

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:

```
cat /etc/network/interfaces
2: enp0s3: <BROADCAST,MULTICAST>
link/ether 08:00:27:1e:8
inet 192.168.43.131/24 b
```

```
Connecting to www.iltasanomat.fi (www.iltasanomat.fi)13.32.43.51:80... connected.
HTTP request sent, awaiting response... 301 Moved Permanently
Location: https://www.is.fi/ [following]
--2020-03-31 18:00:33-- https://www.is.fi/
Resolving www.is.fi (www.is.fi)... 13.32.43.35, 13.32.43.60, 13.32.43.90, ...
Connecting to www.is.fi (www.is.fi)13.32.43.35:443... connected.
HTTP request sent, awaiting response... 200 OK
Length: 202296 (198K) [text/html]
Saving to: 'index.html'

100%[=====] 202,296      --.-K/s   in 0.1s

2020-03-31 18:00:34 (1.44 MB/s) - 'index.html' saved [202296/202296]

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

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

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

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

- **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



192.168.43.131 - PuTTY

```
login as: root
root@192.168.43.131's password:

The programs included with the Debian GNU/Linux system have been
the exact distribution terms for each file, see the files
individual files in /usr/share/doc/*/*-*.txt for more details.

Debian GNU/Linux comes with ABSOLUTELY NO WARRANTY, but all rights
permitted by applicable law.
root@svr04:~#
```

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

```
root@svr04:~# touch M3426.txt
root@svr04:~# wget www.iltasanomat.fi
--2020-03-31 18:04:57-- http://www.iltasanomat.fi
Connecting to www.iltasanomat.fi:80... connected.
HTTP request sent, awaiting response... [('SSL routines', 'ssl3_get_record', 'wrong
ong version number')]
2020-03-31 18:04:58 ERROR 301: Moved Permanently
root@svr04:~# wget www.iltalehti.fi
--2020-03-31 18:05:32-- http://www.iltalehti.fi
Connecting to www.iltalehti.fi:80... connected.
HTTP request sent, awaiting response... [('SSL routines', 'ssl3_get_record', 'wrong
number')]
2020-03-31 18:05:32 ERROR 301: Moved Permanently
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=41.9 ms
64 bytes from google.com (29.89.32.244): icmp_seq=2 ttl=50 time=40.5 ms
64 bytes from google.com (29.89.32.244): icmp_seq=3 ttl=50 time=43.1 ms
64 bytes from google.com (29.89.32.244): icmp_seq=4 ttl=50 time=43.2 ms
^C
--- google.com ping statistics ---
4 packets transmitted, 4 received, 0% packet loss, time 907ms
rtt min/avg/max/mdev = 48.264/50.352/52.441/2.100 ms
root@svr04:~#
```

```

Changing the user information for teppo
Enter the new value, or press ENTER for the default
    Username []: teppo
    Full Name []: asd
    Room Number []: 23
    Work Phone []: 234
    Home Phone []: 234
    Mobile Phone []: 2345
    Country []: asdf
    City []: sdgrg
    Language []: asrg
    Favorite movie []: argasr
    Other []: asrgasrg
Is the information correct? [Y/n] Y
Ok, starting over

Changing the user information for teppo
Enter the new value, or press ENTER for the default
    Username []: adr
    Full Name []: █

```

it was an endless loop trying to create a user

- **Exploring logs**

Use VM's console and navigate to `$COWRIE_VAR/log/cowrie`. In this directory locates `cowrie.json` file.

```

[root@localhost.localdomain ~]# cd $COWRIE_VAR/log/cowrie
[root@localhost.localdomain cowrie]# ls
cowrie.json  cowrie.json.2020-03-04
[root@localhost.localdomain cowrie]# _

```

What information this log file contains?

information about the servers/users activity. wgets, commands, logins etc...

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.

```

{"eventId":"cowrie.login.failed","username":"root","password":"root66","message":"login attempt [root/root66] failed"},
{"eventId":"cowrie.login.failed","username":"admin","password":"admin","message":"login attempt [admin/admin] failed"},
{"eventId":"cowrie.login.failed","username":"test","password":"test","message":"login attempt [test/test] failed"},
{"eventId":"cowrie.login.success","username":"jamk","password":"tts0800","message":"login attempt [jamk/tts0800] succeeded"},

```

Which account was used to successful login on honeypot?

```
{"eventId":"cowrie.login.success","username":"jamk","password":"ttks0800","message":"login attempt [jamk/ttks0800] succeeded",
```

jamk/ttks0800

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",
```

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. 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 tty]# $COWRIE_BIN/playlog $COWRIE_BIN/e21e50dc9329ce84cad25751b123824ad17118efa068ccc6c789692cc21f3d85

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:~# touch M3426.txt
root@svr04:~# wget www.iltasanomat.fi
--2020-03-31 18:04:57-- http://www.iltasanomat.fi
Connecting to www.iltasanomat.fi:80... connected.
HTTP request sent, awaiting response... [({SSL routines', 'ssl3_get_record', 'wrong version number'})]
2020-03-31 18:04:58 ERROR 301: Moved Permanently
root@svr04:~# ^CTraceback (most recent call last):
  File "/opt/cowrie/bin/playlog", line 132, in <module>
    playlog(logfd, settings)
  File "/opt/cowrie/bin/playlog", line 61, in playlog
    time.sleep(sleeptime)
KeyboardInterrupt
[root@localhost.localdomain tty]#
```

shows a replay of my actions. Pretty cool

- **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 2020 is the actual SSH service. Find a way to do that. Hint: Redirect the traffic from port 22 to 2222 using iptables and change the actual SSH service to listen port 2020.

root@localhost:/etc/ssh

```
GNU nano 2.3.1 File: sshd_config

$OpenBSD: sshd_config,v 1.100 2016/08/15 12:32:04 naddy Exp $

# This is the sshd server system-wide configuration file.  See
# sshd_config(5) for more information.

# This sshd was compiled with PATH=/usr/local/bin:/usr/bin

# The strategy used for options in the default sshd_config shipped with
# OpenSSH is to specify options with their default value where
# possible, but leave them commented.  Uncommented options override the
# default value.

# If you want to change the port on a SELinux system, you have to tell
# SELinux about this change.
# semanage port -a -t ssh_port_t -p tcp #PORTNUMBER
#
Port 2020
```

```
[root@localhost.localdomain ssh]# systemctl restart sshd
[root@localhost.localdomain ssh]# systemctl status sshd
● sshd.service - OpenSSH server daemon
   Loaded: loaded (/usr/lib/systemd/system/sshd.service; enabled; vendor preset: enabled)
   Active: active (running) since Tue 2020-03-31 18:56:55 UTC; 7s ago
     Docs: man:sshd(8)
           man:sshd_config(5)
  Main PID: 2243 (sshd)
    Tasks: 1
   Memory: 1.0M
   CGroup: /system.slice/sshd.service
           └─2243 /usr/sbin/sshd -D

Mar 31 18:56:55 localhost.localdomain systemd[1]: Stopped OpenSSH server daemon.
Mar 31 18:56:55 localhost.localdomain systemd[1]: Starting OpenSSH server daemon...
Mar 31 18:56:55 localhost.localdomain sshd[2243]: Server listening on 0.0.0.0 port 2020.
Mar 31 18:56:55 localhost.localdomain sshd[2243]: Server listening on :: port 2020.
Mar 31 18:56:55 localhost.localdomain systemd[1]: Started OpenSSH server daemon.
[root@localhost.localdomain ssh]#
```

```
[root@localhost.localdomain ~]# iptables -t nat -A PREROUTING -p tcp --dport 22 -j REDIRECT --to-port 2222
[root@localhost.localdomain ~]#
```

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
# '!' at the start
# '/' can be used
#
root:x:!root
root:x:123456
root:x:!/honeypot/
jamk:x:ttks0800
tomcat:x:!tomcat
oracle:x:!oracle
M3426:x:M3426
```

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

```
[root@localhost.localdomain ~]# docker-compose -f /opt/docker-cowrie/docker-compose.yml restart cowrie
Restarting cowrie ... done
[root@localhost.localdomain ~]#
```

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

```
login as: M3426
M3426@192.168.43.131'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.
M3426@svr04:~$ ls -la
d-wxrw--wt 1 5508 5508 4096 2020-03-31 19:33 .
d-wxrw--wt 1 5508 5508 4096 2020-03-31 19:33 ..
M3426@svr04:~$ hostname
svr04
M3426@svr04:~$
```

```
({"eventid":"cowrie.login.success","username":"M3426","password":"M3426","message":"login attempt [M3426/M3426] succeeded","sens
{"eventid":"cowrie.client.size","width":80,"height":24,"message":"Terminal Size: 80 24","sensor":"5ac69e16f286","timestamp":"20
{"eventid":"cowrie.session.params","arch":"linux-x64-lsb","message":[],"sensor":"5ac69e16f286","timestamp":"2020-03-31T19:32:10
{"eventid":"cowrie.command.input","input":"ls -la","message":"CMD: ls -la","sensor":"5ac69e16f286","timestamp":"2020-03-31T19:3
{"eventid":"cowrie.command.input","input":"hostname","message":"CMD: hostname","sensor":"5ac69e16f286","timestamp":"2020-03-31T
{"eventid":"cowrie.session.connect","src_ip":"172.30.0.1","src_port":46314,"dst_ip":"172.30.0.2","dst_port":2222,"session":"2f1
{"eventid":"cowrie.client.version","version":"b'SSH-2.0-PuTTY_Release_0.70'", "message":"Remote SSH version: b'SSH-2.0-PuTTY_Rel
{"eventid":"cowrie.client.kex","hassh":"0425c279610910d2d9aef3b285e59b86","hasshAlgorithms":"curve25519-sha256@libssh.org,ecdh-
{"eventid":"cowrie.login.success","username":"M3426","password":"M3426","message":"login attempt [M3426/M3426] succeeded","sens
{"eventid":"cowrie.client.size","width":80,"height":24,"message":"Terminal Size: 80 24","sensor":"5ac69e16f286","timestamp":"20
{"eventid":"cowrie.session.params","arch":"linux-x64-lsb","message":[],"sensor":"5ac69e16f286","timestamp":"2020-03-31T19:33:00
{"eventid":"cowrie.command.input","input":"ls -la","message":"CMD: ls -la","sensor":"5ac69e16f286","timestamp":"2020-03-31T19:3
{"eventid":"cowrie.command.input","input":"hostname","message":"CMD: hostname","sensor":"5ac69e16f286","timestamp":"2020-03-31T
```


- When you create SSH connection to port 2222 and use credentials <your-student-id>/<your-student-id> for login, you end up inside honeypot

```
login as: M3426
M3426@192.168.43.131'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.
M3426@svr04:~$ ls -la
d-wxrw--wt 1 5508 5508 4096 2020-03-31 19:32 .
d-wxrw--wt 1 5508 5508 4096 2020-03-31 19:32 ..
M3426@svr04:~$ hostname
svr04
M3426@svr04:~$
```

```
{
  "eventid": "cowrie.login.success",
  "username": "M3426",
  "password": "M3426",
  "message": "login attempt [M3426/M3426] succeeded",
  "sensor": "5ac69e16f286",
  "timestamp": "2020-03-31T19:32:10"
},
{
  "eventid": "cowrie.session.params",
  "arch": "linux-x64-lsb",
  "message": [],
  "sensor": "5ac69e16f286",
  "timestamp": "2020-03-31T19:32:10"
},
{
  "eventid": "cowrie.command.input",
  "input": "ls -la",
  "message": "CMD: ls -la",
  "sensor": "5ac69e16f286",
  "timestamp": "2020-03-31T19:32:10"
},
{
  "eventid": "cowrie.command.input",
  "input": "hostname",
  "message": "CMD: hostname",
  "sensor": "5ac69e16f286",
  "timestamp": "2020-03-31T19:32:10"
},
{
  "eventid": "cowrie.session.connect",
  "src_ip": "172.30.0.1",
  "src_port": 46314,
  "dst_ip": "172.30.0.2",
  "dst_port": 2222,
  "session": "2f1"
},
{
  "eventid": "cowrie.client.version",
  "version": "b'SSH-2.0-PuTTY Release 0.70'",
  "message": "Remote SSH version: b'SSH-2.0-PuTTY Rel",
  "sensor": "5ac69e16f286",
  "timestamp": "2020-03-31T19:32:10"
},
{
  "eventid": "cowrie.client.kex",
  "hassh": "0425c279610910d2d9aef3b285e59b86",
  "hasshAlgorithms": "curve25519-sha256@libssh.org,ecdh-",
  "sensor": "5ac69e16f286",
  "timestamp": "2020-03-31T19:32:10"
},
{
  "eventid": "cowrie.login.success",
  "username": "M3426",
  "password": "M3426",
  "message": "login attempt [M3426/M3426] succeeded",
  "sensor": "5ac69e16f286",
  "timestamp": "2020-03-31T19:32:10"
},
{
  "eventid": "cowrie.client.size",
  "width": 80,
  "height": 24,
  "message": "Terminal Size: 80 24",
  "sensor": "5ac69e16f286",
  "timestamp": "2020-03-31T19:32:10"
},
{
  "eventid": "cowrie.session.params",
  "arch": "linux-x64-lsb",
  "message": [],
  "sensor": "5ac69e16f286",
  "timestamp": "2020-03-31T19:32:10"
},
{
  "eventid": "cowrie.command.input",
  "input": "ls -la",
  "message": "CMD: ls -la",
  "sensor": "5ac69e16f286",
  "timestamp": "2020-03-31T19:32:10"
},
{
  "eventid": "cowrie.command.input",
  "input": "hostname",
  "message": "CMD: hostname",
  "sensor": "5ac69e16f286",
  "timestamp": "2020-03-31T19:32:10"
}
```

- When you create SSH connection to port 2020 and use credentials root/root66 for login, you end up inside host as a real root user

```
login as: root
root@192.168.43.131's password:
Last login: Tue Mar 31 19:20:24 2020 from m3426
[root@localhost.localdomain ~]# ls -la
total 12608
dr-xr-x---.  6 root root    4096 Mar 31 18:00 .
dr-xr-xr-x. 17 root root    4096 Mar  4 19:10 ..
-rw-----.  1 root root    957 Oct  5  2015 anaconda-ks.cfg
-rw-----.  1 root root  15547 Mar  4 19:50 .bash_history
-rw-r--r--.  1 root root    18 Dec 29  2013 .bash_logout
-rw-r--r--.  1 root root   176 Dec 29  2013 .bash_profile
-rw-r--r--.  1 root root   362 Mar  4 19:39 .bashrc
drwxr-xr-x.  4 root root    27 Mar  2 13:35 .cache
drwxr-xr-x.  3 root root    17 Aug 11  2016 .config
-rw-r--r--.  1 root root   100 Dec 29  2013 .cshrc
-rw-r--r--.  1 root root 202296 Mar 31 18:00 index.html
-rw-----.  1 root root    45 Nov 21  2018 .lessht
-rw-----.  1 root root   197 Nov 21  2018 .mysql_history
drwxr-----. 3 root root    18 Mar  2 13:30 .pki
drwx-----. 2 root root     6 Mar  4 19:44 .ssh
-rw-r--r--.  1 root root 12637098 Nov 22  2018 target-server.tar.gz
-rw-r--r--.  1 root root    129 Dec 29  2013 .tcshrc
-rw-----.  1 root root   6869 Mar  4 19:41 .viminfo
[root@localhost.localdomain ~]# hostname
localhost.localdomain
[root@localhost.localdomain ~]#
```