CHAPTER ONE

INTRODUCTION

1.1 Background to the Study

Effective network security is largely dependent on understanding the internet’s current and emerging threats. To protect information systems and their users, precise, concise, high quality data n malicious operations is of vital significance (Yegneswaran, Barford, and Paxson 2005). Discovering vulnerabilities such as Heartbleed, Shellshock, and Poodle and their extensive prevalence across a number of operating systems attracts government attention to safety of the system. As noted with Heartbleed, attackers were much quicker than suppliers could generate and roll out patches in exploiting the vulnerabilities. Relying solely on traditional defense lines like Intrusion Detection Systems and dynamic firewalls alone does not provide a holistic coverage on the detection of novel and emerging patterns of assaults.

(Schneier, 2000) “Security is a process, not a product.” This popular quote is well echoed by the phenomenon that although there are a big range of safety instruments available today (whether as business or open-source alternatives), none of these instruments can address all of an organization's safety objectives on a one-handed basis. As a result, safety experts are looking for more sophisticated instruments that are efficient in recovering from safety gaps and detecting them. In order to observe the activities of a hacker, the methodology chosen is to mislead them, by giving them some emulated set of activities on a system which appears to be authentic. Their activities are then logged and monitored to acquire insight into the plan of action of the blackhat community. In honeypots, this idea is adopted- system whose value lies in being tested, attacked or compromised.

Honeypots give insights on the attacker’s actions and motivation and are able to spot zero-day attacks. The field of honeypot research consists of two main pillars: a) the development of the honeypot software and its competent deployment: b) Investigating the log information acquired in a structured way.

Honeypots have recently gained a great deal of attention from the research community because of their use in capturing and logging questionable networking practices that can be used to obtain concrete data about hackers ' conduct and operations. In addition to its use as a research instrument, it was also implemented as a research instrument in academic organizations. For instance, the Georgia Institute of Technology's Honeynet Project has been used in network security courses to educate learners how to use instruments like ethereal and tcp dump to examine the traffic of attacks. (Honeynet Project,h.n.).

1.2 Statement of the Problem

In other, for academic institutions to keep up with their counterparts especially in the area of information and communication technology, most of them have embraced the numerous benefits that using the internet and other digital media has to offer. Most of the informations they put online are critical but since these information run through a network, they are prone to various forms of vulnerability such as unwarranted access. There is a growing need for such threats to be minimized or totally eliminated. Due to network traffic, packets which are received can contain virus which can appear as signature rules and can corrupt the entire honeypot infrastructure. Since most institutions do not really give network security the priority that it deserves during budgeting, they remain prone to some of these threats and if left unchecked could result into loss of resources. Deploying only a single intrusion prevention technique such as a firewall may not always be effective. Hence, the focus of this work is to develop a honeypot for improved network security.

1.3 Aim and Objectives of the Study

The aim of this research is to develop a honeypot for improved network security. The specific objectives are to:

1. Design a framework that can be used to monitor network traffic
2. Implement the designed framework

1.4 Scope of the Study

The scope of the project covers the development of network for use at a university (MTU) to secure and provide extra safety for information and other valuable resources. The requirement include designing a honeypot on the real network thereby preventing unauthorized access to the real network and giving the network administrator enough time to study the attacker’s aim, purpose and objectives of attacking the network, with these knowledge the network administrator properly safeguard the valuable information and documents and sourcing for more methods to deny these attackers access to the real network.

1.5 Significance of the Study

Security is currently a trending issue for different network types. The main reason behind the advent of honeypot is that firewalls and access control on their own do not provide an adequate defense against attack. Aside the fact that this work will provide a way for improving network security, it will also help researchers and novel users understand the concept of honeypot.

1.6 Definition of Terms

Attacks: An attack on system safety that stems from a smart threat; that is, a smart act that is a deliberate effort (particularly in the context of a method or technique) to avoid safety facilities and violate a system's security policy.

Denial of Service (DoS) Attacks: Denial of Service is an attack whereby the systems receiving too many requests cannot return communication with the requestors. Then the system consumes resources waiting for the completion of the handshake.

Firewall: A firewall is a typical defense mechanism for border control or perimeter.

Heartbleed: In the famous OpenSSL cryptographic software library, it is a severe vulnerability. This weakness makes it possible to steal the data protected by the SSL / TLS encryption used to safeguard the Internet under ordinary circumstances.

IP Spoofing: Spoofing implies having the computer's address mirror the address of a trusted computer so that other computers can be accessed. The intruder's identity is concealed by distinct means, making it hard to detect and prevent.

Network Security: Network security relates to all hardware and software functions, features, operating processes, accountability measures, access controls, administrative and management policies needed to provide an appropriate level of hardware, software and network data protection. Packet Filtering: A firewall operates tightly with a TCP / IP protocol and works with an algorithm to divide information obtained from network apps or more obviously from protocol services (Telnet, SMTP, DNS, SMNP, NFS, etc.) into information packets.**Packet**

Filtering: A firewall operates closely with a TCP/IP protocol and works with an algorithm to split data received from applications on the network, or more clearly from services run on protocols (Telnet, SMTP, DNS, SMNP, NFS, etc.) into data packets.

POODLE attack is an exploit that exploits how some browsers handle encryption. POODLE **(Padding Oracle on Downgraded Legacy Encryption)** is the name of the exploit vulnerability.

Shellshock is component of the security bug family, also known as **Bashdoor**. The first of which was released on 24 September 2014 in the commonly used UNIX bash shell. Many Internet-facing services, such as some deployments on web servers, use Bash to process certain requests, enabling an attacker to cause vulnerable Bash versions to implement arbitrary commands. This can enable attackers to obtain unauthorized unwanted access through a network to a computer system.

Threats: A threat relates to anything that has the ability to cause severe damage to a computer system in the framework of computer security. Danger may or may not occur, but it has the ability to cause severe harm. Threats can lead to computer systems, networks and more being attacked.

CHAPTER TWO

LITERATURE REVIEW

2.0 Introduction

This review introduces a review of network security, overview of network monitoring, honeypot technology, honeypot work principle, classification of honeypots, advantages and disadvantages of honeypot and survey of related works.

Late investigations have demonstrated that the absence of security of data has made genuine harms to associations, government, and academic institutions and has become a huge concern in this age (Almutairi (2016); Michal, Eva and Zuzana, 2017). In the course of recent years, the web has advanced and individuals have been confronting difficulties of network security. This is a major issue for some associations and establishments who need to shield their useful and private information from threats inside or outside the association. Ongoing examinations have additionally demonstrated that human and association factors likewise sway on network security. Network professionals confronted difficulties to oversee security and they use extraordinary devices like firewall, antivirus, Intrusion Prevention System (IPS), Honeypots among others.

Among every one of these devices, Honeypot assumes a significant job to distinguish the malicious activities quickly and reinforce response for real time assault and attack. Be that as it may, regularly Honeypot users find it hard to utilize Honeypot and unfit to exploit every one of its functionalities. So as to improve the convenience of Honeypot a few difficulties which affect security must be comprehended. It has raised the likelihood that vindictive users increase illicit access to associations to take private data they are keen on or demolish it by infusing applications called malware. Those applications are made to enable vindictive users to control organizations' computers remotely. The past strategies which have been utilized to secure data/information have now been debilitated, which has prompted interest for better techniques to improve and prevent access to unapproved data from attackers. In this way, a Honeypot is required for the detection of all undesirable and suspicious traffic that can't be recognized by devices that have been conveyed commonly, for example, firewall.

2.1 Conceptual Review

2.1.1 A Brief Review of Network Security

Network security refers to any exercises intended to ensure your network. It comprises of the innovations and procedures that are conveyed to shield networks from interior and outer dangers. Network security includes all exercises that associations, ventures, and establishments embrace to ensure the esteem and continuous ease of use of benefits and the uprightness and coherence of tasks. Effective network security focuses on an assortment of dangers and prevents them from entering or spreading on your network. (Amanpreet and Monika, 2014).

Network assaults have been found to be as changed as the framework that they endeavor to enter. Assaults are known to either be purposeful or accidental and actually skillful interlopers have been keen on focusing on the conventions utilized for secure correspondence between networks administration gadgets. (Suliamon, 2014). The least demanding ways computers have been shielded and secure from assaults, for example installing virus protection, utilizing solid passwords, utilizing a firewall to upgrade settings, and programming has demonstrated to be inadequate and not fit for keeping unauthorized users from gaining access to data.

The essential objective of network security is to give controls at all points along the network border which enable access to the network and possibly given traffic a chance to pass if that is approved, substantial and of worthy risk. The reason for network security is to protect networks, network gadgets and network messages from unapproved access, usually by outsiders.

2.1.1 Overview of Network Monitoring

Network monitoring is the function of network management to collect data. Applications for network surveillance are developed to retrieve information for applications for network management. The purpose of monitoring the network is to collect useful information from different parts of the network so that the collected information can be used to manage and control the network. Most devices on the network are placed in distant places. When a network failure happens, surveillance officers are required to identify, isolate, and correct network malfunctions and potentially retrieve the failure. Commonly, administrators should be warned by agents to fix the issues within a minute. With the stable network, the tasks of the administrator continue to be constantly monitored when there is a danger either from within or outside the network. In addition, if the network devices are overloaded, they must inspect the network performance frequently. In order to create a network plan for short-term and long-term future improvements, data on network utilization can be used before a failure due to overload. These devices usually do not have terminals that are directly connected so that the application for network management can not readily track their status. Network monitoring methods are thus created to allow apps for network management to verify the status of their network devices. As network devices are increasingly being used to construct larger networks, network monitoring methods are being extended to monitor networks as a whole.

(Edmund, 2000) outlined three major basic goals for network monitoring: Performance monitoring, account monitoring and Fault monitoring.

Performance tracking deals with network performance measurement. Performance surveillance has three significant problems. First, data on performance surveillance is generally used to schedule future network development and to identify current network utilization issues. Second, the performance surveillance timeframe must be sufficiently long to create a model of network conduct. Third, it's essential to choose what to evaluate. In a network, there are too many measurable items. Fault monitoring deals with the measurement of network issues. Account monitoring is about how the network is used by users. Network administrators are constantly striving to keep their networks running smoothly. If a network were to be down even for a short time, productivity would decline within a company, and the ability to provide essential services would be compromised in the case of public service departments. Administrators need to monitor traffic movement and efficiency across the network to be proactive rather than reactive and check that safety breaches do not happen within the network.

Active network monitoring adds option to modify the data on the line. Passive network monitoring exists in several forms. Simple surveillance can be simple as the quantity of information monitored and generated is low for manual evaluation. Monitoring all kinds of details about the network and its traffic carries a comparable obstacle; data is collected about faults and attackers, but there is so much data that it gets lost in the ocean. Figure 2.1 shows the general architecture of network monitoring.

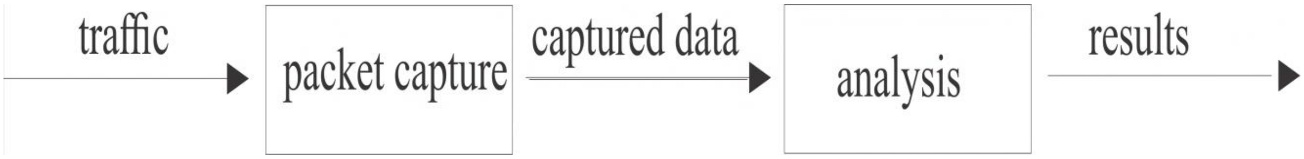


Fig 2.1: Architecture of network monitoring (Edmund, 2000)

2.2 Theoretical Review

2.2.1 Honeypot Technology

A Honeypot is a decoy, positioned out on a network to attract attackers. Honeypots are designed as the emulation of the real machines, creating the advent of running full activities and programs, with open ports that might be found on a normal system or server on a network. This manner honeypot mimics the actual gadget, create confusion for attackers and monitor the intruder without danger to manufacturing servers or records. Honeypot era isn't always to replace the traditional security mechanisms and defense technology, however, it's helping and complementary. Honeypot generations proactively discover and reply to network intrusion and assaults. (Bao, Jian, Chang, and Mo, 2010).

A honeypot device can locate attack behavior and redirect such assaults to a strictly controlled surrounding to protect the practical working systems. (Koch, Robert, Mario, and Gabi, 2013).

This gadget collects intrusion records to look at and report the conduct of the attacker. It also examines the level, equipment, reason, and intrusion methods of the attack such that proof can be received and possible criminal moves can be taken. Cautiously set by using the Honeypot system to draw hackers, and hackers to tune, the intruder may be observed record system (Bao et al 2010). Honeypot can be a computer simulation of a regarded hollow or a carrier computer, also can simulate an expansion of running device and its corresponding features, or just a regular general running system, and simplest through unique processing can be a complete file of the attacker's attack.

2.2.2 Honeypot Work Principle

A honeypot works by means of fooling attackers into believing it is a legitimate device. So attackers assault the system without understanding that they're being discovered completely. Honeypot seems like a sincerely host provided critical provider, so it has more enchantment to the hacker. Through its enchantment to hackers and being attacked, the related records of the attackers consisting of the IP address, motives of the attackers getting into the machine and assault conduct of the attacker might be collected. That's accomplished typically through the implementation of the heritage software program. (Li, Zhang, 2009) Which video display units and statistics the network communication information among the attackers and honeypot host, and uses some analytical tools to interpret and analyze these facts. Facts capture is a difficult section to any honeypot that has the capability to seize everything the attacker is doing. it is able to additionally capture the packets and packet payloads worried inside the attack. This records can show crucial in analyzing the attackers' activities.

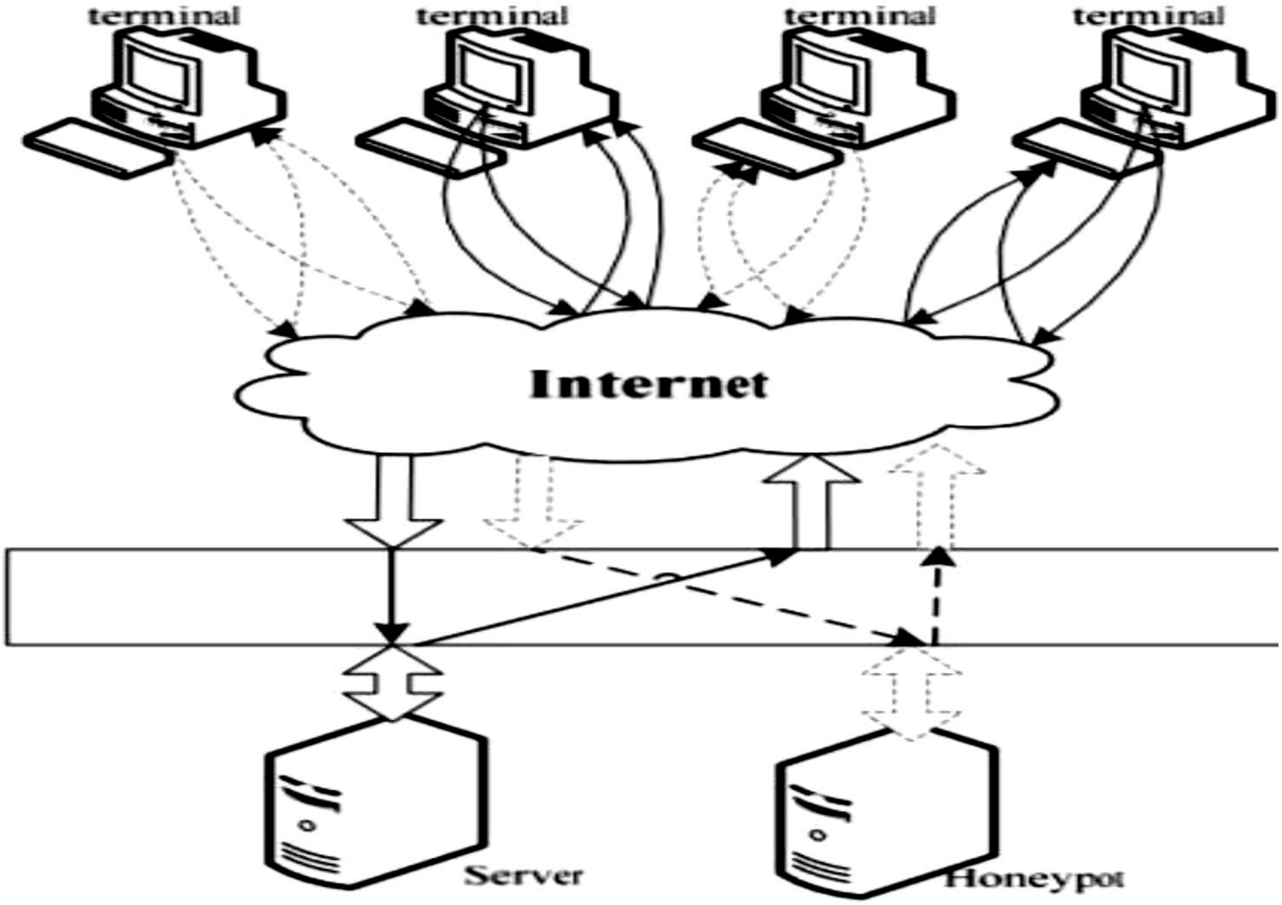


Figure 2.2 Honeypot Work principle

Honeypot system comprises of normally three modules which are induced, deceive and analysis. The induced module is used to draw attackers to attack the Honeypot system. The deceived module calls the simulation facts from the database for the deceived host to generate false data on the way to be sent to the attackers (Li, Zhang, 2009). All the induction and deception occasions of the device are recorded within the faraway (remote) log server and analyzed by way of the evaluation module for adjusting the induction and deception strategy.

2.2.3 CLASSIFICATION OF HONEYPOT

Consistent with the layout, Deployment of honeypot can be categorized into production and research honeypots.

Production Honeypot

A production honeypot is one used within a business enterprise's surroundings to shield the corporation and assist mitigate danger (Karthik, Samudrala, and Yang, 2004). Production honeypots emulate the production network of the enterprise. Attackers interact with them in order to show the vulnerabilities of the production network. Uncovering these weaknesses and alerting administrators of assaults can provide early caution of attacks and help reduce the chances of intrusion (Gubbels, Kecia, 2002). It is located miles within the production network with different production servers like a firewall to improve their security. Production honeypot enables to reduce the dangers of intrusion and upload values to the security measures of an employer. Production honeypot calls for less functionality than a research honeypot. They're easier to construct and set up. Even though they discover assault styles, they do not deliver a great deal of data about the attackers than research honeypots. You could learn from which system attackers are coming from and what exploits are being released, but maybe not who they may be, how they may be prepared, or what gear they may be the usage of (Mokube, Iyatiti, and Michele Adams, 2007).

Research Honeypot

Research honeypots are real running systems and services that attackers can have interaction with. Generally it is designed to get understanding about the blackhat community. They contain higher risk, accumulatesizeable statistics and Genius on new attack strategies and methods. So it affords a greater accurate image of the types of attacks. It is used to look up the threats companies face and helps to furnish better safety towards these threats. Research Honeypot is extra complex to deploy and maintain. They are used chiefly by means of research, military, or government organizations.

Research honeypots add extremely good cost to research by means of presenting a platform to learn about cyber threats. Attackers can be watched in motion and recorded step via step as they attack and compromise the system. This intelligence gathering is one of the most unique and thrilling characteristics of honeypots. (Spitzner, Lance, 2001) According to the Honeypot with Different Attacker Interaction Level, we might also divide honeypots into three most important classes: low-interaction, medium interaction, and high-interaction.

Low- interaction Honeypot

Low-interaction honeypot systems no longer supply intruders with the real operating device for remote login (Koch, Robert, Mario Golling, and Gabi Dreo, 2013).

They are used for simulating the specific characteristic or service which is running in the present system, attackers can solely have motion in this controlled range. A low-interaction honeypot presents unique analog offerings that can be carried out by way of monitoring a unique port (R. Berthieret. al, 2008). Low interaction honeypots emulate network services on preconfigured port, such as FTP, SQL, Web, SSH, etc. Example: Honeyd, Specter.

Medium–interaction Honeypots

Medium interaction honeypot supply the attacker with a higher illusion of a working system because there is more for the attacker to have interaction with.

More complex assaults can consequently be logged and analyzed (Mokube, Iyatiti, and Michele Adams, 2007). They can capture extra information, and have improved concealment than low interactive honeypots. They more correctly have interaction with intruder than do low-interaction honeypots but much less functionality than high-interaction honeypots.

This kind of honeypot system emulates a precise carrier which causes intruders to assume that they are attacking the real operating system**.**

It enables the system to accumulate excessive amounts of information however will increase the threat of intrusion. Example: mwcollect, nepenthes and honeytrap.

High-interaction Honeypots

High interactive honeypots are configured with actual operating system and grant an actual operating system for attackers. They are a complicated solution and contain the deployment of actual operating systems and applications (Chawda, Kartik, and Ankit, 2014). High interactive honeypot permits attackers running all the instructions in the actual operating system. So there are excessive probabilities for collecting massive amounts of information, as all actions can be logged and analyzed. Any error in the system may additionally permit a hacker to manipulate the full operating system, assault other systems, or intercept messages in the application system (E. Cooke et al, 2004). High-interactive honeypots are more beneficial to seize the important points of vulnerabilities or exploits that are unknown to the outside world. This honeypots are quality in the case of Zero Day attacks. Examples: Honeynets Sebek.

2.2.4 Advantages and Disadvantages of Honeypot

Advantages

1. Honeypot creates confusion for attackers by giving them bogus data.
2. It can provide forensic evidence in a court of law that is admissible. As long as it is deployed correctly and is not advertised, it can be used as legal evidence.
3. You can use honeypots to intrude assaults. Knowing that a system is set up to capture and log all activities may scare away would be intruders.
4. The properly designed and configured Honey Pot will collect data such as the IP address, the attackers ' motives entering the attacker's system and attack behavior.
5. Honeypots distract intruders from the scheme of manufacturing, making them harmlessly use all their attempts.
6. Honeypots don't cost much. Some basic versions can be downloaded free of charge.
7. Honeypots can detect insider attacks by offering precious insider pattern data.

Disadvantages

1. Honeypots can only monitor interacting activity with it. They've got a tight viewing field. They see only what activity is aimed at them.
2. Honeypots are also at danger as attackers may misuse honeypot to damage other systems. (Spitzner, Lance, 2003)
3. Fingerprinting is another disadvantage of honeypots. Fingerprinting is when an attacker can identify a honeypot's true identity because it has certain features or behaviors that are expected.(Spitzner, Lance, 2003)
4. Another disadvantage is that honeypots must be maintained like any other networking equipment and services.
5. Building a honeypot needs you to have at least an entire system devoted to it, and for some corporations this may be a costly resource.

2.3 Survey of related works

Honeypots play a great role in the area of network security. Honeypots have developed in a variety of ways to deal with numerous fresh safety threats on the Internet today not only against safety defenders but also novice users. New types of honeypots are being introduced to cope with the recent changes in network security, they are acting against the new vulnerable activities.

Portokalidis et al (2006) proposed a honeypot called “Argos”. It automates the surveillance, detection and generation of signatures for intrusion detection of fresh unidentified malware. It is intended to slow the spread of new, and therefore unknown, malware such as worms, viruses, and bug designs. When Argos detects vulnerable data, assembly codes, called "shellcode," are also dynamically inserted into the process to extract detailed information about the process so that the process is slowed down or trapped in an infinite loop to minimize its harm.

Alosefer and Rana (2010) proposed “Honeyware”. To detect malicious web servers, it is a low-client-side honeypot interaction.Alosefer tested Honeyware in advance against 94 URLs in which 84 were malicious and 10 were benign. Honeyware identified 83 URLs that were malicious. Since Honeyware is a low honeypot interaction, the data it collects must be processed by a time-consuming external processing engine.

Adachi and Oyama (2009) proposed “BitSaucer”. It is a hybrid honeypot, i.e. providing the facility for both low honeypot interaction to attain fewer resource demands and elevated honeypot interaction to emulate complete answers.

Zhuge et al (2007) proposed a new honeypot called "HoneyBow" to automatically detect and capture malware, such as viruses and worms, without needing manually researching honeypots output information from human security professionals. HoneyBow detects file changes by comparing their original MD5 hash after deliberately allowing malware to change their files. The process that produced the modification is recorded as malware when any modification is identified and its MmFetcher element restores the original copy of the documents. Another element, MmWatcher, monitors system calls that create and modify files, triggering detection of intrusion. Finally, to detect suspect operations of malware, MmHunter monitors code being performed as a debugger.

Anagnostakis et al (2005) proposed “Shadow Honeypots”. They are true apps for the manufacturing network, but they contain integrated honeypot codes. They are centered in high-interaction honeypot on the trade-off issue as false positive and false negative.All incoming requests to a server running the honeypot shadow will be performed as if they were performed by a server operating the manufacturing. If a request is determined to be innocent by the shadow honeypot, it will forward the application to the manufacturing server.

LaBreais another type of honeypot intended to slow or prevent assaults by acting as a sticky honeypot for detecting and trapping worms and other malicious codes. It can run on both Windows and UNIX.

Vinu V. Das (2009) proposed a solution to mitigate denial of service attacks by hiding production servers behind an access gateway, called “Active Server (AS)”. Each AS authenticates its customers and a path is launched between the client and a server once a client is authenticated. If an AS does not authenticate a client, the client will be trapped there as a honeypot. The client can be authenticated by any AS if a client has access to various ASes. Honeypots trap attackers, preventing, reducing and delaying the DoS attack effects.

NielsProvos (2004) suggested an open source honeypot with low contact called "Honeyd." It is a strong honeypot and can be operated on systems such as UNIX and Windows. It can monitor unused IPs, simulate TCP / IP stack-level operating systems, simulate thousands of virtual hosts simultaneously, and monitor all ports depending on UDP and TCP.

Nazario (2009) suggested "PhoneyC" a fresh form of honeypot. It expands in two directions current honeypots. The first is to activate honeypots, meaning honeypots on the client side. The second is the dynamic parser of web content to interpret binary dynamic content, particularly client-side scripts such as JavaScript, VB Script, and even Active-X controls. By integrating the two extensions to web applications, active honeypots on the client side become web "clawers" that visit a large number of web servers to detect malicious web servers automatically. As a consequence, many malicious script / control operations were detected during tests by PhoneyC.

Rowe et al. (2007) suggested the "Fake Honeypot" concept. The objective of fake honeypot is to repel attackers by deliberately revealing themselves to attackers from a manufacturing network. It looks like a true honeypot, but it does not execute any true characteristic typical honeypots. A mathematical model was implemented to maximize the impact of fake honeypots, using certain parameters such as the likelihood of a scheme being a honeypot, the advantage anticipated by an intruder from compromising a host of manufacturing, and the price of compromising a host.

CHAPTER THREE

METHODOLOGY

3.0 Introduction

This chapter talks about the methods used to carry out the honeypot system. It also shows the tools used and design methods used to implement the design, this methods will also be used for data collection and data analysis. With the improvement to information and technology and security, the process by which hackers attacks servers and systems has changed drastically.

The honeypot system, which has been developed to monitor threat prone ports on a server or system, works on the premise that when any unwanted access is detected in the Mountain Top University network, the system sends the data to the server, and the server will check if it is a malicious data or not using the packet sniffer. If the data sent is malicious, such packet will be discarded and if not the data will be sent to the destination system.

This model was introduced using an open source software called Pentbox a Security Suite that packages network and system-oriented safety and stability testing instruments. Nmap, brief for Network Mapper, a free open-source tool for scanning vulnerability and discovering network. Wireshark is a free and open source protocol analyzer that allows users to browse information traffic on a computer network interactively.

3.1 Flow Chart of Honeypot System

The study will start by designing a framework of a Honeypot which would use an open source software called pentbox. The framework explains the flow of packets in and out of a protected network. Figure 3.1 shows the framework of honeypot’s activities in an institution:

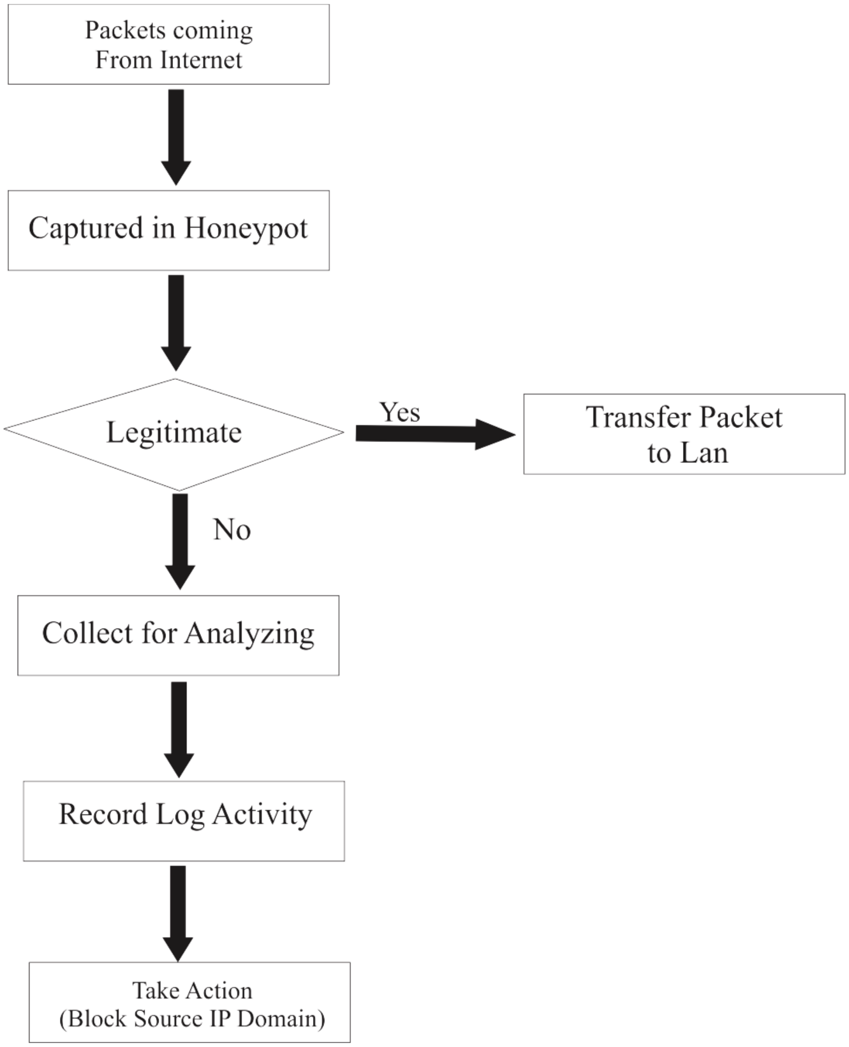


Fig 3.1 flow chart of honeypot technology

Honeypot actually cannot prevent cyber-attacks against the network but helps in identifying and detecting them when used with other defense oriented tools such as Firewall and Intrusion detection system (IDS).

3.1.1 Firewall

Firewall defines a single chock point that keeps unauthorized users out of the protected network, prohibits the entry or exit of potentially vulnerable services, and provides protection against various types of IP spoofing and routing attacks. Single choke point simplifies safety management by consolidating safety capacities on a single system or system set. The firewall is immune to penetration itself. This means using a trusted system with a secure operating system. Basically, the number of firewalls can be deployed for integrated, cooperative, and deep network security protection in the proper positions of the managed network. It is notice that it does not protect against internal threads or against the transfer of virus infected programs or files.

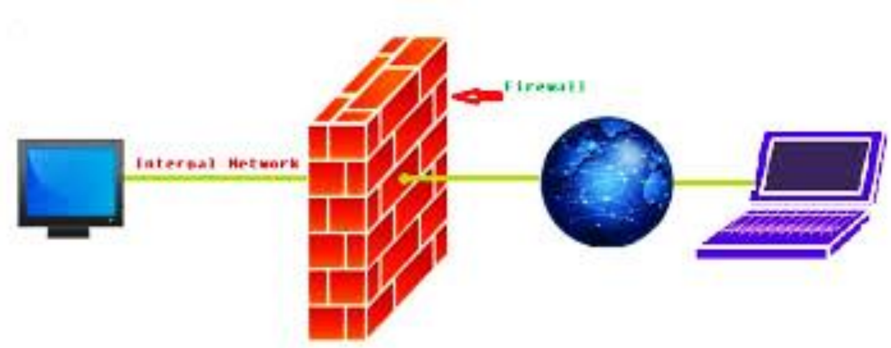


Fig 3.1.1 firewall

3.1.2 Intrusion Detection System

An IDS inspects all inbound and outbound network activity and identifies suspect patterns that may suggest a network or system assault by someone trying to break into or compromise a system. Individuals and companies have used intrusion detection in a number of ways throughout the year, including erecting ways and fences around valuable resources with sentry boxes to monitor activities around the resource's premises. IDS are easier to deploy as it does not affect existing systems or infrastructure but it is not a solution to all security concerns because it meet with the problem of false positive and false negative.

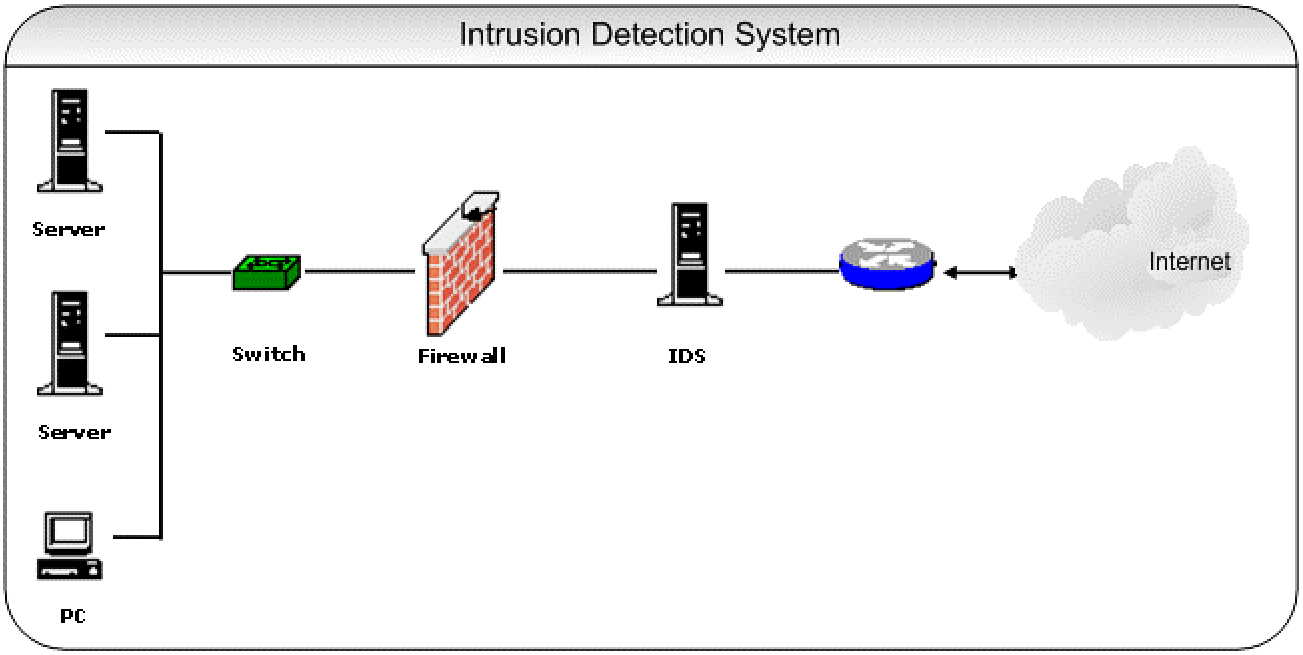


Fig 3.1.2 Intrusion Detection System

3.2 Kali Linux

Kali Linux is a sophisticated penetration testing and security auditing platform based on Debian. Kali includes several hundred instruments for multiple information security functions, such as Penetration Testing, Security Research, Computer Forensics, and Reverse Engineering. Offensive Security, a major information security training firm, develops, finances and maintains Kali Linux.

On March 13, 2013, Kali Linux was published as a full, top-to-bottom BackTrack Linux reconstruction that fully adheres to Debian development norms.

This open-source software contains some of the tools that will be used for this work which include

1. Wireshark
2. Pentbox
3. Nmap/Zenmap
4. DIGT

3.3 PENTBOX 1.8 FRAMEWORK

Pentbox is a framework that consists of security and stability testing oriented tools that are commonly used in networking. It is developed in ruby and oriented to GNU/Linux systems, but is compatible with every systems where Ruby works. Tools in pentbox 1.8

I .Cryptography tools

- Base64 Encoder & Decoder
- Multi-Digest (MD5, SHA1, SHA256, SHA384, SHA512, RIPEMD-160)
- Hash Password Cracker (MD5, SHA1, SHA256, SHA384, SHA512, RIPEMD-160)
- Secure Password Generator

II .Network tools

- Net DoS Tester
- TCP port scanner
- Honeypot
- Fuzzer
- DNS and host gathering
- MAC address geolocation (samy.pl)

III .Web

• HTTP directory brute force

• HTTP common files brute force

3.4 Method for Data Collection

The Pentbox, which has a module for data capture, deals with the collection and recording of the Honeypot operations. It deceives the intruder by detecting all exercise within the honeypot and the data entering and leaving the honeypot, without being known to the attackers. Attacker does not know that in a fake system he is working. Collected information is stored in the database. Captures particular quantities of information through traffic reduction. Captures the attacker's activity on honeypot itself. Captures the attacker's activity even if it is in encrypted form.

3.5 Procedure for Data Analysis

Wireshark is a tool (also known as a network sniffer) for network or protocol and data analysis. It is used to analyze the structure of various network protocols and is capable of demonstrating encapsulation. The analyzer operates on Unix, Linux and Microsoft Windows operating systems, and employs the GTK+ widget toolkit and pcap for packet capturing.

Wireshark shares many characteristics with tcpdump. The difference is that it supports a graphical user interface (GUI) and has information filtering features. In addition, Wireshark permits the user to see all the traffic being passed over the network

Features of Wireshark include:

- Data is evaluated either from the network link wire or from information files that already have data packets captured.
- Supports live information reading and analysis for a broad spectrum of networks (including Ethernet, IEEE 802.11, PPP and loopback).
- Users can browse captured data networks with the assistance of GUI or other variants.
- Users can use command line switches to edit and convert the captured documents to the editcap application.
- The filter screen is used to filter and organize the display of information.
- The creation of plug-ins can scrutinize new protocols.
- Captured traffic is also capable of tracking Internet Voice (VoIP) calls.
- Rough USB traffic can also be captured when using Linux.

CHAPTER FOUR

IMPLEMENTATION AND RESULT

This chapter shows the method that have been used for the implementation and showing the result of the objectives mentioned. This chapter presents an overview of the Honeypot System, which was used to ensure the detection, prevention of the confidentiality of the information against unwanted access and intrusion in the network

4.1 System requirements

The system requirement is divided into two parts the Software and the hardware.

Software Requirements

Software Requirements As the software on the market is witnessing geometric progression, a major element in running a system is the compatible software part. The company and one user should accept selected software as well as be liable for the scheme.

Operating System: Kali Linux

Hardware Requirements

Hardware Requirements Hardware configuration section is a significant software development job that inadequate random access memory can adversely influence the velocity and effectiveness of the whole system. To manage all the activities, the method should be strong. The hard disk should have sufficient capacity to store the file and application.

Processor:	Core i3, Core i5, Core i7 and above
Processor speed:	2.9GHz onwards
RAM:	8GB (Minimum)
Hard disk:	500GB
Monitor Display:	LED
Mouse:	Touchpad with multi-touch gesture support, USB or PS/2

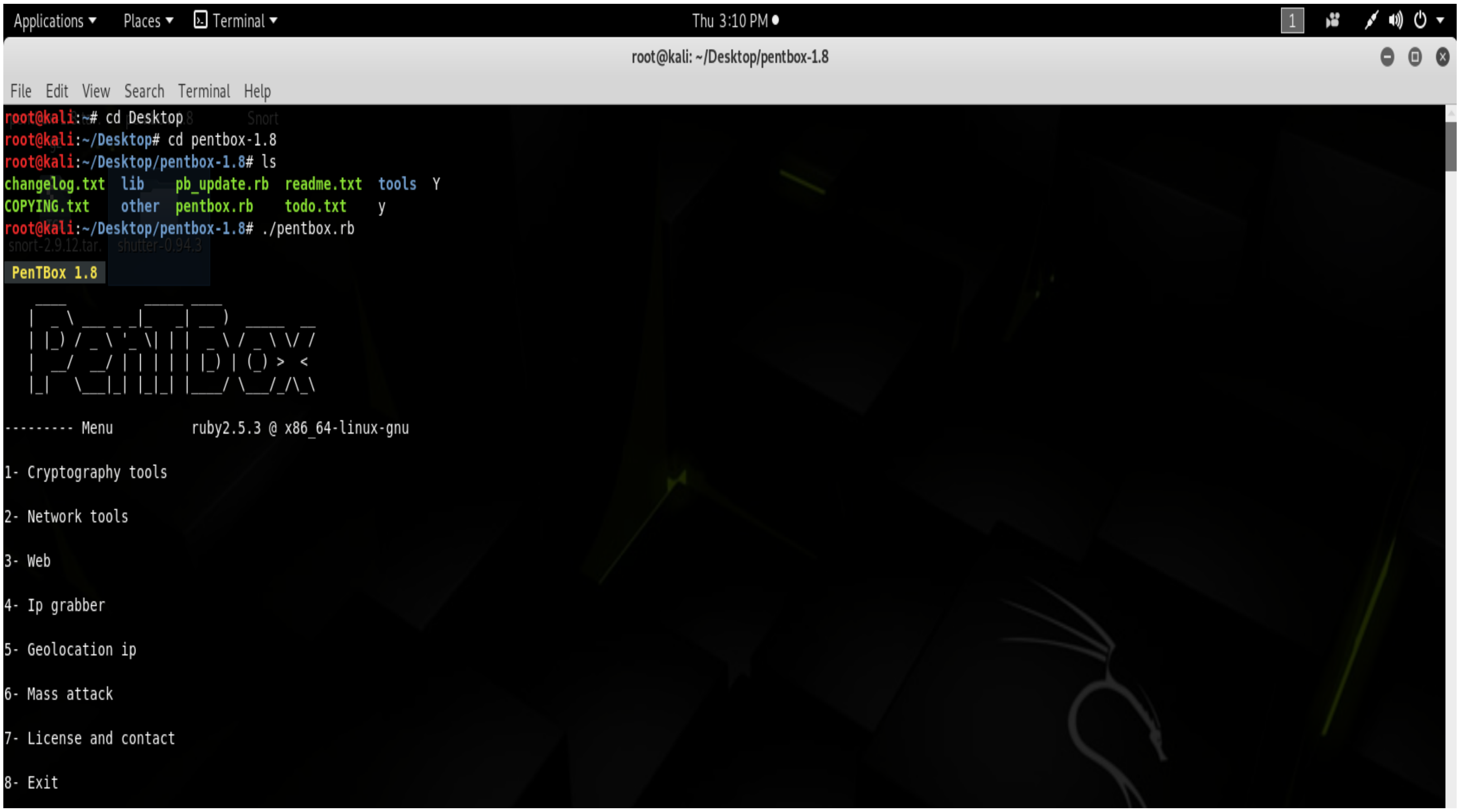
4.2 Configuration of Honeypot (Pentbox).

We have used Kali Linux operating system for setting up the server. After firing up the terminal we start the pentbox 1.8 framework .Pentbox honeypot will only work if the terminal is given administrative privileges. First go to the pentbox directory and the path of its ruby module.

Step1. cd pentbox-1.8

Step2. ls

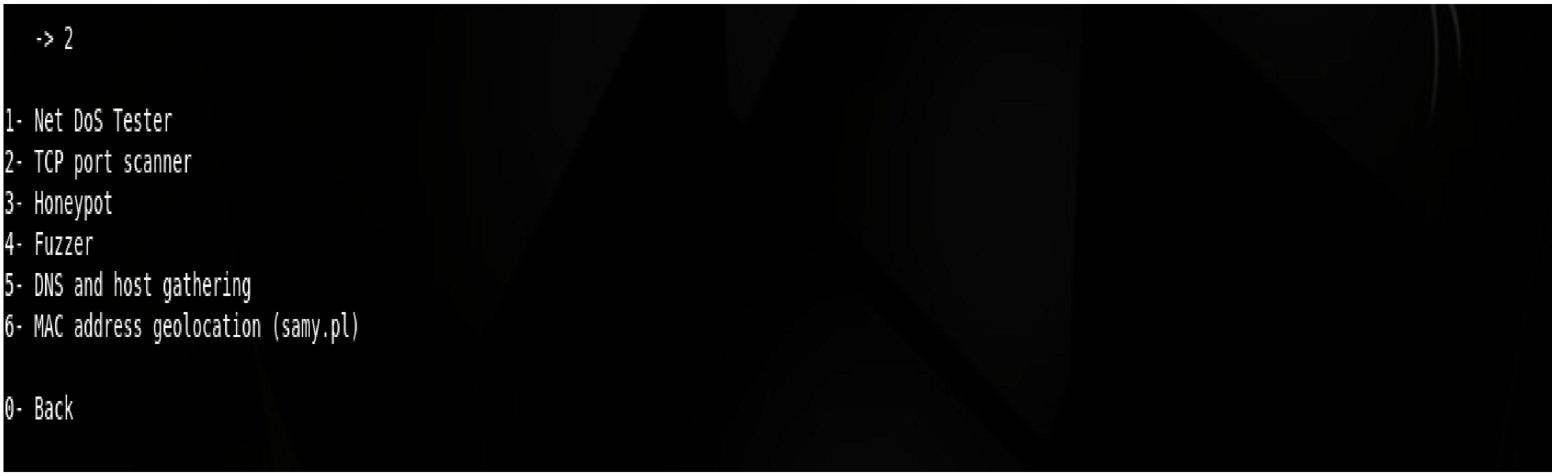
Step3. Now type ./pentbox.rb



```
root@kali:~# cd Desktop
root@kali:~/Desktop# cd pentbox-1.8
root@kali:~/Desktop/pentbox-1.8# ls
changelog.txt  lib    pb_update.rb  readme.txt  tools  Y
COPYING.txt    other  pentbox.rb    todo.txt     y
root@kali:~/Desktop/pentbox-1.8# ./pentbox.rb

 PenTBox 1.8

  ____            _____ ____
 |  _ \ ___ _ __ |_   _| __ )  _____  __
 | |_) / _ \ '_ \  | | |  _ \ / _ \ \/ /
 |  __/  __/ | | | | | | |_) | (_) >  <
 |_|   \___|_| |_| |_| |____/ \___/_/\_\

--------- Menu          ruby2.5.3 @ x86_64-linux-gnu

1- Cryptography tools

2- Network tools

3- Web

4- Ip grabber

5- Geolocation ip

6- Mass attack

7- License and contact

8- Exit
```

Fig 4.1 opening pentbox directory

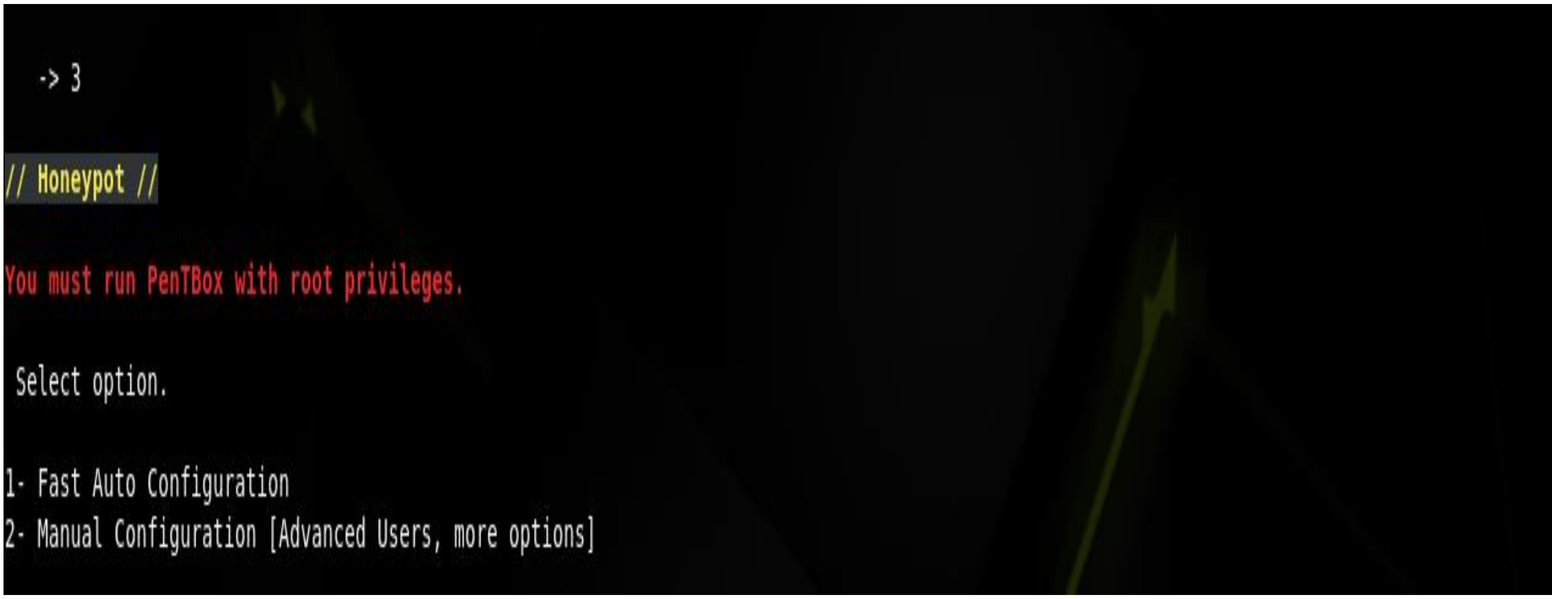
Step4. Select option 2



```
  -> 2

1- Net DoS Tester
2- TCP port scanner
3- Honeypot
4- Fuzzer
5- DNS and host gathering
6- MAC address geolocation (samy.pl)

0- Back
```

Fig 4.2 pentbox tools

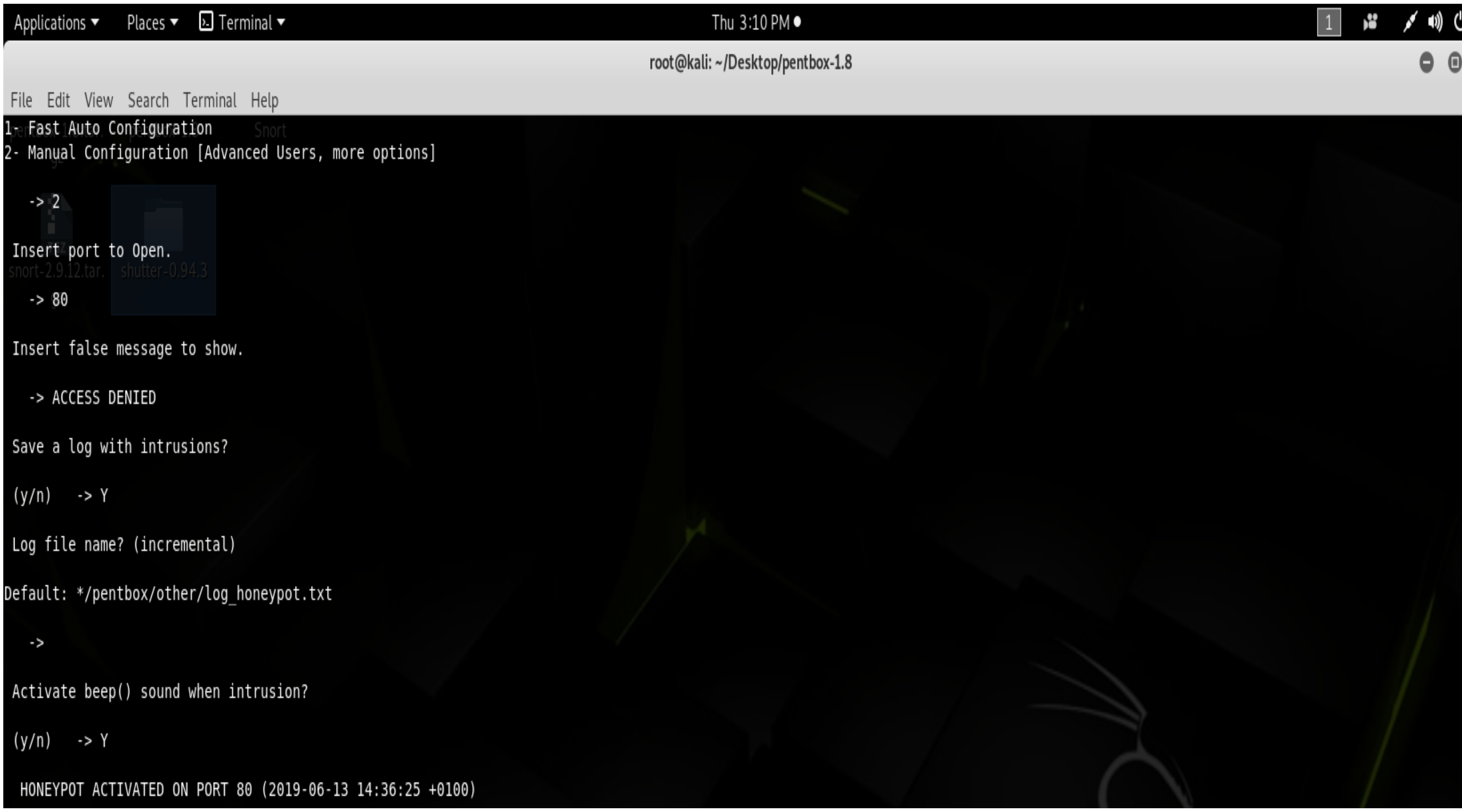
Step5. Then select option 3 Honeypot



```
  -> 3

// Honeypot //

You must run PenTBox with root privileges.

 Select option.

1- Fast Auto Configuration
2- Manual Configuration [Advanced Users, more options]
```

Fig 4.3 configuring the honeypot

Step6. Then select option 2 Manual configuration



```
1- Fast Auto Configuration
2- Manual Configuration [Advanced Users, more options]

   -> 2

 Insert port to Open.

   -> 80

 Insert false message to show.

   -> ACCESS DENIED

 Save a log with intrusions?

 (y/n)   -> Y

 Log file name? (incremental)

Default: */pentbox/other/log_honeypot.txt

   ->

 Activate beep() sound when intrusion?

 (y/n)   -> Y

  HONEYPOT ACTIVATED ON PORT 80 (2019-06-13 14:36:25 +0100)
```

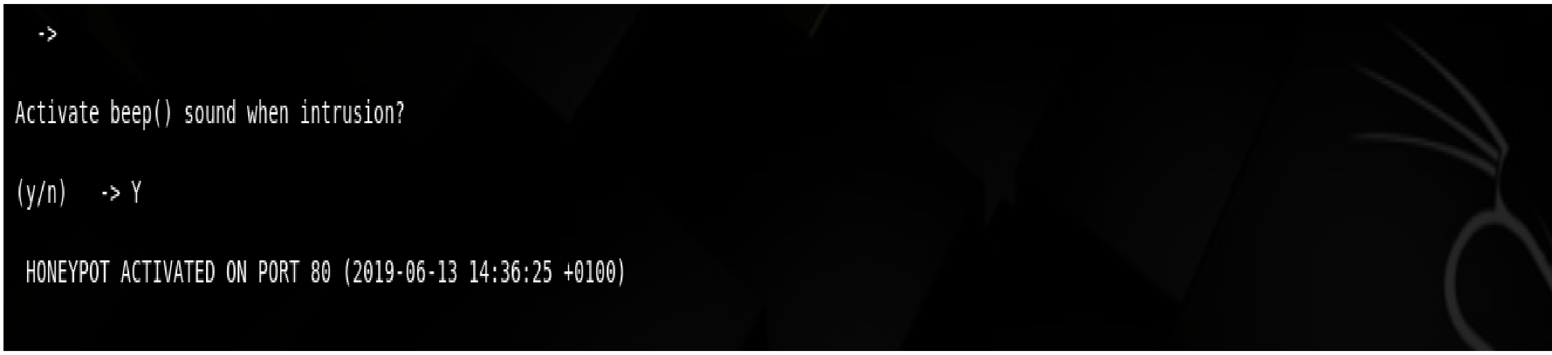
Fig 4.4 Configuring the Honeypot

Step 7. Insert port to open

Step 8. Insert false message to show

Step 9. Save a log file

Step 10. Activate intrusion sound (optional)



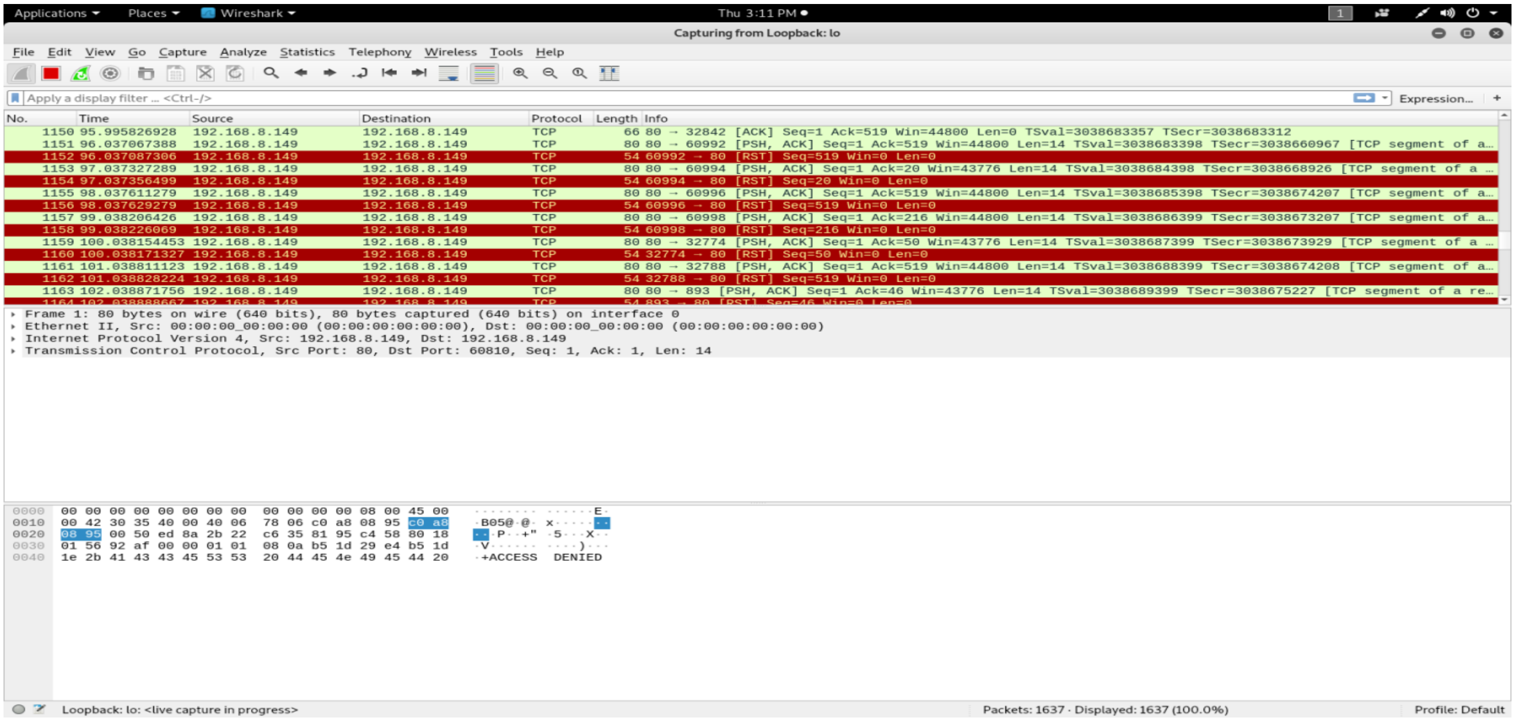
```
   ->

 Activate beep() sound when intrusion?

 (y/n)   -> Y

  HONEYPOT ACTIVATED ON PORT 80 (2019-06-13 14:36:25 +0100)
```

Fig 4.5 Honeypot activated

Port 80 is the port number assigned to Hypertext Transfer Protocol (HTTP), a widely used Internet communication protocol. It is the port from which a computer sends and gets communication and emails from a Web server based on a Web client and is used to send and receive HTML pages or information.

4.3 Configuring Wireshark

Start Wireshark and then click on the network interface you want to use to capture the data. On a wired network, it will likely be eth0. On the honeypot we use Wireshark to monitor the network traffic.



```
No.    Time            Source          Destination     Protocol Length Info
  1150 95.995826928    192.168.8.149   192.168.8.149   TCP      66 80 → 32842 [ACK] Seq=1 Ack=519 Win=44800 Len=0 TSval=3038683357 TSecr=3038683312
  1151 96.037067388    192.168.8.149   192.168.8.149   TCP      80 80 → 60992 [PSH, ACK] Seq=1 Ack=519 Win=44800 Len=14 TSval=3038683398 TSecr=3038660967 [TCP segment of a…
  1152 96.037087306    192.168.8.149   192.168.8.149   TCP      54 60992 → 80 [RST] Seq=519 Win=0 Len=0
  1153 97.037327289    192.168.8.149   192.168.8.149   TCP      80 80 → 60994 [PSH, ACK] Seq=1 Ack=20 Win=43776 Len=14 TSval=3038684398 TSecr=3038668926 [TCP segment of a …
  1154 97.037356499    192.168.8.149   192.168.8.149   TCP      54 60994 → 80 [RST] Seq=20 Win=0 Len=0
  1155 98.037611279    192.168.8.149   192.168.8.149   TCP      80 80 → 60996 [PSH, ACK] Seq=1 Ack=519 Win=44800 Len=14 TSval=3038685398 TSecr=3038674207 [TCP segment of a…
  1156 98.037629279    192.168.8.149   192.168.8.149   TCP      54 60996 → 80 [RST] Seq=519 Win=0 Len=0
  1157 99.038206426    192.168.8.149   192.168.8.149   TCP      80 80 → 60998 [PSH, ACK] Seq=1 Ack=216 Win=44800 Len=14 TSval=3038686399 TSecr=3038673207 [TCP segment of a…
  1158 99.038226069    192.168.8.149   192.168.8.149   TCP      54 60998 → 80 [RST] Seq=216 Win=0 Len=0
  1159 100.038154453   192.168.8.149   192.168.8.149   TCP      80 80 → 32774 [PSH, ACK] Seq=1 Ack=50 Win=43776 Len=14 TSval=3038687399 TSecr=3038673929 [TCP segment of a …
  1160 100.038171327   192.168.8.149   192.168.8.149   TCP      54 32774 → 80 [RST] Seq=50 Win=0 Len=0
  1161 101.038811123   192.168.8.149   192.168.8.149   TCP      80 80 → 32788 [PSH, ACK] Seq=1 Ack=519 Win=44800 Len=14 TSval=3038688399 TSecr=3038674208 [TCP segment of a…
  1162 101.038828224   192.168.8.149   192.168.8.149   TCP      54 32788 → 80 [RST] Seq=519 Win=0 Len=0
  1163 102.038871756   192.168.8.149   192.168.8.149   TCP      80 80 → 893 [PSH, ACK] Seq=1 Ack=46 Win=43776 Len=14 TSval=3038689399 TSecr=3038675227 [TCP segment of a re…
  1164 102.038888667   192.168.8.149   192.168.8.149   TCP      54 893 → 80 [RST] Seq=46 Win=0 Len=0
```

```
▸ Frame 1: 80 bytes on wire (640 bits), 80 bytes captured (640 bits) on interface 0
▸ Ethernet II, Src: 00:00:00_00:00:00 (00:00:00:00:00:00), Dst: 00:00:00_00:00:00 (00:00:00:00:00:00)
▸ Internet Protocol Version 4, Src: 192.168.8.149, Dst: 192.168.8.149
▸ Transmission Control Protocol, Src Port: 80, Dst Port: 60810, Seq: 1, Ack: 1, Len: 14
```

```
0000  00 00 00 00 00 00 00 00  00 00 00 00 08 00 45 00   ········ ······E·
0010  00 42 30 35 40 00 40 06  78 06 c0 a8 08 95 c0 a8   ·B05@·@· x·······
0020  08 95 00 50 ed 8a 2b 22  c6 35 81 95 c4 58 80 18   ···P··+" ·5···X··
0030  01 56 92 af 00 00 01 01  08 0a b5 1d 29 e4 b5 1d   ·V······ ····)···
0040  1e 2b 41 43 43 45 53 53  20 44 45 4e 49 45 44 20   ·+ACCESS  DENIED
```

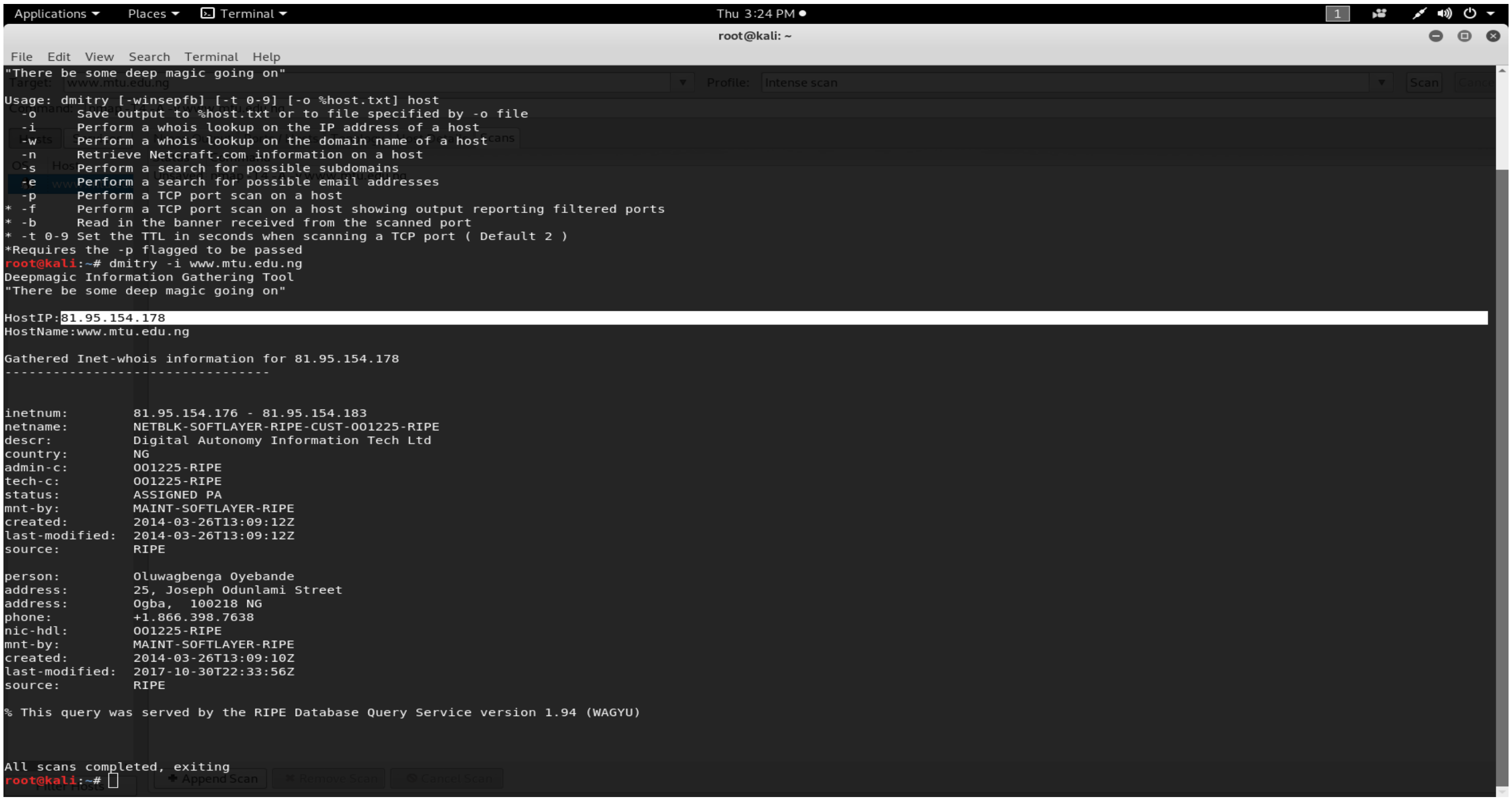
```
Loopback: lo: <live capture in progress>    Packets: 1637 · Displayed: 1637 (100.0%)    Profile: Default
```

Fig 4.6 Screen shot of Real time capture of tcp packets using Wireshark

4.4 Finding the Ip Address of the Domain Using Deep Information Gathering Tool

To view the ip address of the www.mtu.edu.ng page, the Dmitry (Deepmagic Information Gathering Tool) a UNIX/(GNU)Linux Command Line Application coded in C. Dmitry has the capacity to collect as much data about a host as possible. Base feature is capable of gathering possible subdomains, email addresses, uptime data, tcp port scanning, who is searches, and more by typing the following command on kali Linux terminal.

dmitry – I www.mtu.edu.ng.

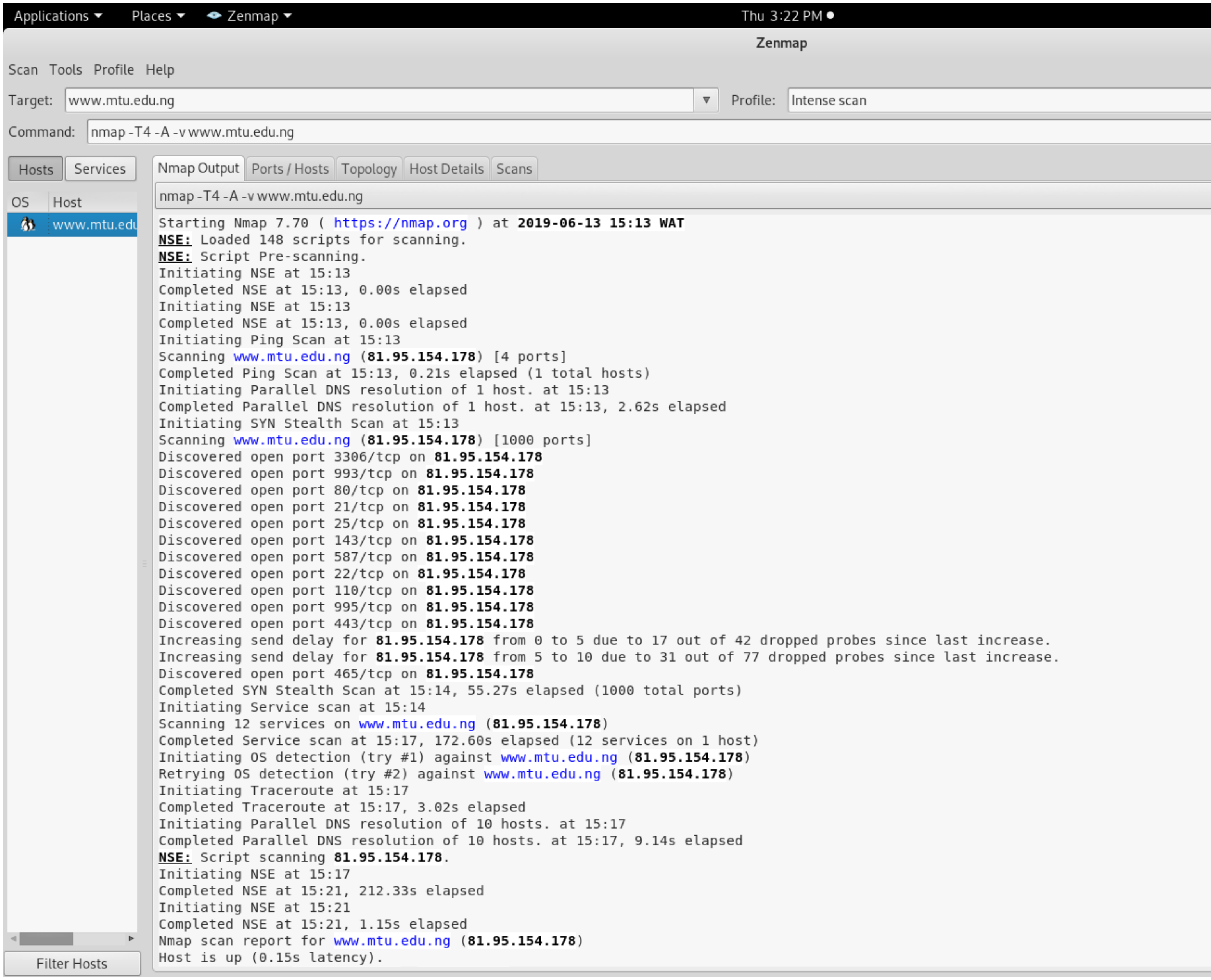


```
"There be some deep magic going on"

Usage: dmitry [-winsepfb] [-t 0-9] [-o %host.txt] host
  -o     Save output to %host.txt or to file specified by -o file
  -i     Perform a whois lookup on the IP address of a host
  -w     Perform a whois lookup on the domain name of a host
  -n     Retrieve Netcraft.com information on a host
  -s     Perform a search for possible subdomains
  -e     Perform a search for possible email addresses
  -p     Perform a TCP port scan on a host
* -f     Perform a TCP port scan on a host showing output reporting filtered ports
* -b     Read in the banner received from the scanned port
* -t 0-9 Set the TTL in seconds when scanning a TCP port ( Default 2 )
*Requires the -p flagged to be passed
root@kali:~# dmitry -i www.mtu.edu.ng
Deepmagic Information Gathering Tool
"There be some deep magic going on"

HostIP:81.95.154.178
HostName:www.mtu.edu.ng

Gathered Inet-whois information for 81.95.154.178
---------------------------------


inetnum:        81.95.154.176 - 81.95.154.183
netname:        NETBLK-SOFTLAYER-RIPE-CUST-001225-RIPE
descr:          Digital Autonomy Information Tech Ltd
country:        NG
admin-c:        001225-RIPE
tech-c:         001225-RIPE
status:         ASSIGNED PA
mnt-by:         MAINT-SOFTLAYER-RIPE
created:        2014-03-26T13:09:12Z
last-modified:  2014-03-26T13:09:12Z
source:         RIPE

person:         Oluwagbenga Oyebande
address:        25, Joseph Odunlami Street
address:        Ogba,  100218 NG
phone:          +1.866.398.7638
nic-hdl:        001225-RIPE
mnt-by:         MAINT-SOFTLAYER-RIPE
created:        2014-03-26T13:09:10Z
last-modified:  2017-10-30T22:33:56Z
source:         RIPE

% This query was served by the RIPE Database Query Service version 1.94 (WAGYU)


All scans completed, exiting
root@kali:~#
```

Fig 4.7 screen shot of finding the domain ip address

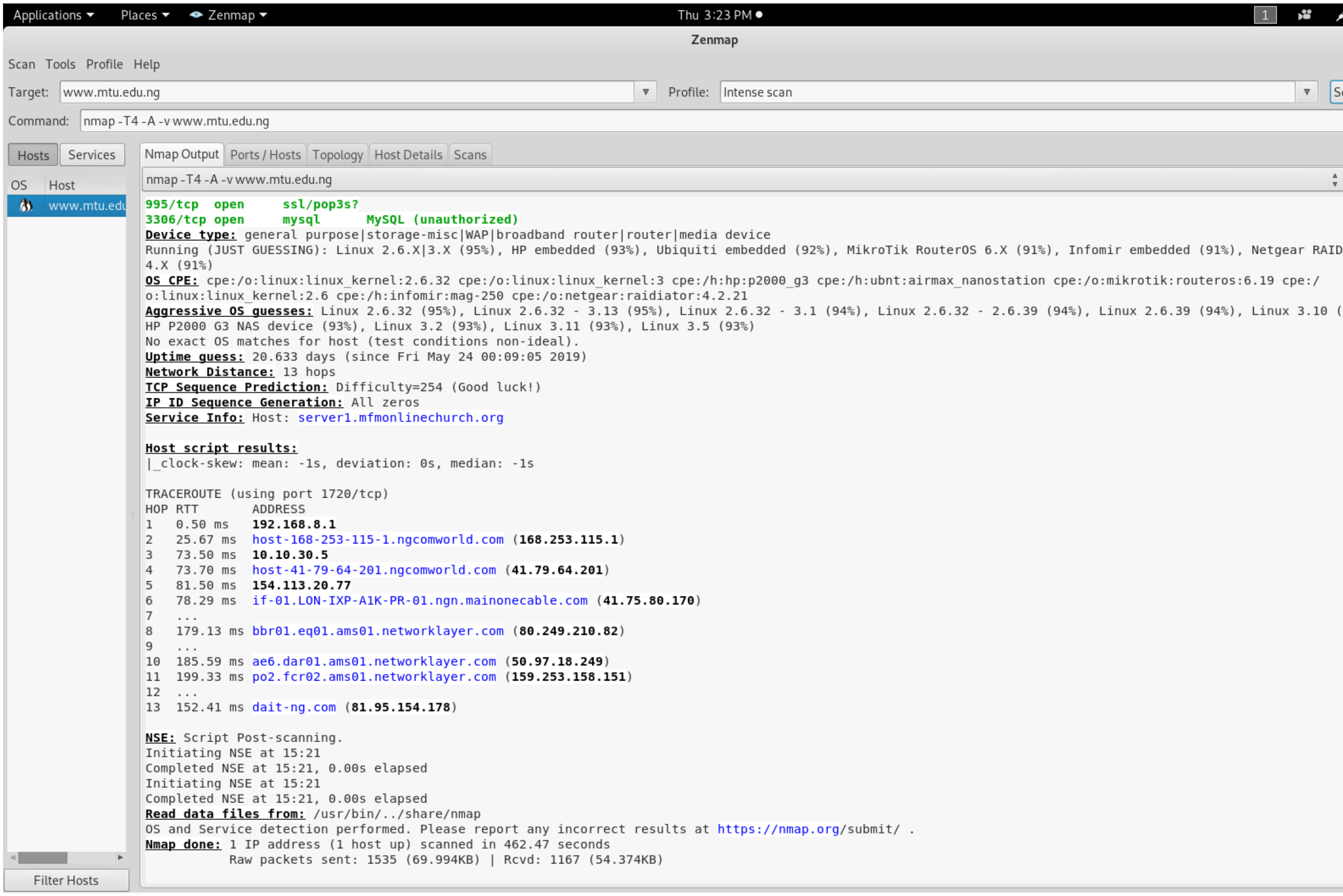
4.5 Starting Nmap/Zenmap Gui.

Enter in the target for your scan. Choose your target which is the Mountain Top University domain (www.mtu.edu.ng), or the IP address (81.95.154.178), then select intense scan for profile and then scan.



```
Starting Nmap 7.70 ( https://nmap.org ) at 2019-06-13 15:13 WAT
NSE: Loaded 148 scripts for scanning.
NSE: Script Pre-scanning.
Initiating NSE at 15:13
Completed NSE at 15:13, 0.00s elapsed
Initiating NSE at 15:13
Completed NSE at 15:13, 0.00s elapsed
Initiating Ping Scan at 15:13
Scanning www.mtu.edu.ng (81.95.154.178) [4 ports]
Completed Ping Scan at 15:13, 0.21s elapsed (1 total hosts)
Initiating Parallel DNS resolution of 1 host. at 15:13
Completed Parallel DNS resolution of 1 host. at 15:13, 2.62s elapsed
Initiating SYN Stealth Scan at 15:13
Scanning www.mtu.edu.ng (81.95.154.178) [1000 ports]
Discovered open port 3306/tcp on 81.95.154.178
Discovered open port 993/tcp on 81.95.154.178
Discovered open port 80/tcp on 81.95.154.178
Discovered open port 21/tcp on 81.95.154.178
Discovered open port 25/tcp on 81.95.154.178
Discovered open port 143/tcp on 81.95.154.178
Discovered open port 587/tcp on 81.95.154.178
Discovered open port 22/tcp on 81.95.154.178
Discovered open port 110/tcp on 81.95.154.178
Discovered open port 995/tcp on 81.95.154.178
Discovered open port 443/tcp on 81.95.154.178
Increasing send delay for 81.95.154.178 from 0 to 5 due to 17 out of 42 dropped probes since last increase.
Increasing send delay for 81.95.154.178 from 5 to 10 due to 31 out of 77 dropped probes since last increase.
Discovered open port 465/tcp on 81.95.154.178
Completed SYN Stealth Scan at 15:14, 55.27s elapsed (1000 total ports)
Initiating Service scan at 15:14
Scanning 12 services on www.mtu.edu.ng (81.95.154.178)
Completed Service scan at 15:17, 172.60s elapsed (12 services on 1 host)
Initiating OS detection (try #1) against www.mtu.edu.ng (81.95.154.178)
Retrying OS detection (try #2) against www.mtu.edu.ng (81.95.154.178)
Initiating Traceroute at 15:17
Completed Traceroute at 15:17, 3.02s elapsed
Initiating Parallel DNS resolution of 10 hosts. at 15:17
Completed Parallel DNS resolution of 10 hosts. at 15:17, 9.14s elapsed
NSE: Script scanning 81.95.154.178.
Initiating NSE at 15:17
Completed NSE at 15:21, 212.33s elapsed
Initiating NSE at 15:21
Completed NSE at 15:21, 1.15s elapsed
Nmap scan report for www.mtu.edu.ng (81.95.154.178)
Host is up (0.15s latency).
```

Fig 4.8 screen shot of zenmap intense scan



Target: www.mtu.edu.ng

Profile: Intense scan

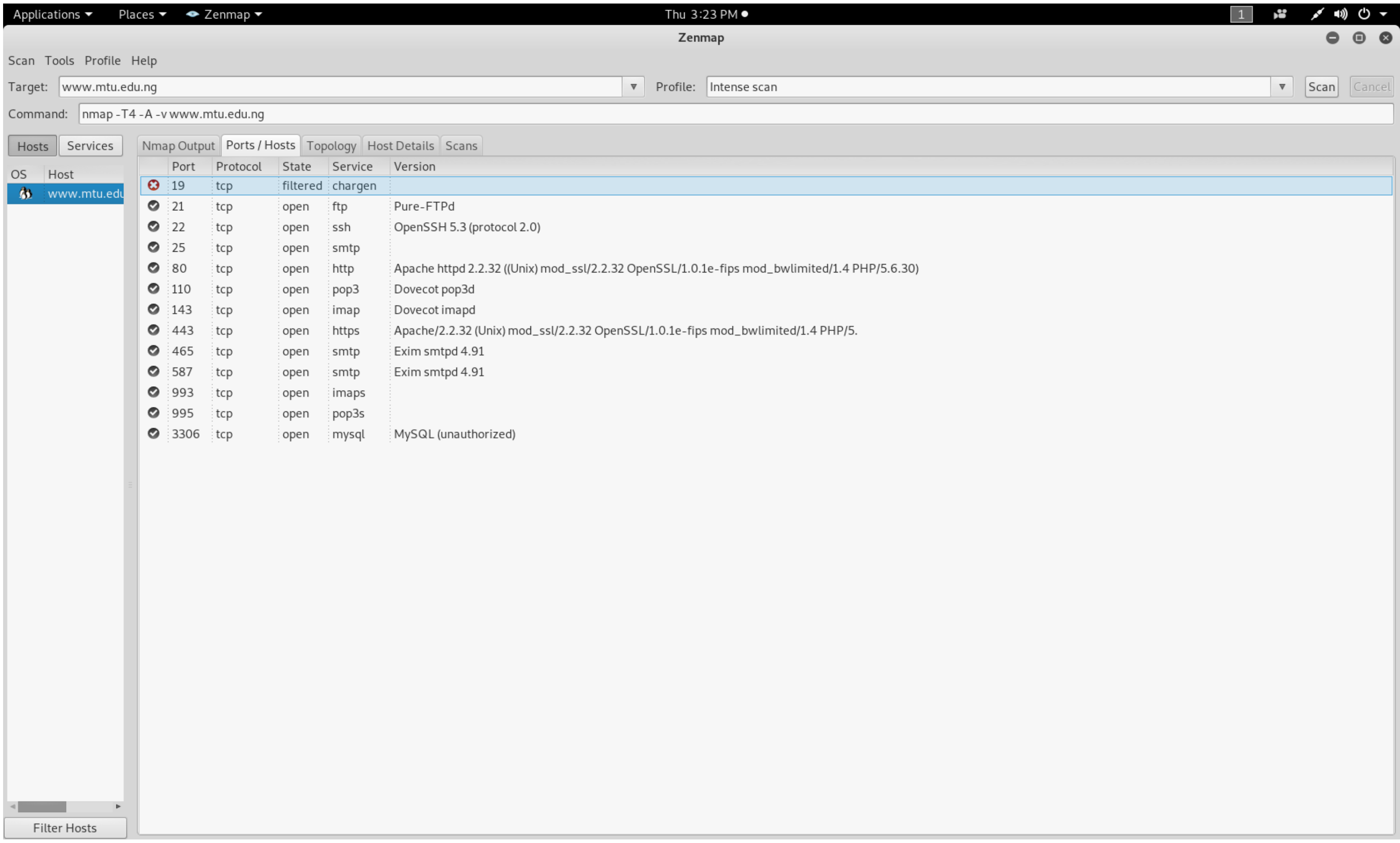
Command: nmap -T4 -A -v www.mtu.edu.ng

```
nmap -T4 -A -v www.mtu.edu.ng
```

```
995/tcp  open     ssl/pop3s?
3306/tcp open     mysql      MySQL (unauthorized)
Device type: general purpose|storage-misc|WAP|broadband router|router|media device
Running (JUST GUESSING): Linux 2.6.X|3.X (95%), HP embedded (93%), Ubiquiti embedded (92%), MikroTik RouterOS 6.X (91%), Infomir embedded (91%), Netgear RAID
4.X (91%)
OS CPE: cpe:/o:linux:linux_kernel:2.6.32 cpe:/o:linux:linux_kernel:3 cpe:/h:hp:p2000_g3 cpe:/h:ubnt:airmax_nanostation cpe:/o:mikrotik:routeros:6.19 cpe:/
o:linux:linux_kernel:2.6 cpe:/h:infomir:mag-250 cpe:/o:netgear:raidiator:4.2.21
Aggressive OS guesses: Linux 2.6.32 (95%), Linux 2.6.32 - 3.13 (95%), Linux 2.6.32 - 3.1 (94%), Linux 2.6.32 - 2.6.39 (94%), Linux 2.6.39 (94%), Linux 3.10 (
HP P2000 G3 NAS device (93%), Linux 3.2 (93%), Linux 3.11 (93%), Linux 3.5 (93%)
No exact OS matches for host (test conditions non-ideal).
Uptime guess: 20.633 days (since Fri May 24 00:09:05 2019)
Network Distance: 13 hops
TCP Sequence Prediction: Difficulty=254 (Good luck!)
IP ID Sequence Generation: All zeros
Service Info: Host: server1.mfmonlinechurch.org

Host script results:
|_clock-skew: mean: -1s, deviation: 0s, median: -1s

TRACEROUTE (using port 1720/tcp)
HOP RTT       ADDRESS
1   0.50 ms   192.168.8.1
2   25.67 ms  host-168-253-115-1.ngcomworld.com (168.253.115.1)
3   73.50 ms  10.10.30.5
4   73.70 ms  host-41-79-64-201.ngcomworld.com (41.79.64.201)
5   81.50 ms  154.113.20.77
6   78.29 ms  if-01.LON-IXP-A1K-PR-01.ngn.mainonecable.com (41.75.80.170)
7   ...
8   179.13 ms bbr01.eq01.ams01.networklayer.com (80.249.210.82)
9   ...
10  185.59 ms ae6.dar01.ams01.networklayer.com (50.97.18.249)
11  199.33 ms po2.fcr02.ams01.networklayer.com (159.253.158.151)
12  ...
13  152.41 ms dait-ng.com (81.95.154.178)

NSE: Script Post-scanning.
Initiating NSE at 15:21
Completed NSE at 15:21, 0.00s elapsed
Initiating NSE at 15:21
Completed NSE at 15:21, 0.00s elapsed
Read data files from: /usr/bin/../share/nmap
OS and Service detection performed. Please report any incorrect results at https://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 462.47 seconds
           Raw packets sent: 1535 (69.994KB) | Rcvd: 1167 (54.374KB)
```

Fig 4.9 screen shot of zenmap intense scan output

Step 11. Click on the ports/host to view the open and closed ports



Target: www.mtu.edu.ng

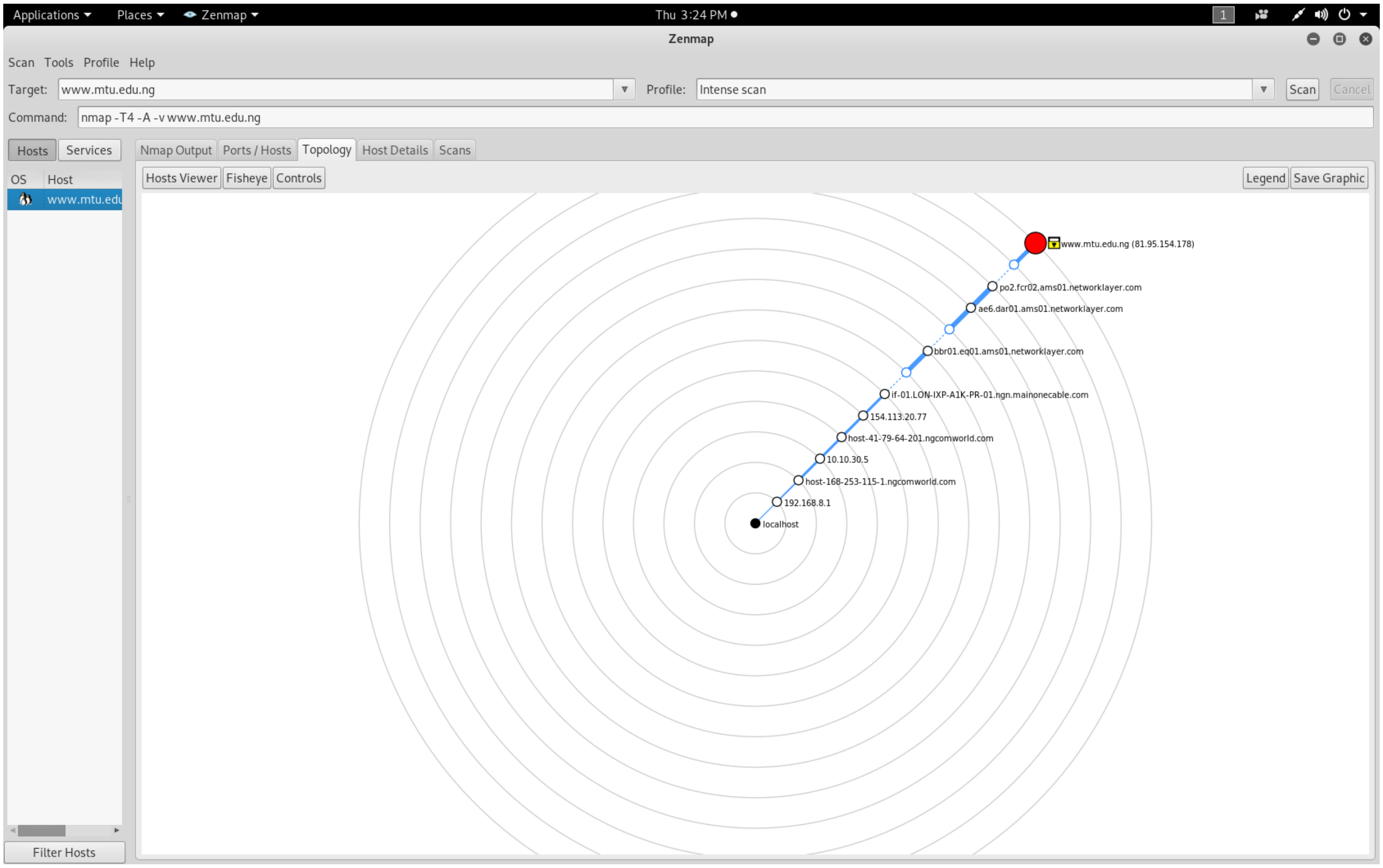
Profile: Intense scan

Command: nmap -T4 -A -v www.mtu.edu.ng

	Port	Protocol	State	Service	Version
☒	19	tcp	filtered	chargen	
✓	21	tcp	open	ftp	Pure-FTPd
✓	22	tcp	open	ssh	OpenSSH 5.3 (protocol 2.0)
✓	25	tcp	open	smtp	
✓	80	tcp	open	http	Apache httpd 2.2.32 ((Unix) mod_ssl/2.2.32 OpenSSL/1.0.1e-fips mod_bwlimited/1.4 PHP/5.6.30)
✓	110	tcp	open	pop3	Dovecot pop3d
✓	143	tcp	open	imap	Dovecot imapd
✓	443	tcp	open	https	Apache/2.2.32 (Unix) mod_ssl/2.2.32 OpenSSL/1.0.1e-fips mod_bwlimited/1.4 PHP/5.
✓	465	tcp	open	smtp	Exim smtpd 4.91
✓	587	tcp	open	smtp	Exim smtpd 4.91
✓	993	tcp	open	imaps	
✓	995	tcp	open	pop3s	
✓	3306	tcp	open	mysql	MySQL (unauthorized)

Fig. 4.10 screen shot of result for Ports/Hosts scan

Step 12. Click on topology to view the network map.



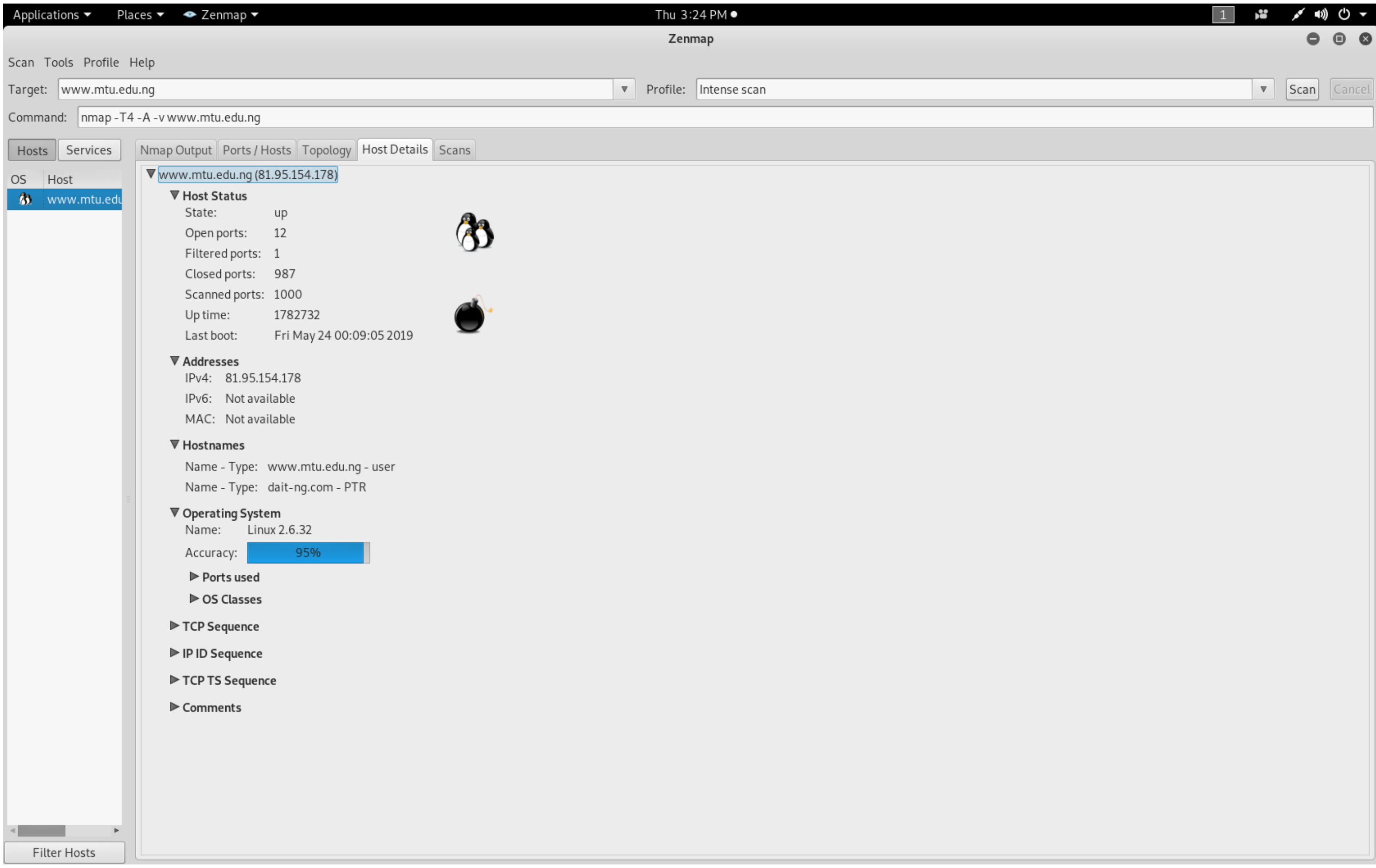
Target: www.mtu.edu.ng

Profile: Intense scan

Command: nmap -T4 -A -v www.mtu.edu.ng

Fig. 4.11 screen shot of topology scan

Step 13. Click on Host details to view the summary of the scan



Target: www.mtu.edu.ng

Profile: Intense scan

Command: nmap -T4 -A -v www.mtu.edu.ng

www.mtu.edu.ng (81.95.154.178)

Host Status

```
State:          up
Open ports:     12
Filtered ports: 1
Closed ports:   987
Scanned ports:  1000
Up time:        1782732
Last boot:      Fri May 24 00:09:05 2019
```

Addresses

```
IPv4:  81.95.154.178
IPv6:  Not available
MAC:   Not available
```

Hostnames

```
Name - Type:  www.mtu.edu.ng - user
Name - Type:  dait-ng.com - PTR
```

Operating System

```
Name:      Linux 2.6.32
Accuracy:  95%
```

Ports used

OS Classes

TCP Sequence

IP ID Sequence

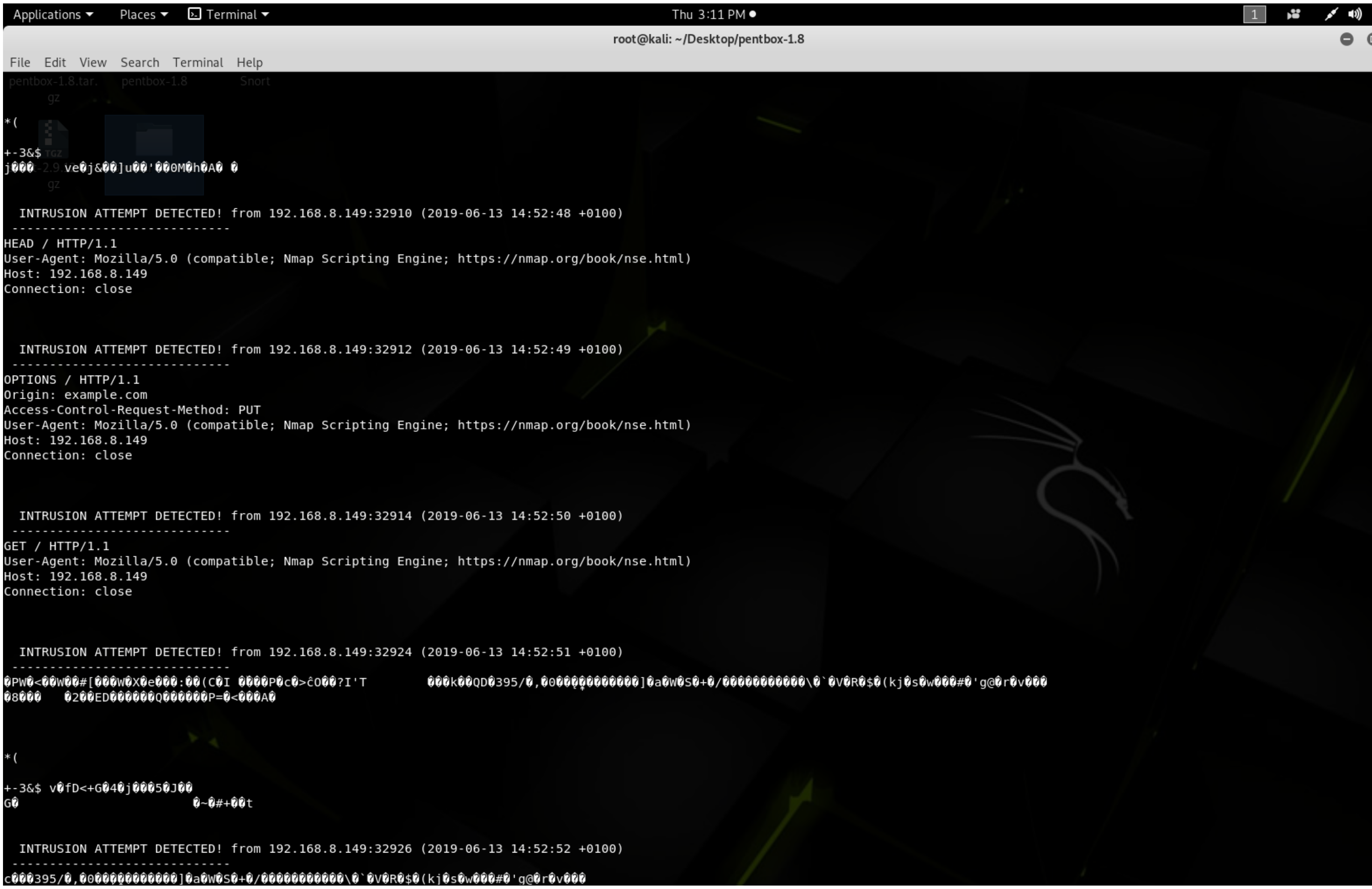
TCP TS Sequence

Comments

Fig. 4.12 Screen shot of host details

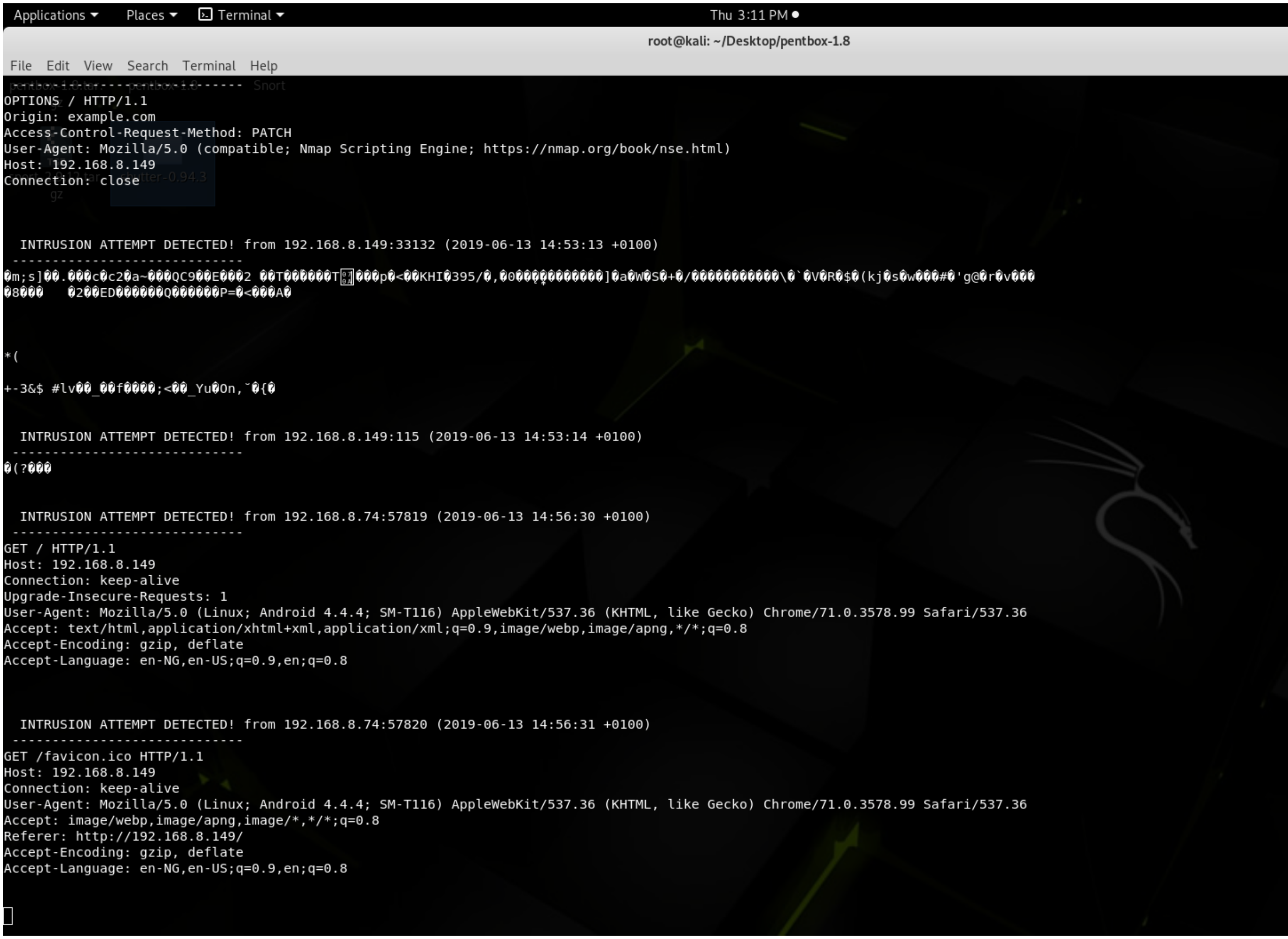
4.6 Result and Conclusion

After the service runs we will see how pentbox will record every connection made. Honeypots, by definition, see only "bad" traffic. Honeypots only report the connections they receive and most of these will be real attacks. Whenever a system tries to connect with the honeypot server, the server records its information.



```
*(

+-3&$
j���   ve�j&��]u��'��0M�h�A� �


  INTRUSION ATTEMPT DETECTED! from 192.168.8.149:32910 (2019-06-13 14:52:48 +0100)
 -----------------------------
HEAD / HTTP/1.1
User-Agent: Mozilla/5.0 (compatible; Nmap Scripting Engine; https://nmap.org/book/nse.html)
Host: 192.168.8.149
Connection: close



  INTRUSION ATTEMPT DETECTED! from 192.168.8.149:32912 (2019-06-13 14:52:49 +0100)
 -----------------------------
OPTIONS / HTTP/1.1
Origin: example.com
Access-Control-Request-Method: PUT
User-Agent: Mozilla/5.0 (compatible; Nmap Scripting Engine; https://nmap.org/book/nse.html)
Host: 192.168.8.149
Connection: close



  INTRUSION ATTEMPT DETECTED! from 192.168.8.149:32914 (2019-06-13 14:52:50 +0100)
 -----------------------------
GET / HTTP/1.1
User-Agent: Mozilla/5.0 (compatible; Nmap Scripting Engine; https://nmap.org/book/nse.html)
Host: 192.168.8.149
Connection: close



  INTRUSION ATTEMPT DETECTED! from 192.168.8.149:32924 (2019-06-13 14:52:51 +0100)
 -----------------------------
�PW�<��W��#[���W�X�e���:��(C�I ����P�c�>ĉO��?I'T        ���k��QD�395/�,�0������������]�a�W�S�+�/�����������\�`�V�R�$�(kj�s�w���#�'g@�r�v���
�8���   �2��ED�������Q�������P=�<���A�



*(

+-3&$ v�fD<+G�4�j���5�J��
G�                       �~�#+��t


  INTRUSION ATTEMPT DETECTED! from 192.168.8.149:32926 (2019-06-13 14:52:52 +0100)
 -----------------------------
c���395/�,�0������������]�a�W�S�+�/�����������\�`�V�R�$�(kj�s�w���#�'g@�r�v���
```

Fig 4.13 Screen shot of Honeypot showing the information about every connection that is detected.



```
 -----------------------------
OPTIONS / HTTP/1.1
Origin: example.com
Access-Control-Request-Method: PATCH
User-Agent: Mozilla/5.0 (compatible; Nmap Scripting Engine; https://nmap.org/book/nse.html)
Host: 192.168.8.149
Connection: close



  INTRUSION ATTEMPT DETECTED! from 192.168.8.149:33132 (2019-06-13 14:53:13 +0100)
 -----------------------------
�m;s]��.���c�c2�a~���QC9��E���2 ��T������T▯���p�<��KHI�395/�,�0������������]�a�W�S�+�/������������\�`�V�R�$�(kj�s�w���#�'g@�r�v���
�8���   �2��ED������Q������P=�<���A�



*(

+-3&$ #lv��_��f����;<��_Yu�On,˘�{�


  INTRUSION ATTEMPT DETECTED! from 192.168.8.149:115 (2019-06-13 14:53:14 +0100)
 -----------------------------
�(?���


  INTRUSION ATTEMPT DETECTED! from 192.168.8.74:57819 (2019-06-13 14:56:30 +0100)
 -----------------------------
GET / HTTP/1.1
Host: 192.168.8.149
Connection: keep-alive
Upgrade-Insecure-Requests: 1
User-Agent: Mozilla/5.0 (Linux; Android 4.4.4; SM-T116) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/71.0.3578.99 Safari/537.36
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/webp,image/apng,*/*;q=0.8
Accept-Encoding: gzip, deflate
Accept-Language: en-NG,en-US;q=0.9,en;q=0.8



  INTRUSION ATTEMPT DETECTED! from 192.168.8.74:57820 (2019-06-13 14:56:31 +0100)
 -----------------------------
GET /favicon.ico HTTP/1.1
Host: 192.168.8.149
Connection: keep-alive
User-Agent: Mozilla/5.0 (Linux; Android 4.4.4; SM-T116) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/71.0.3578.99 Safari/537.36
Accept: image/webp,image/apng,image/*,*/*;q=0.8
Referer: http://192.168.8.149/
Accept-Encoding: gzip, deflate
Accept-Language: en-NG,en-US;q=0.9,en;q=0.8
```

Fig 4.14 Screen shot of Honeypot showing the information about every connection that is detected.

Information recorded by our honeypot.

INTRUSION ATTEMPT DETECTED! From

192.168.8.149:32419 (2019-06-13 12:16:03 +0100)

GET / HTTP/1.1

Host: 192.168.8.149

Connection: keep-alive

Upgrade-Insecure-Requests: 1

```
User-Agent: Mozilla/5.0 (Linux; Android 4.4.4; SM-T116)

AppleWebKit/537.36 (KHTML, like Gecko)

Chrome/71.0.3578.99 Safari/537.36

Accept:

text/html,application/xhtml+xml,application/xml;q=0.9,ima ge/webp,image/apng,*/*;q=0.8

Accept-Encoding: gzip, deflate

Accept-Language: en- NG, en- US; q=0.9, en;q=0.8

----------------------------
```

After knowing which IPs are trying to attack, we can do prevention by blocking the ip or applying our tools as required. As you can see, we now know where the attacker came from, what exploit method was used and the time the exploit was attempted. Honeypots can be used in securing the network of Mountain Top University. Honeypots acts as an early alarming tools to secure the organization technologies and private data. Still IDS technology is not such powerful to protect global information infrastructure completely. In this work we have reviewed a basic honeypot. Hence we can safely conclude that different types of honeypots can be used to detect different types of attack signatures and rules can be defined to filter the traffic based on information gathered.

CHAPTER FIVE

SUMMARY AND CONCLUSION AND RECOMMENDATION

5.1 Summary

Network safety zone has made the greatest strides in latest years because no one expects their system to be assaulted by intruders. Honeypot technology is a significant and helpful component of a general network security approach if safety experts and scientists are to understand their opponents and ensure that network security keeps pace with the fast modifications in network assaults. No other mechanism is similar in a honeypot's effectiveness if the main objective is to collect data.

This work describes fresh methods of enhancing network security policies with honeypot, but we must also remember the reality that if an intruder knows about or bypasses such a scheme, the entire mechanism is irrelevant, so create a honeypot in such a manner that the intruder certainly believes it is the initial server of manufacturing. Strong control mechanism is needed because if the intruder is effective in managing the honeypot scheme, it will not be used by the attacker for further attacking purposes.

5.2 Conclusion

Honeypot is not a network security option, but a useful instrument complements other safety systems to create an additional active network security protection scheme. Working with IDS and firewall, Honeypot offers fresh ways to prevent, detect and respond to assaults. Because of its capacity to trap attackers into a decoy scheme, Honeypot can function as a useful deception instrument for item system avoidance. Honeypot, supplemented by IDS, decreases false positives and false negatives. Routing monitoring of intelligence offers flexible response to assaults. Different types of honeypots share prevalent information control and information capture techniques. Experts focus on the two to make the deployment of honeypot easier and harder to detect. From the developments in honeypot studies and manufacturing now days, I expect the inclusion, virtualization and delivery characteristics of the potential honeypot. Integrated honeypot encapsulates in a single unit of all the species. Virtual honeypot produces in one computer, a big amount of honeypot structures. Distributed honeypot includes various honeypot systems in a real network to provide strong communication between the assaults and system. They all create potential honeypot less expensive to apply, and simpler to keep.

Network security is not a route that many learners take, but when we talk about computing, we see it as one of the most significant subjects. This work has trained me a lot about the society of black hat and white hat. It also gave an understanding of how the forensic jobs are enormous and complicated. Every day new threats are found and keeping up to date is the best way to remain safe. Most assaults will have no impact on the scheme by doing this easy job. Nowadays, the issue is that individuals using an operating system's pirated version contribute to botnets. They don't sustain critical changes in their scheme and are more vulnerable to automated assaults. The detection of a honeypot scheme is not difficult these days, most of the job should concentrate on creating this technology stealthier.

5.3 Recommendation for Further Study

In the future, attempts can be produced to add some more algorithms and methods such as connection tracking, protocol analysis, and pattern detection in stream content etc. depending on which security administrator can conduct the assessment and retrieve the signature even more accurately. To create Honeypot more flexible, some more parameters such as enabling adverse input interpretation like Port! 445 that can also be attached to all ports except 445. There also requires to be a quantitative comparison between the current technique and the suggested approach to show the benefits of the suggested scheme over the current scheme.

5.4 Limitations

i. Time Constraints

ii. Scarcity of previous works

iii. Lack of constant wired or wireless network

iv. Lack of access to live servers.

REFERENCES

Adachi, Y., and Oyama, Y., (2009) *Malware Analysis System using Process-Level Virtualization:* Proceedings of IEEE Symposium on Computers and Communications, pp. 550-556.

Alosefer, Y., andRana, O., (2010) *Honeyware – WebbasedLow Interaction Client Honeypot*: Proceedings of the International Conference on Software Testing, Verification, and Validation Workshops, pp. 410-417.

Anagnostakis, K.G., et al, (2005) *Detecting Targeted Attacks Using Shadow Honeypots:* Proceedings of the Conference on USENIX Security Symposium, pp. 9-23.

Artail H., et al, (2006) *A hybrid honeypot framework for improving intrusion detection systems in protecting organizational networks.* Vol. 25 Pages 274-288

Berthier, R., et al (2008) *An Empirical Analysis on the Comparison of Network Attack Datasets: 11th IEEE High Assurance Systems Engineering Symposium*. *(HASE 2008), pp.39-48.*

Chawda, Kartik, and Ankit D. (2014) *Dynamic & hybrid honeypot model for scalable network monitoring: Information Communication and Embedded Systems (ICICES), International Conference on IEEE*

Cooke, E., et al (2004) *Understanding Distributed Blackhole Placement: Proceedings of the 2004 ACM workshop on Rapid malcode*, ACM. New York. pp. 54-64.

Das, V., (2009) *Honeypot Scheme for Distributed Denialof- Service*: Proceedings of the International Conference on Advanced Computer Control, pp. 497-501.

Gubbels, Kecia.,(2002).*Hands in the Honeypot.* GIAC Security Essentials Certification (GSEC).

Honeynet Project, http://www.honeynet.org. Accessed on 17/2/19. 13:24

http://labrea.sourceforge.net/labrea-info.html. Accessed 22/02/19. 14:18

Jian,B., et al. (2010)*Research on network security of defense based on Honeypot: International Conference on Computer Application and System Modeling (ICCASM 2010).Vol. 10.IEEE.*

Karthik, S., et al. (2004*) Design of Network Security Projects Using Honeypots: Journal of Computing Sciences in Colleges*

Koch, R., et al. (2013) *Attracting sophisticated attacks to secure systems: A New Noneypot Architecture: Communications and Network Security (CNS) on IEEE*

Li-Juan, Z.,(2009) *Honeypot-based defense system research and design: Computer Science and Information Technology .(ICCSIT 2009*)*.2nd IEEE International Conference on. IEEE*

Mokube, et al. (2007) *Honeypots: concepts, approaches, and challenges. Proceedings of the 45th annual southeast regional conference*. ACM.

Nazario, J., (2009) *PhoneyC: A Virtual Client Honeypot*: Proceedings of USENIX Workshop on Large-Scale and Emergent Threats, pp. 1-8.

Portokalidis, G., et al(2006) *Argos: an Emulator for Fingerprinting Zero-Day Attacks*. ACM SIGOPS Operating Systems Review, Vol. 40, No. 4, pp. 15-27.

Provos, N., (2004) *A virtual honeypot framework*: Proceedings of the 13th conference on USENIX Security Symposium, Vol. 13, SSYM'04, Berkeley, CA, USA.

Rowe, N., et al (2007) *Defending Cyberspace with Fake Honeypots*. Journal of Computers, Vol. 2, No. 2, pp. 25-36.

Schneier, B., (2000) *Secrets and Lies: Digital Security in a Networked World*. John Wiley & Sons.

Spitzner, L., (2003)*Honeypots: tracking hackers. Vol. 1. Reading*: Addison-Wesley.

Spitzner, L., (2003) Th*e Honeynet Project: Trapping the hackers.* pp. 15-23.IEEE Security & Privacy99.2

Spitzner, L., (2001).*The value of honeypots, part one: Definitions and values of honeypots.* Security Focus

Yegneswaran, V., et al (2005) *Using honeynets for internet situational awareness*: *Proceedings of the Fourth Workshop on Hot Topics in Networks (HotNets IV).Citeseer, pp. 17–22.*

Y.K.Jain and S. Singh (2011) *Honeypot based Secure Network SystemIn IJCSE*. Vol 3. No.2 Feb 2011.

Zhuge, J., et al (2007) *Collecting Autonomous Spreading Malware using High-Interaction Honeypots:* Proceedings of the International Conference on Information and Communications Security, pp. 438-451.