Honeypots

Carles Mateu

Requirements

To do this labs you'll need:

- To have the CentOS 8 virtual machine for the lab. Available at: http://codeandbeer.org/virtual/Seguretat/OVA/centos-IDS.ova

> **IMPORTANT**
> We should have VirtualBox running with a CentOS virtual machine, with the network on NAT network, the provided one is already at step 3 of this guide (to save time).

Cowrie (SSH honeypot) installation

1. We start by installing and upgrading our CentOS system, and adding some requirements:

```
dnf upgrade -y


dnf install -y python3 python3-virtualenv python3-pip git
    firewalld
```

2. **After upgrade reboot**
3. We will use a special user for the setup:

```
useradd cowrie
```

4. Enable the firewall and start it

```
systemctl enable --now firewalld.service
```

5. Add access to port 2222 (used by cowrie), not permanent, we will redirect later:

```
firewall-cmd --add-port 2222/tcp
```

6. Become the user just created:

```
su - cowrie
```

7. We will get the last cowrie release from github:

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

8. Cowrie is written in python, so we will keep the required libraries for cowrie apart of the system libreries. To do so, we will create a virtualenv.

```
cd cowrie
virtualenv-3 --python=/usr/bin/python3 cowrie-venv
. cowrie-venv/bin/activate
```

IMPORTANT
Pay attention to the activation command, it's a dot (.) followed by space, followed by the call to activate.

and install the requirements

```
pip3 install -r requirements.txt
```

Testing cowrie

9. Try to run cowrie:

```
bin/cowrie start
```

Result shoud be similar to

```
(cowrie-venv) [cowrie@ids cowrie]$ bin/cowrie start
Using activated Python virtual environment "/home/cowrie/
    cowrie/cowrie-venv"
Starting cowrie: [twistd   --umask=0022 --pidfile=var/run
    /cowrie.pid --logger cowrie.python.logfile.logger
    cowrie ]...
(cowrie-venv) [cowrie@ids cowrie]$ ss -lntp
State  Recv-Q  Send-Q      Local Address:Port     Peer
    Address:Port
LISTEN 0       128               0.0.0.0:22
    0.0.0.0:*
LISTEN 0       50                0.0.0.0:2222
    0.0.0.0:*      users:(("twistd",pid=4610,fd=11))
```

```
LISTEN 0       128                   [::]:22
                  [::]:*
(cowrie-venv) [cowrie@ids cowrie]$
```

10. With cowrie running we can log into our machine (with SSH)

```
ssh root@192.168.1.206 -p 2222
```

With this result:

```
(cowrie-venv) [cowrie@ids cowrie]$ ssh root@192.168.1.206 -p 2222
The authenticity of host '[192.168.1.206]:2222 ([192.168.1.206]:2222)' can't be established.
RSA key fingerprint is SHA256:VvySOAIonIGqSHejTU+w3uejoRN58SzOKKqkv/A1Gis.
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added '[192.168.1.206]:2222' (RSA) to the list of known hosts.
root@192.168.1.206'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:~# pwd
/root
root@svr04:~# id
uid=0(root) gid=0(root) groups=0(root)
root@svr04:~#
root@svr04:~# Connection to 192.168.1.206 closed.
```

As you can see, we have a “linux” system running and attending port 2222 by SSH.

11. Stop cowrie now.

```
bin/cowrie stop
```

Persistent systemd configuration

12. Edit bin/cowrie to change

```
#COWRIE_VIRTUAL_ENV=my-env
DAEMONIZE=""
STDOUT="no"
```

to

```
COWRIE_VIRTUAL_ENV=cowrie-venv
DAEMONIZE="-y"
STDOUT="yes"
```

13. As root:

```
cat << EOF > /etc/systemd/system/cowrie-honeypot.socket
[Unit]
Description=Cowrie SSH and Telnet honeypot socket
PartOf=cowrie-honeypot.service

[Socket]
ListenStream=0.0.0.0:2222

[Install]
WantedBy=sockets.target
EOF
```

and

```
cat << EOF > /etc/systemd/system/cowrie-honeypot.service
[Unit]
Description=Cowrie SSH
Wants=network.target
Requires=cowrie.socket

[Service]
Type=simple
User=cowrie
Group=cowrie
ExecStart=/home/cowrie/cowrie/bin/cowrie start
Restart=on-failure
RestartSec=5
[Install]
WantedBy=multi-user.target
EOF
```

14. We can start cowrie then:

```
systemctl daemon-reload
systemctl start cowrie-honeypot.service
systemctl status cowrie-honeypot.service
```

It will fail (SELinux prevents it from running). To solve it change selinux from enforcing to permissive (it's beyond the scope today) in /etc/selinux/config

Port change

15. In real life we would want Cowrie to use port 22 (and we can use port 222, for example). To do that (not on the lab):

```
firewall-cmd --add-port 222/tcp --permanent
firewall-cmd --reload
firewall-cmd --list-ports
```

16. Change sshd from 22/TCP to 222/TCP. In /etc/ssh/sshd_config, uncomment the Port line to read:

```
Port 222
```

And restart sshd

```
systemctl restart sshd.service
ss -lntp
```

17. Log out and log back in, now in port 222 if connected by SSH.

18. Now remove the old 22/TCP permissions, and redirect SSH to cowrie:

```
firewall-cmd --remove-service ssh --permanent
firewall-cmd --reload
firewall-cmd --add-masquerade --permanent
firewall-cmd --add-forward-port=port=22:proto=tcp:toport
    =2222 --permanent
firewall-cmd --reload
```

Now ssh attacks will go to cowrie.

Uninstall 22/TCP to cowrie.

19. To undo the port redirects:

```
firewall-cmd --add-service ssh --permanent
firewall-cmd --remove-forward-port=port=22:proto=tcp:
    toport=2222 --permanent
firewall-cmd --reload
```

Check with

```
firewall-cmd --list-all-zones
```

and change back /etc/ssh/sshd_config

Logs

20. We can check what is going on on the cowrie logs:

/home/cowrie/cowrie/var/log/cowrie/

End

21. By now, stop cowrie (we will try another one)

```
systemctl stop cowrie-honeypot.service
```

Canary

22. We will now install Canary, another honeypot (also python), similarly to cowrie

```
dnf install -y gcc gcc-c++ python2 python2-devel libpcap-
    devel  redhat-rpm-config
useradd canary
su  - canary
virtualenv -p python2 canary-venv
. ./canary-venv/bin/activate
pip install opencanary
pip install pcapy scapy
opencanaryd --copyconfig
cp .opencanary.conf opencanary.conf
```

Last line is due to a bug in opencanary (not gets the right path when using su -).

23. We can now edit the config file (/home/canary/opencanary.conf) and enable http.

As root:

```
firewall-cmd --add-service http --permanent
firewall-cmd --reload
```

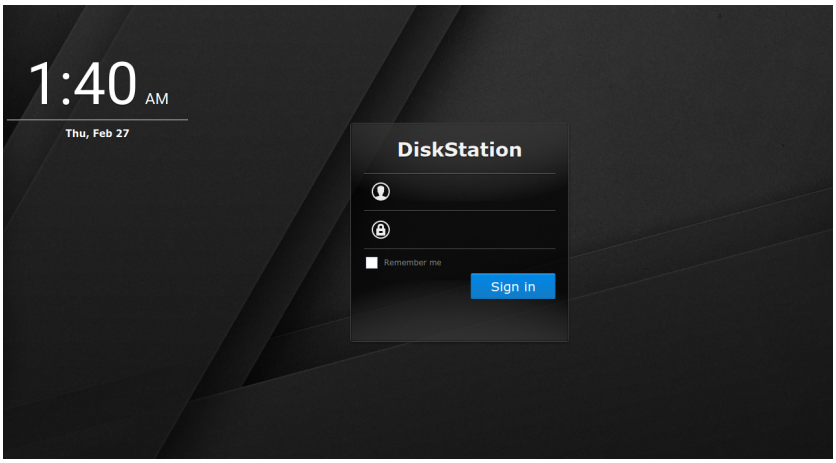
Back as canary:

```
opencanaryd --start
```

24. You will see that it starts an http server:

```
ss -tlnp
```

25. Connect to it with a browser:



26. And try to login, then check the logs in /var/tmp/opencanary.log