Building a Cowrie Honeypot Simulating SSH and Telnet Services to Monitor Attacker Activity

Shree Varshaa R M

March 12, 2025

Overview:

This project involves setting up a Cowrie honeypot to simulate SSH and Telnet services, monitor attacker activity, and analyze logs. Cowrie is a medium-interaction honeypot designed to log brute-force attacks, shell commands, and other malicious activities.

Features:

- Simulates SSH and Telnet services.
- Logs attacker interactions (e.g., login attempts, commands executed).
- Easy to deploy and customize.
- Provides a fake shell environment to trap attackers.

Installation

Prerequisites

- A Linux system (e.g., Ubuntu, Kali Linux).
- Python 3.x installed.

Step 1: Set Up Your Environment

- Use a spare computer, a Raspberry Pi, or a virtual machine (VM) with a Linux distribution (e.g., Ubuntu, Debian).
- If using a VM, you can use VirtualBox, VMware, or any cloud service like AWS, Azure, or Google Cloud.
- **Install Linux:** Download and install Ubuntu Server or Desktop (or any Linux distribution you're comfortable with).
- Ensure the system is updated:by giving the command **sudo apt update && sudo apt upgrade -y**

- **Isolate the System:** Place the honeypot in a DMZ (Demilitarized Zone) or on a separate network segment to avoid exposing your main network to risks.
- If you're testing at home, use a separate VLAN or a virtual network.

Step 2: Install Cowrie Honeypot

- Install the required dependencies and packages

```
┌──(kali㉿kali)-[~]
└─$ sudo apt install git python3-venv python3-pip
[sudo] password for kali:
git is already the newest version (1:2.45.2-1).
git set to manually installed.
python3-venv is already the newest version (3.12.6-1).
python3-venv set to manually installed.
python3-pip is already the newest version (24.3.1+dfsg-1).
python3-pip set to manually installed.
Summary:
  Upgrading: 0, Installing: 0, Removing: 0, Not Upgrading: 0
```

- Clone the Cowrie repository from GitHub

```
┌──(kali㉿kali)-[~]
└─$ git clone https://github.com/cowrie/cowrie
Cloning into 'cowrie'...
remote: Enumerating objects: 18823, done.
remote: Counting objects: 100% (61/61), done.
remote: Compressing objects: 100% (55/55), done.
remote: Total 18823 (delta 41), reused 5 (delta 5), pack-reused 18762 (from 3)
Receiving objects: 100% (18823/18823), 10.35 MiB | 1.02 MiB/s, done.
Resolving deltas: 100% (13246/13246), done.
```

- Create a Python virtual environment to isolate Cowrie's dependencies
- Install the python required packages

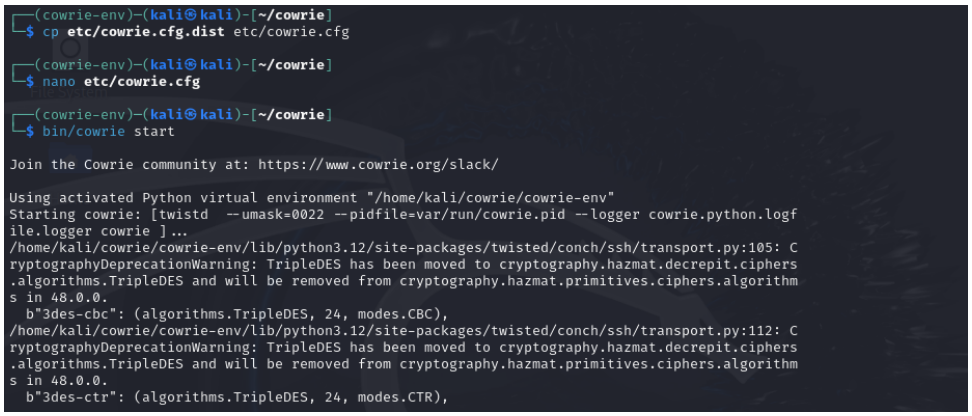
```
┌──(kali㉿kali)-[~]
└─$ cd cowrie

┌──(kali㉿kali)-[~/cowrie]
└─$ python3 -m venv cowrie-env

┌──(kali㉿kali)-[~/cowrie]
└─$ source cowrie-env/bin/activate

┌──(cowrie-env)─(kali㉿kali)-[~/cowrie]
└─$ pip install -r requirements.txt
Collecting attrs==25.1.0 (from -r requirements.txt (line 1))
  Downloading attrs-25.1.0-py3-none-any.whl.metadata (10 kB)
Collecting bcrypt==4.2.1 (from -r requirements.txt (line 2))
  Downloading bcrypt-4.2.1-cp39-abi3-manylinux_2_28_x86_64.whl.metadata (9.8 kB)
Collecting cryptography==44.0.2 (from -r requirements.txt (line 3))
  Downloading cryptography-44.0.2-cp39-abi3-manylinux_2_34_x86_64.whl.metadata (5.7 kB)
Collecting hyperlink==21.0.0 (from -r requirements.txt (line 4))
  Downloading hyperlink-21.0.0-py2.py3-none-any.whl.metadata (1.5 kB)
Collecting idna==3.10 (from -r requirements.txt (line 5))
  Downloading idna-3.10-py3-none-any.whl.metadata (10 kB)
Collecting packaging==24.2 (from -r requirements.txt (line 6))
```

Step:3 Configure Cowrie

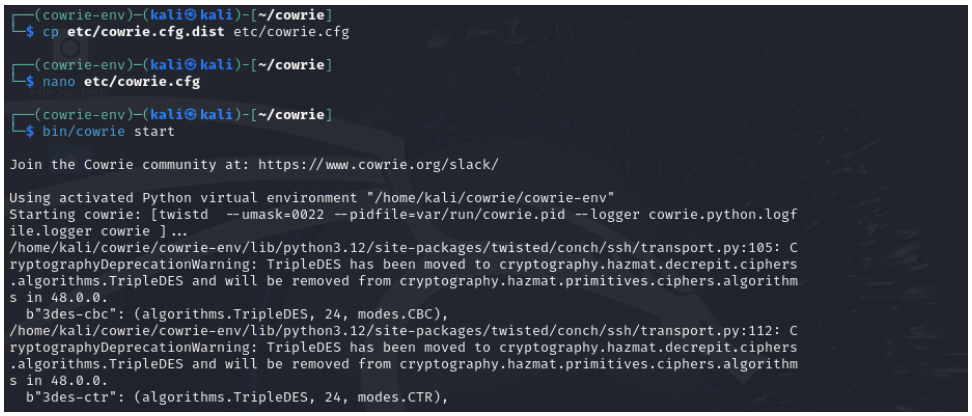
- Cowrie's configuration file is located at **etc/cowrie.cfg.dist**. Copy it to create a new configuration file
- Open it with a text editor
- Change the following settings in the configuration file:
  - **hostname:** Set a fake hostname (e.g., ubuntu-server).

- **listen_endpoints:** Set the IP and port for Cowrie to listen on (e.g., tcp:22:interface=0.0.0.0 for SSH).
- **logfile:** Specify where logs will be stored (e.g., /var/log/cowrie/).

- Save the changes



Step 4: Start the Honeypot



- **Verify It's Running:** Check if Cowrie is listening on port 22
- You can verify with the output that Cowrie is running

Step 5: Monitor and Analyze Activity

- **View Logs:**
  - Cowrie logs all activity in the var/log/cowrie/ directory.
  - Check the logs to see if attackers are connecting

```
┌──(cowrie-env)─(kali㉿kali)-[~/cowrie]
└─$ tail -f var/log/cowrie/cowrie.log
2025-03-12T06:58:49.729564Z [cowrie.ssh.session.HoneyPotSSHSession#info] channel open
2025-03-12T06:58:49.729692Z [cowrie.ssh.connection.CowrieSSHConnection#debug] got global b'no-mo
re-sessions@openssh.com' request
2025-03-12T06:58:49.762236Z [twisted.conch.ssh.session#info] Handling pty request: b'xterm-256co
lor' (24, 80, 0, 0)
2025-03-12T06:58:49.762386Z [SSHChannel session (0) on SSHService b'ssh-connection' on HoneyPotS
SHTransport,1,10.11.140.72] Terminal Size: 80 24
2025-03-12T06:58:49.762824Z [SSHChannel session (0) on SSHService b'ssh-connection' on HoneyPotS
SHTransport,1,10.11.140.72] request_env: LANG=en_US.UTF-8
2025-03-12T06:58:49.763192Z [twisted.conch.ssh.session#info] Getting shell
2025-03-12T06:59:00.379644Z [HoneyPotSSHTransport,1,10.11.140.72] CMD: ls
2025-03-12T06:59:00.380143Z [HoneyPotSSHTransport,1,10.11.140.72] Command found: ls
2025-03-12T06:59:03.521944Z [HoneyPotSSHTransport,1,10.11.140.72] CMD: whoami
2025-03-12T06:59:03.522356Z [HoneyPotSSHTransport,1,10.11.140.72] Command found: whoami
2025-03-12T06:59:59.375798Z [HoneyPotSSHTransport,1,10.11.140.72] CMD: ls
2025-03-12T06:59:59.376329Z [HoneyPotSSHTransport,1,10.11.140.72] Command found: ls
2025-03-12T07:00:42.071919Z [HoneyPotSSHTransport,1,10.11.140.72] CMD: pwd
2025-03-12T07:00:42.073038Z [HoneyPotSSHTransport,1,10.11.140.72] Command found: pwd
2025-03-12T07:01:10.750270Z [HoneyPotSSHTransport,1,10.11.140.72] CMD: uname -a
2025-03-12T07:01:10.750870Z [HoneyPotSSHTransport,1,10.11.140.72] Command found: uname -a
2025-03-12T07:01:19.122251Z [HoneyPotSSHTransport,1,10.11.140.72] CMD: cat /etc/passwd
2025-03-12T07:01:19.122872Z [HoneyPotSSHTransport,1,10.11.140.72] Command found: cat /etc/passwd
2025-03-12T07:01:26.237135Z [HoneyPotSSHTransport,1,10.11.140.72] CMD: ifconfig
2025-03-12T07:01:26.237885Z [HoneyPotSSHTransport,1,10.11.140.72] Command found: ifconfig
2025-03-12T07:03:37.675936Z [HoneyPotSSHTransport,1,10.11.140.72] CMD: exit
2025-03-12T07:03:37.677452Z [HoneyPotSSHTransport,1,10.11.140.72] Command found: exit
2025-03-12T07:03:37.677769Z [twisted.conch.ssh.session#info] exitCode: 0
2025-03-12T07:03:37.678492Z [cowrie.ssh.connection.CowrieSSHConnection#debug] sending request b'exit-status'
2025-03-12T07:03:37.680350Z [HoneyPotSSHTransport,1,10.11.140.72] Closing TTY Log: var/lib/cowrie/tty/46d805c2ad6e1
f9ac6742b9ec8660a0df8e68e9b8531849d1032f4ccfb2ddb5f after 287.9 seconds
2025-03-12T07:03:37.680858Z [cowrie.ssh.connection.CowrieSSHConnection#info] sending close 0
2025-03-12T07:03:37.684443Z [cowrie.ssh.session.HoneyPotSSHSession#info] remote close
2025-03-12T07:03:37.688336Z [HoneyPotSSHTransport,1,10.11.140.72] Got remote error, code 11 reason: b'disconnected
by user'
2025-03-12T07:03:37.689106Z [HoneyPotSSHTransport,1,10.11.140.72] avatar root logging out
2025-03-12T07:03:37.689492Z [cowrie.ssh.transport.HoneyPotSSHTransport#info] connection lost
2025-03-12T07:03:37.690337Z [HoneyPotSSHTransport,1,10.11.140.72] Connection lost after 290.2 seconds
```

Step 6: Connect to the Honeypot:

- Use SSH to connect to the honeypot **ssh root@honeypot-ip -p 2222**
- Here, our atatcker's machine IP address is **10.11.130.184**
- Use any username and password (e.g., root:password).

```
varshini@varsh:~$ ssh root@10.11.130.184 -p 2222
root@10.11.130.184'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@ubuntu:~#
```

Interact with the Honeypot:

- Run commands like ls, whoami, ifocnfig or exit to simulate attacker activity.

```
root@ubuntu:~# ifconfig
eth0      Link encap:Ethernet  HWaddr 3d:dc:d3:b9:dc:26
          inet addr:10.11.130.184  Bcast:10.11.130.255  Mask:255.255.255.0
          inet6 addr: fe09::12c:dcff:fe7d:3401/64 Scope:Link
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:457795 errors:0 dropped:0 overruns:0 frame:0
          TX packets:551815 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:177944365 (177.9 MB)  TX bytes:15383341 (15.4 MB)


lo        Link encap:Local Loopback
          inet addr:127.0.0.1  Mask:255.0.0.0
          inet6 addr: ::1/128 Scope:Host
          UP LOOPBACK RUNNING  MTU:65536  Metric:1
          RX packets:110 errors:0 dropped:0 overruns:0 frame:0
          TX packets:110 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:0
          RX bytes:28771810 (28.8 MB)  TX bytes:28771810 (28.8 MB)
```

```
root@ubuntu:~# uname -a
Linux ubuntu 3.2.0-4-amd64 #1 SMP Debian 3.2.68-1+deb7u1 x86_64 GNU/Linux
root@ubuntu:~# pwd
/root
root@ubuntu:~# cat /etc/passwd
root:x:0:0:root:/root:/bin/bash
daemon:x:1:1:daemon:/usr/sbin:/bin/sh
bin:x:2:2:bin:/bin:/bin/sh
sys:x:3:3:sys:/dev:/bin/sh
sync:x:4:65534:sync:/bin:/bin/sync
games:x:5:60:games:/usr/games:/bin/sh
man:x:6:12:man:/var/cache/man:/bin/sh
lp:x:7:7:lp:/var/spool/lpd:/bin/sh
mail:x:8:8:mail:/var/mail:/bin/sh
news:x:9:9:news:/var/spool/news:/bin/sh
uucp:x:10:10:uucp:/var/spool/uucp:/bin/sh
proxy:x:13:13:proxy:/bin:/bin/sh
www-data:x:33:33:www-data:/var/www:/bin/sh
backup:x:34:34:backup:/var/backups:/bin/sh
list:x:38:38:Mailing List Manager:/var/list:/bin/sh
irc:x:39:39:ircd:/var/run/ircd:/bin/sh
gnats:x:41:41:Gnats Bug-Reporting System (admin):/var/lib/gnats:/bin/sh
nobody:x:65534:65534:nobody:/nonexistent:/bin/sh
libuuid:x:100:101::/var/lib/libuuid:/bin/sh
sshd:x:101:65534::/var/run/sshd:/usr/sbin/nologin
phil:x:1000:1000:Phil California,,,:/home/phil:/bin/bash
root@ubuntu:~# Connection to 10.11.130.184 closed by remote host.
Connection to 10.11.130.184 closed.
```

Analyze Attacks:

- Look for Usernames and passwords attackers are trying.
- Commands they're executing.
- IP addresses of attackers.

Findings:

- During the project, the following observations were made:
  - **Common Attack Patterns:**
  - Brute force attempts using common usernames (root, admin) and passwords (password, 1234).
  - Attackers executing commands like ls, whoami, and uname -a.
  - **Interesting Logs:**IP 10.11.130.184 attempted to exploit a vulnerability using a custom script.
  - Multiple login attempts from the same IP within a short time frame.

Use Additional Tools:

- Use tools like fail2ban to block repeated attack attempts
- Use Wireshark to capture and analyze network traffic.

Conclusion:

This project provided hands-on experience with deploying a honeypot, monitoring attacker activity, and analyzing logs. It highlighted the importance of understanding attacker behavior and securing systems against common threats.