

Honey Pot

Abstract

The purpose of this project was to design, implement, and execute a honeypot to enhance understanding of cybersecurity threats and defensive measures. A honeypot, a decoy system intended to attract and analyze potential cyber-attacks, was created using Pentbox. The objective was to simulate a vulnerable environment to study malicious activities and gather data on attackers' tactics, techniques, and procedures (TTPs).

The project aimed to achieve two key outcomes:

1. **Threat Intelligence:** Collecting actionable data on real-world attack methods, including IP addresses, timestamps, and attack vectors.
2. **Skill Development:** Enhancing practical knowledge of network security, logging, and forensic analysis.

The findings contribute to improving defensive strategies and raising awareness about the importance of proactive security measures. This project demonstrates the potential of honeypots in educational contexts and as tools for real-time monitoring in cybersecurity infrastructures.

What is a HoneyPot

A honeypot is a cybersecurity tool designed to attract, detect, and analyze malicious activity by simulating a vulnerable system or network. It serves as a decoy that mimics real systems, making it an attractive target for cyber attackers. The idea is to engage with attackers in a controlled environment to study their tactics, techniques, and procedures (TTPs) without risking actual assets or data.

Objectives of the Honeypot Project:

1. Simulate a Vulnerable Environment:

To create a controlled, decoy system that mimics a real network or service, designed to attract and engage attackers.

2. Collect Threat Intelligence:

To gather data on malicious activities, including attack methods, tools, IP addresses, and techniques used by potential attackers.

3. Study Attacker Behavior:

To analyze how attackers interact with the honeypot, identifying patterns and understanding their tactics, techniques, and procedures (TTPs).

4. Enhance Defensive Strategies:

To use the data collected from the honeypot to improve detection systems, hone security measures, and develop stronger proactive defense mechanisms.

5. Skill Development and Practical Experience:

To gain hands-on experience with cybersecurity tools and techniques, such as honeypot deployment, network traffic analysis, and forensic investigation.

6. Raise Awareness:

To highlight the importance of honeypots in real-time cybersecurity monitoring and threat intelligence gathering.

Steps to Deploy a Honey Pot

Step 1 :- Download a tool from github called **Pentbox**

```
kali@kali: ~/Tools
File Actions Edit View Help
(kali@kali)-[~/Tools]
$ ls
(kali@kali)-[~/Tools]
$ sudo git clone https://github.com/technicaldada/pentbox.git
Cloning into 'pentbox' ...
remote: Enumerating objects: 25, done.
remote: Counting objects: 100% (8/8), done.
remote: Compressing objects: 100% (8/8), done.
remote: Total 25 (delta 1), reused 0 (delta 0), pack-reused 17 (from 1)
Receiving objects: 100% (25/25), 2.11 MiB | 1.57 MiB/s, done.
Resolving deltas: 100% (3/3), done.
```

Step 2 :- Change into the Pentbox Directory

```
(kali@kali)-[~/Tools]
$ ls
pentbox
(kali@kali)-[~/Tools]
$ cd pentbox
(kali@kali)-[~/Tools/pentbox]
$ ls
pentbox.tar.gz  README.md
(kali@kali)-[~/Tools/pentbox]
$
```

Step 3 :- Extract the Zip file of the Pentbox

sudo tar -zxvf pentbox.tar.gz

```
kali@kali: ~/Tools/pentbox
File Actions Edit View Help
(kali@kali)-[~/Tools/pentbox]
$ sudo tar -zxvf pentbox.tar.gz
pentbox-1.8/lib/racket/racket/12/.svn/text-base/llc.rb.svn-base
pentbox-1.8/lib/racket/racket/12/.svn/text-base/vlan.rb.svn-base
pentbox-1.8/lib/racket/racket/12/.svn/text-base/snap.rb.svn-base
pentbox-1.8/lib/racket/racket/12/.svn/text-base/vtp.rb.svn-base
pentbox-1.8/lib/racket/racket/12/.svn/text-base/misc.rb.svn-base
pentbox-1.8/lib/racket/racket/12/.svn/text-base/eighttwothree.rb.svn-base
pentbox-1.8/lib/racket/racket/12/.svn/text-base/ethernet.rb.svn-base
pentbox-1.8/lib/racket/racket/12/.svn/prop-base/llc.rb.svn-base
pentbox-1.8/lib/racket/racket/12/.svn/prop-base/vlan.rb.svn-base
pentbox-1.8/lib/racket/racket/12/.svn/prop-base/snap.rb.svn-base
pentbox-1.8/lib/racket/racket/12/.svn/prop-base/vtp.rb.svn-base
pentbox-1.8/lib/racket/racket/12/.svn/prop-base/misc.rb.svn-base
pentbox-1.8/lib/racket/racket/12/.svn/prop-base/eighttwothree.rb.svn-base
pentbox-1.8/lib/racket/racket/12/.svn/prop-base/ethernet.rb.svn-base
pentbox-1.8/lib/racket/racket/15/.svn/text-base/dns.rb.svn-base
pentbox-1.8/lib/racket/racket/15/.svn/text-base/harp.rb.svn-base
pentbox-1.8/lib/racket/racket/15/.svn/text-base/ntp.rb.svn-base
pentbox-1.8/lib/racket/racket/15/.svn/text-base/misc.rb.svn-base
pentbox-1.8/lib/racket/racket/15/.svn/text-base/bootp.rb.svn-base
pentbox-1.8/lib/racket/racket/15/.svn/prop-base/dns.rb.svn-base
pentbox-1.8/lib/racket/racket/15/.svn/prop-base/harp.rb.svn-base
pentbox-1.8/lib/racket/racket/15/.svn/prop-base/ntp.rb.svn-base
pentbox-1.8/lib/racket/racket/15/.svn/prop-base/misc.rb.svn-base
pentbox-1.8/lib/racket/racket/15/.svn/prop-base/bootp.rb.svn-base
pentbox-1.8/lib/racket/racket/13/.svn/text-base/arp.rb.svn-base
pentbox-1.8/lib/racket/racket/13/.svn/text-base/egp.rb.svn-base
pentbox-1.8/lib/racket/racket/13/.svn/text-base/ipv4.rb.svn-base
pentbox-1.8/lib/racket/racket/13/.svn/text-base/ipv6.rb.svn-base
pentbox-1.8/lib/racket/racket/13/.svn/text-base/cdp.rb.svn-base
pentbox-1.8/lib/racket/racket/13/.svn/text-base/stp.rb.svn-base
```

Step 4 :- Change the Directory to the Pentbox-1.8

```
kali@kali: ~/Tools/pentbox/pentbox-1.8

File Actions Edit View Help

pentbox-1.8/lib/json/.svn/
pentbox-1.8/lib/net/dns/
pentbox-1.8/lib/net/shodan/
pentbox-1.8/lib/net/.svn/
pentbox-1.8/lib/bit-struct/bit-struct/
pentbox-1.8/lib/bit-struct/.svn/
pentbox-1.8/lib/.svn/tmp/
pentbox-1.8/lib/.svn/text-base/
pentbox-1.8/lib/.svn/prop-base/
pentbox-1.8/lib/.svn/props/
pentbox-1.8/tools/network/dos_exploits/
pentbox-1.8/tools/network/.svn/
pentbox-1.8/tools/cryptography/.svn/
pentbox-1.8/tools/.svn/tmp/
pentbox-1.8/tools/.svn/text-base/
pentbox-1.8/tools/.svn/prop-base/
pentbox-1.8/tools/.svn/props/
pentbox-1.8/tools/web/.svn/
pentbox-1.8/other/log/.svn/
pentbox-1.8/other/.svn/tmp/
pentbox-1.8/other/.svn/text-base/
pentbox-1.8/other/.svn/prop-base/
pentbox-1.8/other/.svn/props/
pentbox-1.8/lib/rsocket/
pentbox-1.8/lib/json/
pentbox-1.8/lib/net/
pentbox-1.8/lib/bit-struct/
pentbox-1.8/lib/.svn/
pentbox-1.8/tools/network/
pentbox-1.8/tools/cryptography/
pentbox-1.8/tools/.svn/
pentbox-1.8/tools/web/
pentbox-1.8/other/log/
pentbox-1.8/other/.svn/
pentbox-1.8/lib/
pentbox-1.8/tools/
pentbox-1.8/other/

(kali@kali)-[~/Tools/pentbox]
$ ls
pentbox-1.8  pentbox.tar.gz  README.md

(kali@kali)-[~/Tools/pentbox]
$ cd pentbox-1.8

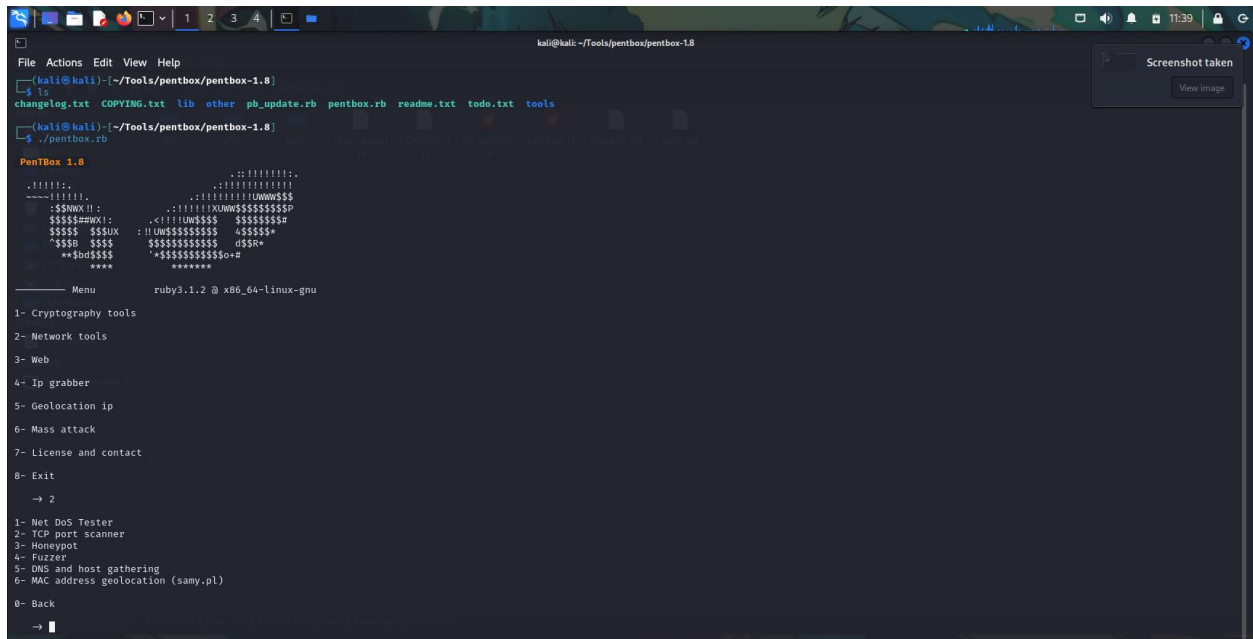
(kali@kali)-[~/Tools/pentbox/pentbox-1.8]
$
```

Step 5 :- Executing the Honey Pot

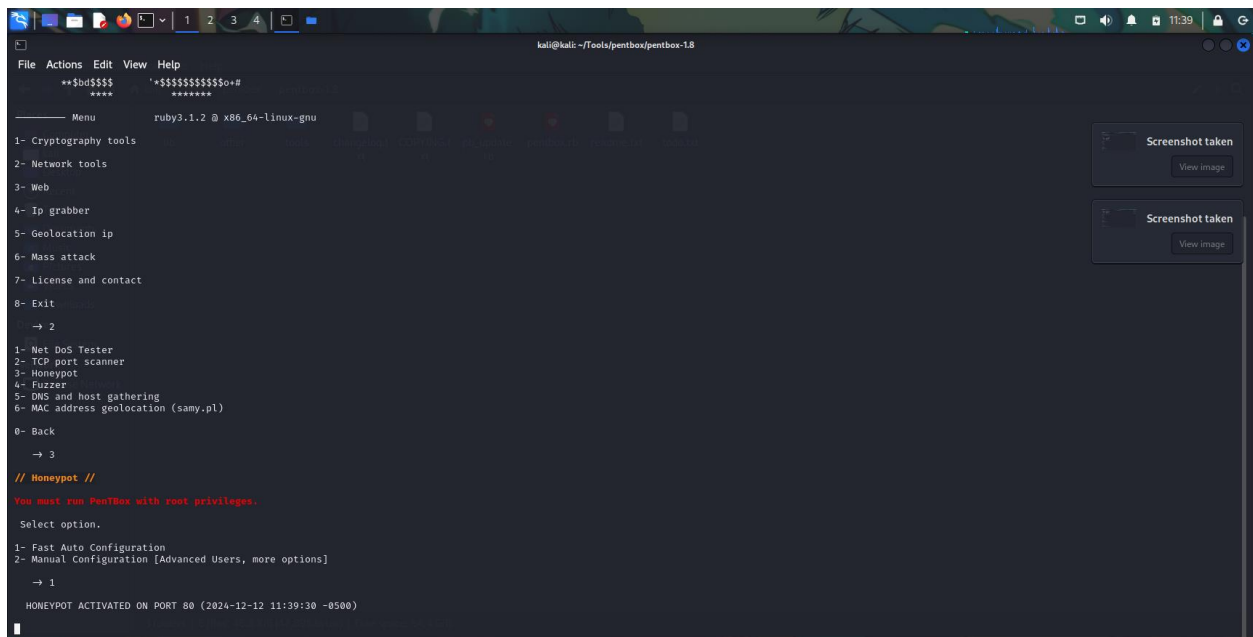
The screenshot shows a Kali Linux desktop environment with a terminal window open. The terminal displays the following commands and output:

```
kali@kali: ~/Tools/pentbox/pentbox-1.8  
$ ls  
changelog.txt  COPYING.txt  lib  other  pb_update.rb  pentbox.rb  readme.txt  todo.txt  tools  
$ ./pentbox.rb  
  
PentBox 1.0  
::: :::  
.:::.:::  
-----  
$Sawx !! :      .:::!!UwWSS$  
$$$$$umwx! : <!!!!UW$$$ $$$$$$  
$$$$$ $$$X : !! UW$$$$$$ $$$$$$  
*$S$ $$$$ $$$$$$$$$$ o$R*  
**$d$$$$ $$$$$$$$$$o+  
****          *****  
  
Menu                ruby3.1.2 @ 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  
  
->
```

Step 6 :- Select the option 2 Network Tools



Step 7 :- Now Select the option 3 HoneyPot



Step 8 :- Wait for the attacker to attack and eventually we can get his information

```
kali@kali: ~/Tools/pentbox/pentbox-1.8
File Actions Edit View Help
2- TCP port scanner
3- Honeypot
4- Fuzzer
5- DNS and host gathering
6- MAC address geolocation (samy.pl)
0- Back
  → 3
// Honeypot //
You must run PentBox with root privileges.
Select option.
1- Fast Auto Configuration
2- Manual Configuration [Advanced Users, more options]
  → 1
HONEYPOT ACTIVATED ON PORT 80 (2024-12-12 11:39:30 -0500)

INTRUSION ATTEMPT DETECTED! from 192.168.1.19:9384 (2024-12-12 11:39:55 -0500)
GET / HTTP/1.1
Host: 192.168.1.37
Connection: keep-alive
Upgrade-Insecure-Requests: 1
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/131.0.0.0 Safari/537.36
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,image/apng,*/*;q=0.8,application/signed-exchange;v=b3;q=0.7
Accept-Encoding: gzip, deflate
Accept-Language: en-GB,en-US;q=0.9,en;q=0.8,kn;q=0.7

INTRUSION ATTEMPT DETECTED! from 192.168.1.19:9385 (2024-12-12 11:39:56 -0500)
GET /favicon.ico HTTP/1.1
Host: 192.168.1.37
Connection: keep-alive
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/131.0.0.0 Safari/537.36
Accept: image/avif,image/webp,image/apng,image/svg+xml,image/*,*/*;q=0.8
Referer: http://192.168.1.37/
Accept-Encoding: gzip, deflate
Accept-Language: en-GB,en-US;q=0.9,en;q=0.8,kn;q=0.7
```

Conclusion

The successful implementation and execution of a honeypot using Pentbox provided valuable insights into the nature of modern cybersecurity threats and attacker behavior. By simulating a vulnerable environment, the project effectively attracted and recorded malicious activity, enabling the collection of critical data such as IP addresses, attack methods, and interaction patterns.

The findings demonstrate the effectiveness of honeypots as tools for threat intelligence gathering and proactive defense. Additionally, the project highlighted the importance of monitoring and analyzing attacker behavior to strengthen overall cybersecurity measures.