WSE 380: Intrusion Detection

Session 4 — Cowrie Honeypot Setup

Today, we are going to setup and launch a medium-interaction SSH honeypot named Cowrie.

1 Cowrie

Unlike the low-interaction honeypot we've been using up until this point, Cowrie emulates a real Linux system after authentication is complete. It also gives us more control over the honeypot by allowing us to specify which credentials are valid, and which files appear in the file system.

1.1 Downloading Cowrie

To download Cowrie, we are going to clone its Github repository:

```
wse380@wse380-22spr:~$ git clone https://github.com/cowrie/cowrie
```

We will then change directory into the newly created *cowrie* folder:

```
wse380@wse380-22spr:~$ cd cowrie
```

1.2 Setting up Cowrie

Now that we successfully downloaded Cowrie, we need to modify a few configuration files. Let's start by creating a copy of the example configuration files:

```
wse380@wse380-22spr:~/cowrie$ cp etc/cowrie.cfg.dist etc/cowrie.cfg
wse380@wse380-22spr:~/cowrie$ cp etc/userdb.example etc/userdb.txt
```

Although there are other files and tools included with Cowrie, which you can use to completely customize your honeypot, we are going to focus on these two files as they are the most important. The *cowrie.cfg* file is the main configuration file for Cowrie, while *userdb.txt* allows you to specify the user credentials attackers can login with. Open up the *cowrie.cfg* file and ensure that line 599 is the following:

```
listen_endpoints = tcp:2222:interface=0.0.0.0
```

Once you are happy with the configuration of Cowrie, there is one more step we need to take before we can start it. We set Cowrie to listen for connections on port 2222. However, SSH servers are typically located on port 22. As such, we want our honeypot to accept connections on that port as well. To accomplish this, we are going to set a firewall rule to redirect all traffic from port 22 to port 2222. Enter the following command into your terminal:

```
wse380@wse380-22spr:~$ sudo iptables -t nat -A PREROUTING -p tcp --dport 22 -j REDIRECT --to-port 2222
```

Now, any time an attacker tries to connect to port 22, their traffic will be redirected to your honeypot.

1.3 Installing Dependencies

Before we can actually run `cowrie`, we need to install its external dependencies/libraries. It is quite common to run into conflicting (version) dependencies when we work with multiple projects, so we create *virtual environments* that isolate all the dependencies for a given project. We will do this as well for cowrie (because we do in fact have a version conflict!). Run the following commands:

```
wse380@wse380-22spr:~/cowrie$ sudo apt install python3.8-venv
wse380@wse380-22spr:~/cowrie$ python3 -m venv ./venv
wse380@wse380-22spr:~/cowrie$ source venv/bin/activate
(venv) wse380@wse380-22spr:~/cowrie$ python -m pip install -r requirements.txt
```

1.4 Starting and Stopping Cowrie

Once you have finished configuring Cowrie, you can now start it. To do so, simply enter the following command from the root of the Cowrie directory:

```
(venv) wse380@wse380−22spr:˜/cowrie$ ./bin/cowrie start
```

This will start Cowrie in the background listening on port 2222. You can then stop Cowrie using a similar command from the root of the Cowrie directory:

```
(venv) wse380@wse380−22spr:˜/cowrie$ ./bin/cowrie stop
```

1.5 Testing Cowrie Connection

To test your `cowrie` deployment, you will need to attempt to SSH from a different machine, because the firewall rule that we added does not apply to loopback connections. Try to SSH again to your remote machine, this time with port 22 — what kind of output do you get? Let me know when you have reached this step, and I can help you verify your `cowrie` deployment.