

Lab9 – Cowrie Honeypot (@Home version)

Document your commands or take screenshots. Answer questions in english or finnish.

The lab uses preconfigured CentOS7 Virtual Machine with Docker. Use the following credentials for the VM:
root/root66

• Initial steps

Retrieve the pre-installed VM image for Cowrie SSH Honeypot from [\\ghost.labranet.jamk.fi\virtuaalikoneet\TTKS\Cowrie-Honeypot.ova](http://ghost.labranet.jamk.fi/virtuaalikoneet\TTKS\Cowrie-Honeypot.ova) and import appliance to VirtualBox. Set the network interface to Bridged and start the VM. If the VirtualBox nags about the interface configurations, press OK.

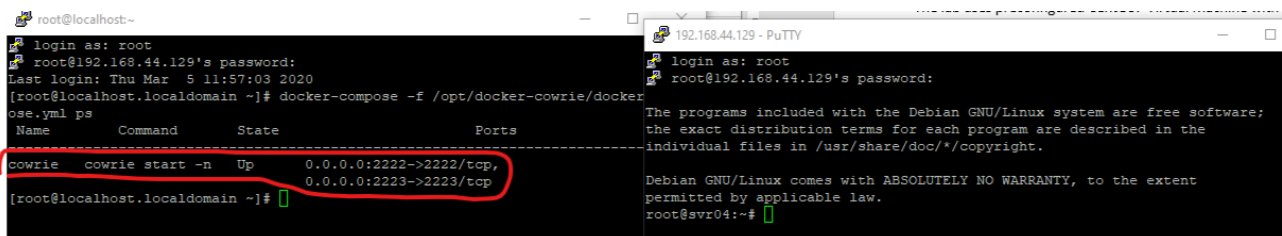
When you have login to VM check that your VM has retrieved IP address from the DHCP server and try to wget www.iltasanomat.fi, so you can verify that VM have access on Internet. Then run the following command:

```
docker-compose -f /opt/docker-cowrie/docker-compose.yml ps
```

You should be informed that there is Docker container named “cowrie” up and running.

• Create a connection to Virtual Machine using Host computer

Use your Putty or other SSH client such as PowerShell and create connection using VM machine’s IP address and port 2222. Use the following credentials: root/123456



```
root@localhost:~# docker-compose -f /opt/docker-cowrie/docker-compose.yml ps
```

Name	Command	State	Ports
cowrie	cowrie start -n	Up	0.0.0.0:2222->2222/tcp, 0.0.0.0:2223->2223/tcp

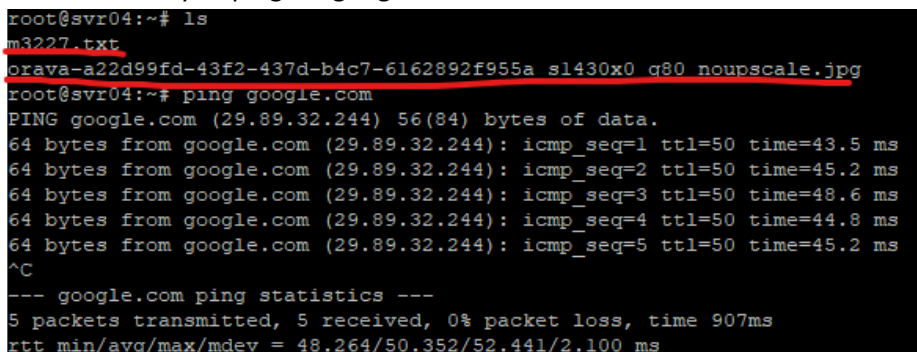
```
[root@localhost.localdomain ~]#
```

```
192.168.44.129 - PuTTY
```

```
login as: root
root@192.168.44.129's password:
Last login: Thu Mar  5 11:57:03 2020
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:~#
```

- docker, and ssh to cowrie

When you have established the connection, leave a mark that you were inside the machine using following command: touch <your-student-id>.txt, where <your-student-id> is your actual student ID. Then wget some web site and try to ping i.e. google.com.



```
root@svr04:~# ls
m3227.txt
brava-a22d99fd-43f2-437d-b4c7-6162892f955a_s1430x0_q80_noupscale.jpg
root@svr04:~# ping google.com
PING google.com (29.89.32.244) 56(84) bytes of data:
64 bytes from google.com (29.89.32.244): icmp_seq=1 ttl=50 time=43.5 ms
64 bytes from google.com (29.89.32.244): icmp_seq=2 ttl=50 time=45.2 ms
64 bytes from google.com (29.89.32.244): icmp_seq=3 ttl=50 time=48.6 ms
64 bytes from google.com (29.89.32.244): icmp_seq=4 ttl=50 time=44.8 ms
64 bytes from google.com (29.89.32.244): icmp_seq=5 ttl=50 time=45.2 ms
^C
--- google.com ping statistics ---
5 packets transmitted, 5 received, 0% packet loss, time 907ms
rtt min/avg/max/mdev = 48.264/50.352/52.441/2.100 ms
```

- txt file, wget:ted jpg file and ping

In addition, try to create a new user using command adduser or useadd. Can you? Finally, exit the SSH session.

```
Must enter a value!
    Other []: ei
Is the information correct? [Y/n] Y
Ok, starting over
```

- You cannot create new users. it creates a loop

• Exploring logs

Use VM's console and navigate to \$COWRIE_VAR/log/cowrie. In this directory locates cowrie.json file. What information this log file contains?

- file contains log info about what users do in target workstation.

Find the information about who has tried to login before you and which credentials were used. Find at least 4 attempts and write them down. Which account was used to successful login on honeypot?

Credentials	Success
jamk / ttk0800	yes
test / test	no
admin / admin	no
root / root66	no

Verify also that you can find your own login information from the log file (root/123456).

```
"eventId":"cowrie_login_success","username":"root","password":"123456","message":"Login attempt (root/123456) succeeded","source":"5ac59e1f226f","timestamp":"2020-03-05T12:00:36.5387150","src_ip":"192.168.44.35","session":"553c1646704"
```

- found

List the contents of the tty directory (\$COWRIE_VAR/lib/cowrie/tty). Directory should now contain a file that holding records about the actions that you took on honeypot. Copy the file from the tty directory to the \$COWRIE_BIN directory.

```
[root@localhost.localdomain tty]# cp de91dd65ebbd7c3aab3d06292eefb9da25acf6a6ecc0806640742e704f3da8a $COWRIE_BIN
[root@localhost.localdomain tty]# cd $COWRIE_BIN
[root@localhost.localdomain bin]# ls
asciinema  createdynamicprocess.py  de91dd65ebbd7c3aab3d06292eefb9da25acf6a6ecc0806640742e704f3da8a  playlog
cowrie     createfs                 fctcl
```

- file is found and is named de91...

Then replay the copied log file using the executable using following syntax:

\$COWRIE_BIN/playlog \$COWRIE_BIN/<name-of-the-log-file>

What does it show? Can you see the replay of the commands that you made before?

```
[root@localhost.localdomain bin]# $COWRIE_BIN/playlog $COWRIE_BIN/de91dd65ebbda7c3742e704f3da8a42e704f3da8a42e704f3da8a742e704f3da8a0

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
```

- it shows a replay of my input commands

• Changing Cowrie settings

By default, Cowrie is running at port 2222, and the actual SSH service is listening on its default port 22. We want to change things so that the SSH connection to the port 22 goes inside to the honeypot, and behind the port 2222 is the actual SSH service. Find a way to do that. Hint: Redirect the traffic from port 22 to 2222 and from port 2222 to 22 using iptables.

```
iptables -t nat -A PREROUTING -i eth0 -p tcp --dport 25 -j REDIRECT --to-port 2525
```

```
684 iptables -t nat -A PREROUTING -i eth0 -p tcp --dport 22 -j REDIRECT --to-port 2222
685 iptables -t nat -A PREROUTING -i eth0 -p tcp --dport 2222 -j REDIRECT --to-port 22
686 iptables -t nat -L -n -v
```

Find out how to add “fake user” for Cowrie, by fake means user that gets inside honeypot. Use following syntax for the credentials: <your-student-id>:x:<your-student-id>. Hint: Modify \$COWRIE_ETC/userdb.txt file and restart the Docker container using command:

```
# '*' for password allows any password
# '!' at the start of a password will not grant this password access
# '/' can be used to write a regular expression
#
root:x:!root
root:x:l23456
root:x:!/honeypot/i
jamk:x:ttks0800
tomcat:x:!tomcat
oracle:x:!oracle
m3227:x:m3227
```

docker-compose -f /opt/docker-cowrie/docker-compose.yml restart cowrie

```
Debian GNU/Linux comes with ABSOLUTELY NO WARRANTY, to the extent
permitted by applicable law.
m3227@svr04:~$
```

Prove that the changes you made works as expected:

- When you create SSH connection to port 22 and use credentials <your-student-id>/<your-student-id> for login, you end up inside honeypot
- When you create SSH connection to port 2222 and use credentials root/root66 for login, you end up inside host as a real root user
- I couldn't make it work, but im pretty sure this should do it.

```
[root@localhost.localdomain ~]# iptables -t nat -L -n -v
Chain PREROUTING (policy ACCEPT 0 packets, 0 bytes)
pkts bytes target      prot opt in     out     source    destination
  7   364 DOCKER      all  --  *      *        0.0.0.0/0  0.0.0.0/0      ADDRTYPE match dst-type LOCAL
  0     0 REDIRECT    tcp  --  eth0    *        0.0.0.0/0  0.0.0.0/0      tcp dpt:22 redir ports 2222
  0     0 REDIRECT    tcp  --  eth0    *        0.0.0.0/0  0.0.0.0/0      tcp dpt:22 redir ports 2222
  0     0 REDIRECT    tcp  --  eth0    *        0.0.0.0/0  0.0.0.0/0      tcp dpt:22 redir ports 2222
  0     0 REDIRECT    tcp  --  eth0    *        0.0.0.0/0  0.0.0.0/0      tcp dpt:2222 redir ports 22
```

- screenshot of vms iptables.s