Syrian Private University

Faculty of Computer &

Informatics & Engineering

Department of Software

Building a device to operate the trap for hackers III

بناء جهاز لتشغيل مصيدة مخترقي الشبكات III

Prepare by

Mohammad Borhan Almalek

Adnan Almarje

Supervised by

Dr. Wassim Ahmad

Eng. Amjad Hijazi

Academic Year

2022-2023

SUPERVISOR CERTIFICATION

I certify that the preparation of this project entitled **Building a device to operate the trap for hackers**, prepared by **Mohammad Borhan Almalek** and **Adnan Almarje**, was made under my supervision at Department of Software & Information System Engineering/Faculty of Computer & Informatics & Engineering in partial fulfillment of the Requirements for the Degree of Bachelors of Software and Information System Engineering.

Dr. Wassim Ahmad

Thursday, 25 May 2023

Abstract

Building a device to operate a trap for hackers is a crucial step in detecting and mitigating cyber-attacks.

The device is designed to mimic a real-world target or valuable resource, in order to lure attackers and trap them in a controlled environment.

By doing this, the device can be used to study the tactics, techniques, and procedures of attackers and provide valuable information for improving security defenses and incident response capabilities. Additionally, the device can act as an early warning system, alerting security personnel to potential threats before they can cause significant harm. The device must be carefully designed and configured to appear as a realistic and attractive target to potential attackers while minimizing the risk of a successful compromise.

An effective project will have a design that can be updated and modified as the threat landscape evolves and the device should be able to provide real-time analysis and alerts to the security team.

ملخص

يعد بناء جهاز لتشغيل مصيدة للمتسللين خطوة حاسمة في اكتشاف الهجمات الإلكترونية والتخفيف من حدتها.

تم تصميم الجهاز لتقليد هدف في العالم الحقيقي أو مورد ثمين، من أجل جذب المهاجمين وحبسهم في بيئة خاضعة للرقابة.

من خلال القيام بذلك، يمكن استخدام الجهاز لدراسة تكتيكات وتقنيات وإجراءات المهاجمين وتوفير معلومات قيمة لتحسين الدفاعات الأمنية وقدرات الاستجابة للحوادث. بالإضافة إلى ذلك، يمكن للجهاز أن يعمل كنظام إنذار مبكر، حيث ينبه أفراد الأمن إلى التهديدات المحتملة قبل أن يتسببوا في ضرر كبير. يجب تصميم الجهاز وتكوينه بعناية ليظهر كهدف واقعي وجذاب للمهاجمين المحتملين مع تقليل مخاطر التسوية الناجحة.

سيكون للمشروع الفعال تصميم يمكن تحديثه وتعديله مع تطور مشهد التهديدات ويجب أن يكون الجهاز قادرًا على توفير تحليلات وتنبيهات في الوقت الفعلي لفريق الأمان

Contents

Table of Figures

List of Abbreviations

IDS	Intrusion Detection Systems
MHN	Modern Honey network
HIDS	Host-based IDS
NIDS	Network-based IDS
HTTP	Hyper Text Transfer Protocol
SMTP	Simple Mail Transfer Protocol
DMZ	Demilitarized Zone
HTML	Hyper Text Markup Language
SSH	SECURE SHELL
LAN	Local Area Network

Chapter 1
Introduction

1.1 Introduction:

First of all, we would like to build a honeypot on a machine. One of us will try to find security flaws that exist on the system. After defining all those, we will try to attack the system. Once the hacker will be able to have access into the system, one of us will have the role of forensic examiner. Using useful forensic investigation tools, he will try to find out the changes that occurred on the victim system by looking at the tracks left behind the hacker. Furthermore, we will go deeper into the subject thinking about its problems bringing to the system. It will be helpful for network security administrators to create more and more secure systems and be aware of the threats.

1.2 Motivation:

We are very interested in this subject field of study. So, our motivation for this project is to understand how security systems are working and how an organization can be protected and being aware of the risks of security flaws in the system. We will learn how a system is working and how it can be developed. Once we have the results, we will examine the output with forensic science tools. While trying all these, we will come across some problems and we will try to solve it. At the same time, we will have experience on creating and managing this kind of systems for the future. If we see similar problems in a network, we will be able to handle the system and recover the loss.

1.3 The Goal of The Project:

Is the honeypot device secure? Does the hacker know that it is a trap system? If the hacker realizes that it is a trap system, does he continue attacking to it? What does he gain from attacking it? Is it possible for the hacker to reach other systems and compromise them? We define our goals with :

- Design of a trap to deceive the attacker through operating systems and services that imitate what is within the system
- Deceive the attacker and divert his attention away from the actual production systems

- Research the techniques and methods used by the attacker and prevent their implementation on the actual network

Chapter 2
Theoretical Study

2.1 Introduction:

With the increasing risks to which any network is exposed, significantly and its impact on the work of the system in general, and despite the existence of mechanisms and tools to reduce them and prevent them from causing damage to the network, it cannot be said that there is a system that is free from risks or able to continue with that, as the presence of antidote Viruses, firewalls and intrusion detection systems do not mean that the network will be immune from the attacker’s access to it, all because of the presence of viruses and new attacks that rely on various methods and techniques to penetrate the network’s security infrastructure that cannot be discovered through the existing and previously mentioned techniques. It is necessary to have some techniques that are designed to be a trap for the attacker in order to be able to learn the new techniques he uses and motivate him to interact with them more to get the largest amount of valuable information. Design is to deceive the attacker through operating systems and services that imitate what is within the system in order to interact with it and extract as much valuable information as possible about the task. Agim, its used techniques, and the software tools used, as any data traffic with intrusion traps is considered suspicious because there is no security resource within it and it does not provide any real service

2.2 Intrusion Detection Systems:

An **Intrusion Detection System (IDS)** is a system that monitors **network traffic** for suspicious activity and issues alerts when such activity is discovered.

It is a software application that scans a network or a system for the harmful activity or policy breaching. Any malicious venture or violation is normally reported either to an administrator or collected centrally using a security information and event management (SIEM) system. A SIEM system integrates outputs from multiple sources and uses alarm filtering techniques to differentiate malicious activity from false alarms.

Although intrusion detection systems monitor networks for potentially malicious activity, they are also disposed to false alarms. Hence, organizations need to fine-tune their IDS products when they first install them.

It means properly setting up the intrusion detection systems to recognize what normal traffic on the network looks like as compared to malicious activity.

Intrusion prevention systems also monitor network packets inbound the system to check the malicious activities involved in it and at once send the warning notifications. (Cobb, 2021)

2.2.1 Classification of Intrusion Detection System:

IDS are classified into 5 types:

2.2.1.1 Network Intrusion Detection System (NIDS):

Network intrusion detection systems (NIDS) are set up at a planned point within the network to examine traffic from all devices on the network. It performs an observation of passing traffic on the entire subnet and matches the traffic that is passed on the subnets to the collection of known attacks. Once an attack is identified or abnormal behavior is observed, the alert can be sent to the administrator. An example of a NIDS is installing it on the subnet where firewalls are located in order to see if someone is trying to crack the firewall.

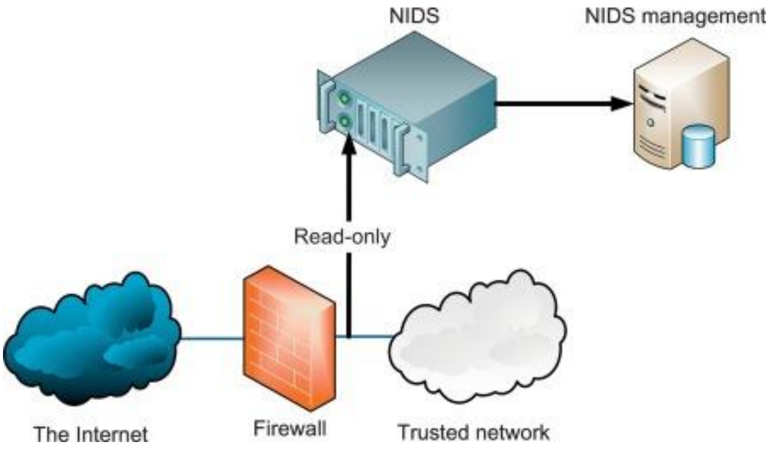


Figure 1 Network Intrusion Detection System

2.2.1.2 Host Intrusion Detection System (HIDS):

Host intrusion detection systems (HIDS) run on independent hosts or devices on the network. A HIDS monitors the incoming and outgoing packets from the device only and will alert the administrator if suspicious or malicious activity is detected. It takes a snapshot of existing system files and compares it with the previous snapshot. If the analytical system files were edited or deleted, an alert is sent to the administrator to investigate. An example of HIDS usage can be seen on mission-critical machines, which are not expected to change their layout.

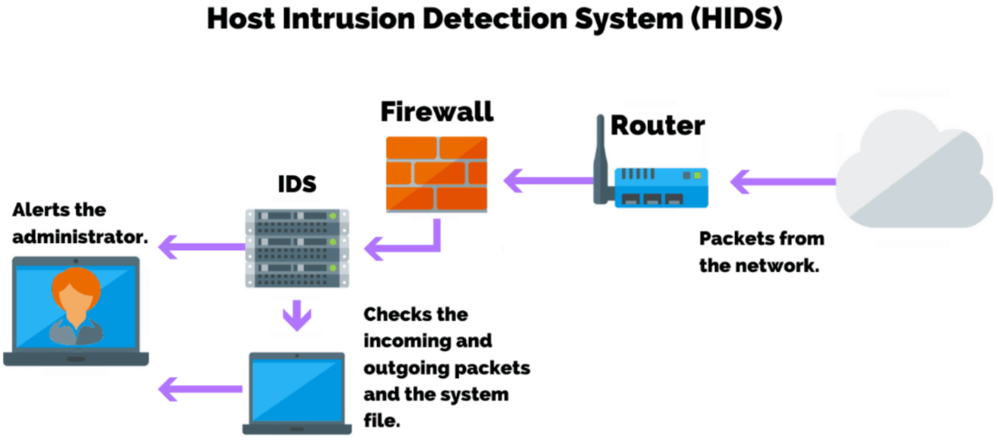

Host Intrusion Detection System (HIDS)
Firewall
Router
IDS
Alerts the administrator.
Packets from the network.
Checks the incoming and outgoing packets and the system file.

Figure 2 Host Intrusion Detection System

2.2.1.3 Protocol-based Intrusion Detection System (PIDS):

Protocol-based intrusion detection system (PIDS) comprises a system or agent that would consistently resides at the front end of a server, controlling and interpreting the protocol between a user/device and the server. It is trying to secure the web server by regularly monitoring the HTTPS protocol stream and accept the related HTTP protocol. As HTTPS is un-encrypted and before instantly entering its web presentation layer then this system would need to reside in this interface, between to use the HTTPS.

2.2.1.4 Application Protocol-based Intrusion Detection System (APIDS):

Application Protocol-based Intrusion Detection System (APIDS) is a system or agent that generally resides within a group of servers. It identifies the intrusions by monitoring and interpreting the communication on application-specific protocols. For example, this would monitor the SQL protocol explicit to the middleware as it transacts with the database in the web server.

2.2.1.5 Hybrid Intrusion Detection System:

Hybrid intrusion detection system is made by the combination of two or more approaches of the intrusion detection system. In the hybrid intrusion detection system, host agent or system data is combined with network information to develop a complete view of the network system. Hybrid intrusion detection system is more effective in comparison to the other intrusion detection system. Prelude is an example of Hybrid IDS.

2.2.2 Detection Method of IDS:

2.2.2.1 Signature-based Method:

Signature-based IDS detects the attacks on the basis of the specific patterns such as number of bytes or number of 1's or number of 0's in the network traffic. It also detects on the basis of the already known malicious instruction sequence that is used by the malware. The detected patterns in the IDS are known as signatures.

Signature-based IDS can easily detect the attacks whose pattern (signature) already exists in system but it is quite difficult to detect the new malware attacks as their pattern (signature) is not known.

2.2.2.2 Anomaly-based Method:

Anomaly-based IDS was introduced to detect unknown malware attacks as new malware are developed rapidly. In anomaly-based IDS there is use of machine learning to create a trustful activity model and anything coming is compared with that model and it is declared suspicious if it is not found in model. Machine learning-based method has a better-generalized property in comparison to signature-based IDS as these models can be trained according to the applications and hardware configurations

2.2.2.3 Comparison of IDS with Firewalls:

IDS and firewall both are related to network security but an IDS differs from a firewall as a firewall looks outwardly for intrusions in order to stop them from happening. Firewalls restrict access between networks to prevent intrusion and if an attack is from inside the network it doesn't signal. An IDS describes a suspected intrusion once it has happened and then signals an alarm.

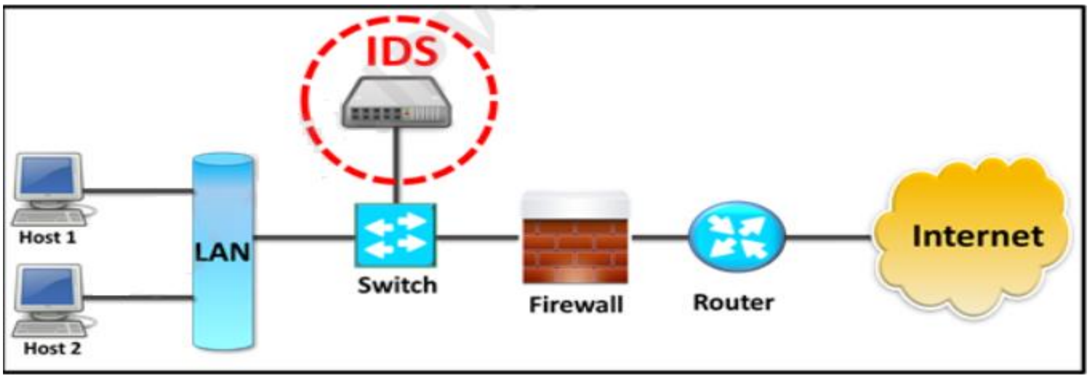


Figure 3 Intrusion Detection System

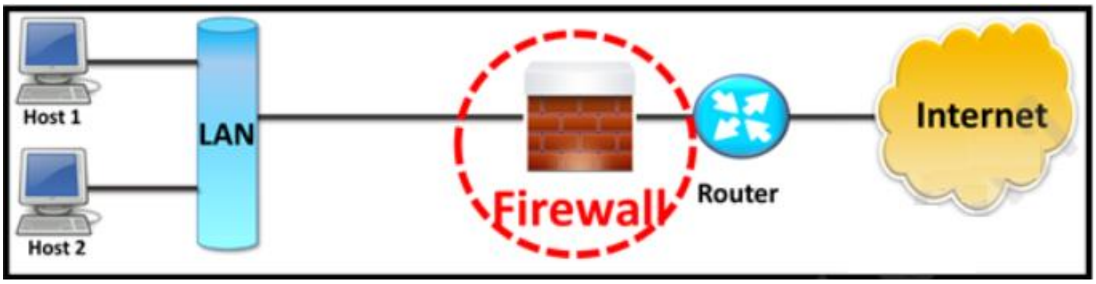


Figure 4 firewall

2.3What are a Honeypots

Some researchers consider this method a technique to detect attacks, while others consider it a real system designed to hack and learn from it.

According to the researcher Lance Spitzer, author of the book Honeypots Tracking Hackers, he gave an accurate description of them, where he defined these traps as a unique security resource that is considered part of the security environment of any system.

The interaction rate between it and the attacker is higher, the higher the effectiveness and value of the information extracted. It is useful to note that hack traps are not only a security solution, but rather a set of software tools whose use and dissemination in the network depends on the purpose to be used or according to the network section to be monitored.

It is able to imitate specific network applications or services, or to imitate entire operating systems with default settings, or to imitate a network that includes different systems with their services.

Honeypots are not considered as a tool for a security solution to a single specific problem, but rather they are designed and deployed within the system according to work requirements, unlike most traditional protection tools that are used to address specific problems and cannot be developed, as the traps are characterized by their high flexibility and adaptation, and this calls for explaining the ambiguity regarding the definition in general.(Honeypots, 2018)

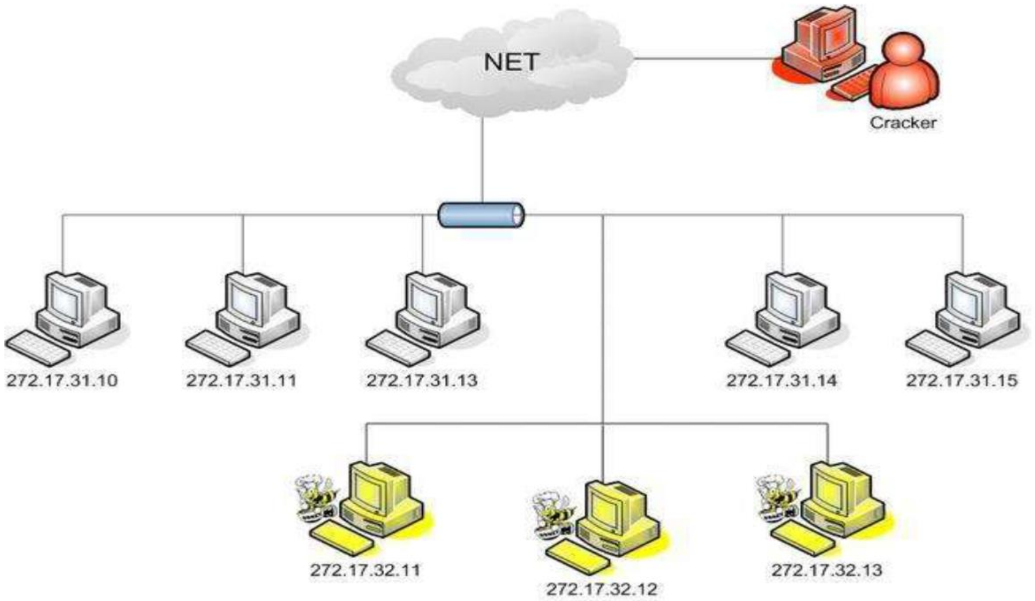


Figure 5 network with honeypots

2.4The Need for Honeypots

Although there are many traditional protection techniques dedicated to protecting the security infrastructure of any system

We need a new concept that aims to attract attackers and know their techniques more than trying to prevent them, as all traditional techniques can prevent specific types of attacks based on past events or a reference database of previous and known attacks, Therefore, it is difficult to detect any new attack or reveal advanced smart techniques for penetration, when studying any network, we find that the incidents that occur on it are much greater than its known weaknesses, because the attackers succeed by exploiting known and unknown weaknesses for the system administrator, so it is necessary to There is a monitoring mechanism that deals with real and actual incidents in real time and works on analyzing them, extracting valuable information from them, discovering the gaps that caused this breach, and trying to treat or prevent them from happening again. (GRIMES, 2005)

2.5Types of Honeypots

Honeypots are classified based on their deployment and the involvement of the intruder. Based on their deployment, honeypots are divided into:

2.5.1 Research honeypots:

These are used by researchers to analyze hacker attacks and deploy different ways to prevent these attacks. Research honeypots are mostly used by military, research and government organizations. They are capturing a huge amount of information. Their aim is to discover new threats and learn more about the Blackhat motives and techniques. The objective is to learn how to protect a system better, they do not bring any direct value to the security of an organization

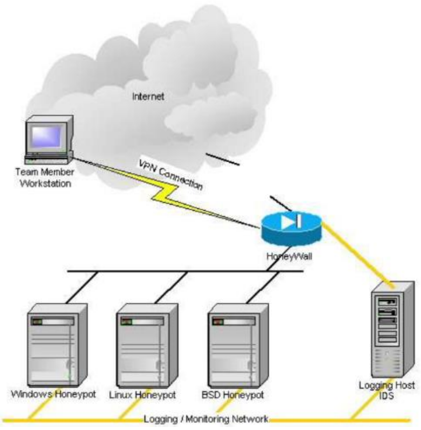


Figure 6 Honeypot Research Project

2.5.2 Production honeypots:

Production honeypots are deployed in production networks along with the server. These honeypots act as a frontend trap for the attackers, consisting of false information and giving time to the administrators to improve any vulnerability in the actual system. Production honeypots should imitate existing applications, services, and servers. If your production assets are fully patched, the honeypots should be fully patched as well. The key is to bait malicious hackers into thinking your honeypot system is a legitimate asset. If done correctly and with high interaction, it will be hard for hackers to know they have interacted with a honeypot. For example, suppose your network's server farm consists of Windows Server 2003 running IIS 6.0, Windows 2000 Server running Microsoft SQL Server 2000, Windows NT 4.0 Server, and a Windows 2000 Server running IIS 5.0. A production honeynet would attempt to mimic these same servers and services, as illustrated in Figure 1.

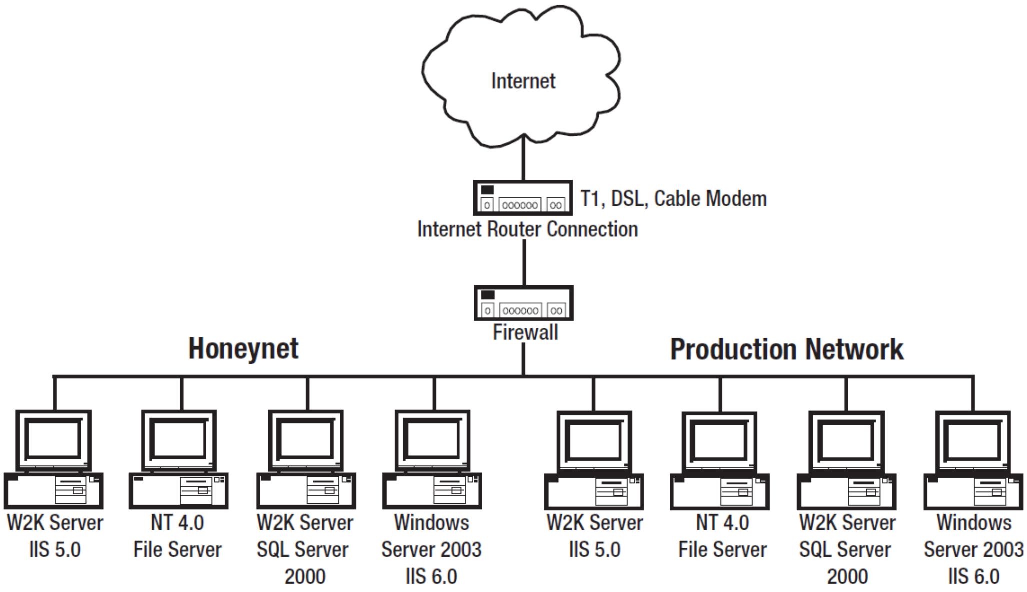


Figure 7 Example of a production honeynet

Based on interaction, honeypots are classified into:

2.5.2.1 Low interaction honeypots:

Low interaction honeypots give very little insight and control to the hacker about the network. It simulates only the services that are frequently requested by the attackers. The main operating system is not involved in the low interaction systems and therefore it is less risky. They require very fewer resources and are easy to deploy. The only disadvantage of these honeypots lies in the fact that experienced hackers can easily identify these honeypots and can avoid it.

2.5.2.2 Medium Interaction Honeypots:

Medium interaction honeypots allow more activities to the hacker as compared to the low interaction honeypots. They can expect certain activities and are designed to give certain responses beyond what a low-interaction honeypot would give.

2.5.2.3 High Interaction honeypots:

A high interaction honeypot offers a large no. of services and activities to the hacker, therefore, wasting the time of the hackers and trying to get complete information about the hackers. These honeypots involve the real-time operating system and therefore are comparatively risky if a hacker identifies the honeypot. High interaction honeypots are also very costly and are complex to implement. But it provides us with extensively large information about hackers.

2.5.3 Advantages of honeypot:

1. Acts as a rich source of information and helps collect real-time data.
2. Identifies malicious activity even if encryption is used.
3. Wastes hackers’ time and resources.
4. Improves security

2.6 Placement of the Honeypot

Honeypots do not need a specific environment in order to adapt to it because they are considered as standard systems without any specific needs, as they can be located anywhere within the network according to the required design and the level of protection to be achieved. In the figure, we show three main areas in which traps can be located

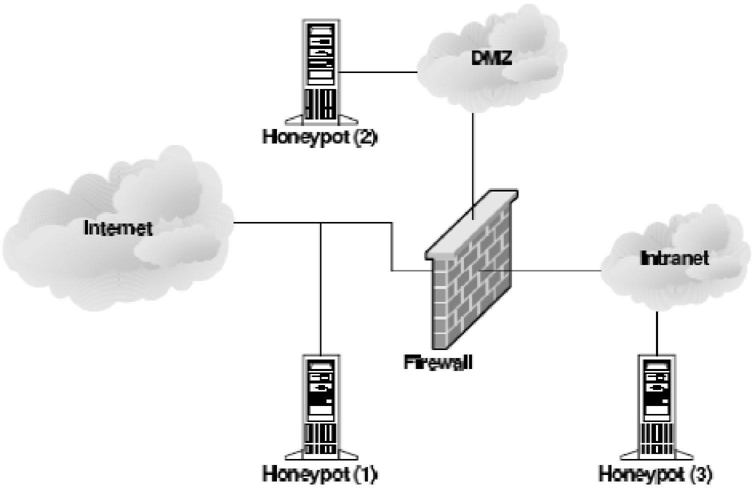


Figure 8 The main cases of honeypot

2.6.1 In front of the firewall

In this case, the danger to the internal network does not increase, and the presence of a malicious system behind it becomes a bad thing Possible, but this can cause a special problem especially if there is no other internal firewall that protects some important resources from any internal attack, in such cases traps will document all unwanted activity from outside the network such as a port scan mechanism (Port Scan), then these activities will not be documented by the firewall or internal IDS system so it will not generate internal alerts, in general this model does not increase the risk to the network and it reduces the possibility of a new attack, but the only problem is It lies in the event that there are internal tasks, it can easily cause damage within the network, especially if the firewall prevents the passage of data from inside the network to the Honeypots

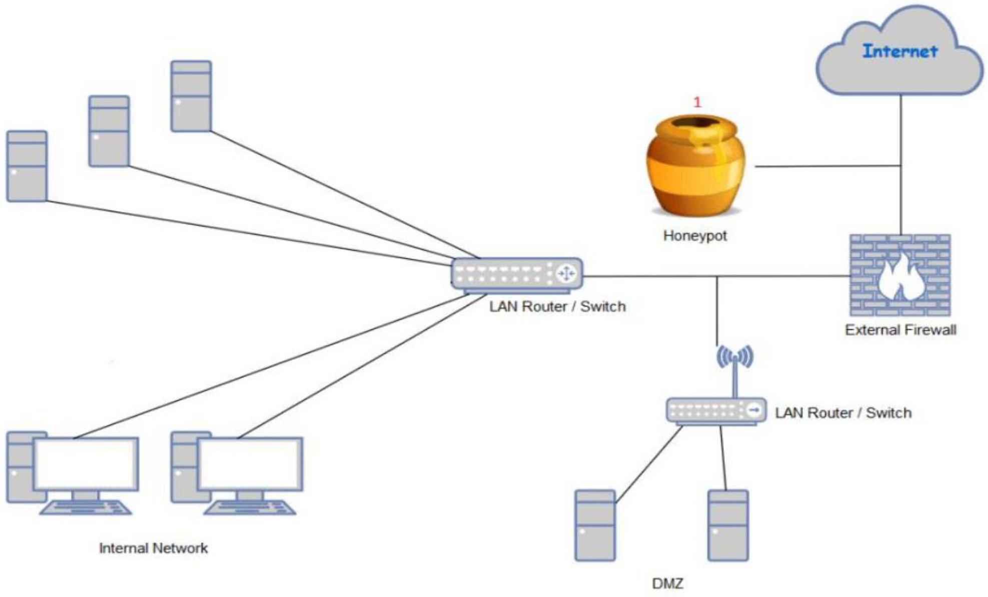


Figure 9. In front of the firewall

2.6.2 inside DMZ

The presence of traps inside the DMZ is considered one of the best solutions where all servers are secured and not accessible from an external or internal attacker. In addition, it facilitates the process of imitating these servers, but it is necessary in the case of traps behind the firewall that all ports are opened through the firewall, Which calls for greater risk and longer time, but by being inside the DMZ, these risks are eliminated. The only drawback here is the increased burden and demand on these devices, which requires extensive management.

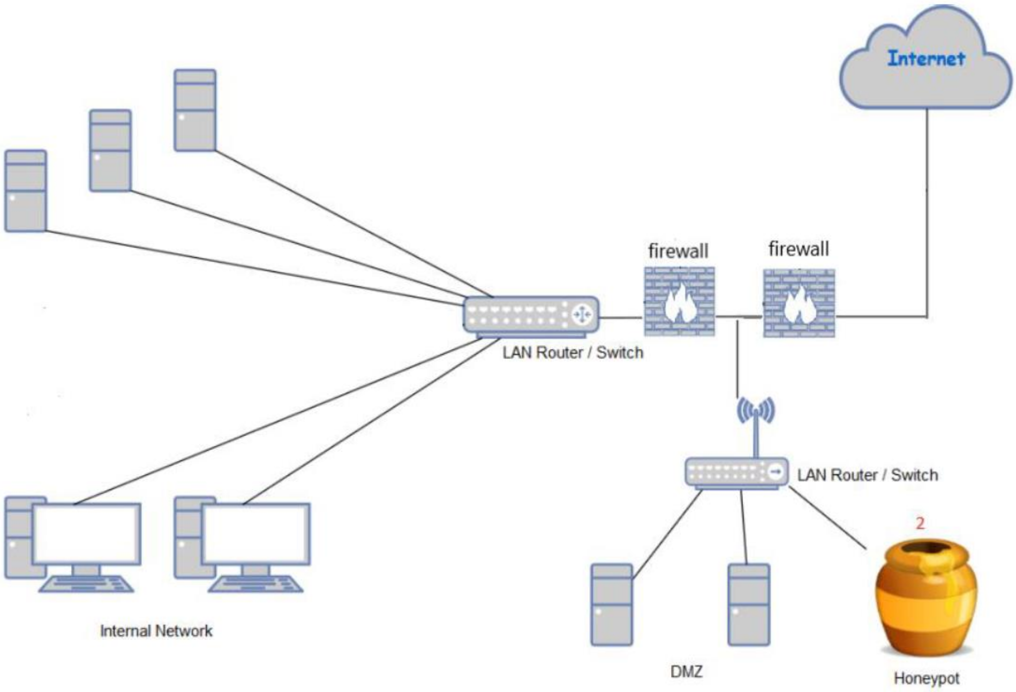


Figure 10 honeypot inside DMZ

2.6.3 Behind the firewall

The reason behind this is to detect internal attacks that the network may be exposed to, penetration traps provide many fake services, and in order to be breached, the firewall rules must be set to allow these activities to pass through, and not to generate alerts whenever they are attacked Intrusion traps, the danger here lies in the failure of the traps and the attacker's ability to enter the network, then it becomes like a port for the attacker into the network, and then the firewall cannot detect this because it has already considered it as belonging to the traps and allowed it to cross, so it is important to protect the network internally and make this protection mandatory, especially If the high level traps interact.

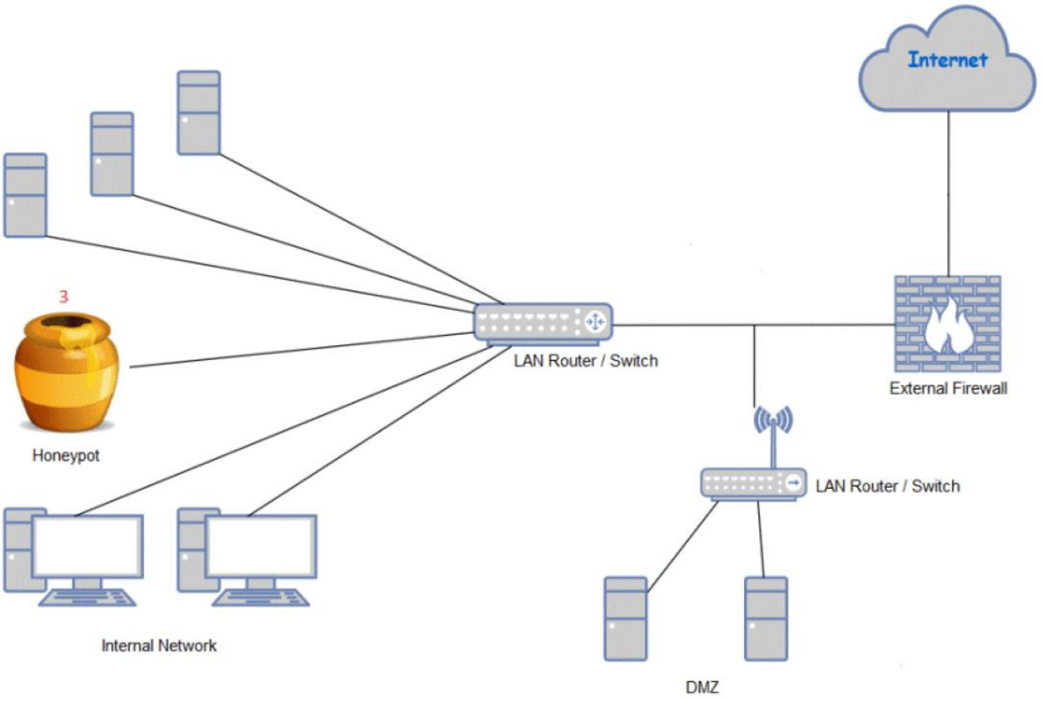


Figure 11 Behind the firewall

Chapter 3
System Analysis & Design

3.1 Suggested Framework

3.1.1 Modern Honey Network:

MHN is a enterprise ready honeypot management system which enables organizations to create a fully functional active-defense network in minutes.

Honeypots have not received wide adoption as an enterprise defense largely because the deployment and management has been a complicated process reserved for security companies and security researchers.

MHN is a centralized server for management and data collection of honeypots. MHN allows you to deploy sensors quickly and to collect data immediately, viewable from a neat web interface. Honeypot deploy scripts include several common honeypot technologies, including Snort, Cowrie, Dionaea, and glastopf, among others.

3.1.2 Advantages of Modern Honey Network:

- Enables organizations to create honeypots in minutes.
- Ease of deploying more than one Honeypot in the same server.
- Making the process of designing penetration traps easier and simpler than before.
- Easily manage and collect information from Honeypots and display them in a graphical interface.
- Provide a variety of results to the attackers and their methods, thus knowing what they are trying access to it in order to protect it.
- Open source under the GNU GPLv3 license

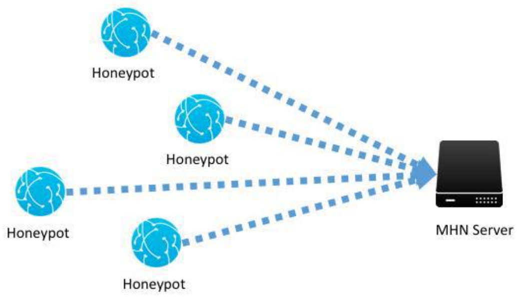


Figure 12 MHN central server managing

3.1.3 Types of Honeypots used in project:

- SSH−Honeypot−(Cowrie)
- Python−Honeypot−(Amun)
- Python−Honeypot−(Dionaea)

3.1.3.1 SSH-Honeypot-(Cowrie):

Cowrie is a medium to high interaction SSH and Telnet honeypot designed to log brute force attacks and the shell interaction performed by the attacker. In medium interaction mode (shell) it emulates a UNIX system in Python, in high interaction mode (proxy) it functions as an SSH and telnet proxy to observe attacker behavior to another system.

Cowrie is maintained by Michel Oosterhof.

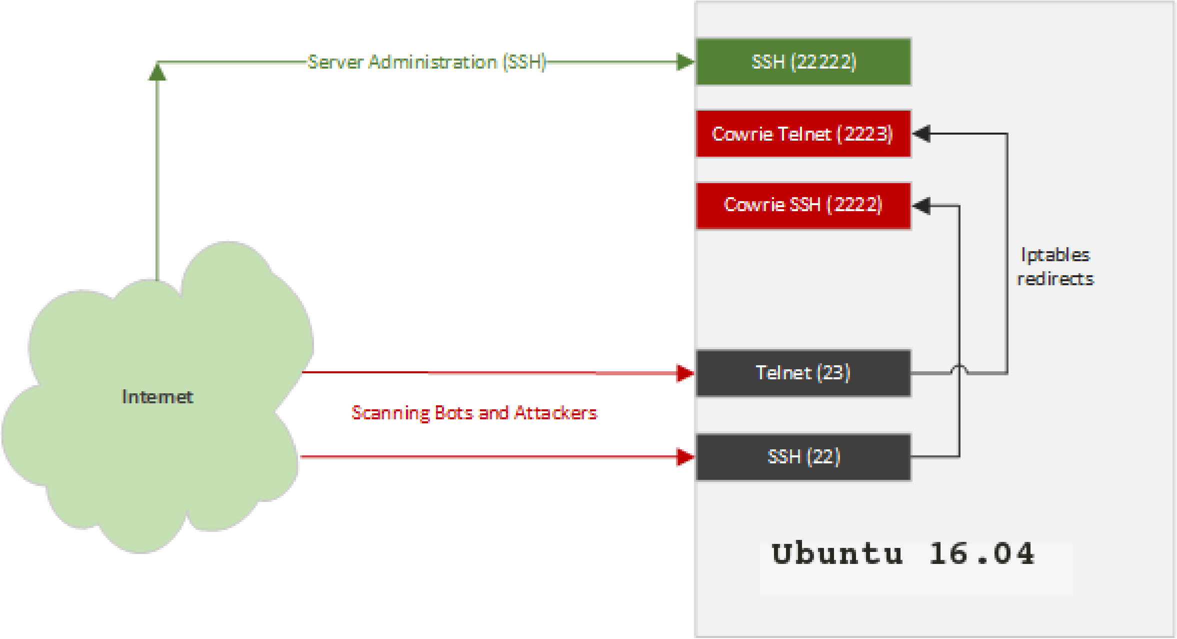


Figure 13 SSH-Honeypot

Cowrie Features:

Choose to run as an emulated shell:

- Fake filesystem with the ability to add/remove files. A full fake filesystem resembling a Debian 5.0 installation is included
- Possibility of adding fake file contents so the attacker can cat files such as /etc/passwd. Only minimal file contents are included
- Cowrie saves files downloaded with wget/curl or uploaded with SFTP and scp for later inspection

Or proxy SSH and telnet to another system

- Run as a pure telnet and ssh proxy with monitoring

For both settings:

- Session logs are stored in an UML Compatible format for easy replay with the bin/playlog utility.
- SFTP and SCP support for file upload
- Support for SSH exec commands
- Logging of direct-tcp connection attempts (ssh proxying)
- Forward SMTP connections to SMTP Honeypot (e.g. mailoney)
- JSON logging for easy processing in log management solutions

3.1.3.2 Python honeypot-(Amun):

Amun was the first python-based low-interaction honeypot, following the concepts of Nepenthes but extending it with more sophisticated emulation and easier maintenance.

Fake services are run on it and rely only on eavesdropping and do not provide interaction with the attacker.

It is used only to generate alerts to suspicious activity coming to the network. These services are eavesdropped through a specific port, such as eavesdropping on port number 83 of the http protocol, meaning that it can be considered as a one-way communication only that is eavesdropped but not answered, this method reduces the risk to the system by A way to distract the attacker and take advantage of his time.

3.1.3.3 Python honeypot-(Dionaea):

A Dionaea honeypot is a type of honeypot used to trap and monitor malicious activity on a computer network. It is designed to emulate vulnerabilities in common Internet services, such as those found in the Windows operating system or the Simple Network Management Protocol (SNMP). The goal of a Dionaea honeypot is to attract and detect hackers, malware, and other malicious activity by mimicking real vulnerabilities and services. This allows organizations to identify and track potential security threats, as well as gather information that can be used to improve the security of their networks.

3.2 Suggested for hardware

3.2.1 Raspberry pi:

The Raspberry Pi is a low cost, credit-card sized computer that plugs into a computer monitor or TV, and uses a standard keyboard and mouse. It is a capable little device that enables people of all ages to explore computing, and to learn how to program in languages like Scratch and Python. It’s capable of doing everything you’d expect a desktop computer to do, from browsing the internet and playing high-definition video, to making spreadsheets, word-processing, and playing games. What's more, the Raspberry Pi has the ability to interact with the outside world, and has been used in a wide array of digital maker projects, from music machines and parent detectors to weather stations and tweeting birdhouses with infra-red cameras.

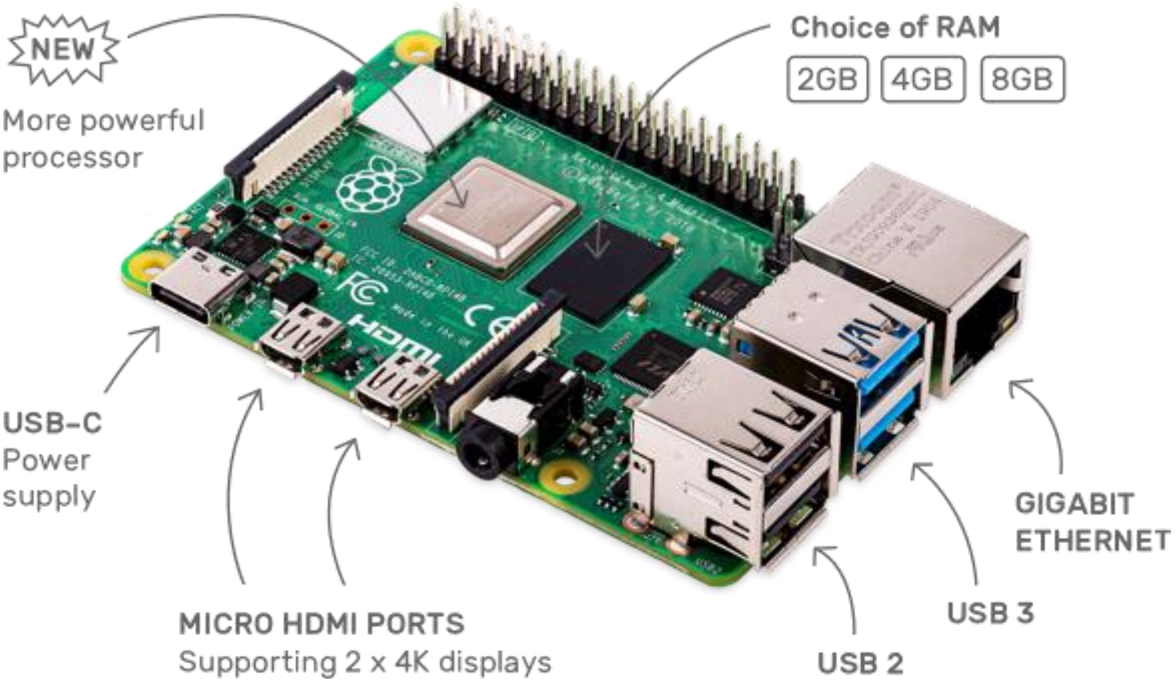


Figure 14 Raspberry pi

Raspberry Pi is the name of a series of single-board computers made by the Raspberry Pi Foundation, a UK charity that aims to educate people in computing and create easier access to computing education.

The Raspberry Pi launched in 2012, and there have been several iterations and variations released since then. The original Pi had a single-core 700MHz CPU and just 256MB RAM, and the latest model has a quad-core CPU clocking in at over 1.5GHz, and 4GB RAM

All over the world, people use the Raspberry Pi to learn programming skills, build hardware projects, do home automation, implement Kubernetes clusters and Edge computing, and even use them in industrial applications.

The Raspberry Pi is a very cheap computer that runs Linux, but it also provides a set of GPIO (general purpose input/output) pins, allowing you to control electronic components for physical computing and explore the Internet of Things (IoT).

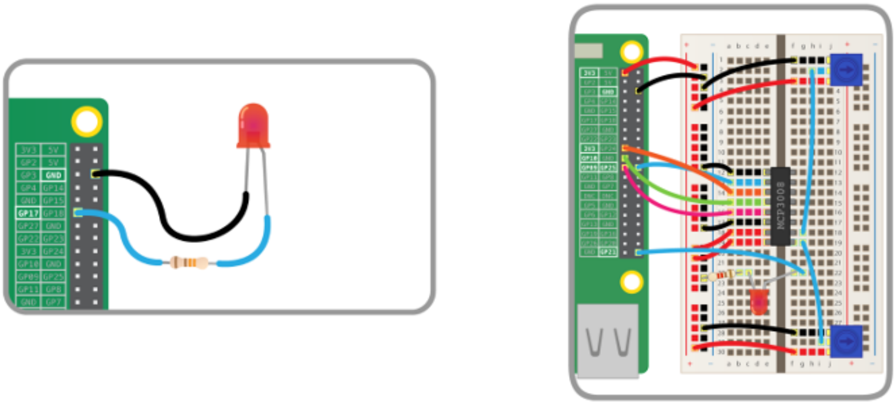


Figure 15 GPIO Raspberry pi

3.3 Analytical Study

3.3.1 Introduction:

Information security is the way in which information is preserved from theft, hacking, deletion, alteration and forgery, ensuring that it remains secure or shared with others without being subjected to interception, spying, hacking and hacking, and ensuring that it does not reach a person who does not have access to it.

Implementing requires careful planning to assure that it meets expectations. These are the basic steps:

1. We have defined the requirements.
2. We've been looking for an environment that accommodates what we're going to do.
3. We have studied the system that we will work on and apply it.
4. We have studied and researched the hardware parts we need.
5. Finally, we simulated what we're going to do.

3.3.2 Define requirements:

- Detect a breach before it happens
- Know the goals of the hacker
- Hacking methods review
- Not affected by hacking
- Protection from intrusions
- Transferring the concept of honeypot from software to device.

3.3.3 Work environment:

When you simulate a job, there is nothing better than Windows 10

We used it and installed VMware workstation:

Figure 16 VMware icon

We installed Ubuntu 18.4

To have a work-ready environment

3.3.4 The operating system:

Ubuntu is a Linux distribution based on Debian and composed mostly of free and open-source software. Ubuntu is officially released in three editions: Desktop, Server, and Core for Internet of things devices and robots. Ubuntu is a popular operating system for cloud computing, with support for OpenStack. Ubuntu's default desktop has been GNOME since version 17.10.

Ubuntu 18.04 is codenamed Bionic Beaver. This is not surprising considering the logic behind the codename and versioning of Ubuntu releases.

Ubuntu’s founder Mark Shuttleworth dedicated the hardworking attribute of beaver to the Ubuntu team

Figure 17 Ubuntu icon

We have adopted this version because it meets the operating requirements and works properly

3.3.5 Hardware:

We've been looking for a device that runs all the equipment we've collected

And we thought it was the right Raspberry Pi

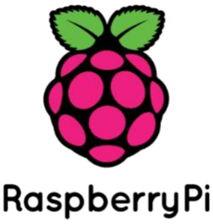


Figure 18 Raspberry Pi icon

MHN hardware requirements

MHN server:

1. 4 GB Ram
2. Dual Core Processor
3. 40 Gb Drive

Honey node:

1. 512 Mb – 1 Gb
2. Dual Core CPU
3. 20 Gb Drive

The specific deployment depends on the location of the honeypot, behind the firewall or directly exposed to the Internet, the number of attacks is different, and the resources consumed are definitely different.

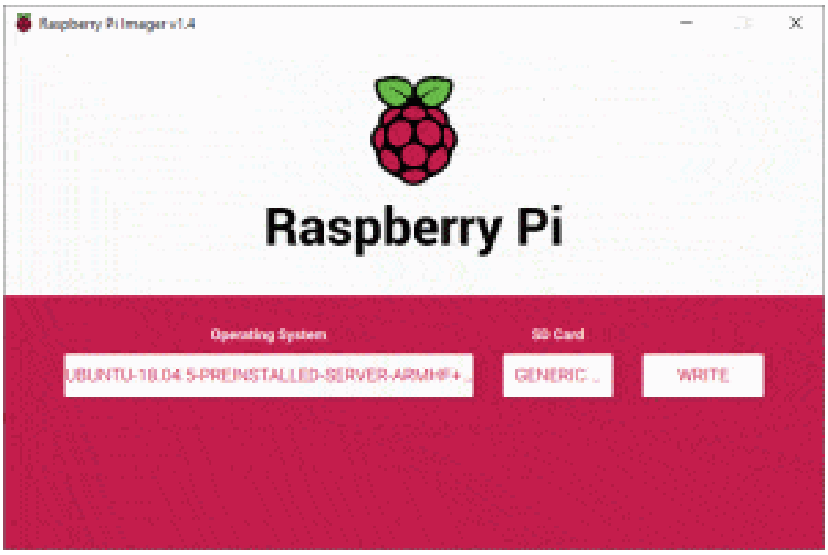
Chapter 4
Implementation

4.1 OS Installation:

As mentioned above, I opted for the “Ubunto 18.6” (headless) version which means it comes with no desktop or gui interface – its command line only. I did this because I wanted the best performance from the device, no unnecessary applications/services and

Finally, MHN Server own installation steps suggest running MHN server in a virtual container. Given that its unlikely I’m going to be using my Raspberry Pi for an additional workload, I install directly to keep things simple.

Prepare the SD card



Operating System: UBUNTU-18.04.5-PREINSTALLED-SERVER-ARMHF+...

SD Card: GENERIC...

Figure 19 Raspberry Pi Imager

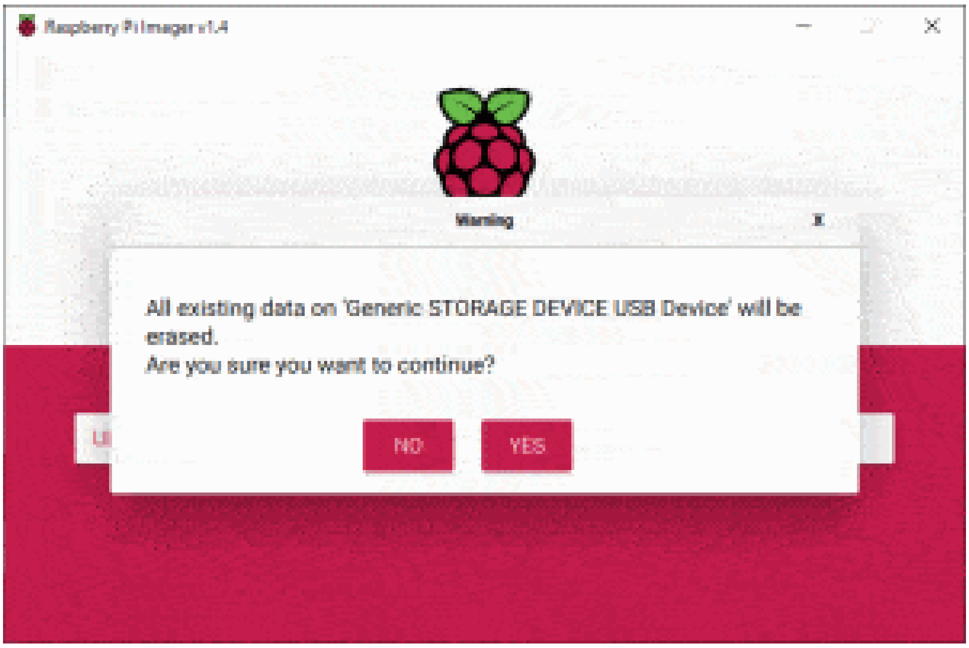
Run the Raspberry PI Imager software

Insert your SD card in to your reader

On the Raspberry PI Imager, select the **Raspberry PI OS (other)** option from the Operating System menu

Select **Ubuntu-18.04-server**

Select your SD card (double check, personally I tend to remove any other flash drives or SD cards just in case!)



Raspberry Pi Imager v1.4

Warning

All existing data on 'Generic STORAGE DEVICE USB Device' will be erased.
Are you sure you want to continue?

Figure 20 Raspberry Pi Imager

Click Write

Click Yes to confirm you understand all data on the SD card will be destroyed

This will take a while so go grab a cup of tea (and biscuits if you have them)

Enable SSH

By default, SSH is disabled on Raspberry PI devices so if you are going to be configuring this remotely, you must turn this on first.

The easiest way to do this is while you still have the SD card in your computer after formatting it.

Simply open the partition called "boot" in Windows Explorer (or equivalent) and create an empty file there with a filename of either ssh or ssh.txt.

When your Raspberry PI boots up, if it finds either of those files, it enables the SSH service (and deletes the files)

4.2 MHN installation:

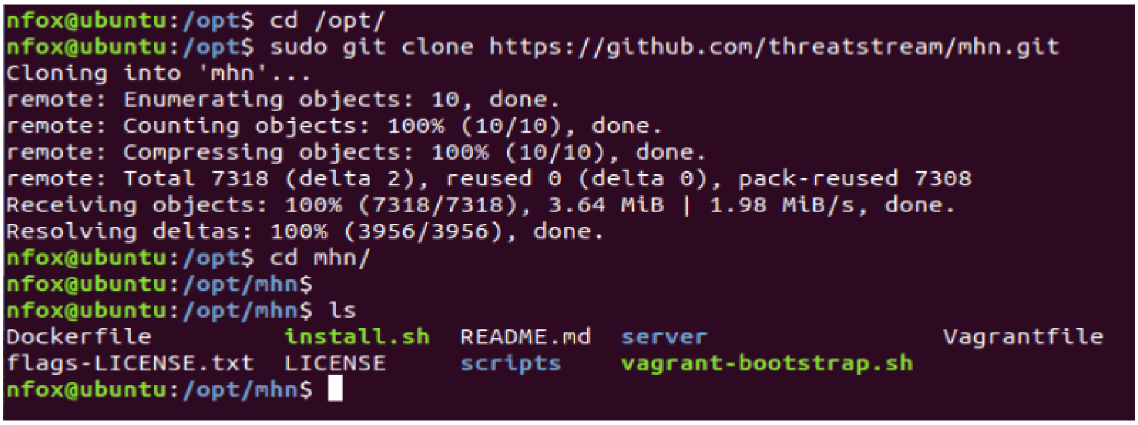
it is very simple to install MHN. We test it on Ubuntu 18.04 LTS (64 bits) with OpenSSH only. The installation steps are as follows:

Install Git

```
# on Debian or Ubuntu
$ sudo apt install git -y
```

Install MHN

```
$ cd /opt/
$ sudo git clone https://github.com/pwnlandia/mhn.git
$ cd mhn/
```



```
nfox@ubuntu:/opt$ cd /opt/
nfox@ubuntu:/opt$ sudo git clone https://github.com/threatstream/mhn.git
Cloning into 'mhn'...
remote: Enumerating objects: 10, done.
remote: Counting objects: 100% (10/10), done.
remote: Compressing objects: 100% (10/10), done.
remote: Total 7318 (delta 2), reused 0 (delta 0), pack-reused 7308
Receiving objects: 100% (7318/7318), 3.64 MiB | 1.98 MiB/s, done.
Resolving deltas: 100% (3956/3956), done.
nfox@ubuntu:/opt$ cd mhn/
nfox@ubuntu:/opt/mhn$
nfox@ubuntu:/opt/mhn$ ls
Dockerfile         install.sh  README.md  server                  Vagrantfile
flags-LICENSE.txt  LICENSE     scripts    vagrant-bootstrap.sh
nfox@ubuntu:/opt/mhn$
```

Figure 21 Raspberry Pi setup

Run the following script to complete the installation. While this script runs, you will be prompted for some configuration options. See below for how this looks.

```
$ sudo ./install.sh
```

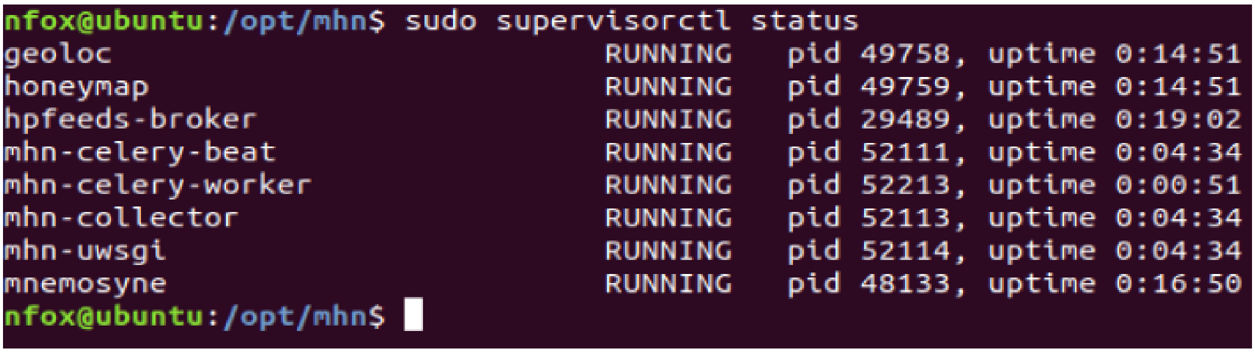
4.3 Configuration:

```
MHN Configuration

Do you wish to run in Debug mode?: y/n n

Superuser email: YOUR_EMAIL@YOURSITE.COM

Superuser password:

Server base url ["http://1.2.3.4"]:

Honeymap url ["http://1.2.3.4:3000"]:

Mail server address ["localhost"]:

Mail server port [25]:

Use TLS for email?: y/n n

Use SSL for email?: y/n n

Mail server username [""]:

Mail server password [""]:

Mail default sender [""]:

Path for log file ["mhn.log"]:
```

4.4 Running:

If the installation scripts ran successfully, you should have a number of services running on your MHN server. See below for checking these.

```
user@precise64:/opt/mhn/scripts$ sudo /etc/init.d/nginx status
 * nginx is running
user@precise64:/opt/mhn/scripts$ sudo /etc/init.d/supervisor status
 is running
user@precise64:/opt/mhn/scripts$ sudo supervisorctl status
geloc                          RUNNING    pid 31443, uptime 0:00:12
honeymap                       RUNNING    pid 30826, uptime 0:08:54
hpfeeds-broker                 RUNNING    pid 10089, uptime 0:36:42
mhn-celery-beat                RUNNING    pid 29909, uptime 0:18:41
mhn-celery-worker              RUNNING    pid 29910, uptime 0:18:41
mhn-collector                  RUNNING    pid 7872,  uptime 0:18:41
mhn-uwsgi                      RUNNING    pid 29911, uptime 0:18:41
mnemosyne                      RUNNING    pid 28173, uptime 0:30:08
```



```
nfox@ubuntu:/opt/mhn$ sudo supervisorctl status
geoloc                           RUNNING    pid 49758, uptime 0:14:51
honeymap                         RUNNING    pid 49759, uptime 0:14:51
hpfeeds-broker                   RUNNING    pid 29489, uptime 0:19:02
mhn-celery-beat                  RUNNING    pid 52111, uptime 0:04:34
mhn-celery-worker                RUNNING    pid 52213, uptime 0:00:51
mhn-collector                    RUNNING    pid 52113, uptime 0:04:34
mhn-uwsgi                        RUNNING    pid 52114, uptime 0:04:34
mnemosyne                        RUNNING    pid 48133, uptime 0:16:50
nfox@ubuntu:/opt/mhn$ 
```

Figure 22 Raspberry Pi setup 2

4.5 Troubleshooting:

If you find that the service is not working properly, you can use some commands and check some logs to determine where the problem is. The first command is supervisorctl, you can see that those processes are out of order, and those are running normally.

```
#!bash
supervisorctlv status

 #List the status of all processes
supervisorctl  restart [process|all]
 # Restart single or all processes
supervisorctl start [process|all]
 #Start a single process or all processes
supervisorctl stop [process|all]
 #STOP single process or all processes
```

4.6 MHN web interface:

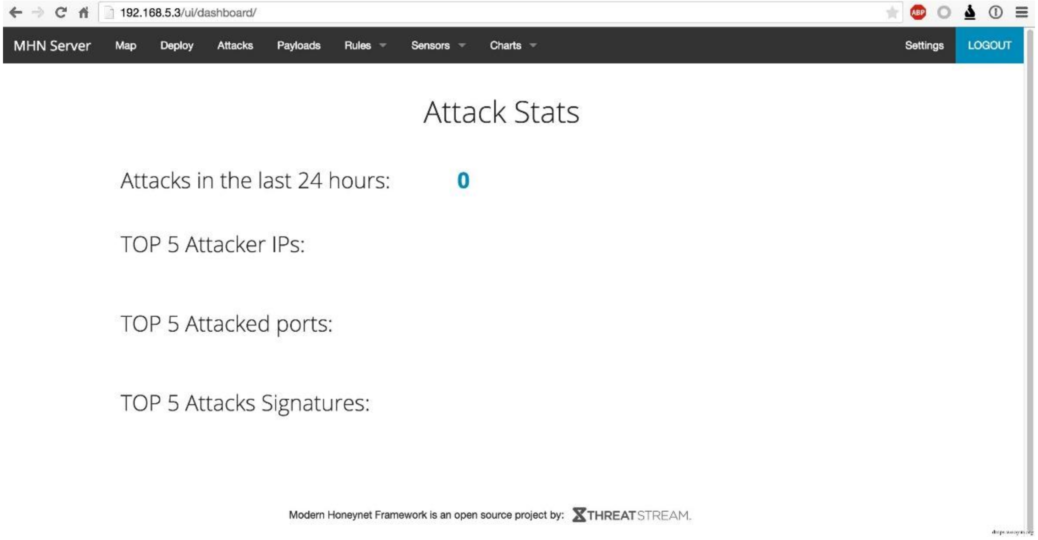
The web interface of MHN is very simple and clear. When you access the web profile for the first time, you need to enter the account name and password. After logging in successfully, you will see a summary page.

1. How many attacks have been attacked in the last 24 hours?

2. The number of attacks ranked in the top five IP

3. The port that is attacked by the top five ports

4. top 5 attack signature



Attack Stats

Attacks in the last 24 hours: 0

TOP 5 Attacker IPs:

TOP 5 Attacked ports:

TOP 5 Attacks Signatures:

Modern Honeynet Framework is an open source project by: THREATSTREAM.

Figure 23 MHN Dash

There are also some menu options to configure or get more attack details.

Map: View the global distribution of attackers' IPs

Deploy: Add, edit, and use honeypot deployment scripts

Attacks: a list of all attackers

Payloads: All attack payloads, there are only three honeypots that can collect payloads (snort, dionaea.glastofp)

Rules: all snort and suricata rules

Seneors: related records for installing honeypot node operations

Settings: Settings for the MHN service

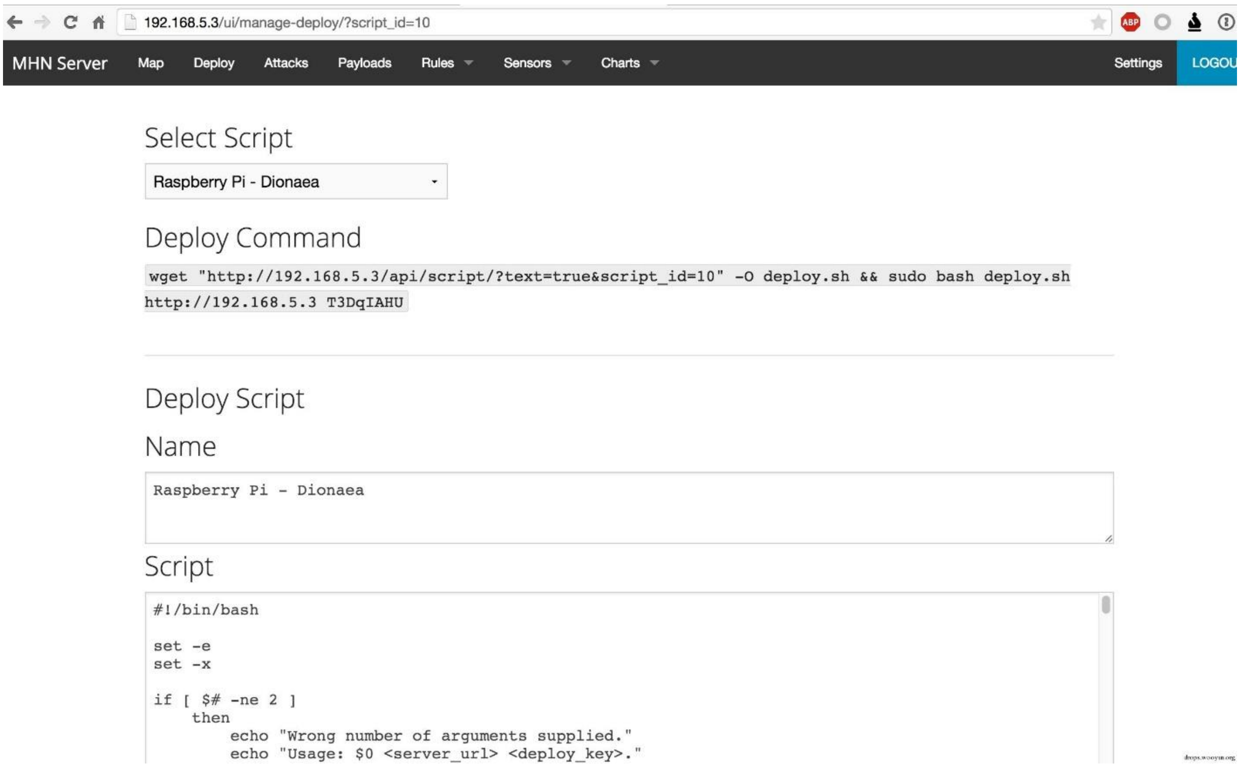
The global distribution map of the attack source, running on port 3000, can be seen without authentication, and can be controlled by ACL open control.

4.7 Installing Honeypot Nodes:

Installing honeypot nodes is easy. All nodes are based on the same platform. There are corresponding installation scripts for each honeypot on the MHN, so it is very easy to install and only requires one server to install.

It is recommended to use ssh to access. If not, please install ssh, modify the corresponding 22 ports, and add firewall protection.

The defined installation script is in the web interface of the MHN server. There are all honeypot installation scripts under the "Deploy" option



Select Script

Raspberry Pi - Dionaea

Deploy Command

```
wget "http://192.168.5.3/api/script/?text=true&script_id=10" -O deploy.sh && sudo bash deploy.sh
http://192.168.5.3 T3DqIAHU
```

Deploy Script

Name

Raspberry Pi - Dionaea

Script

```
#!/bin/bash

set -e
set -x

if [ $# -ne 2 ]
    then
        echo "Wrong number of arguments supplied."
        echo "Usage: $0 <server_url> <deploy_key>."
```

Figure 24 MHN Deploy

After the installation is complete, you can check it with the following command.

#!bash

sudo netstat –tunlp

#production see the current network connection

supervisorctl status

#View the running status

We have probably got an overview of the general overview of mhn, how to install and how to deploy honeypot nodes, and some of these honeypots. If you know what you want to do, deploying a honeynet is very useful. You can get more accurate information from the attacker and then do defense.

In the deployment of honeypot nodes, the deployment is combined as much as possible according to the specific situation, so that the attacker can spend more time, get more information from it, and get more response time.

For example, snort, Glastopf, Dionaea, and kippo should be tested before deployment to avoid unexpected problems in the real environment.

Personal opinion: I have done a similar thing, but it is a bit rough compared to this. There are many kinds of open source honeypots supported by MHN, which basically covers all existing open source honeypots, which can be based on specific business. Scenes to combine. It is also possible to do secondary development according to the specific scenario, because the data returned is somewhat simple. If it is placed in the internal network, the alarm function is necessary. The honeypot in the internal network is only to delay the progress of the attack, and to find the intrusion in time, thus cutting off the entrance.

Chapter 5
Testing

5.1 Attacking:

Attacker will attack the network using the following steps:

- Network Scanning: Online or Offline
- Port Scanning: what is Service on ports Up
- Service Scanning: Attack any Service

Now, through the Kali-Attacker system, we will install the Nmap tool:

```
kali@attacker:~$ sudo apt install nmap

[sudo] password for kali:

Reading package lists... Done

Building dependency tree

Reading state information... Done

Check the network in which the servers are located 3.23.23.233/24
```

The command sP -nmap: It will send a set of Ping requests to determine which devices are open or closed

```
kali@attacker:~$ nmap -sP 200.50.30.0/24

Starting Nmap 7.80 ( https://nmap.org ) at 2020-05-05 08:59 EDT

mass_dns: warning: Unable to determine any DNS servers. Reverse DNS
is disabled. Try

using --system-dns or specify valid servers with --dns-servers

Nmap scan report for 200.50.30.1

Host is up (0.022s latency).

Nmap scan report for 200.50.30.2

Host is up (0.013s latency).

Nmap scan report for 200.50.30.14

Host is up (0.24s latency).

Nmap scan report for 200.50.30.15
```

```
Host is up (0.60s latency).

Nmap done: 256 IP addresses (4 hosts up) scanned in 91.82 seconds
```

Now, check the ports for all devices with which online communication is available

```
kali@attacker:~$ sudo nmap -sS -A 200.50.30.1

Starting Nmap 7.80 ( https://nmap.org ) at 2020-05-05 08:54 EDT

mass_dns: warning: Unable to determine any DNS servers. Reverse DNS is disabled. Try

using --system-dns or specify valid servers with --dns-servers

Nmap scan report for 200.50.30.1

Host is up (0.060s latency).

All 1000 scanned ports on 200.50.30.1 are closed

Warning: OSScan results may be unreliable because we could not find at least 1 open and 1

closed port

Aggressive OS guesses: Cisco 1812, 3640, or 3700 router (IOS 12.4) (95%), Cisco DOCSIS

cable modem termination server (IOS 12.1) (95%), Cisco Catalyst 3560 or 6500-series

switch (IOS 12.1 - 12.2) (95%), Cisco ASR 1002 router (94%), Cisco SOHO 97 ADSL

router (94%), Cisco uBR10012 broadband router (94%), Cisco 1841 router (IOS 12) (94%),

Cisco 1841 router (IOS 12.4) (94%), Cisco 2801 router (IOS 12.4) (94%), Cisco 7600 router

(IOS 12.2) (94%)

No exact OS matches for host (test conditions non-ideal).

Network Distance: 3 hops

TRACEROUTE (using port 199/tcp)

HOP RTT ADDRESS

1 6.85 ms 192.168.100.1
```

```
2 67.78 ms 140.90.60.1

3 37.31 ms 200.50.30.1

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 9.15 seconds

kali@attacker:~$
```

All 1000 scanned ports on 200.50.30.1 are closed

```
kali@attacker:~$ sudo nmap -sS -A 200.50.30.2

Starting Nmap 7.80 ( https://nmap.org ) at 2020-05-05 08:55 EDT

mass_dns: warning: Unable to determine any DNS servers. Reverse DNS
is disabled. Try

using --system-dns or specify valid servers with --dns-servers

Nmap scan report for 200.50.30.2

Host is up (0.019s latency).

All 1000 scanned ports on 200.50.30.2 are closed

Too many fingerprints match this host to give specific OS details

Network Distance: 2 hops

TRACEROUTE (using port 8888/tcp)

HOP RTT ADDRESS

1 3.67 ms 192.168.100.1

2 34.38 ms 200.50.30.2

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 12.24 seconds

kali@attacker:~$
```

All 1000 scanned ports on 200.50.30.2 are closed

Note:

The command A -sS -nmap is considered one of the best and most used types in the process of checking ports, and it is faster

Technical and very secure because it can hide from the Firewall and the connection is not recorded in the Files Log because it does not

Three Way Handshakes

```
kali@attacker:~$ sudo nmap -sS -A 200.50.30.14

Starting Nmap 7.80 ( https://nmap.org ) at 2020-05-05 08:56 EDT

mass_dns: warning: Unable to determine any DNS servers. Reverse DNS
is disabled. Try

using --system-dns or specify valid servers with --dns-servers

Nmap scan report for 200.50.30.14

Host is up (0.036s latency).

Not shown: 966 closed ports

PORT STATE SERVICE VERSION

21/tcp open ftp?

23/tcp open telnet?

25/tcp open smtp?

|_smtp-commands: Couldn't establish connection on port 25

42/tcp open nameserver?

80/tcp open http?

|_http-title: It works!

143/tcp open imap?

443/tcp open tcpwrapped

445/tcp open microsoft-ds?

554/tcp open rtsp?

|_rtsp-methods: ERROR: Script execution failed (use -d to debug)

587/tcp open submission?
```

```
|_smtp-commands: Couldn't establish connection on port 587
617/tcp open sco-dtmgr?
1023/tcp open netvenuechat?
1025/tcp open NFS-or-IIS?
1111/tcp open lmsocialserver?
2103/tcp open zephyr-clt?
2105/tcp open eklogin?
2107/tcp open msmq-mgmt?
2967/tcp open symantec-av?
2968/tcp open enpp?
3128/tcp open squid-http?
3268/tcp open globalcatLDAP?
3372/tcp open msdtc?
6129/tcp open unknown
8080/tcp open tcpwrapped
9999/tcp open abyss?
38292/tcp open landesk-cba?
Network Distance: 3 hops
Host script results:
|_smb2-time: Protocol negotiation failed (SMB2)
TRACEROUTE (using port 199/tcp)
HOP RTT ADDRESS
1 11.19 ms 192.168.100.1
2 21.58 ms 140.90.60.1
3 31.49 ms 200.50.30.14
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 500.39 seconds

kali@attacker:~$
```

Honeypot-Amun responded to Attacker to trick that he has open ports and services available

For communication, but in fact they are fake ports and services that do not represent any danger

```
kali@attacker:~$ sudo nmap -sS -A 200.50.30.15

Starting Nmap 7.80 ( https://nmap.org ) at 2020-05-05 09:08 EDT

mass_dns: warning: Unable to determine any DNS servers. Reverse DNS is disabled. Try

using --system-dns or specify valid servers with --dns-servers

Nmap scan report for 200.50.30.15

Host is up (0.039s latency).

Not shown: 999 closed ports

PORT STATE SERVICE VERSION

22/tcp open ssh OpenSSH 6.7p1 Ubuntu 5ubuntu1.3 (Ubuntu Linux)

Network Distance: 3 hops

Service Info: OS: Linux; CPE: cpe:/o:linux:linux_kernel

TRACEROUTE (using port 587/tcp)

HOP RTT ADDRESS

1 3.95 ms 192.168.100.1

2 54.78 ms 140.90.60.1

3 24.33 ms 200.50.30.15

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 40.64 seconds
```

Honeypot-SSH responded to the Attacker to trick it into having an SSH service available for the connection but

In fact, it is a fake service that lures the attacker to keep him away from important systems, and when he is detected

Immediately, we take the necessary measures to prevent him from trying to connect to the network again, and his IP is added to the Firewall in the Block LIST.

```
kali@attacker:~/Documents$ ls

Passwords.txt Usernames.txt

kali@attacker:~/Documents$

kali@attacker:~/Documents$ nmap --script ssh-brute --script-args

userdb=./Usernames.txt,passdb=./Passwords.txt 200.50.30.15 -p 22

Starting Nmap 7.80 ( https://nmap.org ) at 2020-05-05 10:00 EDT

NSE: [ssh-brute] Trying username/password pair: admin:admin

NSE: [ssh-brute] Trying username/password pair: administrator:administrator

NSE: [ssh-brute] Trying username/password pair: toor:toor

NSE: [ssh-brute] Trying username/password pair: root:root

NSE: [ssh-brute] Trying username/password pair: passw@rd:passw@rd

Nmap scan report for 200.50.30.15

Host is up (0.038s latency).

PORT STATE SERVICE

22/tcp open ssh

| ssh-brute:

|_ Statistics: Performed 6823 guesses in 901 seconds, average tps: 7.4

Nmap done: 1 IP address (1 host up) scanned in 902.43 seconds

kali@attacker:~/Documents$
```

Attacker was able to obtain the username and password of the honeypot server, which is completely isolated from the network

The real server, i.e. it is a fake system, there is no danger on the network because the Attacker will not benefit anything

He will drain his energy, resources and time into a fictitious place where there is no important information and it is all fictitious to shadow him

and extract as much information from it

Honeypot-SSH

```
kali@attacker:~/Documents$ ssh root@200.50.30.15
root@200.50.30.15's password:
The programs included with the Debian GNU/Linux system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*/copyright.
Debian GNU/Linux comes with ABSOLUTELY NO WARRANTY, to the extent
permitted by applicable law.
root@server:~#
root@server:~# cd ..
root@server:/# ls
bin boot dev etc home initrd.img lib
lost+found media mnt opt proc root run
sbin selinux srv sys tmp usr var
vmlinuz
root@server:/# cd home/
root@server:/home# ls
richard
root@server:/home# cd richard/
root@server:/home/richard# ls
root@server:/home/richard#
root@server:/home/richard# cd ..
root@server:/home# cd ..
root@server:/# cd root/
```

```
root@server:~# ls
root@server:~#
root@server:~# whoami
root
root@server:~#
```

They are all fake files that make the Attacker think that he is connected to a real server, where he can create files or delete them

files in Server as in the following illustration:

Add a file named SPUData.DB using the following command:touch SPUData.DB

```
root@server:~# cd ..
root@server:/# ls
bin boot dev etc home initrd.img lib
lost+found media mnt opt proc root run
sbin selinux srv sys tmp usr var
vmlinuz
root@server:/#
root@server:/# touch SPUData.DB
root@server:/# ls
SPUData.DB bin boot dev etc
home initrd.img lib lost+found media
mnt opt proc root run
sbin selinux srv sys tmp
usr var vmlinuz
root@server:/#
```

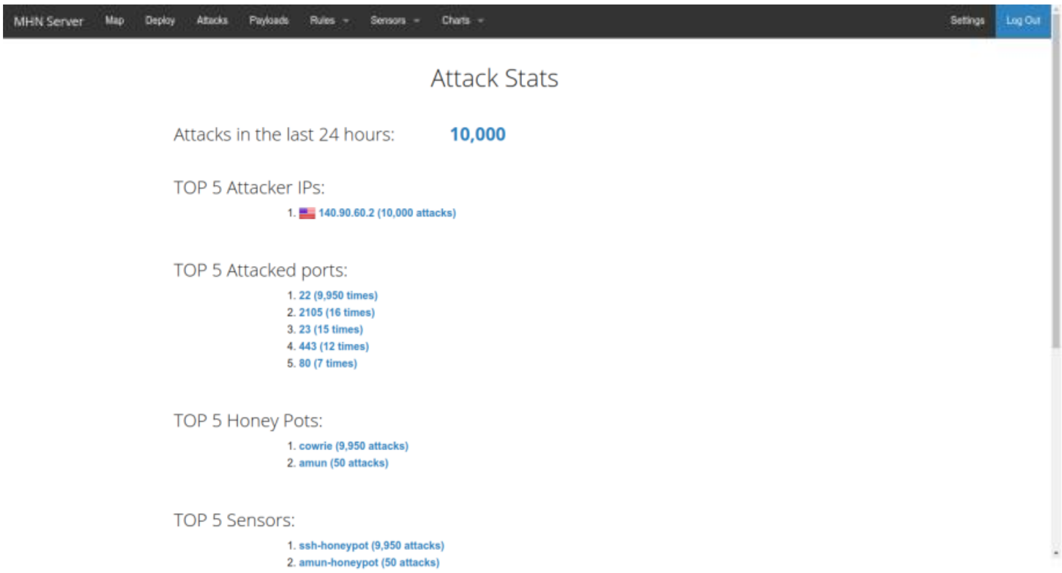
rm SPUData.DB will delete file

```
root@server:/# rm SPUData.DB
root@server:/#
root@server:/# ls
```

```
bin boot dev etc home initrd.img lib
lost+found media mnt opt proc root run
sbin selinux srv sys tmp usr var
vmlinuz
root@server:/#
root@server:/# exit
Connection to 200.50.30.15 closed.
kali@attacker:~/Documents$
```

5.2 Results:

After the hacking, we enter to Server-MHN to see the results



Attack Stats

Attacks in the last 24 hours: **10,000**

TOP 5 Attacker IPs:

1. 🇺🇸 140.90.60.2 (10,000 attacks)

TOP 5 Attacked ports:

1. 22 (9,950 times)
2. 2105 (16 times)
3. 23 (15 times)
4. 443 (12 times)
5. 80 (7 times)

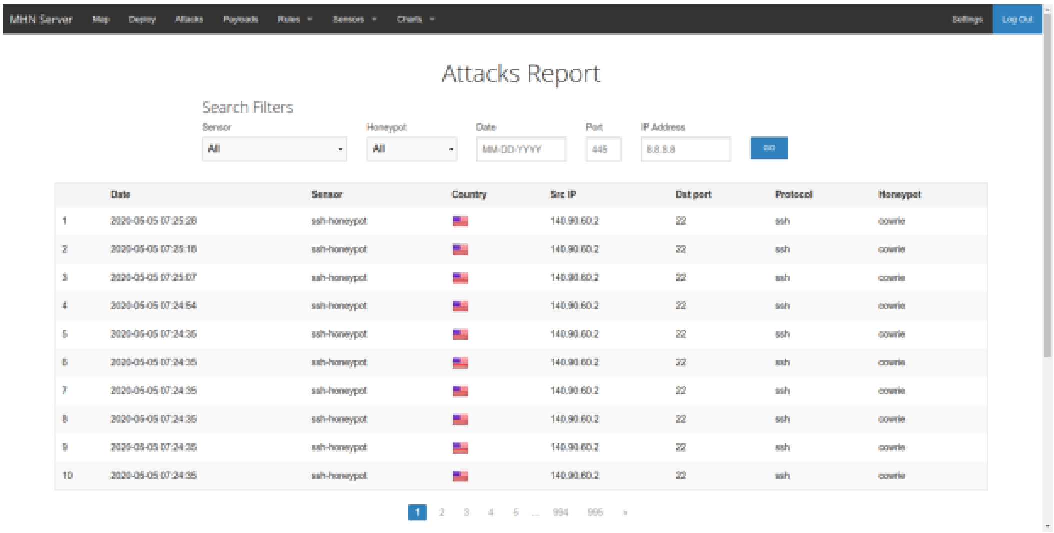
TOP 5 Honey Pots:

1. cowrie (9,950 attacks)
2. amun (50 attacks)

TOP 5 Sensors:

1. ssh-honeypot (9,950 attacks)
2. amun-honeypot (50 attacks)

This is the last record detail



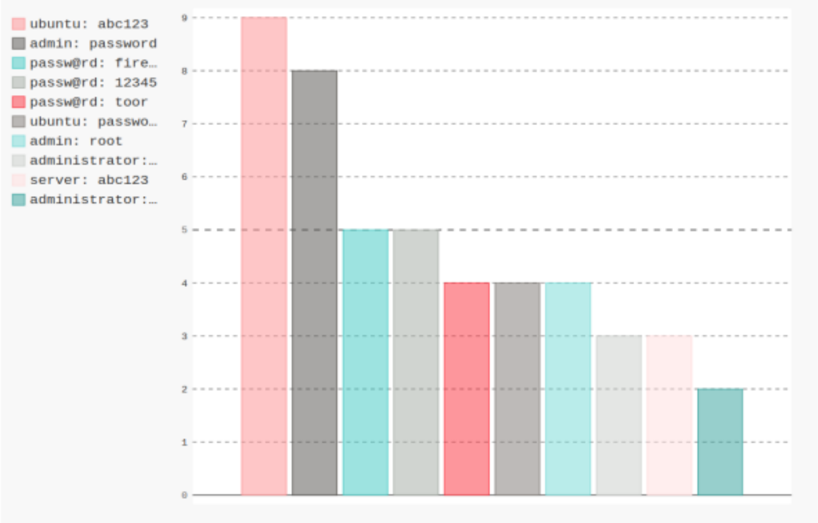
Attacks Report

Search Filters

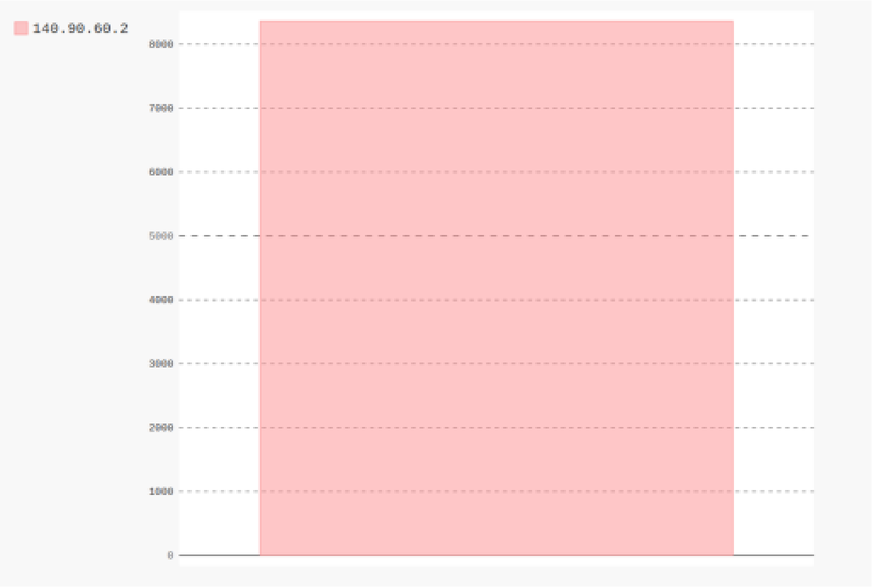
Sensor: All
Honeypot: All
Date: MM-DD-YYYY
Port: 445
IP Address: 8.8.8.8

	Date	Sensor	Country	Src IP	Dst port	Protocol	Honeypot
1	2020-05-05 07:25:28	ssh-honeypot	🇺🇸	140.90.60.2	22	ssh	cowrie
2	2020-05-05 07:25:18	ssh-honeypot	🇺🇸	140.90.60.2	22	ssh	cowrie
3	2020-05-05 07:25:07	ssh-honeypot	🇺🇸	140.90.60.2	22	ssh	cowrie
4	2020-05-05 07:24:54	ssh-honeypot	🇺🇸	140.90.60.2	22	ssh	cowrie
5	2020-05-05 07:24:35	ssh-honeypot	🇺🇸	140.90.60.2	22	ssh	cowrie
6	2020-05-05 07:24:35	ssh-honeypot	🇺🇸	140.90.60.2	22	ssh	cowrie
7	2020-05-05 07:24:35	ssh-honeypot	🇺🇸	140.90.60.2	22	ssh	cowrie
8	2020-05-05 07:24:35	ssh-honeypot	🇺🇸	140.90.60.2	22	ssh	cowrie
9	2020-05-05 07:24:35	ssh-honeypot	🇺🇸	140.90.60.2	22	ssh	cowrie
10	2020-05-05 07:24:35	ssh-honeypot	🇺🇸	140.90.60.2	22	ssh	cowrie

This chart showing the most usernames and passwords used in hacking



This chart shows as the most hackers IP’s



Conclusion

Honeypots are part of a new kind of security technology, often called Deception Technology, that are a way to mitigate damage once the attacker has broken inside the network already, and are an increasingly important aspect of the modern enterprise network. It adds an additional layer of security on top of IDS, firewalls etc., and is a valuable source of information about attackers and impending attacks. This information can be used to further enhance the security of the network. Honeypots are still being actively researched, and in recent years, honeypot technology has been expanded to create world-wide networks of honeynets and honeyfarms, which are being put to use against various different kinds of malicious activity on the internet.

Previously, we conducted research on the application of Honeypot on an external device, we studied how it works as a software to convert it to Hardware, we have reached a result in simulating the subject via computer

In this project, we have implemented what we have studied on a hard piece and are running it

We will complete the project in the following phases.

References

[1] Cobb, M. (2021). *www.techtarget.com*. Retrieved from TechTarget.

[2] Franco, J. (2021). A Survey of Honeypots and Honeynets for.

[3] sakshiagarwal4. (2020). *www.geeksforgeeks.org*. Retrieved from Geeks for Geeks.

[4] Honeypots. (2018). In M. Turner, *Taking the Offensive in Network Security* (p. 13).

[5] Vishal Joshi, P. K. (2017). *Honeypot Based Intrusion Detection System.*

[6] Thalgott, D. A.-F. (2010). *Honeypots in Network Security.*

[7] GRIMES, R. A. (2005). *Honeypots for Windows.*