**Honeypots – Cowrie and Pentbox**

Network Security

Spring 2019

May 5th

Jordan Anthony – 0618991 – [anthonjt@clarkson.edu](mailto:anthonjt@clarkson.edu)

## 

Summary:

This project report shows the implementation of 2 different honeypots. These honeypots are called Pentbox and Cowrie. This report will show how to implement and attack these honeypots. The operating system used to implement Pentbox was Arch Linux and the O.S. used for Cowrie was Ubuntu 18.04 LTS. This project can be done on other systems using bash terminal. This project uses git, nmap, ssh and other basic Unix commands. This report is written with the assumption that the reader has basic knowledge of Unix systems and has git, openssh and nmap installed on their system.

**Pentbox**

Setup:

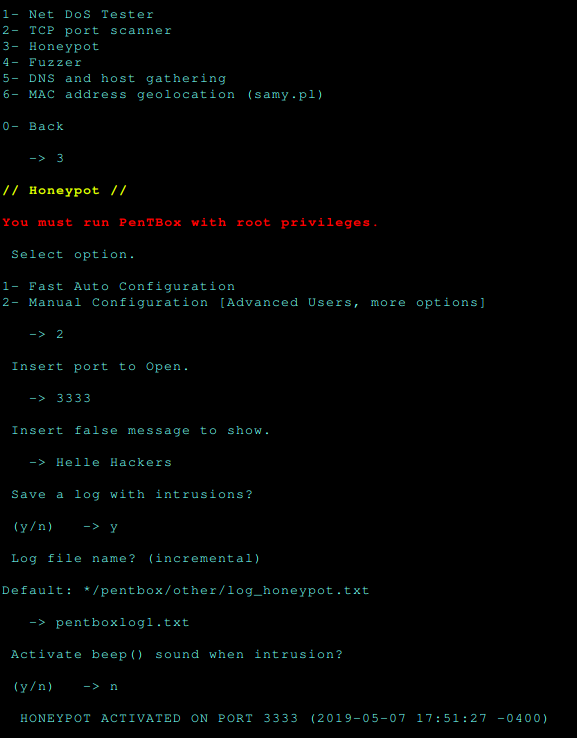
To be able to run pentbox, you will need to first download pentbox from github. The link is provided as Reference[1]. After downloading pentbox, you need to download ruby if you do not already have it. For my system, the command was pacman -Sy ruby. Replace pacman -Sy with the package manager for the machine implementing pentbox. After downloading ruby and pentbox, It is time get into pentbox. First you need to unzip the tar.gz file. This is done by using the command below.

(1) tar -zxvf pentbox-1.8.tar.gz

The above command unpacks the tar.gz file. The version of pentbox may be different. After unpacking, simply cd into the newly created pentbox folder and now pentbox can be run. To do this simply run ./pentbox.rb. After running the pentbox.rb file, the following window of options will show.

**Figure 1.01 (Pentbox view on initial run)**

To enable the honeypot, the Network tools option needs to be selected. After doing this, the next steps are shown below.



**Figure 1.02 (Pentbox full setup)**

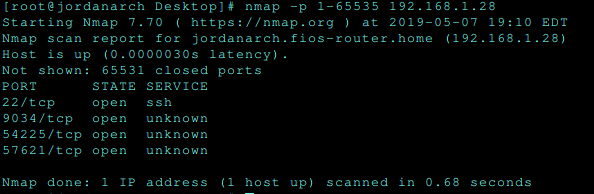
As shown above, simply select the Honeypot option, select 1 or 2 for configuration and follow the options given by pentbox. To be able to have actual attempted attacks, pentbox has to be used on an actual server or port forwarding has to be setup with a router. Since I do not own a server or have access to Clarkson University’s router setting, I was not able to do this.

Attacking:

Now that pentbox is setup, I will attempt to attack it using a machine on the same network. To do this, nmap must be used. Since I do not have my device connected to a server, I did a scan of my devices local I.P. The command I used is shown below.

(2) nmap -p 1-65535 192.168.1.28

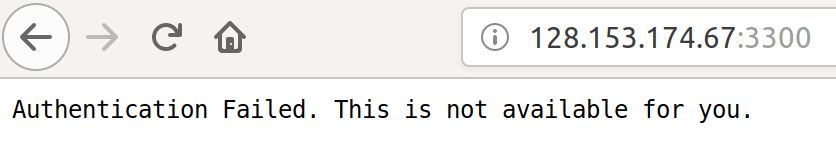
After running this command, the open port created by pentbox will be shown. The output of this command is shown below.



**Figure 1.03 (Nmap output)**

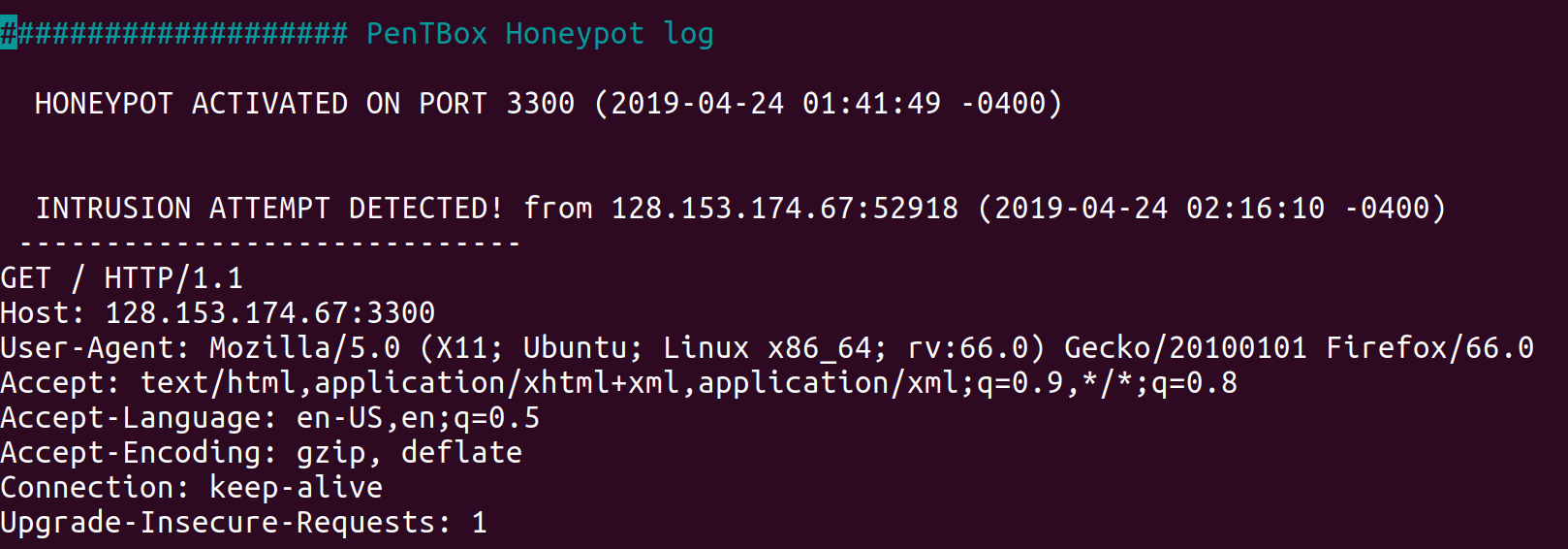
After finding the open ports using nmap, I ran every port found open to see which would connect. Since 9034 in the port that the honeypot was setup on, this is the only one that showed anything. This was done by using firefox browser. The command used is shown below. The message shown on the browser is also shown below.

(3) firefox 192.168.1.28:{22,9034,54225,57621}

**Figure 1.04 (Message from pentbox)**

Logging:

Pentbox logs the attacker when the attacker tries to gain entry using the port and denies the attackers from doing anything at all. It shows the user created message to the attacker. Pentbox logs the attacker I.P., type of connection, O.S., browser if used, language and a few other details. The log for using firefox on my Ubuntu is shown below.

**Figure 1.05 (Pentbox log for firefox on Ubuntu)**

Summary:

The pentbox honeypots works by opening a tcp port. Pentbox denies access to everyone who tries to connect. It simply displays a message and logs everything that it can from the user. This does not get as much details as someone would want but it can be deployed very easily and many can be deployed in a small amount of time. Pentbox can help create signatures for attack I.P. addresses, common operating systems used, and common ports used to attack with. An open port can simply be attacked by using nmap and a browser so it is very important to protect ports.

**Cowrie**

Setup:

To be able to run cowrie, you will need to first configure ssh for cowrie to use its default port. The ssh config file is located in /etc/ssh/sshd\_config. Simply make sure the port is not 22 because this is the port that will be used by cowrie. After configuring the ssh port, some dependencies need to be downloaded. I used Ubuntu 18.04 LTS to implement cowrie. The command I used for these dependencies is shown below.

(4) sudo apt install python-virtualenv libssl-dev libffl-dev build-essential libpython-dev python2.7-minimal authbind

After downloading the dependencies, the next step is to create a user cowrie. The command used is shown below.

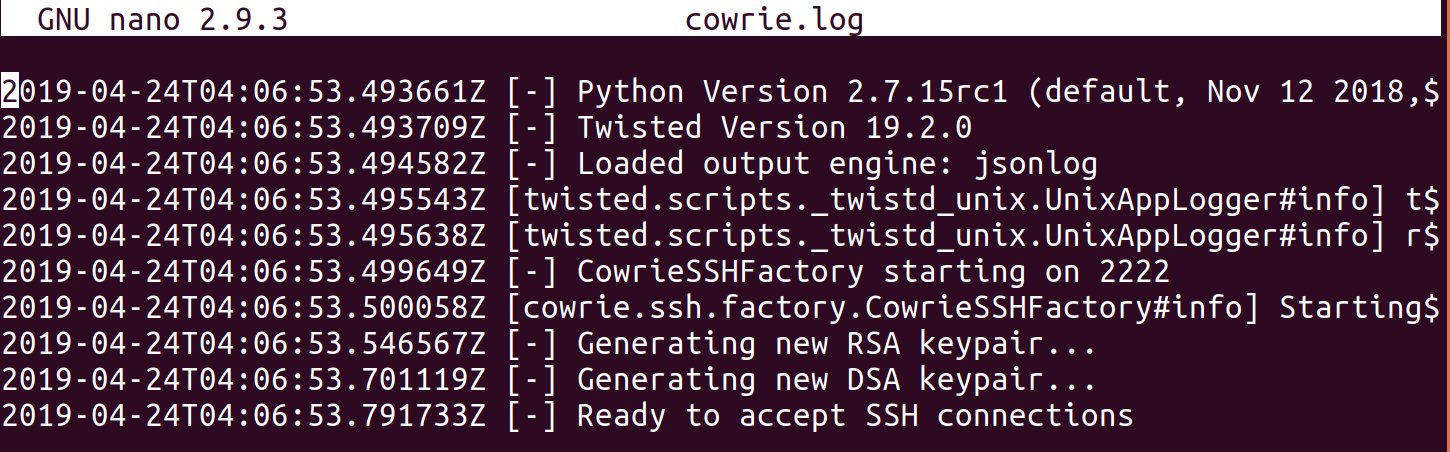
(5) sudo adduser --disabled-password cowrie

After adding a user, su the su – cowrie command to switch user as cowrie. Now cowrie needs to be downloaded. This link is provided as Reference[2]. Now that cowrie is downloaded, enter the cowrie directory. After this, 2 commands need to be run.

(6) sudo adduser --disabled-password cowrie

(7) pip install --upgrade -r requirements.txt

The last step is to simply copy the cowrie.cfg.dlist to cowrie.cfg and change the ssh port in this file from 2222 to 22. Now cowrie is set to run. This is simply done by running bin/cowrie start. After cowrie is started, the log file can be found in var/log/cowrie/cowrie.log. The initial log file after cowrie is started is shown below.

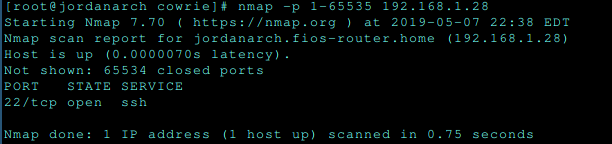
**Figure 2.01 (Initial cowrie log)**

Now that the cowrie log is working and cowrie is running, cowrie will be attacked.

Attacking:

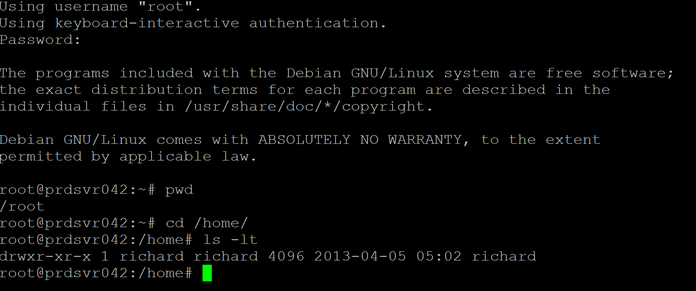
To start the attack, an nmap command has to be used to find the open ports. The command ran is shown below. Again, I had to use the local I.P. because I do not have a server or access to router settings.

(8) nmap -p 1-65535 192.168.1.28



**Figure 2.02 (Nmap for attacking cowrie)**

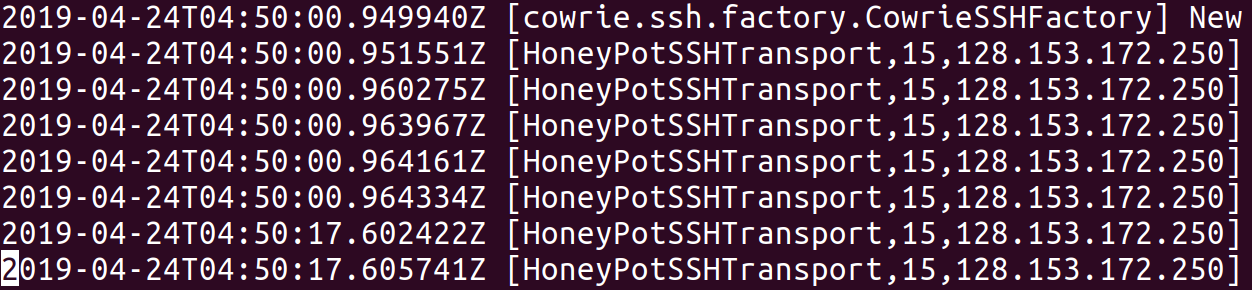
Now that the only open port is an ssh port, this will be attempted to break into. This will be done simply by running ssh -p [192.168.1.28](mailto:root@192.168.1.28):22. This is using my local I.P. address. This prompts the user for the password for root. Cowrie accepts any password and lets the user in. The user gets the screen below when logged in.

**Figure 2.03 (Cowrie attacker shell)**

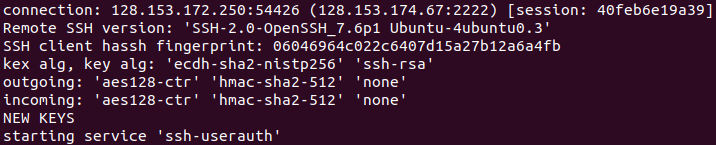
The attacker gets to use an actual shell that looks like a real linux system. It lets the attacker use commands and simulates the commands that they use. The user thinks that all of the commands are working even though none of the commands are actually working.

Logging:

Cowrie logs show more details than pentbox logs because the user can actually maneuver around in a bash shell. The cowrie log tracks the date, time and I.P. address every time the user does anything in the system. This part of the log is shown below.

**Figure 2.04 (Cowrie log left side)**

The log above only shows one part of the data that cowrie logged. The other side of the log is shown below with the same lines. Both of these logs show the data collected when the user initially runs the ssh command. This is before the user logs in or enter a password.

**Figure 2.05 (Cowrie log right side)**

Cowrie logs a lot of other data such as password used, username used, ssh service used, every command the attacker does, unsuccessful connections and much more shown in the very small print figure shown below. Cowrie also logs when the user exits the bash shell and how the the user was connected for.

**Figure 2.06 (Entire cowrie log)**

Summary:

The cowrie honeypot works well to create signatures for ssh and telnet attacks. In this project, the ssh ability of this honeypot was used but this honeypot can also be implemented for telnet connections. Cowrie lets the attacker do more than in pentbox thus more information can be logged from the attacker. Since the attacker can enter commands, you can find patterns in the commands entered to create better signatures than you could with pentbox. Cowrie cannot be implemented as quickly as pentbox because of it’s complexity but it is still fairly easy to make more than 1 honeypot with cowrie. Since it is easy for an attacker to get in, implementing cowrie on a server or using port forwarding will get a lot of hits from attackers.

References

1. [https://github.com/royaflash/pentbo](https://github.com/royaflash/pentbox)x
2. <https://github.com/cowrie/cowrie>
3. <https://www.thewindowsclub.com/what-are-honeypots>
4. <https://www.blackmoreops.com/2016/05/06/setup-honeypot-in-kali-linux/>
5. [https://null-byte.wonderhowto.com/how-to/use-cowrie-ssh-honeypot-catch](https://null-byte.wonderhowto.com/how-to/use-cowrie-ssh-honeypot-catch-)-[attackers-your-network-0181600/](https://null-byte.wonderhowto.com/how-to/use-cowrie-ssh-honeypot-catch-attackers-your-network-0181600/)