INSTITUTE FOR ADVANCED COMPUTING
AND
SOFTWARE DEVELOPMENT
AKURDI, PUNE

DOCUMENTATION ON

"HONEYPOT WITH SNORT"

PG-DITISS March-2023

SUBMITTED BY:

GROUP NO: 19

ROHAN KAMBLE (233438)

ANIKET THORAT (233447)

MR. KARTIK AWARI	**MR. ROHIT PURANIK**
PROJECT GUIDE	**CENTRE CO-ORDINATOR**

TABLE OF CONTENT

1. INTRODUCTION

The "Honeypot with Snort" project aims to create a controlled cybersecurity environment to attract and analyze potential attackers. By deploying a honeypot, which emulates vulnerable services, and integrating Snort as an intrusion detection system, we can observe attacker behaviors and enhance our understanding of their tactics. The project leverages tools like Cowrie (a honeypot) and Snort (an IDS) to simulate attacks and capture valuable data for analysis.

In the realm of cybersecurity, honeypots and intrusion detection systems play a crucial role in detecting and analyzing potential threats. This project aims to create a controlled environment by combining a honeypot, specifically Cowrie, with the intrusion detection system Snort, to monitor and collect data on attacker behavior. By simulating real-world attack scenarios, we seek to gain insights into attacker tactics, techniques, and procedures (TTPs), ultimately enhancing our understanding of cybersecurity threats.

2.EXISTING SYSTEM

In the absence of a dedicated honeypot environment and an integrated intrusion detection system, organizations often rely on reactive methods for identifying and mitigating cyber threats. Traditional cybersecurity measures typically involve signature-based systems that detect known attack patterns, leaving them susceptible to novel and evolving threats. The lack of a controlled testing environment hampers the ability to analyze and learn from attacker behavior effectively.

3. PROPOSED SYSTEM

The proposed system encompasses the integration of a honeypot environment, powered by Cowrie, and an intrusion detection system, Snort, to establish a proactive and controlled approach to threat detection and analysis. Unlike the existing reactive systems, the proposed environment enables the following enhancements:

1.Simulated Attack Scenarios:
The honeypot, emulating various services, attracts potential attackers.
The controlled environment allows ethical testing of different attack scenarios.

2.Real-time Alerting and Logging:
Snort's intrusion detection capabilities enable real-time alerting.
Detailed logs provide valuable insights into attacker tactics and techniques.

3.Threat Analysis and Mitigation:
By analyzing attack patterns and behaviors, organizations can identify vulnerabilities.
Improved threat intelligence helps in enhancing mitigation strategies.

4.Proactive Defense:
The proposed system shifts from reactive to proactive cybersecurity.
Organizations can anticipate and counter emerging threats more effectively.

4. TECHNOLOGY USED

a. **Hardware Requirement :**

- RAM: 16 GB
- HDD: 512GB
- Host Machine and Honeypot vm (AWS Ubuntu Instance):
  Instance Type: t2.micro or higher
  RAM: Minimum 1 GB
  Storage: At least 20 GB EBS volume

b. **Software Requirement :**

• .**Operating System:**
AWS Ubuntu 20.04 LTS instances for both the host machine and virtual environment.

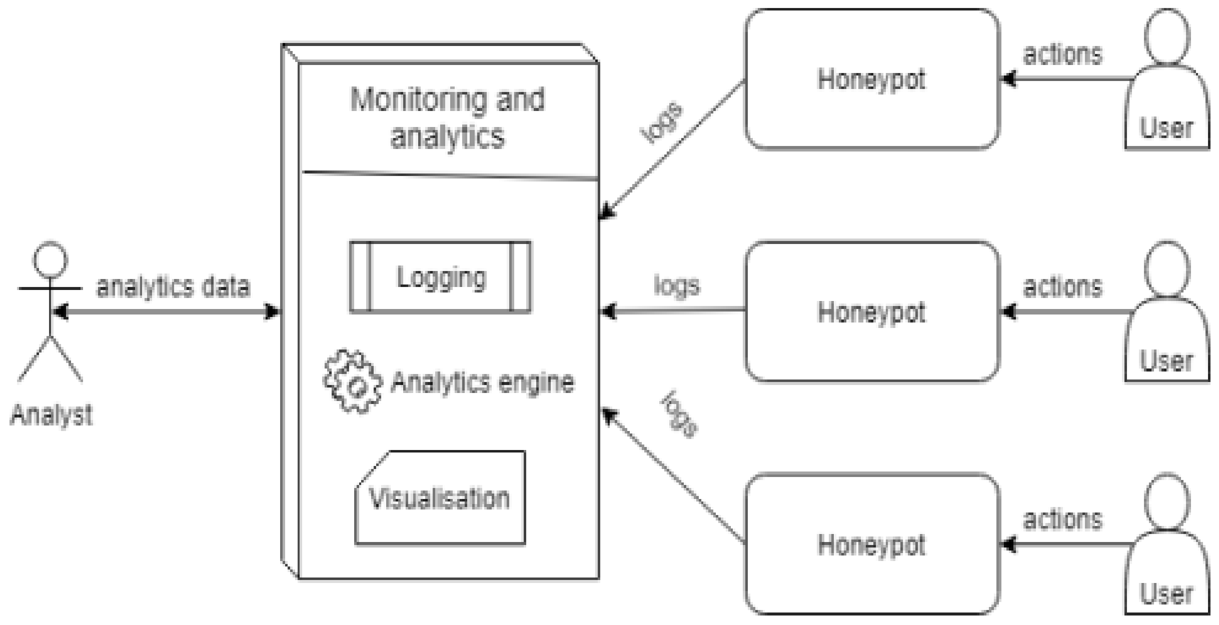
• **Honeypot Software**:
Cowrie: A low-interaction SSH and Telnet honeypot.
Snort: An open-source intrusion detection system for real-time alerting.

• **Web Service Emulation:**
Apache2: To emulate HTTP services and host the custom login page.
PHP: Required for server-side scripting on the login page.

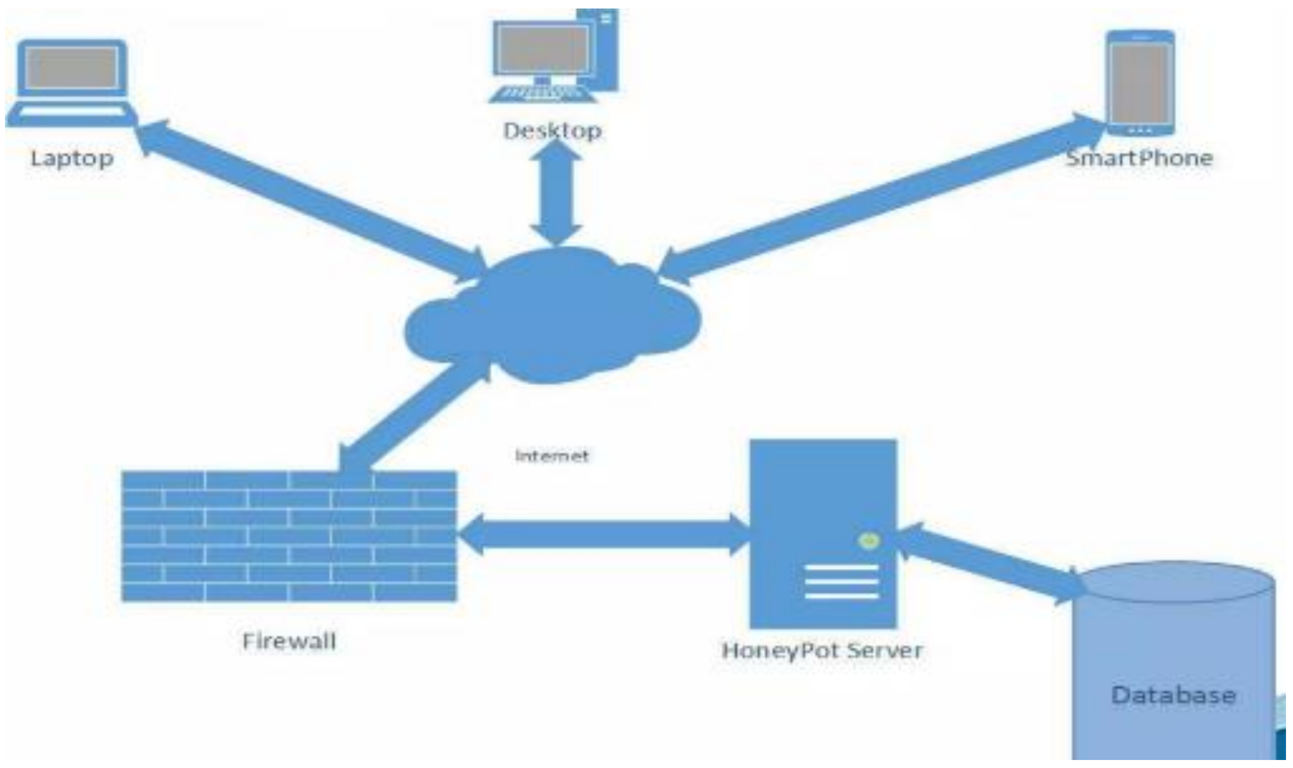
• **Database:**
MySQL: Used for Cowrie's data storage and Snort's alert logging.

5. UML DIAGRAM

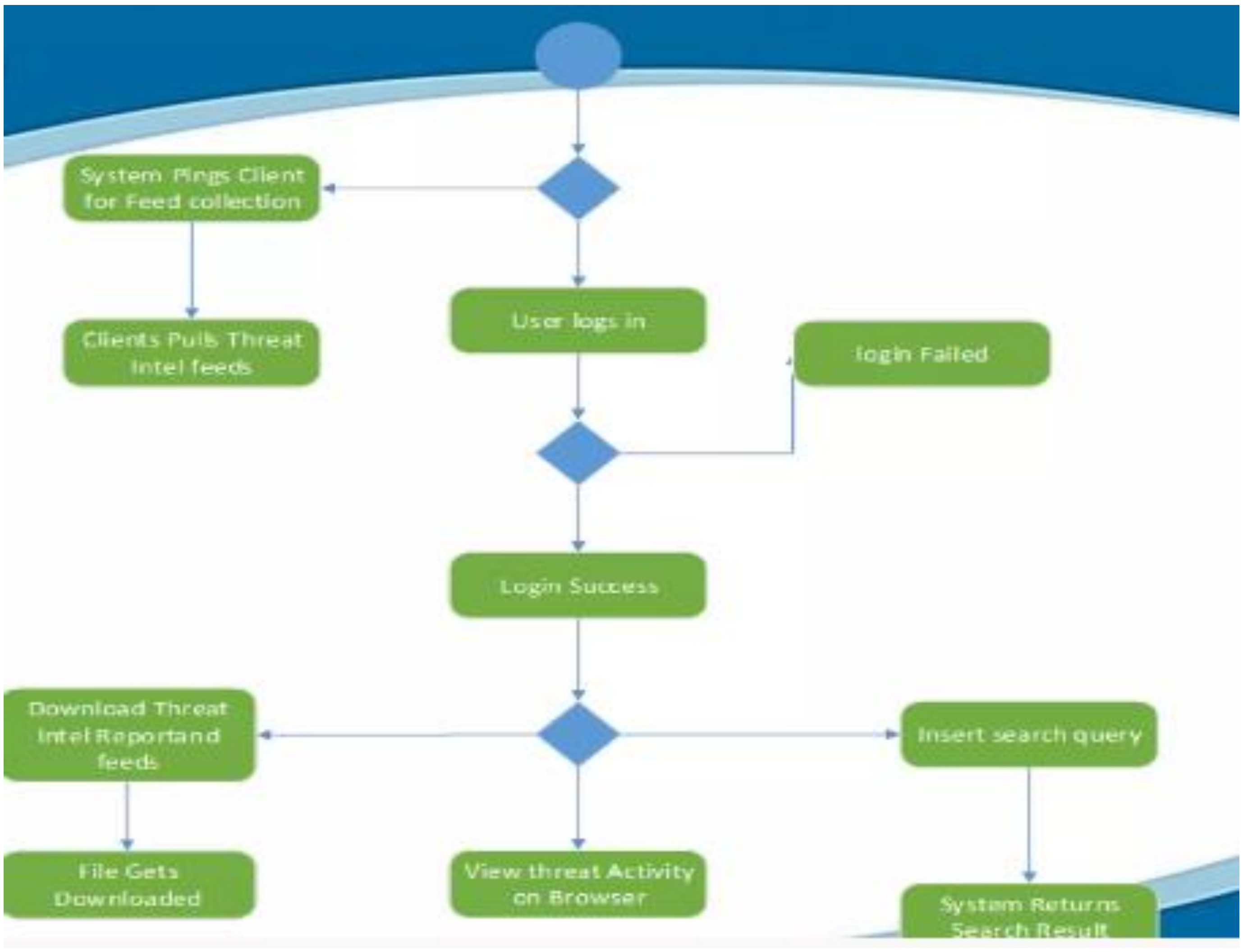
5.1 Use-case Diagram



5.2 Deployment Diagram



5.3 User Activity Diagram


System Pings Client for Feed collection
Clients Pulls Threat Intel feeds
User logs in
login Failed
Login Success
Download Threat Intel Report and feeds
Insert search query
File Gets Downloaded
View threat Activity on Browser
System Returns Search Result

6. SYSTEM IMPLEMENTATION

6.1 . Setting up AWS Environment:

In this initial phase, the AWS cloud environment is established. Two Ubuntu instances are launched – one for the host machine and the other for the honeypot virtual machine. Security groups are configured to control incoming and outgoing traffic, forming the foundational infrastructure for the project. These instances will serve as the infrastructure on which the honeypot and Snort will be deployed.

6.2 . Installing Required Softwares:

After setting up the AWS environment, the focus shifts to software installation. The host machine is accessed using SSH with a private key. This allows secure remote access to the host machine. Then, Cowrie and Snort are installed on the host machine. Cowrie serves as the honeypot, while Snort acts as the intrusion detection system. This step forms the core of the cybersecurity experimentation environment.

6.3 . Configuring Cowrie :

Configuration is vital to tailor Cowrie's behavior and interactions with potential attackers. By navigating to the Cowrie directory and editing the configuration file, specific settings related to SSH and Telnet emulation, banners, and log paths are customized. This configuration ensures that the honeypot accurately mimics real services and attracts potential attackers.

Cowrie's configuration is pivotal in simulating real services and attracting potential attackers. The configuration file is edited to customize settings such as SSH and Telnet emulation, banners, and log paths. These configurations ensure the honeypot's accurate representation of real services and its ability to capture attacker activities.

6.4. Emulating Web Services:

To diversify the honeypot's attractiveness, web services are emulated. Apache2 and PHP are installed on the host machine to create a simulated web application. A custom login page, designed with HTML and PHP, provides a vulnerable entry point for attackers. This step extends the honeypot's scope to web-based attacks, enhancing the detection capabilities of Snort. Web services are emulated to expand the honeypot's attractiveness. Apache2 and PHP are installed on the host machine to create a simulated web application. A customized login page is designed with HTML and PHP, acting as a vulnerable entry point for attackers interested in web-based attacks.

6.5. Configuring Snort:

The configuration of Snort is crucial for efficient threat detection. Editing the Snort configuration file allows customization of rules and settings that dictate how Snort analyzes network traffic. By configuring Snort to log alerts to specific directories, the system is set up to detect and report potentially malicious activities, serving as a crucial component of the intrusion detection system. Snort's configuration is vital to efficient threat detection. By editing its configuration file, rules and settings are customized to dictate how network traffic is analyzed. These configurations enable Snort to log alerts to specific directories, allowing it to detect and report potentially malicious activities.

6.6. Testing The Environment:

This step is focused on practical testing and validation. The Cowrie honeypot is accessed via SSH or Telnet from external systems, simulating real-world scenarios where attackers might attempt unauthorized access. By triggering attacks and analyzing alerts and logs, the effectiveness of the project's cybersecurity mechanisms can be evaluated. Practical testing and validation are essential. By triggering attacks and analyzing alerts and logs, the effectiveness of the project's cybersecurity mechanisms is evaluated.

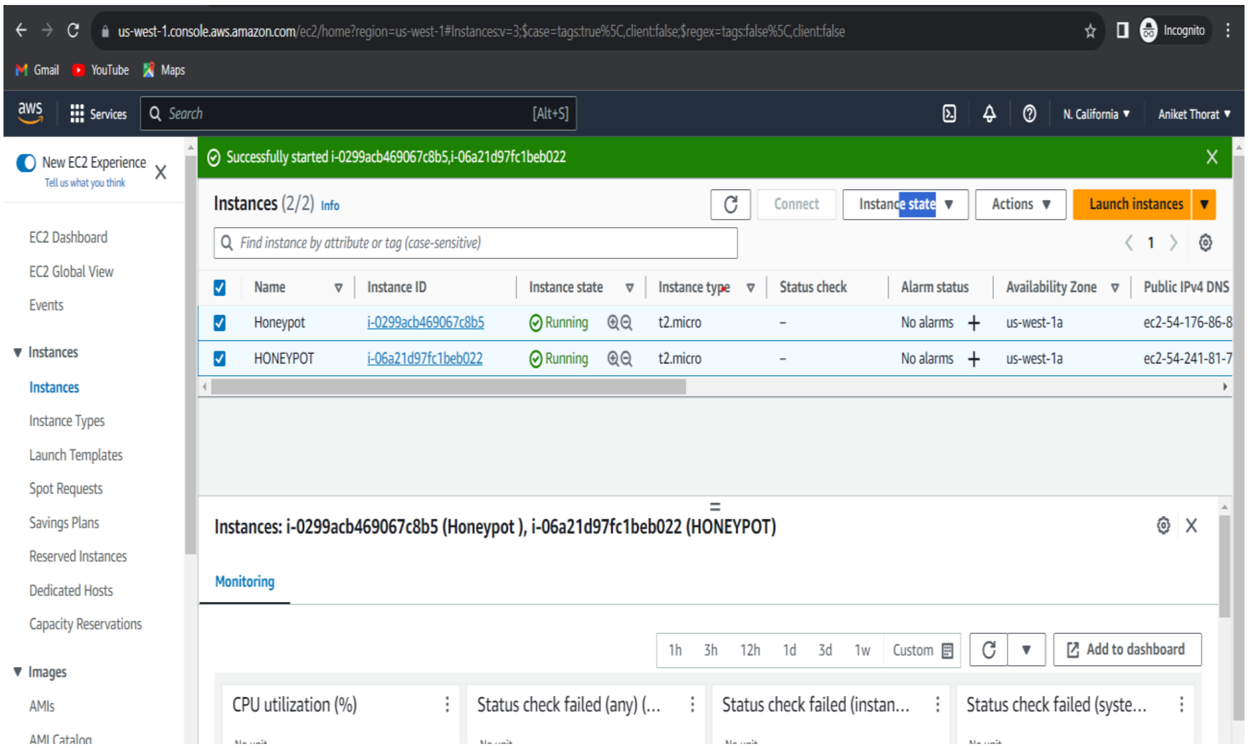
6.7. Database Configuration:

In this step, we will configure a MySQL database to store and manage the data generated by the Cowrie honeypot. This database will play a crucial role in collecting and organizing the logs and information about potential attacker behavior, enhancing our ability to analyze and understand the threat landscape. A MySQL database is configured to store and manage Cowrie's log data. This step enhances the project's data collection and analysis capabilities, enabling more efficient querying and correlation of different logs. The database empowers the project to gain deeper insights into attacker techniques and patterns.

.

7.OUTPUT

7.1. Setting up AWS Environment:


us-west-1.console.aws.amazon.com/ec2/home?region=us-west-1#Instances:v=3;$case=tags:true%5C,client:false;$regex=tags:false%5C,client:false

Successfully started i-0299acb469067c8b5,i-06a21d97fc1beb022

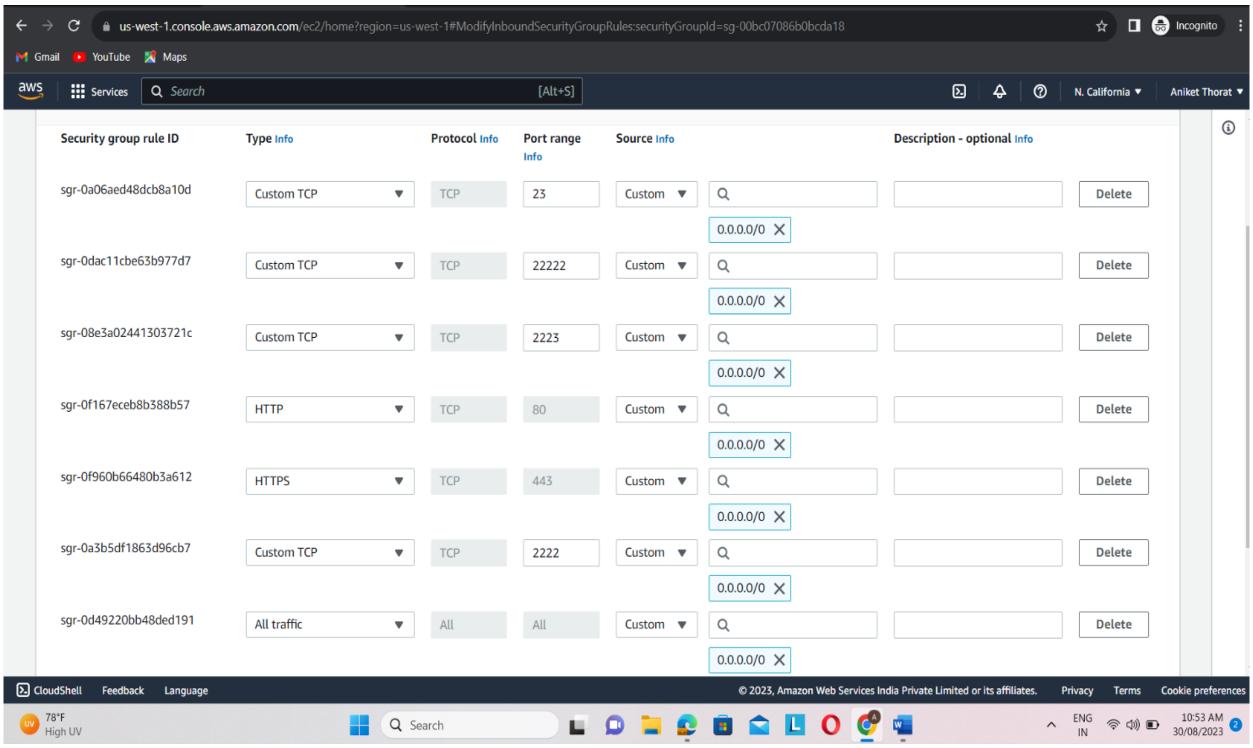
Instances (2/2) Info

	Name	Instance ID	Instance state	Instance type	Status check	Alarm status	Availability Zone	Public IPv4 DNS
✓	Honeypot	i-0299acb469067c8b5	Running	t2.micro	–	No alarms	us-west-1a	ec2-54-176-86-8
✓	HONEYPOT	i-06a21d97fc1beb022	Running	t2.micro	–	No alarms	us-west-1a	ec2-54-241-81-7

Instances: i-0299acb469067c8b5 (Honeypot), i-06a21d97fc1beb022 (HONEYPOT)

Monitoring

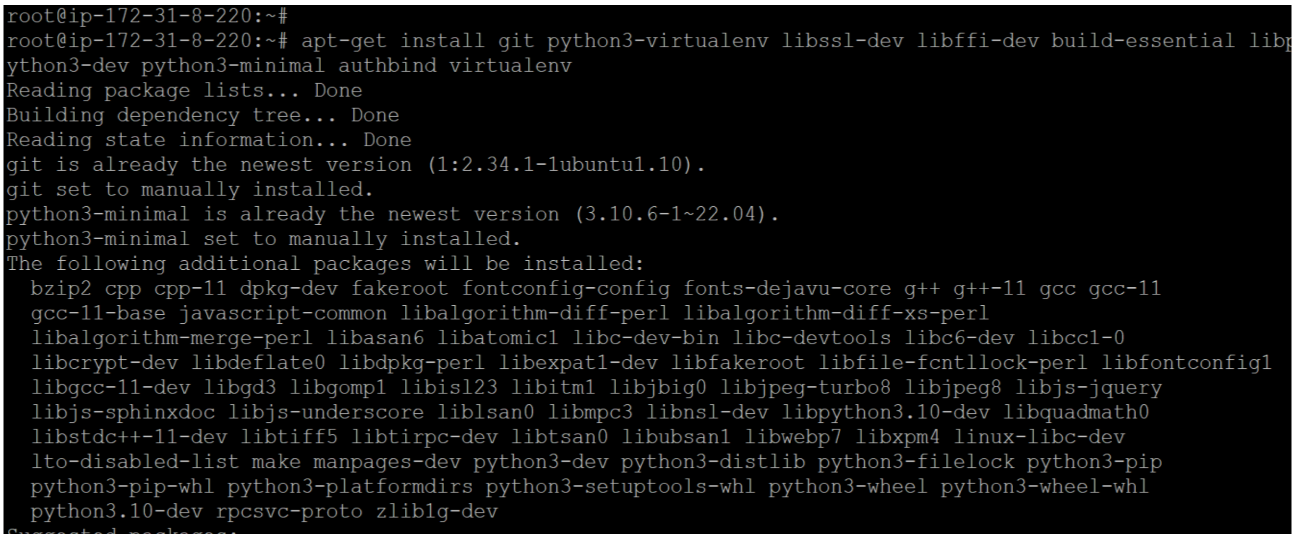
CPU utilization (%) · Status check failed (any) (... · Status check failed (instan... · Status check failed (syste...


us-west-1.console.aws.amazon.com/ec2/home?region=us-west-1#ModifyInboundSecurityGroupRules:securityGroupId=sg-00bc07086b0bcda18

Security group rule ID	Type	Protocol	Port range	Source		Description - optional
sgr-0a06aed48dcb8a10d	Custom TCP	TCP	23	Custom	0.0.0.0/0	
sgr-0dac11cbe63b977d7	Custom TCP	TCP	22222	Custom	0.0.0.0/0	
sgr-08e3a02441303721c	Custom TCP	TCP	2223	Custom	0.0.0.0/0	
sgr-0f167eceb8b388b57	HTTP	TCP	80	Custom	0.0.0.0/0	
sgr-0f960b66480b3a612	HTTPS	TCP	443	Custom	0.0.0.0/0	
sgr-0a3b5df1863d96cb7	Custom TCP	TCP	2222	Custom	0.0.0.0/0	
sgr-0d49220bb48ded191	All traffic	All	All	Custom	0.0.0.0/0	

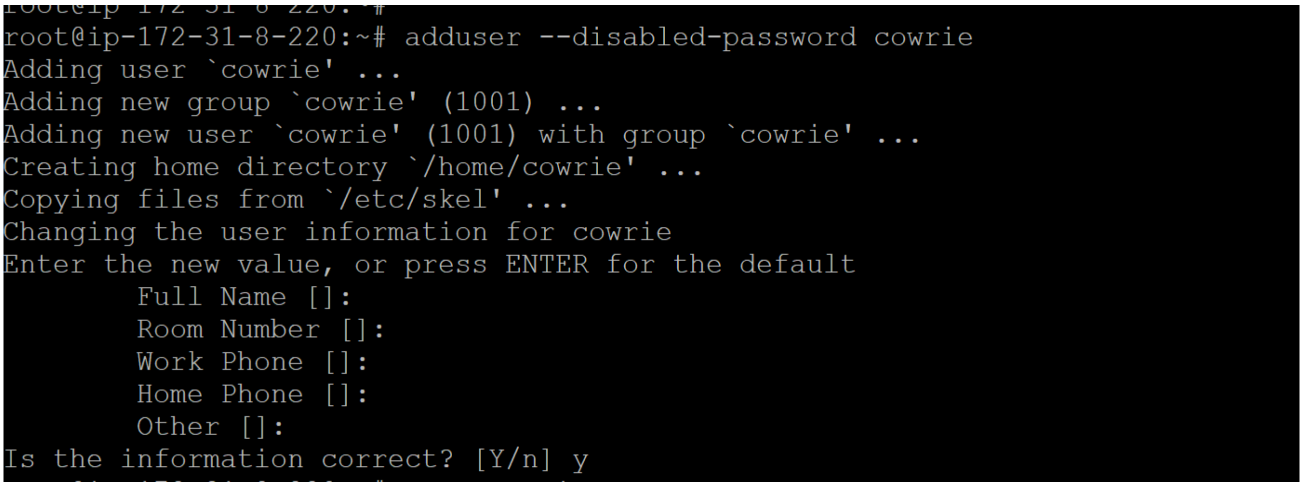
© 2023, Amazon Web Services India Private Limited or its affiliates.

7.2. Installing Required Softwares:

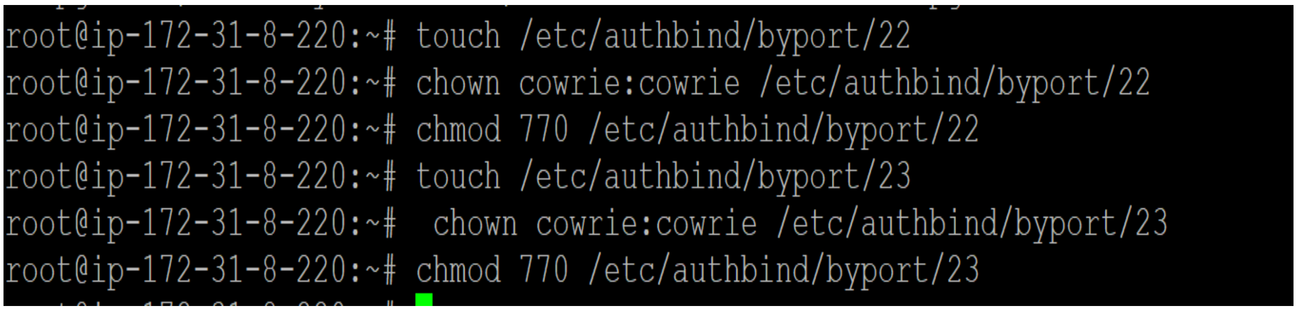


```
root@ip-172-31-8-220:~#
root@ip-172-31-8-220:~# apt-get install git python3-virtualenv libssl-dev libffi-dev build-essential libp
ython3-dev python3-minimal authbind virtualenv
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
git is already the newest version (1:2.34.1-1ubuntu1.10).
git set to manually installed.
python3-minimal is already the newest version (3.10.6-1~22.04).
python3-minimal set to manually installed.
The following additional packages will be installed:
  bzip2 cpp cpp-11 dpkg-dev fakeroot fontconfig-config fonts-dejavu-core g++ g++-11 gcc gcc-11
  gcc-11-base javascript-common libalgorithm-diff-perl libalgorithm-diff-xs-perl
  libalgorithm-merge-perl libasan6 libatomic1 libc-dev-bin libc-devtools libc6-dev libcc1-0
  libcrypt-dev libdeflate0 libdpkg-perl libexpat1-dev libfakeroot libfile-fcntllock-perl libfontconfig1
  libgcc-11-dev libgd3 libgomp1 libisl23 libitm1 libjbig0 libjpeg-turbo8 libjpeg8 libjs-jquery
  libjs-sphinxdoc libjs-underscore liblsan0 libmpc3 libnsl-dev libpython3.10-dev libquadmath0
  libstdc++-11-dev libtiff5 libtirpc-dev libtsan0 libubsan1 libwebp7 libxpm4 linux-libc-dev
  lto-disabled-list make manpages-dev python3-dev python3-distlib python3-filelock python3-pip
  python3-pip-whl python3-platformdirs python3-setuptools-whl python3-wheel python3-wheel-whl
  python3.10-dev rpcsvc-proto zlib1g-dev
```

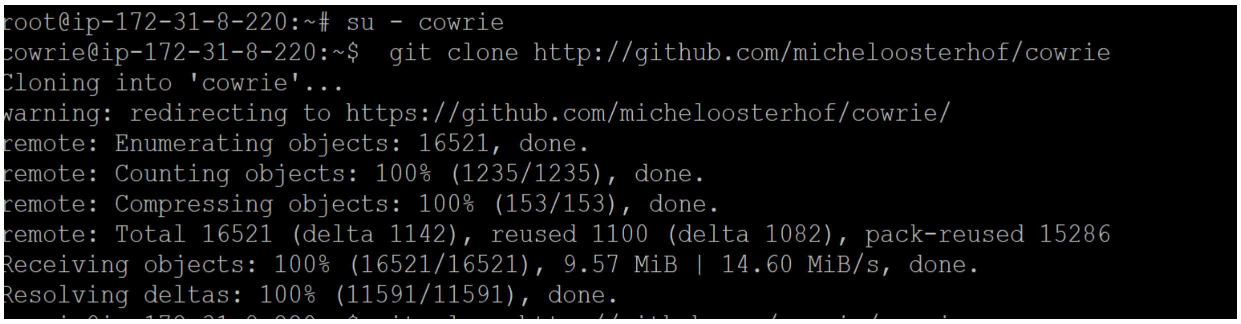
7.3. Configuring Cowrie :



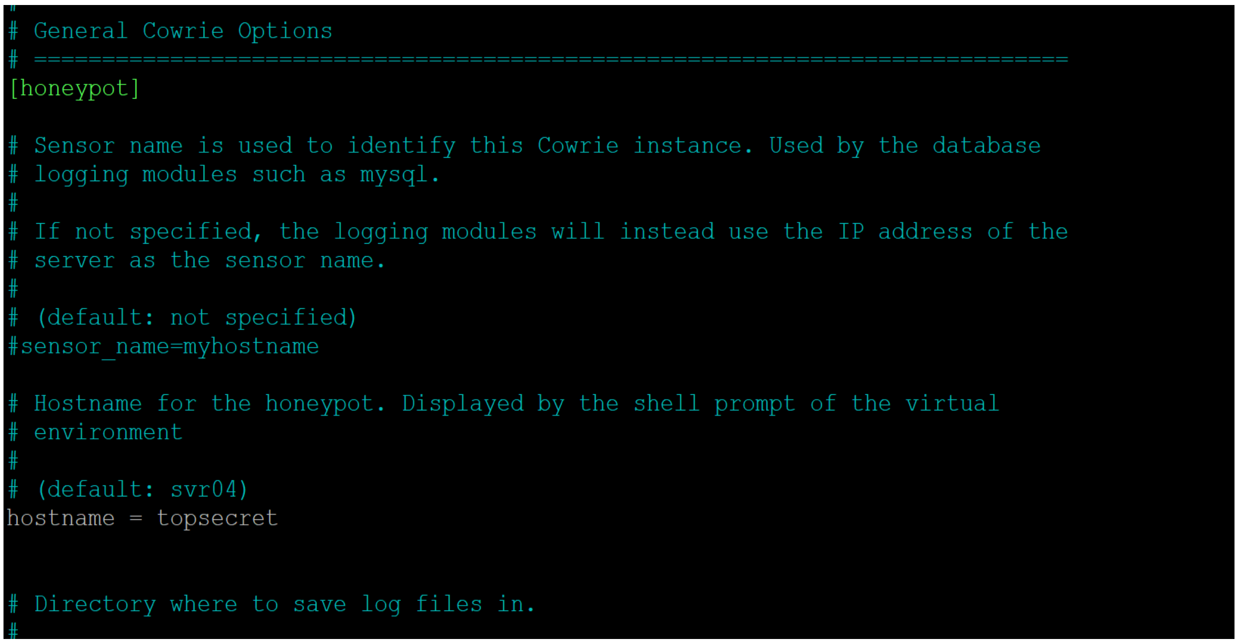
```
root@ip-172-31-8-220:~# adduser --disabled-password cowrie
Adding user `cowrie' ...
Adding new group `cowrie' (1001) ...
Adding new user `cowrie' (1001) with group `cowrie' ...
Creating home directory `/home/cowrie' ...
Copying files from `/etc/skel' ...
Changing the user information for cowrie
Enter the new value, or press ENTER for the default
        Full Name []:
        Room Number []:
        Work Phone []:
        Home Phone []:
        Other []:
Is the information correct? [Y/n] y
```



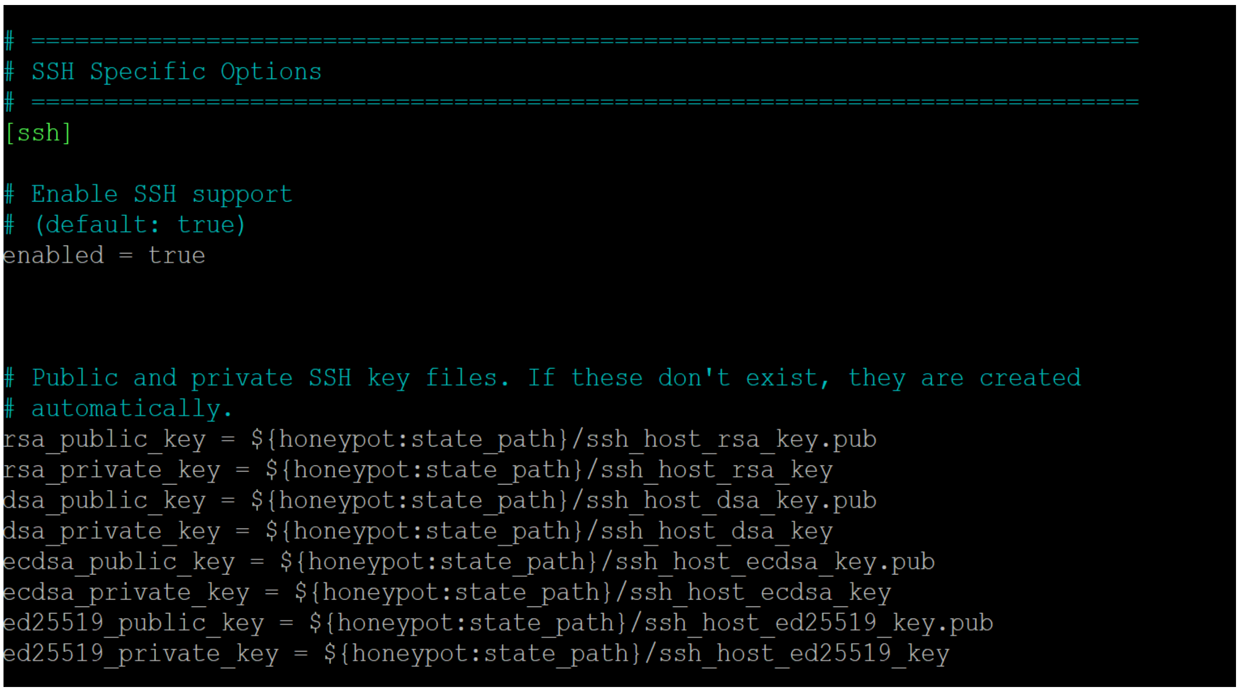
```
root@ip-172-31-8-220:~# touch /etc/authbind/byport/22
root@ip-172-31-8-220:~# chown cowrie:cowrie /etc/authbind/byport/22
root@ip-172-31-8-220:~# chmod 770 /etc/authbind/byport/22
root@ip-172-31-8-220:~# touch /etc/authbind/byport/23
root@ip-172-31-8-220:~#  chown cowrie:cowrie /etc/authbind/byport/23
root@ip-172-31-8-220:~# chmod 770 /etc/authbind/byport/23
```



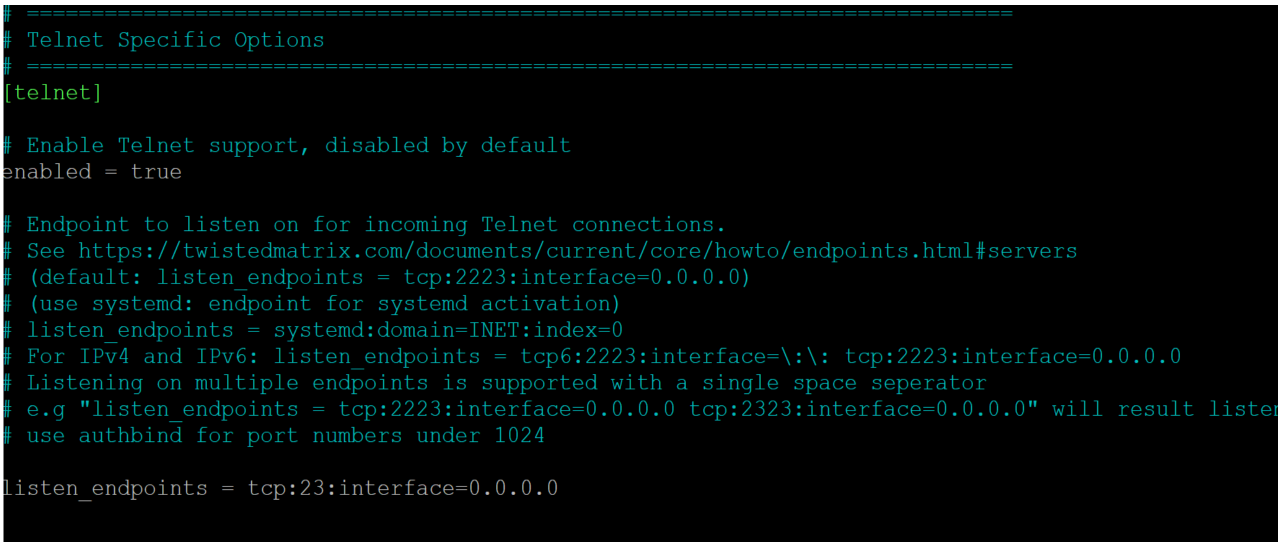
```
root@ip-172-31-8-220:~# su - cowrie
cowrie@ip-172-31-8-220:~$  git clone http://github.com/micheloosterhof/cowrie
Cloning into 'cowrie'...
warning: redirecting to https://github.com/micheloosterhof/cowrie/
remote: Enumerating objects: 16521, done.
remote: Counting objects: 100% (1235/1235), done.
remote: Compressing objects: 100% (153/153), done.
remote: Total 16521 (delta 1142), reused 1100 (delta 1082), pack-reused 15286
Receiving objects: 100% (16521/16521), 9.57 MiB | 14.60 MiB/s, done.
Resolving deltas: 100% (11591/11591), done.
```



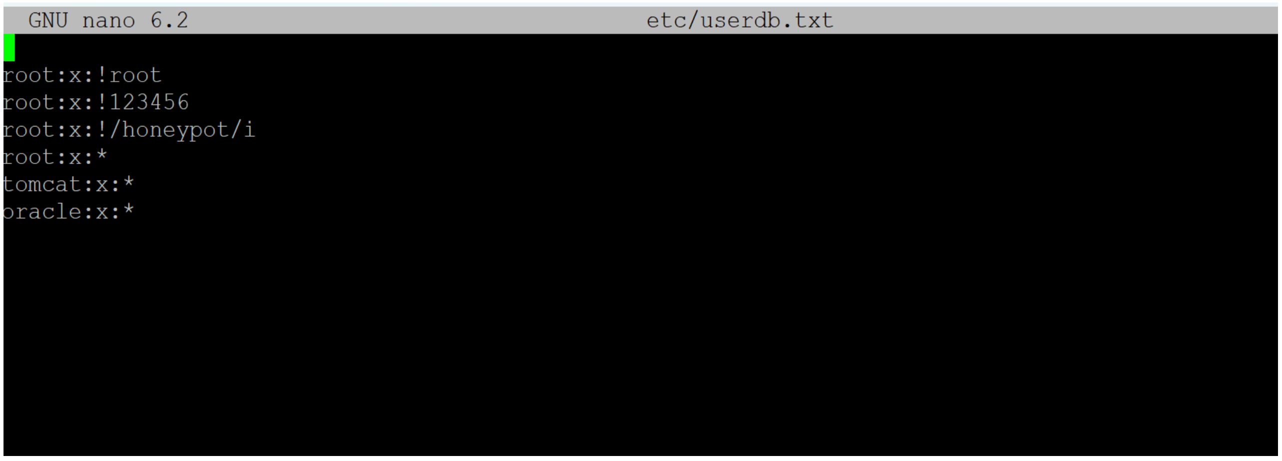
```
# General Cowrie Options
# ============================================================================
[honeypot]

# Sensor name is used to identify this Cowrie instance. Used by the database
# logging modules such as mysql.
#
# If not specified, the logging modules will instead use the IP address of the
# server as the sensor name.
#
# (default: not specified)
#sensor_name=myhostname

# Hostname for the honeypot. Displayed by the shell prompt of the virtual
# environment
#
# (default: svr04)
hostname = topsecret


# Directory where to save log files in.
#
```



```
# ============================================================================
# SSH Specific Options
# ============================================================================
[ssh]

# Enable SSH support
# (default: true)
enabled = true



# Public and private SSH key files. If these don't exist, they are created
# automatically.
rsa_public_key = ${honeypot:state_path}/ssh_host_rsa_key.pub
rsa_private_key = ${honeypot:state_path}/ssh_host_rsa_key
dsa_public_key = ${honeypot:state_path}/ssh_host_dsa_key.pub
dsa_private_key = ${honeypot:state_path}/ssh_host_dsa_key
ecdsa_public_key = ${honeypot:state_path}/ssh_host_ecdsa_key.pub
ecdsa_private_key = ${honeypot:state_path}/ssh_host_ecdsa_key
ed25519_public_key = ${honeypot:state_path}/ssh_host_ed25519_key.pub
ed25519_private_key = ${honeypot:state_path}/ssh_host_ed25519_key
```

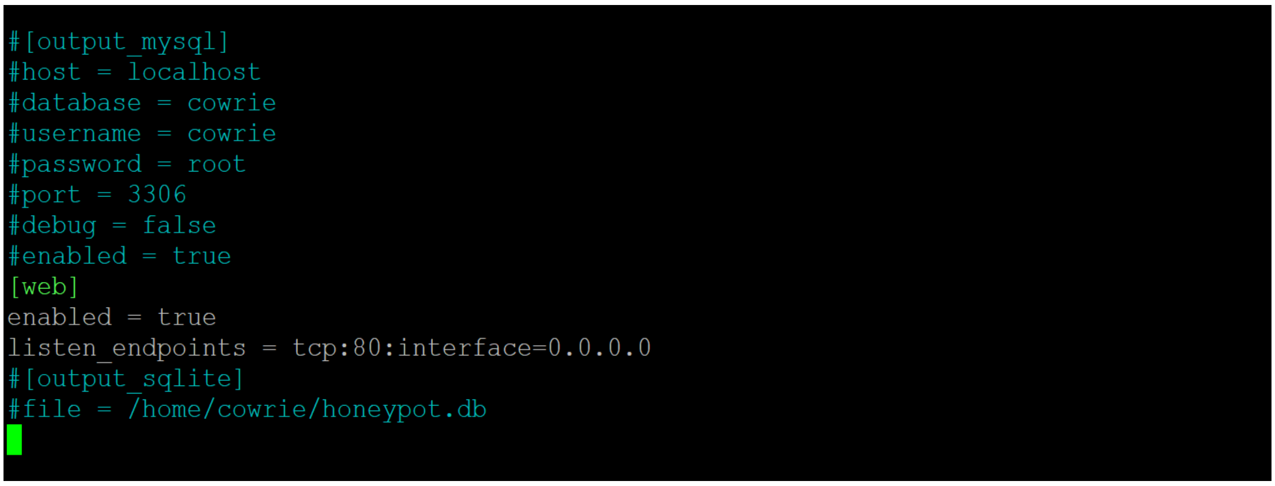


```
# ============================================================================
# Telnet Specific Options
# ============================================================================
[telnet]

# Enable Telnet support, disabled by default
enabled = true

# Endpoint to listen on for incoming Telnet connections.
# See https://twistedmatrix.com/documents/current/core/howto/endpoints.html#servers
# (default: listen_endpoints = tcp:2223:interface=0.0.0.0)
# (use systemd: endpoint for systemd activation)
# listen_endpoints = systemd:domain=INET:index=0
# For IPv4 and IPv6: listen_endpoints = tcp6:2223:interface=\:\: tcp:2223:interface=0.0.0.0
# Listening on multiple endpoints is supported with a single space seperator
# e.g "listen_endpoints = tcp:2223:interface=0.0.0.0 tcp:2323:interface=0.0.0.0" will result liste
# use authbind for port numbers under 1024

listen_endpoints = tcp:23:interface=0.0.0.0
```

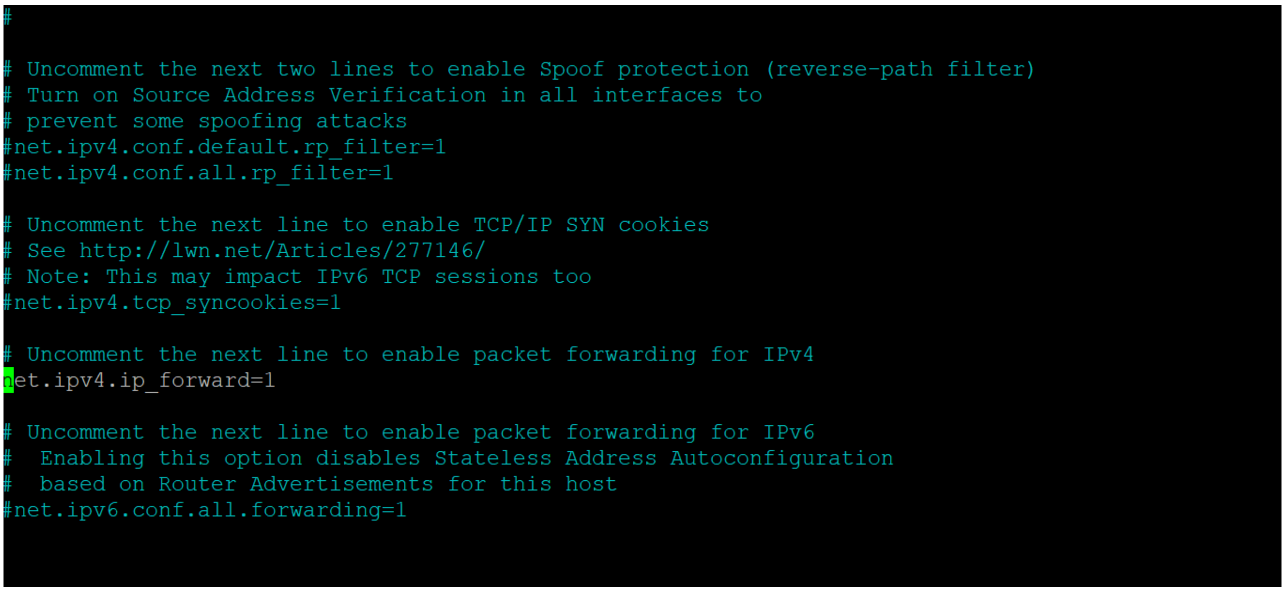


```
  GNU nano 6.2                              etc/userdb.txt

root:x:!root
root:x:!123456
root:x:!/honeypot/i
root:x:*
tomcat:x:*
oracle:x:*
```

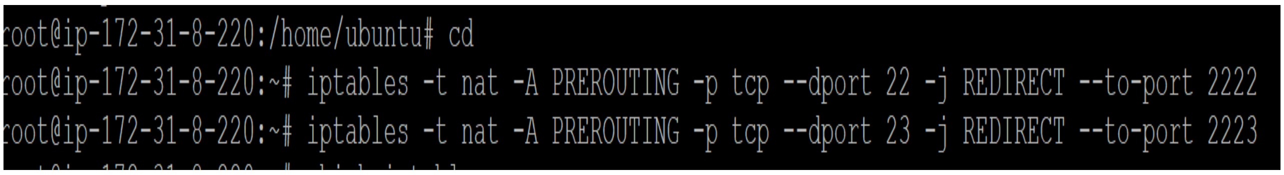
7.4 Emulating Web Services:



```
#[output_mysql]
#host = localhost
#database = cowrie
#username = cowrie
#password = root
#port = 3306
#debug = false
#enabled = true
[web]
enabled = true
listen_endpoints = tcp:80:interface=0.0.0.0
#[output_sqlite]
#file = /home/cowrie/honeypot.db
```

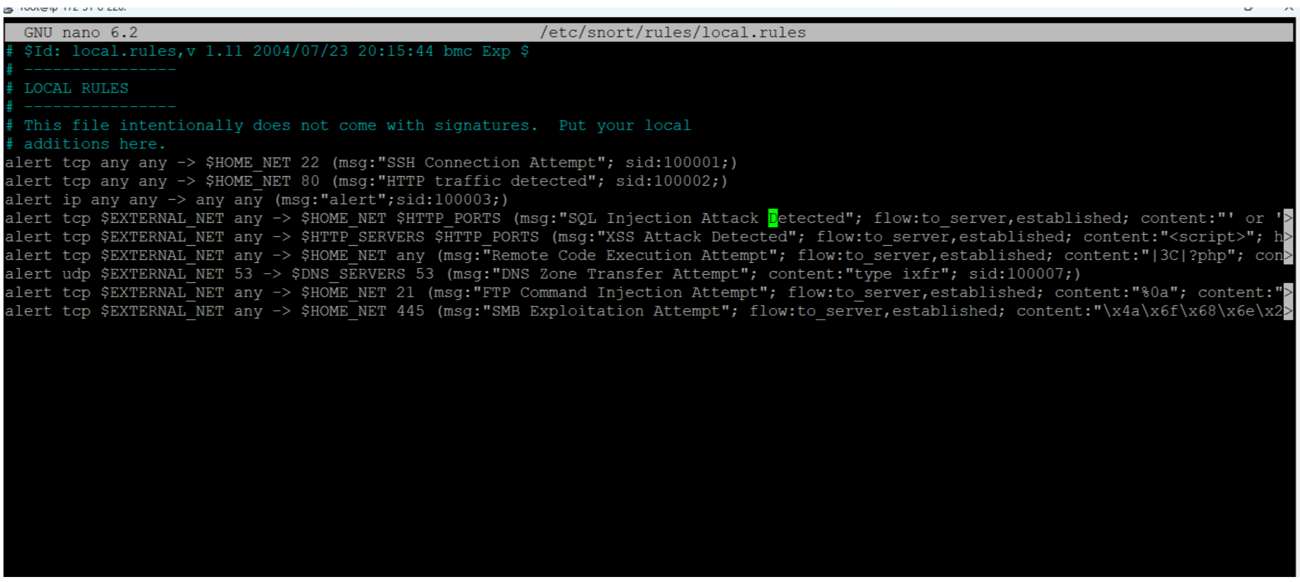


```
#
# Uncomment the next two lines to enable Spoof protection (reverse-path filter)
# Turn on Source Address Verification in all interfaces to
# prevent some spoofing attacks
#net.ipv4.conf.default.rp_filter=1
#net.ipv4.conf.all.rp_filter=1

# Uncomment the next line to enable TCP/IP SYN cookies
# See http://lwn.net/Articles/277146/
# Note: This may impact IPv6 TCP sessions too
#net.ipv4.tcp_syncookies=1

# Uncomment the next line to enable packet forwarding for IPv4
net.ipv4.ip_forward=1

# Uncomment the next line to enable packet forwarding for IPv6
#  Enabling this option disables Stateless Address Autoconfiguration
#  based on Router Advertisements for this host
#net.ipv6.conf.all.forwarding=1
```



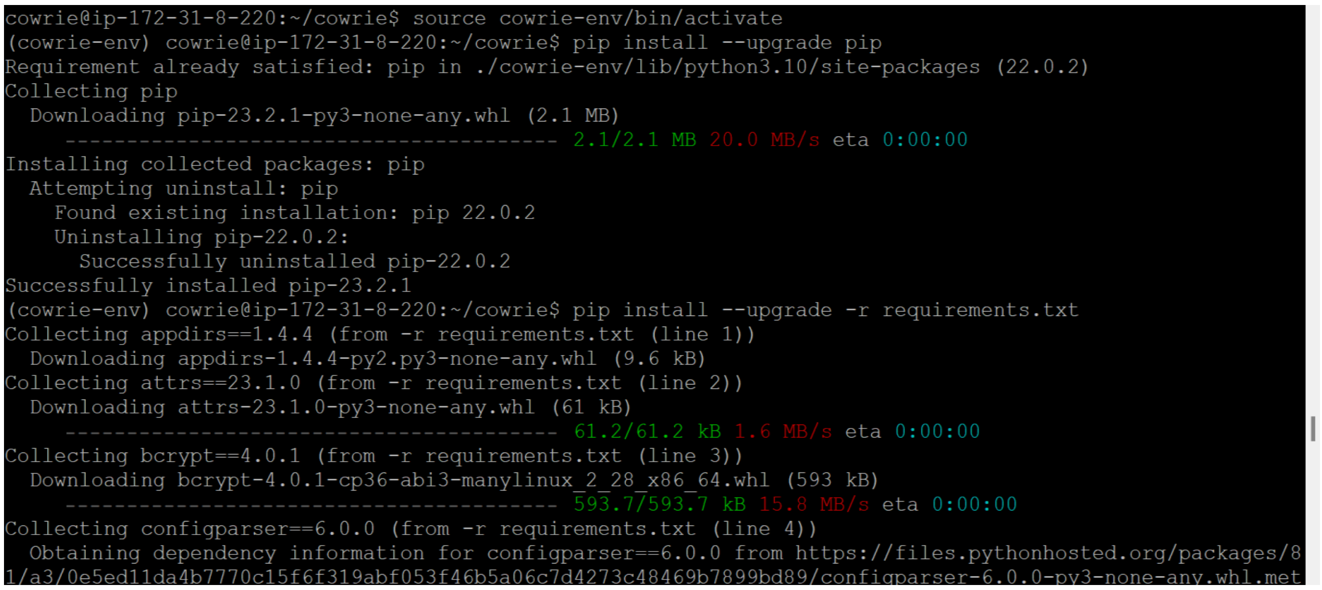
```
root@ip-172-31-8-220:/home/ubuntu# cd
root@ip-172-31-8-220:~# iptables -t nat -A PREROUTING -p tcp --dport 22 -j REDIRECT --to-port 2222
root@ip-172-31-8-220:~# iptables -t nat -A PREROUTING -p tcp --dport 23 -j REDIRECT --to-port 2223
```

7.5. Configuring Snort:

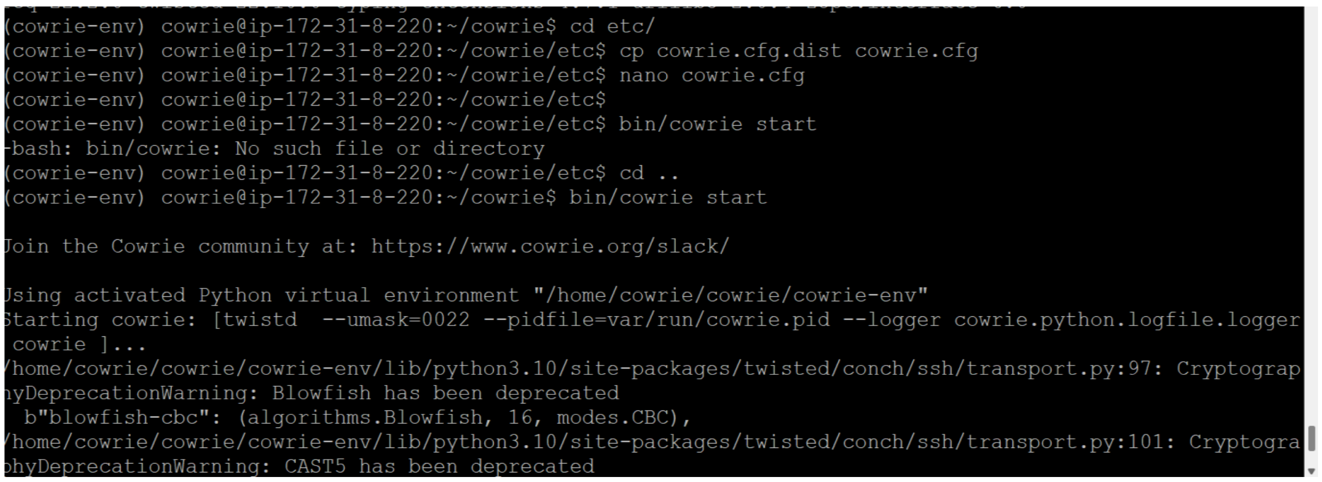


```
  GNU nano 6.2                                  /etc/snort/rules/local.rules
# $Id: local.rules,v 1.11 2004/07/23 20:15:44 bmc Exp $
# ----------------
# LOCAL RULES
# ----------------
# This file intentionally does not come with signatures.  Put your local
# additions here.
alert tcp any any -> $HOME_NET 22 (msg:"SSH Connection Attempt"; sid:100001;)
alert tcp any any -> $HOME_NET 80 (msg:"HTTP traffic detected"; sid:100002;)
alert ip any any -> any any (msg:"alert";sid:100003;)
alert tcp $EXTERNAL_NET any -> $HOME_NET $HTTP_PORTS (msg:"SQL Injection Attack Detected"; flow:to_server,established; content:"' or '>
alert tcp $EXTERNAL_NET any -> $HTTP_SERVERS $HTTP_PORTS (msg:"XSS Attack Detected"; flow:to_server,established; content:"<script>"; h>
alert tcp $EXTERNAL_NET any -> $HOME_NET any (msg:"Remote Code Execution Attempt"; flow:to_server,established; content:"|3C|?php"; con>
alert udp $EXTERNAL_NET 53 -> $DNS_SERVERS 53 (msg:"DNS Zone Transfer Attempt"; content:"type ixfr"; sid:100007;)
alert tcp $EXTERNAL_NET any -> $HOME_NET 21 (msg:"FTP Command Injection Attempt"; flow:to_server,established; content:"%0a"; content:">
alert tcp $EXTERNAL_NET any -> $HOME_NET 445 (msg:"SMB Exploitation Attempt"; flow:to_server,established; content:"\x4a\x6f\x68\x6e\x2>
```

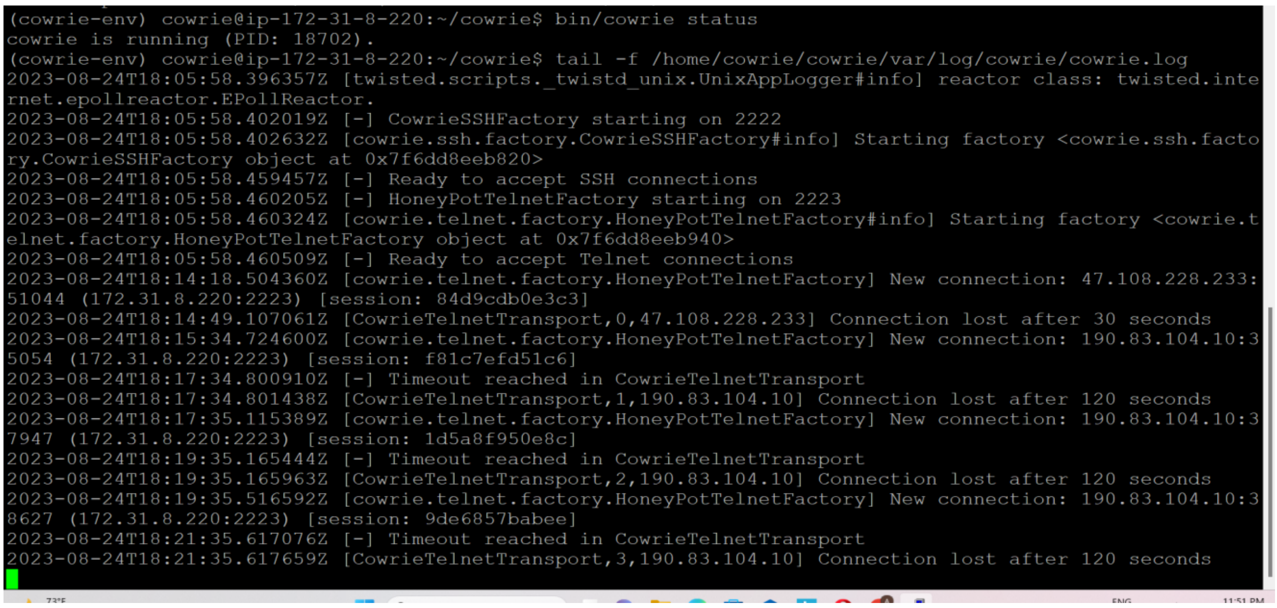
7.6. Testing The Environment:



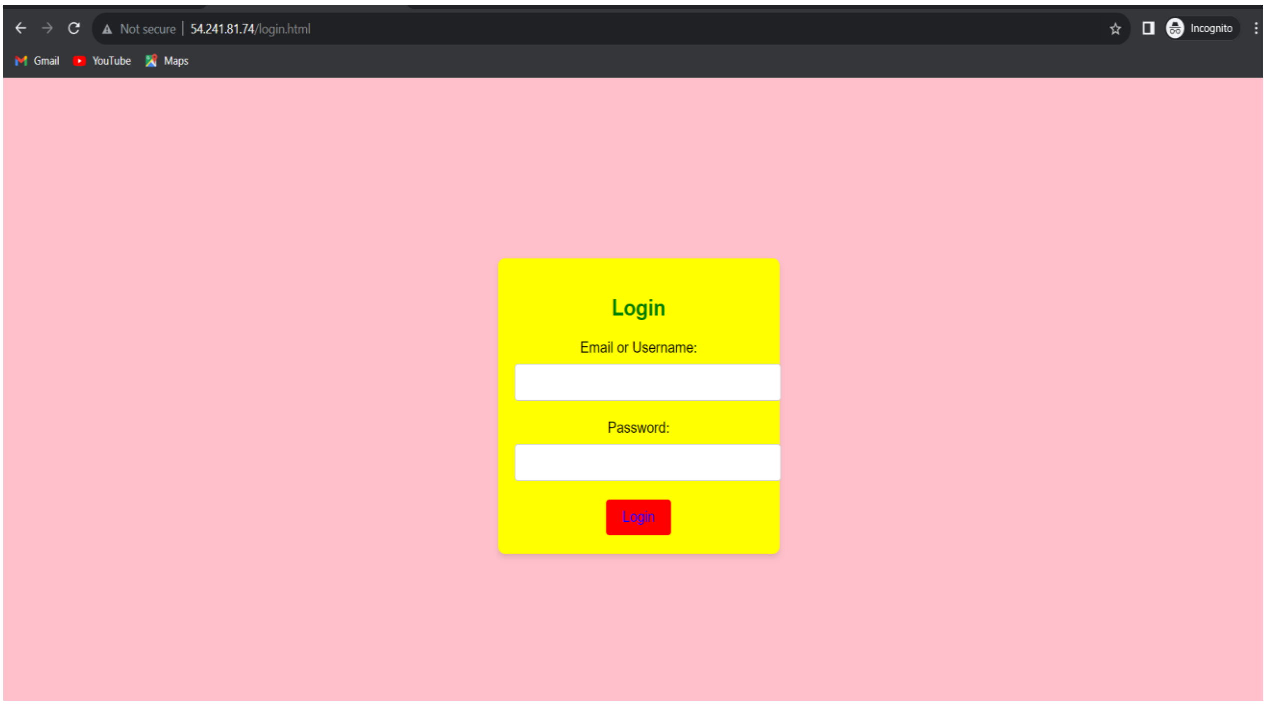
```
cowrie@ip-172-31-8-220:~/cowrie$ source cowrie-env/bin/activate
(cowrie-env) cowrie@ip-172-31-8-220:~/cowrie$ pip install --upgrade pip
Requirement already satisfied: pip in ./cowrie-env/lib/python3.10/site-packages (22.0.2)
Collecting pip
  Downloading pip-23.2.1-py3-none-any.whl (2.1 MB)
     ---------------------------------------- 2.1/2.1 MB 20.0 MB/s eta 0:00:00
Installing collected packages: pip
  Attempting uninstall: pip
    Found existing installation: pip 22.0.2
    Uninstalling pip-22.0.2:
      Successfully uninstalled pip-22.0.2
Successfully installed pip-23.2.1
(cowrie-env) cowrie@ip-172-31-8-220:~/cowrie$ pip install --upgrade -r requirements.txt
Collecting appdirs==1.4.4 (from -r requirements.txt (line 1))
  Downloading appdirs-1.4.4-py2.py3-none-any.whl (9.6 kB)
Collecting attrs==23.1.0 (from -r requirements.txt (line 2))
  Downloading attrs-23.1.0-py3-none-any.whl (61 kB)
     ---------------------------------------- 61.2/61.2 kB 1.6 MB/s eta 0:00:00
Collecting bcrypt==4.0.1 (from -r requirements.txt (line 3))
  Downloading bcrypt-4.0.1-cp36-abi3-manylinux_2_28_x86_64.whl (593 kB)
     ---------------------------------------- 593.7/593.7 kB 15.8 MB/s eta 0:00:00
Collecting configparser==6.0.0 (from -r requirements.txt (line 4))
  Obtaining dependency information for configparser==6.0.0 from https://files.pythonhosted.org/packages/8
1/a3/0e5ed11da4b7770c15f6f319abf053f46b5a06c7d4273c48469b7899bd89/configparser-6.0.0-py3-none-any.whl.met
```

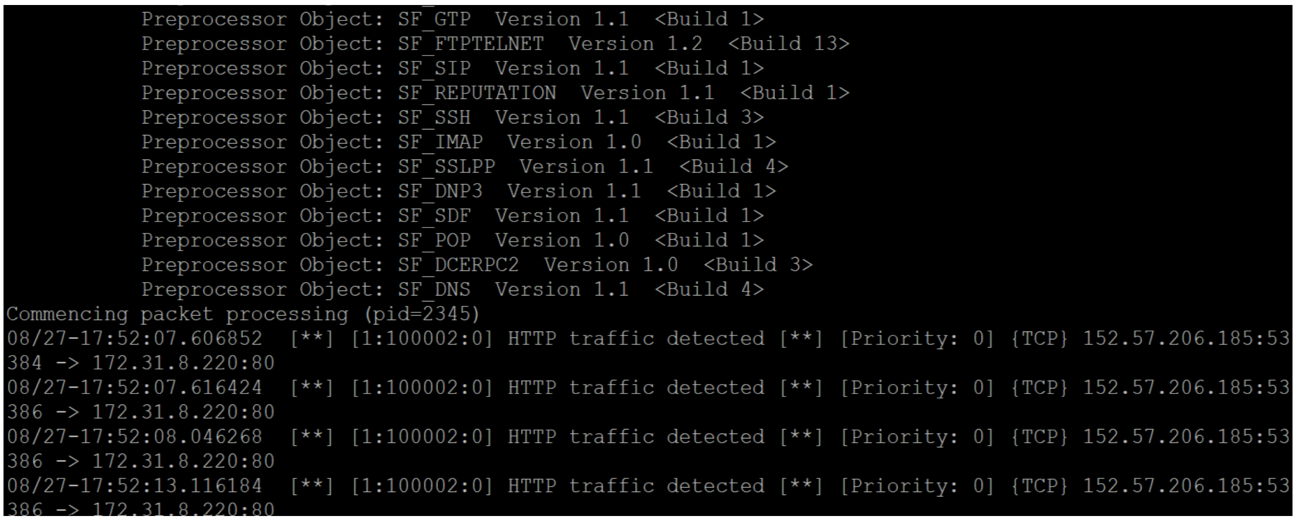


```
(cowrie-env) cowrie@ip-172-31-8-220:~/cowrie$ cd etc/
(cowrie-env) cowrie@ip-172-31-8-220:~/cowrie/etc$ cp cowrie.cfg.dist cowrie.cfg
(cowrie-env) cowrie@ip-172-31-8-220:~/cowrie/etc$ nano cowrie.cfg
(cowrie-env) cowrie@ip-172-31-8-220:~/cowrie/etc$
(cowrie-env) cowrie@ip-172-31-8-220:~/cowrie/etc$ bin/cowrie start
-bash: bin/cowrie: No such file or directory
(cowrie-env) cowrie@ip-172-31-8-220:~/cowrie/etc$ cd ..
(cowrie-env) cowrie@ip-172-31-8-220:~/cowrie$ bin/cowrie start

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

Using activated Python virtual environment "/home/cowrie/cowrie/cowrie-env"
Starting cowrie: [twistd  --umask=0022 --pidfile=var/run/cowrie.pid --logger cowrie.python.logfile.logger
 cowrie ]...
/home/cowrie/cowrie/cowrie-env/lib/python3.10/site-packages/twisted/conch/ssh/transport.py:97: Cryptograp
hyDeprecationWarning: Blowfish has been deprecated
  b"blowfish-cbc": (algorithms.Blowfish, 16, modes.CBC),
/home/cowrie/cowrie/cowrie-env/lib/python3.10/site-packages/twisted/conch/ssh/transport.py:101: Cryptogra
phyDeprecationWarning: CAST5 has been deprecated
```

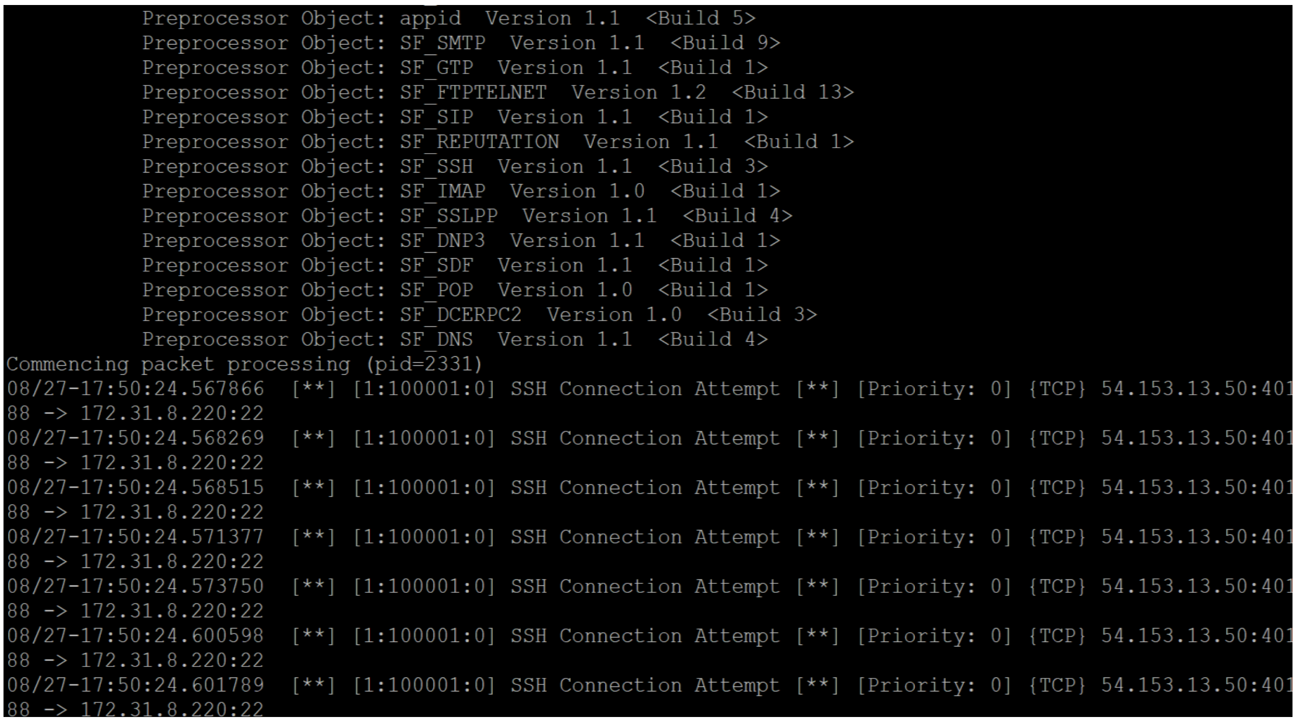


```
(cowrie-env) cowrie@ip-172-31-8-220:~/cowrie$ bin/cowrie status
cowrie is running (PID: 18702).
(cowrie-env) cowrie@ip-172-31-8-220:~/cowrie$ tail -f /home/cowrie/cowrie/var/log/cowrie/cowrie.log
2023-08-24T18:05:58.396357Z [twisted.scripts._twistd_unix.UnixAppLogger#info] reactor class: twisted.inte
rnet.epollreactor.EPollReactor.
2023-08-24T18:05:58.402019Z [-] CowrieSSHFactory starting on 2222
2023-08-24T18:05:58.402632Z [cowrie.ssh.factory.CowrieSSHFactory#info] Starting factory <cowrie.ssh.facto
ry.CowrieSSHFactory object at 0x7f6dd8eeb820>
2023-08-24T18:05:58.459457Z [-] Ready to accept SSH connections
2023-08-24T18:05:58.460205Z [-] HoneyPotTelnetFactory starting on 2223
2023-08-24T18:05:58.460324Z [cowrie.telnet.factory.HoneyPotTelnetFactory#info] Starting factory <cowrie.t
elnet.factory.HoneyPotTelnetFactory object at 0x7f6dd8eeb940>
2023-08-24T18:05:58.460509Z [-] Ready to accept Telnet connections
2023-08-24T18:14:18.504360Z [cowrie.telnet.factory.HoneyPotTelnetFactory] New connection: 47.108.228.233:
51044 (172.31.8.220:2223) [session: 84d9cdb0e3c3]
2023-08-24T18:14:49.107061Z [CowrieTelnetTransport,0,47.108.228.233] Connection lost after 30 seconds
2023-08-24T18:15:34.724600Z [cowrie.telnet.factory.HoneyPotTelnetFactory] New connection: 190.83.104.10:3
5054 (172.31.8.220:2223) [session: f81c7efd51c6]
2023-08-24T18:17:34.800910Z [-] Timeout reached in CowrieTelnetTransport
2023-08-24T18:17:34.801438Z [CowrieTelnetTransport,1,190.83.104.10] Connection lost after 120 seconds
2023-08-24T18:17:35.115389Z [cowrie.telnet.factory.HoneyPotTelnetFactory] New connection: 190.83.104.10:3
7947 (172.31.8.220:2223) [session: 1d5a8f950e8c]
2023-08-24T18:19:35.165444Z [-] Timeout reached in CowrieTelnetTransport
2023-08-24T18:19:35.165963Z [CowrieTelnetTransport,2,190.83.104.10] Connection lost after 120 seconds
2023-08-24T18:19:35.516592Z [cowrie.telnet.factory.HoneyPotTelnetFactory] New connection: 190.83.104.10:3
8627 (172.31.8.220:2223) [session: 9de6857babee]
2023-08-24T18:21:35.617076Z [-] Timeout reached in CowrieTelnetTransport
2023-08-24T18:21:35.617659Z [CowrieTelnetTransport,3,190.83.104.10] Connection lost after 120 seconds
```



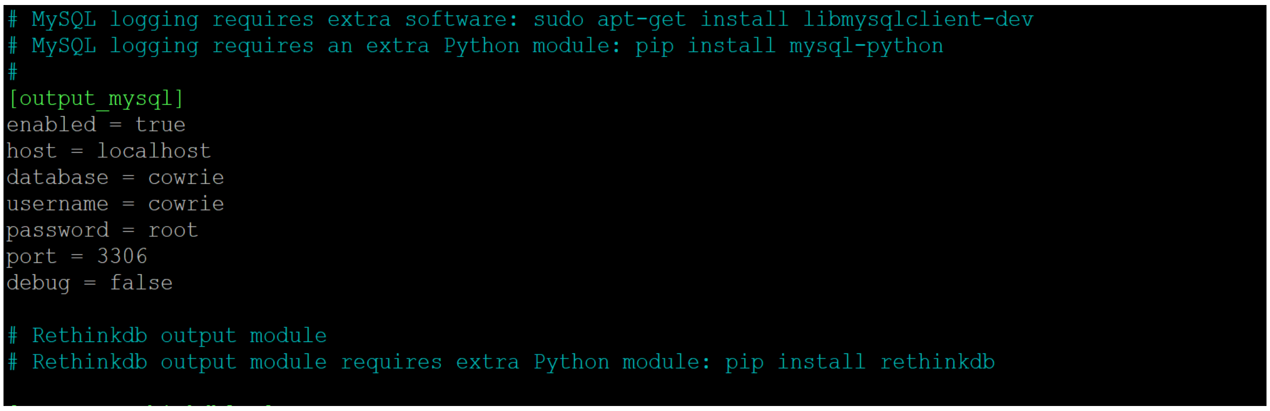


```
           Preprocessor Object: SF_GTP  Version 1.1  <Build 1>
           Preprocessor Object: SF_FTPTELNET  Version 1.2  <Build 13>
           Preprocessor Object: SF_SIP  Version 1.1  <Build 1>
           Preprocessor Object: SF_REPUTATION  Version 1.1  <Build 1>
           Preprocessor Object: SF_SSH  Version 1.1  <Build 3>
           Preprocessor Object: SF_IMAP  Version 1.0  <Build 1>
           Preprocessor Object: SF_SSLPP  Version 1.1  <Build 4>
           Preprocessor Object: SF_DNP3  Version 1.1  <Build 1>
           Preprocessor Object: SF_SDF  Version 1.1  <Build 1>
           Preprocessor Object: SF_POP  Version 1.0  <Build 1>
           Preprocessor Object: SF_DCERPC2  Version 1.0  <Build 3>
           Preprocessor Object: SF_DNS  Version 1.1  <Build 4>
Commencing packet processing (pid=2345)
08/27-17:52:07.606852  [**] [1:100002:0] HTTP traffic detected [**] [Priority: 0] {TCP} 152.57.206.185:53
384 -> 172.31.8.220:80
08/27-17:52:07.616424  [**] [1:100002:0] HTTP traffic detected [**] [Priority: 0] {TCP} 152.57.206.185:53
386 -> 172.31.8.220:80
08/27-17:52:08.046268  [**] [1:100002:0] HTTP traffic detected [**] [Priority: 0] {TCP} 152.57.206.185:53
386 -> 172.31.8.220:80
08/27-17:52:13.116184  [**] [1:100002:0] HTTP traffic detected [**] [Priority: 0] {TCP} 152.57.206.185:53
386 -> 172.31.8.220:80
```

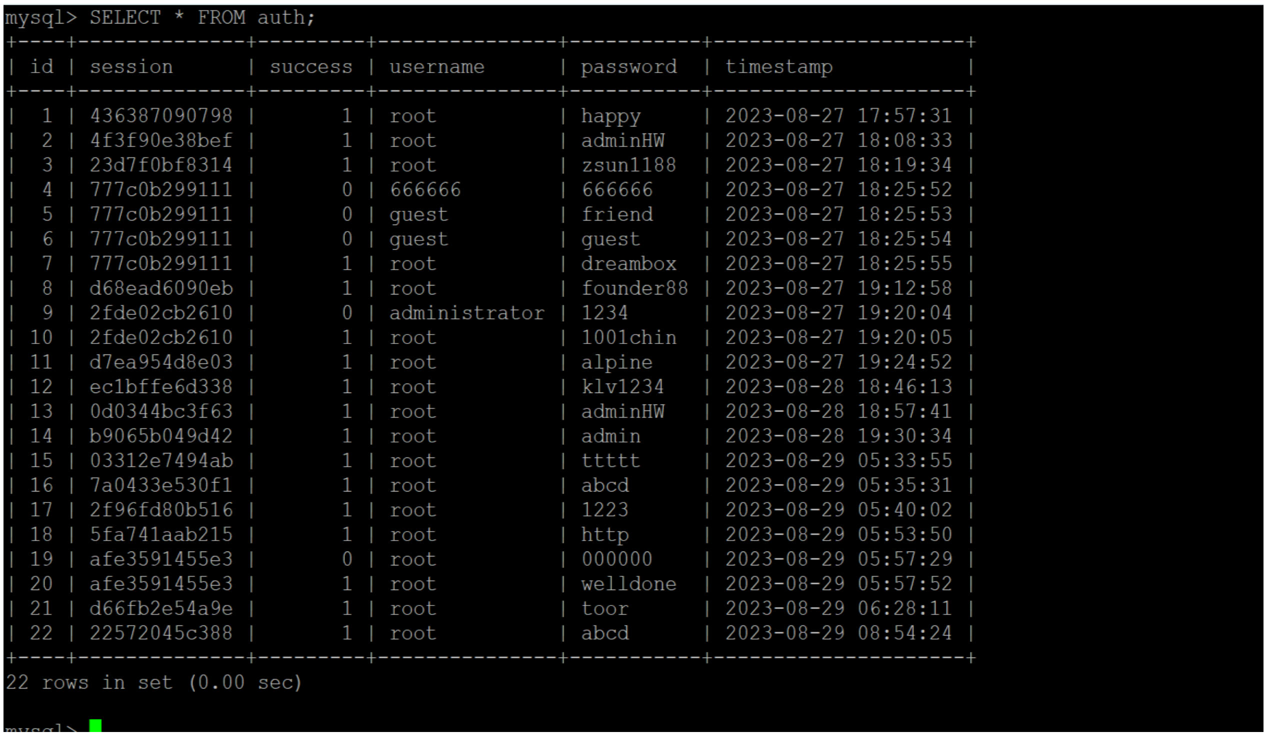


```
           Preprocessor Object: appid  Version 1.1  <Build 5>
           Preprocessor Object: SF_SMTP  Version 1.1  <Build 9>
           Preprocessor Object: SF_GTP  Version 1.1  <Build 1>
           Preprocessor Object: SF_FTPTELNET  Version 1.2  <Build 13>
           Preprocessor Object: SF_SIP  Version 1.1  <Build 1>
           Preprocessor Object: SF_REPUTATION  Version 1.1  <Build 1>
           Preprocessor Object: SF_SSH  Version 1.1  <Build 3>
           Preprocessor Object: SF_IMAP  Version 1.0  <Build 1>
           Preprocessor Object: SF_SSLPP  Version 1.1  <Build 4>
           Preprocessor Object: SF_DNP3  Version 1.1  <Build 1>
           Preprocessor Object: SF_SDF  Version 1.1  <Build 1>
           Preprocessor Object: SF_POP  Version 1.0  <Build 1>
           Preprocessor Object: SF_DCERPC2  Version 1.0  <Build 3>
           Preprocessor Object: SF_DNS  Version 1.1  <Build 4>
Commencing packet processing (pid=2331)
08/27-17:50:24.567866  [**] [1:100001:0] SSH Connection Attempt [**] [Priority: 0] {TCP} 54.153.13.50:401
88 -> 172.31.8.220:22
08/27-17:50:24.568269  [**] [1:100001:0] SSH Connection Attempt [**] [Priority: 0] {TCP} 54.153.13.50:401
88 -> 172.31.8.220:22
08/27-17:50:24.568515  [**] [1:100001:0] SSH Connection Attempt [**] [Priority: 0] {TCP} 54.153.13.50:401
88 -> 172.31.8.220:22
08/27-17:50:24.571377  [**] [1:100001:0] SSH Connection Attempt [**] [Priority: 0] {TCP} 54.153.13.50:401
88 -> 172.31.8.220:22
08/27-17:50:24.573750  [**] [1:100001:0] SSH Connection Attempt [**] [Priority: 0] {TCP} 54.153.13.50:401
88 -> 172.31.8.220:22
08/27-17:50:24.600598  [**] [1:100001:0] SSH Connection Attempt [**] [Priority: 0] {TCP} 54.153.13.50:401
88 -> 172.31.8.220:22
08/27-17:50:24.601789  [**] [1:100001:0] SSH Connection Attempt [**] [Priority: 0] {TCP} 54.153.13.50:401
88 -> 172.31.8.220:22
```

7.6 Database Configuration:



```
# MySQL logging requires extra software: sudo apt-get install libmysqlclient-dev
# MySQL logging requires an extra Python module: pip install mysql-python
#
[output_mysql]
enabled = true
host = localhost
database = cowrie
username = cowrie
password = root
port = 3306
debug = false

# Rethinkdb output module
# Rethinkdb output module requires extra Python module: pip install rethinkdb
```



```
mysql> SELECT * FROM auth;
+----+--------------+---------+---------------+-----------+---------------------+
| id | session      | success | username      | password  | timestamp           |
+----+--------------+---------+---------------+-----------+---------------------+
|  1 | 436387090798 |       1 | root          | happy     | 2023-08-27 17:57:31 |
|  2 | 4f3f90e38bef |       1 | root          | adminHW   | 2023-08-27 18:08:33 |
|  3 | 23d7f0bf8314 |       1 | root          | zsun1188  | 2023-08-27 18:19:34 |
|  4 | 777c0b299111 |       0 | 666666        | 666666    | 2023-08-27 18:25:52 |
|  5 | 777c0b299111 |       0 | guest         | friend    | 2023-08-27 18:25:53 |
|  6 | 777c0b299111 |       0 | guest         | guest     | 2023-08-27 18:25:54 |
|  7 | 777c0b299111 |       1 | root          | dreambox  | 2023-08-27 18:25:55 |
|  8 | d68ead6090eb |       1 | root          | founder88 | 2023-08-27 19:12:58 |
|  9 | 2fde02cb2610 |       0 | administrator | 1234      | 2023-08-27 19:20:04 |
| 10 | 2fde02cb2610 |       1 | root          | 1001chin  | 2023-08-27 19:20:05 |
| 11 | d7ea954d8e03 |       1 | root          | alpine    | 2023-08-27 19:24:52 |
| 12 | ec1bffe6d338 |       1 | root          | klv1234   | 2023-08-28 18:46:13 |
| 13 | 0d0344bc3f63 |       1 | root          | adminHW   | 2023-08-28 18:57:41 |
| 14 | b9065b049d42 |       1 | root          | admin     | 2023-08-28 19:30:34 |
| 15 | 03312e7494ab |       1 | root          | ttttt     | 2023-08-29 05:33:55 |
| 16 | 7a0433e530f1 |       1 | root          | abcd      | 2023-08-29 05:35:31 |
| 17 | 2f96fd80b516 |       1 | root          | 1223      | 2023-08-29 05:40:02 |
| 18 | 5fa741aab215 |       1 | root          | http      | 2023-08-29 05:53:50 |
| 19 | afe3591455e3 |       0 | root          | 000000    | 2023-08-29 05:57:29 |
| 20 | afe3591455e3 |       1 | root          | welldone  | 2023-08-29 05:57:52 |
| 21 | d66fb2e54a9e |       1 | root          | toor      | 2023-08-29 06:28:11 |
| 22 | 22572045c388 |       1 | root          | abcd      | 2023-08-29 08:54:24 |
+----+--------------+---------+---------------+-----------+---------------------+
22 rows in set (0.00 sec)
```

8. CONCLUSION

The project's software and hardware requirements are designed to create a robust and controlled environment for simulating and analyzing cyber threats. By meeting these requirements, the project ensures the successful implementation of the honeypot with Snort, contributing to the proactive enhancement of cybersecurity strategies. The "Honeypot with Snort" project represents a comprehensive approach to understanding attacker behavior in a controlled environment. By combining the power of a honeypot, an IDS, log visualization tools, and a database, the project contributes to a more informed cybersecurity landscape. The steps outlined in this report illustrate the journey from environment setup to data analysis, empowering researchers to uncover insights that can be applied to real-world cybersecurity strategies.

1.Ethical Considerations:

Throughout this project, ethical guidelines were strictly followed. All testing was conducted within the boundaries of legal and responsible behavior. The honeypot environment was clearly labeled as a research environment to prevent confusion with production systems.

2.Future Enhancements:

Further customization of Cowrie's behavior to attract a wider range of attacks. Integration with additional tools for more comprehensive threat analysis. Exploration of advanced Snort rule configurations for detecting sophisticated attack

9. REFRENCES

- *Cowrie Documentation: https://github.com/cowrie/cowrie*
- *Snort Official Documentation: https://www.snort.org/documents*
- *MySQL Documentation. (n.d.). MySQL :: MySQL 8.0 Reference Manual. https://dev.mysql.com/doc/refman/8.0/en/*

Books:

1."Practical Packet Analysis: Using Wireshark to Solve Real-World Network Problems" by Chris Sanders and Gary Smith
2."Network Security Essentials: Applications and Standards" by William Stallings
3."The Web Application Hacker's Handbook: Finding and Exploiting Security Flaws" by Dafydd Stuttard and Marcus Pinto
4."Snort IDS and IPS Toolkit" by Andrew Baker, Brian Caswell, and Jay Beale