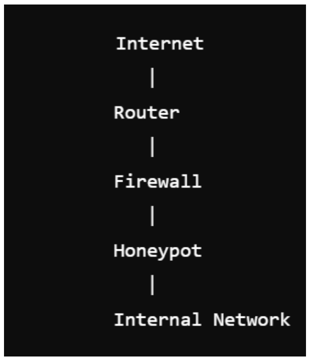
Honeypot Tools

A honeypot is a deliberately vulnerable system, device, network, or application connected to the network but not accessible from external parts of that network of interest. The honeypot is configured to be an enticing target for potential attackers. The attacker will be monitored and reviewed.

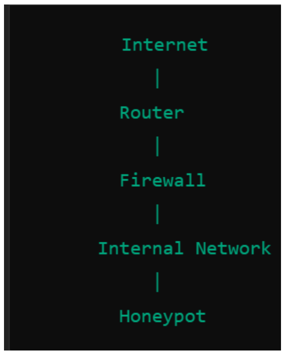
-Diagrams illustrating different topologies for placing honeypots in a network:

DMZ Placement:



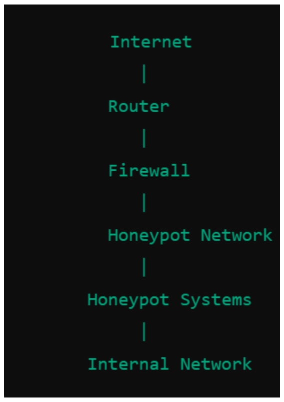
```
Internet
   |
Router
   |
Firewall
   |
Honeypot
   |
Internal Network
```

Inside the Internal Network:



```
Internet
   |
Router
   |
Firewall
   |
Internal Network
   |
Honeypot
```

Separate Network Segment:



```
Internet
   |
Router
   |
Firewall
   |
Honeypot Network
   |
Honeypot Systems
   |
Internal Network
```

Honeypot cybersecurity tools are designed to attract and detect cyber attackers by simulating real systems or networks. I will be testing the basic functionality of the following five honeypot tools in a virtual environment.

Here are the five honeypot tools that will be tested:

1. Endlessh = endlessh/README.md at master · skeeto/endlessh (github.com)
2. Hellpot = HellPot/README.md at main · yunginnanet/HellPot (github.com)
3. Honeyhttp = bocajspear1/honeyhttpd: HoneyHTTPD is a Python-based web server honeypot/service imitation builder. Great for honeypots or faking HTTP services. (github.com)
4. Cowrie = cowrie/cowrie: Cowrie SSH/Telnet Honeypot https://cowrie.readthedocs.io (github.com)
5. Conpot = conpot/README.md at master · mushorg/conpot (github.com)

1.Endlessh:

Description: Endlessh is a simple, lightweight honeypot that creates an endless connection to deter automated SSH attacks.

Features:

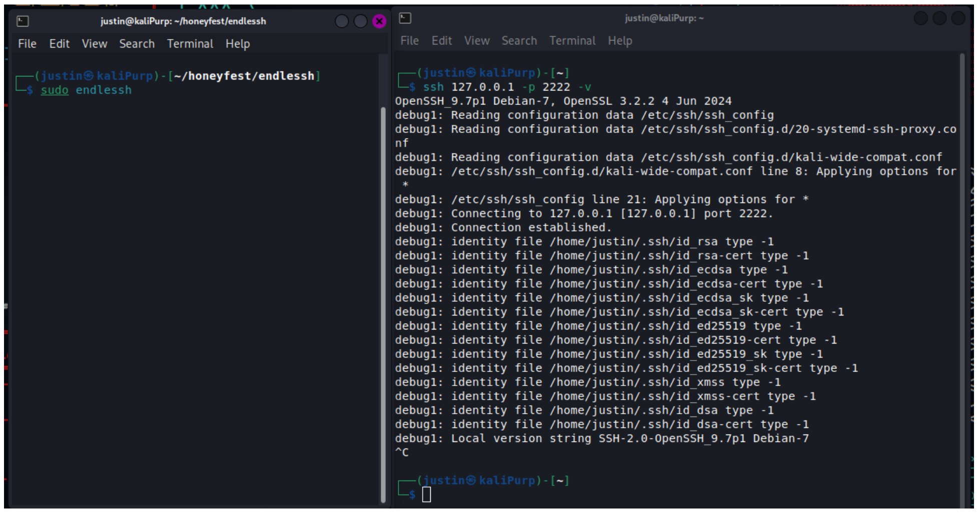
- Creates endless SSH connections
- Low resource usage
- Simple setup and configuration

Honeypot Terminal

-Listening on port 2222

Attacker Terminal

- ***Output*:** The attacker is kept in an endless loop, preventing further progress.


justin@kaliPurp: ~/honeyfest/endlessh
File Edit View Search Terminal Help
justin@kaliPurp: ~
File Edit View Search Terminal Help

```
┌──(justin㉿kaliPurp)-[~/honeyfest/endlessh]
└─$ sudo endlessh
```

```
┌──(justin㉿kaliPurp)-[~]
└─$ ssh 127.0.0.1 -p 2222 -v
OpenSSH_9.7p1 Debian-7, OpenSSL 3.2.2 4 Jun 2024
debug1: Reading configuration data /etc/ssh/ssh_config
debug1: Reading configuration data /etc/ssh/ssh_config.d/20-systemd-ssh-proxy.co
nf
debug1: Reading configuration data /etc/ssh/ssh_config.d/kali-wide-compat.conf
debug1: /etc/ssh/ssh_config.d/kali-wide-compat.conf line 8: Applying options for
 *
debug1: /etc/ssh/ssh_config line 21: Applying options for *
debug1: Connecting to 127.0.0.1 [127.0.0.1] port 2222.
debug1: Connection established.
debug1: identity file /home/justin/.ssh/id_rsa type -1
debug1: identity file /home/justin/.ssh/id_rsa-cert type -1
debug1: identity file /home/justin/.ssh/id_ecdsa type -1
debug1: identity file /home/justin/.ssh/id_ecdsa-cert type -1
debug1: identity file /home/justin/.ssh/id_ecdsa_sk type -1
debug1: identity file /home/justin/.ssh/id_ecdsa_sk-cert type -1
debug1: identity file /home/justin/.ssh/id_ed25519 type -1
debug1: identity file /home/justin/.ssh/id_ed25519-cert type -1
debug1: identity file /home/justin/.ssh/id_ed25519_sk type -1
debug1: identity file /home/justin/.ssh/id_ed25519_sk-cert type -1
debug1: identity file /home/justin/.ssh/id_xmss type -1
debug1: identity file /home/justin/.ssh/id_xmss-cert type -1
debug1: identity file /home/justin/.ssh/id_dsa type -1
debug1: identity file /home/justin/.ssh/id_dsa-cert type -1
debug1: Local version string SSH-2.0-OpenSSH_9.7p1 Debian-7
^C

┌──(justin㉿kaliPurp)-[~]
└─$
```

2.Hellpot:

Description: Hellpot is designed to mislead attackers by mimicking a vulnerable server and recording their actions.

Features:

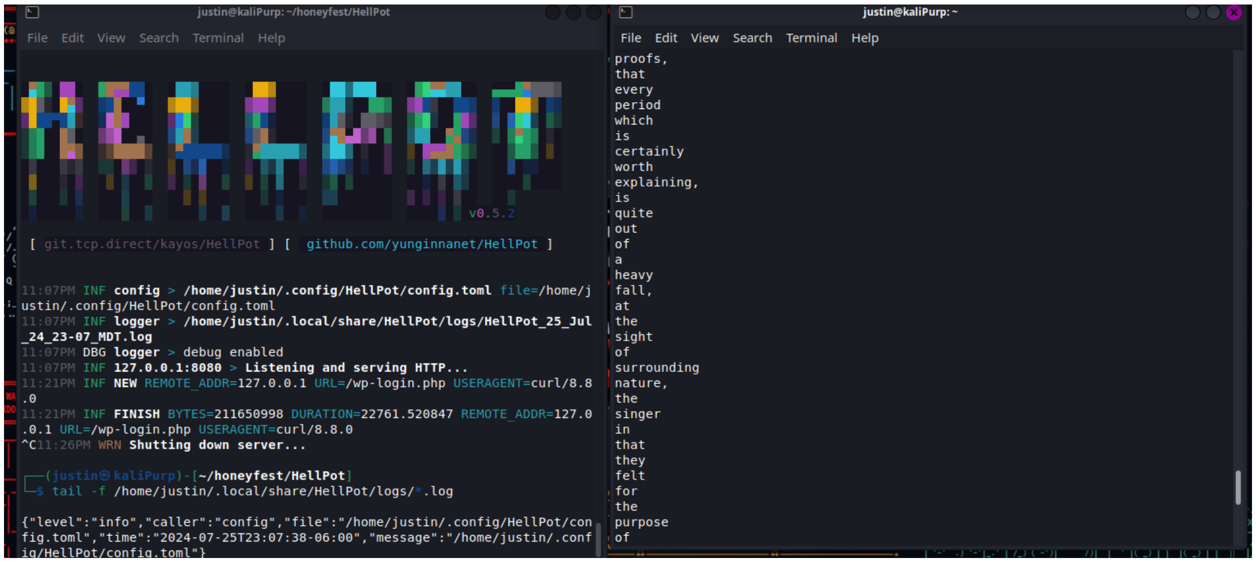
- Mimics vulnerable servers
- Logs attacker interactions
- Configurable deceptive responses

Honeypot Terminal

-Listening on port 8080

Attacker Terminal

- *Output*: Attackers receive deceptive responses, and their activities are logged.


justin@kaliPurp: ~/honeyfest/HellPot
File Edit View Search Terminal Help
v0.5.2
justin@kaliPurp: ~
File Edit View Search Terminal Help

```
 [ git.tcp.direct/kayos/HellPot ] [  github.com/yunginnanet/HellPot ]


11:07PM INF config > /home/justin/.config/HellPot/config.toml file=/home/j
ustin/.config/HellPot/config.toml
11:07PM INF logger > /home/justin/.local/share/HellPot/logs/HellPot_25_Jul
_24_23-07_MDT.log
11:07PM DBG logger > debug enabled
11:07PM INF 127.0.0.1:8080 > Listening and serving HTTP...
11:21PM INF NEW REMOTE_ADDR=127.0.0.1 URL=/wp-login.php USERAGENT=curl/8.8
.0
11:21PM INF FINISH BYTES=211650998 DURATION=22761.520847 REMOTE_ADDR=127.0
.0.1 URL=/wp-login.php USERAGENT=curl/8.8.0
^C11:26PM WRN Shutting down server...

┌──(justin㉿kaliPurp)-[~/honeyfest/HellPot]
└─$ tail -f /home/justin/.local/share/HellPot/logs/*.log

{"level":"info","caller":"config","file":"/home/justin/.config/HellPot/con
fig.toml","time":"2024-07-25T23:07:38-06:00","message":"/home/justin/.conf
ig/HellPot/config.toml"}
```

```
proofs,
that
every
period
which
is
certainly
worth
explaining,
is
quite
out
of
a
heavy
fall,
at
the
sight
of
surrounding
nature,
the
singer
in
that
they
felt
for
the
purpose
of
```

3.HoneyHTTP:

Description: HoneyHTTPD is a Python-based web server honeypot/service imitation builder that logs interactions.

Features:

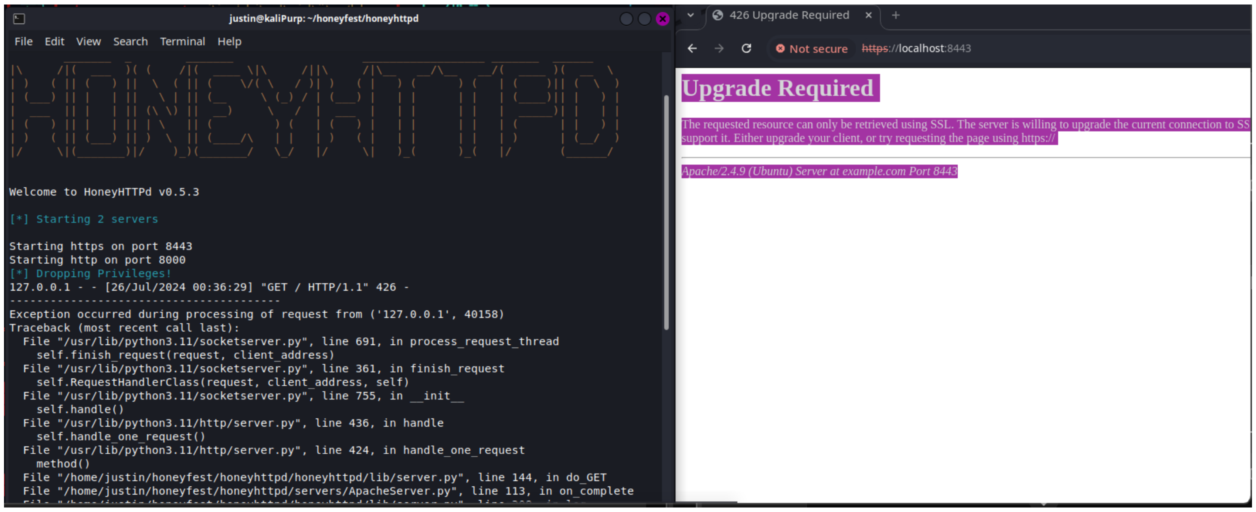
- Python-based web server honeypot
- Customizable responses
- Logs interactions to files, ElasticSearch, or AWS S3

Honeypot Terminal

-Created http & https servers

Attacker Terminal

- ***Output*:** Attackers receive custom responses; interactions are logged.


justin@kaliPurp: ~/honeyfest/honeyhttpd
File Edit View Search Terminal Help

```
Welcome to HoneyHTTPd v0.5.3

[*] Starting 2 servers

Starting https on port 8443
Starting http on port 8000
[*] Dropping Privileges!
127.0.0.1 - - [26/Jul/2024 00:36:29] "GET / HTTP/1.1" 426 -
----------------------------------------
Exception occurred during processing of request from ('127.0.0.1', 40158)
Traceback (most recent call last):
  File "/usr/lib/python3.11/socketserver.py", line 691, in process_request_thread
    self.finish_request(request, client_address)
  File "/usr/lib/python3.11/socketserver.py", line 361, in finish_request
    self.RequestHandlerClass(request, client_address, self)
  File "/usr/lib/python3.11/socketserver.py", line 755, in __init__
    self.handle()
  File "/usr/lib/python3.11/http/server.py", line 436, in handle
    self.handle_one_request()
  File "/usr/lib/python3.11/http/server.py", line 424, in handle_one_request
    method()
  File "/home/justin/honeyfest/honeyhttpd/honeyhttpd/lib/server.py", line 144, in do_GET
  File "/home/justin/honeyfest/honeyhttpd/servers/ApacheServer.py", line 113, in on_complete
```

Upgrade Required

The requested resource can only be retrieved using SSL. The server is willing to upgrade the current connection to SS
support it. Either upgrade your client, or try requesting the page using https://

Apache/2.4.9 (Ubuntu) Server at example.com Port 8443

4. Cowrie:

Description: Cowrie is an SSH and Telnet honeypot that logs brute force attacks and shell interaction performed by the attacker.

Features:

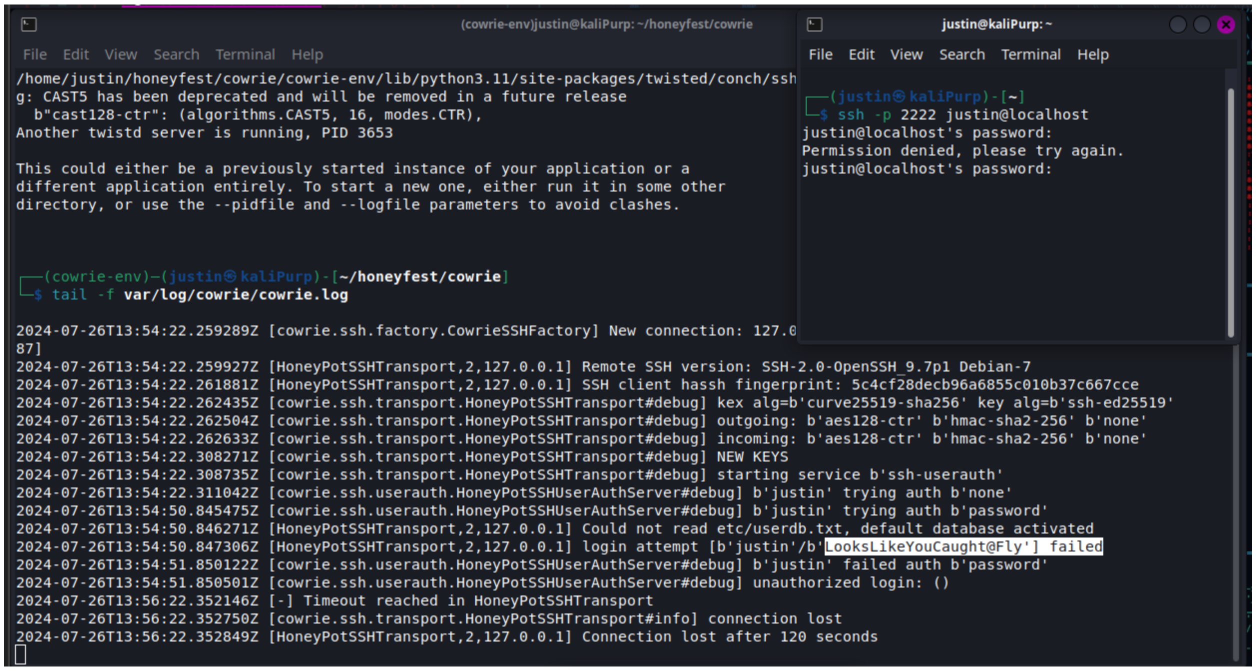
- Emulates an SSH and Telnet server
- Logs attacker commands and interactions
- Provides detailed logging and session replay

Honeypot Terminal

-Listening on port 2222

Attacker Terminal

-**Output:** Attackers' login attempt and interaction logged and monitored



(cowrie-env)justin@kaliPurp: ~/honeyfest/cowrie

```
/home/justin/honeyfest/cowrie/cowrie-env/lib/python3.11/site-packages/twisted/conch/ssh
g: CAST5 has been deprecated and will be removed in a future release
  b"cast128-ctr": (algorithms.CAST5, 16, modes.CTR),
Another twistd server is running, PID 3653

This could either be a previously started instance of your application or a
different application entirely. To start a new one, either run it in some other
directory, or use the --pidfile and --logfile parameters to avoid clashes.



┌──(cowrie-env)─(justin㉿kaliPurp)-[~/honeyfest/cowrie]
└─$ tail -f var/log/cowrie/cowrie.log

2024-07-26T13:54:22.259289Z [cowrie.ssh.factory.CowrieSSHFactory] New connection: 127.0
87]
2024-07-26T13:54:22.259927Z [HoneyPotSSHTransport,2,127.0.0.1] Remote SSH version: SSH-2.0-OpenSSH_9.7p1 Debian-7
2024-07-26T13:54:22.261881Z [HoneyPotSSHTransport,2,127.0.0.1] SSH client hassh fingerprint: 5c4cf28decb96a6855c010b37c667cce
2024-07-26T13:54:22.262435Z [cowrie.ssh.transport.HoneyPotSSHTransport#debug] kex alg=b'curve25519-sha256' key alg=b'ssh-ed25519'
2024-07-26T13:54:22.262504Z [cowrie.ssh.transport.HoneyPotSSHTransport#debug] outgoing: b'aes128-ctr' b'hmac-sha2-256' b'none'
2024-07-26T13:54:22.262633Z [cowrie.ssh.transport.HoneyPotSSHTransport#debug] incoming: b'aes128-ctr' b'hmac-sha2-256' b'none'
2024-07-26T13:54:22.308271Z [cowrie.ssh.transport.HoneyPotSSHTransport#debug] NEW KEYS
2024-07-26T13:54:22.308735Z [cowrie.ssh.transport.HoneyPotSSHTransport#debug] starting service b'ssh-userauth'
2024-07-26T13:54:22.311042Z [cowrie.ssh.userauth.HoneyPotSSHUserAuthServer#debug] b'justin' trying auth b'none'
2024-07-26T13:54:50.845475Z [cowrie.ssh.userauth.HoneyPotSSHUserAuthServer#debug] b'justin' trying auth b'password'
2024-07-26T13:54:50.846271Z [HoneyPotSSHTransport,2,127.0.0.1] Could not read etc/userdb.txt, default database activated
2024-07-26T13:54:50.847306Z [HoneyPotSSHTransport,2,127.0.0.1] login attempt [b'justin'/b'LooksLikeYouCaught@Fly'] failed
2024-07-26T13:54:51.850122Z [cowrie.ssh.userauth.HoneyPotSSHUserAuthServer#debug] b'justin' failed auth b'password'
2024-07-26T13:54:51.850501Z [cowrie.ssh.userauth.HoneyPotSSHUserAuthServer#debug] unauthorized login: ()
2024-07-26T13:56:22.352146Z [-] Timeout reached in HoneyPotSSHTransport
2024-07-26T13:56:22.352750Z [cowrie.ssh.transport.HoneyPotSSHTransport#info] connection lost
2024-07-26T13:56:22.352849Z [HoneyPotSSHTransport,2,127.0.0.1] Connection lost after 120 seconds
```

justin@kaliPurp: ~

```
┌──(justin㉿kaliPurp)-[~]
└─$ ssh -p 2222 justin@localhost
justin@localhost's password:
Permission denied, please try again.
justin@localhost's password:
```

5.Conpot:

Description: Conpot is a low-interaction honeypot designed to simulate industrial control systems (ICS) to attract and interact with potential attackers.

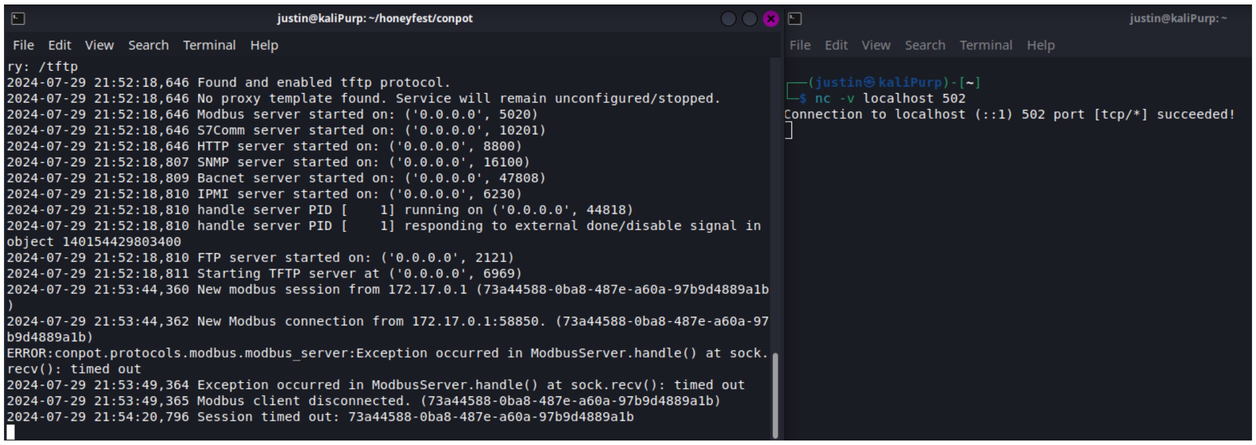
Features:

- Simulates various industrial control protocols (e.g., Modbus, S7comm)
- Logs interactions and provides detailed activity reports
- Configurable templates to simulate different types of ICS

Honeypot Terminal

-Listening on multiple ports (e.g., 5020 for Modbus, 10201 for S7comm, etc.)

-**Output**: Logs detailed interactions with the simulated ICS environment


justin@kaliPurp: ~/honeyfest/conpot
File Edit View Search Terminal Help
justin@kaliPurp: ~
File Edit View Search Terminal Help

```
ry: /tftp
2024-07-29 21:52:18,646 Found and enabled tftp protocol.
2024-07-29 21:52:18,646 No proxy template found. Service will remain unconfigured/stopped.
2024-07-29 21:52:18,646 Modbus server started on: ('0.0.0.0', 5020)
2024-07-29 21:52:18,646 S7Comm server started on: ('0.0.0.0', 10201)
2024-07-29 21:52:18,646 HTTP server started on: ('0.0.0.0', 8800)
2024-07-29 21:52:18,807 SNMP server started on: ('0.0.0.0', 16100)
2024-07-29 21:52:18,809 Bacnet server started on: ('0.0.0.0', 47808)
2024-07-29 21:52:18,810 IPMI server started on: ('0.0.0.0', 6230)
2024-07-29 21:52:18,810 handle server PID [    1] running on ('0.0.0.0', 44818)
2024-07-29 21:52:18,810 handle server PID [    1] responding to external done/disable signal in
object 140154429803400
2024-07-29 21:52:18,810 FTP server started on: ('0.0.0.0', 2121)
2024-07-29 21:52:18,811 Starting TFTP server at ('0.0.0.0', 6969)
2024-07-29 21:53:44,360 New modbus session from 172.17.0.1 (73a44588-0ba8-487e-a60a-97b9d4889a1b
)
2024-07-29 21:53:44,362 New Modbus connection from 172.17.0.1:58850. (73a44588-0ba8-487e-a60a-97
b9d4889a1b)
ERROR:conpot.protocols.modbus.modbus_server:Exception occurred in ModbusServer.handle() at sock.
recv(): timed out
2024-07-29 21:53:49,364 Exception occurred in ModbusServer.handle() at sock.recv(): timed out
2024-07-29 21:53:49,365 Modbus client disconnected. (73a44588-0ba8-487e-a60a-97b9d4889a1b)
2024-07-29 21:54:20,796 Session timed out: 73a44588-0ba8-487e-a60a-97b9d4889a1b
```

```
┌──(justin㉿kaliPurp)-[~]
└─$ nc -v localhost 502
Connection to localhost (::1) 502 port [tcp/*] succeeded!
```

Attacker Terminal:

-**Output:** Attackers attempting to interact with the simulated ICS services receive responses and their actions are logged for analysis

Conclusion:

This document provides an overview of the honeypot tools tested, their features, and the expected interactions from both the honeypot and attacker perspectives. The included screenshots for each honeypot will further illustrate the functionality and effectiveness of these tools in detecting and logging unauthorized access attempts.

There are many more types of honeypots available, each tailored for different purposes and environments. For a comprehensive list of honeypot resources, you can refer to the following link:

Awesome Honeypots = paralax/awesome-honeypots: an awesome list of honeypot resources (github.com)