ISSN: 2278 – 1323
International Journal of Advanced Research in Computer Engineering & Technology
Volume 1, Issue 4, June 2012

Design & Implementation of Linux based Network Forensic System using Honeynet

Jatinder Kaur, Gurpal Singh, Manpreet Singh

SMCA, Thapar University, Patiala -147004, India

CSE , Ramgharia College, Phagwara, India

jyoti929@gmail.com, gurpalsingh123@gmail.com, sunny_minhas@rediifmail.com

Abstract— Network Forensics is scientifically confirmed techniques to collect, detect, identify, examine, correlate, analyze, and document digital evidence from multiple systems for the purpose of finding the fact of attacks and other problem incident as well as perform the action to recover from the attack. Network Forensic measures the 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.
In this paper we designed a Linux based Network Forensic system in which contented virtual honeynet system to solve the information gathering in the past . This system is totally based on traditional server honeypot. It helps organizations in investigating outside and inside network attacks. It is also important for law enforcement investigations.

***Index Terms*—Network Forensics, Malware, Honeypot, Log Analysis, Honeywall .**

I. INTRODUCTION

Network Forensics is the science that deals with capture, recording and analysis of network traffic. The concept of network forensics deals with the data found across the network connection and egress traffic from one host to another. Network forensic tries to analyze traffic data logged through firewalls or intrusion detection system or at network devices like routers and switches.
Researchers utilize the open source software to collect and analyze malicious network behaviours from the Internet, and to collect the real time log information about the malware attacking. Honeypots play an important role for forensics and miscellaneous traffic. Network forensic can be considered as an essential part of the Network security[1]. The Honeypot has proved to be a very effective tool in proving more about Internet crime like credit card fraud or malware propagation. Earlier, the data for forensic analysis was collected from security products like firewalls, and intrusion detection system only. But With their evolution, Honeypots have become key contributor in capturing the attack data which is analyzed and investigated[2].

For increasing the sample numbers of malicious attacking information, we decide to use high interaction honeypot to collect and analyze the category of the attacks in the form of logs, through the attacker. Our contribution in this research has automated implementation of Linux based virtual Honeynet in context of network forensic system by using honeynet technology to collect the network attack traces which can lead to further investigation either using some tools or manually. In this paper we have used the Honeynet technology for implementation of Linux based network forensic system in our context using Open Source Virtual Box for virtualizations. The collected information can also provide to the network forensic investigators as the evidence of crime.

II. BACKGROUND AND RELATED WORK

Network forensics is defined in [2] as "the use of scientifically verified techniques to collect, combine, identify, examine, correlate, analyze, and document digital evidence from multiple, actively processing and transmitting digital sources for the purpose of finding 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.
Network Forensics system can prove valuable investigation 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.
In some studies we found researchers used some software, such as honeytraps and NFAT [3] as the network forensic tools on collecting log part, and some use the Multi-source logs as the foundation analysis.
Honeytrap is a low-interaction honeypot that also aims to collect malware in an automated way. Some studies use the capturing methods to collect evidences and logs on networks. They also built the information platform for network

administrators to use on network forensics. As the result, this study is different from other normal analysis in monitoring all information of traffic packets. We provide the evidences by processing the network forensics method to collect malware behaviours, in order to ensure the effectiveness of digital evidence and credibility of the evidence on judicial review.

A. MALWARE ANALYSIS

There are mainly two approaches in malware analysis. One is static analysis and the other is dynamic analysis [4]. The major difference is the dynamic analysis has to simulate a network environment on the server, but not the static analysis. The static analysis is a white-box approach in which the purpose of analysing the malware samples to help the network administrators or IT staffs understands the function of the malware. The most difficult part in this analysis is how it can be done to analyze the malware when it's well unseen. Therefore, we need to use virus scanner, such as AVIRA, and BitDefender to analyze and define the categories of the threats.
The dynamic analysis is a black-box approach, in which apply the sample malware in an emulation network environment on the server. Using the dynamic analysis tool such as Autoruns and Capture-BAT observes the action detail of the malware. Internally, it saves and access the file, DLLs, registry, and API procedure call. Externally, it monitors the server access, malware comportment scanning, and malware downloading. For getting a great quantity of information, finding and analyzing various threats programs compartment is necessary.

B. 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 [5]. According to the level of intruder's interaction, the category of honeypot system has three different types in interaction frequency, low interaction honeypot, medium interaction and high interaction honeypot. In functional point of view, it is divided into production honeypots and research honeypot[6].
We used high interaction honeypot-client honeypot 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 [7].

C. HONEYNET : HIGH INTEREACTION 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, Honeynets is 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.

D. HONEYPOT BASED NETWORK FORENSIC SYSTEM

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 and honeynets 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 [10]. 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

CAPTURE-HPC :

Using Capture-HPC , one kind of honey client, as a high interaction web page testing is because it collects different categories of threats than the Nepenthes [11][12]. It focuses on drive-by-download or the threats links in the web page. These attacking happen in an unaware or miss understand situation when users are browsing a web page or reading a HTML type files. These threats attack the common vulnerability in the application software, such as web browser, Flash, PDF, Office, etc. Through these common vulnerabilities can affect the user computers by making the client application apply the threats programs. These web page attacking are also called client-side attack.

E. 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 securi ty 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[9].

F. HONEYPOT AND NETWORK ANALYSIS TOOLS

• 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 [13]:

G. DESCRIPTION OF NETWORK SECURITY AND MONITORING TOOL

• TCPDump: 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 d ata 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.
• TCPReplay: 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, TCP Replay 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.

H. 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 interact to the attacker via honeypot system and all the activities will be observed through honeywall in the database. When attacker will attack or interact 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 pcap file which shows the all log file of attacker .

III. IMPLEMENTED SYSTEM ARCHITECTURE

The system architecture is composed of several systems in major server. The system architecture is shown as Figure 1.
Analyzing the malware network activities can use the low interaction honeypot (Nepenthes) and high interaction honeypot (Capture-HPC). Capture-HPC is client honeypot; they have different point usages. Using this honeypot system we can obtain the analysed information of malware activities on the network. We also create two honeypot systems in our honeypot system module, and locate them in two different IP network on the Internet. In this system we created Honeynet, there is honeywall in which there is integration of various tools of network data capturing like standard tcpdump packet capturing, walleye schema, GUI for data analysis.

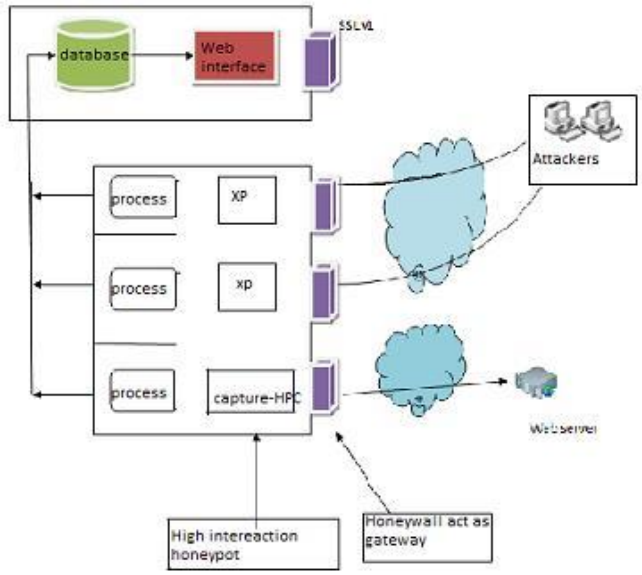

Figure 1 : Network Forensic System architecture

.
In this study , Capture-HPC and XP for the detection to avoid been aware by the attacker or threats program. These are the high -interaction honeypot system. In this there is honeywall in which there is integration of various tools of network data capturing like standard tcpdump packet capturing, walleye schema, GUI for data analysis.
To complement the information that honeypot system cannot find out, we use Capture-HPC program to acquire the other parts of malware information, because the Capture-HPC is analyzing the embedded language through the web crawler, and getting the malware activities information and systems through the client-side attack.
The information will send back to the Linux server when the honeypot system catches the malware log, but the affected malware log information will be kept in the database and we can check that information through the honeywall . This honeywall is built in the virtual box which is the integration of many tools. With the help of these tools the attackers logs can analyzed in the form of Pcap data.

IV. EXPERIMENT OF TEST SETUP AND RESULTS

A. SYSTEM FUNCTION DESCRIPTION

To login the system needs to use the SSLVPN to identify the user. After login to the system, the user has to verify the account name and the password to access the information. On the malware system web page, the main function is to search all the information collected by the honeynet. The system provided the other tool Nmap, NSLookup, Tcp dump and Traceroute. When the attackers will attack or interact with the system they will pass through the gatewall and interact with virtual honeypot. All the information of the attackers(OS, snort alert, ids alert, ip address, time , cp and udp packets) can be accessed through web interface.
We can use the wireshark tool to read the information of pcap file.

B. RESULTS

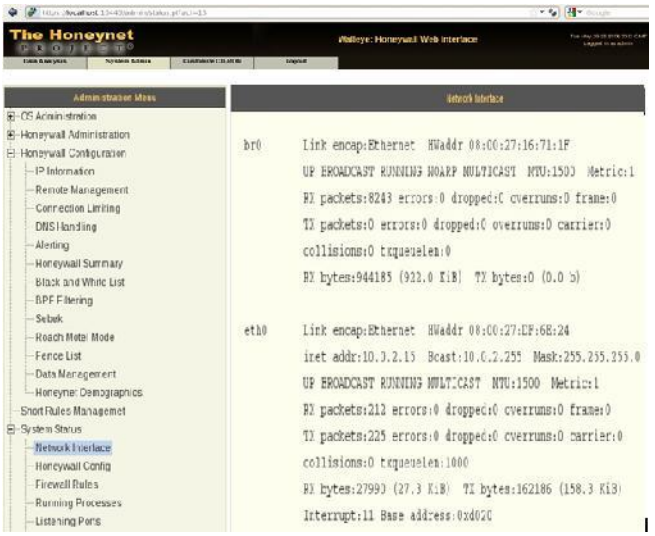

Figure 2 : Data of Network interface

In this Figure it shows the network interface information whether the bridge is created or not and which lan connection is build on the system.

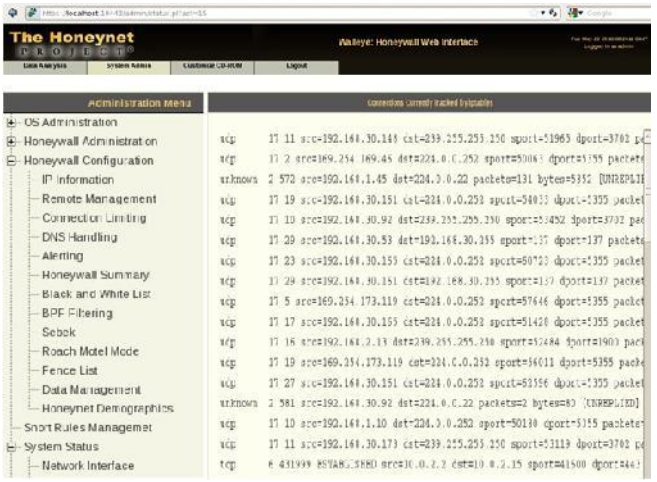

Figure 3 : Shows the result of Udp packets .

This figure shows the information of outbound Udp packets and also tells the date when these Udp packets were sent.
In this system daily we can check how many snort alerts and ids alerts are generated .

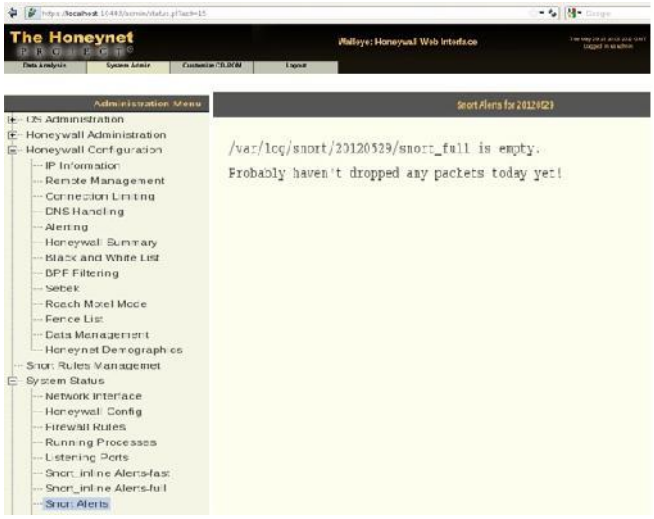

Figure 4 : Snort alert generated file

Figure 4 shows the generation of snort alert on daily. We can check these on the web interface.

V. 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.

VI. FUTURE SCOPE

In this research paper, 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.

REFERENCES

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

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

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

[4] D. Inoue, K. Yoshioka, M. Eto, Y. Hoshizawa, and K. Nakao, "Malware Behavior Analysis in Isolated Miniature Network for Revealing Malware's Network Activity," *Communications, 2008 ICC '08. IEEE International Conference*, Bejing ,pp. 1715-1721,May 2008.

[5] C.H. Yeh, and C.H. Yang, "Design and Implementation of Honeypot System Based on Open-Source Software," *IEEE International Conference on Intelligence and Security Informatics (IEEE ISI 2008)*, June 2008.

[6] B. Scottberg, W. Yurcik, and D. Doss, "Internet honeypots: Protection or entrapment?" in *Proceedings of the IEEE International Symposium on Technology and Society (ISTAS)*, 2002.

[7] L. Spitzner, "The honeynet project," http://www.honeynet.org, (Last visited: May 26, 2007)

[8] A. Yasinsac and Y. Manzano, "Honeytraps, a network forensic tool," in *Sixth Multi-Conference on Systemics, Cybernetics and Informatics*,
2002.

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

[10] "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/.

[11] Client Honyepot,
http://en.wikipedia.org/wiki/Client_honeypot

[12] C. Seifert, I. Welch, and P. Komisarczuk, "Identification of Malicious Web Pages with Static Heuristics," *Telecommunication Networks and Applications Conference*, pp. 91-96, 2008.

[13] 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.