Design And Implementation Of Linux Based Network Forensics System Using Virtual Honeynet

Thesis submitted in partial fulfillment of the requirements for the award of degree of

Master Of Technology

Submitted By

Jatinder kaur

(Roll No. 601003010)

Under the Supervision of
Mr. Gurpal Singh Chhabra
Lecturer

SCHOOL OF MATHEMATICS AND COMPUTER APPLICATIONS
THAPAR UNIVERSITY

PATIALA – 147004

JULY 2012

Certificate

I hereby certify that the work which is being presented in the thesis entitled, "Design and Implementation Of Network Forensic System using Honeypot", in partial fulfillment of the requirements for the award of degree of Master of Technology in Computer Science and Applications submitted in School of Mathematics and Computer Applications of Thapar University, Patiala, is an authentic record of my own work carried out under the supervision of Mr.Gurpal Singh Chhabra and refers other researcher's work which are duly listed in the reference section.

The matter presented in the thesis has not been submitted for award of any other degree of this or any other University.

Jatinder Kaur

(Jatinder Kaur)

This is to certify that the above statement made by the candidate is correct and true to the best of my knowledge.

(Mr. Gurpal Singh Chhabra)

Lecturer,

School of Mathematics and Computer Applications

Countersigned by

(Dr. S.S Bhatia)

Head

School of Mathematics and Computer Applications

Thapar University

Patiala

(Dr. S. K. Mohapatra)

Dean (Academic Affairs)

Thapar University

Patiala

ii

Abstract

Use of computer and Internet is now a mandatory perquisite for doing any kind of research, business, transactions, or precisely any daily routine. This thesis report focuses on Network Forensics & its related area. Reports also highlight some of the basic tools that can be used for the purpose of controlling & investigating Cyber crime and network attacks by the hacker community , which is growing exponentially.

With the fast growing crime in the virtual world, tools and techniques to collect preserve and analyze digital data on the Internet for investigation and law enforcement purposes (termed as Network forensics) are also developed. Network forensics deals with the capture and analysis of the trace and log data of network intrusions from the current network security products and provides information to characterize intrusion or misbehavior features. To figure out such activities, people responsible for computer security incident response and digital forensic examination need to continually update their skills, tools, and knowledge to keep pace with changing technology. In M.Tech Thesis work, we will focus on this implementation of network forensic system using virtual honeynet to find out the activities of attackers.

Acknowledgement

First of all, I thank the almighty for his blessings and showing me the right direction. With His mercy, it has been made possible for me to reach so far.

It gives me great pleasure to express my gratitude towards the guidance and help I have received from Mr.Gurpal Singh Chhabra. I am thankful for his continual support, encouragement, and invaluable suggestion. He not only provided me help whenever needed, but also the resources required to complete this thesis report on time.

I am also thankful to Dr. S.S Bhatia, Head, School of Mathematics and Computer Applications for his kind help and cooperation. I express my gratitude to all the staff members of School of Mathematics and Computer Applications for providing me all the facilities required for the completion of my thesis work.

I would like to say thanks to all my friends especially for their support. I want to express my appreciation to every person who contributed with either inspirational or actual work to this thesis.

Last but not the least I am highly grateful to all my family members for their inspiration and ever encouraging moral support, which enables me to purse my studies.

Jatinder Kaur

Jatinder kaur

iv

Table of Contents

List of contents — Page No.

List Of Tables

List Of Figures

CHAPTER 1.

I. Introduction

Within last couple of decades, industry as well as governments uses Internet at an increasing pace in basic functions and core activities, as shown in the graph below (ref Figure1.1). Governments use Internet to provide citizens and businesses with public services. Electronic government services that are provided to citizens typically include paying income tax, demanding and issuing personal documentation such as birth and marriage certificates, issuing and renewing driving licenses, participating in election processes and so forth. Business has become depended on Internet not only to communicate and provide their product and services to customers but also to enact new business models which are entirely dependent on the use of the Internet, such as electronic marketplaces, online auctions, online bartering and information brokerage.

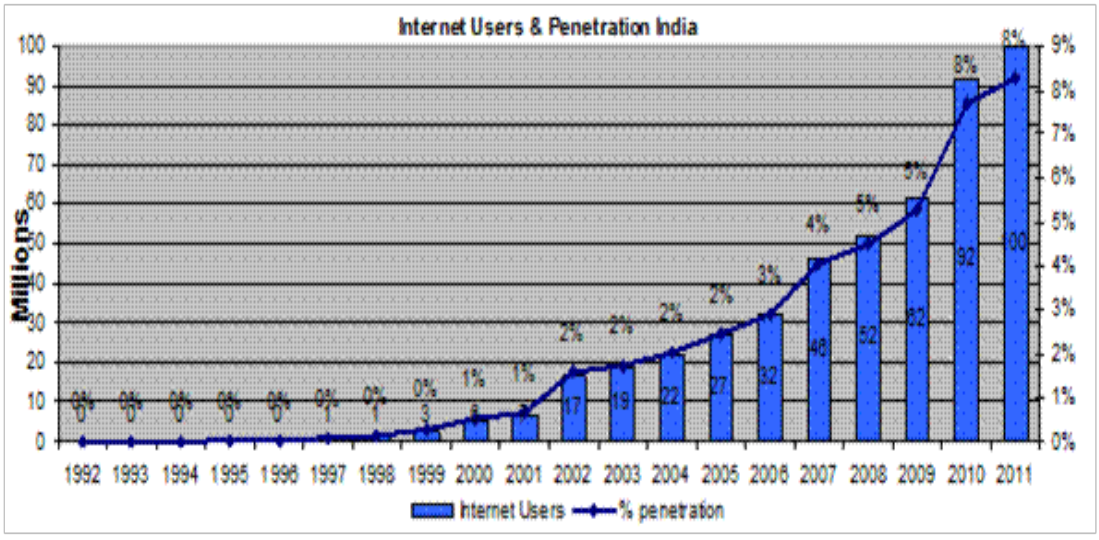

Figure 1.1 Growth of Internet usage & penetration in India

However, technological developments have also "a dark side": Since crime tends to follow opportunity and the Internet provides many new opportunities, new crimes as well as new ways to commit "traditional crimes" by means of new technologies emerge [1]. Due to the "anonymity" of the cyber-criminal activities and to the fact that these new (types of) crimes are not restricted by geographical boundaries, they have far-reaching consequences. In a networked world, where all points are equidistant from all others and are accessible from everywhere, the principles of the legal system cannot impose obligations on everyone to comply with all law [2]. As a result, governments and business become increasingly vulnerable to threats originating from the Internet.

The 2011 CyberSecurity Watch Survey uncovered that more attacks (58%) are caused by outsiders (those without authorized access to network systems and data) versus 21 % of attacks caused by insiders (employees or contractors with authorized access) and 21% from an unknown source; however 33% view the insider attacks to be more costl y, compared to 51% in 2010. Insider attacks are becoming more sophisticated, with a growing number of insiders (22%) using rootkits or hacker tools compared to 9% in 2 010, as these tools are increasingly automated and readily available[4].

'Internet forensics' (also referred to as Cyber forensics or Network forensics) is a sub-category of computer-forensics. Computer-forensics refers to the collection, preservation and analysis of computer-derived evidence to ensure its admissibility as evidence in a legal proceeding [3]. Internet forensics includes techniques and methodologies to collect, preserve and analyze digital data on the Internet for investigation and law enforcement purposes. It is a relatively recent field of research and practice that has evolved as a result of the increasing use of Internet and the move of criminal activity. It is also argued that Internet forensics evolved as a response to the hacker community [5].

The amount of data in the world is increasing. In 2002, about 5 exabytes of new data were created, growing 30% yearly since 1999 [6]. The proliferation of digital

information storing devices is incredible, and shows no sign of slowing. With smart-phones, iPods and digital cameras, the average American has more storage on his or her body than the first Cray supercomputers[7]. In 2003, 62 million US households owned computers connected to the Internet[8]. In 2008, one terabyte hard drives are sold for $100. With cheap storage, broadband internet connections for sharing, and ease of digital creation thanks to still and video cameras, some researchers warn of the oncoming "Exaflood" of information[6] .As storage becomes cheaper and personal hard drives become larger, the task of the digital investigator grows geometrically harder. Individuals own more and larger media devices than ever before, and sometimes when a crime is committed, evidence may exist somewhere on one of the devices. It is the job of the digital investigator to find what evidence exists and recover that evidence in a sound manner. One only has to look to the FBI to realize how important digital forensics has become in serious criminal cases. In 2007, the FBI opened it's 14th Regional Computer Forensics Laboratory (RCFL) and handled 76,581 pieces of media, up from 59,677 in 2006. Nine of the 14 labs reported a larger backlog at the end of 2007 than at the start, despite the increases in funding and staffing levels. In their 2007 annual report, the RCFLs state that "The capacity of electronic devices continues to increase, examiners must review more and more data. Therefore, even if the number of requests decreases, the workload either remains steady or actually increases in many cases". That same report emphasizes the importance of computer forensics in high profile cases, such as the Bike Path rapist, the foiled Fort Dix terrorist attack, and Operation

As the search space grows faster than law enforcement's ability to search, there exists an opportunity for software to step in and take some of the load from law enforcement personnel. This thesis argues that the current generation of forensic tools are not up to the task because these tools are designed for manual operation by a trained operator. What is needed, instead, is a new generation of tools that can perform automated analysis and reporting.

1.1 Cyber Crime

An electronic crime is defined as an illegal act that is carried out using a computer or electronic media. A cyber crime is an electronic crime that is carried out using the Internet, or a crime whose "crime scene" is the Internet. Cyber crimes are not necessarily new crimes; many cases involve rather classic types of crimes where criminals exploit computing power and accessibility to information. However, it seems that the anonymity provided through the Internet encourages crimes that involve the use of computer systems, since criminals believe that there is a small chance of being prosecuted, let alone being caught for their actions. Criminals are also increasingly taking advantage of hacker techniques and malicious code. Cyber crimes can be automated (such as spam, worms, Trojans, viruses, spyware) or specifically targeted such as theft of proprietary information or intellectual property, sabotage etc. It is estimated that computer fraud is merely committed by relatively unsophisticated individuals [8], while Internet fraud, on the other hand, and is believed to be the deed of highly sophisticated individuals [4].

A simple yet sturdy definition of cyber crime would be "unlawful acts wherein the computer is either a tool or a target or both". Defining cyber crimes, as "acts that are punishable by the information Technology Act" would be unsuitable as the Indian Penal Code also covers many cyber crimes, such as e-mail spoofing, cyber defamation etc. Cyber Crime refers to all activities done with criminal intent in cyberspace. These fall into three slots[9].

- Those against persons.
- Against Business and Non-business organizations.
- Crime targeting the government.

Hence, precisely we can say, any criminal offence that involves a computer/network can be known as a cyber crime or a computer crime. All the cyber crime activities can be summarized under the following categories. .

Unauthorised access: This occurs when a user/hacker deliberately gets access into someone else's network either to monitor or data destruction purposes.

Denial of service attack: It involves sending of disproportionate demands or data to the victims server beyond the limit that the server is capable to handle and hence causes the server to crash.

Virus, worms and Trojan attacks: viruses are bascially programs that are attached to a file which then gets circulated to other files and gradually to other computers in the network.

Worms unlike viruses do not need a host for attachments they make copies themselves and do this repeatedly hence eating up all the memory of the computer.

1.2 Virtual World

Cyber crime occurs in the virtual world, so it presents different issues that need to be addressed or taken care of. For eg., theft in the virtual world (often referred as cyber-theft), is the phenomenon of copying or creating image of the property. Mainly, the issues that need to be taken care in order to minimize the probability of any cyber-theft are number of transactions conducted daily, and the Phishing attacks to acquire sensitive consumer information such as username, password, and credit card information.

Computer forensics is the application of scientifically proven methods to gather, process, interpret, and to use digital evidence to provide a conclusive description of cyber crime activities. Network forensics also includes the act of making digital data suitable for inclusion into a criminal investigation. Today ***Network forensics*** is a term used in conjunction with law enforcement, and is offered as courses at many colleges and universities worldwide.

Computer Forensics is simply the application of computer investigation and analysis techniques in the interests of determining potential legal evidence."[1] Another is "a process to answer questions about digital states and events"[2]. This process often involves the investigation and examination computer system(s), including. The

forensic examiner renders an opinion, based upon the examination of the material that has been recovered. After rendering an opinion and report, to determine whether they are or have been used for criminal, civil or unauthorized activities. Mostly, computer forensics experts investigate data storage devices, these include but are not limited to hard drives, portable data devices (USB Drives, External drives, Micro Drives and many more) [9]. Digital evidences needs to be handled in a systematical & controlled way in order to make them accountable in the legal proceedings, as shown in the Figure 3. Cyber Analyst performs the following tasks while working with digital evidences:

1. Identify: Any digital information or artifacts that can be used as evidence.
2. Collect, observe & preserve.
3. Analyze , identify and organize.
4. Rebuild the evidence and verify the result every time [10].

Fig 1.2 Handling of Evidences by Cyber Analyst

Computer forensics is done in a fashion that adheres to the standards of evidence that are admissible in a court of law [9]. Thus, computer forensics must be techno-legal in nature rather than purely technical or purely legal. Some examples of such activities are

- Recovering thousand of deleted e-mails.
- Recovering evidence post formatting hard drive.
- Performing investigation after multiple users had taken over the system

1.3 Network Forensics

Network forensics deals with the capture and analysis of the trace and log data of network intrusions from the current network security products and provides information to characterize intrusion or misbehavior features. The power of various network forensic analysis tools available as open source can be integrated so that the investigator can have an edge over the attacker. The storage to handle large volumes of data and computing power to analyze the same is now available at cheaper rate. An effective network forensic system will increase the cost of the network crimes for the attacker and thus reduce network crime rates. A network forensics system can prove valuable investigative tools on malware attacking information collection. Forensics is not by itself a science. The word forensics means *"to bring to the court"*. Computer forensics, also sometimes referred as Network Forensics enables the systematic and careful identification of evidence in computer related crime and abuse cases. This may range from tracing the tracks of a hacker through a client's systems, to tracing the originator of defamatory emails, to recovering signs of fraud. This requires the necessary skills to identify an intruder's footprints and to properly gather the necessary evidence to prosecute in the court of law[11].

- collection,
- detection
- identification,
- examination
- correlation
- analyze
- documentation

of digital evidence from multiple systems for the purpose of uncovering the fact of attacks and other problem incident as well as perform the action to recover from the attack.

1. Identification – recognizing an incident from indicators and determining its type. This is not explicitly within the field of forensics, but significant because it impacts other steps.

2. Preservation – isolate, secure and preserve the state of physical and digital evidence. This includes preventing people from using the digital device or allowing other electromagnetic devices to be used within an affected radius.

3. Collection – record the physical scene and duplicate digital evidence using standardized and accepted procedures.

4. Examination – in-depth systematic search of evidence relating to the suspected crime. This focuses on identifying and locating potential evidence, possibly within unconventional locations. Construct detailed documentation for analysis.

5. Analysis – determine significance, reconstruct fragments of data and draw conclusions based on evidence found. It may take several iterations of examination and analysis to support a crime theory. The distinction of analysis is that it may not require high technical skills to perform and thus more people can work on this case.

6. Presentation – summarize and provide explanation of conclusions. This should be written in a layperson's terms using abstracted terminology. All abstracted terminology should reference the specific details.

7. Incident Response – The response to crime or intrusion detected is initiated based on the information gathered to validate and assess the incident

1.3.1 Network Forensics Systems Can Be Of Two Kinds :

1. "Catch-it-as-you-can" systems, in which all packets passing through certain traffic point are captured and written to storage with analysis being done subsequently in batch mode. This approach requires large amounts of storage.

2.“Stop, look and listen” systems, in which each packet is analyzed in a rudimentary way in memory and only certain information saved for future analysis. This approach requires a faster processor to keep up with incoming traffic[12].

1.3.2 The different types of Network Forensics Systems

1.**Distributed Systems Based Network Forensic System**

Internet and LANs are distributed in nature and networks attack events are logged in clients at various locations. There is a need to collect these logs, fuse them and analyze on a central server.

2. Soft Computing Based Network Forensic System

The soft computing implementations are used to analyze captured data and classify the attack data. Neural network and Fuzzy tools are used for validation of attack occurrence.

3. Honeypot Based Network Forensic System

Honeypot based system is used to attract the attackers so that their process methodology can be observed and analyzed to improve defense mechanisms.

1.3.3 Network Forensic Analysis Tools

Network Forensic Analysis Tools (NFATs) [13] allow administrators to monitor the networks, gather all information about anomalous traffic, and help in network forensics. NFATs synergizes with IDSs and firewalls making preserving long term record of network traffic possible and allowing quick analysis of trouble spots identified by IDSs and firewalls. A few functions of an NFAT

Network traffic recording and analysis

1) Network performance evaluation
2) Data aggregation from multiple sources

3) Anomaly detection

4) Determination of network protocols in use

5) Detection of employee misuse of resources

6) Security investigations and incident response

7) Intellectual property protection

Intellectual property protection

- The commercial NFATs available in the market are – NetIntercept , NetDetector , NetFlow , SilentRunner , EnCase , and VisualRoute .
- The open source / freeware NFATs are – TCPDump / Libpcap / WinDump, Snort , Nmap , P0f , Tcpstat , Tcptrace, Tcpflow, Wireshark

1.4 Computer Forensics Vs Network Forensics

Computer forensics is the application of scientifically proven methods to gather, process, interpret, and to use digital evidence to provide a conclusive description of any crime or criminal activities that may or may not have used the virtual world for committing crime. Whereas Network forensics is scientifically proven techniques to collect, detect, identify, examine, correlate, analyze, and document digital evidence from multiple systems for the purpose of uncovering the fact of attacks and other problem incident as well as perform the action to recover from the attack.. For e.g.

Computer forensics: If any murder has occurred and we use the computer related software to figure out the fact & evidences

Network Forensics*:* If anybody enters a desktop using unauthorized access via internet connection & copy data to his own laptop to improve their overall state of security. Normally, production honeypots are low-interaction honeypots, Hence, the difference between the computer and Network forensic lies on the fact, whether the crime have been conducted the virtual world or the physical world.

The computer forensic field is very active in both the commercial and research sectors. Commercial tools provide standardized interfaces, storage formats, training, certification and support to the forensics community. Some are formally reviewed, tested and analyzed by the National Institute of Standards and Technology (NIST) Computer Forensic Tool Testing (CFTT) program other tools are used without formal testing, raising potential problems if the results are challenged in court.

1.5 Study of Honeypot/Honeynet

1.5.1 Honeypot Based Network Forensic System

Honeypot based system is used to attract the attackers so that their process methodology can be observed and analyzed to improve defense mechanisms.“A honeypot consists in an environment where vulnerabilities have been deliberately introduced in order to observe attacks and intrusions.” A honey pot is a computer system on the Internet that is expressly set up to attract and "trap" people who attempt to penetrate other people's computer systems. Put in other words honeypots are a great environment to observe malicious traffic.They can do everything from detecting encrypted attacks in IPv6 networks to capturing the latest in on-line credit card fraud. Its is this flexibility that gives honeypots their true power. It is also this flexibility that can make them challenging to define and understand. As such, I use the following definition to define what a honeypot is. A honeypot is an information system resource whose value lies in unauthorized or illicit use of that resource.

1.5.2 Types of Honeypots

Honeypots can be classified based on their deployment and based on their level of involvement. Based on deployment, honeypots may be classified as:

1. production honeypots
2. research honeypots

Production honeypots are easy to use, capture only limited information, and are used primarily by companies or corporations; Production honeypots are placed inside the

production network with other production servers by an organization which are easier to deploy. They give less information about the attacks or attackers than research honeypots do.

Research honeypots are run to gather information about the motives and tactics of the Blackhat community targeting different networks. These honeypots do not add direct value to a specific organization; instead, they are used to research the threats organizations face and to learn how to better protect against those threats. to collect the packets and network logs which are later being sent to the base machine from virtual honeypots for further analysis and investigations.

Research honeypots are complex to deploy and maintain, capture extensive information, and are used primarily by research, military, or government organizations.

To set up a honey pot, it is recommended that you:

- Install the operating system without patches installed and using typical defaults and options
- Make sure that there is no data on the system that cannot safely be destroyed
- Add the application that is designed to record the activities of the invader

Based on design criteria, honeypots can be classified as

1. pure honeypots
2. high-interaction honeypots
3. low-interaction honeypots

Pure honeypots are full-fledged production systems. The activities of the attacker are monitored using a casual tap that has been installed on the honeypot's link to the network. No other software needs to be installed. Even though a pure honeypot is useful, stealthiness of the defense mechanisms can be ensured by a more controlled mechanism.

High-interaction honeypots imitate the activities of the real systems that host a variety of services and, therefore, an attacker may be allowed a lot of services to waste his time. According to recent researches in high interaction honeypot technology, by employing virtual machines, multiple honeypots can be hosted on a single physical machine. Therefore, even if the honeypot is compromised, it can be restored more quickly. In general, high interaction honeypots provide more security by being difficult to detect, but they are highly expensive to maintain. If virtual machines are not available, one honeypot must be maintained for each physical computer, which can be exorbitantly expensive. Example: Honeynet.

Low-interaction honeypots simulate only the services frequently requested by attackers. Since they consume relatively few resources, multiple virtual machines can easily be hosted on one physical system, the virtual systems have a short response time, and less code is required, reducing the complexity of the security of the virtual systems. Example: Honeyd.

Honeyd: Low-interaction honeypot

Honeyd is a low-interaction honeypot. Developed by Niels Provos, Honeyd is OpenSource and designed to run primarily on Unix systems (though it has been ported to Windows). Honeyd works on the concept of monitoring unused IP space. Anytime it sees a connection attempt to an unused IP, it intercepts the connection and then interacts with the attacker, pretending to be the victim. By default, Honeyd detects and logs any connection to any UDP or TCP port. In addition, you can configure emulated services to monitor specific ports, such as an emulated FTP server monitoring TCP port 21. When an attacker connects to the emulated service, not only does the honeypot detect and log the activity, but it captures all of the attacker's interaction with the emulated service. In the case of the emulated FTP server, we can potentially capture the attacker's login and password, the commands they issue, and perhaps even learn what they are looking for or their identity. It all depends on the level of emulation by the honeypot. Most emulated services work the same way. They expect a specific type of behavior, and then are programmed to react in a

predetermined way. If attack A does this, then react this way. If attack B does this, then respond this way. The limitation is if the attacker does something that the emulation does not expect, then it does not know how to respond. Most low-interaction honeypots, including Honeyd, simply generate an error message. You can see what commands the emulated FTP server for Honeyd supports by review the source code.

Honeynets

Two or more honeypots on a network form a **honeynet**. Typically, a honeynet is used for monitoring a larger and/or more diverse network in which one honeypot may not be sufficient. Honeynets and honeypots are usually implemented as parts of larger network intrusion detection systems. A **honeyfarm** is a centralized collection of honeypots and analysis tools.The concept of the honeynet first began in 1999 when Lance Spitzner, founder of the Honeynet Project, published the paper "To Build a Honeypot":

A honeynet is a network of high interaction honeypots that simulates a production network and configured such that all activity is monitored, recorded and in a degree, discreetly regulate.[

Honeynet: High-interaction honeypot

Honeynets are a prime example of high-interaction honeypot. Honeynets are not a product, they are not a software solution that you install on a computer. Instead, Honeyents are an architecture, an entire network of computers designed to attacked. The idea is to have an architecture that creates a highly controlled network, one where all activity is controlled and captured. Within this network we place our intended victims, real computers running real applications. The bad guys find, attack, and break into these systems on their own initiative. When they do, they do not realize they are within a Honeynet. All of their activity, from encrypted SSH sessions to emails and files uploads, are captured without them knowing it. This is done by inserting kernel modules on the victim systems that capture all of the attacker's

actions. At the same time, the Honeynet controls the attacker's activity. Honeynets do this using a Honeywall gateway. This gateway allows inbound traffic to the victim systems, but controls the outbound traffic using intrusion prevention technologies. This gives the attacker the flexibility to interact with the victim systems, but prevents the attacker from harming other non-Honeynet computers

1.5.3 Advantages

Honeypots are a tremendously simply concept, which gives them some very powerful strengths.

Small data sets of high value*:* Honeypots collect small amounts of information. Instead of logging a one GB of data a day, they can log only one MB of data a day. Instead of generating 10,000 alerts a day, they can generate only 10 alerts a day. Remember, honeypots only capture bad activity, any interaction with a honeypot is most likely unauthorized or malicious activity. As such, honeypots reduce 'noise' by collecting only small data sets, but information of high value, as it is only the bad guys. This means its much easier (and cheaper) to analyze the data a honeypot collects and derive value from it.

New tools and tactics*:* Honeypots are designed to capture anything thrown at them, including tools or tactics never seen before.

Minimal resources*:* Honeypots require minimal resources, they only capture bad activity. This means an old Pentium computer with 128MB of RAM can easily handle an entire class B network sitting off an OC-12 network.

Encryption or IPv6*:* Unlike most security technologies (such as IDS systems) honeypots work fine in encrypted or IPv6 environments. It does not matter what the bad guys throw at a honeypot, the honeypot will detect and capture it.

Information: Honeypots can collect in-depth information that few, if any other technologies can match.

Simplicty: Finally, honeypots are conceptually very simple. There are no fancy algorithms to develop, state tables to maintain, or signatures to update. The simpler a technology, the less likely there will be mistakes or misconfigurations.

1.5.4 Disadvantages

Like any technology, honeypots also have their weaknesses. It is because of this they do not replace any current technology, but work with existing technologies.

Limited view*:* Honeypots can only track and capture activity that directly interacts with them. Honeypots will not capture attacks against other systems, unless the attacker or threat interacts with the honeypots also.

Risk: All security technologies have risk. Firewalls have risk of being penetrated, encryption has the risk of being broken, IDS sensors have the risk of failing to detect attacks. Honeypots are no different, they have risk also. Specifically, honeypots have the risk of being taken over by the bad guy and being used to harm other systems. This risk is of various types for different honeypots. Depending on the type of honeypot, it can have no more risk than an IDS sensor, while some honeypots have a great deal of risk.

CHAPTER 2.

II. Literature survey

Internet has become not only a crime scene, but also a breeding ground for primary and secondary sources of evidence. “Cyberspace has become the neighborhood, wherein law enforcement officers must regularly interact with their constituency” [3]. A forensics investigation requires the use of disciplined investigative techniques to discover and analyze traces of evidence left behind after a committed crime [7].

The Cyber Security watch Survey 2011 states that, outsiders has the maximum percentage of the attacks followed by the insiders and unknown, as compared to the 2006 report, that declared the maximum damage by the insiders than the outsiders, a shown in the Figure 2.1(b) below. Whereas, if we compare the cost of damage by the outsiders & insiders, then the difference is relatively low, as show in the Figure 2.1(a) [6].

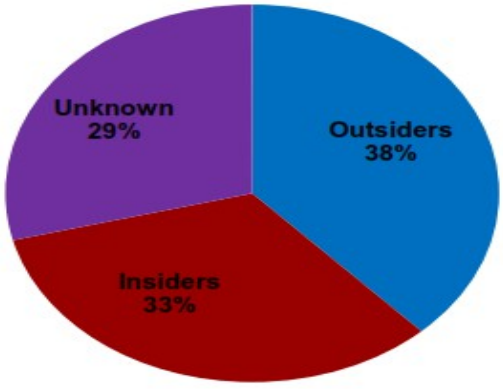

Figure 2.1(a)

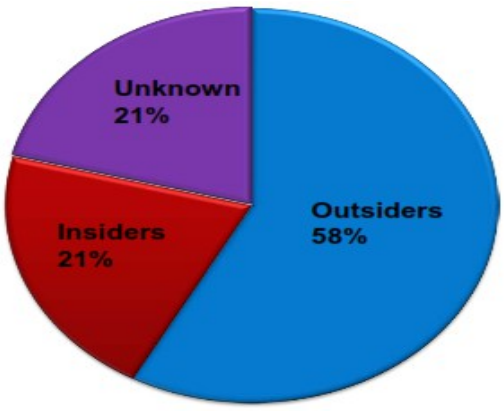

Figure 2.1(b)

Figure 2.1(a): Electronic Crimes were more costly or damaging to your organization

Figure 2.1(b): percent of the Electronic Crime events are known or suspected

Hanan Hibshi, Timothy Vidas and Lorrie Cranor in the paper titled "*Usability of Forensics Tools : A User Study*" that exploits the Digital Forensics has become a critical part of almost every investigation, and users of digital forensics tools are becoming more diverse in their backgrounds and interests. This paper examines the usability aspect of forensics tools through interviews and surveys designed to obtain feedback from professionals using these tools as part of their regularly assigned duties. The study results highlight a number of usability issues that need to be taken into consideration when designing and implementing digital forensics tools. This paper presents that current digital forensics tools are not considered user friendly and they lack intuitive interfaces.

Peter Stephenson and Richard Walter introduced a *"Cyber Crime Assessment***".** The ongoing research reported in this paper will begin the process of translating physical crime scene assessment techniques into digital practice. the translation between physical investigation and digital investigation is not straightforward. The addition of particular techniques pioneered by crime assessment experts in physical investigations has turned out to be far more promising than application of psychological profiling. This research-in-progress paper presents a new approach to cyber crime assessment. It views the computer as the crime scene and it applies a crime typology.

Vasilios Katos , Peter M. Bednar introduced a method in the paper titled "*A cyber-crime investigation framework*". This paper presents an approach adopting elements of the Strategic Systems Thinking Framework(SST) by which conflicting information due to the unavoidable uncertainty can be captured and processed, in support of the investigation process.A formal description of this approach is proposed as a basis for developing a cyber-crime investigation support system. The purpose of this paper is to introduce the aforementioned concepts in the cyber-crime investigations domain

and the suitability of the underlying tools is studied. Therefore a generic cyber-crime investigation framework is presented. An ongoing area of research is the application of the framework to specific cyber-crime classes, which could be based either on the technical nature of the malicious activity such as a DDoS, or on the type and purpose of the offence, such as identity theft.

Bhavesh Patel, Sanjay.M.Shah and Sameer Singh Chauhan introducing a "*Comparitive Analysis of Network Forensics Systems* " that shows the three types of model of Network Forensics systems. In distributed model the capturing of packet is on multiple host while in Soft computing model it is on the single host. Distributed model decision making is based on statistical data while in it is based on non statistical data. Time and cost involve in forensic analysis is less in soft computing model compare to distributed model. In soft computing model, if the rules are such that we can differentiate between an attack and legitimate traffic then we get desirable results. In distributed model it is very hard to differentiate between an attack and legitimate traffic, so desirable results are not possible every time. All network traffic is captured in distributed model; while in soft computing approach some data may be lost due to centralize capturing system. Incident response can be easily handled in soft computing model compare to distributed model. Distributed model is efficient in capturing the complete network traffic. Soft computing model is efficient in differentiating the attack and legitimate traffic.

B. Skaggs, B. Blackburn**,** G. Manes, S. Shenoi introduced a *"Network Vulnerabiltiy Analysis"*. Current hacker tools and technologies warrant reengineering to address cyber crime and homeland security. The creation of network scanners is necessary to secure the information infrastructure by gathering network topology, intelligence, internallexternal vulnerability analysis, and penetration testing. Scanners must be able to function on a variety of networks: Internet (IP), wireless, and converged networks. Scanners should be extendable and upgradeable to facilitate use by a broad spectrum of users and platforms; such flexibility allows users to keep up with current hacker technology. This paper describes one such scanner, referred to **as** NetGlean. NetGlean

uses a relational database, SQL, to store and retrieve host information. These converged networks of the future will have telephone, wireless, IP, video, and other services running together. Future versions of NetGlean should prototype a converged network scanner, which would utilize the same features as NetGlean.

Ziming Zhao, Gail-Joon Ahn and Hongxin Hu proposed a method in "*Automatic Extraction of Secrets from Malware*" to extracts binary code relevant to secret hiding behaviors. Then relocate and reuse the extracted binary code in a self-contained fashion to reveal hidden information. It demonstrate the feasibility of our approach through a proof-of-concept prototype called ASES (Automatic and Systematic Extraction of Secrets). The proposed approach consists of three major tasks to identify secret-related code in binary, relocate and reuse it without symbol information. We also developed a prototype system, to extract external API information from malware binaries.

Ahmad Almulhem proposed a challenges in the paper titled "*Network Forensics: Notions and Challenges*" . This paper, presents various aspects of network forensics are reviewed as well as related technologies and their limitations and challenges in deploying a network forensics infrastructure are highlighted**.** Network forensics has been proposed to introduce investigation capabilities in current networks. It refers to a dedicated investigation infrastructure that allows for the collection and analysis of network packets and events for investigative purposes. In this paper, various aspects of network forensics were reviewed as well as related technologies and their limitations. Also, challenges in deploying network forensics infrastructure were highlighted.

Chih-Hung Lin and Chung-Huang Yang proposed a method "*Implementation of network forensics based on honeypot*". This study, designed a network forensics system which contented honeypot system to solve the information gathering problems in the past, and distributed the honeypot system module at the same time, and even combine with the Capture-HPC program to check out the traverse web sites

in search of client-side malware. Therefore, the study result can help investigators gather the evidences about the network crimes to allow criminals to be prosecuted in court . This technique can analyze the logs that are cause by the malware attacking. To combine with the Capture-HPC program, the system can even check out the traverse web sites in search of client-side malware.

F. Pouget, M. Dacier proposed a new forensic algrothim in paper titled "*Honeypot based forensics*".This present an algorithm to characterize the root causes of these attacks. This algorithm enables us to obtain precious and non trivial information to identify the various attacks targeting our environment. This algorithm identifies root causes of the data collected from our honeypot environment. We demonstrate that identifying the root causes is a prerequisite for a better understanding of malicious activity observed thanks to honeypots environments . This paper present that simple clustering techniques can be applied to obtain more in-depth information on observed attacks. Results are very promising and confirm that such an analysis is meaningful. Indeed, it obtain clusters representing root causes of attacks. This approach shown that a large amount of malicious traffic corresponds to a few number of root causes. Such root causes are important information on the attack tools in use. Other applications such as tool fingerprinting or analysis of tools evolution were also presented.

M.I. Cohen proposed a method in the paper titled "*PyFlag – An advanced network forensic framework"*. This paper describes the PyFlag architecture and in particular how that is used in the network forensics context. The novel processing of HTML pages is described and the PyFlag page rendering is demonstrated. PyFlag's novel processing of complex web applications such as Gmail and other web applications is described. Network forensics processing presents many challenges including the large number of protocols present on the Internet and the need for adapting to new protocols flexibly and quickly. The use of complex web applications such as web mail requires further processing beyond the HTTP protocol to support further decoding specific to the applications themselves.

Emmanuel S. Pilli, R.C. Joshi and Rajdeep Niyogi introduced a method in the paper titled "*A Generic Framework for Network Forensics*". The Internet needs to be protected from these attacks and an appropriate response has to be generated to handle them to reduce the impact. Network forensics is the science that deals with capture, recording, and analysis of network traffic for investigative purpose and incident response. This paper presents a generic framework for network forensic analysis by specifically identifying the steps connected only to network forensics from the already proposed models for digital investigation. Each of the phases in the framework is elucidated. A comparison of the proposed model is done with the existing models for digital investigation.

Hong-Ming Wang, Chung-Huang Yang introduced a new idea to develop a forensic system in the paper titled "*Design and Implementation of a Network Forensics System for Linux*" In this paper,it presents idea to develop a network forensics system for Linux, which is used to collect and protect evidences when the cyber crime occurred. It consists of a live system, a friendly graphical launch menu, strengthen PyFlag software, and integrate required tools of system and network. This system can expand its volatile, report presentation functionalities, and provide investigator to perform network forensics work quickly and correctly. The result of the forensics in this system can not only preserve evidences of the cyber crime, but also help organizations and institutions to understand the whole context of network security incidents and to strengthen the network host defense and security policy.

Natarajan Meghanathan, Sumanth Reddy Allam and Loretta A. Moore introduces new methods and techniques in the paper titled "*Tools and Techniques For Network Forensics*". This paper discusses the different tools and techniques available to conduct network forensics. Some of the tools discussed include: eMailTrackerPro – to identify the physical location of an email sender; Web Historian – to find the duration of each visit and the files uploaded and downloaded from the visited website; packet sniffers like Ethereal – to capture and analyze the data exchanged among the different

computers in the network.The second half of the paper presents a survey of different IP traceback techniques like packet marking that help a forensic investigator to identify the true sources of the attacking IP packets. This paper discusses the use of Honeypots and Honeynets that gather intelligence about the enemy and the tools and tactics of network intruders.

CHAPTER 3.

III. Problem Statement and Objective

Network forensics deals with the capture and analysis of the trace and log data of network intrusions from the current network security products and provides information to characterize intrusion or misbehavior features. This is exactly what we will be implementing to get the internal working of server honeypot technology and network forensics. Honeypot based system is used to attract the attackers so that their process methodology can be observed and analyzed to improve defense mechanisms. Network Forensic allow administrators to monitor the networks, gather all the intelligent information about all the abnormal traffic, and helps in network forensics. Here we have tried to implement the Linux based Network Forensic System Based on Virtual Honeynet Technology. We have developed a prototype system which is used for gathering the network logs , are highly malicious in nature using honeynet infrastructure . The end result of the system is collected network data which are highly malicious in nature and which can be used for further investigation to get the intelligent information about the attackers.

3.2 Objectives:

1. Research and design of network forensic system.
2. Design of Honeypot/Honeynet Technology as Network Forensic System.
3. Development of automated network forensic system using Virtual Honeynet.

CHAPTER 4.

IV. Background and Existing Tools

4.1 EnCase

One type of software available for forensic analysis is EnCase (www.encase.com/products/ee_index.asp). EnCase was originally developed for law enforcement personnel, but has matured to support commercial needs, as well. The EnCase Enterprise Edition is a network enabled incident response system which offers immediate and complete forensic analysis of volatile and static data on compromised servers and workstations anywhere on the network, without disrupting operations. It consists of three components. The first of these components is the Examiner software. This software is installed on a secure system where Investigations and audits are performed. The second component is called SAFE, which stands for Secure Authentication of EnCase. SAFE is a server which is used to authenticate users, administer access rights, maintain logs of EnCase transactions, and provide for secure data transmission. The final component is Servlet, an efficient software component installed on network workstations and servers to establish connectivity between the Examiner, SAFE, and the networked workstations, servers, or devices being investigated[14].

These components work to provide the acquisition and analysis of volatile data on workstations and servers suspected to be compromised. This includes running applications, open files and other data in RAM, as well as acquiring and analyzing attached drive media, including files, operating systems artifacts, and data in file slack and unallocated spaces. It quickly isolates, identifies, assesses and rectifies both internal and external security breaches and provides non-intrusive forensic functionality to ensure that investigations withstand internal or external scrutiny regarding thoroughness, accuracy and authenticity.

In summary, the EnCase Enterprise Edition conducts comprehensive investigations, uncovering information and evidence pertaining to incidents that other tools cannot find. EnCase will find information despite efforts made to hide or delete it[15].

4.2 Sleuth Kit

The Sleuth Kit (TSK) is a C library and a collection of command line tools. TSK can be integrated into automated forensics systems in many ways, including as a C library and by using the SQLite database that it can can create. It is available for both windows and unix platform. It is actually is a set of tools, that integrates, The Sleuth Kit, for example Allin1 and NBTempo [14]. Allin1 is used in Sleuth Lit for Extract strings (ASCII and Unicode) from allocated and unallocated, Sort by file types or Sort by images and create thumbnails, Make foremost run on images. NBTempo is a GUI (Graphical User Interface) Bash script for making files timelines and reporting them in CSV (electronic sheet) format [16].

It is the file system tool that allows you to examine file systems of a suspect computer in a non-intrusive fashion. Because the tools do not rely on the operating system to process the file systems, deleted and hidden content is shown.

The Sleuth Kit is written in C and Perl and uses some code and design from The Coroner's Toolkit (TCT). The Sleuth Kit has been tested on Linux, Mac OS X, Windows (Visual Studio and mingw), Open & FreeBSD, Solaris, and Open source software allows you to customize the tools for your environment and validate the code. The Sleuth Kit , Analyzes raw (i.e. dd), Expert Witness (i.e. EnCase) and AFF file system and disk images. Tools can be run on a live Windows or UNIX system during Incident Response. These tools will show files that have been "hidden" by rootkits and will not modify the A-Time of files that are viewed. The searching techniques used are:

- List allocated and deleted ASCII and Unicode file names.
- Display the details and contents of all NTFS attributes (including all Alternate Data Streams).

- Display file system and meta-data structure details.
- Create time lines of file activity, which can be imported into a spread sheet to create graphs and reports.
- Lookup file hashes in a hash database, such as the NIST NSRL, Hash Keeper, and custom databases that have been created with the 'md5sum' tool.
- Organize files based on their type (for example all executables, jpegs, and documents are separated). Pages of thumbnails can be made of graphic images for quick analysis.

The ***Autopsy*** Forensic Browser is a web front-end graphical interface to the tools in The Sleuth Kit, which allows you to more easily conduct an investigation. Autopsy provides case management, image integrity, keyword searching, and other automated operations

Name	Scripting	Cost
Encase	YES	$2,708.31
FTK	NO	$2,557.44
SleuthKit	YES	Free
Pyflag	YES	Free

Table 4.1. Forensic tools, their costs and customization ability

4.3 FTK

The Forensic Toolkit (FTK) offers law enforcement and corporate security professionals the ability to perform complete and thorough computer forensic examinations. The FTK features powerful file filtering and search functionality. FTK's customizable filters allow you to sort through thousands of files to quickly find

the evidence you need[17]. FTK is recognized as the leading forensic tool to perform e-mail analysis. Some of the important features of FTKM are as follows:

- FTK Explorer allows you to quickly navigate through acquired images and generate audit logs and case reports.
- Compatible with the Password Recovery Toolkit TM and Distributed Network Attack.
- Full text indexing powered by dt Search yields instant text search results.
- Automatically recover deleted files and partitions and target key files quickly by creating custom file filters.
- Supported File & Acquisition Formats: NTFS, NTFS compressed, FAT 12/16/32, and Linux ext2 & ext3.
- Image formats include: Encase, SMART, Snapback, Safeback (up to but not including v.3), and Linux DD.
- Supports: Outlook, Outlook Express, AOL, Netscape, Yahoo, Earthlink, Eudora, Hotmail, and MSN e-mail and help to View, search, print, and export e-mail messages and attachments..It also recovers deleted and partially deleted e-mail.
- Automatically extract data from PKZIP, WinZip, WinRAR, GZIP, and TAR compressed files.
- Identify and flag standard operating system and program files and known child pornography and other potential evidence files
- Registry Viewer that helps in accessing and decrypting storage data and vies independent registry files, and finally generate reports [16].

4.4 Pyflag

The Australian government has released the Python Forensic and Log Analysis GUI (PyFlag). PyFlag is an open source forensics suite designed for media and network analysis. PyFlag imports a case image into a back-end database where persistent information is stored for access through a web browser from a client workstation. In practice the database can be on the same system as the client, allowing for a mobile deployment, or on a central server, allowing many investigators to work n the same case at the same time. PyFlag uses TSK for underlying image access and builds individual file analysis, extraction and reporting on top of TSK[18]. Pyflag provides its own scripting language called PyFlash and also allows users to write their own extensions to the suite in python.[19].

4.5 Foremost

It is linux based program that recover specific file types from disk images (like all JPG files). It uses a configuration file to specify headers and footers to search for. Intended to be run on disk images, foremost can search through most any kind of data without worrying about the format. Foremost is a console program to recover files based on their headers, footers, and internal data structures. This process is commonly referred to as data carving. Foremost can work on image files, such as those generated by dd, Safeback, Encase, etc, or directly on a drive. The headers and footers can be specified by a configuration file or you can use command line switches to specify built-in file types. These built-in types look at the data structures of a given file format allowing for a more reliable and faster recovery.One important limitation of the foremost tool is that, due to programming difficulties, foremost is limited to processing files smaller than 2GB.

4.6 Fatback

Fatback is a tool used to get back the FAT files that have been deleted from the drive. FATBACK easily installs on linux and it performs file recovery on FAT12, FAT16

and FAT32 file systems from a linux forensics platform. Some of the important features of the fatback are:

- Log file name support.
- Recursive undeletion of directory.
- Ability to work with single partitions or entire disk.

Figure 4.1 shows how to use the fatback tool to get the list of all the files that have been deleted. It has use several FatBack command line options:

-a: runs fatback in automatic undelete mode. Or in other words it will attempt to recover all deleted files in a given partition.

-o: places recovered files in a specified directory (in the example below its "undeleted").

```
root@localhost chapter11]# fatback -a -o undeleted -s evidencefloppy.bir
lo audit log specified, using "./fatback.log"
  (Done)
root@localhost chapter11]# cd undeleted
root@localhost undeleted]# ls -al
otal 28
rwxr-xr-x    2 root     root         4096 Apr  9 16:37 .
rwxr-xr-x    5 root     root         4096 Apr  9 16:36 ..
rw-r--r--    1 root     root        20480 Apr  9 16:37 ?OCUMENT.DOC
```

Figure 4.1 : Use of FATBACK TOOL

V. An Analysis of Current batch Report

This chapter presents the batch forensic reports created by today's top-of-the-line commercial and open source forensic tools. To test these tools with data that could be published here we built a clean computer system running Windows XP in a VMWare Virtual machine, populated the machine with two user accounts and a standard set of software, and then used the accounts to engage in simulated real-world communications. We also conducted a pilot user study of PyFlag.

The analysis here confirms what many forensic practitioners already know: significant training is needed to use today's forensic tools. The tools have many reports, each one highly specific with a lot of information which easily overwhelms users and customers alike, delaying the finding and use of actionable intelligence. Each report that comes with EnCase internalizes its information, making sharing with other reports difficult, and FTK has no way to script it's multiple reports together into the single report desired.

5.1 Building a Test Image Containing Realistic Data

To evaluate these tools against each other for ease of use and value of automated reports, we created a test image populated with realistic data. The test image is a VMW are Fusion virtual machine running Microsoft Windows XP Service Pack 3 with an assortment of messaging and communication applications installed as seen in

Software Package	Version
Windows XP	Service Pack3
Internet Explorer	6.0.2900.5512.xpsp.080413-211
Mozilla Firefox	3.0.3
Google Chrome	0.3.154.9

Table 5.1. Test system that was used to generate realistic data image

5.2 Building the Test Image

In our hypothetical scenario, a computer is discovered or acquired and the analyst needs to determine who was using the drive, what they were doing with it, and with whom they were communicating. To enable testing of this scenario, we created a test image with realistic but fictional data.

The image has two users, domexuser1 and domexuser2. (The user accounts were created from the default administrator account and labeled domexuser1 and domexuser2 for simplicity.) Each account was created as a standard user with no administrator privileges.

On a separate computer, a domexuser3 account was created for third party communications, a requirement for portraying a communications network in which there are unknown or not yet analyzed identities. We created a hotmail and gmail account for each user account, for a total of 6 email accounts.After the user accounts were created, the administrator account was used to install common communication and productivity software, including Microsoft Office 2008, Mozilla Firefox, Mozilla Thunderbird, AOL Instant Messenger, and Pidgin were installed for all users. The installation files were deleted and the recycle bin was emptied, and the administrator account logged off.

Over a course of several days, an experimenter playing the role of one user and then the second exchanged instant messages and emails with domexuser3, which resided on a separate system. The two accounts received, edited and saved office document files as well as various media files. Some of these files were then deleted. Email and instant messenger conversations were saved locally on the system. The accounts also visited web pages for news and webmail.

Forum and message board accounts were created and used. Domexuser1 was the first to log in, and immediately went online to create communication paths. Using Mozilla Firefox, domexuser1 created a hotmail account, a gmail account, agchat account and an AOL instant messenging (IM) account, all with the name domexuser1. Thunderbird was configured for the accounts and left open to synchronize. AOL instant messenger was connected to the internet and domexuser1 setup was completed.

The Switch User option in the XP menu was then used to log in as domexuser2 created for the username domexuser2. Outlook Express was configure and synchronized for the mail accounts and Pidgin was logged on for the AOL IM and gchat accounts. Switching between the three accounts, test emails and instant messages were sent back and forth to establish that all were participating. Domexuser1 then created three Microsoft Word documents and three Microsoft Excel Documents. One document of each type was sent to the email threads, one of each was deleted, and one of each was left alone.

The VMWare image then needed to be imaged for analysis. VMWare Fusion allows allows the vmdk files to be mounted within userspace. The unix command dd was then used with the options noerror and sync to provide a raw copy of the image. The final image was 40 Gigabytes in size, with much of that space empty.

5.3 Why a Test Image

This thesis does not use real data, instead, we created a realistic test image to provide a standardized image to test current tools on and to develop PyFlag modules against. Using realistic data avoids the man hours required to scrub the dataset and results of Personally Identifiable Information (PII). Finally, the use of realistic communications data would require consent of all communicating parties, which was deemed to be too high a hurdle for a single research project such as this.

5.4 An Analysis of Current Batch Reports

All of the reports that were analyzed for this thesis have a similar flaw: the reports are filled with data, but they do not present intelligence. The forensic tools analyzed here are able to recover tremendous amounts of raw data, files, and streams and associated metadata. This data by itself yields little value in an investigation. There are pages and pages of file names, time stamps, attributes, and other low-level details. None of this data is presented with any higher level of summarization or analysis. Data must be analyzed in the context of the current case to yield actionable intelligence. This analysis should occur before the tool presents its results back to the user: it should not be the task of the user to try to weed through all of the data and make sense of it. Of course it is risky to take the human out of the loop completely—the tool should not hide data from the user, nor be biased towards any type of intelligence conclusion. But some decision must be made, for the simple fact that all of the information cannot fit on a single page. Today's tools typically order their output alphabetically or chronologically. In most cases this is not the optimum ordering.

Each tool analyzed here presents very different reports containing different kinds of information. Both the reports and the interfaces used to obtain them correspond to the forensic tool's internal program architecture, rather than the way that an analyst would hope a tool to work or a report to be organized. To use these tools, the analyst must learn to think the way the tool's original programmer thought.

Today's forensic tools are diverse and designed for use by digital forensic tool experts. The commercial tools are designed with complex interfaces that are confusing to both inexperienced users and to experts who have merely been trained on other tools. Documentation and tutorials are available for all of these programs, but it is highly specific to each tool. This is not surprising:

Commercial training and certification on individual tools is significant source of revenue for tool makers, so there is little incentive to standardize. Both the complexity of the commercial tools and their differences from one another are made clear by comparing the EnCase user interface(Figure 5.1) and the FTK user interface(Figure 5.3) .

The purpose of this thesis is to describe a next-generation forensic tool that can be used by a person who is familiar with digital forensic abilities and limitations, but who is not an expert in any technique or tool set. The user will need the tool to produce a report with useful information with a minimum of configuration and interaction. Ideally the tool will be a "one-click" solution, producing a report with minimal involvement. The user can then decide if a piece of media requires deeper analysis by an expert, ideally with a focused search for certain information or file types.

5.4.1 EnCase Batch Report

EnCase data analysis begins by ingesting the data into the EnCase program. Each ingest then EnCase data analysis begins by ingesting the data into the EnCase program. Each ingest then must be indexed before a useful report can be generated. This indexing uses a word dictionary that the user can add to and then search the image, creating a list of found words. The reporting tab is then populated with the results of the index. The default report (Figure 3.2) shows a file directory tree, which has far more data than is useful, with little filtering or layout consideration. Though this report could be better organized and searched through with more scripting, the built-in EnCase report provides little intelligence to the investigator. Other default reports are a gallery view of all images found and a timeline view of all files on the media. The reports allows filtering by the user according to different file properties.

The next step is to process the case,which runs file type specific searches against the media, as well as regular expression searches. EnCase can also mount container files in this process. These reports are then saved as html files in a user chosen directory, and must be viewed by an external web browser.

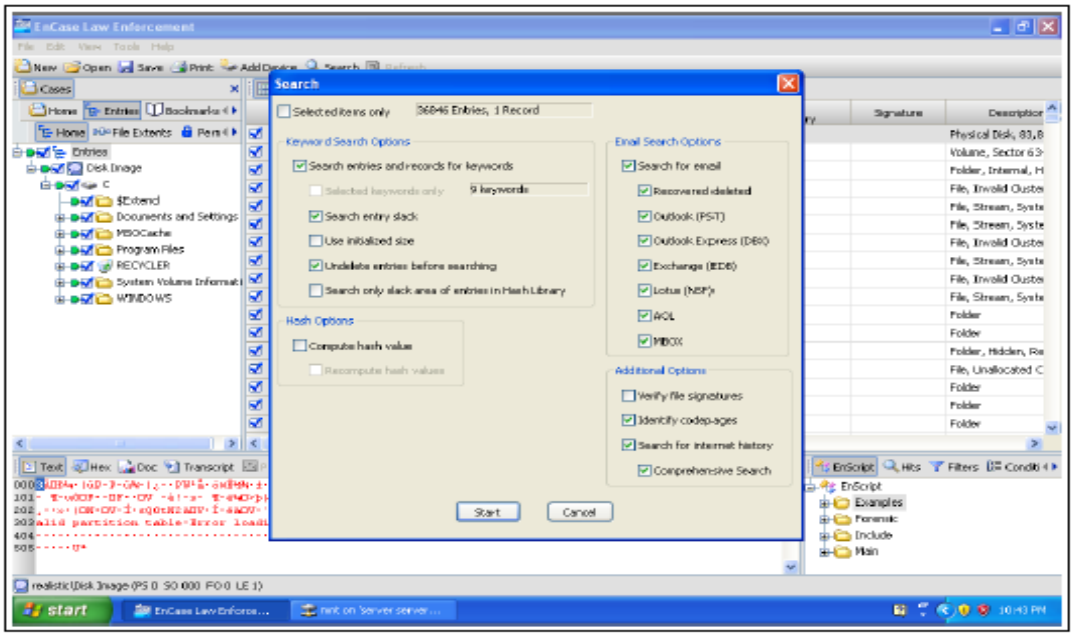

Figure 5.1: The encase user interface

5.4.2 FTK Batch Report

Like EnCase, FTK must ingest a media file to start analysis. At the ingest phase, FTK allows the user to select how in depth the case index is created, as well as what files are returned in the results. Carving options are also available on ingest, and a list of known types to carve for is presented to the user. In our tests, FTK 2.0 would hang when analyzing files shared across a network, even when the file share was mounted as a drive in windows. We were only able to achieve results with FTK when the data to be analyzed was present on a local drive. This is a significant limitation when distributed operations are desired. FTK then presents the user several views on the data analyzed. The user can explore the file structure, see all email information, see all graphics, see general file information and statistics, and run searches on the index

or the actual media based on regular expressions or words. No additional analysis can be performed, and custom scripts to extract data from known file types or to carve for other file types do not exist. FTK presents the data it finds in an easy to read view, but still presents a lot of data to the end user, and the user must search and filter the data for relevant information.

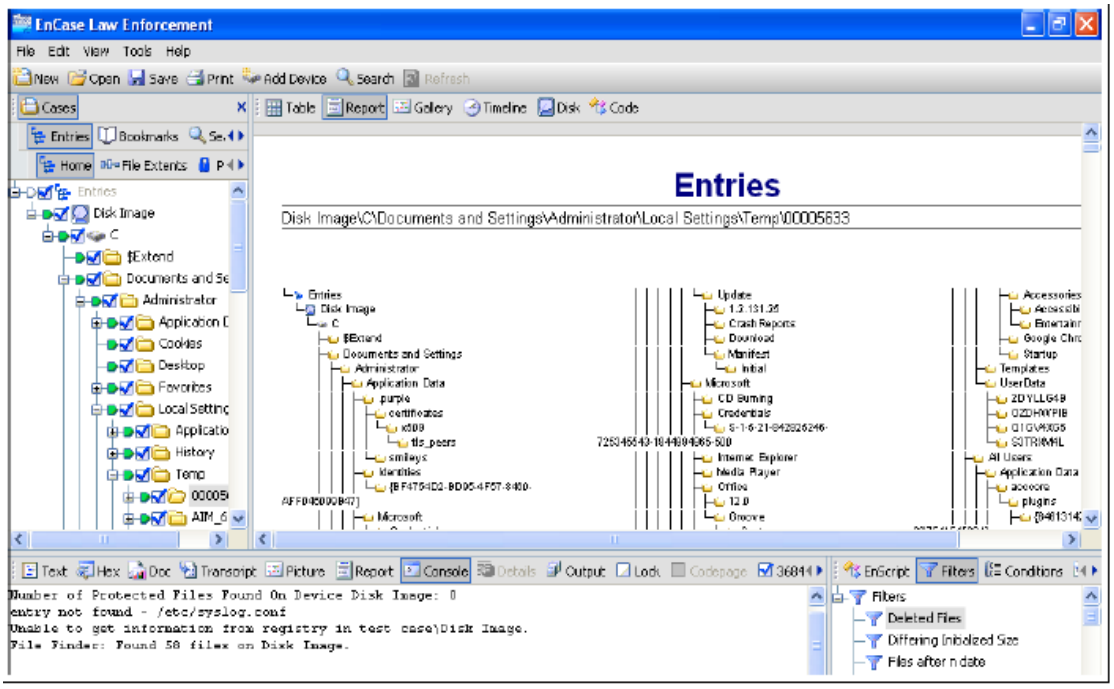

Figure 5.2: Encase Batch Report

5.4.3 TSK/Autopsy Report

Autopsy ingests do not have the same index requirement the other applications have. Autopsy performs its analysis as the user navigates the web interface, performing the analysis on the fly. However, Autopsy does require precursor steps for timeline related viewing, but only for that mode. Through the web interface, this procedure is described as “process the file system images, collect the temporal data, and save the

Table 5.2 : Analysis Time of Realistic image

data to a single file." Once processed, Autopsy recommends the user view the timeline in a separate application such as a text editor. Autopsy also provides a general report about the ingested media's metadata. Users can browse the file structure through autopsy, which issues TSK commands as the requests are made. Autopsy also allows the user to extract all strings from the media image into a file for faster searching,notherwise the media image is searched in real time for the user. Once all the scanners had run and the database was populated, the user could request that the system create a report. At this point the user can create reports from one of three categories:Disk Forensics, Network Forensics, memory Forensics. The most useful batch reporst for thisimage were the "I'm Feeling Lucky" reports—a collection of five disk forensics reports.

Name	Time
Encase	48 hours
FTK	30 hours
Sleuthkit/Autopsy	N/A(real time)
Pyflag	26 hours

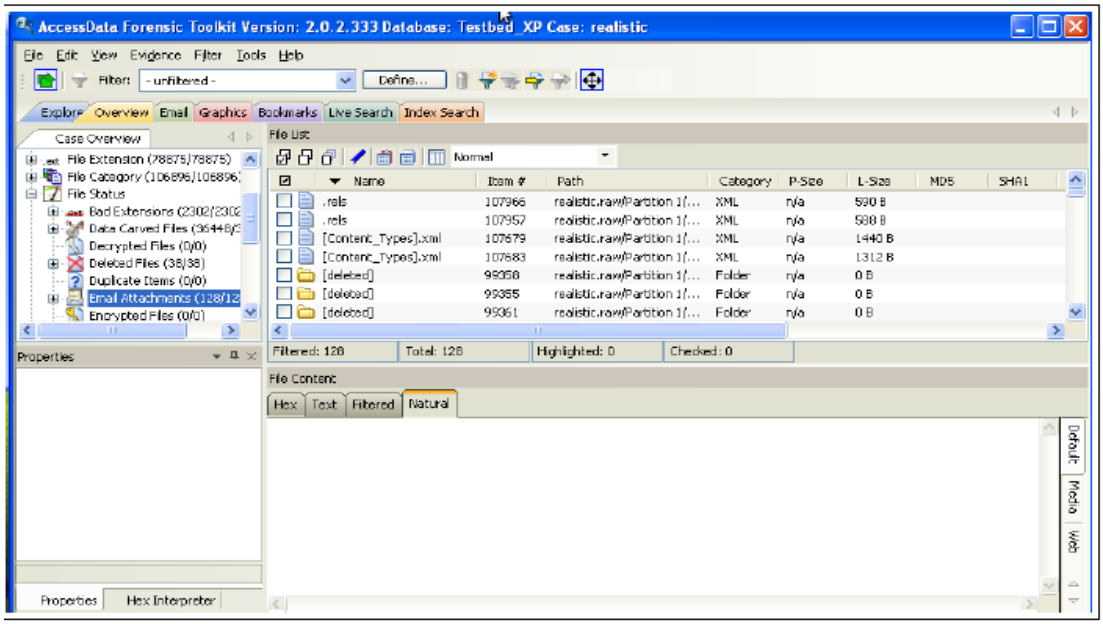

Figure 5.3 : FTK user interface

5.4.4 PyFlag Batch Report

PyFlag is a significantly more sophisticated open source forensic tool than Autopsy. Like Autopsy, PyFlag uses SleuthKit for extracting files from disk images, but it then analyzes the files using its own recursive analytical framework. PyFlag is also

capable of analyzing intercepted network packets and memory images. All in all, PyFlag is much more comparable to EnCase and FTK than to other open source alternatives. We ingested the raw image using PyFlag running on an Unbuntu Virtual Machine insideVMWare Fusion on a 2.0 GHz Core 2 Duo Macbook with 4GB of RAM. To test the limits of PyFlag's analysis, all scanners were turned on except for network scanners.

We conducted pilot user study to test the usability of the PyFlag interface for the average user.Two computer science students were selected to perform a very basic analysis of a 64MB USB flash drive with the files deleted. (A small flash drive was used so as to avoid overwhelming the students with the large realistic disk image.) The drive was formatted in FAT32 and the files deleted using the Windows command line DEL command. The drive was then imaged using dd. We also analyzed the drive image to verify each file could be recovered[20].

The primary goal of this pilot was to analyze the current PyFlag interface from the point of view of a non-expert user. The non-expert user is often overlooked by forensic tool developers, yet this is important. A case investigator often knows what information they are looking for as evidence. It could be a document containing a phone number, credit card numbers, or email communication between certain people. It could also be certain kinds of images, such as child pornography. In these cases, the case investigator may not need a digital forensic expert to find the files, they need a computer user to operate the tools available. By designing the tools in such a way that provide a forensically sound manner yet still retain ease of use for "standard" investigating, the workload could be distributed across a larger group of users. Only in cases where the subject has used methods to hide data or in cases where advanced techniques are necessary, would a forensic expert be required to step in.

The second purpose of this test was to analyze the current PyFlag reporting format from the point of view of a non-expert user. The current reporting format uses tables to show results. Each table is presented as a frame in the browser window, with the PyFlag navigation bar at the top. The table may also be alone in a separate browser window. The table view is powerful in that it lets users filter by every attribute PyFlag has recorded through a point and click menu, and view the raw SQL query

that generated the results, but it is designed for presenting tabular data in a fixed format, not for data fusion and readable display.

The list of tasks performed by the users appears in Table 3.3; The steps required are listed in Appendix with screenshots. User results and responses were then recorded quantitatively via Likert Scale scoring and qualitatively via conversation style dialogue where the author asked the user's opinions, what they liked and disliked, and what they would change about the interface for a better experience.

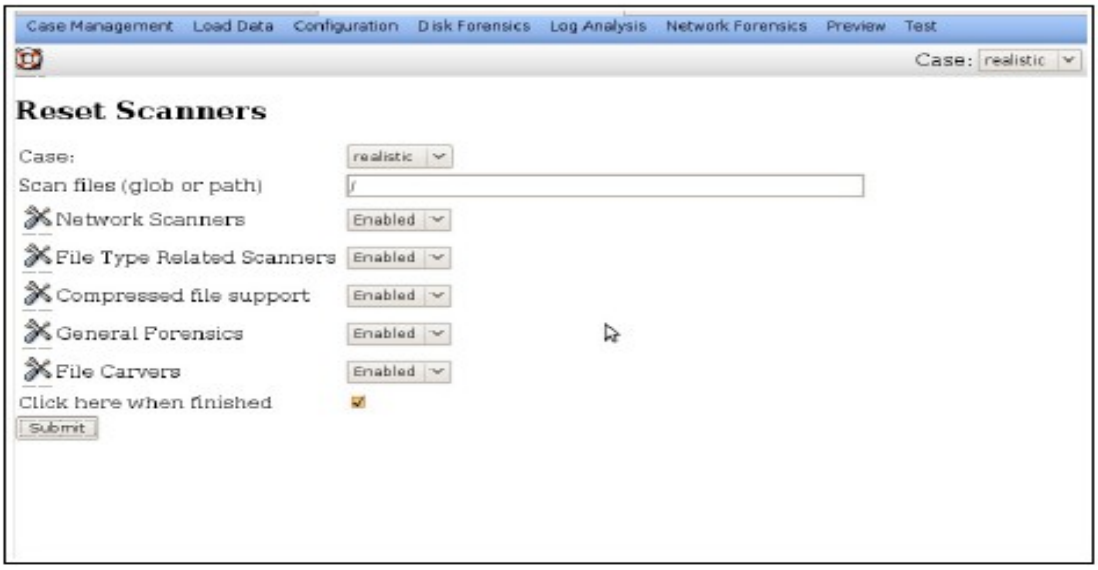

Figure 5.4: Pyflag scanning options

At the conclusion of the tasks the students were debriefed. Based on their comments and difficulties accomplishing the tasks with the provided software, we drew the following conclusions:

• PyFlag lacks a summary report to give a snapshot of the image being examined. Such a summary report would be useful[21].

• The generic report option allows the user to select from every attribute capture by PyFlag in the back end database. This allows useful information to be reported on

along with the generic information per image, but overwhelmed the users with too much information.

• For PyFlag, the Disk Forensics menu offers an option called “I’m Feeling Lucky,” (similar to the Google search option). This menu option allows searching for certain communication streams, html pages or image files in the ingested image. Calling this option “I’m Feeling Lucky” caused the users to skip over it; this was unfortunate, because it contained the most useful reports for the standard tasks at hand.

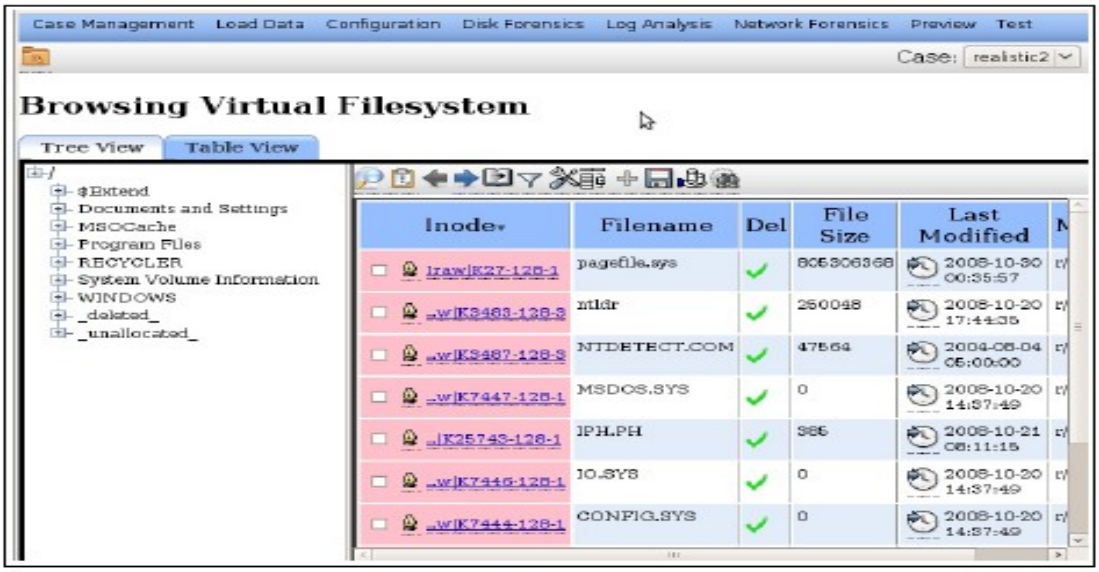

Figure 5.5: Pyflag file view

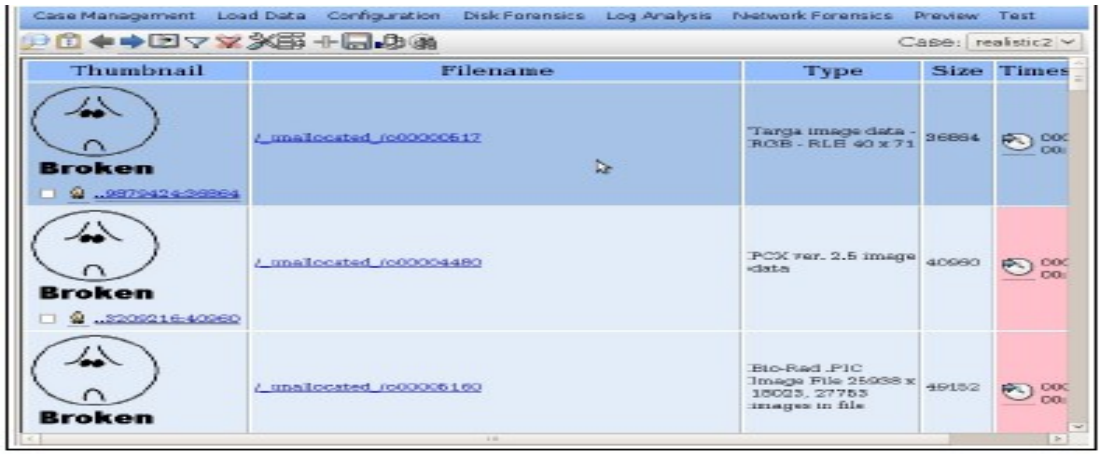

Figure 5.6 : Pyflag Image Report

- Create a new case.
- Ingest a data image
- Load a file system image
- Scan the image
- Produce a report of the file timeline of the data image
- Produce a report of the deleted items of the data image
- Produce a report of the images on the data image

Table 5.3: Tasks performed by user in pyflag user study

CHAPTER 6.

VI. Design and Implementation of Network Forensic System using Virtual Honeynet

In network security, an attack is associated with pre-event (before the attack), during-event (during the attack), and post-event (after the attack) scenarios. In earlier chapters Honeypots were shown playing a critical role against attacks in pre-event and during event scenarios. This chapter elaborates the role of honeypots in post

event scenarios. Network forensics is the science that deals with detection and investigation of intrusions in the post-event scenarios. In the starting a background of network forensics is presented. Then, Honeypots as a network forensic analysis tool (NFATs) is discussed. The chapter explains the general process model of forensics and describes the role of Honeypot in its different phases. Next, it outlines the Honeypot based Network Forensics Frameworks. Finally the major challenges in this upcoming and young discipline are presented.

Network forensics is the science that deals with capture, recording, and analysis of network traffic. Honeypots play an important role for forensics and investigation of networks. The network log data is collected from existing security products like Wireshark, etc., analyzed for attack characterization and investigated to traceback the perpetuator. Network forensics is not another term for network security. Network forensics can be considered as an essential part in Network security. Earlier, the data for forensic analysis was collected from security products like firewalls and intrusion detection systems only. With their evolution, Honeypots have become key contributor in capturing the attack data which is analyzed and investigated, hence facilitating the process of network forensics. Network forensics, however, may involve certain crimes which are legally prosecutable but which may not breach network security policies [35].

Network security protects the system against attack while network forensics does not. Network security products are generalized and look for possible harmful behaviors and they monitor the network 24 hours a day. Network forensics is post mortem investigation of the attack. It is also case specific as each crime scenario may be different and the process is initiated notitia criminis(after crime notification).Network forensics is a natural extension of computer forensics. Computer forensics was introduced by law enforcement and has many guiding principles from the investigative methodology of judicial system. Computer forensics involves preservation, identification, extraction, documentation, and interpretation of computer data. Network forensics evolved as a response to the hacker community and involves the capture, recording, and analysis of network events in order to discover the source

of security attacks. In computer forensics, the investigator and the person being investigated are on two different levels with the investigator at an advantage. In network forensics, the network investigator and the attacker are at the same skill level. The network forensic specialist uses many of the same tools and engages in the same set of practices as the person being investigated [36]. The difference in them is based on ethical standards and not on technical skills. Network forensics involves monitoring network traffic and determining if there is an anomaly in the traffic and ascertaining whether it indicates an attack. If it is so then the nature of the attack is also determined. When attacks are successful, forensic techniques enable investigators to catch the attackers. The ultimate goal is to provide sufficient evidence to allow the perpetrator to be prosecuted.

6.1 NETWORK FORENSICS

The concept of network forensics deals with the data found across a network connection, mostly ingress and egress traffic from one host to another. Network forensics tries to analyze the traffic data logged through firewalls or intrusion detection systems or at network devices like routers and switches as shown in Figure 6.1.

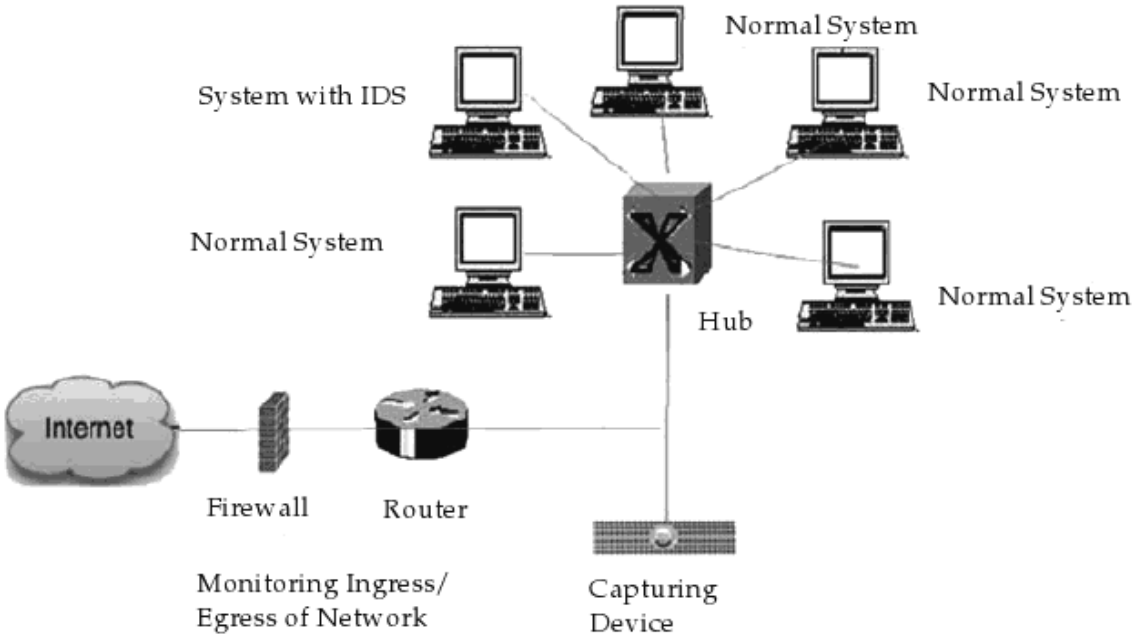

Figure 6.1: Network with traffic monitoring tools.

Network forensics is defined in as "the use of scientifically proven techniques to collect, fuse, identify, examine, correlate, analyze, 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" [37]. Marcus Ranum is credited with coining the word network forensics as "the capture, recording, and analysis of network events in order to discover the source of security attacks or other problem incidents" [38].

6.2 Motivation for Network Forensics

There are a number of forces driving network forensics and some of them are:

The increase in the number of security incidents affecting many organizations and individuals and the increase in the sophistication of the cyber attacks[39]

• The attacker is covering the tracks used to cause the attacks, making it more difficult to traceback.

• Companies doing business on the Internet cannot hide a security breach and are now expected to prove the state of their security as a compliance measure for regulatory purposes

• Internet Service Providers (ISPs) are also being made responsible for what passes over their network

• The defensive approaches of network security like firewalls and intrusion detection systems can address attacks only from the prevention, detection and reaction perspectives.

The alternate approach of network forensics becomes important as it has the investigative features. Network forensics ensures that the attacker spends more time and energy to cover his tracks making the attack costly. Network criminals are also cautious to avoid prosecution for their illegal actions. This acts as a deterrent and reduces network crime rate thus improving security.

6.2.1 Honeypot Approaches for Network Forensics

There are two ways of developing a network forensic process. One way is to reactively use traditional security products like firewalls & intrusion detection systems, analyze the data and investigate. The other way is to proactively lure the attacker by means of honeypots [40] and honeynets [41] and observe the attack patterns. The behavioral profiles of attackers are created and their exploitation mechanisms are understood. Since, a Honeynet (or high interaction honeypots) is a highly controlled network of computers, involving real operating systems and applications, designed in a way to capture all activity when attacked so full extent of the attackers' behavior can be learnt by letting these high-interaction honeypots to

interact with them [42]. The Honeynet controls the attacker's activity by using a honeywall gateway allowing inbound traffic to the victim systems and controlling the outbound traffic using intrusion prevention technologies. Virtual honeynet is another solution that allows us to run multiple platforms needed on a single computer. The term virtual is used because all the different operating systems have the 'appearance' to be running on their own, independent computer.The virtualization software allows running multiple operating systems at the same time, on the same hardware. The advantages of virtual honeynets are cost reduction and easier management, as everything is combined on a single system.

6.2.2 Honeypots and Network Forensics

In Network Forensics, Virtual honeynets can be used to gather data about the intruder's attack strategy and study the applications used to carry out the attack [43]. This approach gives an edge to network forensic investigator over the attacker, as he is aware of the attack methodology and the tools being used. The investigator is able to fingerprint the attacker with this knowledge.

6.3 Honeypot As Network Forensics Analysis Tools

Network Forensic Analysis Tools (NFATs) allow administrators to monitor the networks, gather all information about anomalous traffic, assist in network crime investigation and help in generating a suitable incident response. NFATs also help in the following
functions:

- insider theft and misuse of resources
- prediction of future attack targets
- protect intellectual property
- perform risk assessments
- exploit (break-in) attempt detection
- data aggregation from multiple sources

- network traffic recording and analysis
- determination of hardware and network protocols in use

NFATs capture the entire network traffic, allow the users to analyze the network traffic according to their needs and discover significant features about the traffic. NFATs synergize with IDSs and Firewalls and make possible the long term preservation of network traffic records for quick analysis [44].

The attack traffic can be replayed and attackers' moves can be analyzed for malicious intent. NFATs facilitate organization of the captured network traffic packets to be viewed as individual transport layer connections between machines. This enables the analysis of protocol layers and packet content. Traffic patterns are extracted between various machines. A description of a partial list of the NFATs [45] is given below:

6.3.1 Description of Network Forensic Analysis Tools

• NetIntercept: Captures network traffic and stores in pcap format, reassembles the individual data streams, analyzes them by parsing to recognize the protocol and detect spoofing and generates a variety of reports from the results.

• NetDetector: Captures intrusions, integrates signature-based anomaly detection, reconstructs application sessions and performs multi time-scale analysis on diverse applications and protocols.It has an intuitive management console and full standards based reporting tools. It imports and exports data in a variety of formats.

• NetWitness: Captures all network traffic, reconstructs the network sessions to the application layer for automated alerting, monitoring, interactive analysis and review.

• NetworkMiner: Network traffic capture by live sniffing, performs host discovery, reassembles transferred files, identifying rogue hosts and assesses how much data leakage was affected by an attacker.

• SilentRunner: Captures, analyzes and visualizes network activity by uncovering break-in attempts, abnormal usage, misuse and anomalies. It generates an interactive graphical representation of the series of events and correlates actual network traffic. It also plays back and reconstructs security incidents in their exact sequence.

• Iris: Collects network traffic and reassembles it as its native session based format, reconstructs the actual text of the session, replays traffic for audit trial of suspicious activity, provides a variety of statistical measurements and has advanced search and filtering mechanism for quick identification of data.

6.3.2 Honeypots and Network Forensics

• Xplico: Captures internet traffic, dissects the data at the protocol level, reconstructs and normalizes it for use in manipulators. The manipulators transcode, correlate and aggregate it for analysis and presents the results in a visualized form.
• Solera DS 5150 with DeepSee Suite: DS 5150 is an appliance for high speed data capture, complete indexed record of network traffic, filtering, regeneration and playback. DeepSee forensic suite has three softwares—Reports, Sonar and Search—to index, search and reconstruct all network traffic.
• PyFlag: Python Forensic Log Analysis GUI is an advanced forensic tool to analyze network captures in libpcap format while supporting a number of network protocols. It has the ability to recursively examine data at multiple levels and is ideally suited for network protocols which are typically layered. PyFlag parses the pcap files, extracts the packets and dissects them at low level protocols (IP, TCP or UDP). Related packets are collected into streams using reassembler. These streams are then dissected with higher level protocol dissectors (HTTP, IRC, etc.).
There are many other open source network security and monitoring tools which help in specific activities. These tools were designed with information security in mind rather than evidence processing and hence do not have a forensic standing. A description about a partial list of network security tools is given below [44]:

6.3.3 Description of Network Security and Monitoring Tools

• TCP Dump: A common packet sniffer and analyzer, runs in command line, intercepts and displays packets being transmitted over a network. It captures,

displays, and stores all forms of network traffic in a variety of output formats. It will print packet data like
timestamp, protocol, source and destination hosts and ports, flags, options, and sequence numbers.

• TCP Flow: Captures data transmitted as part of TCP connections (flows) and stores the data for protocol analysis. It reconstructs the actual data streams and stores in a separate file. TCP Flow understands sequence numbers and will correctly reconstruct data streams regardless of retransmissions or out-of-order delivery.

• TCP Stat: Reports network interface statistics like bandwidth, number of packets, packets per second, average packet size, standard deviation of packet size and interface load by monitoring an interface or reading from libpcap file.

• TCP Replay: Suite of tools with the ability to classify previously captured traffic as client or server, rewrite Layer 2, 3 and 4 headers and finally replay the traffic back onto the network. TCPPrep is a multi-pass pcap file pre-processor which determines packets as client or server, TCPRewrite is the pcap file editor which rewrites packet headers, TCPReplay replays pcap files at arbitrary speeds onto the network and TCPBridge bridges two network segments.

• IOS NetFlow: Collects and measures IP packet attributes of each packet forwarded through routers or switches, groups similar packets into a flow, to help understand who, what, when, where and how the traffic is flowing. It also detects network anomalies and vulnerabilities.

• Flow-tools: Library to collect, send, process and generate reports from NetFlow data. Few important tools in the suite are—flow- capture which collects and stores exported flows from a router, flow-cat concatenates flow files, flow-report generates reports for NetFlow datasets, and flow-filter filters flows based on export fields.

• NMap: Utility for network exploration and security auditing. It supports many types of port scans and can be used as on OS fingerprinting tool. It uses raw IP packets in novel ways to determine hosts available on the network, services being offered, operating systems running, firewalls in use and many other characteristics.

• Ngrep: A pcap-aware tool that allows specifying extended regular or hexadecimal expressions to match against data payloads. It can debug plaintext protocol

interactions to identify and analyze anomalous network communications and to store, read and reprocess pcap dump files while looking for specific data patterns.

• Ntop: Used for traffic measurement, traffic monitoring, network optimization & planning, and detection of network security violations. It provides support for both tracking ongoing attacks and identifying potential security holes including IP spoofing, network cards in promiscuous mode, denial of service attacks, trojan horses and port scan attacks.

• Wireshark: Most popular network protocol analyzer. It can perform live capture in libpcap format, inspect and dissect hundreds of protocols, do offline analysis, and work on multiple platforms. It can read and write files in different file formats of other tools.

• Snort: Network intrusion prevention / detection system capable of performing packet logging, sniffing and real-time traffic analysis. It can perform protocol analysis, content searching & matching and application level analysis.

• Bro: A network IDS which detects intrusions by parsing network traffic. It extracts its application-level semantics and executes event- oriented analyzers to compare the activity with patterns deemed troublesome. It is primarily a research platform for traffic analysis and network forensics.

• Argus: Processes packets in capture files or live data and generates detailed status reports of the 'flows' detected in the packet stream. The flow reports capture much of the semantics of every flow with a great deal of data reduction. The audit data is good for network forensics, non-repudiation, detecting very slow scans, and supporting zero day events.

• P0f: Passive OS fingerprinting by capturing traffic coming from a host to the network. It can also detect the presence of firewall, use of NAT, existence of a load balancer setup, the distance to the remote system and its uptime.

• SiLK: System internet Level Knowledge supports efficient capture, storage and analysis of network flow data based on Cisco NetFlow. The tool suite, consisting of collection and analysis tools, provides analysts with the means to understand, query, and summarize both recent and historical traffic data in network flow records. SiLK supports network forensics in identifying artifacts of intrusions, vulnerability exploits,

worm behavior, etc. SiLK has performance as a key element and manages the large volume of traffic by storing only the security related information and splits files into predefined categories to reduce lookup time.

6.4 Design Architecture of Network Forensic System using Virtual Honeynet

6.4.1 Honeypot System

Honeypot system is also called "Malware Collection System". The purpose of honeypot system is to protect the network, detect and scatter attacks from external attackers and delay the attack on the real objective, to reduce information security risks. At the same time, the system simulates the system vulnerability for the attackers to attack, and find out the attacker. According to the level of intruder's interaction, the category of honeypot system has three different types in interaction frequency [45], low interaction honeypot, medium interaction and high interaction honeypot. In functional point of view, it is divided into production honeypot and research honeypot.

We used high interaction honeypot-client honeypot and low interaction honeypot system to provide a different aims system for other users and researchers to use. We also create two honeypot systems in our honeypot system module, and locate them in two different IP network on the Internet.

6.4.2 Honeywall System

It is an open source tool and it act as a gateway for honeypots. All the attackers will pass through this gateway when they will attack the system. All the logs are generated in the database through this honeywall. The architecture working is totally based on Traditional server . In this System the honeypot attract the attackers so that their process methodology can be observed and analyzed to improve defense mechanisms. So attackers first will go through honeywall and then honeypot system will activate and it will the machine through which we will intereact to the attacker via honeypot system and all the activities will be observed through honeywall in the database.

When attacker will attack or intereact with the system Network packets are being logged and dumps are being created ,Connections are being logged and IDS alerts are being generated on the web interface we can download the pacap file which shows the all log file of attacker[47].

6.4.3 Development efforts:

- Design of Network Forensic System Based on Virtual Honeynet
- Implemented and setup has been created using Virtual Box running multiple honeypots in virtualized environment.
- Implemented Honeywall (Open Source Tool) which have an integration of multiple tools and acts as gateway for honeypots.
- Developed Virtual Honeynet based framework for network trace collection and analysis. (Investigation will be Offline using deep packet inspection).

6.4.4 System Requirements:

- Red Hat Enterprise Edition OS.
- Virtual Box
- Fuse, bridge utilities and other required packages
- Honeywall Roo
- Honeypot Images (Window XP, Nepenthes etc)
- IP addresses to assign to Virtual Honeypot (at least 2 IP address)

6.4.5 Features of Honeypot based Network Forensic System:

- Attract the attackers so that their process methodology can be observed and analyzed to improve defense mechanisms.
- Network packets are being logged and dumps are being created
- Connections are being logged

- IDS alerts are being generated.
- Network statistics can be generated.
- Honeypot model is helpful in improving the defensive mechanism.

6.5 **System Design Architecture**

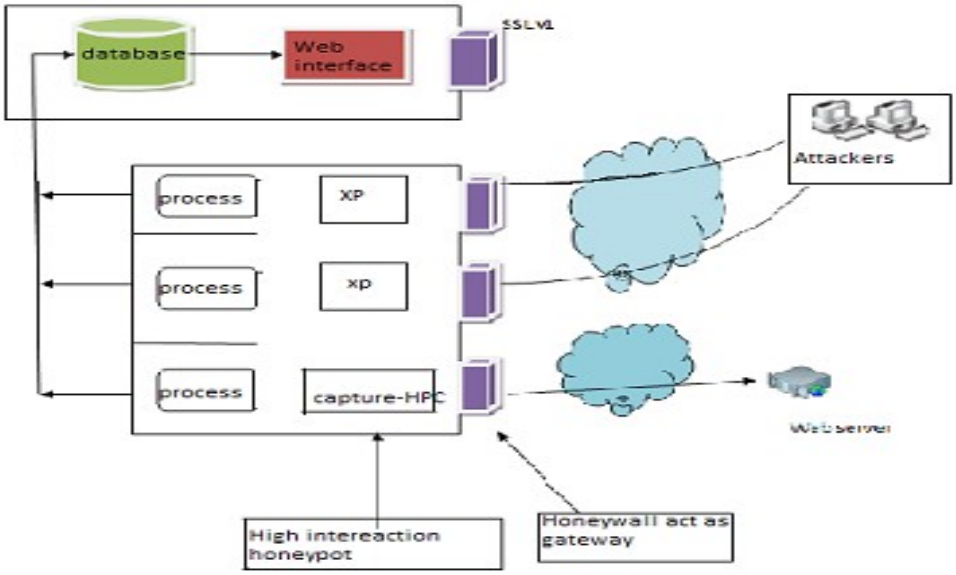

Figure 6.2: Architecture of Network Forensic System

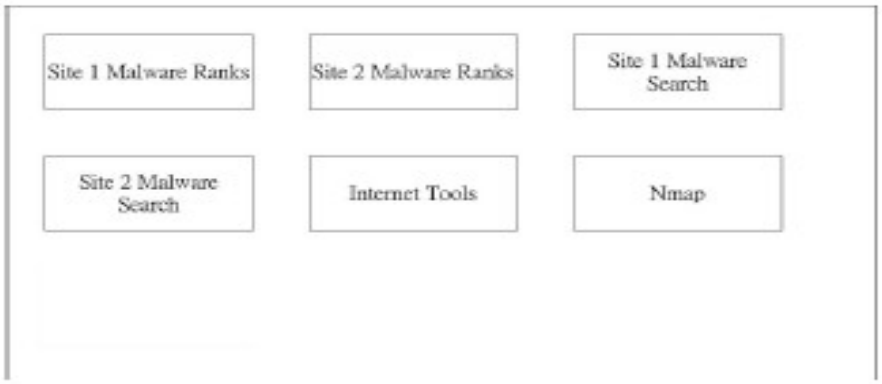

Figure 6.3 : Web Interface Function

6.6 Performance and Results

6.6.1 Snapshots

In my thesis work we build a test setup “Network Forensic System using Honeypots” which shows the log file in the database log and we can check these logs on the web interface through which we check the attackers activities**.**

Step 1: In this step we used a command “./startnet.sh setup” in the root terminal to generate our test setup.

Figure 6.4: To generate the Test setup

Step 2: In the next step we used a command "./Startnet.sh start" to start the working of our test setup.

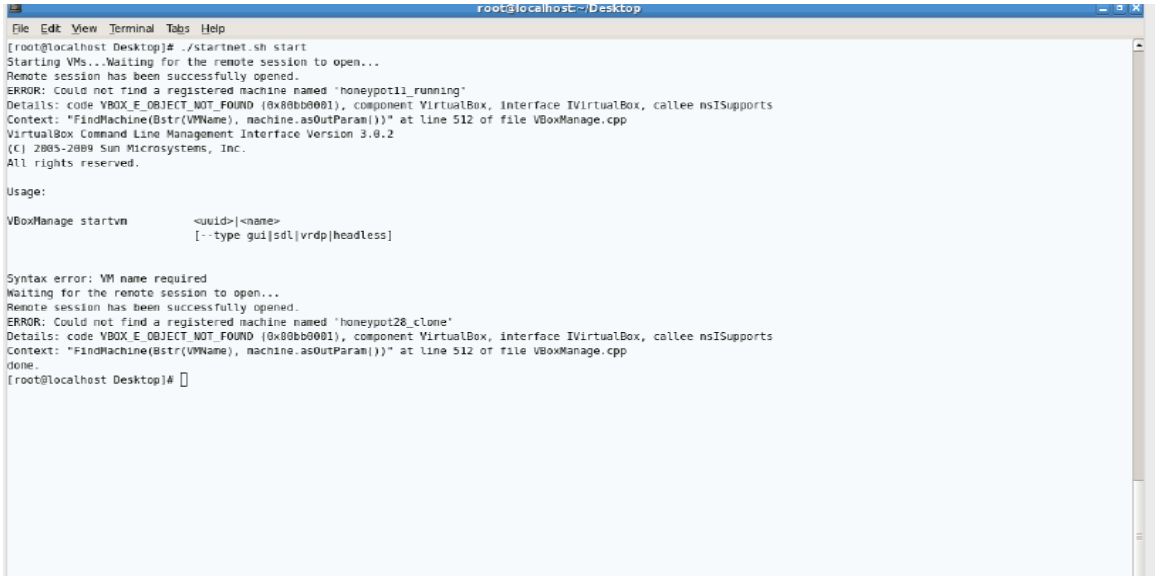

Figure 6.5: Starting of the test setup

Step 3:In this step assign two different ip address one to the base machine and one to the honeypot system

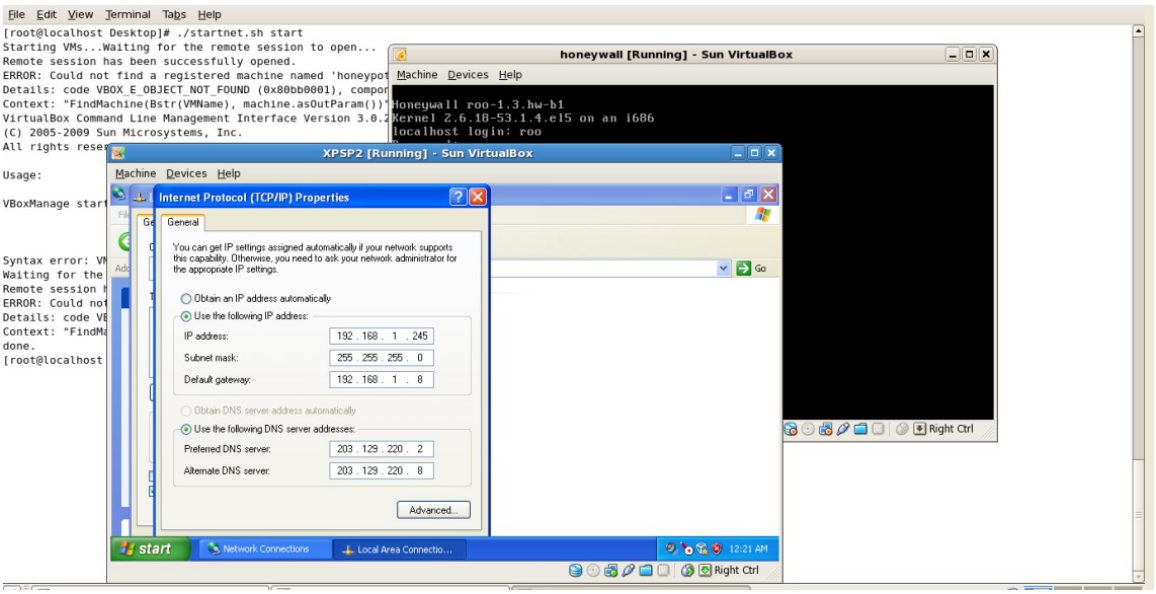

Figure 6.6: Assigning of ip address

Step 4: Honeypot and Honeywall in running mode

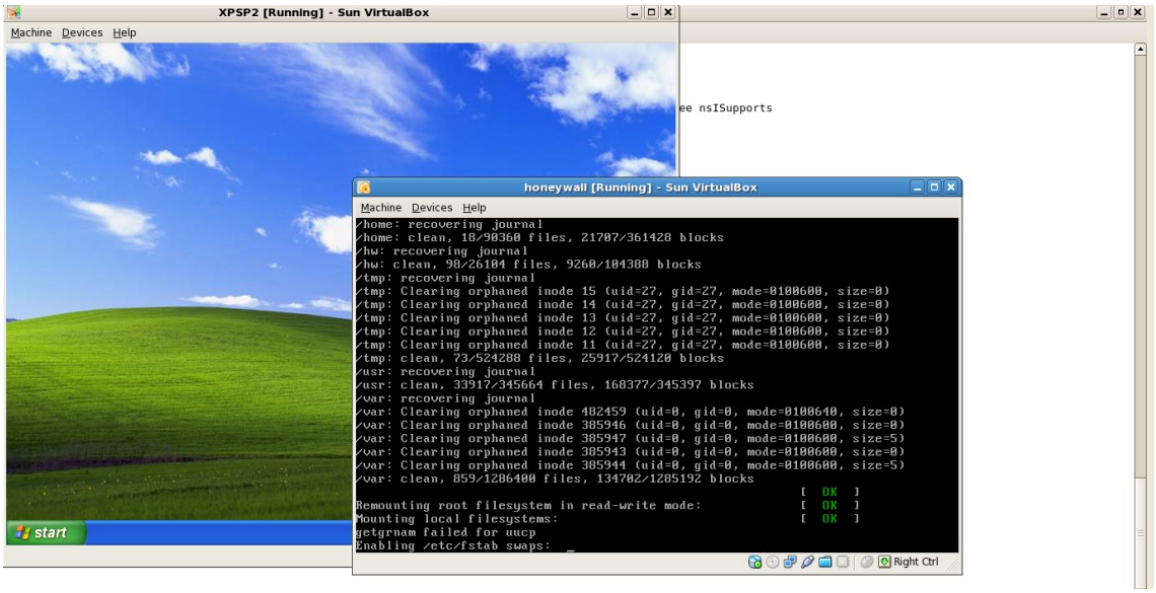

Figure 6.7: running state of honeywall & honeypot

Step 5: Continued……with working of honeywall and honeypot

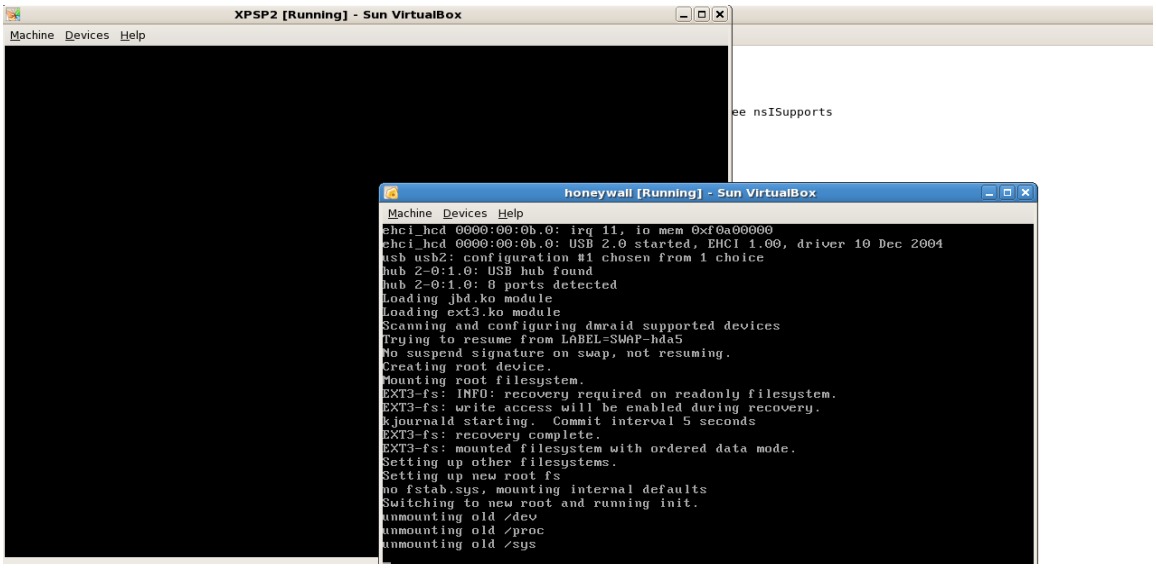

Figure 6.8: Interface of honeywall and honeypot

Step 6: After that to download pcap file we used command cd /var/log/pcap

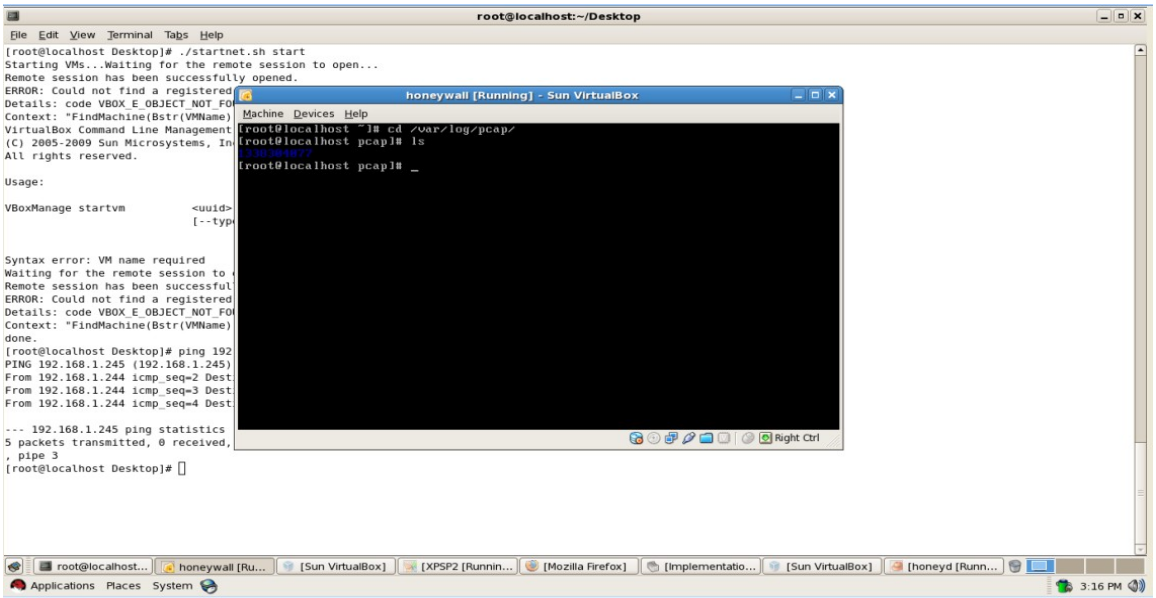

Figure 6.9: checking of pcap file

Step 7:In this Step we can check the snort alerts generated using the given command

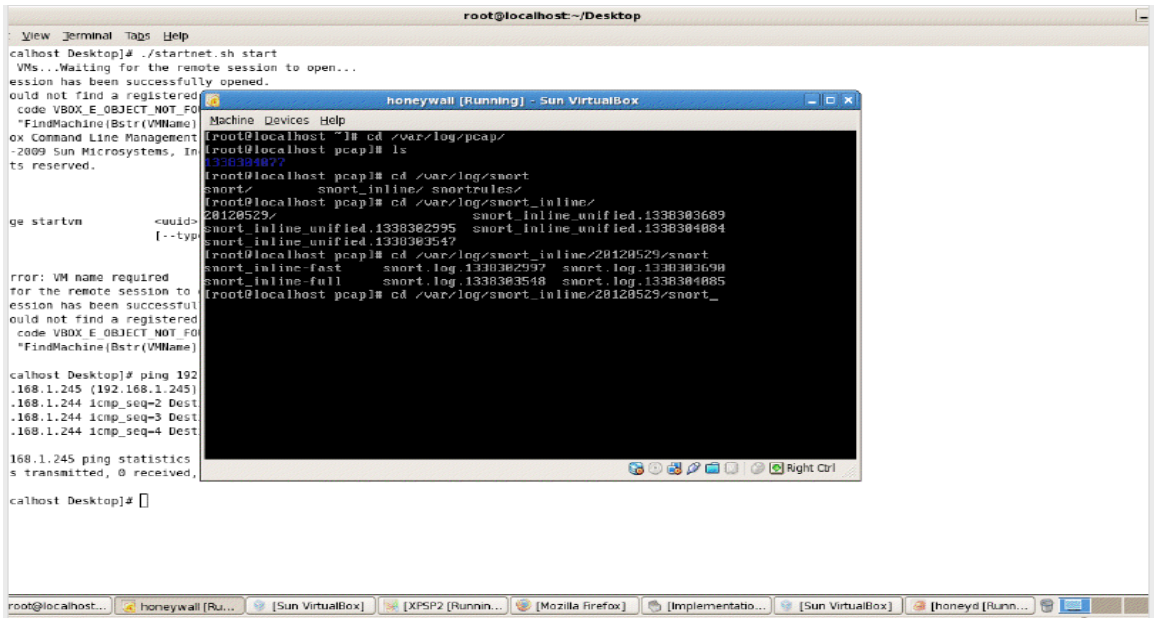

Figure 6.10: Snort alert generation checking

Step 8: In this step we will put the local host address and port number on the web interface.

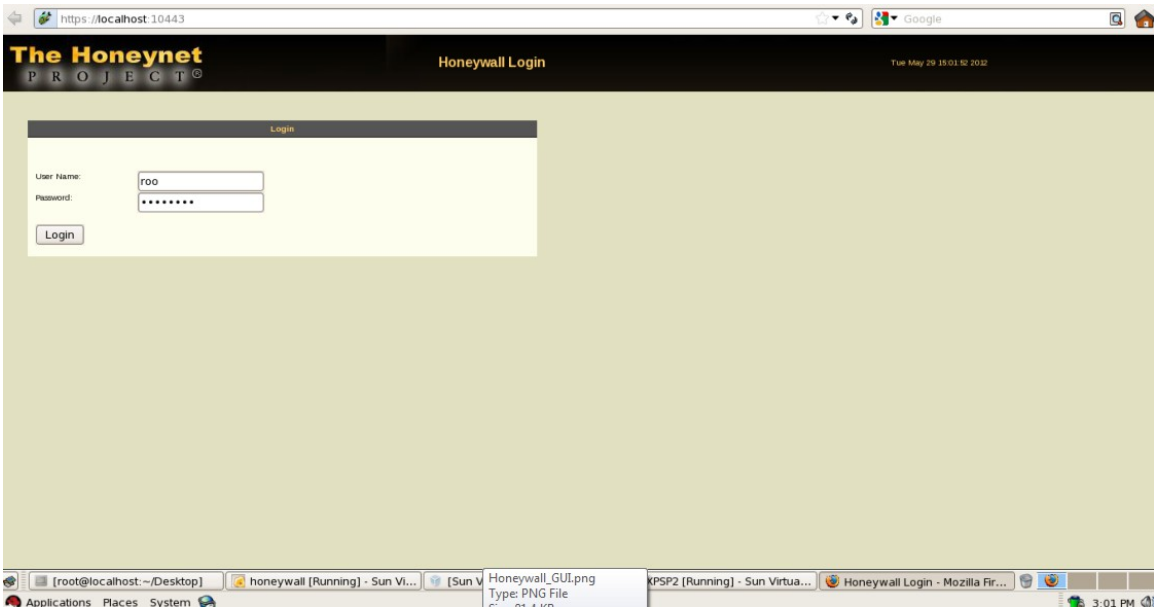

Figure 6.11: Honeywall login interface on web interface

Step 9: After login it will show the pcap file

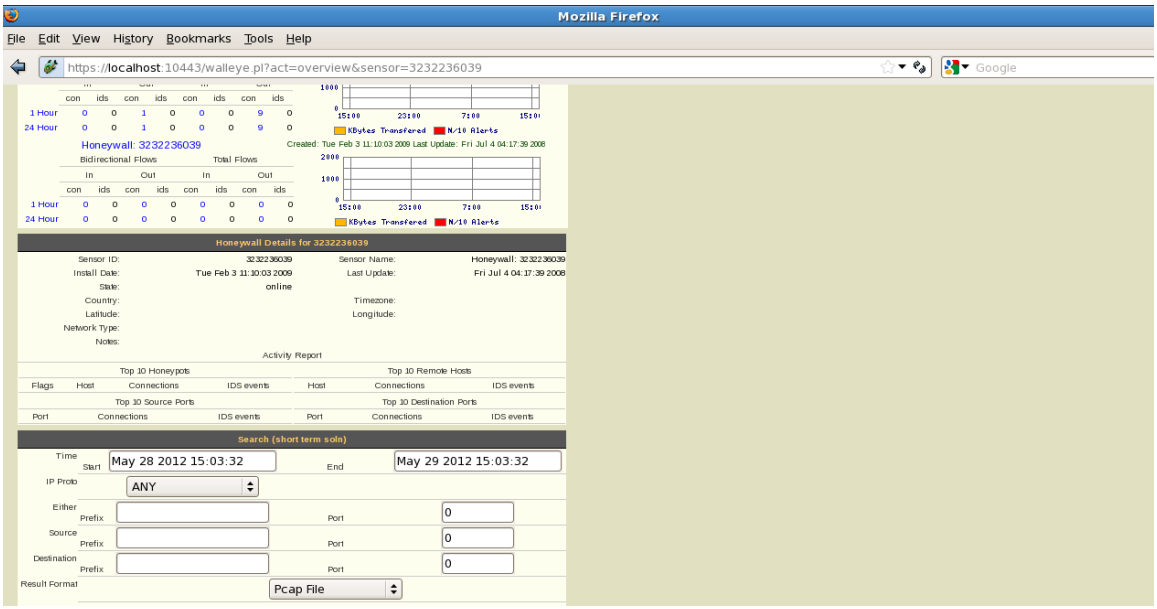

Figure 6.12: Downloading of pcap file

Step10:clicking on pcap file it will show you the option to save the pcap file

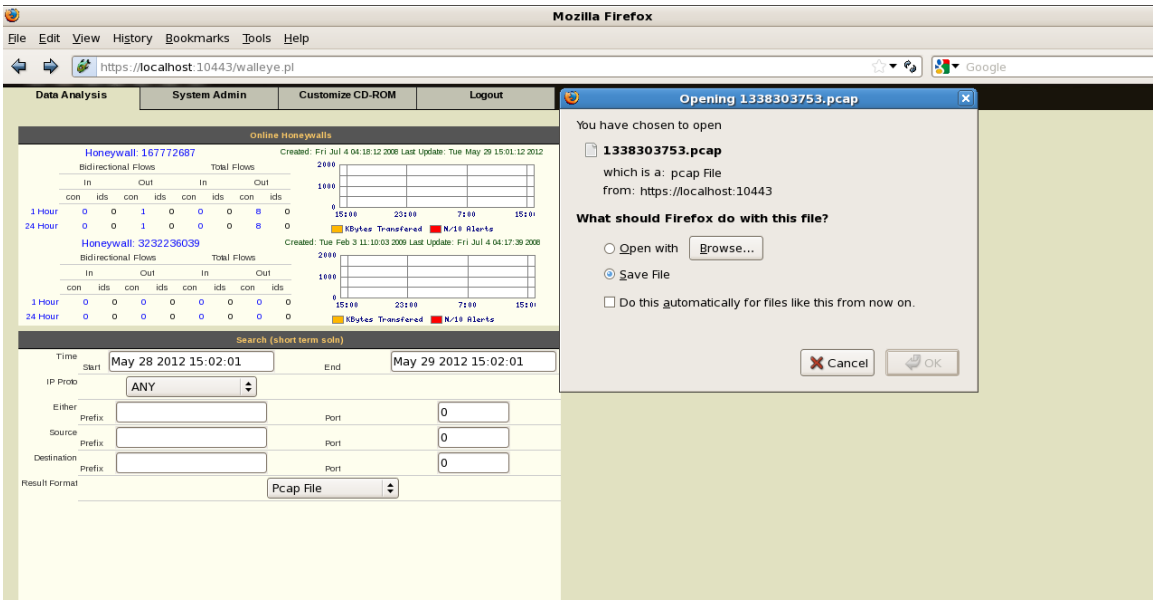

Figure 6.13 :Saving of pcap file

Step11:In this step in the database management we can set manually the time period to generate the pcap file database according to investigation.

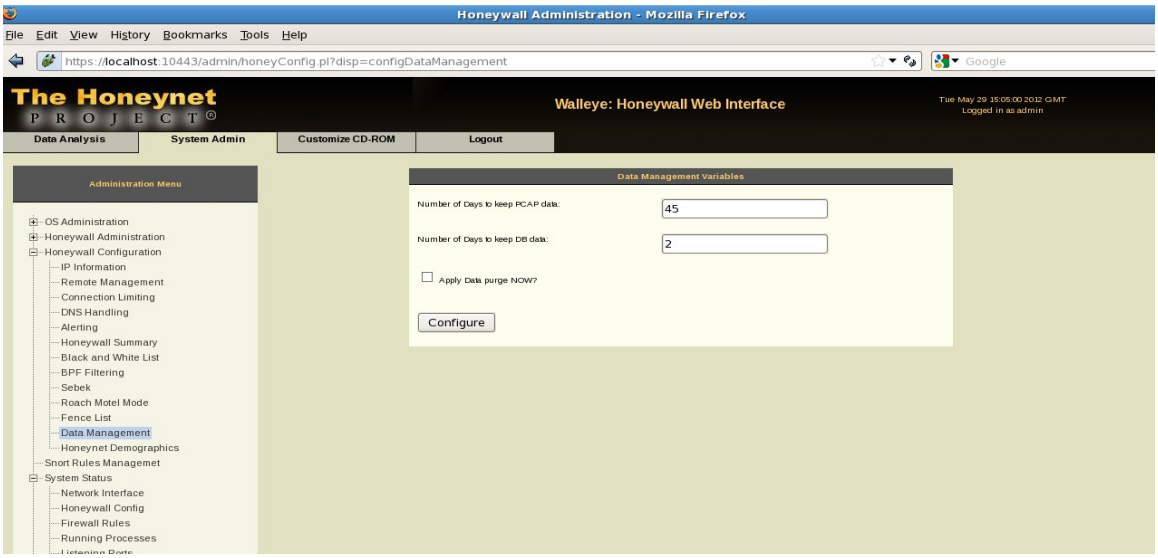

Figure 6.14 :Time management to database management

Step 12: Limit given to udp and tcp packets for the transferring of data

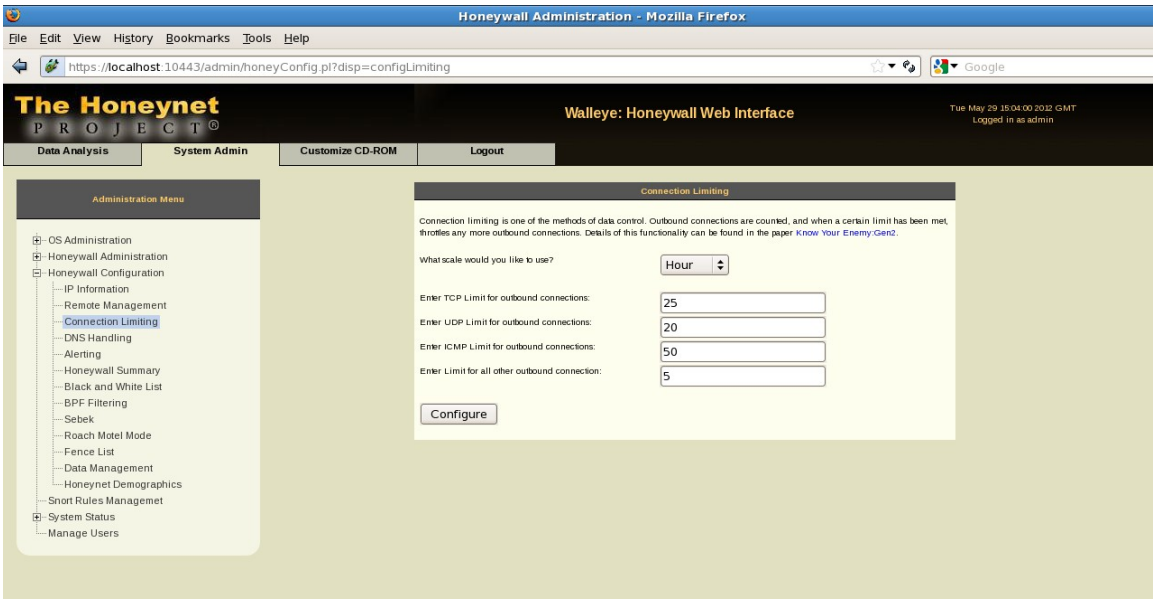

Figure 6.15 : Connection limiting

Step13: System Admin information the network interface

Figure 6.16 : Network interface information

Step14: udp and tcp packets transferring information

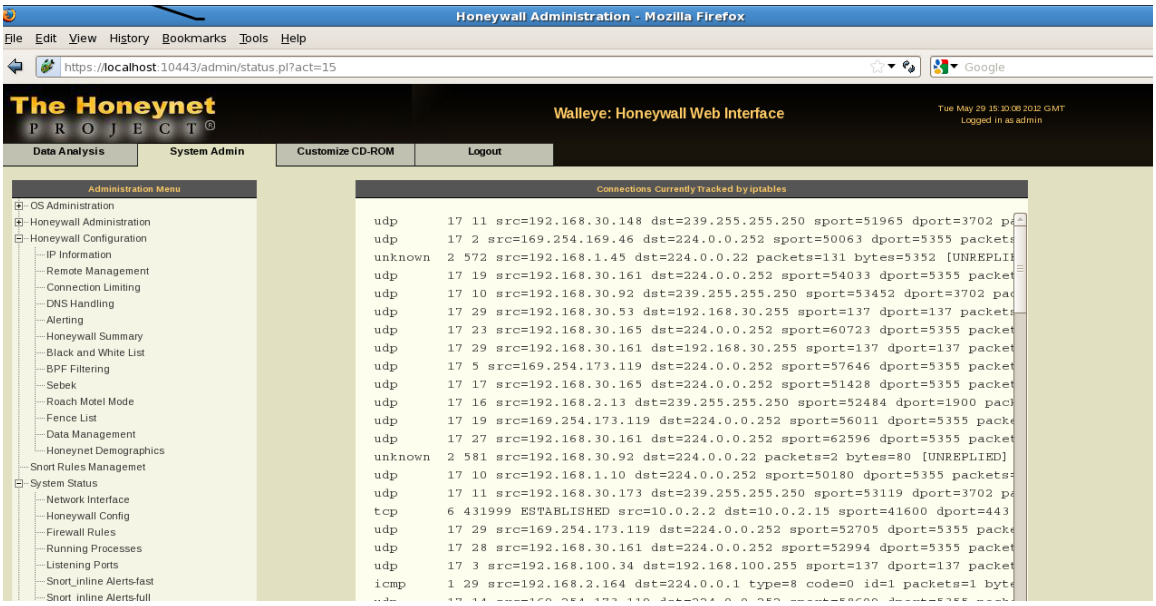

Figure 6.17 : Transfer of Udp & Tcp packets

Step15: To check the snort log analysis generated on web interface

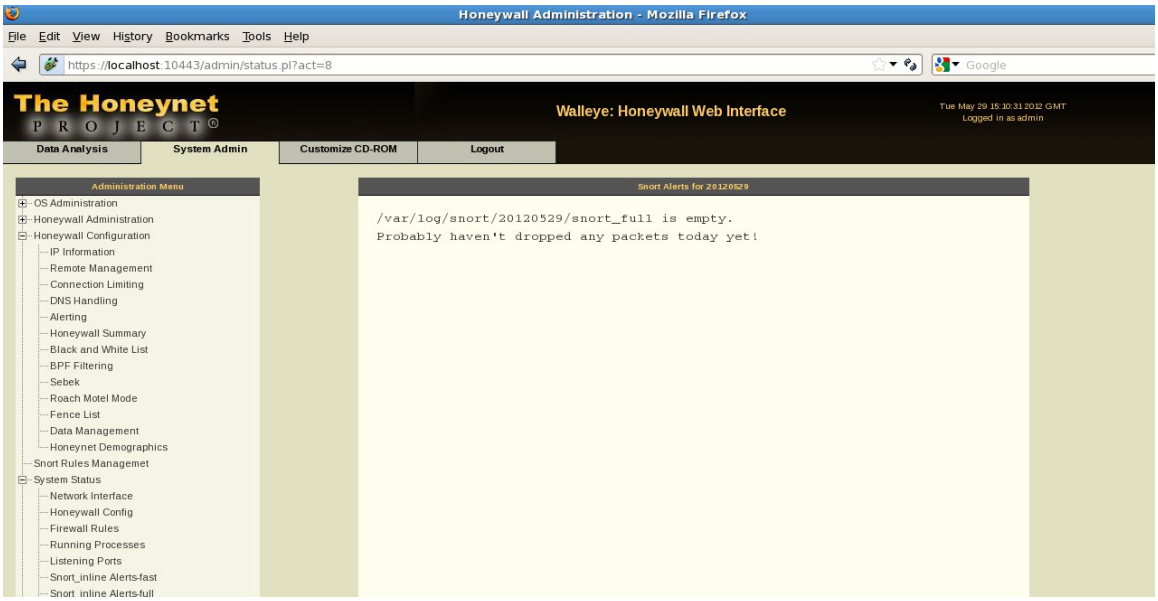

Figure 6.18: Snort alert generated file

CHAPTER 7.

7.1 Conclusion

Network forensics ensures investigation of the attacks by tracing the attack back to the source and attributing the crime to a person, host or a network. It has the ability to predict future attacks by constructing attack patterns from existing traces of intrusion data. The incident response to an attack is much faster. The preparation of authentic evidence, admissible into a legal system, is also facilitated. We have analyzed and compared different approaches used for network forensic system. We have developed automated prototype for network attack data collection based on Virtual Honeynet and we found Honeypot model is helpful in improving the defensive mechanism. Honeypots based Model can be very useful to collect the attacker traces as anything coming on the honeypot is malicious by nature. From an investigative perspective, a honeypot is an ideal tool to closely study attackers and capture their tools, keystrokes, etc.

7.2 Future Scope

In this research thesis, we have presented detailed study and an exhaustive survey of the several tools and techniques available to conduct network forensics and develop a solution which is better suitable to collect the attackers' traces so that we can further investigate the attack traces. We described the Honeynet Architecture and the use of Honeypots, both and physical and virtual, in detecting malicious attack traffic and protecting the production systems. In general, the security and forensic personnel need to keep up pace with the latest attack tools and techniques adopted by the attackers. With the developed solution, the deployment in distributed environment would lead to better and good volume of attack data which are always useful for investigation purpose. Future work would also involve exploring the tools and techniques available for wireless network forensics. And also it is just our initial efforts to develop the network based forensic system, scalability is also one of major future work involved.

List Of Publications

[1] Jatinder kaur, Gurpal Singh, “Comprehensive Study Of Digital Forensics”, in IJARCET 2012 (International Journal Of Advanced Research In Computer Engineering And Technology), ISSN Bhopal section (Status-Accepted).

[2] Jatinder kaur, Gurpal Singh, Manpreet Singh “Design And Implementation Of Linux Based Network Forensics System Using Honeynet”, in IJARCET 2012 (International Journal Of Advanced Research In Computer Engineering And Technology), ISSN Bhopal section (JUNE 2012) (Status-Accepted).

References

[1] Council of Europe (CoE), "*Explanatory Report to the Convention on Cybercrime*" (ETS 185), 2001.

[2] Post, D., Text presented at the Symposium "*The Internet and the Law: a Global conversation*", Ottawa 1-2October 2004.

[3] Kenneally, E.K., "*The Internet is the Computer: the role of forensics in bridging the digital and physical divide*", Digital Investigation, Vol. 2, 2005, pp. 41-44.

[4] E-crime Watch Survey 2011, Press Release , U.S. Secret Service, CERT Coordination Center.

[5] Berghel H., "*The Discipline of Internet Forensics*", Digital Village, Communications of the ACM, August 2003/Vol. 46, No. 8, pp. 15-20

[6] Hal R. Varian and Peter Lyman, 2003. http://www2.sims.berkeley.edu/ research/projects/how-much-info-2003/. [Online; accessed 28 September 2008].

[7] Cyberlaw India, Available at http://www.cyberlawsindia.net/computer-forensics.html

[8] Bret Swanson. The coming exaflood-bret swanson. The Wall Street Journal, 2(26), 2007

[9] Peter Stephenson, Richard Walter, "*Cyber Crime Assessment*" 45th Hawaii International Conference on System Sciences, 2012

[10] http://www.slideshare.net/tabrezahmad/cybercrime-investigation

[11] Emmanuel S. Pilli "Network forensic frameworks: Survey and research challenges" Journal of Elsevier Ltd. 2010

[12] S. Garfinkel, "*Network Forensics: Tapping the Internet*"

[13] V. Broucek and P. Turner, "*Forensic computing: Developing a conceptual approach for an emerging academic disciple Australian Security Research Symposium*", July, 2001

[14] Simson L. Garfinkel, David J. Malan, Karl-Alexander Dubec, Christopher C. Stevens,and Cecile Pham " *Disk imaging with the advanced forensic format, library and tools*" Springer, January 2006

[15] Guidance Software. Guidance software form 10-k, 2007. http://investors.guidancesoftware.com/secfiling.cfm

[16] Dan Farmer and Wietse Venema. The coroner's toolkit (tct). http://www. porcupine.org/forensics/tct.html.

[17] Forensic toolkit 2.0, 2008. http://www.accessdata.com/Products/ftk2test.aspx.

[18] M.I. Cohen. Pyflag " *An advanced network forensic framework*" In Proceedings of the Digital Forensics Research Workshop 2008

[19] Digital Forensic Research Workshop *" forensics challenge results"* , 2008

[20] Cohen M, Collett D. Python forensic log analysis GUI (PyFlag)

[21] James, S., Nordby, J., et al. Forensic Science – "*An Introduction to Scientific and Investigative Techniques*" , Press 2005

[22] Richard Bassett, Linda Bass and Paul O'Brien, "*Computer Forensics:An Essential Ingredient for Cyber Security*" Journal of Information science and Technology, Vol 3(1), 2006

[23] U.S. Department of Justice (DOJ), "*Electronic Crime Scene Investigation: A Guide for first responders*", United States Department of Justice, Washington DC, 2001

[24] A. Yasinsac, R. Erbacher, D. Marks, M. Pollitt, and P. Sommer, "*Computer Forensics Education*", Security & Privacy, IEEE, pp. 15–23, 2003

[25] F. Pouget, M. Dacier " *Honeypot-based Forensics*" , Institut Eurécom. 2229, route des Crętes; BP 193

[26] http://scripts4cf.sourceforge.net/tools.html

[27] http://www.netmon.ch/allin1.html

[28] http://www.sleuthkit.org/

[29] http://wiki.sleuthkit.org/index.php?title=Tools_Using_TSK_or_Autopsy.

[30] http://s-t-d.org/tools.html

[31] http://www.ilook-forensics.org/

[32] http://www.downloadatoz.com/utility_directory/forensic-toolkit/

[33] 2011 Cyber Security watch Survey, Software Engineering Institute, Carnegie Mellon University. Available at www.cert.org

[34] Disley V.N., "Nailing the Intruder", SANS Institute, 2001. Available at http://www.sans.org

[35] V. Broucek and P. Turner, "*Forensic computing: Developing a conceptual approach for an emerging academic discipline*", in 5th Australian Security Research Symposium, 2001.

[36] H. Berghel, "*The discipline of Internet forensics*", Communications of the ACM, vol. 46, p. 20, 2003.

[37] G. Palmer, "*A road map for digital forensic research*", in First Digital Forensic Research Workshop, Utica, New York, 2001, pp. 27–30.

[38] M. Ranum, "network forensics " in http://www.ranum.com.

[39] S. Perry, "*Network forensics and the inside job*", Network Security, vol. 2006, pp. 11–13, 2006.

[40] "*Honeynet Project: Know Your Enemy: Honeynets—What a honeynet is, its value, how it works, and risk involved*," in http://old.honeynet.org/ papers/honeynet/.

[41] L. Spitzner, "*Know Your Enemy: Defining Virtual Honeynets*," in http:// www.honeynet.org/.

[42] D. Moore, C. Shannon, G.M. Voelker, and S. Savage, "*Network telescopes: Technical report*", CAIDA, April, 2004.

[43] W. Harrop and G. Armitage, "*Defining and evaluating greynets (sparse darknets)*", in Local Computer Networks 2005. 30th Anniversary. The IEEE Conference on, Sydney,NSW, 2005, pp. 344–350.

[44] V. Corey, C. Peterman, S. Shearin, M.S. Greenberg, and J. Van Bokkelen, "*Network forensics analysis*", IEEE Internet Computing, vol. 6, pp. 60–66, 2002.

[45] E.S. Pilli, R.C. Joshi, and R. Niyogi, "*Network forensic frameworks: Survey and research challenges*", Digital Investigation, vol. 7, pp. 1–12, April 2010.

[46] "ISO/IEC 27001: 2005," in http://www.iso.org/iso/catalogue_detail. htm? csnumber=42103

[47] W.Y. Chin, E.P. Markatos, S. Antonatos, and S. Ioannidis ,"*HoneyLab: Large-Scale Honeypot Deployment and Resource Sharing*", Network and System Security, pp. 381-388, Jul 2008

[48] A. Almulhem, "*Network Forensics: Notions and Challenges*" ,in Proceedings of 9th IEEE International Symposium on Signal Processing and Information Technology (ISSPIT 2009) UAE, Dec 2009

[49] Yanet Manzano and Alec Yasinsac, "*Policies to Enhance Computer and Network Forensics"*, 2nd Annual IEEE Systems, Man, Cybernetic Information Assurance Workshop,June 2001