PUNE INSTITUTE OF COMPUTER TECHNOLOGY, PUNE

ACADEMIC YEAR: 2023-24

DEPARTMENT OF COMPUTER ENGINEERING DEPARTMENT

CLASS: B.E. **SEMESTER: I**

SUBJECT: LP-IV

ASSIGNMENT NO.	A7
TITLE	**Study of Honeypot**
PROBLEM STATEMENT /DEFINITION	Study of Honeypot
OBJECTIVE	1. study malicious behavior on the Internet and to identify potential bad actors—often by IP address. 2. Understand the honeypot security used to protect the company from attacks, they are implemented inside the production network to improve the overall security.
OUTCOME	Students will be able to - 1. understand the computer security-defense tool 2. learn to understand how computer network trap by attackers and gather data 3. learn steps to manage and prevent attacks.
S/W PACKAGES AND HARDWARE APPARATUS USED	Windows 10 (64-bit), Intel I5 4GB RAM 256 GB SSD, Setup HoneyPot on windows 10 using KFSensor
REFERENCES	1. https://www.diva-portal.org/smash/get/diva2:327476/fulltext01 2. https://www.youtube.com/watch?v=0WUaI2pNiPI 3. https://www.youtube.com/watch?v=ULgcOnelE6E (setup honeypot on windows 10)
STEPS	Refer to theory, algorithm, test input, test output
INSTRUCTIONS FOR WRITING JOURNAL	1. Date 2. Assignment no. 3. Problem definition 4. Learning objective 5. Learning Outcome 6. Concepts related Theory 7. Algorithm 8. Test cases 9. Conclusion/Analysis

Prerequisites:

Concepts related Theory:

1. INTRODUCTION-

A honeypot is a computer system that is set up to act as a decoy to lure cyber attackers, and to detect, deflect or study attempts to gain unauthorized access to information systems. Generally, it consists of a

computer, applications, and data that simulate the behavior of a real system that appears to be part of a network but is actually isolated and closely monitored. All communications with a honeypot are considered hostile, as there's no reason for legitimate users to access a honeypot. Viewing and logging this activity can provide an insight into the level and types of threat a network infrastructure faces while distracting attackers away from assets of real value. Honeypots can be classified based on their deployment (use/action) and based on their level of involvement.
Based on deployment, honeypots may be classified as:
1. Research honeypots
2.Production honeypots

1. Research honeypots

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.

2. Production honeypots

Production honeypots are used to protect the company from attacks, they are implemented inside the production network to improve the overall security. They are capturing a limited amount of information, mostly low interaction honeypots are used. Thus, security administrator watches the hacker's movements carefully and tries to lower the risks that may come from it towards the company. At this point, we will try to discuss and find out the risks of using production honeypots. Because while testing the security of the systems existing in an organization, unexpected actions may happen such as misusing other systems using honeypot features. If the network administrator is not aware of this problem, they put organization in a big trouble.

Spitzner L.(2002) claims that it is easier to break the honeypot phases into groups and refers that Bruce Schneier model is good for understanding the honeypots. He groups the security issues into several steps, which are prevention, detection and response.

2.1. Prevention

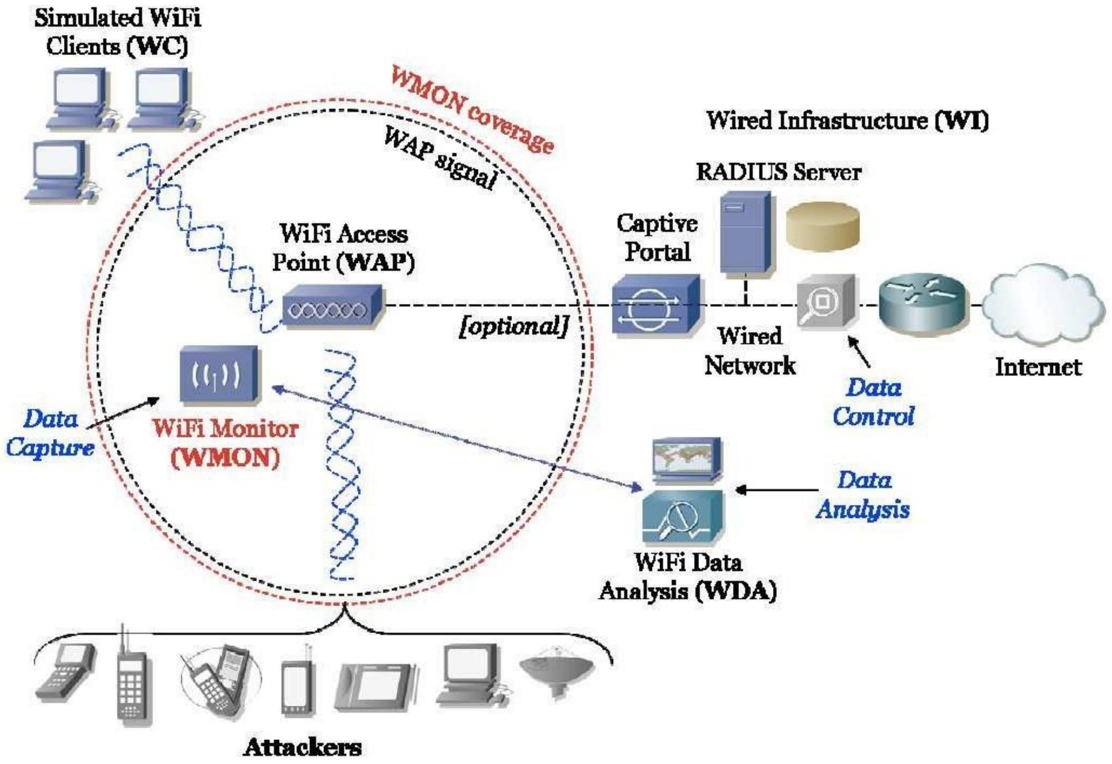
Prevention is the first thing to consider in our security model. As a definition, it means to prevent the hackers to hack the system. So, we will try not to allow them to access the system. There are many ways to do this in security. One can use firewall to control the network traffic and put some rules to block or allow it. Using authentication methods, digital certificates or having strong passwords are the most common and well-known security prevention techniques. There are also encryption algorithms that encrypt data. It is a good way to use it since it encrypts the messages and make them impossible to read.

The relation between using prevention and honeypot can be explained as following. If the hacker understands the company he is trying to hack is using honeypots and they are aware of today's security problems, it will make them think about it. It will be confusing and scary for a hacker. Even if a company uses the methods that we discussed in the first paragraph in order to stay secure, it is still good to have honeypot in an organization since security issues are concerned and handled professionally. As the security is very significant, it is always good to be conscious. There is no tolerance when there is a problem, it can give a lot of damage to any company. Because every company has private and important data, there is a need to protect the data from intruders.

2.2 Detection

Detection is the act of detecting any malicious activity in the system. We are assuming that prevention did not work so one way or another, a hacker compromised the system. There are some wa ys for detecting those attacks. The well-known detection solution is Network IntrusionDetection Systems. This technology will help users to know if the network is compromised, but it will not prevent hackers from attacking the system. For companies, such detection systems are expensive. At this point, honeypots are valuable to monitor the activity.

2.3 Honeypot project: Honeyspot is the well known wireless honeypot project supported by Spanish Honeynet Project. The term comes from honeypot and hotspot. Basically, honeyspot was created to watch the hacker and his attacks towards the wireless network. Thus, the traffic that is through the honeyspot is considered as malicious. However, like any other honeypot structures, professional hackers may understand that it is not a real system. So, honeyspot should look as real as possible for the best results. The Honeyspot team would like to know the attack type, intruder's ideas, tools, logic, and his approaches. It is very beneficial to get as much information as possible to identify the attack and help to understand any other further attacks in the future. From all these information, honeyspot can answer the questions about the security flaws in WEP wireless connections and attacks targeted to it. IP address spoofing, web session hacking, MAC address spoofing can be identified. It can also answer the special approaches to hack wireless clients. Thanks to all these information, more secure systems can be created



.

The figure is showing the architecture used in honeyspot.WAP which is in the middle is the wireless access point. It gives wireless networks to the users for internet connection. Attackers can connect to it.

WC (Wireless Client) are the devices that are able to connect to the honeyspot network. The purpose of this is to create traffic that is flooding through the network. It is to show the attacker that there is traffic. The real traffic makes sense for the attacker as it looks like a real system. Furthermore, attacker can attack on this stage by using his monitoring tools.

WMON is wireless monitor module. This module captures the traffic in order to have the network traffic information. It helps to understand the attacks, so this module is quite significant at this point.

WDA is wireless data analysis module. This module works with WMON as a team. As WMON is supposed to capture the traffic, there must be a module which is responsible for examining it. Therefore, WMON has the records and saves them in order to send them to WDA for obtaining the information.

WI module is wired structure and it is up to the person to put it in the structure or not. If you wish to create a wired network in your structure, it is also possible. So, WI module gives you a different aspect to your structure.

3. Advantages of honeypots

Here are the reasons why we should choose honeypots according to Mokube I. and Adams M. (2007):

a. Honeypots can capture attacks and give information about the attack type and if needed, thanks to the logs, it is possible to see additional information about the attack.

b. New attacks can be seen and new security solutions can be created by looking at them.
c. More examinations can be obtained by looking at the type of the malicious behaviors. It helps to understand more attacks that may happen.
d. Honeypots are not bulky in terms of capturing data. They are only dealing with the incoming malicious traffic. Therefore, the information that has been caught is not as much as the whole traffic.
e. Focusing only on the malicious traffic makes the investigation far easier. Therefore, this makes honeypots very useful.
f. They are simple to understand, to configure and to install. They do not have complex algorithms. There is no need for updating or changing some things.
g. As honeypots can capture anything malicious, it can also capture new tools for detecting attacks too. It gives more ideas and deepness of the subject proving that it is possible to discover different point of views and apply them for our security solutions.

4. Disadvantages of honeypots

We are continuing with Mokube I. & Adams M. (2007)'s studies:

a. We can only capture data when the hacker is attacking the system actively. If he does not attack the system, it is not possible to catch information. If there is an attack occurring in another system, our honeypot will not be able to identify it. So, attacks not towards our honeypot system may damage other systems and cause big problems.
b. There is a fingerprinting disadvantage of honeypots. It is easy for an experienced hacker to understand if he is attacking a honeypot system or a real system. Fingerprinting allows us to distinguish between these two. It is not a wanted result of our experiment.
c. The honeypot may be used as a zombie to reach other systems and compromise them.This can be very dangerous.

4. STUDY OF PRACTICAL IMPLEMENTATION

We are starting with a low interaction honeypot and then continue on a middle level of interaction to finally conclude with a high level of interaction.

4.1 Starting to honeypots

We started with Honeyd as a low level interaction honeypot and then we will move on to medium level interaction honeypots. Every honeypot has specific and different attitudes. We will explain them one by one.

4.2 Starting with low level interaction honeypots : Honeyd

Low interaction honeypots are emulating the services of a real operating system. We started with deploying Honeyd. It is the most well known low level interaction honeypot.

We thought it is a good starting point; it is easy to configure and understand its logic. More explanation can be found as the following.

Therefore, as a starting point, we worked on Honeyd.

- Honeyd is developed by Niels Provos from University of Michigan and used mainly as a production honeypot.
- Honeyd is an open source solution and designed for Unix systems. Like the other low level interaction honeypots, there is no operating system installed in Honeyd. They are just some services running on it.
- It is configurable, so anyone can create their own services and decide which ports to open and listen as well.
- As hacker will not find any real computer with real operating system, the key point here is to configure a virtualized network stack.
- Honeyd basically captures TCP traffic that hacker is generating. On Honeyd , we configured a template which looks like a real system with Windows XP operating system, and IP address.
- Thus, when the hacker establishes the connection with Honeyd, Honeyd generates fake messages and return them to the hacker to fool the hacker.
- Honeyd is able to create many fake IP addresses and simultaneously run them for hackers trying to attack the machine. Unlike other low interaction honeypots, Honeyd can also handle several different operating systems at the same time.
- There are two other major advantages to use Honeyd.
    - First of all, it can capture the connection on any port. This utility makes detection of the network traffic easier and better.
    - Second advantage of it is that being able to change services. Following figure clearly explains the process of honeypot. As it is seen processes are designed to create some returns which are created by personality engine to make it look good and logical according to our template for Honeyd.

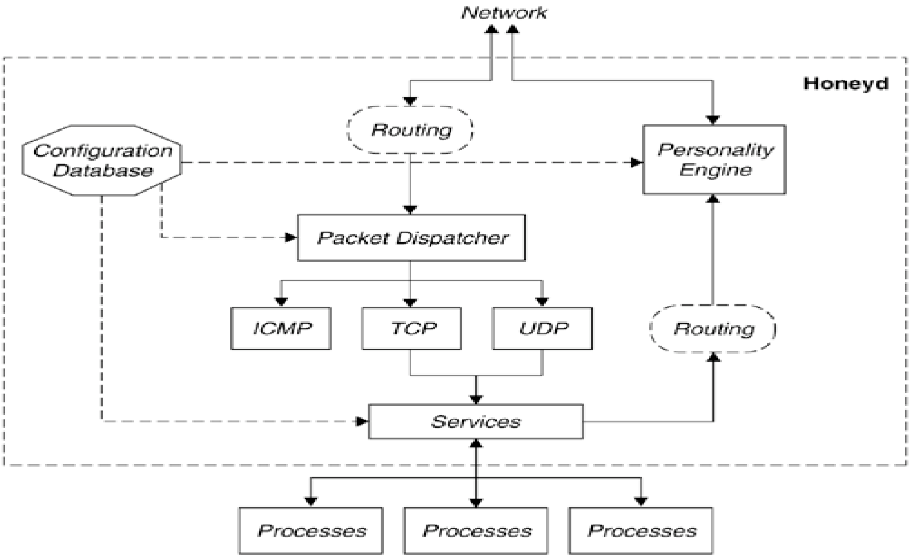


Figure 4.1 Honeyd structure from *Virtual Honeypots: From Botnet Tracking to Intrusion Detection, Provos* N., Holz T. (2007)

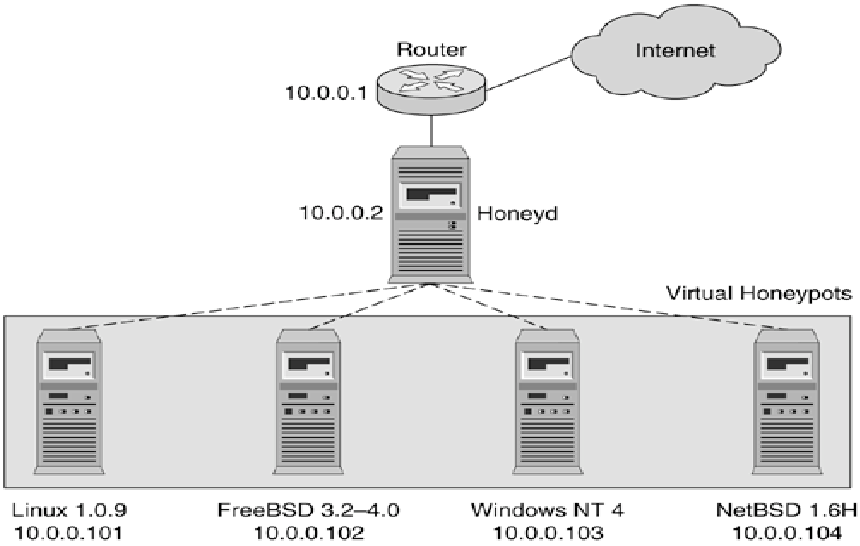


Figure 4.2 Honeyd virtual honeypots from *Virtual Honeypots: From Botnet Tracking to Intrusion Detection, Provos* N., Holz T. (2007)

4.3 Continuing with medium level of interaction honeypots : Nepenthes

Medium level of interaction honeypots are mostly used on learning new threats for the users that is on internet such as worms and new viruses and being aware of them. Thus, these kinds of honeypots are used to detect those malware and botnet. Their simulation algorithm is based on virtualizing logical responses for incoming requests. They are not virtualizing the whole operating system needs and they are not simulating application protocols in detail. When the request arrives to the medium interaction honeypot, that message is watched and examined, and fake responses are created. We will explain the most well known medium level of interaction honeypots as following.

As we stated it above, Nepenthes is developed with Mwcollectd. According to Maggi F. and Zanero S. (2008), Nepenthes is working on five modules which are vulnerability, shellcode parsing, fetching, logging and submission modules. Vulnerability function allows us to create vulnerable services. Shellcode parsing takes the payload and examine on it and get information about the extracted data. If any important data is found to examine, then fetch functionality gets the malware and submits to the center part. You can log the information that you have by using the logging function of Nepenthes. Nepenthes is used for mostly malicious software that are spreading over internet automatically. Figure 4.3 is explaining Nepenthes architecture.

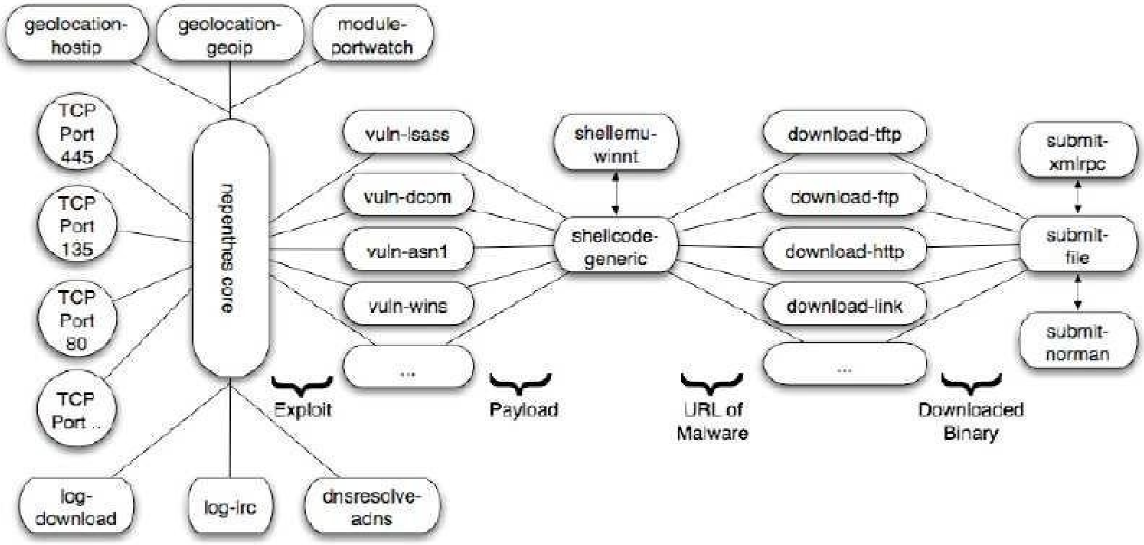


Figure 4.3 : Nepenthes architecture from Maggi F. and Zanero S. (2008)

We decided to install this software on Ubuntu operating system. The installation itself is simple as Nepenthes is present on the ubuntu repositories. To install the software, we used the command *apt-get install Nepenthes.* Once the process finished, we had to customize the configuration files. All of them are included in the folder /etc/Nepenthes/. The first one to check is Nepenthes.conf, it includes all the basic configuration on the software. The other ones are:

- submit-file.conf in which it is possible to set in which directory the downloaded malware will be stored.
- submit-norman.conf in which we set our email address. Norman Sandbox is an automated malware analyser. When Nepenthes will download a new malware, it will automatically be submitted to norman sandbox and the report will be send to our email address.
- log-download.conf in which we set the path of the logs for downloaded malware and malware submissions.

One of the strength of Nepenthes is that it emulates FTP and TFTP servers so the bot/attacker can upload the malicious software to the honeypot which allows the forensic party to analyze the threat.

After finishing the configuration, the last step is to put the honeypot on a DMZ and wait for the results.

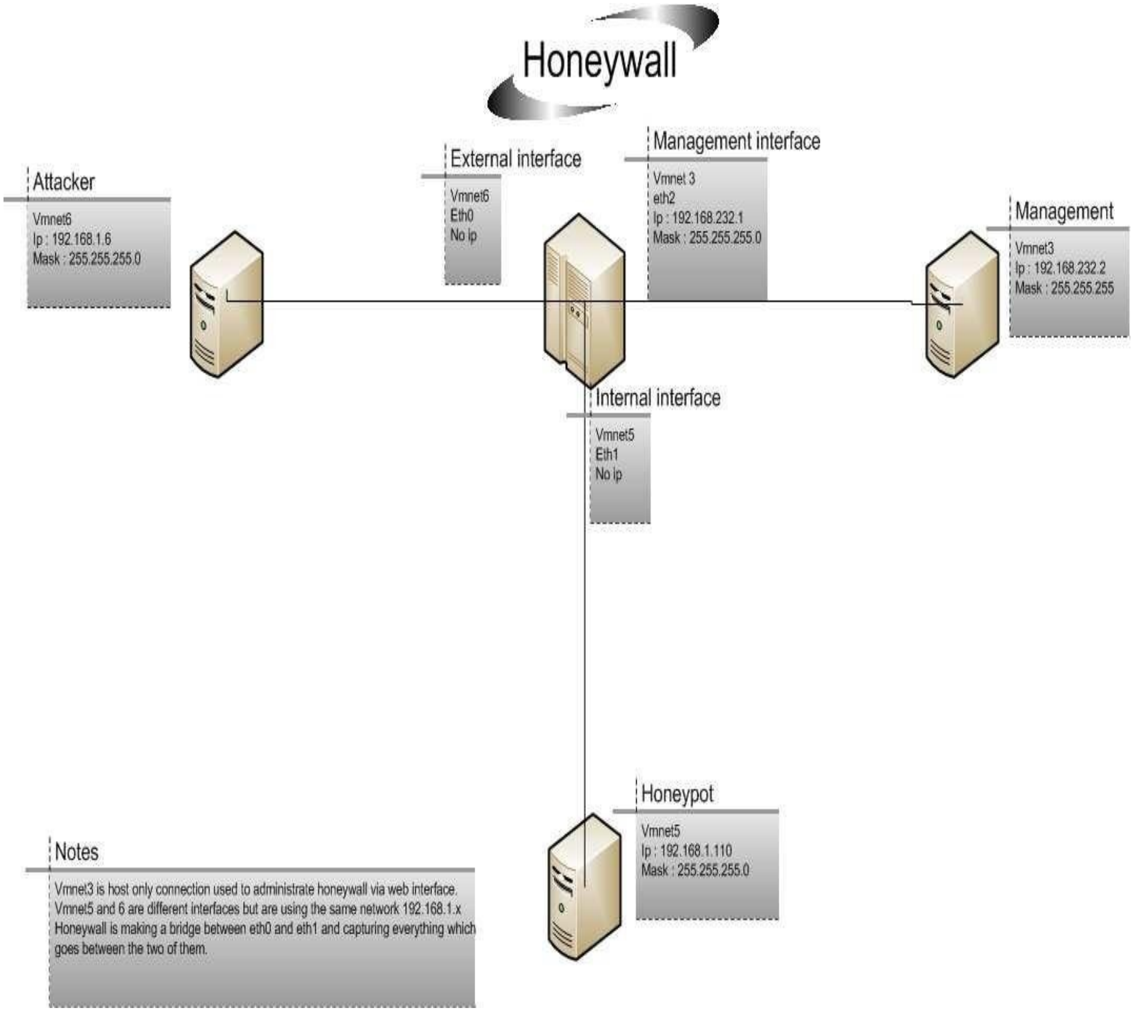
An experience lead by Jean-Michel Phillipe in 2007 for 192 days using Nepenthes showed the following results:

- One malware downloaded every 17 seconds.
- More or less 10 new malware per day and only a few detected as malicious.
- Almost only malware targeting Windows operating systems.

4.4 High level of interaction honeypots : Honeywall

Our last experiment will be based on high level of interaction honeypots. As we examined two types of interaction honeypots, we will move on further on implementation. Both low level and medium level of interaction honeypots offer more or less the same things. Services are emulated and you have restrictions. Thanks to several network monitoring tools, it is easy to understand what is going on throughout the traffic and understand that they are honeypots. Now, with high level of interaction honeypots, we will discover more on honeypots and with real operating system we will be able to catch more useful and interesting findings. Hackers will be freer with a real system without restrictions. Implementation will be time consuming and complicated. Our aim is to investigate if it is difficult to hack it and understand its structure, and detecting possible problems related to it and finding appropriate solutions or actions. Now, we will get to know available high level of interaction honeypot products currently exist in the market.

In high level of interaction honeypots part, we experimented on Honeywall. Our implementation is shown on the figure 4.4 below.



Vmnet3 is host only connection used to administrate honeywall via web interface.
Vmnet5 and 6 are different interfaces but are using the same network 192.168.1.x
Honeywall is making a bridge between eth0 and eth1 and capturing everything which goes between the two of them.

Figure 4.4 Our Honeywall implementation

For our experimentation, we decided to use virtual machines. It allows us to create our network without much physical equipment. We created three virtual machines:

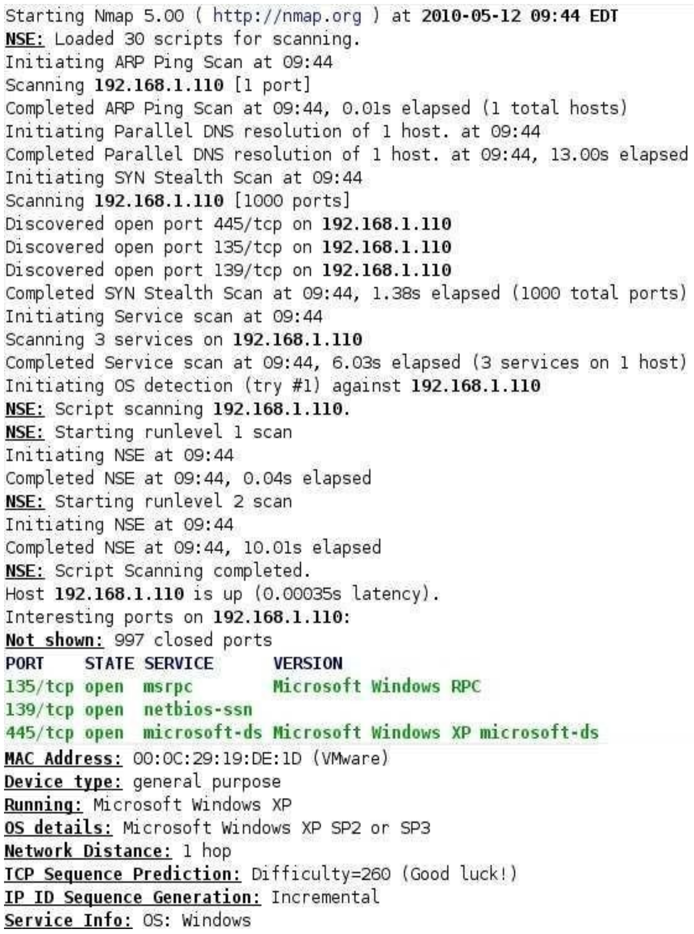
-One hosting Honeywall

-One hosting the attacker machine, we installed backtrack4 on it

-One hosting our honeypot, an unpatched Windows XP sp3

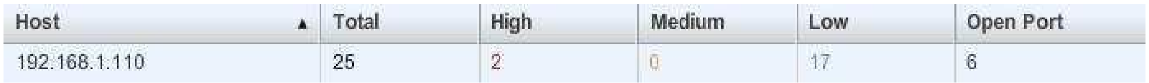
- The Honeywall has three virtual network interfaces. eth0 is bridged to vmnet6, it is the attacker side.
- Eth1 is bridged on vmnet5, it is the honeypot side. Finally, eth2 is bridged to vmnet3, it is the management administration, and it allows remote administration of Honeywall.
- We did not create a virtual machine for the management part, we used a host only connection with the computer hosting all the virtual machines, this way we did not need another virtual machine just to administrate the Honeywall.
- Eth0 and eth1 are making a bridge, thus none of these interfaces have a network address making these two interfaces invisible. Honeywall does not give a choice on that part, but it is the best way to keep it undetected.
- Once we managed to install and run all the virtual machines properly, we used the attacker machine in order to hack the honeypot. The first step is to detect any security flow that we could exploit. In order to do that, we used two tools very known: Nmap and Nessus.
- Nmap is a port scanner offering a lot of options (type of scan, level of detail about the target, etc…). The result of our scan is the figure 4.5 below:

Many precious information have been gathered, thanks to the scan we could identify the operating system running on the target (Windows XP sp2 or sp3) and which port are open (135, 139 and 445). We also obtained some information about the network card used, we used the default value for the MAC address of the machine, so Nmap detected it as a



```
Starting Nmap 5.00 ( http://nmap.org ) at 2010-05-12 09:44 EDT
NSE: Loaded 30 scripts for scanning.
Initiating ARP Ping Scan at 09:44
Scanning 192.168.1.110 [1 port]
Completed ARP Ping Scan at 09:44, 0.01s elapsed (1 total hosts)
Initiating Parallel DNS resolution of 1 host. at 09:44
Completed Parallel DNS resolution of 1 host. at 09:44, 13.00s elapsed
Initiating SYN Stealth Scan at 09:44
Scanning 192.168.1.110 [1000 ports]
Discovered open port 445/tcp on 192.168.1.110
Discovered open port 135/tcp on 192.168.1.110
Discovered open port 139/tcp on 192.168.1.110
Completed SYN Stealth Scan at 09:44, 1.38s elapsed (1000 total ports)
Initiating Service scan at 09:44
Scanning 3 services on 192.168.1.110
Completed Service scan at 09:44, 6.03s elapsed (3 services on 1 host)
Initiating OS detection (try #1) against 192.168.1.110
NSE: Script scanning 192.168.1.110.
NSE: Starting runlevel 1 scan
Initiating NSE at 09:44
Completed NSE at 09:44, 0.04s elapsed
NSE: Starting runlevel 2 scan
Initiating NSE at 09:44
Completed NSE at 09:44, 10.01s elapsed
NSE: Script Scanning completed.
Host 192.168.1.110 is up (0.00035s latency).
Interesting ports on 192.168.1.110:
Not shown: 997 closed ports
PORT    STATE SERVICE      VERSION
135/tcp open  msrpc        Microsoft Windows RPC
139/tcp open  netbios-ssn
445/tcp open  microsoft-ds Microsoft Windows XP microsoft-ds
MAC Address: 00:0C:29:19:DE:1D (VMware)
Device type: general purpose
Running: Microsoft Windows XP
OS details: Microsoft Windows XP SP2 or SP3
Network Distance: 1 hop
TCP Sequence Prediction: Difficulty=260 (Good luck!)
IP ID Sequence Generation: Incremental
Service Info: OS: Windows
```

Windows XP is known for its security holes, especially if not patched regularly. In order to check that, we used the security scanner Nessus. This tool is free to use on its home version, the professional version is charged. Nessus is able to detect and report any security problem for the target it is scanning. On the report page, a link is provided to install the right patch and protect the system. On the hacker point of view, it just highlights which exploits he can use. After the scan, we obtained the following result shown in the figure 4.6:

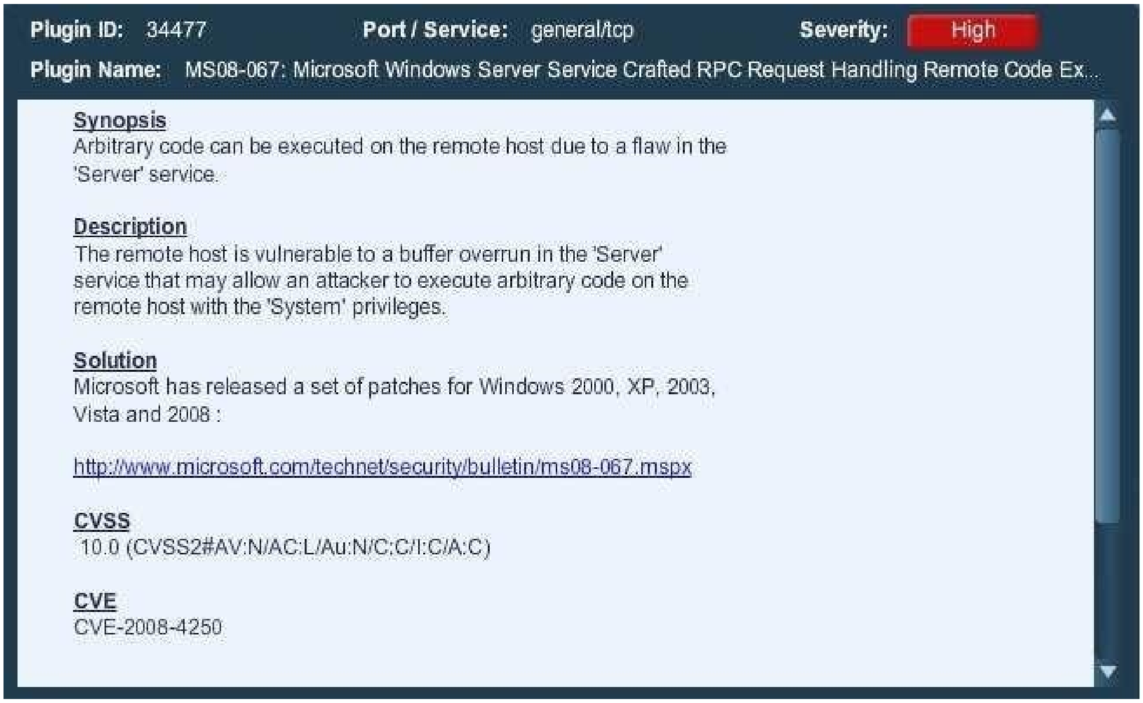


Host ▲	Total	High	Medium	Low	Open Port
192.168.1.110	25	2	0	17	6

Figure 4.6 Nessus result report

Nessus detected two critical problems, by opening the pane to take a closer look at it, we saw that the machine had the vulnerability ms08-067. Figure 4.7 is showing this vulnerability.

Figure 4.7 Nessus vulnerability report MS08-067



Plugin ID: 34477 **Port / Service:** general/tcp **Severity:** High

Plugin Name: MS08-067: Microsoft Windows Server Service Crafted RPC Request Handling Remote Code Ex...

Synopsis
Arbitrary code can be executed on the remote host due to a flaw in the 'Server' service.

Description
The remote host is vulnerable to a buffer overrun in the 'Server' service that may allow an attacker to execute arbitrary code on the remote host with the 'System' privileges.

Solution
Microsoft has released a set of patches for Windows 2000, XP, 2003, Vista and 2008 :

http://www.microsoft.com/technet/security/bulletin/ms08-067.mspx

CVSS
10.0 (CVSS2#AV:N/AC:L/Au:N/C:C/I:C/A:C)

CVE
CVE-2008-4250

As we can see on the screenshot, this vulnerability allows a hacker to execute code with System privileges. Now that we know what to exploit, we updated metasploit framework 3. This tool contains a huge library of exploits and payloads and can very easily hack into unprotected machines. Metasploit can be used via command line, or a user interface, or a web interface. We chose to use the web interface for its conviviality. In the research field for exploits, we wrote ms08-067 and after a few seconds the exploit was displayed. We selected it, and chose automated targeting (if not, we had to chose the operating system we were attacking). Then we chose the payload to use, in our case we wanted to create a new user on the machine who will have administrator privileges. The final step is to fill the ip address of the target and click on exploit. A command shell appeared to tell us that the exploit was successful. We checked on the honeypot the list of users, and the new user was here. This showed us how easy it is to hack an unprotected Windows XP. We used a simple payload to create a user, but we could have used a remote shell or a VNC session to fully control the computer.

If the attacker realizes what he is up against, he may want to hack into the Honeywall itself. Thanks to its architecture, the system is really hard to get into. The only way to take control over it is to have a physical access to the machine or to find a way into the management interface. A prudent administrator would most likely isolate that interface from the rest of the network to ensure its security. With this setup, Honeywall should be perfectly secure. However, if the administrator wishes to have a less restricted access to the remote administration of the system, it would also give an attacker the chance to enter the management as well. For that reason the design of Honeywall has to be very carefully studied before its implementation.RESULTS

5. ALGORITHM

KFSensor is a Windows based honeypot Intrusion Detection System (IDS). It acts as a honeypot to attract and detect hackers and worms by simulating vulnerable system services and trojans. By acting as a decoy server it can divert attacks from critical systems and provide a higher level of information than can be achieved by using firewalls and NIDS alone. KFSensor is a system installed in a network in order to divert and study an attacker's behavior. This is a new technique that is very effective in detecting attacks.

The main feature of KFSensor is that every connection it receives is a suspect hence it results in very few false alerts. At the heart of KFSensor sits a powerful internet daemon service that is built to handle multiple ports and IP addresses. It is written to resist denial of service and buffer overflow attacks. Building on this flexibility KFSensor can respond to connections in a variety of ways, from simple port listening and basic services (such as echo), to complex simulations of standard system services. For the HTTP protocol KFSensor accurately simulates the way Microsoft's web server (IIS) responds to both valid and invalid requests. As well as being able to host a website it also handles complexities such as range requests and client side cache negotiations. This makes it extremely difficult for an attacker to fingerprint, or identify KFSensor as a honeypot

STEP-1: Download KFSensor Evaluation Setup File from KFSensor Website.

STEP-2: Install with License Agreement and appropriate directory path.

STEP-3: Reboot the Computer now. The KFSensor automatically starts during Windows boot.

STEP-4: Click Next to setup wizard.

STEP-5: Select all port classes to include and Click Next.

STEP-6: "Send the email and Send from email", enter the ID and Click Next.

STEP-7: Select the options such as Denial of Service[DOS], Port Activity, Proxy Emulsion, Network Port Analyzer, Click Next.

STEP-8: Select Install as System service and Click Next.

STEP-9: Click finish.

Conclusion: Hence we conclude that the HONEYPOT computer security-defense tool is used to understand the attackers' traps against the computer network and security.

Review Questions:

Q1.What is a honeypot? How it protects against cyber attacks

Q2. How do honeypots help in cyber security?

Q3. How do attackers detect honeypots?

Q4. Which type of data should a honeypot contain?