceptable in investigation procedure about evidence in the LAN networks because there are more possible attacks on the LAN network than exist in the drop down menu. Furthermore, while the GUI does not provide any details or information about the level of the attack, the researcher spent a long on the Internet in order to find these levels.



Alert search options:

○ From: 2016-04-14 18:07 To: 2016-04-14 22:07
● Real time monitoring

Minimum level: All
Category: All categories
Pattern:
Log formats: All log formats
Srcip:
User:
Location:
Rule id:
Max Alerts: 1000

Figure 5.1: OSSEC search feature

For example, if the investigator would like to search for a specific attack that does not exist in the category, the search function is not useful in this case. In addition, the presented alerts in the main page of the GUI are not grouped, for example based on the attack name or the incident ID. This can be improved. These limitations led the researcher to install Splunk in order to have more space in searching and grouping these results based on the IP of the attacker. For example, Splunk has a fixing search for a known activity but also provides an advanced search bar in order to search for more activities that are complex.

The other disadvantage was that OSSEC failed in detecting a DDoS attack, which was unacceptable. After a long search on the Internet, the researcher found that the OSSEC developers wrote in the OSSEC website that Iplog tool will help to detect the DDoS attack, which is not installed in OSSEC. If the tool is useful in detecting DDoS, why it is not installed with OSSEC?

The OSSEC developers’ team must solve these limitations and make the GUI more dynamic for professional use. The main missing features are report exporting in different formats, providing more details about the packets and making the search feature more flexible. Furthermore, the level of attack should be available in the GUI.

The results of OSSEC showed that half of the attacks attempted were detected by OSSEC. However, missing features such as searching by a specific word or selected time make the evaluation of OSSEC low and it is not recommended. OSSEC IDS should not use as the main IDS in a LAN network.

5.2.2.3 Prelude

The Prelude installation is fairly straightforward. It was the easiest IDS to install into a computer, and that computer is recommended to use Red hat Linux as its operating system. The researcher faced some difficulties when installing Prelude in the Linux environment such as Ubuntu and Fedora 21 distribution and it was not functioning well. The researcher contacted the Prelude developer and discussed with them the difficulties and some other technical issues. They replied and mentioned that in order to have a full function of Prelude it is necessary to install it in a Red hat Linux environment (Prelude developer team, personal communication, Jan 10, 2016). The configurations as mentioned were not too many, and it is easy to do. The main things to enact were agent registration and the Prewikka as the GUI in order to see everything in Prelude. Without these registrations of agents, Prelude will not be able to detect any malicious activities. However, Prelude allows other software for IDS to register and works as an agent for Snort, PADS and OSSEC as well. This feature gives Prelude more advantages than the other IDSs used in this research.

Additionally, according to the communication between the researcher and Prelude developer teams about the issue of Prelude not detecting the DDoS attack, they said that the third party software “Suricata IDS” is required (Prelude developer team, personal communication, Jan 21, 2016). Suricata is a high performance Network IDS.

This is not good because the DDoS is one of the most well known attacks in the LAN networks as discussed in Chapter 2 and it is not detected by Prelude.

Prelude teams should add other important missing features such as the function of search and export for the findings. In addition, when the investigator examines the alerts, these alerts have the same colour, which make the investigation processes harder than it should be in order to know which attacks were successful and which were not. Thus, the colour of these must be different based on the status of the attack. This must be clear for any investigators to see inside the alert. Finally, in order to register the email of the network examiner to receive email about the incidents in the network, the examiner should configure the configuration file with the email server and this was not direct or simple to configure.

For security purposes, Prelude showed more significance than other IDSs with consideration for the misdetections of DDoS. Nevertheless, for forensics investigation, both OSSEC and Prelude have the same features and value. This was because Prelude failed to detect DDoS attack and the information on the Packet Sniffing attack was not complete.

5.3.4 Evaluation of Analysis Tools

A variety analysis third party tools (including open source tool) was utilised in this researcher in order to investigate the incidents of the LAN networks and to collect the research findings. These tools were Splunk, Squert, Sguil, and Argus. Some of these tools were used to assist in examining the connection between the victim's and the attacker's machine and to present the results. While, other tools were used to investigate all incidents that occurred in the design system and detected by IDSs. These tools worked in conjunction with the database of the IDSs and added more value to investigation process and presenting the findings. The following table highlights the main features that were used in this research in order to conduct the research and present the findings.

Table 5.7: Features and missing features of Analysis Tools

Tool Name	Features	Missing Features
Splunk	• Easy to install. • Flexible search function. • Exporting the results in different formats. • Monitors file systems. • Dynamic Web interface.	• No visualization (Location).

Tool Name	Features	Missing Features
Splunk	• Display log data	
Squert	• A visual tool. • Easy to install. • Used query to view events. • Dynamic Web interface. • Working in conjunction with Sguil database. • Providing more details about events.	• Working in conjunction with Sguil database. • Not work in Windows operating system direct.
Sguil	• Grouping the events. • Providing more information about incidents. • Work in conjunction with Squert. • Shows more details about incidents.	• Hard to configure • Link with PADS' database. • Not work in Windows operating system direct.
Argus	• Easy to install. • Record all packets. • Visual the record by commands. • Converting the record to be readable by other tools such as Wireshark.	• No user interface. • Hard to visual the record.

The most useful tool used in this research was Splunk. This software linked with OSSECs' database. This assisted in finding out the context of the attack in the designed network system more efficiently. Additionally, by determining the time, type of the attack and the IP of attacker in the search bar, the number of attempts was presented in a readable report, which helped to process the findings faster. The status of agents is presented in the main page, which helped the researcher to know which agents were active.

The second tool was Squert, which was useful as it has the ability to group the attacks and determine the location of the attacker(s) based on the IP, choose a specific time for searching, visualise the connections and many other features. This tool linked with the database of PADS as discussed in Chapter 3 and showed the results of the experiment effectively. This tool also has ability to show the signature of the attack and the port that has been used.

The third best tool was Sguil. This tool also linked with the PADSs' database in order to see alerts of PADS in a visual manner as discussed in Chapter 3. This tool has the ability to show more data about the attacker(s) such as IP resolution, which presents greater detail about the country and Internet provider for the attacker or victim as well and many other advantages. Additionally, the investigator can escalate the events for more investigation. These escalated events will be presented in the Squert main page, which adds more values for both tools.

The fourth tool was Argus. It is used mainly to record all network packets. The reason for this record was to ensure that there was a connection between the attacker and the experimental system components. It can extract information of packets that comes from the attacker. This recording helped the researcher to find more information about the connection of the attack between the attacker and the victim. The record can helpfully be converted to be readable by many network analysers such as Wireshark. This tool provides comprehensive details of connections between the experimental designed system devices and has the ability to present these connections visually. However, the commands to present these connections in graphic view are difficult to view and require a long time to read and practice. The visualizations of the connections during the attacks phase in this research is presented in Appendix 8.

5.4 RECOMMENDATIONS

The researcher, in this section, will discuss the effort needed in the field of the LAN network forensic investigation for successful management of LAN network intrusion events. The research was focused on the four selected attacks for a LAN network and the capability of the IDS to generate alerts about these attacks and then collect data for investigation. However, there is yet more study to do in this area as the following recommendations highlighted.

Firstly, this research was conducted in an isolated experimental design system of a LAN network lab environment with isolated machines. As discussed in the previous section, in order to collect more data from a LAN network intrusion incident, the evidence extracted from multiple IDS devices or components of LAN networks is required. Future research needs to prepare a small LAN network and obtain evidence from IDS components. However, the experimental operation must be more aligned with heavier network traffic. For example, although the security program running on the machine may affect the intrusion experiment, the network records it delivers would comprise of the evidential information for the intrusion incident. Additionally, the results of LAN network effects and affected fields on the machine system will have a more accurate result when examining such an environment if heavier traffic volumes are investigated.

Secondly, the methodology adopted in this research was developed by Pilli et al. (2010) and has been used in many studies such as the study of Nguyen et al. (2014). This model has been adopted in this research in order to proceed in a forensically trusted manner. This model contains useful steps to investigate the LAN network intrusion activities as well as review many tools that can extract the evidence from LAN networks. This methodology consists of nine steps; all of them can be applied easily in this research except the response to network/system administration steps. However, this step is useful but it is out of the scope of this research. This methodology is straightforward and covers all the required steps to investigate in this research area as well as giving suggested tools that may be used in order to extract the evidence. Additionally, it categorizes the tools in two varieties (commercial and open source) as well as highlighting the tools that can be used to extract evidence in different fields such as LAN networks and databases. As a result, this methodology provides much useful and value information, and is recommended to other researchers in this field.

The research results from the research design showed that common features and functionalities are common on the three digital forensic tools. The results have also revealed the different abilities for each IDS. The examination was conducted with the five stages defined in Figure 3.8. Based on testing and findings, the research design implemented in this research was shown to be practicable for digital forensic tools evaluation associated with the investigation of LAN network incidents.

Thirdly, additional research is needed for better knowledge of LAN networks methods and implementation techniques for applications. For instance, converting the raw digital information into useful evidence is the key in order to produce reliable data that can be used in a legal setting. However, a lack of detail during the forensic processes may impact the integrity of the evidence and prevent acceptance.

Finally, the method of comparative analysis used in Chapter 4 has assisted the researcher in realizing and evaluating the ability of each IDS. This technique helped the researcher to evaluate each IDS based on a number of pieces of evidence that can be gathered from LAN network activities when utilising IDS software. LAN network investigators can use this method during similar investigation processes.

5.5 CONCLUSION

Based on the results presented in Chapter 4, the findings have been discussed in Chapter 5. These findings answered the main research question as well as sub-questions and show the chosen three IDSs working together have the same result as when Prelude is working alone to detect four selected attacks on a LAN network. Thus, three selected IDSs working together did not detect more attacks than any one single IDS. Additionally, all sub-questions were answered including the discussion of each answer based on these findings and the table of comparative analysis.

In Section 5.2.1, a review of the evaluation of the research design was made. Then the discussion of findings was reviewed for more work and improvement. Suggestions were made to improve the functionality of the three IDS for advanced processing in LAN network forensic investigation aspects. Extra third party software was required in order to obtain the results and helped the researcher to extract the findings. A number of recommendations for future work were presented based on the discussion of the research results.

CHAPTER 6

Conclusion

6.0 INTRODUCTION

Chapter 6 concludes the thesis which has investigated the capability of three intrusion detection systems (IDSs). Understanding the ability of IDS's to provide reliable evidence can assist the forensic investigators to gather digital evidence from LAN network effectively in a forensic manner. This digital evidence can then be admissible to a court of law. In order to evaluate the abilities for each of the three selected IDSs, four chosen LAN network attacks were performed in a lab environment. The testing aimed to evaluate if the three IDSs working together in order to detect more malicious activities than any one single IDS could detect more than a single IDS.

The motivation and background for this research was discussed in Chapter 1. The review of literature in Chapter 2 highlighted a number of common LAN network attacks that creates more pressure on IDSs to detect these attacks and produce forensic evidence. These two chapters identify the importance of research into the network forensics area. The methodology and experimental devices were discussed in Chapter 3. Chapter 4 presented the findings of the research and produced a comparative analysis table based on the observations from the experiment phase. This analysis answers the main question of the research. These findings are then discussed in Chapter 5.

This chapter will provide an overview of the most significant aspects of the results. Section 6.1 will review the research while Section 6.2 will summarize the findings of the research. The limitations of the research will be discussed in Section 6.3. Section 6.4 will provide recommendations for future research. Finally, recommendations for forensic practice will be identified from the results of this research.

6.1 RESEARCH REVIEW

The literature review provides a comprehensive overview of the knowledge and current issues of using IDS to detect intrusions in a LAN networks. This chapter started by reviewing network security by giving the overview and its objectives. Followed by a section that provides a description of common LAN network attacks in the area. After that the discussion of digital forensic, network forensics and network investigation life cycle were made. Finally, the last two sections described the fundamental of IDS components and how they work, and some current issues from using these IDSs to detect incidents in LAN networks.

The literature review was conducted to present a comprehensive review in precise areas that are essential to be tested for further research. The security attacks and threats toward LAN network devices within have a high importance for rational investigation. All studies and information discussed in the literature review have presented essential facts that led the researcher to design a logical research methodology. Therefore, the proposed research design focuses on forensically gathering the evidence from the intrusion detection systems in form of their detection alerts. Unequivocally, the research assisted to collect the traffic of a wired network that was packets coming and outgoing from the proposed system' devices. The main source of evidence and those packets were typically collected by an intrusion detection system for more analysis on the information of LAN networks attacks.

Chapter 3 has provided comprehensive information about the methodology of the research. It was used to analyse each portion of the target field of gathering evidence from IDS and also the acquiring and preserving of traffic of the LAN networks. Issues of IDSs discussed in Chapter 2 have assisted the researcher to develop the main question of the research as well as sub-questions. The Network investigation life cycle in Section 2.8, and the issues of IDSs discussed in Section 2.11 worked in conjunction to adopt a testing methodology and recommended tools. The proposed system design' components were functioning in a real world environment for small LAN networks. The methods of collecting, acquiring and preserving traffic of LAN networks were supported by other studies that discussed the priority of these steps and avoided any mishandling that might influence the result. Lastly, a design of the experimental system was structured in order to cover components and architecture of the system. This was followed by discussing the limitations that can be identified in the methodology.

Once the experimental phase was accomplished, Chapter 4 presented the analysis, reporting and the presentation of the results obtained from examination. In order to have the validity and integrity of the research results some alterations were made. One of the changes was examining all IDSs simultaneously so that all IDSs face the same packets number and attack technique rather than just testing each one of them individually. The experiment was conducted on isolated computers in a Lab environment. The thesis' stages were divided into five phases worth each of them consisting of several steps as shown Figure 3.1. The extra third party tools played an important role in this research in order to investigate the link between the attacker and the victims' device. Other tools also used to investigate the IDS' alerts and collect the evidence.

In this experiment, most of the IDS presented some important benefits that could detect well-known attacks and capture live traffic. These benefits were presented in a visual method. In order to investigate forensically, the evidence was copied and transferred to the forensic server in order to extract the evidence while all IDSs were still monitoring the LAN network traffic to detect other malicious activities. The hash calculation was applied to all tested alerts of IDSs as well as logs to ensure the copying and transferring of data does not change or alter the evidence's integrity.

The discussions of the findings are carried in Chapter 5 based on the results presented in Chapter 4 in the comparative analysis table. The main research question as well as the sub related questions, are answered in this chapter. The recommendations and issues found in the experiments were presented in order to provide knowledge for similar research and the future research.

The main question works in conjunction with sub-questions and was used to discover IDS' capabilities that assist the investigation in LAN network forensics. The main question of this research has provided knowledge for the network forensic processes. The proposed system was able to detect huge amounts of LAN network' traffic. Additionally, this design system was capable of acquiring evidence from four selected attacks. Regardless of the fact that all the used IDSs in this experiment are capable of acquiring and preserving some trace of the LAN network' attacks, and they have some noted limitations associated with detection of these attacks.

The conclusion of the research is presented in Chapter 6, which includes the significant points of the research and the potential for more research projects. It will include the summary of the research findings observed and obtained from the whole experimental process. The research' limitations are discussed and recommendations for further work in future be presented.

6.2 SUMMARY OF RESEARCH RESULTS

The main function of IDSs are to detect malicious activities in a network. They place at a strategic point(s) within the network to monitor traffic in and out form all machines on the network. Once an attack is detected, the alert can be sent to the network and/or security administrator. The ability of each IDS is different from another and there are many types of IDSs as discussed in Chapter 2.

The main goal of the research was to examine the ability of three IDSs in a LAN network to detect different well-known types of LAN network attacks. In order to achieve the goal of the research, the system design should be created firstly and all software and IDSs installed and then the connectivity should be tested in order to be ensure that all packets can reach their destination. Each IDS has different installations methods and some of them required a specific packages and software to be installed. Some of IDSs alone has limitations but some limitations can be overcome by installing third party software to enhance ease of use and functionality. This testing created in a lab environment. Consequently, the results of the experiment of the research were mainly based on the inspection of this information.

The findings of PADS showed its ability to detect the Port Scanning attack. In order to investigate more about this detection, analysis third party software is required such as Squert and Sguil. Nevertheless, PADS is used by some distributed detection systems such as Security Onion Distribution which is a Network Security Monitoring distribution. However, these systems used PADS as an assistant tool but not a main IDS to detect malicious activities.

In contrast, OSSEC IDS offered more alerts than PADS because it did detect Port Scanning and Dictionary attacks. The ability to generate alerts was significant. However, in order to search, export and investigate evidence forensically, Splunk (third party analysis software) is required.

Finally, Prelude presented more results than PADS and OSSEC in this experiment. These results provided more information about detecting the Port Scanning, Dictionary and Packet Sniffing attack. However, the information on Packet Sniffing was not complete, because the Internet Protocol of the attacker was missing. Thus, this alarm was useless in order to find evidence about the attacker but it was useful to show the capability of detecting the attack.

In summary, installing numerous IDSs in a LAN network may giving a feeling of enhanced security. This testing is vital because every IDS has different approaches to detect malicious activities. The results of the research showed that all selected IDSs failed to detect DDoS attack, therefore, the three selected IDSs working together did not detect more attacks than any one single IDS and this may not increase security at all.

6.3 LIMITATIONS

The evaluation in Chapter 3 helped the researcher to reduce a number of possible limitations in the study. It was used as a basis of the designed system and the research model. There were six limitations discussed before starting the experiment. The first limitation described the location of IDS and firewall in LAN networks. The second one different types of IDS that were used to detect the LAN networks intrusion. The third limitation is the amount of the data that must be collected and analysed. The fourth issue discussed the various ways to design and configure the architecture of the network system. The fifth limitation was technical issues that might face the researcher when installing the proposed system. The last limitation discussed the difficulties of applying all types of LAN network attacks. However, there were two limitations during the experimental phase that were also realized and will be discussed in this section. However, two limitations remain.

The scope of this research was to examine four selected well-known attacks of the LAN networks with three IDSs in an experimental scenario. Four type of attacks are not enough to evaluate the capability of the three selected IDSs. For example, DDoS attack is one of the most well-known attacks in LAN networks; TCP SYN attack been used in this research is a type of DDoS, and all IDSs failed to detect this attack. However, it may be possible the selected IDSs can detect other types of DDoS attacks. Since the purpose of this research is to identify and evaluate the capability of three IDSs to produce evidence with the four developed examination scenarios, no further examination scenario was added. Therefore, there is a limitation in identifying and evaluating the detection capability of the three IDSs for all possible types of LAN network attacks.

The second limitation was when transferring all alerts (evidence) from IDS device to Forensic server device while the IDS device is still working. There was an issue that relate to transfer the evidence in a proper way without impact the integrity of the evidence and not stop the IDS device from working. The researcher created a scripts and used as a transfer tool that has the ability to transfer the data. This tool compressed the IDS data firstly and then generated the calculation of hash value for the data before the transferring in order to ensure the copying and transferring of data does not alter or modify the evidence's integrity. After that transferred the data to Forensic server for analysing. The limitation was that if the volume of the network traffic is high and/or more IDS devices, the copying and transferring will work properly without loss and delay issues.

The third limitation was all IDSs used in this research are open source. These IDSs are developed and supported by vendors. However, the author feels that the commercial IDSs may be able to detect the four selected attacks and produce more reliable evidence than the IDSs used in this research. This limitation made the discussion of the results is limited. The limitations recognized in this research will provide new directions for further study, highlighted in the next section.

6.4 RECOMMENDATIONS FOR FUTURE RESEARCH

Chapter 4 provided findings of the research while Chapter 5 produced the discussion of these results. When combining these two chapters' knowledge of the chosen field of using intrusion detection systems is found. It is sufficient to produce a source of forensic evidence for a LAN network. The main concern of this research was applying a number of IDSs with the aim of obtaining effective digital evidence. The outcomes of this research has led to a number of new ideas for future research.

First of all, the plan of this experiment was created to examine and analyze these IDSs and suggest how to improve the collection of digital forensic evidence with these tests. The experimental procedure was to install a number of IDSs in a LAN network system. Regardless of the number of IDSs, the concept can be applied in studying any type of computer network attack. Thus, this experiment should be extended to study other types of attacks on LAN networks.

Another point is that there are many Security Information and Event Management applications in common use, such as, the Security Onion distribution that provides analysis of security detection alerts in real-time generated by network software and hardware. The main goal of these systems is to produce significant evidence. Log files and events are crucial sources for gathering digital evidence. Additionally, some of these distributions are open source and can be used for future research. New research needs to be done on these systems to evaluate them in terms of their detection of attacks and how to improve these systems, in order to collect forensic evidence.

Finally, MySQLdump has been used to transfer the evidence obtained from databases of the IDSs and zip for log files of the proposed system into the forensic server for more investigations with regard to the calculation of hash value for the files of log and databases. Nevertheless, this MySQLdump cannot be considered as a trusted forensic tool to transfer the evidence gained without a forensic study. Thus, there should be new research to test this tool and consider it as a forensic tool for the purpose of evidence collection. This tool must consider the integrity and validity of the evidence during the transformation.

6.5 RECOMMENDATIONS FOR PRACTICE

Three IDSs (PADS, OSSEC, and Prelude) were used in this experiment. Prelude produced reliable evidence and detect more attacks than others detect. This IDS is compatible and supportable in the Red Hat environment only. The installations and configurations are straightforward. However, Prelude failed to detect DDoS attack in this experiment but it may able to detect by using other agents. Prelude has some shortcomings such as the feature of exporting alerts via the graphical user interface of Prewikka. However, Prelude teams suggest that Prelude can export all/any alert(s) by issuing SQL commands. Therefore, it is recommended to use this as the main IDS in a LAN network.

A diversity of analysis tools (Splunk, Squert, Sguil and Argus) were used in this research. Splunk has the ability to convert a huge of raw data into a readable information in the form of tables. Therefore, this tool can reduce the investigation process time by categorized the findings based on custom research queries. Splunk presents the results of the investigation in an effective way. Consequently, it is recommended to use this tool by LAN network forensic investigators if it is possible.

REFERENCES

Adelstein, F. (2006). Live forensics: diagnosing your system without killing it first. *Commun. ACM, 49*(2), 63-66. doi:10.1145/1113034.1113070

Akhgar, B., Saathoff, G. B., Arabnia, H. R., Hill, R., Staniforth, A., & Bayerl, P. S. (2015). *Application of Big Data for National Security: A Practitioner's Guide to Emerging Technologies*: Elsevier Science. Retrieved from http://books.google.co.nz/books?id=ZPOcBAAAQBAJ

Al-Jarrah, O., & Arafat, A. (2014). Network Intrusion Detection System using attack behaviour classification Symposium conducted at the meeting of the 2014 5th International Conference on Information and Communication Systems (ICICS), doi:10.1109/IACS.2014.6841978

Alkhishali, H. H., Abou El Farag, A., & El Baith Mohamed, A. (2010). Design and implementation of portable, multi mode of operation embedded firewall Symposium conducted at the meeting of the 2010 32nd International Conference on Information Technology Interfaces (ITI).

Allen, W. H., Marin, G. A., & Rivera, L. A. (2005). Automated detection of malicious Reconnaissance to enhance network security Symposium conducted at the meeting of the 2005. Proceedings. IEEE SoutheastCon, doi:10.1109/SECON.2005.1423286

Almulhem, A. (2009). Network forensics: Notions and challenges*IEEE.* Symposium conducted at the meeting of the 2009 IEEE International Symposium on Signal Processing and Information Technology (ISSPIT),

Altaher, A., Ramadass, S., & Ali, A. (2011). A dual stack IPv4/IPv6 testbed for malware detection in IPv6 networks Symposium conducted at the meeting of the 2011 IEEE International Conference on Control System, Computing and Engineering (ICCSCE), doi:10.1109/ICCSCE.2011.6190516

Bace, R., & Mell, P. (2001). NIST special publication on intrusion detection systems: DTIC Document.

Berghel, H. (2003). The discipline of Internet forensics. *Communications of the ACM, 46*(8), 15-20.

Bishop, M. (2003). What is computer security? *Security & Privacy, IEEE, 1*(1), 67-69. doi:10.1109/MSECP.2003.1176998

Bryant, S., & Bryant, R. (2014). *Policing Digital Crime*: Ashgate Publishing, Limited.

Casey, E. (2004). Network traffic as a source of evidence: tool strengths, weaknesses, and future needs. *Digital Investigation, 1*(1), 28-43. doi:http://dx.doi.org/10.1016/j.diin.2003.12.002

Casey, E. (2011). Digital evidence and computer crime: forensic science, computers and the internet: Academic press.

Casey, E., & Stanley, A. (2004). Tool review – remote forensic preservation and examination tools. *Digital Investigation, 1*(4), 284-297. doi:http://dx.doi.org/10.1016/j.diin.2004.11.003

Chang, E. S., Jain, A. K., Slade, D. M., & Tsao, S. L. (1999). Managing cyber security vulnerabilities in large networks. *Bell Labs Technical Journal, 4*(4), 252-272. doi:10.1002/bltj.2202

Chasaki, D., & Wolf, T. (2012). Attacks and Defenses in the Data Plane of Networks. *IEEE Transactions on Dependable and Secure Computing, 9*(6), 798-810. doi:10.1109/TDSC.2012.50

Chen, T. M. (2010). Stuxnet, the real start of cyber warfare? [Editor's Note]. *Network, IEEE, 24*(6), 2-3. doi:10.1109/MNET.2010.5634434

Chen, T. M., & Abu-Nimeh, S. (2011). Lessons from Stuxnet. *Computer, 44*(4), 91-93. doi:10.1109/MC.2011.115

Chia-Mei, C., Han-Wei, H., Peng-Yu, Y., & Ya-Hui, O. (2013). Defending malicious attacks in Cyber Physical Systems Symposium conducted at the meeting of the 2013 IEEE 1st International Conference on Cyber-Physical Systems, Networks, and Applications (CPSNA), doi:10.1109/CPSNA.2013.6614240

Cloudflare.com. (2015). *Affordable advanced DDoS protection and mitigation | CloudFlare | The web performance & security company.* Retrieved 23 April, 2015, from https:// www.cloudflare.com

Convery, S. (2004). *Network security architectures*: Pearson Education India.

Dabbagh, M., Ghandour, A. J., Fawaz, K., Hajj, W. E., & Hajj, H. (2011). Slow Port Scanning detection Symposium conducted at the meeting of the 2011 7th International Conference on Information Assurance and Security (IAS), doi:10.1109/ISIAS.2011.6122824

Daley, K., Larson, R., & Dawkins, J. (2002). A structural framework for modeling multi-stage network attacks Symposium conducted at the meeting of the International Conference on Parallel Processing Workshops, 2002. Proceedings. doi:10.1109/ICPPW.2002.1039705

Dattani, M., Thanthry, N., Best, T., Bhagavathula, R., & Pendse, R. (2004). Route optimized nested mobility solution using PAT Symposium conducted at the meeting of the 2004 IEEE 60th Vehicular Technology Conference, 2004. VTC2004-Fall. doi:10.1109/vetecf.2004.1404633

Eiland, E. E., Evans, S. C., Markham, T. S., Barnett, B., Impson, J., & Steinbrecher, E. (2008). Network Intrusion Detection: Using MDLcompress for deep packet inspection Symposium conducted at the meeting of the IEEE Military Communications Conference, 2008. MILCOM 2008. doi:10.1109/MILCOM.2008.4753180

Fink, G. A., North, C. L., Endert, A., & Rose, S. (2009). Visualizing cyber security: Usable workspaces Symposium conducted at the meeting of the 6th International Workshop on Visualization for Cyber Security, 2009. VizSec 2009. doi:10.1109/VIZSEC.2009.5375542

Fosic, I., & Zagar, D. (2011). VPN network protection by IDS system implementation Symposium conducted at the meeting of the MIPRO, 2011 Proceedings of the 34th International Convention

Gadge, J., & Patil, A. A. (2008). Port scan detection Symposium conducted at the meeting of the 16th IEEE International Conference on Networks, 2008. ICON 2008. doi:10.1109/ICON.2008.4772622

Garfinkel, S. (2002). *Network Forensics: Tapping the Internet - O'Reilly Media*. Retrieved 30 April, 2015, from http://archive.oreilly.com/pub/a/network/2002/04/26/nettap.html

Geer, D. (2005). Malicious bots threaten network security. *Computer, 38*(1), 18-20. doi:10.1109/MC.2005.26

Goodhue, D. L., & Straub, D. W. (1991). Security concerns of system users: A study of perceptions of the adequacy of security. *Information & Management, 20*(1), 13-27. doi:http://dx.doi.org/10.1016/0378-7206(91)90024-V

Guangzhi, Q., Hariri, S., & Yousif, M. (2005). Multivariate statistical analysis for network attacks detection Symposium conducted at the meeting of the 2005. The 3rd ACS/IEEE International Conference on Computer Systems and Applications, doi:10.1109/AICCSA.2005.1387011

Guo, Y., Slay, J., & Beckett, J. (2009). Validation and verification of computer forensic software tools—Searching Function. *Digital Investigation, 6, Supplement*(0), S12-S22. doi:http://dx.doi.org/10.1016/j.diin.2009.06.015

Hales, G. A., Ferguson, R. I., & Archibald, J. M. (2013). On the use of data visualization techniques to support digital forensic analysis: A survey of current approaches.

Hanley, M., & Montelibano, J. (2011). Insider threat control: Using centralized logging to detect data exfiltration near insider termination: DTIC Document.

Heberlein, L. T., Dias, G. V., Levitt, K. N., Mukherjee, B., Wood, J., & Wolber, D. (1990). A network security monitor Symposium conducted at the meeting of the Proceedings. 1990 IEEE Computer Society Symposium on Research in Security and Privacy, 1990. doi:10.1109/RISP.1990.63859

Holste, M. (2015). *Enterprise log search and archive (ELSA)*. Retrieved 1 October, 2015, from https://code.google.com/p/enterprise-log-search-and-archive/

Hopper, L., Hopper, R., & Womble, P. (2009, 11-12 May 2009). Identifying network attacks from a social perspective Symposium conducted at the meeting of the Conference on Technologies for Homeland Security, 2009. HST '09. IEEE doi:10.1109/THS.2009.5168080

Hoque, N., Bhuyan, M. H., Baishya, R. C., Bhattacharyya, D. K., & Kalita, J. K. (2014). Network attacks: Taxonomy, tools and systems. *Journal of Network and Computer Applications*, *40*(0), 307-324. doi:http://dx.doi.org/10.1016/j.jnca.2013.08.001

Hori, Y., Nishide, T., & Sakurai, K. (2011, Nov. 30 2011-Dec. 2 2011). Towards Countermeasure of Insider Threat in Network Security Symposium conducted at the meeting of the 2011 Third International Conference on Intelligent Networking and Collaborative Systems (INCoS), doi:10.1109/INCoS.2011.156

Hosmer, B. (2012). Getting started with salt stack--the other configuration management system built with python. *Linux J., 2012*(223), 3.

Hunt, R., & Slay, J. (2011). A new approach to developing attack taxonomies for network security - including case studies Symposium conducted at the meeting of the 17th IEEE International Conference on Networks (ICON), 2011 doi:10.1109/ICON.2011.6168489

ISO. (2012) ISO/IEC 27037:2012 *Information Technology – Security Techniques – Guidelines for Identification, Collection, Acquisition, and Preservation of Digital Evidence*, International Organization for Standardization, Geneva.

Jiang, L., Tian, G., & Zhu, S. (2012). Design and Implementation of Network Forensic System Based on Intrusion Detection Analysis Symposium conducted at the meeting of the 2012 International Conference on Control

Engineering and Communication Technology (ICCECT), doi:10.1109/ICCECT.2012.51

Kassim, M. M., & Sujitha, A. (2013). ProcurePass: A User Authentication Protocol to Resist Password Stealing and Password Reuse Attack Symposium conducted at the meeting of the 2013 International Symposium on Computational and Business Intelligence (ISCBI), doi:10.1109/ISCBI.2013.14

Keramati, M., & Keramati, M. (2014). Novel security metrics for ranking vulnerabilities in computer networks Symposium conducted at the meeting of the 2014 7th International Symposium on Telecommunications (IST), doi:10.1109/ISTEL.2014.7000828

Kiuchi, T., Hori, Y., & Sakurai, K. (2010). A Design of History Based Traffic Filtering with Probabilistic Packet Marking against DoS Attacks Symposium conducted at the meeting of the 2010 10th IEEE/IPSJ International Symposium on Applications and the Internet (SAINT), doi:10.1109/SAINT.2010.44

Kizza, & Migga, J. (2009). *A guide to computer network security*: Springer.

Klevinsky, T. J., Laliberte, S., & Gupta, A. (2002). *Hack IT: security through penetration testing*: Addison-Wesley Professional.

Kothari, C. (2004). *Research methodology: Methods and techniques*: New Age International.

Lazarevic, A., Kumar, V., & Srivastava, J. (2005). Intrusion detection: A survey. In *Managing Cyber Threats* (pp. 19-78): Springer.

Lesk, M. (2011). Cybersecurity and Economics. *IEEE Security & Privacy, 9*(6), 76-79. doi:10.1109/MSP.2011.160

MaCfee, M. (2015). Net Losses: Estimating the Global Cost of Cybercrime. MaCfee. Retrieved 19 April 2015, from http://www.mcafee.com/ca/resources/reports/rp-economic-impact-cybercrime2.pdf

Maggi, P., Pozza, D., & Sisto, R. (2008). Vulnerability Modelling for the Analysis of Network Attacks Symposium conducted at the meeting of the Third International Conference on Dependability of Computer Systems, 2008. DepCos-RELCOMEX '08. doi:10.1109/DepCoS-RELCOMEX.2008.49

Mana, A., & Munoz, A. (2006). Protected Computing vs. Trusted Computing Symposium conducted at the meeting of the First International Conference on

Communication System Software and Middleware, 2006. Comsware 2006. doi:10.1109/COMSWA.2006.1665152

McRee, R. (2008). Expanding Response: Deeper Analysis for Incident Handlers. SANS Institute.

Mendyk-Krajewska, T., & Mazur, Z. (2010). Problem of network security threats Symposium conducted at the meeting of the Human System Interactions (HSI), 2010 3rd Conference on doi:10.1109/HSI.2010.5514533

Mercuri, R. T. (2003). Analyzing security costs. *Commun. ACM*, *46*(6), 15-18. doi:10.1145/777313.777327

Ming-Hung, W., Chia-Ming, Y., Chia-Liang, L., Chien-Chao, T., & Li-Hsing, Y. (2014). KPAT: A kernel and protocol analysis tool for embedded networking devices Symposium conducted at the meeting of the 2014 IEEE International Conference on Communications (ICC), doi:10.1109/ICC.2014.6883478

Mohay, G. (2005). Technical challenges and directions for digital forensics Symposium conducted at the meeting of the 2005. First International Workshop on Systematic Approaches to Digital Forensic Engineering, doi:10.1109/SADFE.2005.24

Morris, T., Vaughn, R., & Dandass, Y. (2012). A Retrofit Network Intrusion Detection System for MODBUS RTU and ASCII Industrial Control Systems Symposium conducted at the meeting of the 2012 45th Hawaii International Conference on System Science (HICSS), doi:10.1109/HICSS.2012.78

Mukkamala, & Sung. (2003). Identifying significant features for network forensic analysis using artificial intelligent techniques. *International Journal of digital evidence, 1*(4), 1-17.

Mukkamala, S., Janoski, G., & Sung, A. (2002). Intrusion detection using neural networks and support vector machines Symposium conducted at the meeting of the Proceedings of the 2002 International Joint Conference on Neural Networks, 2002. IJCNN '02. doi:10.1109/IJCNN.2002.1007774

NCSC Incident Response | NCSC. (2016). *Ncsc.govt.nz.* Retrieved 24 May 2016, from http://www.ncsc.govt.nz/incidents/

Nguyen, K., Tran, D., Ma, W., & Sharma, D. (2014). An approach to detect network attacks applied for network forensics Symposium conducted at the meeting of the Fuzzy Systems and Knowledge Discovery (FSKD), 2014 11th International Conference on doi:10.1109/FSKD.2014.6980912

NST. (2015). *Network Security Toolkit (NST)*. Retrieved 1 October, 2015, from http://www.networksecuritytoolkit.org/nst/index.html

Onwubiko, C., & Lenaghan, A. P. (2007). Managing Security Threats and Vulnerabilities for Small to Medium Enterprises Symposium conducted at the meeting of the Intelligence and Security Informatics, 2007 IEEE doi:10.1109/ISI.2007.379479

Ossec.net. (2015). Home — OSSEC. Retrieved 3 July 2015, from http://ossec.net

PADS. (2015). *Passive Asset Detection System*. Retrieved 29-09-2015, from http://passive.sourceforge.net/index.php

Palmer, G. (2001). A Road Map for Digital Forensic Research, In First Digital Forensic Research Workshop, Utica, New York. (pp. 27-30).

Petersen, R. R., & Wiil, U. K. (2011). CrimeFighter Investigator: A Novel Tool for Criminal Network Investigation Symposium conducted at the meeting of the 2011 European Intelligence and Security Informatics Conference (EISIC), doi:10.1109/EISIC.2011.55

Petroni, N. L., Walters, A., Fraser, T., & Arbaugh, W. A. (2006). FATKit: A framework for the extraction and analysis of digital forensic data from volatile system memory. *Digital Investigation, 3*(4), 197-210. doi:http://dx.doi.org/10.1016/j.diin.2006.10.001

Pfleeger, S. L., & Stolfo, S. J. (2009). Addressing the Insider Threat. *Security & Privacy, IEEE, 7*(6), 10-13. doi:10.1109/MSP.2009.146

Pilli, E. S., Joshi, R. C., & Niyogi, R. (2010). Network forensic frameworks: Survey and research challenges. *Digital Investigation, 7*(1–2), 14-27. doi:http://dx.doi.org/10.1016/j.diin.2010.02.003

Prelude, (2015). Retrieved 3 August 2015, from https://www.prelude-siem.org/

Qadeer, M. A., Zahid, M., Iqbal, A., & Siddiqui, M. R. (2010). Network Traffic Analysis and Intrusion Detection Using Packet Sniffer Symposium conducted at the meeting of the 2010. ICCSN '10. Second International Conference on Communication Software and Networks, doi:10.1109/ICCSN.2010.104

Rahman, R. (2012). Legal jurisdiction over malware-related crimes1: From theories of jurisdiction to solid practical application. *Computer Law & Security Review, 28*(4), 403-415. doi:http://dx.doi.org/10.1016/j.clsr.2012.03.004

Rekhis, S., & Boudriga, N. (2012). A System for Formal Digital Forensic Investigation Aware of Anti-Forensic Attacks. *IEEE Transactions on*

Information Forensics and Security, 7(2), 635-650. doi:10.1109/TIFS.2011.2176117

Ritchey, R. W., & Ammann, P. (2000). Using model checking to analyse network vulnerabilities Symposium conducted at the meeting of the IEEE Symposium on Security and Privacy, 2000. S&P 2000. Proceedings. 2000 doi:10.1109/SECPRI.2000.848453

Robledo, H. F. G. (2008). Types of Hosts on a Remote File Inclusion (RFI) Botnet Symposium conducted at the meeting of the CERMA '08 Electronics, Robotics and Automotive Mechanics Conference, 2008. doi:10.1109/CERMA.2008.60

Rod, M., & KPMG. (2002). Options in Computer Forensic Tools. *Computer Fraud & Security, 2002*(11), 8-11. doi:http://dx.doi.org/10.1016/S1361-3723(02)01108-9

Russell, D., & Gangemi, G. (1991). *Computer security basics*: "O'Reilly Media, Inc.".

Sallhammar, K., Helvik, B. E., & Knapskog, S. J. (2005). Incorporating Attacker Behavior in Stochastic Models of Security Symposium conducted at the meeting of the Security and Management

SecurityWeek. (2015). *Japan Sees 25 billion Cyberattacks in 2014: Govt Agency | SecurityWeek.Com*. Retrieved 14 April, 2015, from http://www.securityweek.com/japan-sees-25-billion-cyberattacks-2014-govt-agency

Selamat, S. R., Yusof, R., Sahib, S., Hassan, N. H., Abdollah, M. F., & Abidin, Z. Z. (2011). Traceability in digital forensic investigation process Symposium conducted at the meeting of the 2011 IEEE Conference on Open Systems (ICOS), doi:10.1109/ICOS.2011.6079259

Shaikh, S. A., Chivers, H., Nobles, P., Clark, J. A., & Chen, H. (2008). Network Reconnaissance. *Network Security, 2008*(11), 12-16. doi:http://dx.doi.org/10.1016/S1353-4858(08)70129-6

Shi, Z. (2011). The automaton modelling of typical network attacks Symposium conducted at the meeting of the 2011 IEEE International Conference on Computer Science and Automation Engineering (CSAE), doi:10.1109/CSAE.2011.5953213

Shun, J., & Malki, H. A. (2008). Network Intrusion Detection System Using Neural Networks Symposium conducted at the meeting of the 2008. ICNC '08.

Fourth International Conference on Natural Computation, doi:10.1109/ICNC.2008.900

Sibiya, G., Venter, H. S., Ngobeni, S., & Fogwill, T. (2012). Guidelines for procedures of a harmonised digital forensic process in network forensics Symposium conducted at the meeting of the Information Security for South Africa (ISSA), 2012 doi:10.1109/ISSA.2012.6320451

Sivaprasad, A., & Jangale, S. (2012). A complete study on tools & techniques for digital forensic analysis Symposium conducted at the meeting of the 2012 International Conference on Computing, Electronics and Electrical Technologies (ICCEET), doi:10.1109/ICCEET.2012.6203877

Spafford, B., & Carrier, E. H. (2003). Getting Physical with the Digital Investigation Process. *International Journal of digital evidence, 2*(2).

Squert. (2015). *Squert Project* Retrieved 01-10, 2015, from http://www.squertproject.org/

Srivastava, A., Gupta, B. B., Tyagi, A., Sharma, A., & Mishra, A. (2011). A Recent Survey on DDoS Attacks and Defense Mechanisms (pp. 570-580). Berlin, Heidelberg: Springer Berlin Heidelberg. Retrieved from http://dx.doi.org/10.1007/978-3-642-24037-9_57. doi:10.1007/978-3-642-24037-9_57

Stallings, W. (2011). Cryptography and Network Security: Principles and Practice: Prentice Hall.

Sudin, S., Tretiakov, A., Ali, R. H. R. M., & Rusli, M. E. (2008). Attacks on mobile networks: An overview of new security challenge Symposium conducted at the meeting of the International Conference on Electronic Design, 2008. ICED 2008. doi:10.1109/ICED.2008.4786772

Unitec.ac.nz. (2015). *Cyber security in New Zealand moves forward | Unitec*. Retrieved 16 April, 2015, from http://www.unitec.ac.nz/about-us/cyber-security-in-new-zealand-moves-forward

Valjarevic, A., & Venter, H. S. (2012). Harmonised digital forensic investigation process model Symposium conducted at the meeting of the Information Security for South Africa (ISSA), 2012 doi:10.1109/ISSA.2012.6320441

Valjarevic, A., Venter, H. S., & Ingles, M. (2014). Towards a prototype for guidance and implementation of a standardized digital forensic investigation process Symposium conducted at the meeting of the Information Security for South Africa (ISSA), 2014 doi:10.1109/ISSA.2014.6950488

Verma, K., Hasbullah, H., & Kumar, A. (2013). An efficient defense method against UDP spoofed flooding traffic of denial of service (DoS) attacks in VANET Symposium conducted at the meeting of the 2013 IEEE 3rd International Advance Computing Conference (IACC), doi:10.1109/IAdCC.2013.6514286

Wang, Z., & Wang, X. (2008). NetFlow Based Intrusion Detection System Symposium conducted at the meeting of the 2008. MMIT '08. International Conference on MultiMedia and Information Technology, doi:10.1109/MMIT.2008.213

Wazid, M., Katal, A., Goudar, R. H., & Rao, S. (2013). Hacktivism trends, digital forensic tools and challenges: A survey Symposium conducted at the meeting of the 2013 IEEE Conference on Information & Communication Technologies (ICT), doi:10.1109/CICT.2013.6558078

Whitaker, A. J., Valentine, M., & Whitaker, A. (2008). *CCNA: Exam 640-802*: Que Pub. Retrieved from https://books.google.co.nz/books?id=hJ-OF2w5iOMC

Wiil, U. K. (2013). Issues for the Next Generation of Criminal Network Investigation Tools Symposium conducted at the meeting of the 2013 European Intelligence and Security Informatics Conference (EISIC), doi:10.1109/EISIC.2013.9

Wilhelm, T. (2009). *Professional Penetration Testing: Volume 1: Creating and Learning in a Hacking Lab*: Elsevier Science. Retrieved from https://books.google.co.nz/books?id=AcscdZ6Bs40C

Wolf, T., Chandrikakutty, H., Hu, K., Unnikrishnan, D., & Tessier, R. (2014). Securing Network Processors with High-Performance Hardware Monitors. *Transactions on Dependable and Secure Computing, IEEE, PP*(99), 1-1. doi:10.1109/TDSC.2014.2373378

Wu, Y., Feng, G., Wang, N., & Liang, H. (2015). Game of information security investment: Impact of attack types and network vulnerability. *Expert Systems with Applications, 42*(15–16), 6132-6146. doi:http://dx.doi.org/10.1016/j.eswa.2015.03.033

Xiangyu, L. (2011). A Method of Network Topology Visualization Based on SNMP Symposium conducted at the meeting of the Instrumentation, Measurement, Computer, Communication and Control, 2011 First International Conference on doi:10.1109/IMCCC.2011.70

Xiaofeng, H., & Shifei, S. (2012). Study on the Resource Allocation in Urban Defense Engineering with Intentional Threats. *Systems Engineering Procedia, 5*(0), 198-206. doi:http://dx.doi.org/10.1016/j.sepro.2012.04.032

Yinghua, G., & Slay, J. (2010, 15-18 Feb. 2010). A Function Oriented Methodology to Validate and Verify Forensic Copy Function of Digital Forensic Tools Symposium conducted at the meeting of the ARES '10 International Conference on Availability, Reliability, and Security, 2010. doi:10.1109/ARES.2010.16

Yong-Dal, S. (2008). New Digital Forensics Investigation Procedure Model Symposium conducted at the meeting of the NCM '08. Fourth International Conference on Networked Computing and Advanced Information Management, 2008. doi:10.1109/NCM.2008.116

Yongle, W., & JunZhang, C. (2013). Hijacking spoofing attack and defense strategy based on Internet TCP sessions Symposium conducted at the meeting of the 2013 2nd International Symposium on Instrumentation and Measurement, Sensor Network and Automation (IMSNA), doi:10.1109/IMSNA.2013.6743326

Yonglin, S., Yongjun, W., Xin, H., Zhanrui, R., & Jie, L. (2011). A new perspective of network vulnerability analysis using Network Security Gradient Symposium conducted at the meeting of the 2011 International Conference for Internet Technology and Secured Transactions (ICITST),

Zhioua, S. (2013). The Middle East under Malware Attack Dissecting Cyber Weapons Symposium conducted at the meeting of the 2013 IEEE 33rd International Conference on Distributed Computing Systems Workshops (ICDCSW), doi:10.1109/ICDCSW.2013.30

Zhiyong, L., & Yunyan, Z. (2014). The Evaluation Model for Network Security Symposium conducted at the meeting of the Fourth International Conference on Communication Systems and Network Technologies (CSNT), 2014 doi:10.1109/CSNT.2014.145

APPENDICES

APPENDIX 1 OSSEC INSTALLATION

This command will install the ossec server:

```
[root@Emad-Forensic-Server]# wget -U ossec http://www.ossec.net/files/ossec-hids-2.8.1.tar.gz
```

Then decompress the file.

```
[root@Emad-Forensic-Server] # tar -zxvf ossec-hids-*.tar.gz (or gunzip -d; tar -xvf)
[root@Emad-Forensic-Server] # cd ossec-hids-*
```

Last step is installation the ossec.

```
[root@Emad-Forensic-Server] # ./install.sh

  ** Para instalação em português, escolha [br].
  ** 要使用中文进行安装, 请选择 [cn].
  ** Fur eine deutsche Installation wohlen Sie [de].
  ** Για εγκατάσταση στα Ελληνικά, επιλέξτε [el].
  ** For installation in English, choose [en].
  ** Para instalar en Española, eliga [es].
  ** Pour une installation en français, choisissez [fr]
  ** A Magyar nyelvű telepítéshez válassza [hu].
  ** Per l'installazione in Italiano, scegli [it].
  ** 日本語でインストールします．選択して下さい．[jp].
  ** Voor installatie in het Nederlands, kies [nl].
  ** Aby instalować w języku Polskim, wybierz [pl].
  ** Для инструкций по установке на русском, введите [ru].
  ** Za instalaciju na srpskom, izaberi [sr].
  ** Türkçe kurulum için seçin [tr].
  (en/br/cn/de/el/es/fr/hu/it/jp/nl/pl/ru/sr/tr) [en]: en

# OSSEC HIDS v2.8 Installation Script - http://www.ossec.net

 You are about to start the installation process of the OSSEC HIDS.
 You must have a C compiler pre-installed in your system.
 If you have any questions or comments, please send an e-mail
 to dcid@ossec.net (or daniel.cid@gmail.com).

  - System: Linux Emad-Forensic-Server 3.19.8-100.fc20. i686
  - User: root
  - Host: Emad-Forensic-Server

  -- Press ENTER to continue or Ctrl-C to abort. --
```

```
1- What kind of installation do you want (server, agent, local, hybrid or help)?
hybrid
  - Server installation chosen (hybrid).

2- Setting up the installation environment.
 - Choose where to install the OSSEC HIDS [/var/ossec]:
 - Installation will be made at /var/ossec.
 - The installation directory already exists. Should I delete it? (y/n) [y]:

3- Configuring the OSSEC HIDS.
  3.1- Do you want e-mail notification? (y/n) [y]:
   - What's your e-mail address? sts2010@localhost
   - What's your SMTP server ip/host? 127.0.0.1
  3.2- Do you want to run the integrity check daemon? (y/n) [y]:
   - Running syscheck (integrity check daemon).
  3.3- Do you want to run the rootkit detection engine? (y/n) [y]:
   - Running rootcheck (rootkit detection).
  3.4- Active response allows you to execute a specific
       command based on the events received. For example,
       you can block an IP address or disable access for a specific user.
       More information at:
       http://www.ossec.net/en/manual.html#active-response

   - Do you want to enable active response? (y/n) [y]:

     - Active response enabled.

   - By default, we can enable the host-deny and the
     firewall-drop responses. The first one will add
     a host to the /etc/hosts.deny and the second one
     will block the host on iptables (if linux) or on
     ipfilter (if Solaris, FreeBSD or NetBSD).
   - They can be used to stop SSHD brute force scans,
     portscans and some other forms of attacks. You can
     also add them to block on snort events, for example.

   - Do you want to enable the firewall-drop response? (y/n) [y]:

     - firewall-drop enabled (local) for levels >= 6

   - Default white list for the active response:
      -  xxx.xxx.xxx.xxx

   - Do you want to add more IPs to the white list? (y/n)? [n]: y
   - IPs (space separated): xxx.xxx.xxx.xxx/29

  3.5- Do you want to enable remote syslog (port 514 udp)? (y/n) [y]:

   - Remote syslog enabled.

  3.6- Setting the configuration to analyse the following logs:
    -- /var/log/messages
    -- /var/log/secure
    -- /var/log/maillog
    -- /var/log/httpd/error_log (apache log)
    -- /var/log/httpd/access_log (apache log)

 - If you want to monitor any other file, just change
   the ossec.conf and add a new local file entry.
   Any questions about the configuration can be answered
   by visiting us online at http://www.ossec.net.

      --- Press ENTER to continue ---

5- Installing the system
 - Running the Makefile
INFO: Little endian set.

 *** Making zlib (by Jean-loup Gailly and Mark Adler) ***
Make [1]: Entering directory `/home/ossec-hids-2.8.1/src/external'
Make [1]: Leaving directory `/home/ossec-hids-2.8.1/src/external'

 *** Making cJSON (by Dave Gamble) ***
Make [1]: Entering directory `/home/ossec-hids-2.8.1/src/external/cJSON'
Make [1]: Leaving directory `/home/ossec-hids-2.8.1/src/external/cJSON'

 *** Making Lua 5.2 (by team at PUC-Rio in Brazi) ***
     Copyright © 1994–2014 Lua.org, PUC-Rio.
Make [1]: Entering directory `/home/ossec-hids-2.8.1/src/external/lua-5.2.3'
Make [2]: Entering directory `/home/ossec-hids-2.8.1/src/external/lua-5.2.3/src'
Make all SYSCFLAGS="-DLUA_USE_POSIX"
Make [3]: Entering directory `/home/ossec-hids-2.8.1/src/external/lua-5.2.3/src'
Make [3]: Nothing to be done for `all'.
Make [3]: Leaving directory `/home/ossec-hids-2.8.1/src/external/lua-5.2.3/src'
Make [2]: Leaving directory `/home/ossec-hids-2.8.1/src/external/lua-5.2.3/src'
Make [1]: Leaving directory `/home/ossec-hids-2.8.1/src/external/lua-5.2.3'

 *** Making os_xml ***
```

```
Make [1]: Entering directory `/home/ossec-hids-2.8.1/src/os_xml'
Make [1]: `os_xml. a' is up to date.
Make [1]: Leaving directory `/home/ossec-hids-2.8.1/src/os_xml'

 *** Making os_regex ***
Make [1]: Entering directory `/home/ossec-hids-2.8.1/src/os_regex'
Make [1]: `os_regex. a' is up to date.
Make [1]: Leaving directory `/home/ossec-hids-2.8.1/src/os_regex'

 *** Making os_net ***
make [1]: Entering directory `/home/ossec-hids-2.8.1/src/sonnet'
Make [1]: `os_net. a' is up to date.
Make [1]: Leaving directory `/home/ossec-hids-2.8.1/src/os_net'

 *** Making os_crypto ***
Make [1]: Entering directory `/home/ossec-hids-2.8.1/src/os_crypto'
Make [2]: Entering directory `/home/ossec-hids-2.8.1/src/os_crypto/blowfish'
Make [2]: Leaving directory `/home/ossec-hids-2.8.1/src/os_crypto/blowfish'
Make [2]: Entering directory `/home/ossec-hids-2.8.1/src/os_crypto/md5'
Make [2]: Leaving directory `/home/ossec-hids-2.8.1/src/os_crypto/md5'
Make [2]: Entering directory `/home/ossec-hids-2.8.1/src/os_crypto/sha1'
Make [2]: Leaving directory `/home/ossec-hids-2.8.1/src/os_crypto/sha1'
Make [2]: Entering directory `/home/ossec-hids-2.8.1/src/os_crypto/md5_sha1'
Make [2]: Leaving directory `/home/ossec-hids-2.8.1/src/os_crypto/md5_sha1'
Make [2]: Entering directory `/home/ossec-hids-2.8.1/src/os_crypto/shared'
Make [2]: Leaving directory `/home/ossec-hids-2.8.1/src/os_crypto/shared'
Make [1]: Leaving directory `/home/ossec-hids-2.8.1/src/os_crypto'

 *** Making shared ***
Make [1]: Entering directory `/home/ossec-hids-2.8.1/src/shared'
Make [1]: Leaving directory `/home/ossec-hids-2.8.1/src/shared'

 *** Making config ***
Make [1]: Entering directory `/home/ossec-hids-2.8.1/src/config'
Make [1]: Leaving directory `/home/ossec-hids-2.8.1/src/config'

 *** Making os_maild ***
Make [1]: Entering directory `/home/ossec-hids-2.8.1/src/os_maild'
Make [1]: Leaving directory `/home/ossec-hids-2.8.1/src/os_maild'

 *** Making os_dbd ***
Make [1]: Entering directory `/home/ossec-hids-2.8.1/src/os_dbd'
Compiling DB support with:
Make [1]: Leaving directory `/home/ossec-hids-2.8.1/src/os_dbd'
```

```
 *** Making os_csyslogd ***
Make [1]: Entering directory `/home/ossec-hids-2.8.1/src/os_csyslogd'
Make [1]: Leaving directory `/home/ossec-hids-2.8.1/src/os_csyslogd'

 *** Making agentlessd ***
Make [1]: Entering directory `/home/ossec-hids-2.8.1/src/agentlessd'
Make [1]: Leaving directory `/home/ossec-hids-2.8.1/src/agentlessd'

 *** Making os_execd ***
Make [1]: Entering directory `/home/ossec-hids-2.8.1/src/os_execd'
Make [1]: Leaving directory `/home/ossec-hids-2.8.1/src/os_execd'

 *** Making analysisd ***
Make [1]: Entering directory `/home/ossec-hids-2.8.1/src/analysisd'
Make [2]: Entering directory `/home/ossec-hids-2.8.1/src/analysisd/cdb'
Make [2]: Leaving directory `/home/ossec-hids-2.8.1/src/analysisd/cdb'
Make [2]: Entering directory `/home/ossec-hids-2.8.1/src/analysisd/alerts'
Make [2]: Leaving directory `/home/ossec-hids-2.8.1/src/analysisd/alerts'
Make [2]: Entering directory `/home/ossec-hids-2.8.1/src/analysisd/decoders'
Make [3]: Entering directory `/home/ossec-hids-2.8.1/src/analysisd/decoders/plugins'
Make [3]: Leaving directory `/home/ossec-hids-2.8.1/src/analysisd/decoders/plugins'
Make [2]: Leaving directory `/home/ossec-hids-2.8.1/src/analysisd/decoders'
Make [2]: Entering directory `/home/ossec-hids-2.8.1/src/analysisd/compiled_rules'
Make [2]: Leaving directory `/home/ossec-hids-2.8.1/src/analysisd/compiled_rules'
Make [2]: Entering directory `/home/ossec-hids-2.8.1/src/analysisd/cdb'
Make [2]: Leaving directory `/home/ossec-hids-2.8.1/src/analysisd/cdb'
Make [2]: Entering directory `/home/ossec-hids-2.8.1/src/analysisd/decoders'
Make [3]: Entering directory `/home/ossec-hids-2.8.1/src/analysisd/decoders/plugins'
Make [3]: Leaving directory `/home/ossec-hids-2.8.1/src/analysisd/decoders/plugins'
Make [2]: Leaving directory `/home/ossec-hids-2.8.1/src/analysisd/decoders'
Make [2]: Entering directory `/home/ossec-hids-2.8.1/src/analysisd/compiled_rules'
./register_rule.sh build
*Build completed.
Make [2]: Leaving directory `/home/ossec-hids-2.8.1/src/analysisd/compiled_rules'
Make [2]: Entering directory `/home/ossec-hids-2.8.1/src/analysisd/cdb'
Make [2]: Leaving directory `/home/ossec-hids-2.8.1/src/analysisd/cdb'
Make [1]: Leaving directory `/home/ossec-hids-2.8.1/src/analysisd'

 *** Making logcollector ***
Make [1]: Entering directory `/home/ossec-hids-2.8.1/src/logcollector'
Make [1]: Leaving directory `/home/ossec-hids-2.8.1/src/logcollector'

 *** Making remoted ***
```

```
Make [1]: Entering directory `/home/ossec-hids-2.8.1/src/remoted'
Make [1]: Leaving directory `/home/ossec-hids-2.8.1/src/remoted'

 *** Making client-agent ***
Make [1]: Entering directory `/home/ossec-hids-2.8.1/src/client-agent'
Make [1]: Leaving directory `/home/ossec-hids-2.8.1/src/client-agent'

 *** Making addagent ***
Make [1]: Entering directory `/home/ossec-hids-2.8.1/src/addagent'
Make [1]: Leaving directory `/home/ossec-hids-2.8.1/src/addagent'

 *** Making util ***
Make [1]: Entering directory `/home/ossec-hids-2.8.1/src/util'
Make [1]: Leaving directory `/home/ossec-hids-2.8.1/src/util'

 *** Making rootcheck ***

Make [1]: Entering directory `/home/ossec-hids-2.8.1/src/rootcheck'
Make [1]: Leaving directory `/home/ossec-hids-2.8.1/src/rootcheck'

 *** Making syscheckd ***
Make [1]: Entering directory `/home/ossec-hids-2.8.1/src/syscheckd'
Make [1]: Leaving directory `/home/ossec-hids-2.8.1/src/syscheckd'

 *** Making monitord ***
Make [1]: Entering directory `/home/ossec-hids-2.8.1/src/monitord'
Make [1]: Leaving directory `/home/ossec-hids-2.8.1/src/monitord'

 *** Making os_auth ***
Make [1]: Entering directory `/home/ossec-hids-2.8.1/src/os_auth'
Make [1]: Leaving directory `/home/ossec-hids-2.8.1/src/os_auth'
Make [1]: Entering directory `/home/ossec-hids-2.8.1/src/os_maild'
Make [1]: Leaving directory `/home/ossec-hids-2.8.1/src/os_maild'
Make [1]: Entering directory `/home/ossec-hids-2.8.1/src/os_dbd'
Make [1]: Leaving directory `/home/ossec-hids-2.8.1/src/os_dbd'
Make [1]: Entering directory `/home/ossec-hids-2.8.1/src/os_csyslogd'
Make [1]: Leaving directory `/home/ossec-hids-2.8.1/src/os_csyslogd'
Make [1]: Entering directory `/home/ossec-hids-2.8.1/src/agentlessd'
Make [1]: Leaving directory `/home/ossec-hids-2.8.1/src/agentlessd'
Make [1]: Entering directory `/home/ossec-hids-2.8.1/src/os_execd'
Make [1]: Leaving directory `/home/ossec-hids-2.8.1/src/os_execd'
Make [1]: Entering directory `/home/ossec-hids-2.8.1/src/analysisd'
Make [1]: Leaving directory `/home/ossec-hids-2.8.1/src/analysisd'
Make [1]: Entering directory `/home/ossec-hids-2.8.1/src/logcollector'
Make [1]: Leaving directory `/home/ossec-hids-2.8.1/src/logcollector'
Make [1]: Entering directory `/home/ossec-hids-2.8.1/src/remoted'
Make [1]: Leaving directory `/home/ossec-hids-2.8.1/src/remoted'
Make [1]: Entering directory `/home/ossec-hids-2.8.1/src/client-agent'
Make [1]: Leaving directory `/home/ossec-hids-2.8.1/src/client-agent'
Make [1]: Entering directory `/home/ossec-hids-2.8.1/src/addagent'
Make [1]: Leaving directory `/home/ossec-hids-2.8.1/src/addagent'
Make [1]: Entering directory `/home/ossec-hids-2.8.1/src/util'
Make [1]: Leaving directory `/home/ossec-hids-2.8.1/src/util'
Make [1]: Entering directory `/home/ossec-hids-2.8.1/src/rootcheck'
Make [1]: Nothing to be done for `build'.
Make [1]: Leaving directory `/home/ossec-hids-2.8.1/src/rootcheck'
Make [1]: Entering directory `/home/ossec-hids-2.8.1/src/syscheckd'
Make [1]: Leaving directory `/home/ossec-hids-2.8.1/src/syscheckd'
Make [1]: Entering directory `/home/ossec-hids-2.8.1/src/monitord'
Make [1]: Leaving directory `/home/ossec-hids-2.8.1/src/monitord'
Make [1]: Entering directory `/home/ossec-hids-2.8.1/src/os_auth'
Make [1]: Leaving directory `/home/ossec-hids-2.8.1/src/os_auth'

 - System is Redhat Linux.
 - Init script modified to start OSSEC HIDS during boot.

 - Configuration finished properly.

 - To start OSSEC HIDS:
                /var/ossec/bin/ossec-control start

 - To stop OSSEC HIDS:
                /var/ossec/bin/ossec-control stop

 - The configuration can be viewed or modified at /var/ossec/etc/ossec.conf

   Thanks for using the OSSEC HIDS.
   If you have any question, suggestion or if you find any bug,
   contact us at contact@ossec.net or using our public maillist at
   ossec-list@ossec.net
   (http://www.ossec.net/main/support/).

   More information can be found at http://www.ossec.net

   --- Press ENTER to finish (maybe more information below). ---

 - In order to connect agent and server, you need to add each agent to the server.
   Run the 'manage_agents' to add or remove them:
```

```
   /var/ossec/bin/manage_agents

   More information at:
   http://www.ossec.net/en/manual.html

   ------------------------------------------
   Finishing Hybrid setup (agent configuration)
   ------------------------------------------
#
 OSSEC HIDS v2.8 Installation Script - http://www.ossec.net

 You are about to start the installation process of the OSSEC HIDS.
 You must have a C compiler pre-installed in your system.
 If you have any questions or comments, please send an e-mail
 to dcid@ossec.net (or daniel.cid@gmail.com).

  - System: Emad-Forensic-Server
  - User: root
  - Host: IDS-Server

  -- Press ENTER to continue or Ctrl-C to abort. --

2- Setting up the installation environment.

    - Installation will be made at /var/ossec/ossec-agent.

3- Configuring the OSSEC HIDS.

  3.1- What's the IP Address or hostname of the OSSEC HIDS server?
xxx.xxx.xxx.xxx

   - Adding Server IP xxx.xxx.xxx.xxx

  3.2- Do you want to run the integrity check daemon? (y/n) [y]:
   - Not running syscheck (integrity check daemon).

  3.3- Do you want to run the rootkit detection engine? (y/n) [y]:
   - Not running rootcheck (rootkit detection).

  3.4 - Do you want to enable active response? (y/n) [y]:
   - Active response disabled.

5- Installing the system
```

```
 - Running the Makefile
INFO: Little endian set.

 *** Making zlib (by Jean-loup Gailly and Mark Adler) ***
Make [1]: Entering directory `/home/ossec-hids-2.8.1/src/external'
Make [1]: Leaving directory `/home/ossec-hids-2.8.1/src/external'

 *** Making cJSON (by Dave Gamble) ***
Make [1]: Entering directory `/home/ossec-hids-2.8.1/src/external/cJSON'
Make [1]: Leaving directory `/home/ossec-hids-2.8.1/src/external/cJSON'

 *** Making Lua 5.2 (by team at PUC-Rio in Brazi) ***
     Copyright © 1994–2014 Lua.org, PUC-Rio.

Make [1]: Entering directory `/home/ossec-hids-2.8.1/src/external/lua-5.2.3'
Make [2]: Entering directory `/home/ossec-hids-2.8.1/src/external/lua-5.2.3/src'
Make all SYSCFLAGS="-DLUA_USE_POSIX"
Make [3]: Entering directory `/home/ossec-hids-2.8.1/src/external/lua-5.2.3/src'
Make [3]: Nothing to be done for `all'.
Make [3]: Leaving directory `/home/ossec-hids-2.8.1/src/external/lua-5.2.3/src'
Make [2]: Leaving directory `/home/ossec-hids-2.8.1/src/external/lua-5.2.3/src'
Make [1]: Leaving directory `/home/ossec-hids-2.8.1/src/external/lua-5.2.3'

 *** Making os_xml ***
Make [1]: Entering directory `/home/ossec-hids-2.8.1/src/os_xml'
Make [1]: `os_xml.a' is up to date.
Make [1]: Leaving directory `/home/ossec-hids-2.8.1/src/os_xml'

 *** Making os_regex ***
Make [1]: Entering directory `/home/ossec-hids-2.8.1/src/os_regex'
Make [1]: `os_regex. a' is up to date.
Make [1]: Leaving directory `/home/ossec-hids-2.8.1/src/os_regex'

 *** Making os_net ***
Make [1]: Entering directory `/home/ossec-hids-2.8.1/src/os_net'
Make [1]: `os_net. a' is up to date.
Make [1]: Leaving directory `/home/ossec-hids-2.8.1/src/os_net'

 *** Making os_crypto ***
Make [1]: Entering directory `/home/ossec-hids-2.8.1/src/os_crypto'
Make [2]: Entering directory `/home/ossec-hids-2.8.1/src/os_crypto/blowfish'
Make [2]: Leaving directory `/home/ossec-hids-2.8.1/src/os_crypto/blowfish'
Make [2]: Entering directory `/home/ossec-hids-2.8.1/src/os_crypto/md5'
Make [2]: Leaving directory `/home/ossec-hids-2.8.1/src/os_crypto/md5'
Make [2]: Entering directory `/home/ossec-hids-2.8.1/src/os_crypto/sha1'
Make [2]: Leaving directory `/home/ossec-hids-2.8.1/src/os_crypto/sha1'
Make [2]: Entering directory `/home/ossec-hids-2.8.1/src/os_crypto/md5_sha1'
Make [2]: Leaving directory `/home/ossec-hids-2.8.1/src/os_crypto/md5_sha1'
Make [2]: Entering directory `/home/ossec-hids-2.8.1/src/os_crypto/shared'
Make [2]: Leaving directory `/home/ossec-hids-2.8.1/src/os_crypto/shared'
Make [1]: Leaving directory `/home/ossec-hids-2.8.1/src/os_crypto'

 *** Making shared ***
Make [1]: Entering directory `/home/ossec-hids-2.8.1/src/shared'
Make [1]: Leaving directory `/home/ossec-hids-2.8.1/src/shared'

 *** Making config ***
Make [1]: Entering directory `/home/ossec-hids-2.8.1/src/config'
Make [1]: Leaving directory `/home/ossec-hids-2.8.1/src/config'

 *** Making os_maild ***
Make [1]: Entering directory `/home/ossec-hids-2.8.1/src/os_maild'
Make [1]: Leaving directory `/home/ossec-hids-2.8.1/src/os_maild'

 *** Making os_dbd ***
Make [1]: Entering directory `/home/ossec-hids-2.8.1/src/os_dbd'
Make [1]: Leaving directory `/home/ossec-hids-2.8.1/src/os_dbd'

 *** Making os_csyslogd ***
Make [1]: Entering directory `/home/ossec-hids-2.8.1/src/os_csyslogd'
Make [1]: Leaving directory `/home/ossec-hids-2.8.1/src/os_csyslogd'

 *** Making agentlessd ***
Make [1]: Entering directory `/home/ossec-hids-2.8.1/src/agentlessd'
Make [1]: Leaving directory `/home/ossec-hids-2.8.1/src/agentlessd'

 *** Making os_execd ***
Make [1]: Entering directory `/home/ossec-hids-2.8.1/src/os_execd'
Make [1]: Leaving directory `/home/ossec-hids-2.8.1/src/os_execd'

 *** Making analysisd ***
Make [1]: Entering directory `/home/ossec-hids-2.8.1/src/analysisd'
Make [2]: Entering directory `/home/ossec-hids-2.8.1/src/analysisd/cdb'
Make [2]: Leaving directory `/home/ossec-hids-2.8.1/src/analysisd/cdb'
Make [2]: Entering directory `/home/ossec-hids-2.8.1/src/analysisd/alerts'
Make [2]: Leaving directory `/home/ossec-hids-2.8.1/src/analysisd/alerts'
Make [2]: Entering directory `/home/ossec-hids-2.8.1/src/analysisd/decoders'
Make [3]: Entering directory `/home/ossec-hids-2.8.1/src/analysisd/decoders/plugins'
Make [3]: Leaving directory `/home/ossec-hids-2.8.1/src/analysisd/decoders/plugins'
Make [2]: Leaving directory `/home/ossec-hids-2.8.1/src/analysisd/decoders'
Make [2]: Entering directory `/home/ossec-hids-2.8.1/src/analysisd/compiled_rules'
Make [2]: Entering directory `/home/ossec-hids-2.8.1/src/analysisd/cdb'
Make [2]: Leaving directory `/home/ossec-hids-2.8.1/src/analysisd/cdb'
Make [2]: Entering directory `/home/ossec-hids-2.8.1/src/analysisd/decoders'
Make [3]: Entering directory `/home/ossec-hids-2.8.1/src/analysisd/decoders/plugins'
Make [2]: Leaving directory `/home/ossec-hids-2.8.1/src/analysisd/decoders'
Make [2]: Entering directory `/home/ossec-hids-2.8.1/src/analysisd/compiled_rules'
Make [Entering directory `/home/ossec-hids-2.8.1/src/analysisd/cdb]: [….]
Make [Leaving directory `/home/ossec-hids-2.8.1/src/analysisd/cdb]: [….]
Make [Leaving directory `/home/ossec-hids-2.8.1/src/analysisd]: [….]

 *** Making logcollector ***
Make [1]: Entering directory `/home/ossec-hids-2.8.1/src/logcollector'
Make [1]: Leaving directory `/home/ossec-hids-2.8.1/src/logcollector'

 *** Making remoted ***
Make [Entering directory `/home/ossec-hids-2.8.1/src/remoted]: [….]
Make [Leaving directory `/home/ossec-hids-2.8.1/src/remoted]: [….]

 *** Making client-agent ***
Make [Entering directory `/home/ossec-hids-2.8.1/src/client-agent]: [….]
Make [Leaving directory `/home/ossec-hids-2.8.1/src/client-agent]: [….]

 *** Making addagent ***
Make [Entering directory `/home/ossec-hids-2.8.1/src/addagent]: [….]
Make [Leaving directory `/home/ossec-hids-2.8.1/src/addagent]: [….]

 *** Making util ***
Make [Entering directory `/home/ossec-hids-2.8.1/src/util]: [….]
Make [Leaving directory `/home/ossec-hids-2.8.1/src/util]: [….]

 *** Making rootcheck ***
Make [Entering directory `/home/ossec-hids-2.8.1/src/rootcheck]: [….]
Make [Leaving directory `/home/ossec-hids-2.8.1/src/rootcheck]: [….]

 *** Making syscheckd ***
Make [Entering directory `/home/ossec-hids-2.8.1/src/syscheckd]: [….]

 *** Making monitord ***
Make [Entering directory `/home/ossec-hids-2.8.1/src/monitord]: [….]
Make [Leaving directory `/home/ossec-hids-2.8.1/src/monitord]: [….]
```

```
 *** Making os_auth ***
Make [Entering directory `/home/ossec-hids-2.8.1/src/os_auth]: [….]'

 - System is Redhat Linux.
 - Init script modified to start OSSEC HIDS during boot.

 - Configuration finished properly.

 - To start OSSEC HIDS:
                /var/ossec/ossec-agent/bin/ossec-control start

 - To stop OSSEC HIDS:
                /var/ossec/ossec-agent/bin/ossec-control stop

 - The configuration can be viewed or modified at /var/ossec/ossec-
agent/etc/ossec.conf

   Thanks for using the OSSEC HIDS.
   If you have any question, suggestion or if you find any bug,
   contact us at contact@ossec.net or using our public maillist at
   ossec-list@ossec.net
   (http://www.ossec.net/main/support/).

   More information can be found at http://www.ossec.net

   --- Press ENTER to finish (maybe more information below). ---

 - You first need to add this agent to the server so they
   can communicate with each other. When you have done so,
   you can run the 'manage_agents' tool to import the
   authentication key from the server.

   /var/ossec/ossec-agent/bin/manage_agents

   More information at:
   http://www.ossec.net/en/manual.html#ma

[root@Emad-Forensic-Server ossec-hids-2.8.1] #
[root@Emad-Forensic-Server] # /var/ossec/bin/ossec-control start
Starting OSSEC HIDS v2.8 (by Trend Micro Inc.) ...
Started ossec-maild...
Started ossec-execd...
Started ossec-analysisd...
Started ossec-logcollector...
Started ossec-remoted...
Started ossec-syscheckd...
Started ossec-monitord...
Completed.
```

Configure OSSEC for Real-time Alerts on File Modifications

Change the time from 79200 seconds to 20 and add new command to send an email about create a new file or any other malicious activities.

```
[root@Emad-Forensic-Server] # gedit /var/ossec/etc/ossec.conf
```

```
<syscheck>
   <! -- Frequency that syscheck is executed - default to every 22 hours -->
   <frequency>20</frequency>
```

And add this command to arise an email about any changes.

```
   <alert_new_files>yes</alert_new_files>
```

Adding the report_changes="yes" realtime="yes" to enable real-time monitoring as following:

```
<! -- Directories to check (perform all possible verifications) -->
<directories report_changes="yes" realtime="yes"
check_all="yes">/etc/usr/bin/usr/sbin</directories>
<directories report_changes="yes" realtime="yes"
check_all="yes">/bin/sbin</directories>
```

Adding new command that will monitor the home director and the type of the files

```
<directories report_changes="yes" realtime="yes" restrict=".php|.js|.py|.sh|.html"
check_all="yes">/home/nst/var/www</directories>
```

In ossec_rules.xml rules and its path is /var/ossec/rules/local_rules. xm, change the rule ID 554 like this:

```
<rule id="554" level="7" overwrite="yes">
```

Then issues this command in order to restart the ossec

```
[root@Emad-Forensic-Server ~] # /var/ossec/bin/ossec-control restart
```

In order to connect to database this file /etc/my.cnf should be configured by removing /mnt/ram4:

```
[root@Emad-Forensic-Server ~] # gedit /etc/my.cnf
```

These commands should be issued in order to create database that will store all ossec's alters:

```
[root@Emad-Forensic-Server ~] # mysql -u root
Enter password:
```

```
[...]

mysql> create database ossec;
Query OK, 1 row affected (0.00 sec)

mysql> grant INSERT, SELECT, UPDATE, CREATE, DELETE, EXECUTE on
ossec. * to root;
Query OK, 0 rows affected (0.00 sec)

mysql> set password for root = PASSWORD('sguil');
Query OK, 1 row affected (0.00 sec)

mysql> flush privileges;
Query OK, 0 rows affected (0.00 sec)

mysql> quit;
```

Adding schema to database

```
mysql -u root -p ossec < /opt/ossec-hids-2.8.1/src/os_dbd/mysql.schema
```

Then I have to add these lines into the database config /var/ossec/etc/ossec.conf:

```
<ossec_config>
   <database_output>
      <hostname>127.0.0.1</hostname>
      <username>root</username>
      <password>sguil</password>
      <database>ossec</database>
      <type>mysql</type>
   </database_output>
</ossec_config>
```

After that enable the database

```
[root@Emad-Forensic-Server ~] # /var/ossec/bin/ossec-control enable database
```

Then restart the OSSEC

```
[root@Emad-Forensic-Server ~] # /var/ossec/bin/ossec-control restart
```

Installing ossec-wui is also quite simple. Because we've already set up Apache and PHP, we can just download the web UI and extract into /var/www/html:

```
[root@Emad-Forensic-Server ~] # wget http://www.ossec.net/files/ossec-wui-0.8.tar.gz
[root@Emad-Forensic-Server ~] # tar -xf ossec-wui-0.8.tar.gz
[root@Emad-Forensic-Server ~] # mkdir -p /var/www/html/ossec/tmp/
```

```
[root@Emad-Forensic-Server ~] # mv ossec-wui-0.8/* /var/www/html/ossec/
[root@Emad-Forensic-Server ~] # chown apache: apache /var/www/html/ossec/tmp/
[root@Emad-Forensic-Server ~] # chmod 770 /var/www/html/ossec/tmp
```

Make sure the apache user can access the ossec folder:

```
[root@Emad-Forensic-Server ~] # usermod -a -G ossec apache
```

Then I add these lines in /etc/httpd/conf/httpd.conf

```
ServerAdmin root@Emad-Forensic-Server.aut

# These new lines that will create the new website server hosting
<VirtualHost *:80>
ServerName Emad-Forensic-Server.aut
DocumentRoot "/var/www/html/ossec/"
ServerAlias Emad-Forensic-Server
</VirtualHost>
```

The final step is adding the IP in hosts by editing the file of /etc/hosts

```
127.0.0.1        Emad-Forensic-Server.aut
```

Then restart httpd by issue this command httpd -k restart

This command will configure all agents that exist in the network

```
[root@Emad-Forensic-Server ~] # /var/ossec/bin/manage_agents

****************************************
* OSSEC HIDS v2.8 Agent manager.     *
* The following options are available: *
****************************************
   (A)dd an agent (A).
   (E)xtract key for an agent (E).
   (L)ist already added agents (L).
   (R)emove an agent (R).
   (Q)uit.
Choose your action: A, E, L, R or Q: a

- Adding a new agent (use '\q' to return to the main menu).
  Please provide the following:
   * A name for the new agent: Windows 7

** Invalid name 'Windows 7' given. Name must contain only alphanumeric
characters (min=2, max=32).

   * A name for the new agent: Windows-7
   * The IP Address of the new agent:  xxx.xxx.xxx.xxx
   * An ID for the new agent [001]:
Agent information:
   ID:001
   Name: Windows-7
   IP Address: xxx.xxx.xxx.xxx

Confirm adding it? (y/n): y
Agent added.
```

Then will generate key for this agents by chosen E

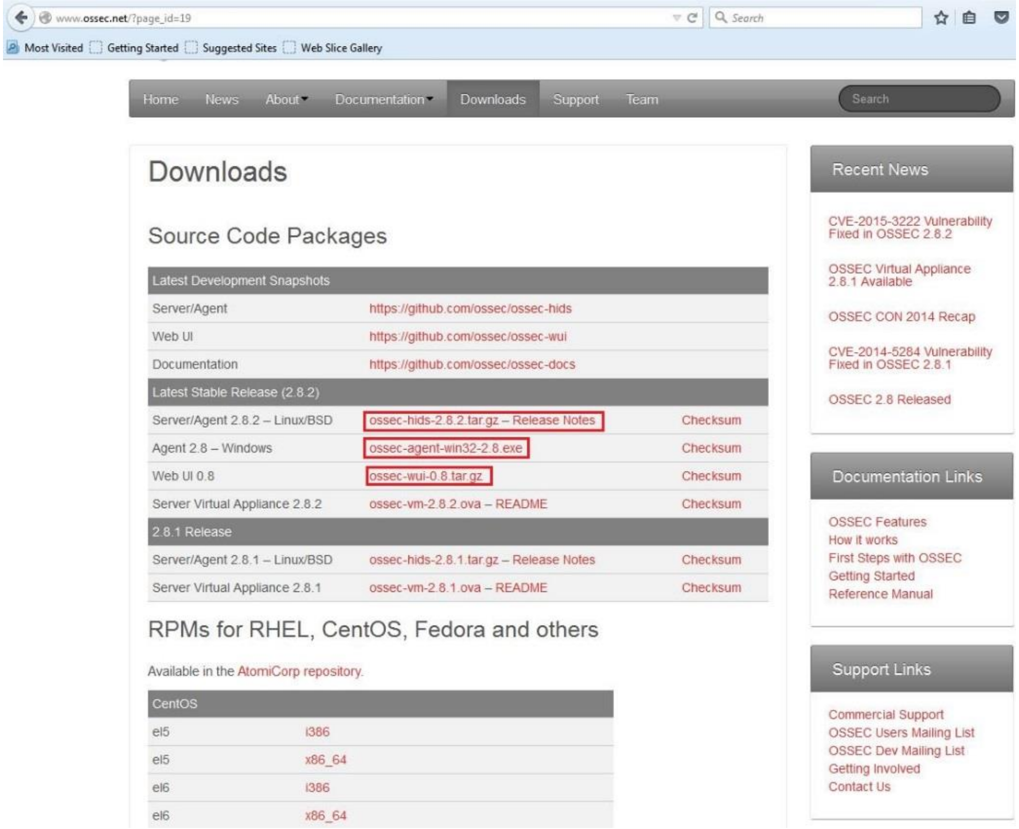
```
****************************************
* OSSEC HIDS v2.8 Agent manager.     *
* The following options are available: *
****************************************
  (A)dd an agent (A).
  (E)xtract key for an agent (E).
  (L)ist already added agents (L).
  (R)emove an agent (R).
  (Q)uit.
Choose your action: A, E, L, R or Q: e

Available agents:
  ID: 001, Name: Windows-7, IP:  xxx.xxx.xxx.xxx
Provide the ID of the agent to extract the key (or '\q' to quit): 001

Agent key information for '001' is:
MDAxIFdpbmRvd3MtNyA2MC4yMzQuNDMuNDQgZGQ0YjgzNjU0MDM2MzV
jMjAwODZlMDNlMzYzMzkwNDM3OTVmZTVjMmY3YjkzZDIyZjZmNTE2M
WNiZWJkMmEyZA==

** Press ENTER to return to the main menu.

****************************************
* OSSEC HIDS v2.8 Agent manager.     *
* The following options are available: *
****************************************
  (A)dd an agent (A).
  (E)xtract key for an agent (E).
  (L)ist already added agents (L).
  (R)emove an agent (R).
  (Q)uit.
Choose your action: A, E, L, R or Q: a
```

```
- Adding a new agent (use '\q' to return to the main menu).
  Please provide the following:
   * A name for the new agent: Windows-Server
   * The IP Address of the new agent:  xxx.xxx.xxx.xxx
   * An ID for the new agent [002]:
Agent information:
   ID:002
   Name: Windows-Server
   IP Address: xxx.xxx.xxx.xxx

Confirm adding it? (y/n): y
Agent added.

***************************************
* OSSEC HIDS v2.8 Agent manager.     *
* The following options are available: *
***************************************
   (A)dd an agent (A).
   (E)xtract key for an agent (E).
   (L)ist already added agents (L).
   (R)emove an agent (R).
   (Q)uit.
Choose your action: A, E, L, R or Q: e

Available agents:
   ID: 001, Name: Windows-7, IP:  xxx.xxx.xxx.xxx
   ID: 002, Name: Windows-Server, IP:  xxx.xxx.xxx.xxx
Provide the ID of the agent to extract the key (or '\q' to quit): 002

Agent key information for '002' is:
MDAyIFdpbmRvd3MtU2VydmVyIDYwLjIzNC40My40NSBiM2RmZTcwNmRiZ
Dg1ZWFiNTk2YTVlYmM1OWJjNzVmMmNkMDUwYjU2YzhjZDQ4ZmVhMzQ
5Nzc3ODViOTExN2Ez

** Press ENTER to return to the main menu.

***************************************
* OSSEC HIDS v2.8 Agent manager.     *
* The following options are available: *
***************************************
   (A)dd an agent (A).
   (E)xtract key for an agent (E).
   (L)ist already added agents (L).
   (R)emove an agent (R).
   (Q)uit.
Choose your action: A, E, L, R or Q: a

- Adding a new agent (use '\q' to return to the main menu).
  Please provide the following:
   * A name for the new agent: Forensic-Server
   * The IP Address of the new agent:  xxx.xxx.xxx.xxx
   * An ID for the new agent [003]:
Agent information:
   ID:003
   Name: Forensic-Server
   IP Address: xxx.xxx.xxx.xxx

Confirm adding it? (y/n): y
Agent added.

***************************************
* OSSEC HIDS v2.8 Agent manager.     *
* The following options are available: *
***************************************
   (A)dd an agent (A).
   (E)xtract key for an agent (E).
   (L)ist already added agents (L).
   (R)emove an agent (R).
   (Q)uit.
Choose your action: A, E, L, R or Q: e

Available agents:
   ID: 001, Name: Windows-7, IP:  xxx.xxx.xxx.xxx
   ID: 002, Name: Windows-Server, IP:  xxx.xxx.xxx.xxx
   ID: 003, Name: Forensic-Server, IP:  xxx.xxx.xxx.xxx
Provide the ID of the agent to extract the key (or '\q' to quit): 003

Agent key information for '003' is:
MDAzIEZvcmVuaWMtU2VydmVyIDYwLjIzNC40My40MyAxYWE5MmI0ZWY
0NmEyNmFjNzMzOGJlZGEwNTNkZWI4NDI2NmY4MjM0YzAyZGRlODQ5M2
Q5OTFlMTI4NGVjYzcw

** Press ENTER to return to the main menu.

***************************************
* OSSEC HIDS v2.8 Agent manager.     *
* The following options are available: *
```

```
****************************************
   (A)dd an agent (A).
   (E)xtract key for an agent (E).
   (L)ist already added agents (L).
   (R)emove an agent (R).
   (Q)uit.
Choose your action: A, E, L, R or Q: q

** You must restart OSSEC for your changes to take effect.

manage_agents: Exiting.
[root@Emad-Forensic-Server ~] #
```

APPENDIX 2 OSSEC’ AGENT INSTALLATION

Picture 1 Download from website



Downloads

Source Code Packages

Latest Development Snapshots		
Server/Agent	https://github.com/ossec/ossec-hids	
Web UI	https://github.com/ossec/ossec-wui	
Documentation	https://github.com/ossec/ossec-docs	
Latest Stable Release (2.8.2)		
Server/Agent 2.8.2 – Linux/BSD	ossec-hids-2.8.2.tar.gz – Release Notes	Checksum
Agent 2.8 – Windows	ossec-agent-win32-2.8.exe	Checksum
Web UI 0.8	ossec-wui-0.8.tar.gz	Checksum
Server Virtual Appliance 2.8.2	ossec-vm-2.8.2.ova – README	Checksum
2.8.1 Release		
Server/Agent 2.8.1 – Linux/BSD	ossec-hids-2.8.1.tar.gz – Release Notes	Checksum
Server Virtual Appliance 2.8.1	ossec-vm-2.8.1.ova – README	Checksum

RPMs for RHEL, CentOS, Fedora and others

Available in the AtomiCorp repository.

CentOS	
el5	i386
el5	x86_64
el6	i386
el6	x86_64

Recent News

CVE-2015-3222 Vulnerability Fixed in OSSEC 2.8.2

OSSEC Virtual Appliance 2.8.1 Available

OSSEC CON 2014 Recap

CVE-2014-5284 Vulnerability Fixed in OSSEC 2.8.1

OSSEC 2.8 Released

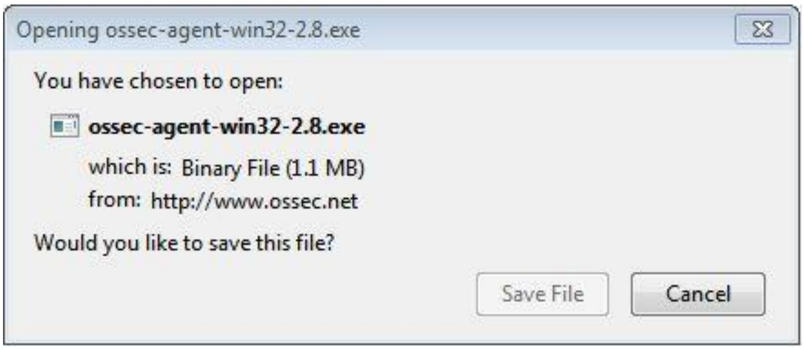
Documentation Links

OSSEC Features
How it works
First Steps with OSSEC
Getting Started
Reference Manual

Support Links

Commercial Support
OSSEC Users Mailing List
OSSEC Dev Mailing List
Getting Involved
Contact Us

Picture 2: Save the file



Opening ossec-agent-win32-2.8.exe

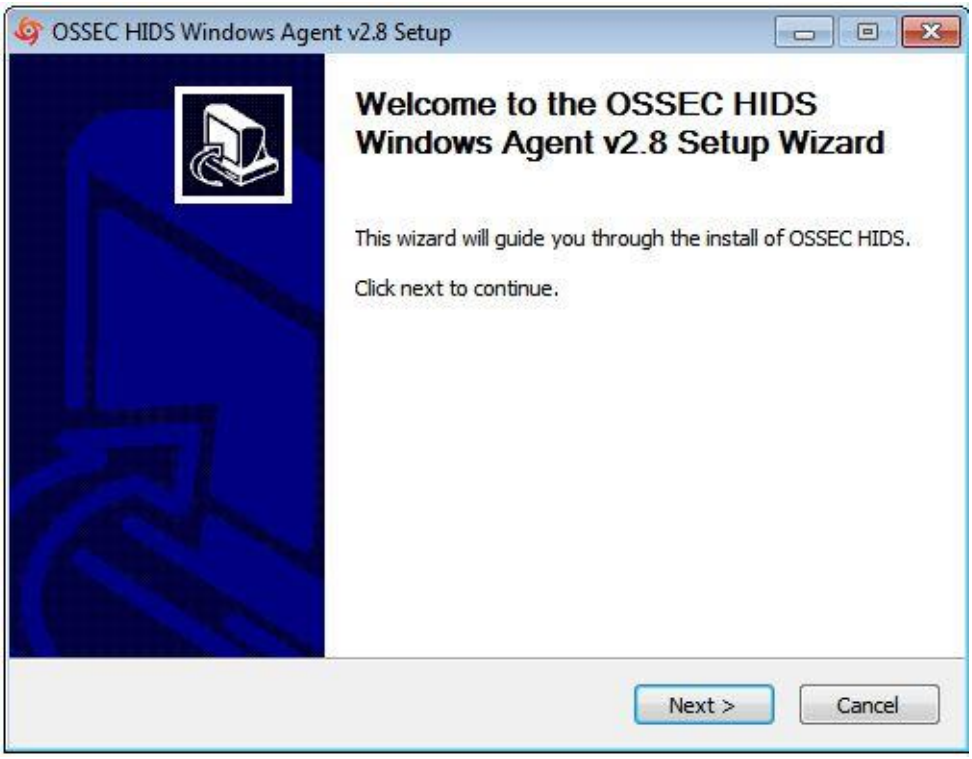
You have chosen to open:

ossec-agent-win32-2.8.exe

which is: Binary File (1.1 MB)
from: http://www.ossec.net

Would you like to save this file?

Picture 3: Click on the file then click on Next

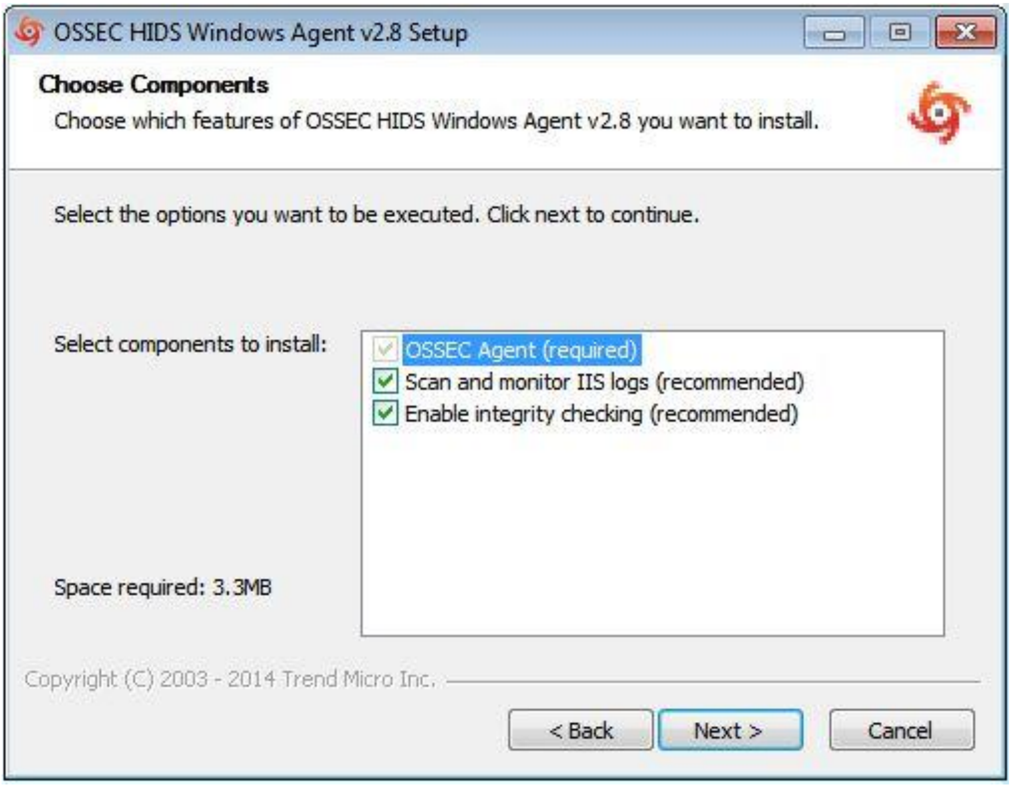


Welcome to the OSSEC HIDS Windows Agent v2.8 Setup Wizard

This wizard will guide you through the install of OSSEC HIDS.

Click next to continue.

Picture 4: Select all components



Choose Components

Choose which features of OSSEC HIDS Windows Agent v2.8 you want to install.

Select the options you want to be executed. Click next to continue.

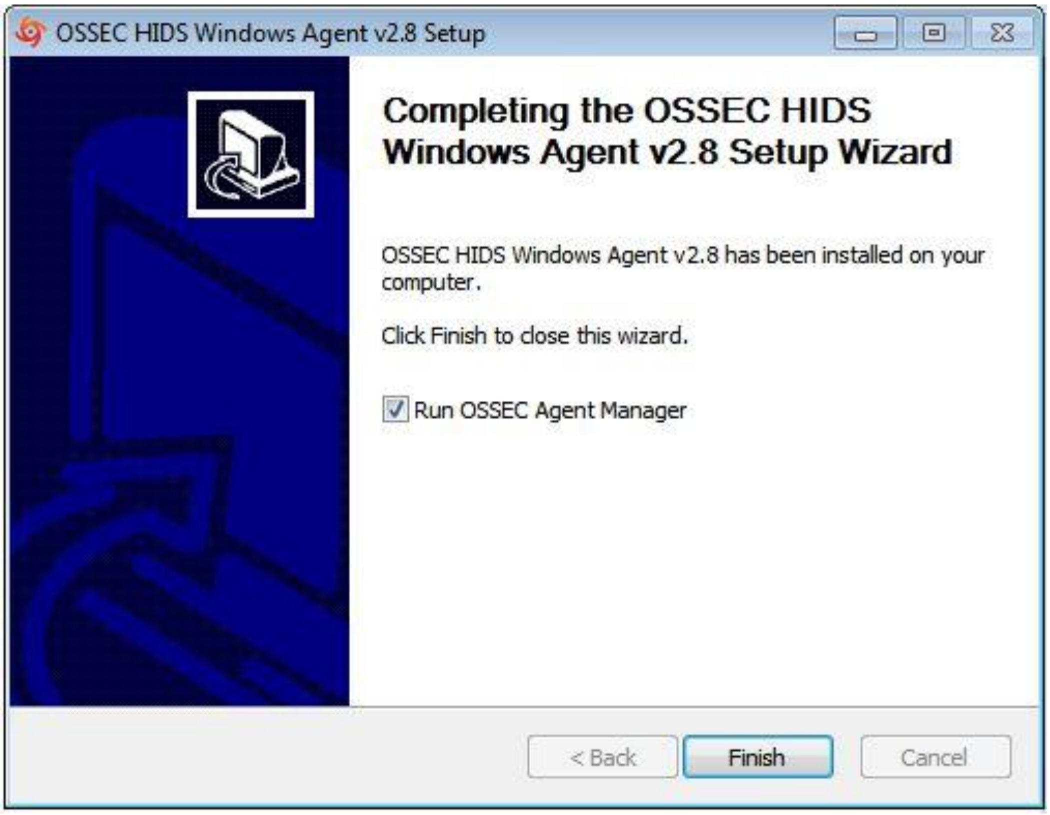
Select components to install:

- ☑ OSSEC Agent (required)
- ☑ Scan and monitor IIS logs (recommended)
- ☑ Enable integrity checking (recommended)

Space required: 3.3MB

Copyright (C) 2003 - 2014 Trend Micro Inc.

Picture: 5 click on Finish



OSSEC HIDS Windows Agent v2.8 Setup

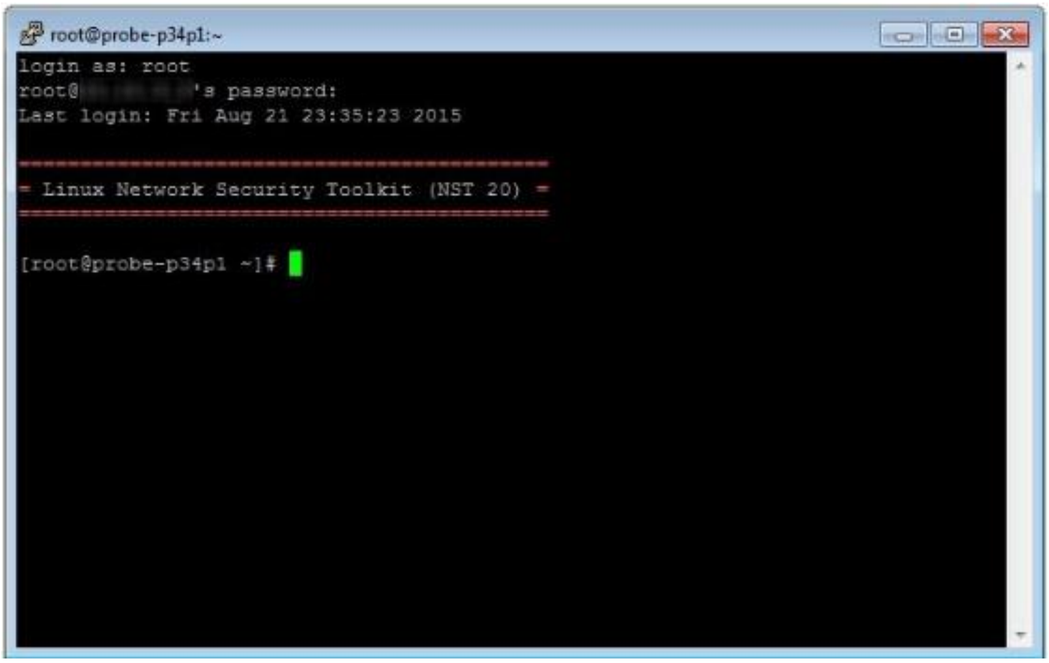
Completing the OSSEC HIDS Windows Agent v2.8 Setup Wizard

OSSEC HIDS Windows Agent v2.8 has been installed on your computer.

Click Finish to close this wizard.

☑ Run OSSEC Agent Manager

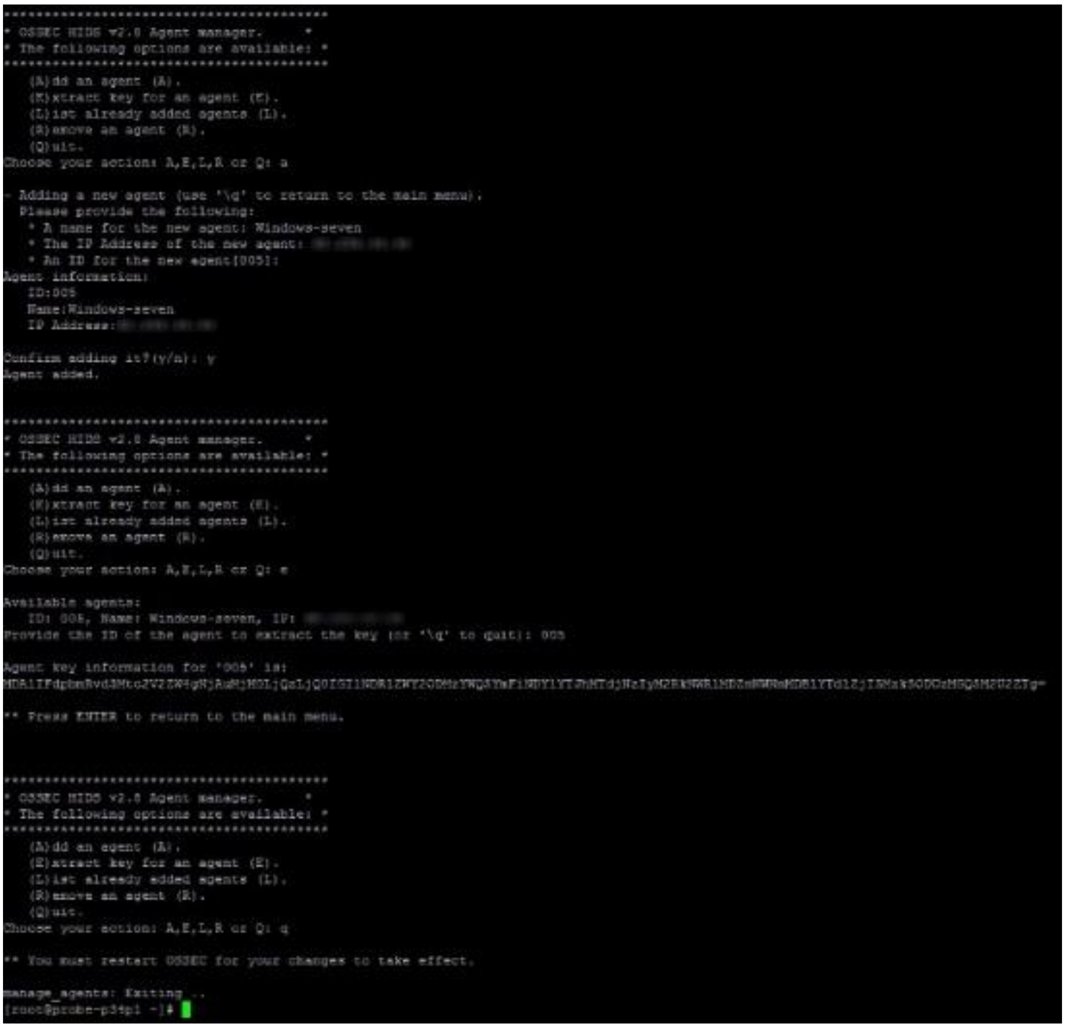
Picture: 6 run putty



root@probe-p34p1:~

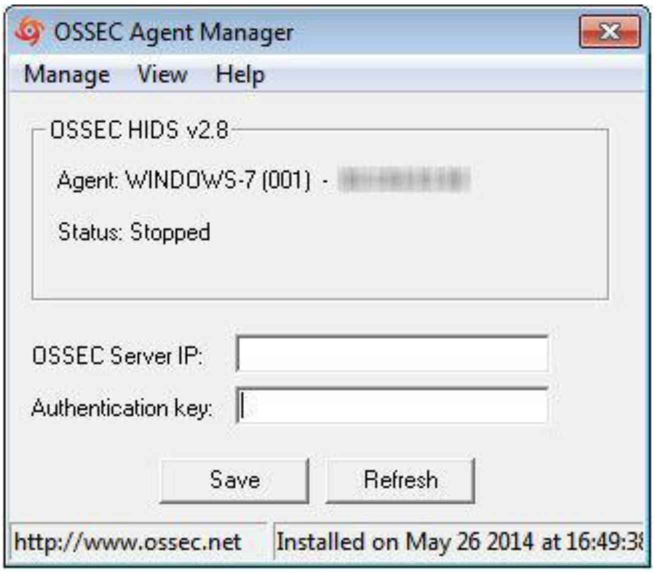
```
login as: root
root@[redacted]'s password:
Last login: Fri Aug 21 23:35:23 2015

===========================================
= Linux Network Security Toolkit (NST 20) =
===========================================

[root@probe-p34p1 ~]#
```

Picture: 7 access the sever and copy the password



```
****************************************
* OSSEC HIDS v2.8 Agent manager.     *
* The following options are available: *
****************************************
   (A)dd an agent (A).
   (E)xtract key for an agent (E).
   (L)ist already added agents (L).
   (R)emove an agent (R).
   (Q)uit.
Choose your action: A,E,L,R or Q: a

- Adding a new agent (use '\q' to return to the main menu).
  Please provide the following:
   * A name for the new agent: Windows-seven
   * The IP Address of the new agent: [redacted]
   * An ID for the new agent[005]:
Agent information:
   ID:005
   Name:Windows-seven
   IP Address:[redacted]

Confirm adding it?(y/n): y
Agent added.


****************************************
* OSSEC HIDS v2.8 Agent manager.     *
* The following options are available: *
****************************************
   (A)dd an agent (A).
   (E)xtract key for an agent (E).
   (L)ist already added agents (L).
   (R)emove an agent (R).
   (Q)uit.
Choose your action: A,E,L,R or Q: e

Available agents:
   ID: 005, Name: Windows-seven, IP: [redacted]
Provide the ID of the agent to extract the key (or '\q' to quit): 005

Agent key information for '005' is:
MDA1IFdpbmRvd3Mtc2V2ZW4gNjAuMjM0LjQzLjQ0IGI1NDR1ZWY2ODMzYWQ3YmFiNDY1YTJhMTdjNzIyM2RkNWR1MDZmNWRmMDB1YTdlZjI5Mzk5ODOzMGQ3M2U2ZTg=

** Press ENTER to return to the main menu.


****************************************
* OSSEC HIDS v2.8 Agent manager.     *
* The following options are available: *
****************************************
   (A)dd an agent (A).
   (E)xtract key for an agent (E).
   (L)ist already added agents (L).
   (R)emove an agent (R).
   (Q)uit.
Choose your action: A,E,L,R or Q: q

** You must restart OSSEC for your changes to take effect.

manage_agents: Exiting ..
[root@probe-p34p1 ~]#
```

Picture: 8 click on the OSSEC Agent Manager



Agent: WINDOWS-7 (001) - [redacted]

Status: Stopped

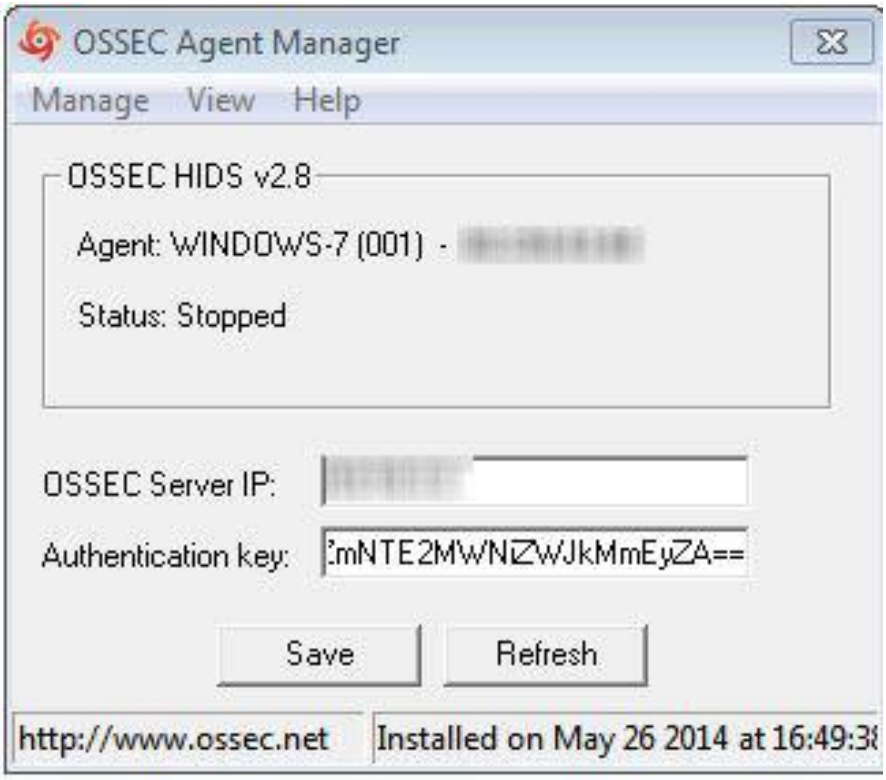
OSSEC Server IP:

Authentication key:

http://www.ossec.net

Installed on May 26 2014 at 16:49:3

Picture: 9insert the IP of the server and type the authentication key



OSSEC HIDS v2.8

Agent: WINDOWS-7 (001) - [redacted]

Status: Stopped

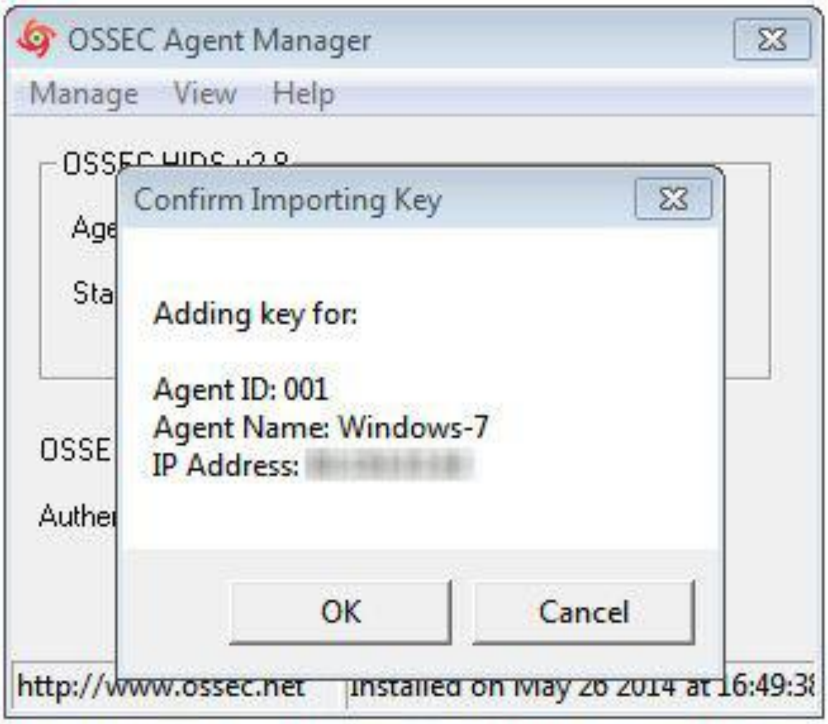
OSSEC Server IP: [redacted]

Authentication key: :mNTE2MWNiZWJkMmEyZA==

http://www.ossec.net

Installed on May 26 2014 at 16:49:3

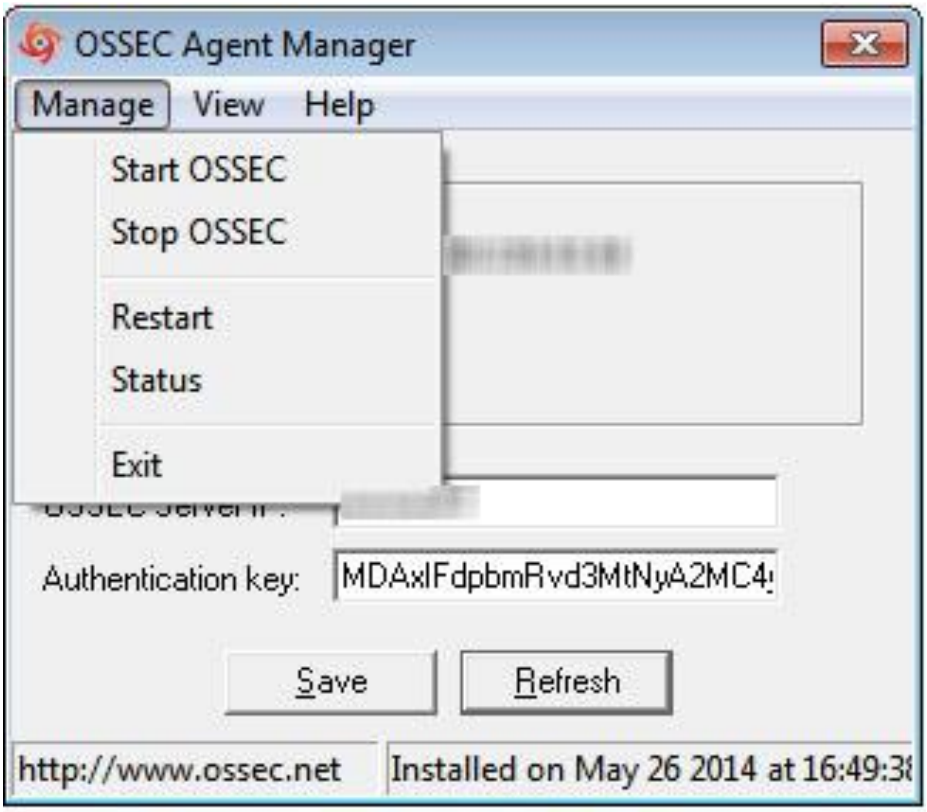
Picture: 10 click on Save



Adding key for:

Agent ID: 001
Agent Name: Windows-7
IP Address: [redacted]

Picture: 11 click on Start OSSEC



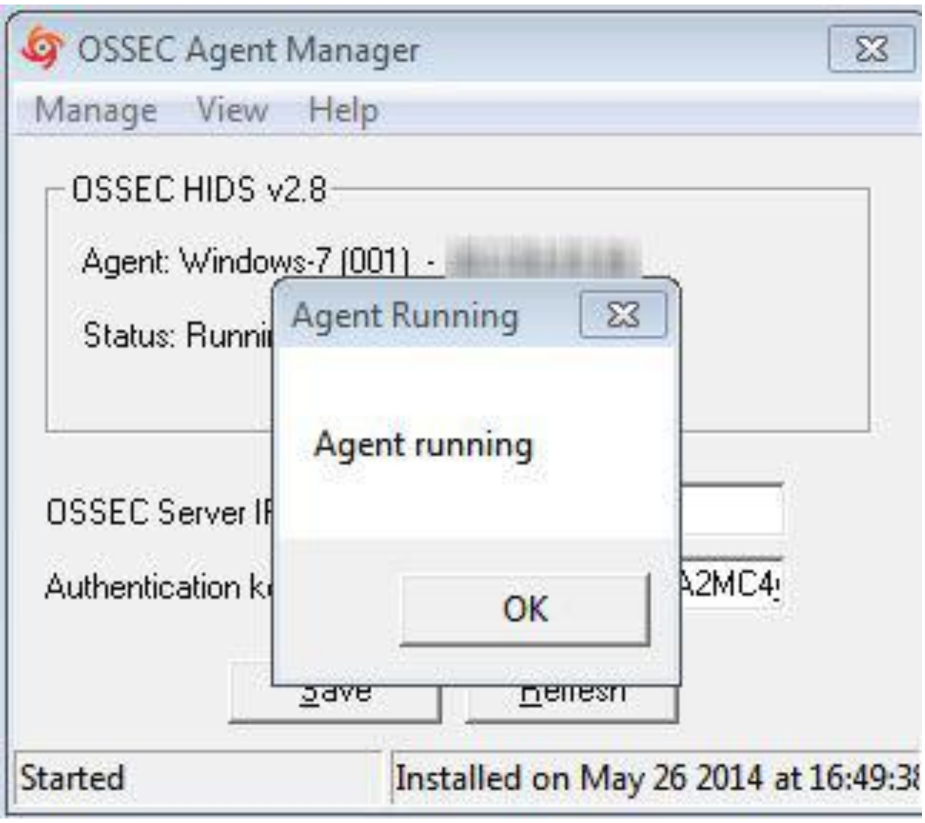
OSSEC Server IP: [redacted]

Authentication key: MDAxlFdpbmRvd3MtNyA2MC4

http://www.ossec.net

Installed on May 26 2014 at 16:49:3

Picture: 12 make sure the Agent is running



OSSEC HIDS v2.8

Agent: Windows-7 (001) - [redacted]

Status: Runni

Agent running

OSSEC Server IP

Authentication k

A2MC4

Started

Installed on May 26 2014 at 16:49:3

APPENDIX 3 PRELUDE INSTALLATION

In order to install Prelude, there are some software or packages that should be downloaded first such as the following packages:

These packages should be installed before install Prelude.

```
[root@Emad-Forensic-Server ~] # yum install tcl setroubleshoot tcltk git gcc-c++
automake autoconf boost-devel cairo-devel libpcap-devel zlib-devel p7zip zip
bzip2.x86_64 audispd-plugins.x86_64 derrick openssl libcap glibc.x86_64 glibc-
devel tcllib tcl-devel glibc wireshark-gnome.x86_64 tk graphviz-tcl tclx tk iwidgets
gcc tcl-devel openssl-devel tclx tk glibc. i686 libX11.i686
```

```
[root@Emad-Forensic-Server ~] # yum install mysql-server MySQL mysql-libs
prewikka audispd-plugins
```

Then download and install the Prelude packages:

```
[root@Emad-Forensic-Server ~] # yum install libprelude libprelude-python
libpreludedb libpreludedb-mysql libpreludedb-python prelude-lml prelude-manager
prelude-manager-db-plugin prelude-correlator prelude-notify
```

Login into database in order to create a user that can manage the database of prelude

```
[root@Emad-Forensic-Server ~] # mysql -u root -p
```

```
mysql> CREATE database prelude;
Query OK, 1 rows affected (0.00 sec)
```

```
mysql> grant all privileges on prelude. * to prelude@'localhost' identified by 'sguil';
Query OK, 1 rows affected (0.00 sec)
```

Then I have to create tables inside the prelude database

```
[root@Emad-Forensic-Server ~] # mysql -u prelude -p prelude <
/usr/share/libpreludedb/classic/mysql.sql
```

To ensure the tables has been created, I have to login into mysql and check by issues these commands

```
[root@Emad-Forensic-Server ~] # mysql -u prelude -p prelude
[root@Emad-Forensic-Server ~] # Enter password:
Reading table information for completion of table and column names
You can turn off this feature to get a quicker start up with -A

Welcome to the MySQL monitor.  Commands end with; or \g.
Your MySQL connection id is 9
Server version: 5.6.27-0ubuntu0.15.04.1 (Ubuntu)

Copyright (c) 2000, 2015, Oracle and/or its affiliates. All rights reserved.

Oracle is a registered trademark of Oracle Corporation and/or its affiliates. Other
names may be trademarks of their respective owners.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

mysql> show databases;
+--------------------+
| Database           |
+--------------------+
| information_schema |
| prelude            |
+--------------------+
2 rows in set (0.00 sec)

mysql> use prelude;
Database changed
mysql> show tables;
+---------------------------------------+
| Tables_in_prelude                     |
+---------------------------------------+
| Prelude_Action                        |
| Prelude_AdditionalData                |
| Prelude_Address               |
| Prelude_Alert                 |
| Prelude_Alertident            |
| Prelude_Analyzer              |
| Prelude_AnalyzerTime          |
| Prelude_Assessment            |
| Prelude_Checksum              |
| Prelude_Classification        |
| Prelude_Confidence            |
| Prelude_CorrelationAler t     |
| Prelude_CreateTime            |
| Prelude_DetectTime            |
| Prelude_File                  |
| Prelude_FileAccess            |
| Prelude_FileAccess_Permission |
| Prelude_Heartbeat             |
| Prelude_Impact                |
| Prelude_Inode                 |
| Prelude_Linkage               |
| Prelude_Node                  |
| Prelude_OverflowAlert         |
| Prelude_Process               |
| Prelude_ProcessArg            |
| Prelude_ProcessEnv            |
| Prelude_Reference             |
| Prelude_Service               |
| Prelude_SnmpService           |
| Prelude_Source                |
| Prelude_Target                |
| Prelude_ToolAlert             |
| Prelude_User                  |
| Prelude_UserId                |
| Prelude_WebService            |
| Prelude_WebServiceArg         |
| _format                       |
+-------------------------------+
37 rows in set (0.00 sec)

mysql> \q
Bye
```

Now I have to create a user that can manage the prelude

```
[root@Emad-Forensic-Server ~] # prelude-admin add prelude-manager --uid 0 --gid 0
Generating 2048 bits RSA private key... This might take a very long time.
[Increasing system activity will speed-up the process].
Generation in progress...

Created profile 'prelude-manager' with analyzerID '1480274811192849'.
```

Now I have to restart the prelude-manager by using two panels the first one will be acting the server and the other will be acting as client

In panel one: -

```
[root@Emad-Forensic-Server ~] # /etc/init. d/prelude-manager restart
[Ok] Restarting prelude-manager (via systemctl): prelude-manager. Service.
```

Now I have to register a sensor. This will register a sensor by copying the password from sensor and paste the password at the server panel as following:

```
[root@Emad-Forensic-Server ~] # prelude-admin register prelude-lml 'idmef: w admin: r' 127.0.0.1 --uid 0 --gid 0
Generating 2048 bits RSA private key... This might take a very long time.
[Increasing system activity will speed-up the process].
Generation in progress...

You now need to start "prelude-admin" registration-server on 127.0.0.1:
For example: "prelude-admin registration-server prelude-manager"

Enter the one-shot password provided on 127.0.0.1:
```

In panel two:

```
[root@Emad-Forensic-Server ~] # prelude-admin registration-server prelude-manager
The "6cista0h" password will be requested by "prelude-admin register"
in order to connect. Please remove the quotes before using it.
```

```
Generating 1024 bits Diffie-Hellman key for anonymous authentication...
Waiting for peers install request on 0.0.0.0:5553...
Waiting for peers install request on:5553…
```

After pasting the password

In panel one “Client”

```
Enter the one-shot password provided on 127.0.0.1:
Confirm the one-shot password provided on 127.0.0.1:

Connecting to registration server (127.0.0.1:5553) ... Authentication succeeded.
```

In panel two “SERVER”

```
Connection from 127.0.0.1:43584...
Registration request for analyzerID="2363504770842041" permission="idmef: w admin: r".
Approve registration? [y/n]: y
```

The registration must be approved by the client BY TYPING Y.

The registration has been completed.

Now we can start the prelude-lml

```
[root@Emad-Forensic-Server ~] #/etc/init. d/prelude-lml start
Disabling temporarily Selina...
Starting prelude-lml: [OK]
Restoring SELinux...
```

This command will show which services of prelude are running

```
[root@Emad-Forensic-Server ~] # ps -ef | grep prel
root     22232     1 0 05:21?       00:00:00 prelude-manager -d
root     22546     1 0 05:32?       00:00:00 /usr/bin/prelude-lml -d -P /var/run/prelude-lml.pid
root     22562 22027 0 05:33 pts/2    00:00:00 grep prel
```

```
[root@Emad-Forensic-Server ~] # prelude-admin list
```

```
Profile                Permission    Issuer       AnalyzerID
------------------------------------------------------------------------------
prelude-manager    n/a                n/a
prelude-lml           idmef: w          admin: r     1480274811192849
```

APPENDIX 4 PADS WITH SGUIL

This section describes how to compile software on your compiler.
Add more packages:

```
# yum install gcc gcc-c++ make flex bison tcllib
# yum install pcre-devel zlib-devel libpcap-devel
# yum install tcl-devel automake libtool
# yum install mysql-devel mysqltcl
# yum install tcl initscripts-systemd
```

libdnet – for sensor

Download libdnet-1.12.tgz from http://libdnet.googlecode.com/files/libdnet-1.12.tgz

```
# tar zxvf libdnet-1.12.tgz
# rm libdnet-1.12.tgz
# cd libdnet-1.12
#. /configure --prefix=/usr/local/libdnet-1.12
# make
# make install
# ln -s /usr/local/libdnet-1.12 /usr/local/libdnet
```

DAQ – for sensor

Download daq-2.0.4.tar.gz from https://www.procyonlabs.com/mirrors/snort/

```
# cd /usr/src/forensic
# tar zxvf daq-2.0.4.tar.gz
# rm daq-2.0.4.tar.gz
# cd daq-2.0.4
#. /configure --prefix=/usr/local/daq-2.0.4 --with-dnet-includes=/usr/local/libdnet/include --with-dnet-libraries=/usr/local/libdnet/lib
# make
# make install
# ln -s /usr/local/daq-2.0.4 /usr/local/daq
# PATH=/usr/local/daq/bin: $PATH
```

Note: PATH command is needed for compiling Snort next.

Snort – for sensor

Download snort-2.9.7.2.tar.gz from http://sourceforge.net/projects/snort/files/OLD%20STUFF%20THAT%20YOU%20SHOULDNT%20USE/

```
# tar zxvf snort-2.9.7.2.tar.gz
# rm snort-2.9.7.2.tar.gz
# cd snort-2.9.7.2
#. /configure --prefix=/usr/local/snort-2.9.7.2 --with-dnet-includes=/usr/local/libdnet/include --with-dnet-libraries=/usr/local/libdnet/lib --with-daq-includes=/usr/local/daq/include --with-daq-libraries=/usr/local/daq/lib
# make
# make install
# ln -s /usr/local/snort-2.9.7.2 /usr/local/snort
# ln -s /usr/local/snort/bin/snort /usr/local/bin/
```

Sguil – for server/sensor/client

Download sguil-0.8.0.tar.gz from http://sourceforge.net/projects/sguil/files/sguil/sguil-0.8.0/

```
# tar xzvf sguil-0.8.0.tar.gz
# rm sguil-0.8.0.tar.gz
```

There is no need to compile. Files will be used on soruce/server/sensor.

4.6 PADS – for sensor

Download pads-1.3.1.tar.gz from https://github.com/gamelinux/pads/tags

```
# tar xzvf pads-1.3.1.tar.gz
# rm pads-1.3.1.tar.gz
# cd pads-1.3.1/
#. /configure --prefix=/usr/local/pads-1.3.1
# make
# make install
# ln -s /usr/local/pads-1.3.1 /usr/local/pads
# ln -s /usr/local/pads/bin/pads /usr/local/bin/
```

4.7 SANCP – for sensor

Download sancp-1.6.1-stable.tar.gz from http://sourceforge.net/projects/sancp/

```
# tar xzvf sancp-1.6.1-stable.tar.gz
# rm sancp-1.6.1-stable.tar.gz
# cd sancp-1.6.1-stable/
```

Modify lines in Makefile file as following:

```
# LINUX and BSD CFLAGS
CFLAGS = -O3 -I/usr/include/pcap -L/usr/lib64 -I./ -L/usr/lib/libsocket.so -g -L/opt/csw/lib -ggdb
# LINUX  LFLAGS
LFLAGS = -lresolv -lnsl -lpcap -L/usr/lib64/libpcap.so
```

Continue:

```
# make linux
# mkdir -p /usr/local/sancp-1.6.1-stable/bin
# cp sancp /usr/local/sancp-1.6.1-stable/bin
# ln -s /usr/local/sancp-1.6.1-stable /usr/local/sancp
# ln -s /usr/local/sancp/bin/sancp /usr/local/bin/
```

mysqltcl – for server

Download mysqltcl-3.052.tar.gz from http://www.xdobry.de/mysqltcl/#downloads

```
# tar xzvf mysqltcl-3.052.tar.gz
# rm mysqltcl-3.052.tar.gz
# cd mysqltcl-3.052/
#. /configure --prefix=/usr/local/mysqltcl-3.052 --exec-prefix=/usr/local/mysqltcl-3.052 --with-tcl=/usr/lib64 --with-mysql-lib=/usr/lib/mysql
# make
# make install
# ln -s /usr/local/mysqltcl-3.052 /usr/local/mysqltcl
```

Note: –enable-64bit option is currently not working.

tls – for server/client

Download tls1.6.7-src.tar.gz from http://tls.sourceforge.net/

```
# tar xzvf tls1.6.7-src.tar.gz
# rm tls1.6.7-src.tar.gz
# cd tls1.6.7/
#. /configure --prefix=/usr/local/tls1.6.7 --exec-prefix=/usr/local/tls1.6.7 --enable-64bit --with-tcl=/usr/lib --with-ssl-dir=/usr
# make
# make install
# ln -s /usr/local/tls1.6.7 /usr/local/tls1.6.7
```

tcllib – for server

Download tcllib-1.13.tar.gz from http://sourceforge.net/projects/tcllib/files/tcllib/1.13/

```
# tar xzvf tcllib-1.13.tar.gz
# rm tcllib-1.13.tar.gz
# cd tcllib-1.13/
#. /configure --prefix=/usr/local/tcllib-1.13
# make
# make install
# ln -s /usr/local/tcllib-1.13 /usr/local/tcllib
```

p0f – for server

p0f is described as a tool which can fingerprint Operating System passively. There are two methods of detecting the type of Operating System a host is running

Download p0f.tgz from https://packetstormsecurity.com/files/22110/p0f.tgz.html

```
# tar xzvf p0f.tgz
# rm p0f.tgz
# cd p0f-1.7/
```

Edit mk/Linux file as following:

```
CLIBS    = -lpcap -I/user/include -L/usr/lib64
```

Continue:

```
# make
# mkdir -p /usr/local/p0f-2.0.8/sbin
# cp p0f /usr/local/p0f-2.0.8/sbin
# ln -s /usr/local/p0f-2.0.8 /usr/local/p0f
```

tcpflow – for server

Download tcpflow-1.3.0.tar.gz from http://pkgs.fedoraproject.org/repo/pkgs/tcpflow/tcpflow-1.3.0.tar.gz/4912d55b7ae20f4971dfcc86547b2824/

```
# tar xzvf tcpflow-1.3.0.tar.gz
# rm tcpflow-1.3.0.tar.gz
# cd tcpflow-1.3.0/
#. /configure --prefix=/usr/local/tcpflow-1.3.0
# make
# make install
# ln -s /usr/local/tcpflow-1.0.2 /usr/local/tcpflow
```

At this point, you should have all software ready under folder /usr/local/.

Set up Sguil Server

```
# yum install tclx mysql-server
```

Create version-less symbolic Links:

```
# ln -s /usr/local/sguil-0.8.0 /usr/local/sguil
```

Add required packages in tcl:

```
# cp -rp /usr/local/mysqltcl-3.052/lib/mysqltcl-3.052 /usr/lib64/tcl8.5/
# cp -rp /usr/local/tcllib/lib/tcllib1.13 /usr/lib64/tcl8.5/
# cp -rp /usr/local/tls1.6.7/lib/tls1.6.7 /usr/lib64/tcl8.5/
```

Verify packages in tcl:

```
# tclsh
% package require Tclx
8.4
% package require mysqltcl
3.05
% package require sha1
2.0.3
% exit
```

```
# useradd -u 400 -d /home/sguil -c "SGUIL User" sguil
# passwd sguil
# mkdir -p /forensic/sguild_data/archive
# mkdir -p /forensic/sguild_data/rules
# mkdir -p /forensic/sguild_data/load
# chown -R sguil. sguil /forensic/sguild_data
```

Set up Database

Configure MySQL database server:

```
# useradd -u 27 -d /var/lib/mysql -s /bin/bash -c "MySQL Server" mysql
# mkdir /forensic/mysql
# chown -R mysql. Mysql /forensic/mysql
# chmod 755 /forensic/mysql
# rm -rf /var/lib/mysql
# ln -s /forensic/mysql /var/lib/mysql
```

Start mysqld:

```
# chkconfig --level 345 mysqld on
# /usr/bin/mysql_install_db --user=mysql
# service mysqld start
Starting MySQL: [OK]
```

Verify mysqld is working:

```
# mysqladmin ping
mysqld is alive
```

Create database users (replace “password” and “sguil_password” with your own passwords):

```
# mysql -u root mysql
mysql> update user set Password = PASSWORD('sguil') where User = 'root';
mysql> flush privileges;
mysql> exit
```

```
# mysql -u root -p mysql
mysql> GRANT ALL PRIVILEGES ON sguildb.* TO sguil@localhost
IDENTIFIED BY "sguil";
mysql> GRANT FILE ON *.* to sguil@localhost;
mysql> update user set Password = PASSWORD("sguil") where User = "sguil";
mysql> FLUSH PRIVILEGES;
mysql> exit
```

Create Sguil database:

```
# mysql -u sguil -p -e "CREATE DATABASE sguildb"
# mysql -u sguil -p -D sguildb < /usr/local/sguil/server/sql_scripts/create_sguildb.sql
gedit /usr/include/mysql/my_sys.h
```

Verify Sguil database:

```
# mysql -u sguil -p -D sguildb -e "show tables"
Enter password: sguil_password
+-------------------+
| Tables_in_sguildb |
+-------------------+
| history           |
| nessus            |
| nessus_data       |
| pads              |
| portscan          |
| sensor            |
| status            |
| user_info         |
| version           |
+-------------------+
```

Configure Sguil Server

Copy files:

```
# mkdir /var/run/sguil
# chown sguil.sguil /var/run/sguil
# mkdir -p /etc/sguild/certs
# cp /usr/local/sguil/server/sguild.conf /etc/sguild
# cp /usr/local/sguil/server/autocat.conf /etc/sguild
# cp /usr/local/sguil/server/sguild.users /etc/sguild
# cp /usr/local/sguil/server/sguild.queries /etc/sguild
# cp /usr/local/sguil/server/sguild.access /etc/sguild
# cp /usr/local/sguil/server/sguild.email /etc/sguild
# cp /usr/local/sguil/server/sguild.reports /etc/sguild
# chown -R sguil.sguil /etc/sguild
```

Modify /etc/sguild/sguild.conf file:

```
set USER sguil
set GROUP sguil
set SGUILD_LIB_PATH /usr/local/sguil/server/lib
set DEBUG 0
set SENSOR_AGGREGATION_ON 0
set RULESDIR /forensic/sguild_data/rules
set DBPASS "sguil"
set DBUSER sguil
set LOCAL_LOG_DIR /forensic/sguild_data/archive
set TMP_LOAD_DIR /forensic/sguild_data/load
set TCPFLOW "/usr/local/tcpflow/bin/tcpflow"
set P0F 1
set P0F_PATH "/usr/local/p0f/sbin/p0f"
```

Setup certificates for sguil components to communicate to each other:

```
# cd /etc/pki/tls/certs
# make sguild.pem
umask 77 ; \
PEM1=`/bin/mktemp /tmp/openssl.XXXXXX` ; \
PEM2=`/bin/mktemp /tmp/openssl.XXXXXX` ; \
/usr/bin/openssl req -utf8 -newkey rsa:2048 -keyout $PEM1 -nodes -x509 -days 365
-out $PEM2 -set_serial 0 ; \
cat $PEM1 >  sguild.pem ; \
echo ""    >> sguild.pem ; \
cat $PEM2 >> sguild.pem ; \
rm -f $PEM1 $PEM2
Generating a 2048 bit RSA private key
...........................+++
..+++
writing new private key to '/tmp/openssl.xkFofx'
-----
You are about to be asked to enter information that will be incorporated
into your certificate request.
What you are about to enter is what is called a Distinguished Name or a DN.
There are quite a few fields but you can leave some blank
For some fields there will be a default value,
If you enter '.', the field will be left blank.
-----
Country Name (2 letter code) [XX]:nz
State or Province Name (full name) []:Auckland
Locality Name (eg, city) [Default City]:Auckland
Organization Name (eg, company) [Default Company Ltd]:AUT
Organizational Unit Name (eg, section) []:MFIT
Common Name (eg, your name or your server's hostname) []:STS2010
Email Address []:sts2010@yahoo.com
```

View the content of /etc/pki/tls/certs/sguild.pem file. Copy everything between the “BEGIN RSA PRIVATE KEY” line and the “END RSA PRIVATE KEY” line (including these two lines) to a new file called sguild.key under /etc/sguild/certs/ directory. Next, copy everything between the “BEGIN CERTIFICATE” and “END CERTIFICATE” lines (including these two lines) to a new file called sguild.pem under /etc/sguild/certs/ directory.

```
# cd /etc/sguild/certs
# touch sguild.key
# touch sguild.pem
```

Set log permission:

```
# chown -R sguil.sguil /var/log/sguild
```

Configure analyst accounts:

```
# /usr/local/sguil/server/sguild -adduser sguil
```

Note: “sguil” is the user account you will use in Sguil Client later.
Now start sguild:

```
# /usr/local/sguil/server/sguild -P /var/run/sguil/sguild.pid -D
```

Verify Server

Verify your server is working properly:

```
# ps -aef | grep sguil | grep -v grep
sguil     1928     1  0 10:00 ?        00:00:00 tclsh /usr/local/sguil/server/sguild -P
/var/run/sguil/sguild.pid -D
sguil     1932  1928  0 10:00 ?        00:00:00 tclsh /usr/local/sguil/server/sguild -P
/var/run/sguil/sguild.pid -D
sguil     1932  1928  0 10:00 ?        00:00:00 tclsh /usr/local/sguil/server/sguild -P
/var/run/sguil/sguild.pid -D
```

Note: You should see 3 processes above.

Set up Sguil Sensor

Create version-less symbolic Links:

```
# ln -s /usr/local/sguil-0.8.0 /usr/local/sguil
```

Add required packages in tcl:

```
# cp -rp /usr/local/tls1.6.7/lib/tls1.6.7 /usr/lib64/tcl8.5/
```

Verify packages in tcl:

```
# tclsh
% package require Tclx
8.4
% exit
```

Note: You should be able to see the versions of packages as shown above, if not, go back to “Add required packages in tcl” and copy the files again.
Add sguil user and folders:

```
# useradd -u 400 -d /home/sguil -c "SGUIL User" sguil
# mkdir -p /forensic/snort-logs/MYSENSOR/OLD
# mkdir -p /forensic/snort_data/MYSENSOR/dailylogs
# mkdir -p /forensic/snort_data/MYSENSOR/sancp
# chown -R sguil.sguil /forensic/snort-logs /forensic/snort_data
# ln -s /forensic/snort-logs/MYSENSOR /var/log/snort-MYSENSOR
# mkdir /var/run/sguil
# chown sguil.sguil /var/run/sguil
```

Configure Sensor Software

Configuration files are located in folder /etc/sguil/. This guide only shows the part of the configuration setting that needs to be changed. Leave everything else in the file as is even if you don’t see them here.
Copy configuration files:

```
# mkdir /etc/sguil
# cp /usr/local/barnyard2-2-1.13/etc/barnyard2.conf /etc/sguil/
# cp /usr/local/pads/etc/pads.conf /etc/sguil/
# cp /usr/local/sguil/sensor/pads_agent.conf /etc/sguil/
# cp /usr/local/sguil/sensor/pcap_agent.conf /etc/sguil/
# cp /usr/local/sguil/sensor/sancp/sancp.conf /etc/sguil/
# cp /usr/local/sguil/sensor/sancp_agent.conf /etc/sguil/
# cp /usr/local/sguil/sensor/snort_agent.conf /etc/sguil/
```

Configure Barnyard2 by editing /etc/sguil/barnyard2.conf:

```
config reference_file: /usr/local/snortrules/etc/reference.config
config classification_file: /usr/local/snortrules/etc/classification.config
config gen_file: /usr/local/snortrules/etc/gen-msg.map
config sid_file: /usr/local/snortrules/etc/sid-msg.map
config hostname: MYSENSOR
config interface: eth0
#output alert_fast
output alert_syslog: LOG_AUTH LOG_ALERT
output sguil: sensor_name=MYSENSOR
```

Configure PADS by editing /etc/sguil/pads.conf:

```
daemon 1
pid_file /var/run/sguil/pads.pid
interface eth0
network 192.168.1.0/24
output fifo: /forensic/snort_data/MYSENSOR/pads.fifo
```

Note: Replace 192.168.1.0/24 with the network address that your sensor is monitoring.

Configure PADS_agent by editing /etc/sguil/pads_agent.conf:

```
set DEBUG 0
set DAEMON 0
set PID_FILE /var/run/sguil/pads_agent.pid
set SERVER_HOST 127.0.0.1
set HOSTNAME MYSENSOR
set NET_GROUP MYSENSOR
set LOG_DIR /forensic/snort_data
```

Note: Replace 127.0.0.1 with your Sguil server address.

Configure SANCP_agent by editing /etc/sguil/sancp_agent.conf:

```
set DEBUG 0
set DAEMON 0
set PID_FILE /var/run/sguil/sancp_agent.pid
set SERVER_HOST 172.17.0.200
set HOSTNAME MYSENSOR
set NET_GROUP MYSENSOR
set LOG_DIR /forensic/snort_data
```

Note: Replace 172.17.0.200 with your Sguil server address.

For snort, we will use /usr/local/snortrules/ folder to hold rules:

```
# mkdir /usr/local/snortrules/
```

Download snort rules *.tar.gz file and place in /usr/local/snortrules folder:

```
# tar xzvf snortrules-snapshot-2912.tar.gz
# touch /usr/local/snortrules/rules/white_list.rules
# touch /usr/local/snortrules/rules/black_list.rules
# mkdir /usr/local/snort-2.9.7.2/lib/snort_dynamicrules
# cp /usr/local/snortrules/so_rules/precompiled/RHEL-6-0/x86-64/2.9.7.5/*
/usr/local/snort-2.9.7.2/lib/snort_dynamicrules/
# cd /usr/local/snortrules/etc/
```

Configure Snort by editing /usr/local/snortrules/etc/snort.conf file. This can be tweaked to suit your needs. Here are the lines you must modify to run snort properly:

```
###################################################
# Step #1: Set the network variables.
###################################################
ipvar HOME_NET 192.168.1.0/24
var WHITE_LIST_PATH /usr/local/snortrules/rules
var BLACK_LIST_PATH /usr/local/snortrules/rules

###################################################
# Step #4: Configure dynamic loaded libraries.
###################################################
dynamicpreprocessor directory /usr/local/snort/lib/snort_dynamicpreprocessor/
dynamicengine /usr/local/snort-2.9.7.2/lib/snort_dynamicengine/libsf_engine.so
dynamicdetection directory /usr/local/snort/lib/snort_dynamicrules

###################################################
# Step #5: Configure preprocessors
###################################################
preprocessor perfmonitor: time 300 file /forensic/snort_data/MYSENSOR/snort.stats
pktcnt 10000

###################################################
# Step #6: Configure output plugins
###################################################
#unified2
output unified2: filename emerged.log, limit 128
```

Once you are done with the rule set, copy *.rules to your IDS server under directory /forensic/sguild_data/rules/MYSENSOR/. This allows you to see the rules in Sguil client console.

Note: Snort rule update is not addressed here. This guide just provides you with basic and manual way to set up snort rules. There are tools available to automate the process, such as Pulled_pork, or Oinkmaster. You could also write your own scripts.

Configure snort_agent by editing /etc/sguil/snort_agent.conf:

```
set DEBUG 0
set DAEMON 0
set PID_FILE /var/run/sguil/snort_agent.pid
set SERVER_HOST 172.17.0.200
set HOSTNAME MYSENSOR
set NET_GROUP MYSENSOR
set LOG_DIR /forensic/snort_data
set PORTSCAN 0
set PORTSCAN_DIR ${LOG_DIR}/${HOSTNAME}/portscans
set SNORT_PERF_STATS 1
set SNORT_PERF_FILE "${LOG_DIR}/${HOSTNAME}/snort.stats"
```

Note: Replace 172.17.0.200 with your Sguil server address.

Copy shell script to run Snort full packet capture:

```
# cp /usr/local/sguil/sensor/log_packets.sh /etc/sguil/
```

Modify /etc/sguil/log_packets.sh file:

```
HOSTNAME="MYSENSOR"
```

```
SNORT_PATH="/usr/local/bin/snort"
LOG_DIR="/forensic/snort_data"
MAX_DISK_USE=90
INTERFACE="eth0"
OPTIONS="-u sguil -g sguil -m 122"
PIDFILE="/var/run/sguil/snort_log.pid"
```

Run Sensor Software

Note: If you prefer, you can jump over to Section 8 “Startup Scripts” instead of running commands manually. The “Startup Scripts” does the same thing as described below. After you are done with “Startup Scripts”, come back to section 6.4 to verify.

Start Sancp:

```
# /usr/local/bin/sancp -d /forensic/snort_data/MYSENSOR/sancp -i eth0 -u sguil -g
sguil -c /etc/sguil/sancp.conf -D
```

Start Sancp_agent:

```
# tclsh /usr/local/sguil/sensor/sancp_agent.tcl -D -c /etc/sguil/sancp_agent.conf
```

Start PADS:

```
# /usr/local/bin/pads -c /etc/sguil/pads.conf -u sguil -g sguil
```

Start PADS_agent:

```
# tclsh /usr/local/sguil/sensor/pads_agent.tcl -D -c /etc/sguil/pads_agent.conf
```

Start pcap_agent:

```
# tclsh /usr/local/sguil/sensor/pcap_agent.tcl -D -c /etc/sguil/pcap_agent.conf
```

Start Snort:

```
# /usr/local/bin/snort -u sguil -g sguil -m 122 -l /var/log/snort-MYSENSOR -c
/usr/local/snortrules/etc/snort.conf -D -i eth0 -q -A none -U --pid-path /var/run/sguil
```

Start Snort_agent:

```
# tclsh /usr/local/sguil/sensor/snort_agent.tcl -D -c /etc/sguil/snort_agent.conf
```

Start Snort full packet capture:

```
# /etc/sguil/log_packets.sh
```

Verify Sensor

Verify your sensor is working properly:

```
# ps -aef | grep sguil | grep -v grep
sguil     1901     1  0 10:00 ?        00:00:00 tclsh /usr/local/sguil/sensor/pads_agent.tcl
-D -c /etc/sguil/pads_agent.conf
sguil     1903 1901 0 10:00 ? 00:00:00 cat
/forensic/snort_data/MYSENSOR/pads.fifo
sguil     1911     1  0 10:00 ?  00:00:00 tclsh /usr/local/sguil/sensor/pcap_agent.tcl -D
-c /etc/sguil/pcap_agent.conf
sguil     1920     1  0 10:00 ?        00:00:00 tclsh
/usr/local/sguil/sensor/sancp_agent.tcl -D -c /etc/sguil/sancp_agent.conf
sguil     1945     1  0 10:00 ?        00:00:00 tclsh /usr/local/sguil/sensor/snort_agent.tcl
-D -c /etc/sguil/snort_agent.conf
sguil     1949  1945  0 10:00 ?        00:00:00 tail -n 1 -f
/forensic/snort_data/MYSENSOR/snort.stats
sguil     1954     1  0 10:00 ?        00:00:00 /usr/local/bin/barnyard2 -c
/etc/sguil/barnyard2.conf -f merged.log --pid-path /var/run/sguil -w /var/log/snort-
MYSENSOR/waldo2.file -l /var/log/snort-MYSENSOR -a /var/log/snort-
MYSENSOR/OLD -d /var/log/snort-MYSENSOR-D
```

```
sguil     1961     1  1 10:00 ?        00:00:00 /usr/local/bin/sancp -d
/forensic/snort_data/MYSENSOR/sancp -i eth0 -u sguil -g sguil -c
/etc/sguil/sancp.conf -D
sguil 1914 1 1 10:00 ? 0:0:0 /usr/local/bin/pads -c /etc/sguil/pads.conf -u sguil -g
sguil
sguil 1934 1 2 10:00 ? 00:00:00 /usr/local/bin/snort -u sguil -g sguil -m 122 -l
/var/log/snort-MYSENSOR -c /usr/local/snortrules/etc/snort.conf -D -i eth0 -q -A
none -U --pid-path /var/run/sguil
sguil 19581 7 10:00 ?  00:00:00 /usr/local/bin/snort -u sguil -g sguil -m 122 -l
/forensic/snort_data/MYSENSOR/dailylogs/2011-12-01 -b -i eth0
```

Note: You should see 11 processes above.

Set up Sguil Client

Download sguil-client-0.8.0.tar.gz file from http://sourceforge.net/projects/sguil/files/sguil/sguil-0.8.0/ and place under folder /usr/local/.

```
# cd /usr/local/
# tar xzvf sguil-client-0.8.0.tar.gz
# rm sguil-client-0.8.0.tar.gz
# ln -s /usr/local/sguil /usr/local/sguil-0.8.0
# ln -s /usr/local/sguil/client/sguil.tk /usr/local/bin
# cp /usr/local/sguil/client/sguil.conf /root/sguil.conf
```

Modify a few lines in /root/sguil.conf:

```
Set SERVERHOST 172.17.0.200
Set SGUILLIB /usr/local/sguil/client/lib
set TLS_PATH "/usr/lib64/tcl/tls1.6/libtls16.so"
Set DEBUG 0
set EXT_DNS_SERVER 8.8.8.8
set HOME_NET "192.168.1.0/24"
Set WIRESHARK_PATH /usr/bin/wireshark
```

Note: Change the IP addresses to suite your own needs. SERVERHOST should be your Sguil server IP.

Run Sguil client:

```
# /usr/local/bin/sguil.tk
```

Log in with ‘sguiluser’ credentials. Check the sensor you want to monitor. Click “Start Sguil” to enter the console. You should be seeing snort alerts and PADS alerts in the console.

Configure services:

```
# chkconfig --add sancp
# chkconfig --add sancp_agent
# chkconfig --add pads_agent
# chkconfig --add pcap_agent
# chkconfig --add snort_agent
# chkconfig --add sguil_logger
# chkconfig --add snort
# chkconfig --add pads

# chkconfig --level 345 sancp on
# chkconfig --level 345 sancp_agent on
# chkconfig --level 345 pads_agent on
# chkconfig --level 345 pcap_agent on
# chkconfig --level 345 snort_agent on
# chkconfig --level 345 sguil_logger on
# chkconfig --level 345 snort on
# chkconfig --level 345 pads on
```

Schedule to restart sguil packet logger hourly to rotate snort logs:

```
# crontab -e
        # Restart the sguil packet logger on a regular basis
0 * * * * /etc/init.d/sguil_logger restart
```

Reboot Sensor and verify all services start properly.

APPENDIX 5 SPLUNK INSTALLATION

First the package of ld-dlinux has to be download then install and configure in order to install the Splunk otherwise the installation will be failure:

The command to install is:

```
yum install /lib/ld-linux.so.2
```

Then download the plunk and agreement file from Splunk website. Now you should have something similar to:

- Splunk-6.2.5-272645.i386.rpm
- reporting-and-management-for-ossec_1189.tgz

The following command will install the splunk and report-and-agreement:

```
[root@Emad-Forensic-Server ~]# rpm -i Splunk-6.2.5-272645.i386.rpm
```

The agreement file should be in folder apps in splunk directory.

```
[root@Emad-Forensic-Server ~]# mv reporting-and-management-for-ossec_1189.tar /opt/Splunk/etc/apps/
[root@Emad-Forensic-Server ~]# cd /opt/Splunk/etc/apps/
[root@Emad-Forensic-Server ~]# tar -xvf reporting-and-management-for-ossec_1189.tar
[root@Emad-Forensic-Server ~]# chown -R Splunk:Splunk ossec
```

The following command enter my agreements of use terms.

```
[root@Emad-Forensic-Server ~]#./Splunk start --accept-license
```

The following command will enable boot when the machine restart

```
[root@Emad-Forensic-Server ~]# /opt/Splunk/bin/Splunk enable boot-start
[root@Emad-Forensic-Server ~]# /opt/Splunk/bin/Splunk start
```

The user name is admin and the password is changeme

APPENDIX 6 PREWIKKA INSTALLATION

Now the GUI should be checked in order to ensure all alerts are save and manipulate in Prewikka

```
[root@Emad-Forensic-Server ~]# MySQL -u root -p
Enter password:
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 59
Server version: 5.1.73 Source distribution

Copyright (c) 2000, 2013, Oracle and/or its affiliates. All rights reserved.

Oracle is a registered trademark of Oracle Corporation and/or its affiliates. Other names may be trademarks of their respective owners.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

mysql> create database prewikka;
Query OK, 1 row affected (0.00 sec)

mysql> grant all privileges on prewikka.* to prewikka@'localhost' identified by 'sguil';
Query OK, 0 rows affected (0.00 sec)

mysql> quit
```

This command will create the scheme of the prewikka database

```
[root@Emad-Forensic-Server ~]# mysql -u root -p prewikka < /usr/share/prewikka/database/mysql.sql
```

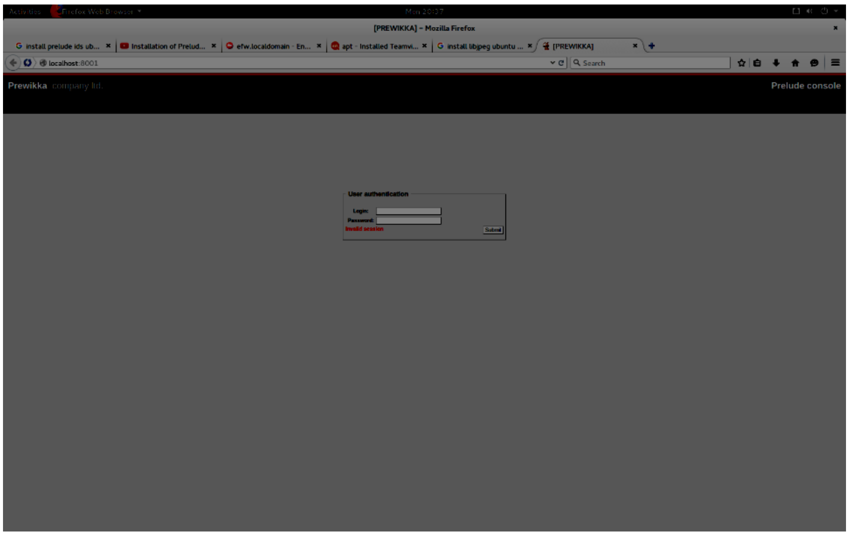
Then login into mysql to check the database scheme by issuing the following commands:

```
[root@Emad-Forensic-Server ~]# mysql -u root -p
Enter password:
Welcome to the MySQL monitor.  Commands end with ; or \g.
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 59
Server version: 5.1.73 Source distribution

Copyright (c) 2000, 2013, Oracle and/or its affiliates. All rights reserved.

Oracle is a registered trademark of Oracle Corporation and/or its affiliates. Other
names may be trademarks of their respective owners.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

mysql> show databases;
+----------------------------+
| Database             |
+----------------------------+
| information_schema  |
| mysql                |
| performance_schema |
| prelude              |
| prewikka            |
+---------------------------+
5 rows in set (0.00 sec)

mysql> use prewikka;
Reading table information for completion of table and column names
You can turn off this feature to get a quicker startup with -A

Database changed
mysql> show tables;

+-------------------------------------+
| Tables_in_prewikka          |
+-------------------------------------+
| Prewikka_Filter              |
| Prewikka_Filter_Criterion    |
| Prewikka_Permission          |
| Prewikka_Session             |
| Prewikka_User                |
| Prewikka_User_Configuration|
| Prewikka_Version             |
+-------------------------------------+
7 rows in set (0.00 sec)

mysql> \q
BYE
```

Start the prewikka by issuing the following command:

```
prewikka-httpd -p 8001 -a localhost
```

In Firefox type localhost:8001 and hit enter



User authentication

Login:

Password:

Invalid session

The initial user name and password is admin

APPENDIX 7 NTOPNG INSTALLATION

Install required packages:

Install groupinstall

```
        # yum groupinstall
```

Install TCL

```
        # yum install tcl
```

Install libpcap

```
        # yum install libpcap libcap-devel
```

Install Redis Server

```
   # wget http://redis.googlecode.com/files/redis-2.6.13.tar.gz
   # tar zxfv redis-2.6.13.tar.gz
   # cd redis-2.6.13
   # make bit
   # make test
   # make install
```

Install ntopng

```
# wget http://sourceforge.net/projects/ntop/files/ntopng/ntopng-1.1_6932.tgz/download
   # tar zxfv ntopng-1.1_6932.tgz
   # cd ntopng-1.1_6932
   # ./configure
   # make geoip
   # make
   # make install
```

Create configuration files for ntopng

```
   # cd /usr/local/etc/
   # mkdir ntopng
   # cd ntopng
   # vi ntopng.start
```

Put these lines:

```
   --local-network 127.0.0.1
   --interface 1
```

```
   # vi ntopng.pid
```

Put this line:

```
   -G=/var/run/ntopng.pid
```

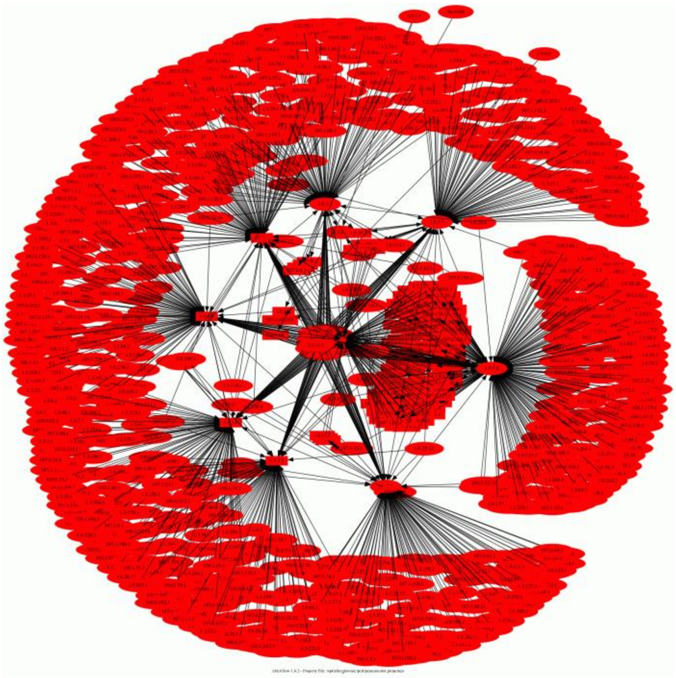
In order to use SSL with ntopng (i.e. HTTPS) you need to do the following commands:

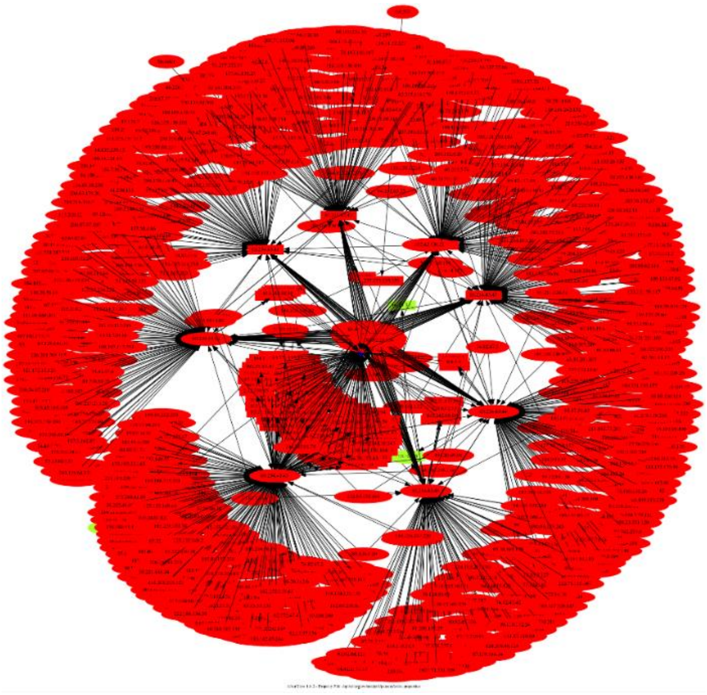
```
cd /tmp/
openssl req -new -x509 -sha256 -extensions v3_ca -nodes -days 365 -out cert.pem
cat privkey.pem cert.pem > /usr/local/share/ntopng/httpdocs/ssl/ntopng-cert.pem
/bin/rm -f privkey.pem cert.pem
cd /usr/local/bin/
ln -s /usr/lib/x86_64-linux-gnu/libssl.so.
```

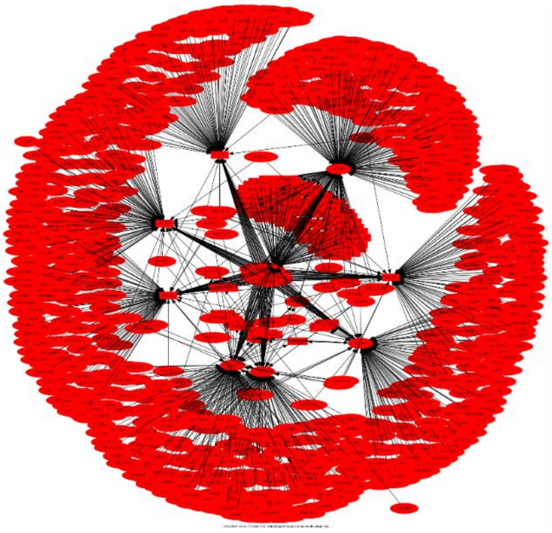
Then run ntopng

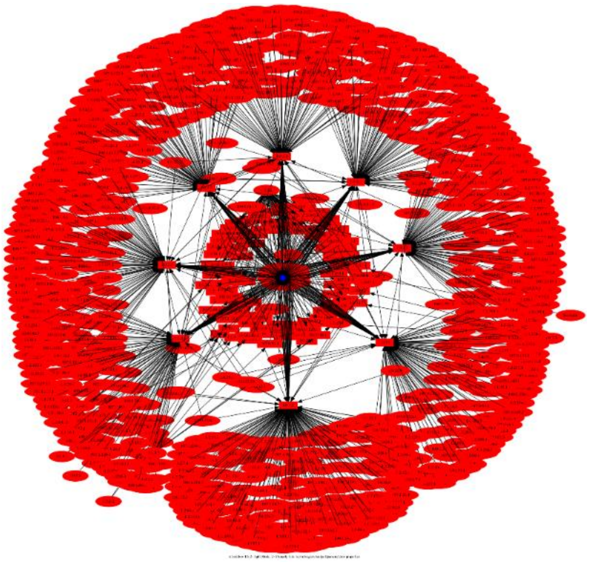
APPENDIX 8 ARGUS INVESTIGATIONS

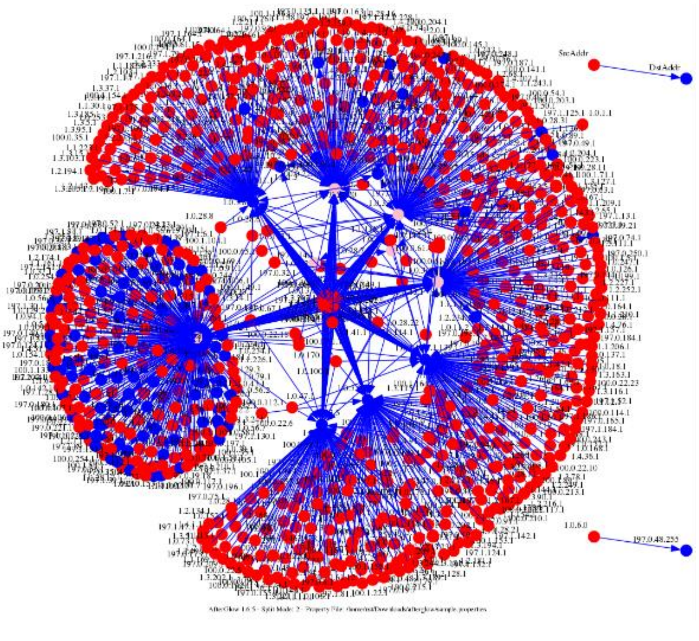
The attack investigation statistics by Argus. The following pictures show all communications that pass through the experimental devices during all attacks. While the center point is the IP of the victims’ computer.







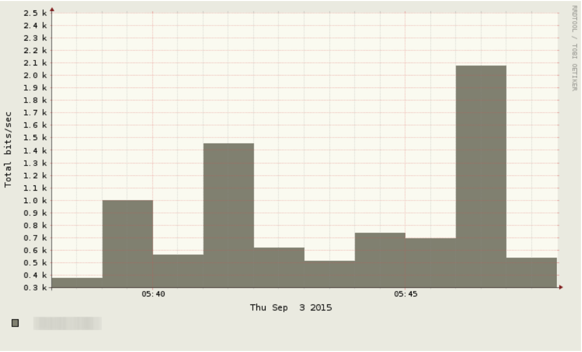



SrcAddr
DstAddr
1.0.6.0
197.0.48.255
AfterGlow 1.6.5 - Split Mode: 2 - Property File: /home/[illegible]/Downloads/afterglow/sample.properties

This picture will present all connection:

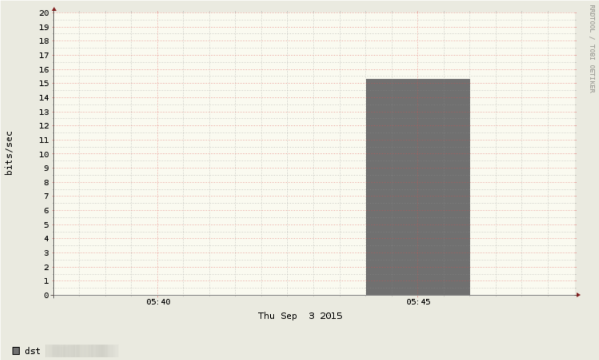
During Reconnaissance Attack (per second)

(ragraph dbytes daddr -M 1s -fill -stack -r argus3.pcap - udp and dst bytes gt 67)



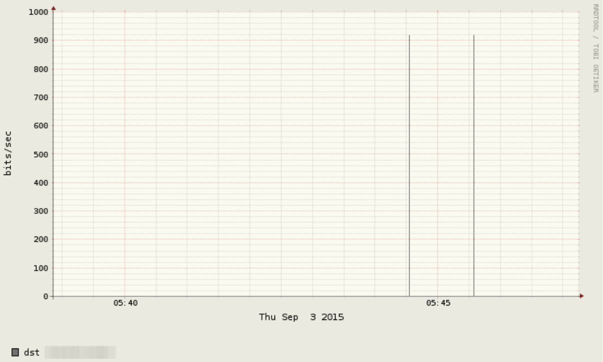
During DDos Attack (per minutes)

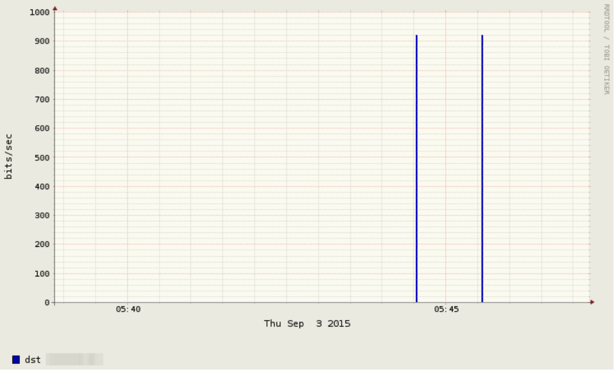
(ragraph bytes daddr -M 1m -fill -stack -r argus3.pcap - udp and dst host xx.xx.xx.xx)



During DDos Attack (per second)

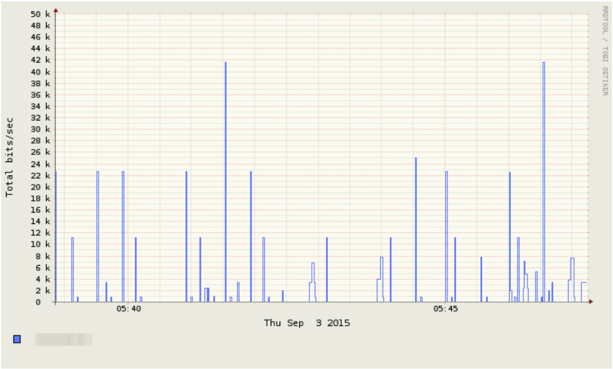
(ragraph bytes daddr -M 1s -fill -stack -r argus3.pcap - udp and dst host xx.xx.xx.xx)





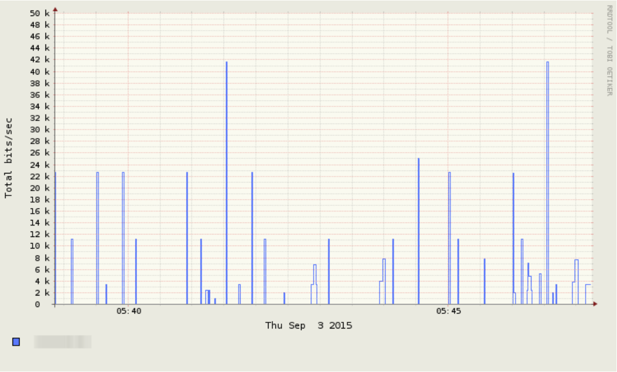
During DDos Attack (per minutes) (ragraph bytes daddr -M 1s -fill -stack -r argus3.pcap - udp and dst host xx.xx.xx.xx)

During Dictionary Attack (per second) (ragraph bytes daddr -M 1s -fill -stack -r argus3.pcap - udp and dst host xx.xx.xx.xx)



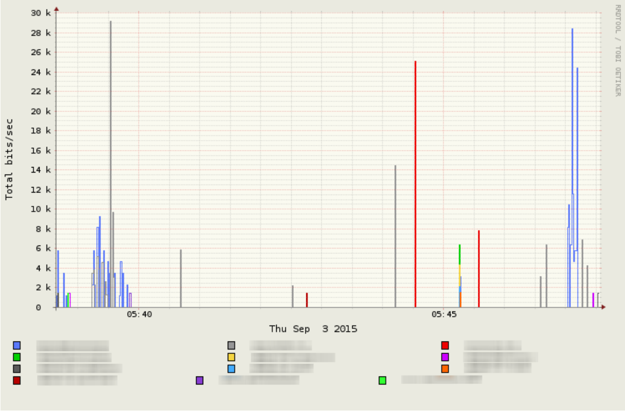
During Dictionary Attack (per minutes) (ragraph bytes daddr -M 1m -fill -stack -r argus3.pcap - udp and dst host xx.xx.xx.xx)

.



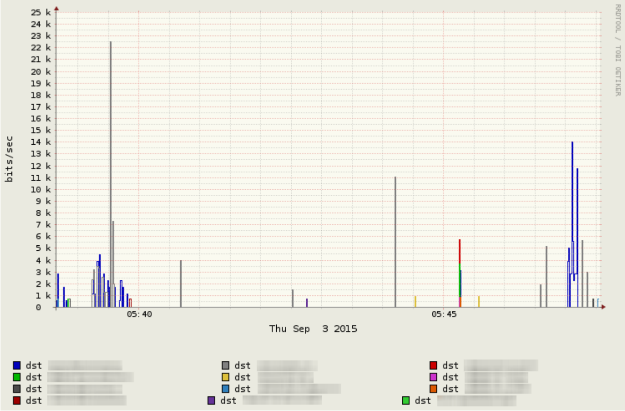
During Packet Sniffing attack (per second)

(ragraph bytes daddr -M 1s -fill -stack -r argus3.pcap - dst host xx.xx.xx.xx)



During Packet Sniffing attack (per minutes)

(ragraph bytes daddr -M 1m -fill -stack -r argus3.pcap - dst host xx.xx.xx.xx)



APPENDIX 9 SQUERT INSTALLATION

1) Extract the squert tarball to a web directory and rename it to "squert"

2) Copy squert/.inc/config.php.sample to squert/.inc/config.php

3) Edit squert/.inc/config.php to match your sguildb and sguild server settings

4) IMPORTANT!! Edit your MySQL server settings to include the following directive:

```
group_concat_max_len = 100000
```

this should be placed in the "[mysqld]" section of my.cnf

Also,

The ip2c TCL scripts uses "LOAD DATA LOCAL INFILE" to dump the results into the database. While most stock MySQL installs are compiled with this, they don't always allow it.

Find the my.cnf that your client is using and add:

```
local-infile=1
```

to the client section. If you just have the client installed and you cannot find this file just create it in /etc and add:

```
[client] local-infile=1
```

Lastly,

You will need to add indexes to the sid and cid columns in Sguils history table:

```
mysql -N -B --user=root -p -e "CREATE INDEX sid ON history (sid);" mysql -N -B --user=root -p -e
"CREATE INDEX cid ON history (cid);"
```

Performance WILL suffer if you do not do this.

5) Create additional tables:

```
cat squert/.scripts/squert.sql | mysql -uroot -p -U sguildb
```

6) Create a mysql user account for squert to access sguildb (what you set in step 3):

```
mysql -N -B --user=root -p -e "GRANT SELECT ON sguildb.* TO 'squert_user'@'localhost'
IDENTIFIED BY 'apassword';"
```

7) Give this user privileges to the ip2c table:

```
mysql -N -B --user=root -p -e "GRANT ALL PRIVILEGES ON sguildb.ip2c TO 'squert_user'@'localhost';"
```

8) Give this user privileges to the mappings table:

```
mysql -N -B --user=root -p -e "GRANT ALL PRIVILEGES ON sguildb.mappings TO 'squert_user'@'localhost';"
```

9) Give this user privileges to the filters table:

```
mysql -N -B --user=root -p -e "GRANT INSERT,UPDATE,DELETE ON sguildb.filters TO 'squert_user'@'localhost';"
```

10) Give this user privileges to sguils user_info table:

```
mysql -N -B --user=root -p -e "GRANT UPDATE ON sguildb.user_info TO 'squert_user'@'localhost';";
```

11) Now populate the ip2c table:

```
squert/.scripts/ip2c.tcl
```

12) Add an index to comment column in Sguils history table:

```
mysql -N -B --user=root -p -e "CREATE INDEX comment ON sguildb.history (comment(50));"
```

13) The read only user needs DELETE access to sguils history table (to delete comments):

```
mysql -N -B --user=root -p -e "GRANT DELETE on sguildb.history to 'readonly'@'localhost';"
```

14) Create a scheduled task to keep the mappings tables up to date:

```
*/5 * * * * /usr/local/bin/php -e /usr/local/www/squert/.inc/ip2c.php 1 > /dev/null 2>&1
```

This entry updates the database every 5 minutes. Make sure you use the correct paths to php and ip2c.php.

15) Create a scheduled task to keep the ip2c table up to date:

```
0 0 1 * * <path_to_squert>/.scripts/ip2c.tcl > /dev/null 2>&1
```

This entry updates the ip2c database on the first day of every month.