Cowrie Honeypot Project Report

PREPARED BY:
NAME: CHETHANA R
DATE :27/06/2024

Introduction to Cowrie Honeypot

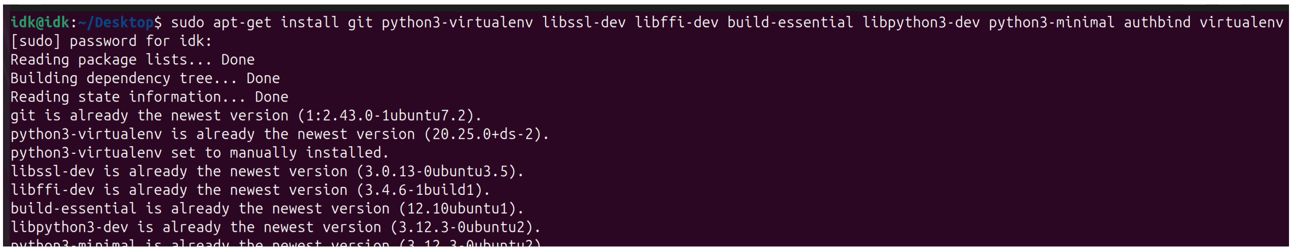
Cowrie is a high-interaction honeypot designed to log SSH and Telnet sessions by attackers. It can capture command-line activities, collect shell interaction, and even download malware samples. Cowrie provides a fake filesystem with the full Debian GNU/Linux system structure, which serves to deceive the attacker into believing that they have gained access to a real system. This allows the collection of comprehensive information about an attacker's activities, thereby providing valuable insights for threat analysis and mitigation. Since Cowrie is open source, it can be customized to meet specific research or security needs.

Practical Setup

1. Installing Dependencies

The necessary dependencies for running Cowrie were installed. This included software packages such as Python and various Python libraries.

COMMAND TO INSTALL DEPENDENCIES:
sudo apt-get install git python3-virtualenv libssl-dev libffi-dev build-essential

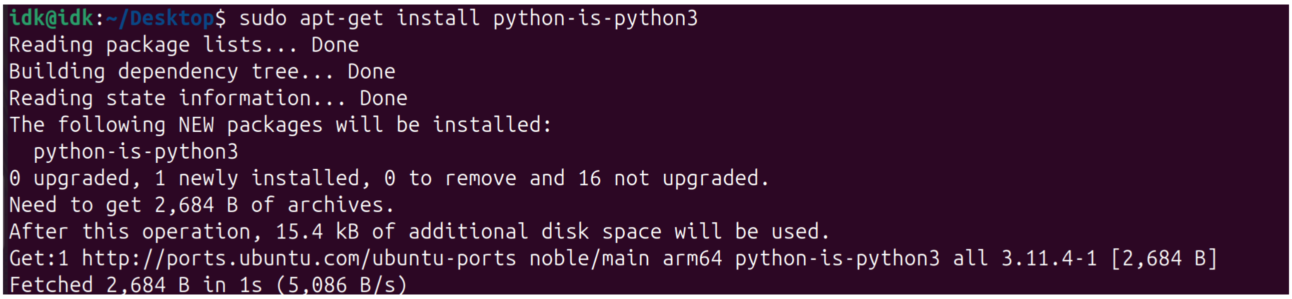


```
idk@idk:~/Desktop$ sudo apt-get install git python3-virtualenv libssl-dev libffi-dev build-essential libpython3-dev python3-minimal authbind virtualenv
[sudo] password for idk:
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
git is already the newest version (1:2.43.0-1ubuntu7.2).
python3-virtualenv is already the newest version (20.25.0+ds-2).
python3-virtualenv set to manually installed.
libssl-dev is already the newest version (3.0.13-0ubuntu3.5).
libffi-dev is already the newest version (3.4.6-1build1).
build-essential is already the newest version (12.10ubuntu1).
libpython3-dev is already the newest version (3.12.3-0ubuntu2).
python3-minimal is already the newest version (3.12.3-0ubuntu2).
```

2. Python Verification

The system was verified to have the python-is-python3 package installed. This package ensures that the python command refers to Python 3 instead of Python 2, which is required for the operation of Cowrie. This package is essential as Cowrie is designed to work with Python 3.

COMMAND TO INSTALL PYTHON-IS-PYTHON3:
sudo apt-get install python-is-python3

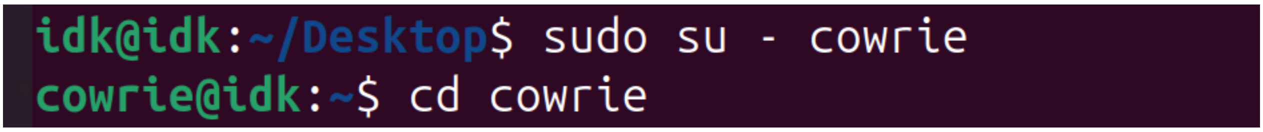


```
idk@idk:~/Desktop$ sudo apt-get install python-is-python3
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
The following NEW packages will be installed:
  python-is-python3
0 upgraded, 1 newly installed, 0 to remove and 16 not upgraded.
Need to get 2,684 B of archives.
After this operation, 15.4 kB of additional disk space will be used.
Get:1 http://ports.ubuntu.com/ubuntu-ports noble/main arm64 python-is-python3 all 3.11.4-1 [2,684 B]
Fetched 2,684 B in 1s (5,086 B/s)
```

3. Adding a User

A new user named 'cowrie' was added to the system. The password for this user was disabled to increase the security of the setup.

COMMAND TO ADD USER:
sudo adduser --disabled-password --gecos "" cowrie

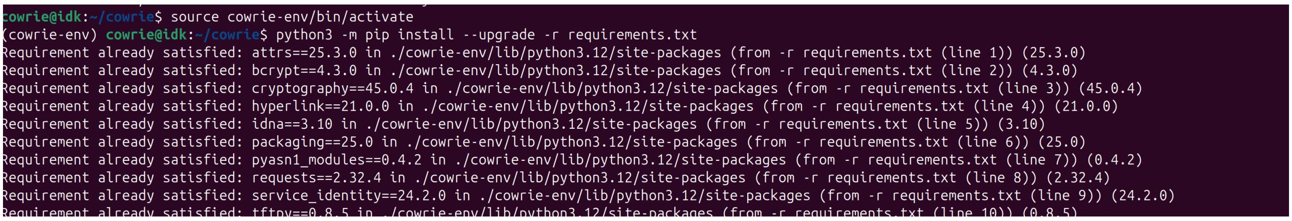


```
idk@idk:~/Desktop$ sudo su - cowrie
cowrie@idk:~$ cd cowrie
```

4. Setting Up Cowrie

- The Cowrie repository was cloned from GitHub.
- A Python virtual environment was created and activated.
- Cowrie dependencies were installed using pip.

COMMANDS:
git clone https://github.com/cowrie/cowrie.git
cd cowrie
python3 -m venv cowrie-env
source cowrie-env/bin/activate
pip install --upgrade pip
pip install -r requirements.txt

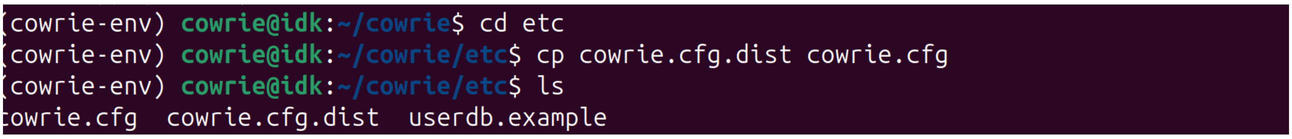


```
cowrie@idk:~/cowrie$ source cowrie-env/bin/activate
(cowrie-env) cowrie@idk:~/cowrie$ python3 -m pip install --upgrade -r requirements.txt
Requirement already satisfied: attrs==25.3.0 in ./cowrie-env/lib/python3.12/site-packages (from -r requirements.txt (line 1)) (25.3.0)
Requirement already satisfied: bcrypt==4.3.0 in ./cowrie-env/lib/python3.12/site-packages (from -r requirements.txt (line 2)) (4.3.0)
Requirement already satisfied: cryptography==45.0.4 in ./cowrie-env/lib/python3.12/site-packages (from -r requirements.txt (line 3)) (45.0.4)
Requirement already satisfied: hyperlink==21.0.0 in ./cowrie-env/lib/python3.12/site-packages (from -r requirements.txt (line 4)) (21.0.0)
Requirement already satisfied: idna==3.10 in ./cowrie-env/lib/python3.12/site-packages (from -r requirements.txt (line 5)) (3.10)
Requirement already satisfied: packaging==25.0 in ./cowrie-env/lib/python3.12/site-packages (from -r requirements.txt (line 6)) (25.0)
Requirement already satisfied: pyasn1_modules==0.4.2 in ./cowrie-env/lib/python3.12/site-packages (from -r requirements.txt (line 7)) (0.4.2)
Requirement already satisfied: requests==2.32.4 in ./cowrie-env/lib/python3.12/site-packages (from -r requirements.txt (line 8)) (2.32.4)
Requirement already satisfied: service_identity==24.2.0 in ./cowrie-env/lib/python3.12/site-packages (from -r requirements.txt (line 9)) (24.2.0)
Requirement already satisfied: tftpy==0.8.5 in ./cowrie-env/lib/python3.12/site-packages (from -r requirements.txt (line 10)) (0.8.5)
```

5. Configuring Cowrie

- The configuration files were copied and edited as needed.
- The fake filesystem was set up using the honeyfs directory.

Example:
cp etc/cowrie.cfg.dist etc/cowrie.cfg



```
(cowrie-env) cowrie@idk:~/cowrie$ cd etc
(cowrie-env) cowrie@idk:~/cowrie/etc$ cp cowrie.cfg.dist cowrie.cfg
(cowrie-env) cowrie@idk:~/cowrie/etc$ ls
cowrie.cfg  cowrie.cfg.dist  userdb.example
```

6. Generating the Fake Filesystem

- The fs.pickle file was generated using the contents of the honeyfs directory.

Command:
mkdir -p share/cowrie
python3 bin/fsctl create share/cowrie/fs.pickle honeyfs/

7. Running Cowrie

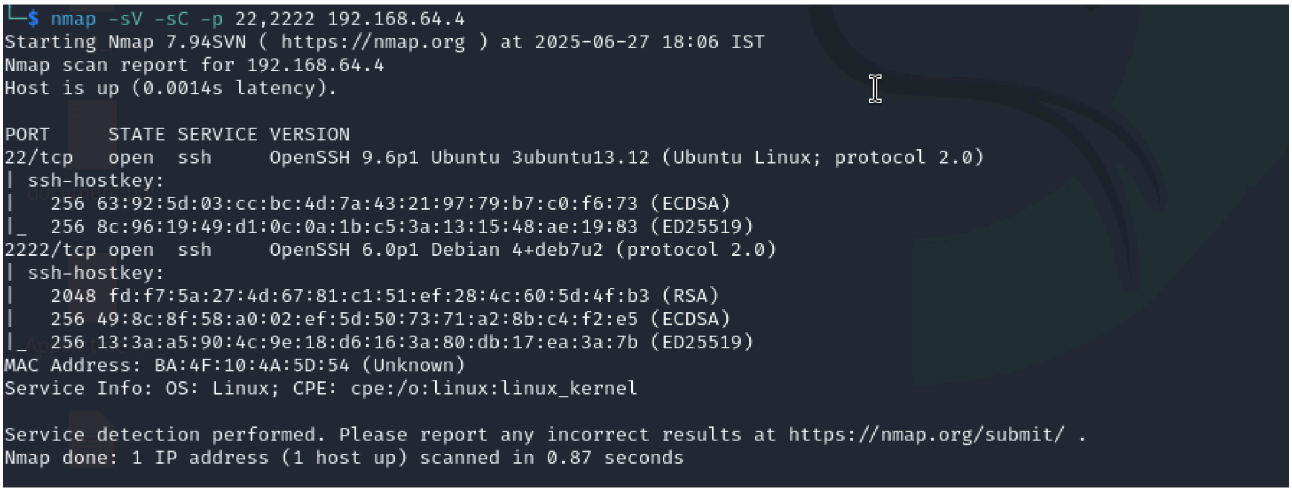
- Cowrie was started using the following command:
bin/cowrie start

8. Attacker Activity from Kali Linux

As an attacker, the following commands were executed from a Kali Linux machine to interact with the Cowrie honeypot:

RECONNAISSANCE
Nmap Scan:
nmap -sV <honeypot-ip>

Other Scanning/Enumeration Commands:
nmap -A <honeypot-ip>
nmap -p 22 <honeypot-ip>

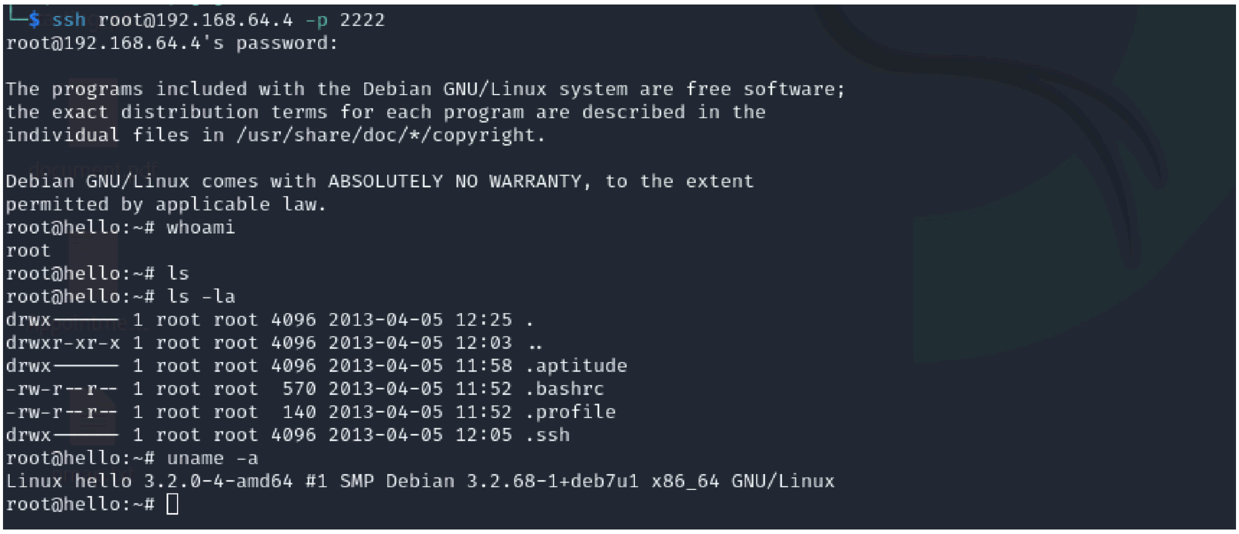


```
└─$ nmap -sV -sC -p 22,2222 192.168.64.4
Starting Nmap 7.94SVN ( https://nmap.org ) at 2025-06-27 18:06 IST
Nmap scan report for 192.168.64.4
Host is up (0.0014s latency).

PORT     STATE SERVICE VERSION
22/tcp   open  ssh     OpenSSH 9.6p1 Ubuntu 3ubuntu13.12 (Ubuntu Linux; protocol 2.0)
| ssh-hostkey:
|   256 63:92:5d:03:cc:bc:4d:7a:43:21:97:79:b7:c0:f6:73 (ECDSA)
|_  256 8c:96:19:49:d1:0c:0a:1b:c5:3a:13:15:48:ae:19:83 (ED25519)
2222/tcp open  ssh     OpenSSH 6.0p1 Debian 4+deb7u2 (protocol 2.0)
| ssh-hostkey:
|   2048 fd:f7:5a:27:4d:67:81:c1:51:ef:28:4c:60:5d:4f:b3 (RSA)
|   256 49:8c:8f:58:a0:02:ef:5d:50:73:71:a2:8b:c4:f2:e5 (ECDSA)
|_  256 13:3a:a5:90:4c:9e:18:d6:16:3a:80:db:17:ea:3a:7b (ED25519)
MAC Address: BA:4F:10:4A:5D:54 (Unknown)
Service Info: OS: Linux; CPE: cpe:/o:linux:linux_kernel

Service detection performed. Please report any incorrect results at https://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 0.87 seconds
```

hydra -l root -P /usr/share/wordlists/rockyou.txt ssh://<honeypot-ip>

Exploitation/Interaction
SSH Login Attempt:
ssh root@<honeypot-ip>

Post-Login Commands (executed after successful login to Cowrie):



```
└─$ ssh root@192.168.64.4 -p 2222
root@192.168.64.4'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@hello:~# whoami
root
root@hello:~# ls
root@hello:~# ls -la
drwx------ 1 root root 4096 2013-04-05 12:25 .
drwxr-xr-x 1 root root 4096 2013-04-05 12:03 ..
drwx------ 1 root root 4096 2013-04-05 11:58 .aptitude
-rw-r--r-- 1 root root  570 2013-04-05 11:52 .bashrc
-rw-r--r-- 1 root root  140 2013-04-05 11:52 .profile
drwx------ 1 root root 4096 2013-04-05 12:05 .ssh
root@hello:~# uname -a
Linux hello 3.2.0-4-amd64 #1 SMP Debian 3.2.68-1+deb7u1 x86_64 GNU/Linux
root@hello:~# 
```

```
ls
pwd
cat /etc/passwd
uname -a
whoami
cd /home
```

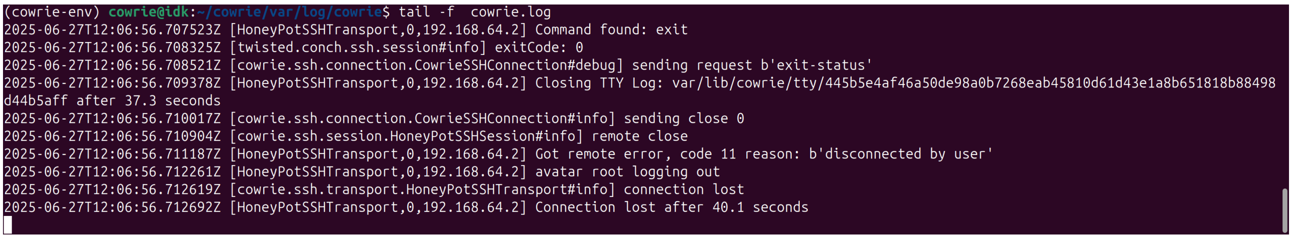
9. Cowrie Logs

Below are sample log entries captured by Cowrie during the attack simulation:

Example log file location:
var/log/cowrie/cowrie.log

Sample log files:

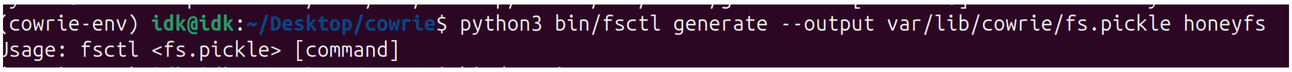
```
2025-06-27T12:06:56.707523Z [HoneyPotSSHTransport,0,192.168.64.2] Command found: exit
2025-06-27T12:06:56.708325Z [twisted.conch.ssh.session#info] exitCode: 0
2025-06-27T12:06:56.708521Z [cowrie.ssh.connection.CowrieSSHConnection#debug] sending request b'exit-status'
2025-06-27T12:06:56.709378Z [HoneyPotSSHTransport,0,192.168.64.2] Closing TTY Log: var/lib/cowrie/tty/445b5e4af46a50de98a0b7268eab45810d61d43e1a8b651818b88498d44b5aff after 37.3 seconds
2025-06-27T12:06:56.710017Z [cowrie.ssh.connection.CowrieSSHConnection#info] sending close 0
2025-06-27T12:06:56.710904Z [cowrie.ssh.session.HoneyPotSSHSession#info] remote close
2025-06-27T12:06:56.711187Z [HoneyPotSSHTransport,0,192.168.64.2] Got remote error, code 11 reason: b'disconnected by user'
2025-06-27T12:06:56.712261Z [HoneyPotSSHTransport,0,192.168.64.2] avatar root logging out
2025-06-27T12:06:56.712619Z [cowrie.ssh.transport.HoneyPotSSHTransport#info] connection lost
```



```
(cowrie-env) cowrie@idk:~/cowrie/var/log/cowrie$ tail -f  cowrie.log
2025-06-27T12:06:56.707523Z [HoneyPotSSHTransport,0,192.168.64.2] Command found: exit
2025-06-27T12:06:56.708325Z [twisted.conch.ssh.session#info] exitCode: 0
2025-06-27T12:06:56.708521Z [cowrie.ssh.connection.CowrieSSHConnection#debug] sending request b'exit-status'
2025-06-27T12:06:56.709378Z [HoneyPotSSHTransport,0,192.168.64.2] Closing TTY Log: var/lib/cowrie/tty/445b5e4af46a50de98a0b7268eab45810d61d43e1a8b651818b88498
d44b5aff after 37.3 seconds
2025-06-27T12:06:56.710017Z [cowrie.ssh.connection.CowrieSSHConnection#info] sending close 0
2025-06-27T12:06:56.710904Z [cowrie.ssh.session.HoneyPotSSHSession#info] remote close
2025-06-27T12:06:56.711187Z [HoneyPotSSHTransport,0,192.168.64.2] Got remote error, code 11 reason: b'disconnected by user'
2025-06-27T12:06:56.712261Z [HoneyPotSSHTransport,0,192.168.64.2] avatar root logging out
2025-06-27T12:06:56.712619Z [cowrie.ssh.transport.HoneyPotSSHTransport#info] connection lost
2025-06-27T12:06:56.712692Z [HoneyPotSSHTransport,0,192.168.64.2] Connection lost after 40.1 seconds
```

10. Issues Encountered & Solutions

- Issue: fs.pickle file not found.
  Solution: Generated using fsctl as described above.
- Issue: Dependency errors.
  Solution: Installed missing packages using apt-get and pip.



```
(cowrie-env) idk@idk:~/Desktop/cowrie$ python3 bin/fsctl generate --output var/lib/cowrie/fs.pickle honeyfs
Jsage: fsctl <fs.pickle> [command]
```

11. References

- Cowrie GitHub Repository: https://github.com/cowrie/cowrie
- Cowrie Documentation: https://cowrie.readthedocs.io/en/latest/