Lab: Investigate Logs using Cowrie

In this lab you will explore how you can investigate the logs to perform digital forensics on a system.

Digital Forensics

Skills Network

Estimated time needed: 15 minutes

In this lab you will explore how you can investigate the logs to perform digital foresics on a system.

Learning Objectives

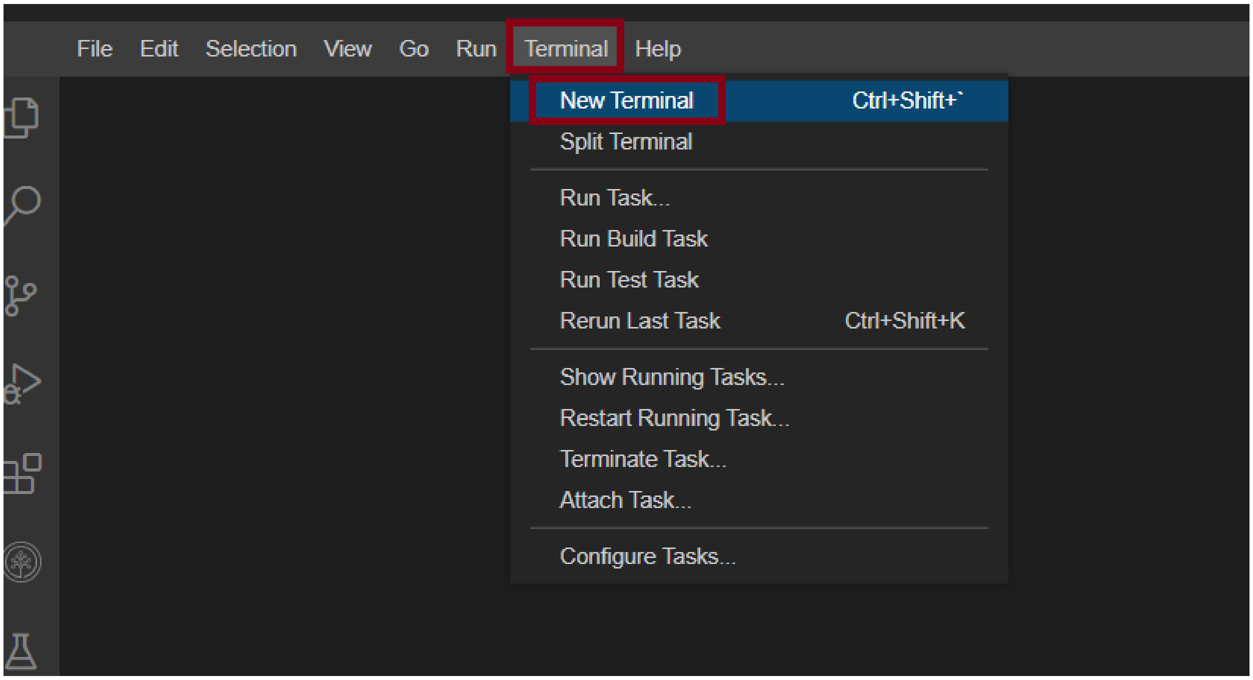
At the end of this lab you will be able to:

- Analyze the logs
- Use ssh to connect and analyze a remote system

Run Cowrie on docker

Cowrie is an SSH and Telnet honeypot which allows interactions designed to log brute force attacks and shell interaction, which in a real system is often performed by an attacker. This helps to analyze the time of attack, type of commands run by the attacker to perform the attack, other security issues and data breaches caused when the system was attacked, etc.,

1. Open a new terminal.



2. In the new terminal window, run the following command to run cowrie in a dockerized container.

```
docker run -p 2222:2222/tcp cowrie/cowrie > honeypotLogs.txt
```

Copied!Wrap Toggled!Executed!

This will start Cowrie and redirect all the logs which are printed on screen to the file named `honeypotLogs.txt` in your local machine.

Telnet to Cowrie

1. Open another terminal.

2. From the terminal, run the following command to telnet to Cowrie through `ssh`.

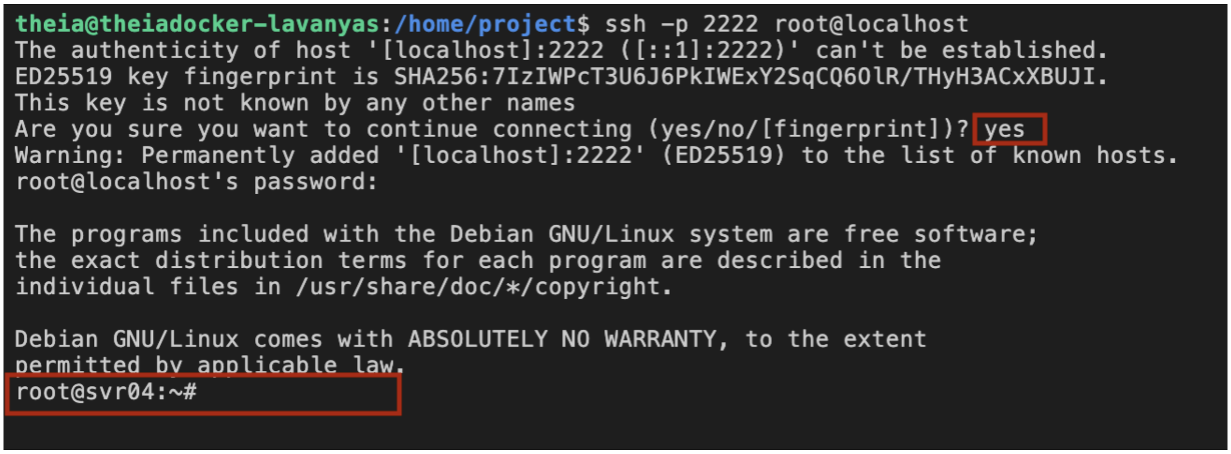
```
ssh -p 2222 root@localhost
```

Copied!Wrap Toggled!Executed!

3. When prompted to confirm `Are you sure you want to continue connecting?`, type **yes** and press `enter`.
4. When prompted for `root@localhost` password, type **admin** and press enter.

The connection will now be established and you will see the root prompt.

This session times out in 3-4 minutes if not used.



```
theia@theiadocker-lavanyas:/home/project$ ssh -p 2222 root@localhost
The authenticity of host '[localhost]:2222 ([::1]:2222)' can't be established.
ED25519 key fingerprint is SHA256:7IzIWPcT3U6J6PkIWExY2SqCQ6OlR/THyH3ACxXBUJI.
This key is not known by any other names
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added '[localhost]:2222' (ED25519) to the list of known hosts.
root@localhost'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:~#
```

5. Now you will be connected to Cowrie and can run commands that can be executed on Cowrie. Run the following command in the `ssh` terminal.

```
ls
```

Copied!Wrap Toggled!Executed!

6. Create a file in the remote system by redirecting echo into a file.

```
echo "The is a file created through ssh">newfilefromRemote.txt
```

Copied!Wrap Toggled!Executed!

7. Check if the file is created and its content.

```
cat newfilefromRemote.txt
```

Copied!Wrap Toggled!Executed!

You should be able to see the content of the file.

8. Delete the file by running the following command.

```
rm -f newfilefromRemote.txt
```

Analyze logs

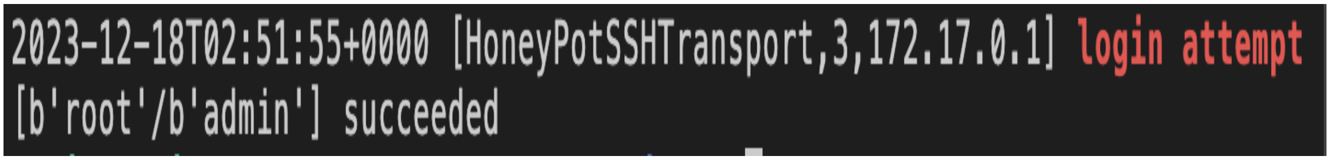
1. Open a new Terminal.

2. In the terminal, run the following commands to check when the ssh request was received.

```
cat honeypotLogs.txt | grep "login attempt"
```

Copied!Wrap Toggled!Executed!

You will see the details about one or more login attempts to Cowrie as shown in the image below.



```
2023-12-18T02:51:55+0000 [HoneyPotSSHTransport,3,172.17.0.1] login attempt
[b'root'/b'admin'] succeeded
```

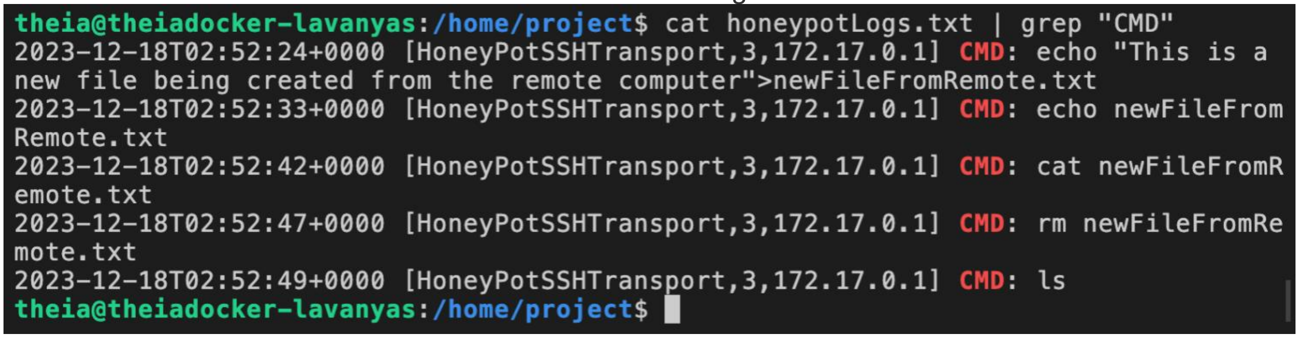
3. Run the following command to check the logs for all the commands executed on the remote terminal.

1. 1

```
cat honeypotLogs.txt | grep "CMD"
```

Copied!Wrap Toggled!Executed!

This will list all the commands executed in the ssh shell as given below.



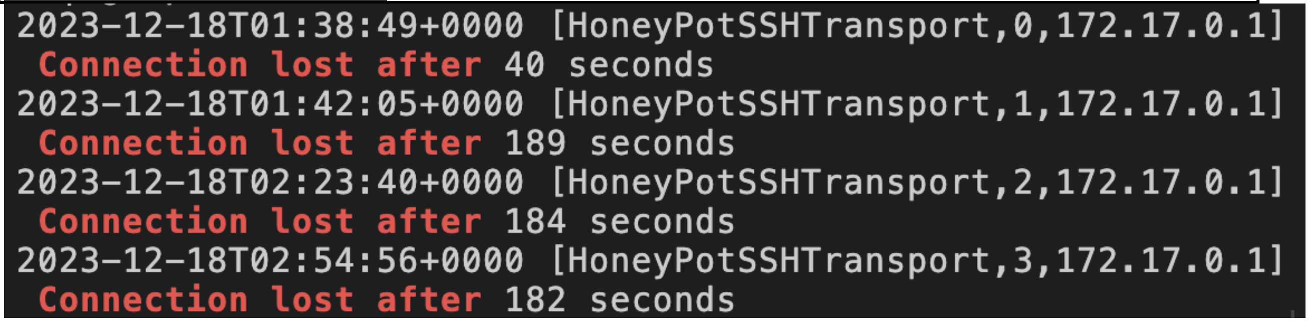
```
theia@theiadocker-lavanyas:/home/project$ cat honeypotLogs.txt | grep "CMD"
2023-12-18T02:52:24+0000 [HoneyPotSSHTransport,3,172.17.0.1] CMD: echo "This is a
new file being created from the remote computer">newFileFromRemote.txt
2023-12-18T02:52:33+0000 [HoneyPotSSHTransport,3,172.17.0.1] CMD: echo newFileFrom
Remote.txt
2023-12-18T02:52:42+0000 [HoneyPotSSHTransport,3,172.17.0.1] CMD: cat newFileFromR
emote.txt
2023-12-18T02:52:47+0000 [HoneyPotSSHTransport,3,172.17.0.1] CMD: rm newFileFromRe
mote.txt
2023-12-18T02:52:49+0000 [HoneyPotSSHTransport,3,172.17.0.1] CMD: ls
theia@theiadocker-lavanyas:/home/project$ 
```

4. Wait for 3 to 4 mins and execute the following command to see how the remote host closes the connection automatically.

1. 1

```
cat honeypotLogs.txt | grep "Connection lost after"
```

Copied!Wrap Toggled!Executed!



```
2023-12-18T01:38:49+0000 [HoneyPotSSHTransport,0,172.17.0.1]
 Connection lost after 40 seconds
2023-12-18T01:42:05+0000 [HoneyPotSSHTransport,1,172.17.0.1]
 Connection lost after 189 seconds
2023-12-18T02:23:40+0000 [HoneyPotSSHTransport,2,172.17.0.1]
 Connection lost after 184 seconds
2023-12-18T02:54:56+0000 [HoneyPotSSHTransport,3,172.17.0.1]
 Connection lost after 182 seconds
```

Terminate Cowrie

1. Go to the original shell where you created the docker instance of Cowrie.

2. Press `ctrl + c` to terminate the instance.

Practice Exercises:

1. Run another Cowrie instance with docker.

1. 1

```
docker run -p 2223:2222/tcp cowrie/cowrie >> honeypotLogs.txt
```

Copied!Wrap Toggled!

Note: The `>>` *ensures that the logs are appended and not over written.*

3. Telnet from 2 different terminal to the instance using same credentials **admin**.

4. Run some commands in both. Analyze the logs in honeypotlogs.txt.

5. Run the following command in a new terminal to see how long it takes for the telnet connections to be terminated.

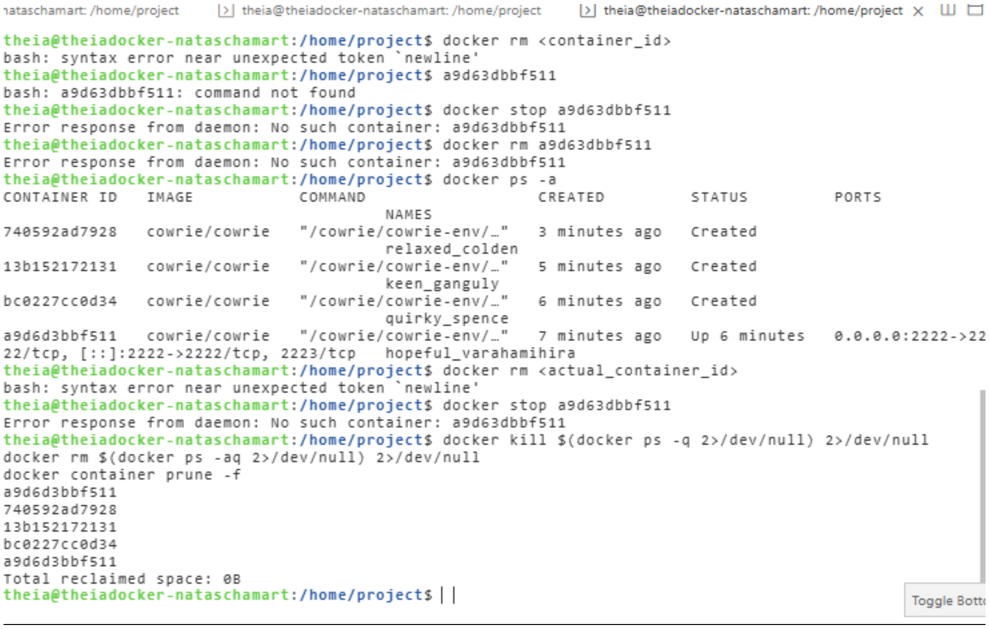
1. 1

```
tail -f honeypotLogs.txt
```

Copied!Wrap Toggled!

Here you are simulating analyzing logs of multiple telnet connections.

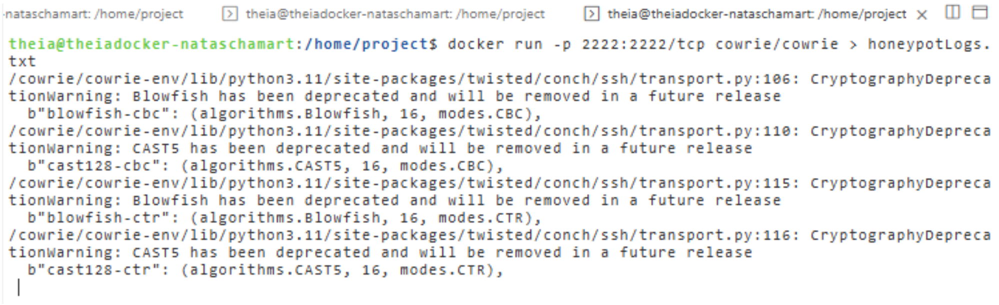
My WORK


nataschamart: /home/project
theia@theiadocker-nataschamart: /home/project
theia@theiadocker-nataschamart: /home/project
Toggle Bott

```
theia@theiadocker-nataschamart:/home/project$ docker rm <container_id>
bash: syntax error near unexpected token `newline'
theia@theiadocker-nataschamart:/home/project$ a9d63dbbf511
bash: a9d63dbbf511: command not found
theia@theiadocker-nataschamart:/home/project$ docker stop a9d63dbbf511
Error response from daemon: No such container: a9d63dbbf511
theia@theiadocker-nataschamart:/home/project$ docker rm a9d63dbbf511
Error response from daemon: No such container: a9d63dbbf511
theia@theiadocker-nataschamart:/home/project$ docker ps -a
CONTAINER ID   IMAGE           COMMAND                  CREATED         STATUS         PORTS
                                         NAMES
740592ad7928   cowrie/cowrie   "/cowrie/cowrie-env/…"   3 minutes ago   Created
                                         relaxed_colden
13b152172131   cowrie/cowrie   "/cowrie/cowrie-env/…"   5 minutes ago   Created
                                         keen_ganguly
bc0227cc0d34   cowrie/cowrie   "/cowrie/cowrie-env/…"   6 minutes ago   Created
                                         quirky_spence
a9d6d3bbf511   cowrie/cowrie   "/cowrie/cowrie-env/…"   7 minutes ago   Up 6 minutes   0.0.0.0:2222->22
22/tcp, [::]:2222->2222/tcp, 2223/tcp    hopeful_varahamihira
theia@theiadocker-nataschamart:/home/project$ docker rm <actual_container_id>
bash: syntax error near unexpected token `newline'
theia@theiadocker-nataschamart:/home/project$ docker stop a9d63dbbf511
Error response from daemon: No such container: a9d63dbbf511
theia@theiadocker-nataschamart:/home/project$ docker kill $(docker ps -q 2>/dev/null) 2>/dev/null
docker rm $(docker ps -aq 2>/dev/null) 2>/dev/null
docker container prune -f
a9d6d3bbf511
740592ad7928
13b152172131
bc0227cc0d34
a9d6d3bbf511
Total reclaimed space: 0B
theia@theiadocker-nataschamart:/home/project$ ||
```

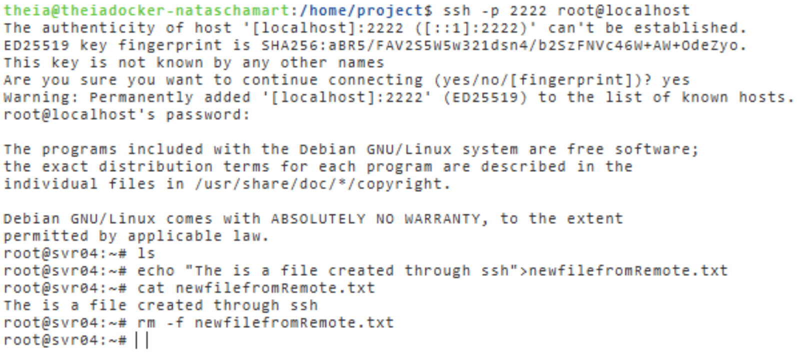
I accidently typed the commands into the main terminal when the lab clearly said to open a new terminal. I had to utilize AI to help me get out of the problem.

Run Cowrie on docker


-nataschamart: /home/project
theia@theiadocker-nataschamart: /home/project
theia@theiadocker-nataschamart: /home/project

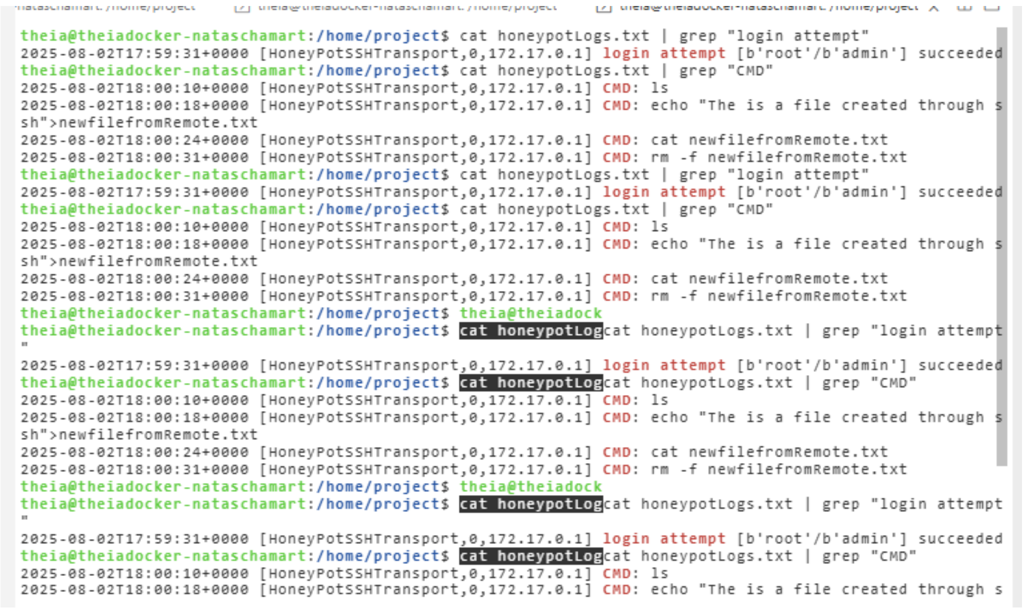
```
theia@theiadocker-nataschamart:/home/project$ docker run -p 2222:2222/tcp cowrie/cowrie > honeypotLogs.
txt
/cowrie/cowrie-env/lib/python3.11/site-packages/twisted/conch/ssh/transport.py:106: CryptographyDepreca
tionWarning: Blowfish has been deprecated and will be removed in a future release
  b"blowfish-cbc": (algorithms.Blowfish, 16, modes.CBC),
/cowrie/cowrie-env/lib/python3.11/site-packages/twisted/conch/ssh/transport.py:110: CryptographyDepreca
tionWarning: CAST5 has been deprecated and will be removed in a future release
  b"cast128-cbc": (algorithms.CAST5, 16, modes.CBC),
/cowrie/cowrie-env/lib/python3.11/site-packages/twisted/conch/ssh/transport.py:115: CryptographyDepreca
tionWarning: Blowfish has been deprecated and will be removed in a future release
  b"blowfish-ctr": (algorithms.Blowfish, 16, modes.CTR),
/cowrie/cowrie-env/lib/python3.11/site-packages/twisted/conch/ssh/transport.py:116: CryptographyDepreca
tionWarning: CAST5 has been deprecated and will be removed in a future release
  b"cast128-ctr": (algorithms.CAST5, 16, modes.CTR),
 |
```

Telnet to Cowrie

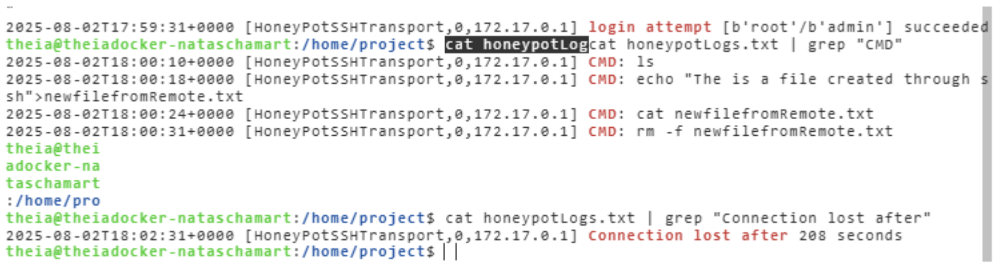


```
theia@theiadocker-nataschamart:/home/project$ ssh -p 2222 root@localhost
The authenticity of host '[localhost]:2222 ([::1]:2222)' can't be established.
ED25519 key fingerprint is SHA256:aBR5/FAV2S5W5w321dsn4/b2SzFNVc46W+AW+OdeZyo.
This key is not known by any other names
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added '[localhost]:2222' (ED25519) to the list of known hosts.
root@localhost'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:~# ls
root@svr04:~# echo "The is a file created through ssh">newfilefromRemote.txt
root@svr04:~# cat newfilefromRemote.txt
The is a file created through ssh
root@svr04:~# rm -f newfilefromRemote.txt
root@svr04:~# ||
```

Analyze logs

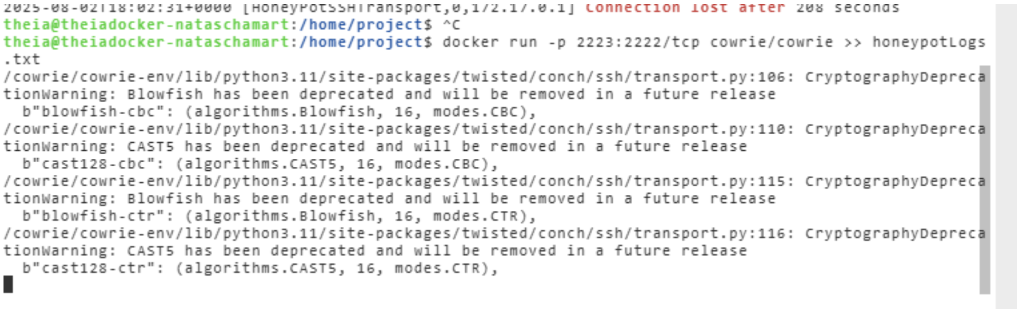


```
theia@theiadocker-nataschamart:/home/project$ cat honeypotLogs.txt | grep "login attempt"
2025-08-02T17:59:31+0000 [HoneyPotSSHTransport,0,172.17.0.1] login attempt [b'root'/b'admin'] succeeded
theia@theiadocker-nataschamart:/home/project$ cat honeypotLogs.txt | grep "CMD"
2025-08-02T18:00:10+0000 [HoneyPotSSHTransport,0,172.17.0.1] CMD: ls
2025-08-02T18:00:18+0000 [HoneyPotSSHTransport,0,172.17.0.1] CMD: echo "The is a file created through s
sh">newfilefromRemote.txt
2025-08-02T18:00:24+0000 [HoneyPotSSHTransport,0,172.17.0.1] CMD: cat newfilefromRemote.txt
2025-08-02T18:00:31+0000 [HoneyPotSSHTransport,0,172.17.0.1] CMD: rm -f newfilefromRemote.txt
theia@theiadocker-nataschamart:/home/project$ cat honeypotLogs.txt | grep "login attempt"
2025-08-02T17:59:31+0000 [HoneyPotSSHTransport,0,172.17.0.1] login attempt [b'root'/b'admin'] succeeded
theia@theiadocker-nataschamart:/home/project$ cat honeypotLogs.txt | grep "CMD"
2025-08-02T18:00:10+0000 [HoneyPotSSHTransport,0,172.17.0.1] CMD: ls
2025-08-02T18:00:18+0000 [HoneyPotSSHTransport,0,172.17.0.1] CMD: echo "The is a file created through s
sh">newfilefromRemote.txt
2025-08-02T18:00:24+0000 [HoneyPotSSHTransport,0,172.17.0.1] CMD: cat newfilefromRemote.txt
2025-08-02T18:00:31+0000 [HoneyPotSSHTransport,0,172.17.0.1] CMD: rm -f newfilefromRemote.txt
theia@theiadocker-nataschamart:/home/project$ theia@theiadock
theia@theiadocker-nataschamart:/home/project$ cat honeypotLogcat honeypotLogs.txt | grep "login attempt
"

2025-08-02T17:59:31+0000 [HoneyPotSSHTransport,0,172.17.0.1] login attempt [b'root'/b'admin'] succeeded
theia@theiadocker-nataschamart:/home/project$ cat honeypotLogcat honeypotLogs.txt | grep "CMD"
2025-08-02T18:00:10+0000 [HoneyPotSSHTransport,0,172.17.0.1] CMD: ls
2025-08-02T18:00:18+0000 [HoneyPotSSHTransport,0,172.17.0.1] CMD: echo "The is a file created through s
sh">newfilefromRemote.txt
2025-08-02T18:00:24+0000 [HoneyPotSSHTransport,0,172.17.0.1] CMD: cat newfilefromRemote.txt
2025-08-02T18:00:31+0000 [HoneyPotSSHTransport,0,172.17.0.1] CMD: rm -f newfilefromRemote.txt
theia@theiadocker-nataschamart:/home/project$ theia@theiadock
theia@theiadocker-nataschamart:/home/project$ cat honeypotLogcat honeypotLogs.txt | grep "login attempt
"

2025-08-02T17:59:31+0000 [HoneyPotSSHTransport,0,172.17.0.1] login attempt [b'root'/b'admin'] succeeded
theia@theiadocker-nataschamart:/home/project$ cat honeypotLogcat honeypotLogs.txt | grep "CMD"
2025-08-02T18:00:10+0000 [HoneyPotSSHTransport,0,172.17.0.1] CMD: ls
2025-08-02T18:00:18+0000 [HoneyPotSSHTransport,0,172.17.0.1] CMD: echo "The is a file created through s
```

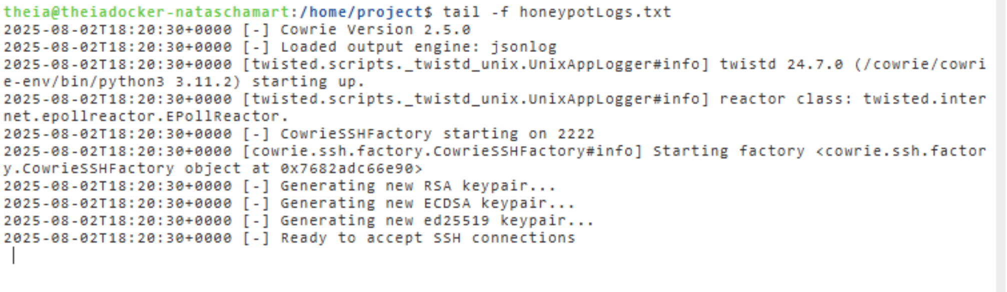


```
2025-08-02T17:59:31+0000 [HoneyPotSSHTransport,0,172.17.0.1] login attempt [b'root'/b'admin'] succeeded
theia@theiadocker-nataschamart:/home/project$ cat honeypotLogcat honeypotLogs.txt | grep "CMD"
2025-08-02T18:00:10+0000 [HoneyPotSSHTransport,0,172.17.0.1] CMD: ls
2025-08-02T18:00:18+0000 [HoneyPotSSHTransport,0,172.17.0.1] CMD: echo "The is a file created through s
sh">newfilefromRemote.txt
2025-08-02T18:00:24+0000 [HoneyPotSSHTransport,0,172.17.0.1] CMD: cat newfilefromRemote.txt
2025-08-02T18:00:31+0000 [HoneyPotSSHTransport,0,172.17.0.1] CMD: rm -f newfilefromRemote.txt
theia@thei
adocker-na
taschamart
:/home/pro
theia@theiadocker-nataschamart:/home/project$ cat honeypotLogs.txt | grep "Connection lost after"
2025-08-02T18:02:31+0000 [HoneyPotSSHTransport,0,172.17.0.1] Connection lost after 208 seconds
theia@theiadocker-nataschamart:/home/project$ | |
```

Terminate Cowrie



```
2025-08-02T18:02:31+0000 [HoneyPotSSHTransport,0,172.17.0.1] Connection lost after 208 seconds
theia@theiadocker-nataschamart:/home/project$ ^C
theia@theiadocker-nataschamart:/home/project$ docker run -p 2223:2222/tcp cowrie/cowrie >> honeypotLogs
.txt
/cowrie/cowrie-env/lib/python3.11/site-packages/twisted/conch/ssh/transport.py:106: CryptographyDepreca
tionWarning: Blowfish has been deprecated and will be removed in a future release
  b"blowfish-cbc": (algorithms.Blowfish, 16, modes.CBC),
/cowrie/cowrie-env/lib/python3.11/site-packages/twisted/conch/ssh/transport.py:110: CryptographyDepreca
tionWarning: CAST5 has been deprecated and will be removed in a future release
  b"cast128-cbc": (algorithms.CAST5, 16, modes.CBC),
/cowrie/cowrie-env/lib/python3.11/site-packages/twisted/conch/ssh/transport.py:115: CryptographyDepreca
tionWarning: Blowfish has been deprecated and will be removed in a future release
  b"blowfish-ctr": (algorithms.Blowfish, 16, modes.CTR),
/cowrie/cowrie-env/lib/python3.11/site-packages/twisted/conch/ssh/transport.py:116: CryptographyDepreca
tionWarning: CAST5 has been deprecated and will be removed in a future release
  b"cast128-ctr": (algorithms.CAST5, 16, modes.CTR),
```



```
theia@theiadocker-nataschamart:/home/project$ tail -f honeypotLogs.txt
2025-08-02T18:20:30+0000 [-] Cowrie Version 2.5.0
2025-08-02T18:20:30+0000 [-] Loaded output engine: jsonlog
2025-08-02T18:20:30+0000 [twisted.scripts._twistd_unix.UnixAppLogger#info] twistd 24.7.0 (/cowrie/cowri
e-env/bin/python3 3.11.2) starting up.
2025-08-02T18:20:30+0000 [twisted.scripts._twistd_unix.UnixAppLogger#info] reactor class: twisted.inter
net.epollreactor.EPollReactor.
2025-08-02T18:20:30+0000 [-] CowrieSSHFactory starting on 2222
2025-08-02T18:20:30+0000 [cowrie.ssh.factory.CowrieSSHFactory#info] Starting factory <cowrie.ssh.factor
y.CowrieSSHFactory object at 0x7682adc66e90>
2025-08-02T18:20:30+0000 [-] Generating new RSA keypair...
2025-08-02T18:20:30+0000 [-] Generating new ECDSA keypair...
2025-08-02T18:20:30+0000 [-] Generating new ed25519 keypair...
2025-08-02T18:20:30+0000 [-] Ready to accept SSH connections
 |
```