Honeypot (Cowrie)

Basic honeypot to attract attackers

lovish

7/27/2025

Table of Contents

Honeypot Project

Introduction

A honeypot is a decoy system or server that simulate a vulnerable system to attract and interact with attackers, with the goal of monitoring the attackers activities, the tool and technique they used without exposing the real system.

How it works:

1. Simulation:

It will simulate a fake service like SSH, FTP, SMB that behaves just like a real one. For eg cowrie emulates an interactive SSH sessions, logs every keystroke and even simulate a fake file system.

2. Isolation:

It is logically isolated to the real network to ensure the attackers cannot get access to the critical system, it is often run in a DMZ or on an isolated network.

3. Logging:

IP addresses, Brute force attempts, Command executed, Uploaded files, Port scans, Malware dropped.

This provides the indicators of compromise(IOCs) and Tactics, Techniques, Procedures(TTP) for threat intelligence.

4. No legitimate user:

Any interaction with the honeypot is considered malicious by default. There is no reason a legitimate user would access it.

Why Honeypots Matter in Cybersecurity

- **Threat Intel**: Capture real-world exploits **before** they hit your production.
- **Early Warning**: Attackers probing your honeypot? They're scanning your network too.
- **Behavior Analysis**: Understand attacker **goals**, **paths**, **payloads**, and **scripts**.
- **SOC Use Case**: Alert on honeypot interaction → high confidence indicator.

Why I Chose This Project

I chose to set up and work with a **honeypot** (**Cowrie**) because it is a real-world, hands-on cybersecurity project that helps simulate and observe malicious behavior in a controlled environment. Honeypots are used in both defensive security and threat intelligence operations, and I wanted to explore how attackers interact with systems once they gain unauthorized access.

It was also an opportunity to **apply core cybersecurity concepts**, such as:

- Intrusion detection
- Log analysis
- Threat behavior modeling
- Network and system monitoring

What I Aimed to Learn or Achieve

My main objectives were:

1. **Understand Attacker Behavior**:
    - I wanted to observe the common actions an attacker might perform after a successful brute-force attack (e.g., commands used, file uploads, malware activity).
2. **Gain Experience with Defensive Tools**:
    - I aimed to gain hands-on experience using **Cowrie**, setting it up on Ubuntu, configuring the network, and collecting logs.
3. **Practice Monitoring and Analysis**:
    - By analyzing Cowrie's log files, I aimed to learn how to extract useful information like IP addresses, used credentials, and commands.
4. **Simulate a Realistic SOC Scenario**:
    - The goal was to create a mini-SOC use case where I could detect attacks, correlate events, and think like a blue team analyst.
5. **Improve my Linux and Networking Skills**:
    - This project involved using SSH, iptables, file permissions, and working with JSON log files, helping me improve both technical and analytical skills.

Tools and Environment

Virtualization & OS

- **VirtualBox:** Used to create and manage the 2 virtual machines.
- **Ubuntu 22.04 LTS VM:** Host the cowrie honeypot

- **Kali Linux VM:** Used as the attacker machine to simulate the brute force attacks, port scanning, and post-exploitation.

Honeypot

- **Cowrie Honeypot** – A medium- to high-interaction SSH and Telnet honeypot to log attacker interactions.

Attack Tools (on Kali Linux)

- **Hydra** – Used for SSH brute-force attacks to test Cowrie's logging.
- **Nmap** – For scanning open ports and detecting services.
- **Basic Linux Commands** – Simulated attacker behavior (ls, pwd, whoami, etc.).

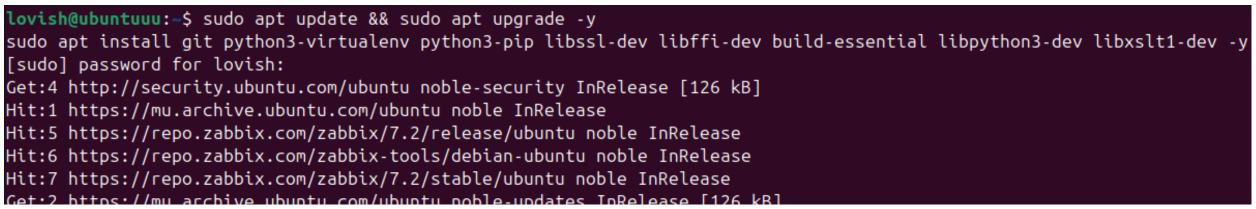
Setup Process to set up cowrie

Step-by-step explanation of what I did with screenshots, e.g.:

sudo apt update && sudo apt upgrade -y && sudo apt install git python3-virtualenv python3-pip libssl-dev libffi-dev build-essential libpython3-dev libxslt1-dev –y

This command does 3 main things:

1. sudo apt update
   Updates the list of available software packages from Ubuntu's servers.
2. sudo apt upgrade -y
   Installs the latest updates for all installed software on the system.
   The -y automatically answers "yes" to confirm updates.
3. sudo apt install ...
   Installs all the required tools and libraries needed to run and build the Cowrie honeypot:
   - git: To download code from GitHub.
   - python3-virtualenv: To create an isolated Python environment.
   - python3-pip: To install Python packages.
   - libssl-dev, libffi-dev: Required for security-related Python packages.
   - build-essential: A group of tools to compile programs from source code.
   - libpython3-dev: Python development files needed by some packages.
   - libxslt1-dev: For parsing and transforming XML files (used by some Python libs).



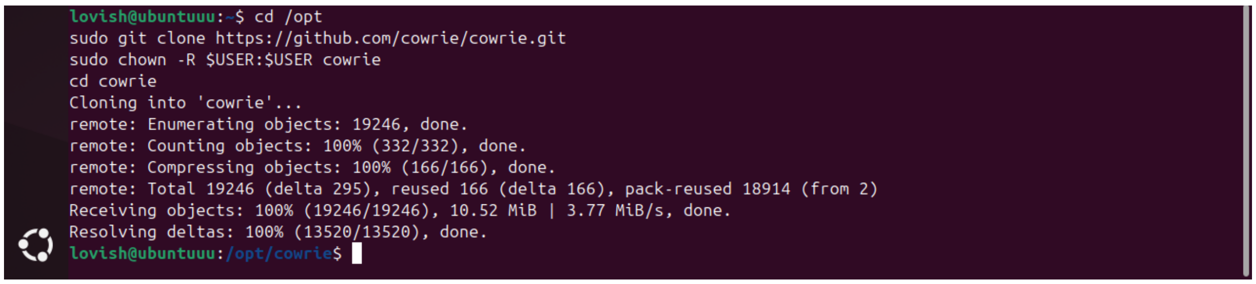
```
lovish@ubuntuu:~$ sudo apt update && sudo apt upgrade -y
sudo apt install git python3-virtualenv python3-pip libssl-dev libffi-dev build-essential libpython3-dev libxslt1-dev -y
[sudo] password for lovish:
Get:4 http://security.ubuntu.com/ubuntu noble-security InRelease [126 kB]
Hit:1 https://mu.archive.ubuntu.com/ubuntu noble InRelease
Hit:5 https://repo.zabbix.com/zabbix/7.2/release/ubuntu noble InRelease
Hit:6 https://repo.zabbix.com/zabbix-tools/debian-ubuntu noble InRelease
Hit:7 https://repo.zabbix.com/zabbix/7.2/stable/ubuntu noble InRelease
Get:2 https://mu.archive.ubuntu.com/ubuntu noble-updates InRelease [126 kB]
```

cd /opt

sudo git clone https://github.com/cowrie/cowrie.git

sudo chown -R $USER:$USER cowrie cd cowrie

1. cd /opt
   Changes directory to /opt, a common location for installing optional software.
2. sudo git clone https://github.com/cowrie/cowrie.git
   Downloads (clones) the Cowrie honeypot source code from GitHub into a folder called cowrie.
3. sudo chown -R $USER:$USER cowrie
   Changes the owner of the cowrie folder and all its files to the current user.
   This is important so you don't get permission errors later.
4. cd cowrie
   Enters the Cowrie directory so we can configure and install it.

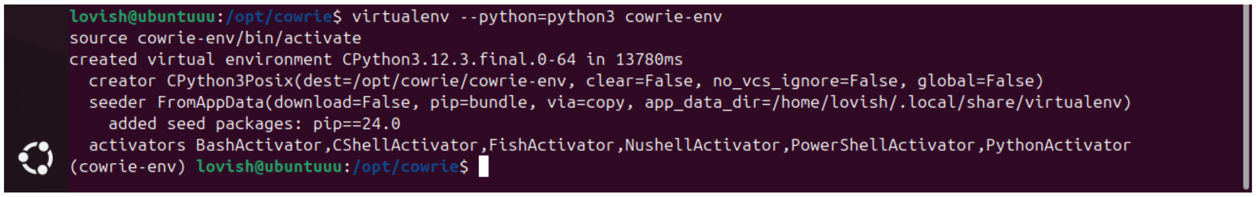


```
lovish@ubuntuu:~$ cd /opt
sudo git clone https://github.com/cowrie/cowrie.git
sudo chown -R $USER:$USER cowrie
cd cowrie
Cloning into 'cowrie'...
remote: Enumerating objects: 19246, done.
remote: Counting objects: 100% (332/332), done.
remote: Compressing objects: 100% (166/166), done.
remote: Total 19246 (delta 295), reused 166 (delta 166), pack-reused 18914 (from 2)
Receiving objects: 100% (19246/19246), 10.52 MiB | 3.77 MiB/s, done.
Resolving deltas: 100% (13520/13520), done.
lovish@ubuntuu:/opt/cowrie$
```

virtualenv --python=python3 cowrie-env

source cowrie-env/bin/activate

1. virtualenv --python=python3 cowrie-env
   Creates a virtual environment named cowrie-env using Python 3.
   This keeps Cowrie's Python packages separate from the system's packages, avoiding conflicts.
2. source cowrie-env/bin/activate
   Activates the virtual environment.
   Once activated, any Python commands or installations will happen inside this environment.

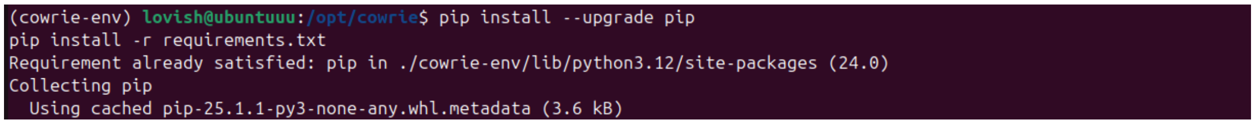


```
lovish@ubuntuuu:/opt/cowrie$ virtualenv --python=python3 cowrie-env
source cowrie-env/bin/activate
created virtual environment CPython3.12.3.final.0-64 in 13780ms
  creator CPython3Posix(dest=/opt/cowrie/cowrie-env, clear=False, no_vcs_ignore=False, global=False)
  seeder FromAppData(download=False, pip=bundle, via=copy, app_data_dir=/home/lovish/.local/share/virtualenv)
    added seed packages: pip==24.0
  activators BashActivator,CShellActivator,FishActivator,NushellActivator,PowerShellActivator,PythonActivator
(cowrie-env) lovish@ubuntuuu:/opt/cowrie$
```

pip install --upgrade pip

pip install -r requirements.txt

1. **pip install --upgrade pip**
   Updates pip (Python's package installer) to the latest version to avoid errors during installation.
2. **pip install -r requirements.txt**
   Installs all the required Python packages listed in the requirements.txt file, which comes with Cowrie.

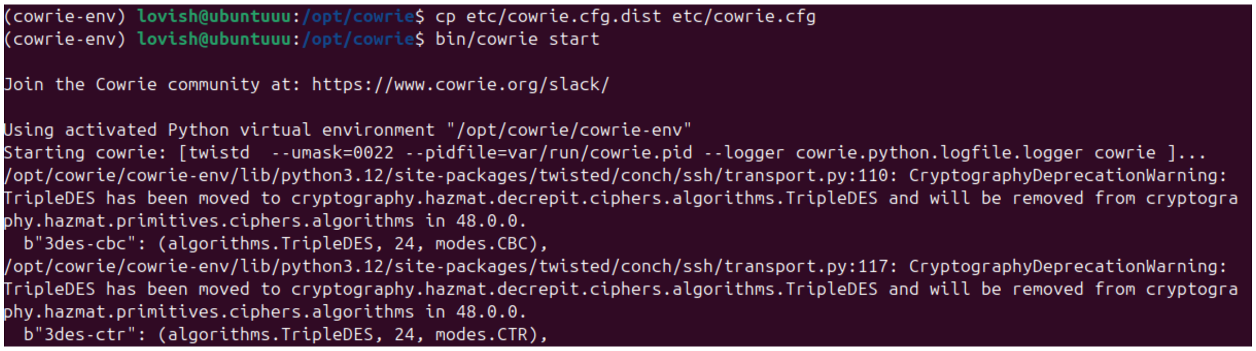


```
(cowrie-env) lovish@ubuntuuu:/opt/cowrie$ pip install --upgrade pip
pip install -r requirements.txt
Requirement already satisfied: pip in ./cowrie-env/lib/python3.12/site-packages (24.0)
Collecting pip
  Using cached pip-25.1.1-py3-none-any.whl.metadata (3.6 kB)
```

cp etc/cowrie.cfg.dist etc/cowrie.cfg

bin/cowrie start

1. **cp etc/cowrie.cfg.dist etc/cowrie.cfg**
   Copies the default config file (cowrie.cfg.dist) and renames it to cowrie.cfg.
2. **Cowrie** begins listening for fake SSH and Telnet connections.

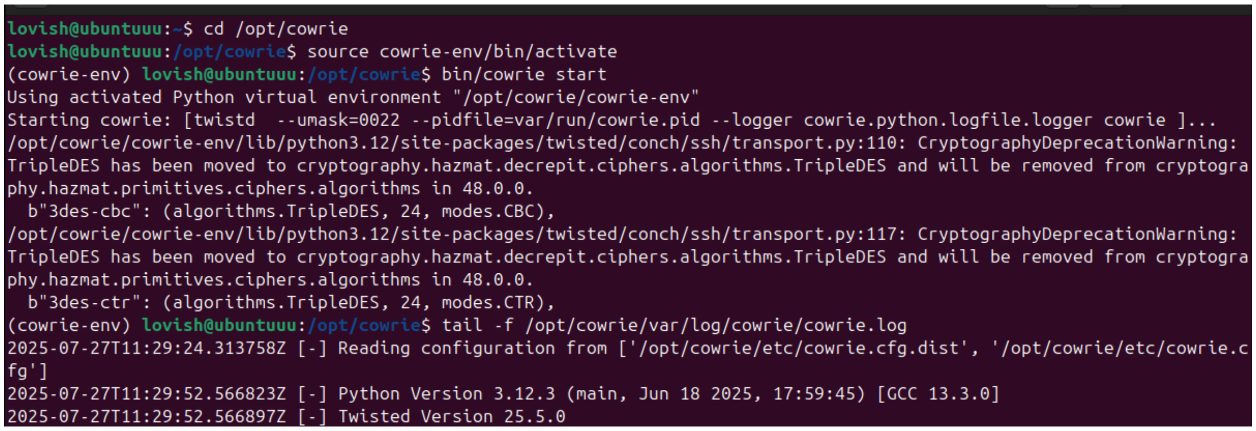


```
(cowrie-env) lovish@ubuntuuu:/opt/cowrie$ cp etc/cowrie.cfg.dist etc/cowrie.cfg
(cowrie-env) lovish@ubuntuuu:/opt/cowrie$ bin/cowrie start

Join the Cowrie community at: https://www.cowrie.org/slack/

Using activated Python virtual environment "/opt/cowrie/cowrie-env"
Starting cowrie: [twistd  --umask=0022 --pidfile=var/run/cowrie.pid --logger cowrie.python.logfile.logger cowrie ]...
/opt/cowrie/cowrie-env/lib/python3.12/site-packages/twisted/conch/ssh/transport.py:110: CryptographyDeprecationWarning:
TripleDES has been moved to cryptography.hazmat.decrepit.ciphers.algorithms.TripleDES and will be removed from cryptogra
phy.hazmat.primitives.ciphers.algorithms in 48.0.0.
  b"3des-cbc": (algorithms.TripleDES, 24, modes.CBC),
/opt/cowrie/cowrie-env/lib/python3.12/site-packages/twisted/conch/ssh/transport.py:117: CryptographyDeprecationWarning:
TripleDES has been moved to cryptography.hazmat.decrepit.ciphers.algorithms.TripleDES and will be removed from cryptogra
phy.hazmat.primitives.ciphers.algorithms in 48.0.0.
  b"3des-ctr": (algorithms.TripleDES, 24, modes.CTR),
```

tail -f /opt/cowrie/var/log/cowrie/cowrie.log

bin/cowrie start

1. **tail:** Displays the end of a file.
2. **-f: Follows the file —** meaning new lines are shown in real time.
3. **/opt/cowrie/var/log/cowrie/cowrie.log**: Path to Cowrie's log file where all attacker activities are recorded

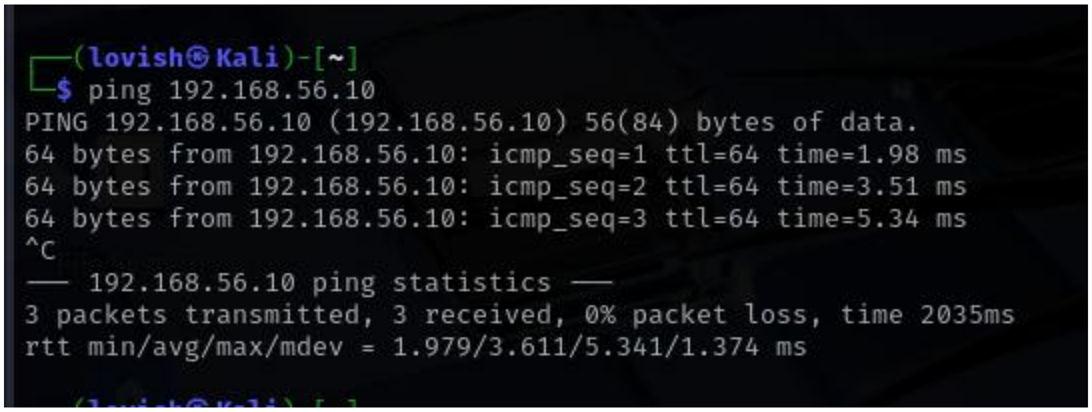


```
lovish@ubuntuuu:~$ cd /opt/cowrie
lovish@ubuntuuu:/opt/cowrie$ source cowrie-env/bin/activate
(cowrie-env) lovish@ubuntuuu:/opt/cowrie$ bin/cowrie start
Using activated Python virtual environment "/opt/cowrie/cowrie-env"
Starting cowrie: [twistd  --umask=0022 --pidfile=var/run/cowrie.pid --logger cowrie.python.logfile.logger cowrie ]...
/opt/cowrie/cowrie-env/lib/python3.12/site-packages/twisted/conch/ssh/transport.py:110: CryptographyDeprecationWarning:
TripleDES has been moved to cryptography.hazmat.decrepit.ciphers.algorithms.TripleDES and will be removed from cryptogra
phy.hazmat.primitives.ciphers.algorithms in 48.0.0.
  b"3des-cbc": (algorithms.TripleDES, 24, modes.CBC),
/opt/cowrie/cowrie-env/lib/python3.12/site-packages/twisted/conch/ssh/transport.py:117: CryptographyDeprecationWarning:
TripleDES has been moved to cryptography.hazmat.decrepit.ciphers.algorithms.TripleDES and will be removed from cryptogra
phy.hazmat.primitives.ciphers.algorithms in 48.0.0.
  b"3des-ctr": (algorithms.TripleDES, 24, modes.CTR),
(cowrie-env) lovish@ubuntuuu:/opt/cowrie$ tail -f /opt/cowrie/var/log/cowrie/cowrie.log
2025-07-27T11:29:24.313758Z [-] Reading configuration from ['/opt/cowrie/etc/cowrie.cfg.dist', '/opt/cowrie/etc/cowrie.c
fg']
2025-07-27T11:29:52.566823Z [-] Python Version 3.12.3 (main, Jun 18 2025, 17:59:45) [GCC 13.3.0]
2025-07-27T11:29:52.566897Z [-] Twisted Version 25.5.0
```

Attack Simulation

ping 192.168.56.10

The ping command is used to check if the target IP address (the Cowrie honeypot in this case) is online and reachable.



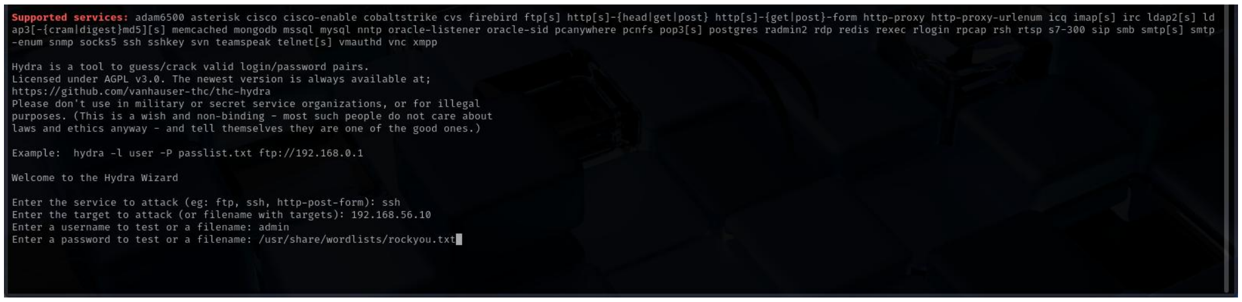
```
┌──(lovish㉿Kali)-[~]
└─$ ping 192.168.56.10
PING 192.168.56.10 (192.168.56.10) 56(84) bytes of data.
64 bytes from 192.168.56.10: icmp_seq=1 ttl=64 time=1.98 ms
64 bytes from 192.168.56.10: icmp_seq=2 ttl=64 time=3.51 ms
64 bytes from 192.168.56.10: icmp_seq=3 ttl=64 time=5.34 ms
^C
── 192.168.56.10 ping statistics ──
3 packets transmitted, 3 received, 0% packet loss, time 2035ms
rtt min/avg/max/mdev = 1.979/3.611/5.341/1.374 ms
```

SSH Brute-Force Attack using Hydra

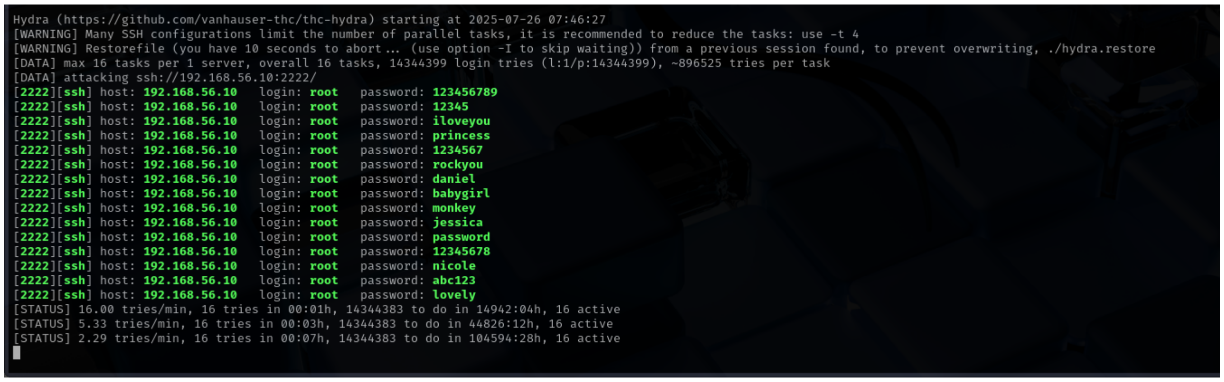
Tool Used: Hydra

hydra -l admin -P /usr/share/wordlists/rockyou.txt ssh://192.168.56.10 -s 2222

1. **-l admin**: Sets the username to try (admin).
2. **-P /usr/share/wordlists/rockyou.txt**: Uses the RockYou password list (a common wordlist) to try different passwords.
3. **ssh://192.168.56.10**: The target IP address and protocol (SSH).
4. **-s 2222**: Specifies port 2222 (Cowrie's fake SSH port).



```
Supported services: adam6500 asterisk cisco cisco-enable cobaltstrike cvs firebird ftp[s] http[s]-{head|get|post} http[s]-{get|post}-form http-proxy http-proxy-urlenum icq imap[s] irc ldap2[s] ld
ap3[-{cram|digest}md5][s] memcached mongodb mssql mysql nntp oracle-listener oracle-sid pcanywhere pcnfs pop3[s] postgres radmin2 rdp redis rexec rlogin rpcap rsh rtsp s7-300 sip smb smtp[s] smtp
-enum snmp socks5 ssh sshkey svn teamspeak telnet[s] vmauthd vnc xmpp

Hydra is a tool to guess/crack valid login/password pairs.
Licensed under AGPL v3.0. The newest version is always available at;
https://github.com/vanhauser-thc/thc-hydra
Please don't use in military or secret service organizations, or for illegal
purposes. (This is a wish and non-binding - most such people do not care about
laws and ethics anyway - and tell themselves they are one of the good ones.)

Example:  hydra -l user -P passlist.txt ftp://192.168.0.1

Welcome to the Hydra Wizard

Enter the service to attack (eg: ftp, ssh, http-post-form): ssh
Enter the target to attack (or filename with targets): 192.168.56.10
Enter a username to test or a filename: admin
Enter a password to test or a filename: /usr/share/wordlists/rockyou.txt
```



```
Hydra (https://github.com/vanhauser-thc/thc-hydra) starting at 2025-07-26 07:46:27
[WARNING] Many SSH configurations limit the number of parallel tasks, it is recommended to reduce the tasks: use -t 4
[WARNING] Restorefile (you have 10 seconds to abort... (use option -I to skip waiting)) from a previous session found, to prevent overwriting, ./hydra.restore
[DATA] max 16 tasks per 1 server, overall 16 tasks, 14344399 login tries (l:1/p:14344399), ~896525 tries per task
[DATA] attacking ssh://192.168.56.10:2222/
[2222][ssh] host: 192.168.56.10   login: root   password: 123456789
[2222][ssh] host: 192.168.56.10   login: root   password: 12345
[2222][ssh] host: 192.168.56.10   login: root   password: iloveyou
[2222][ssh] host: 192.168.56.10   login: root   password: princess
[2222][ssh] host: 192.168.56.10   login: root   password: 1234567
[2222][ssh] host: 192.168.56.10   login: root   password: rockyou
[2222][ssh] host: 192.168.56.10   login: root   password: daniel
[2222][ssh] host: 192.168.56.10   login: root   password: babygirl
[2222][ssh] host: 192.168.56.10   login: root   password: monkey
[2222][ssh] host: 192.168.56.10   login: root   password: jessica
[2222][ssh] host: 192.168.56.10   login: root   password: password
[2222][ssh] host: 192.168.56.10   login: root   password: 12345678
[2222][ssh] host: 192.168.56.10   login: root   password: nicole
[2222][ssh] host: 192.168.56.10   login: root   password: abc123
[2222][ssh] host: 192.168.56.10   login: root   password: lovely
[STATUS] 16.00 tries/min, 16 tries in 00:01h, 14344383 to do in 14942:04h, 16 active
[STATUS] 5.33 tries/min, 16 tries in 00:03h, 14344383 to do in 44826:12h, 16 active
[STATUS] 2.29 tries/min, 16 tries in 00:07h, 14344383 to do in 104594:28h, 16 active
```

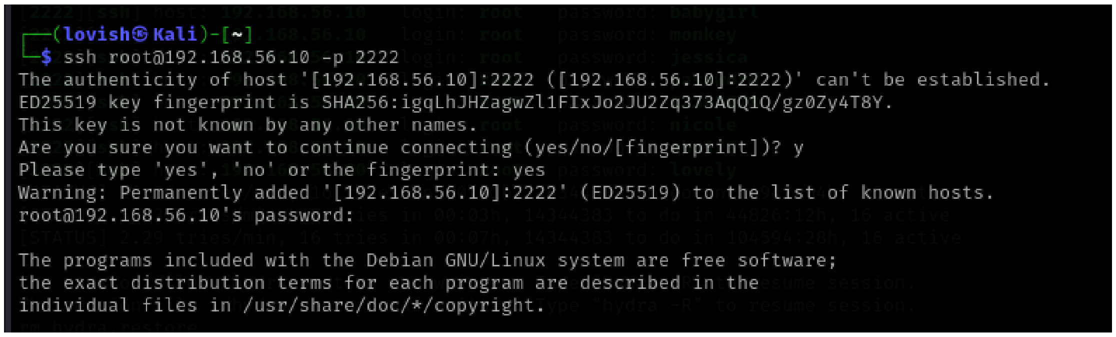
Network Scanning using Nmap

Tool Used: Nmap

Target: Cowrie Honeypot

IP Address: 192.168.56.10

nmap 192.168.56.10

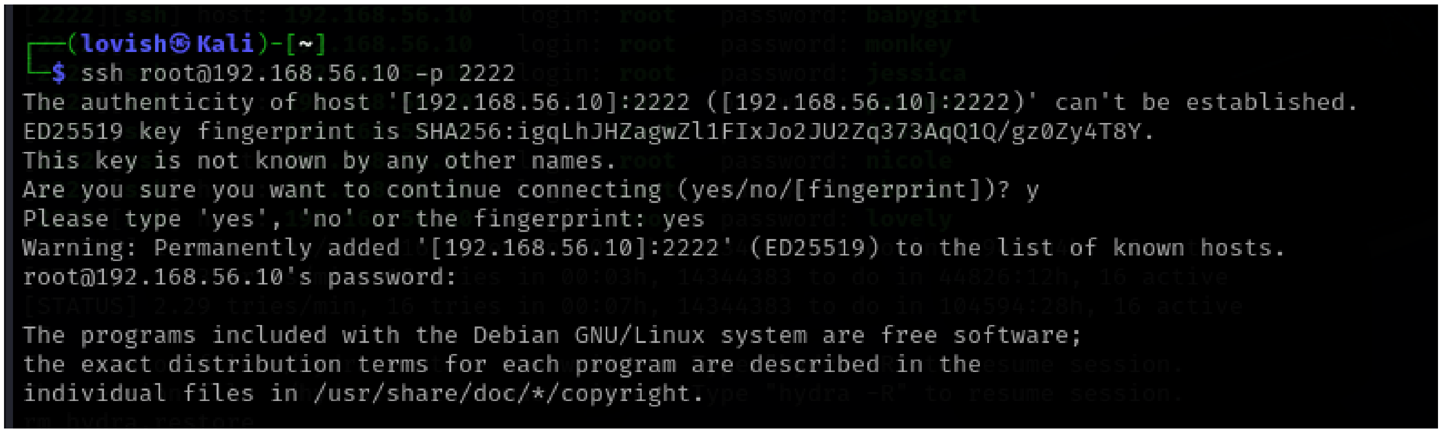


```
┌──(lovish㉿Kali)-[~]
└─$ ssh root@192.168.56.10 -p 2222
The authenticity of host '[192.168.56.10]:2222 ([192.168.56.10]:2222)' can't be established.
ED25519 key fingerprint is SHA256:igqLhJHZagwZl1FIxJo2JU2Zq373AqQ1Q/gz0Zy4T8Y.
This key is not known by any other names.
Are you sure you want to continue connecting (yes/no/[fingerprint])? y
Please type 'yes', 'no' or the fingerprint: yes
Warning: Permanently added '[192.168.56.10]:2222' (ED25519) to the list of known hosts.
root@192.168.56.10'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.
```

SSH Access Attempt — Successful Login

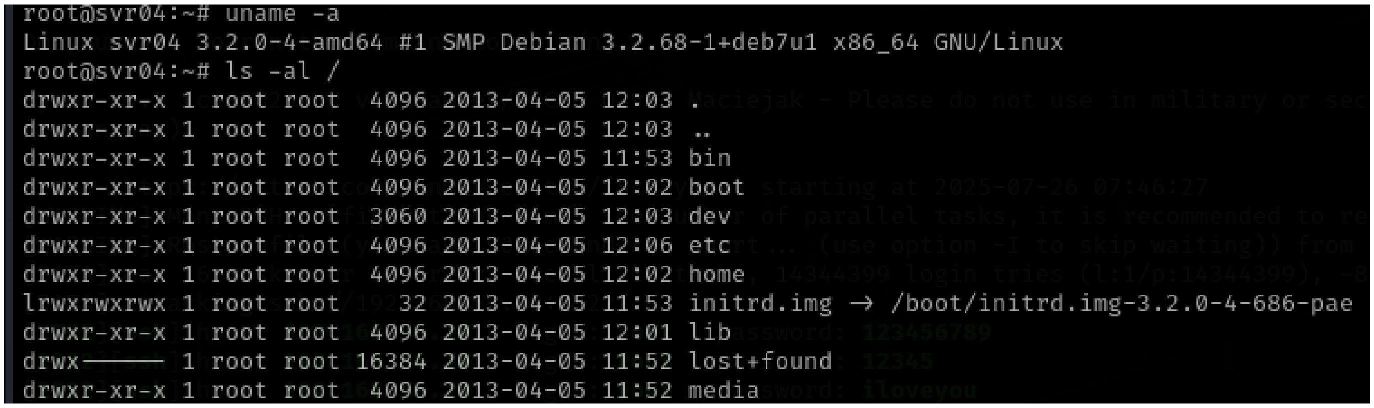
ssh root@192.168.56.10 -p 2222

1. The **Cowrie honeypot accepted your login** (after brute-force or manual guessing).
2. You are now in a **fake shell** emulated by Cowrie — any command you type is logged.



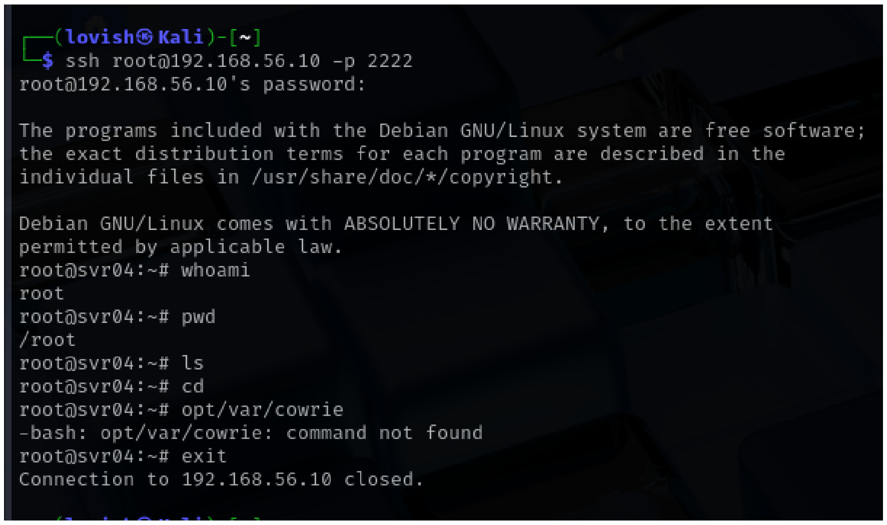
```
┌──(lovish㉿Kali)-[~]
└─$ ssh root@192.168.56.10 -p 2222
The authenticity of host '[192.168.56.10]:2222 ([192.168.56.10]:2222)' can't be established.
ED25519 key fingerprint is SHA256:igqLhJHZagwZl1FIxJo2JU2Zq373AqQ1Q/gz0Zy4T8Y.
This key is not known by any other names.
Are you sure you want to continue connecting (yes/no/[fingerprint])? y
Please type 'yes', 'no' or the fingerprint: yes
Warning: Permanently added '[192.168.56.10]:2222' (ED25519) to the list of known hosts.
root@192.168.56.10'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.
```

ls -al /



```
root@svr04:~# uname -a
Linux svr04 3.2.0-4-amd64 #1 SMP Debian 3.2.68-1+deb7u1 x86_64 GNU/Linux
root@svr04:~# ls -al /
drwxr-xr-x 1 root root  4096 2013-04-05 12:03 .
drwxr-xr-x 1 root root  4096 2013-04-05 12:03 ..
drwxr-xr-x 1 root root  4096 2013-04-05 11:53 bin
drwxr-xr-x 1 root root  4096 2013-04-05 12:02 boot
drwxr-xr-x 1 root root  3060 2013-04-05 12:03 dev
drwxr-xr-x 1 root root  4096 2013-04-05 12:06 etc
drwxr-xr-x 1 root root  4096 2013-04-05 12:02 home
lrwxrwxrwx 1 root root    32 2013-04-05 11:53 initrd.img → /boot/initrd.img-3.2.0-4-686-pae
drwxr-xr-x 1 root root  4096 2013-04-05 12:01 lib
drwx────── 1 root root 16384 2013-04-05 11:52 lost+found
drwxr-xr-x 1 root root  4096 2013-04-05 11:52 media
```

Whoami, pwd ,ifconfig ,ls , cd /var/log && ls



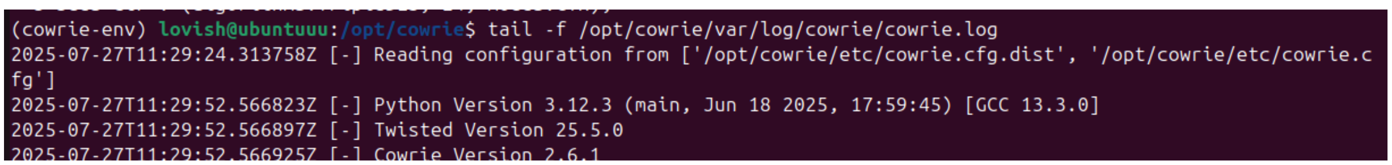
```
┌──(lovish㉿Kali)-[~]
└─$ ssh root@192.168.56.10 -p 2222
root@192.168.56.10's password:

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@svr04:~# whoami
root
root@svr04:~# pwd
/root
root@svr04:~# ls
root@svr04:~# cd
root@svr04:~# opt/var/cowrie
-bash: opt/var/cowrie: command not found
root@svr04:~# exit
Connection to 192.168.56.10 closed.
```

Results and Analysis

- What did the honeypot capture?

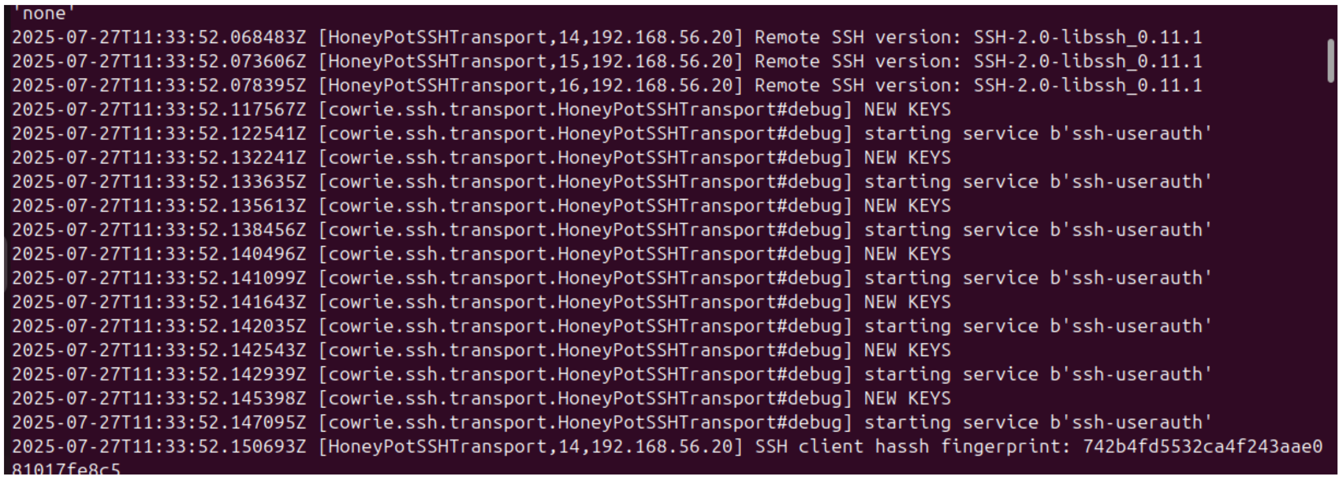
tail -f /opt/cowrie/var/log/cowrie/cowrie.log

To capture all log in plain text



```
(cowrie-env) lovish@ubuntuuu:/opt/cowrie$ tail -f /opt/cowrie/var/log/cowrie/cowrie.log
2025-07-27T11:29:24.313758Z [-] Reading configuration from ['/opt/cowrie/etc/cowrie.cfg.dist', '/opt/cowrie/etc/cowrie.c
fg']
2025-07-27T11:29:52.566823Z [-] Python Version 3.12.3 (main, Jun 18 2025, 17:59:45) [GCC 13.3.0]
2025-07-27T11:29:52.566897Z [-] Twisted Version 25.5.0
2025-07-27T11:29:52.569925Z [-] Cowrie Version 2.6.1
```

1. Entries like NEW KEYS and starting service b'ssh-userauth' indicate key exchange and the start of the SSH user authentication process.
2. The log captures the **handshake** steps (like exchanging keys), which is part of setting up a secure SSH session.
3. The hassh fingerprint entry is a unique fingerprint representing the SSH client software, useful for identifying attack tools.

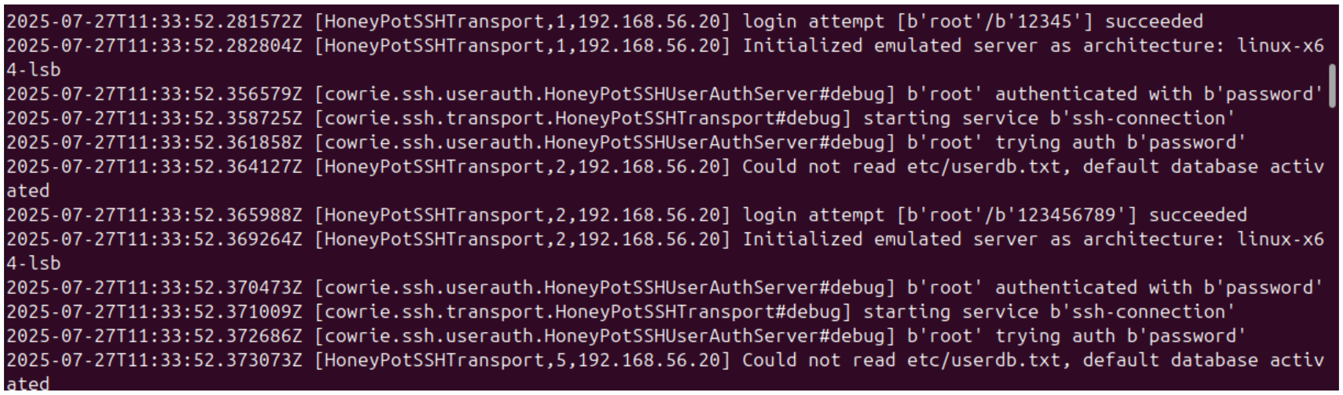


```
'none'
2025-07-27T11:33:52.068483Z [HoneyPotSSHTransport,14,192.168.56.20] Remote SSH version: SSH-2.0-libssh_0.11.1
2025-07-27T11:33:52.073606Z [HoneyPotSSHTransport,15,192.168.56.20] Remote SSH version: SSH-2.0-libssh_0.11.1
2025-07-27T11:33:52.078395Z [HoneyPotSSHTransport,16,192.168.56.20] Remote SSH version: SSH-2.0-libssh_0.11.1
2025-07-27T11:33:52.117567Z [cowrie.ssh.transport.HoneyPotSSHTransport#debug] NEW KEYS
2025-07-27T11:33:52.122541Z [cowrie.ssh.transport.HoneyPotSSHTransport#debug] starting service b'ssh-userauth'
2025-07-27T11:33:52.132241Z [cowrie.ssh.transport.HoneyPotSSHTransport#debug] NEW KEYS
2025-07-27T11:33:52.133635Z [cowrie.ssh.transport.HoneyPotSSHTransport#debug] starting service b'ssh-userauth'
2025-07-27T11:33:52.135613Z [cowrie.ssh.transport.HoneyPotSSHTransport#debug] NEW KEYS
2025-07-27T11:33:52.138456Z [cowrie.ssh.transport.HoneyPotSSHTransport#debug] starting service b'ssh-userauth'
2025-07-27T11:33:52.140496Z [cowrie.ssh.transport.HoneyPotSSHTransport#debug] NEW KEYS
2025-07-27T11:33:52.141099Z [cowrie.ssh.transport.HoneyPotSSHTransport#debug] starting service b'ssh-userauth'
2025-07-27T11:33:52.141643Z [cowrie.ssh.transport.HoneyPotSSHTransport#debug] NEW KEYS
2025-07-27T11:33:52.142035Z [cowrie.ssh.transport.HoneyPotSSHTransport#debug] starting service b'ssh-userauth'
2025-07-27T11:33:52.142543Z [cowrie.ssh.transport.HoneyPotSSHTransport#debug] NEW KEYS
2025-07-27T11:33:52.142939Z [cowrie.ssh.transport.HoneyPotSSHTransport#debug] starting service b'ssh-userauth'
2025-07-27T11:33:52.145398Z [cowrie.ssh.transport.HoneyPotSSHTransport#debug] NEW KEYS
2025-07-27T11:33:52.147095Z [cowrie.ssh.transport.HoneyPotSSHTransport#debug] starting service b'ssh-userauth'
2025-07-27T11:33:52.150693Z [HoneyPotSSHTransport,14,192.168.56.20] SSH client hassh fingerprint: 742b4fd5532ca4f243aae0
81017fe8c5
```

Cowrie Log: Login Attempts and Authentication

1. **login attempt [b'root'/b'12345']**
   An attacker tried to login using the username root and password 12345.
2. **b'root' authenticated with b'pass'**
   The honeypot accepted the login, simulating a successful authentication.
3. **starting service b'ssh-connection'**
   Cowrie started a fake SSH session to mimic a real server connection.
4. **login attempt [b'root'/b'123456789'] succeeded**
   Another password guess for user root succeeded, showing multiple brute-force attempts.
5. **Could not read etc/userdb.txt, default database**
   Cowrie tried to read its user database but didn't find it, so it used the default settings.

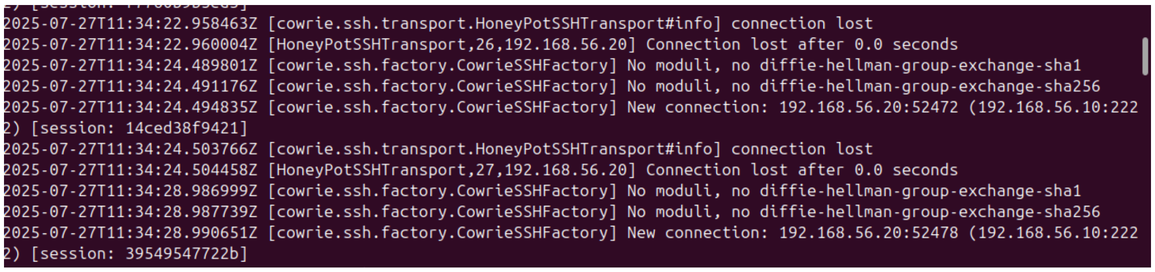
6. **Initialized emulated server as architecture: linux**
   Cowrie sets up a fake Linux environment for the attacker after successful login.



```
2025-07-27T11:33:52.281572Z [HoneyPotSSHTransport,1,192.168.56.20] login attempt [b'root'/b'12345'] succeeded
2025-07-27T11:33:52.282804Z [HoneyPotSSHTransport,1,192.168.56.20] Initialized emulated server as architecture: linux-x6
4-lsb
2025-07-27T11:33:52.356579Z [cowrie.ssh.userauth.HoneyPotSSHUserAuthServer#debug] b'root' authenticated with b'password'
2025-07-27T11:33:52.358725Z [cowrie.ssh.transport.HoneyPotSSHTransport#debug] starting service b'ssh-connection'
2025-07-27T11:33:52.361858Z [cowrie.ssh.userauth.HoneyPotSSHUserAuthServer#debug] b'root' trying auth b'password'
2025-07-27T11:33:52.364127Z [HoneyPotSSHTransport,2,192.168.56.20] Could not read etc/userdb.txt, default database activ
ated
2025-07-27T11:33:52.365988Z [HoneyPotSSHTransport,2,192.168.56.20] login attempt [b'root'/b'123456789'] succeeded
2025-07-27T11:33:52.369264Z [HoneyPotSSHTransport,2,192.168.56.20] Initialized emulated server as architecture: linux-x6
4-lsb
2025-07-27T11:33:52.370473Z [cowrie.ssh.userauth.HoneyPotSSHUserAuthServer#debug] b'root' authenticated with b'password'
2025-07-27T11:33:52.371009Z [cowrie.ssh.transport.HoneyPotSSHTransport#debug] starting service b'ssh-connection'
2025-07-27T11:33:52.372686Z [cowrie.ssh.userauth.HoneyPotSSHUserAuthServer#debug] b'root' trying auth b'password'
2025-07-27T11:33:52.373073Z [HoneyPotSSHTransport,5,192.168.56.20] Could not read etc/userdb.txt, default database activ
ated
```

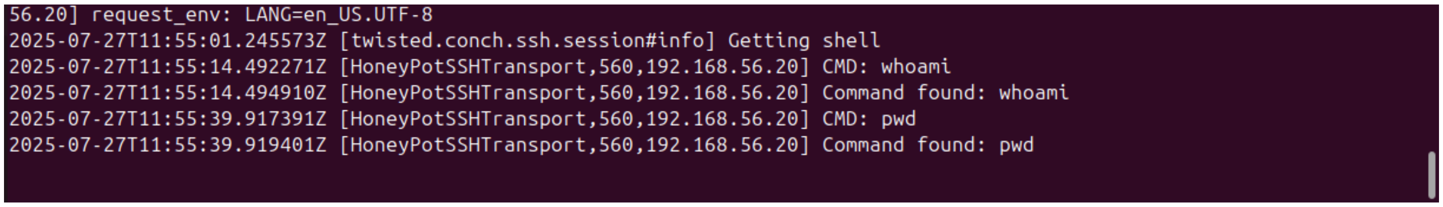
Cowrie Log: Connection Lost and New Connections

1. **connection lost** and **Connection lost after 0.0 seconds**
   These messages mean that the SSH connection from the client at IP 192.168.56.20 was closed very quickly.
   This could be due to the attacker disconnecting, a failed attempt, or network issues.



```
2025-07-27T11:34:22.958463Z [cowrie.ssh.transport.HoneyPotSSHTransport#info] connection lost
2025-07-27T11:34:22.960004Z [HoneyPotSSHTransport,26,192.168.56.20] Connection lost after 0.0 seconds
2025-07-27T11:34:24.489801Z [cowrie.ssh.factory.CowrieSSHFactory] No moduli, no diffie-hellman-group-exchange-sha1
2025-07-27T11:34:24.491176Z [cowrie.ssh.factory.CowrieSSHFactory] No moduli, no diffie-hellman-group-exchange-sha256
2025-07-27T11:34:24.494835Z [cowrie.ssh.factory.CowrieSSHFactory] New connection: 192.168.56.20:52472 (192.168.56.10:222
2) [session: 14ced38f9421]
2025-07-27T11:34:24.503766Z [cowrie.ssh.transport.HoneyPotSSHTransport#info] connection lost
2025-07-27T11:34:24.504458Z [HoneyPotSSHTransport,27,192.168.56.20] Connection lost after 0.0 seconds
2025-07-27T11:34:28.986999Z [cowrie.ssh.factory.CowrieSSHFactory] No moduli, no diffie-hellman-group-exchange-sha1
2025-07-27T11:34:28.987739Z [cowrie.ssh.factory.CowrieSSHFactory] No moduli, no diffie-hellman-group-exchange-sha256
2025-07-27T11:34:28.990651Z [cowrie.ssh.factory.CowrieSSHFactory] New connection: 192.168.56.20:52478 (192.168.56.10:222
2) [session: 39549547722b]
```

commands attacker tried



```
56.20] request_env: LANG=en_US.UTF-8
2025-07-27T11:55:01.245573Z [twisted.conch.ssh.session#info] Getting shell
2025-07-27T11:55:14.492271Z [HoneyPotSSHTransport,560,192.168.56.20] CMD: whoami
2025-07-27T11:55:14.494910Z [HoneyPotSSHTransport,560,192.168.56.20] Command found: whoami
2025-07-27T11:55:39.917391Z [HoneyPotSSHTransport,560,192.168.56.20] CMD: pwd
2025-07-27T11:55:39.919401Z [HoneyPotSSHTransport,560,192.168.56.20] Command found: pwd
```

Conclusion

Summary of What I Learned

Through this project, I learned:

- How to **deploy and configure Cowrie**, a medium-interaction SSH honeypot, on a Ubuntu VM.

- How to **simulate real-world attacks** like SSH brute-force, command execution, and scanning from a Kali Linux attacker machine.
- How Cowrie **captures logs** of attacker behavior (e.g., commands like ls, whoami, cat /etc/passwd) without giving real access.
- How to view and interpret Cowrie logs in JSON and text formats.
- How to use tools like **Hydra** and **Nmap** to simulate attacks and monitor responses.