

FCS REPORT

Yashasvi Chaurasia
2020159

Question 4

a)

Steps:

1. I installed 'knockd' on my system
2. Configured the knockd.conf file to change the default values of the port knocking sequence.
3. Added entries for SSH port on port 22 and FTP port on port 21
4. Used 'ip addr' to check the network interface I would be using knocking on.
In my case it is 'wlan0'
5. Then I edited the knockd control file to enable knocking and modified the file to write my network interface in the file.
6. Then I used 'service knockd status' to check status of the knockd service and then I used 'service knockd start' to enable the knockd service.
7. Later I used my another machine to knock and then connect on the ssh port
8. In the end I used knock to close the ssh port .

Reasoning of iptable rule :

I used

```
"sbin/iptables -I INPUT -s %IP% -p tcp --dport 22 -j ACCEPT"
```

As using the -i flag I can insert the rule at the top of the table rather than appending the rule at the bottom as if any rule above the new connection rule closes the ssh port then the newly made ssh connection cannot execute due to the priority in which the rules are executed.

Hence if I want to open or close a port after knocking then I should insert it at the top of the table rather than appending it to the bottom to avoid any unintended behaviour.

Screenshots of the process:

```

(octops@kali)-[/etc]
$ cat knockd.conf
[options]
    logfile = /var/log/knockd.log

[openSSH]
    sequence      = 5460,1500,9800
    seq_timeout   = 10
    command       = /sbin/iptables -I INPUT -s %IP% -p tcp --dport 22 -j ACCEPT
    tcpflags      = syn

[closeSSH]
    sequence      = 9200,8010,7320
    seq_timeout   = 10
    command       = /sbin/iptables -D INPUT -s %IP% -p tcp --dport 22 -j ACCEPT
    tcpflags      = syn

[openFTP]
    sequence      = 5200,1100,9000
    seq_timeout   = 10
    command       = /sbin/iptables -I INPUT -s %IP% -p tcp --dport 21 -j ACCEPT
    tcpflags      = syn

[closeFTP]
    sequence      = 9000,8010,7020
    seq_timeout   = 10
    command       = /sbin/iptables -D INPUT -s %IP% -p tcp --dport 21 -j ACCEPT
    tcpflags      = syn

(octops@kali)-[/etc]
$

```

Modified conf file

```

(octops@kali)-[/etc]
$ ip addr
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN group default qlen 1000
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
    inet 127.0.0.1/8 scope host lo
        valid_lft forever preferred_lft forever
    inet6 ::1/128 scope host
        valid_lft forever preferred_lft forever
2: eth0: <NO-CARRIER,BROADCAST,MULTICAST,UP> mtu 1500 qdisc fq_codel state DOWN group default qlen 1000
    link/ether 20:47:47:36:5b:39 brd ff:ff:ff:ff:ff:ff
3: wlan0: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc noqueue state UP group default qlen 1000
    link/ether b0:c0:90:1c:4f:af brd ff:ff:ff:ff:ff:ff
    inet 192.168.59.58/20 brd 192.168.63.255 scope global dynamic noprefixroute wlan0
        valid_lft 2511sec preferred_lft 2511sec
    inet6 fe80::b2c0:90ff:fe1c:4faf/64 scope link noprefixroute
        valid_lft forever preferred_lft forever

(octops@kali)-[/etc]

```

Checking network interface

```
octops@kali: /etc/default
# control if we start knockd at init or not
# 1 = start
# anything else = don't start
# PLEASE EDIT /etc/knockd.conf BEFORE ENABLING
START_KNOCKD=1

# command line options
KNOCKD_OPTS="-i wlan0"
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
-- INSERT --
```

Editing knock control file

```
octops@kali: ~
The programs included with the Kali GNU/Linux system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*/copyright.

Kali GNU/Linux comes with ABSOLUTELY NO WARRANTY, to the extent
permitted by applicable law.
Last login: Sat Oct 15 06:34:22 2022 from 192.168.56.59
(octops@kali)-[~]
$ sudo systemctl start knockd
[sudo] password for octops:
sudo: systemctl: command not found

(octops@kali)-[~]
$ service knockd status
o knockd.service - Port-Knock Daemon
   Loaded: loaded (/lib/systemd/system/knockd.service; disabled; vendor prese>
   Active: inactive (dead)
   Docs: man:knockd(1)

(octops@kali)-[~]
$ service knockd start
=== AUTHENTICATING FOR org.freedesktop.systemd1.manage-units ===
Authentication is required to start 'knockd.service'.
Authenticating as: Octops,, (octops)
Password: 
```

Enabling service

```
octops@kali: ~  
$ service knockd status  
○ knockd.service - Port-Knock Daemon  
   Loaded: loaded (/lib/systemd/system/knockd.service; disabled; vendor prese>  
   Active: inactive (dead)  
   Docs: man:knockd(1)  
  
(octops@kali)-[~]  
$ service knockd start  
==== AUTHENTICATING FOR org.freedesktop.systemd1.manage-units ====  
Authentication is required to start 'knockd.service'.  
Authenticating as: Octops,, (octops)  
Password: Failed to start knockd.service: Connection timed out  
See system logs and 'systemctl status knockd.service' for details.  
polkit-agent-helper-1: pam_authenticate failed: Authentication failure  
  
(octops@kali)-[~]  
$ service knockd start  
==== AUTHENTICATING FOR org.freedesktop.systemd1.manage-units ====  
Authentication is required to start 'knockd.service'.  
Authenticating as: Octops,, (octops)  
Password:  
==== AUTHENTICATION COMPLETE ====  
  
(octops@kali)-[~]  
$
```

Enabled knockd service successfully

```
● knockd.service - Port-Knock Daemon  
   Loaded: loaded (/lib/systemd/system/knockd.service; disabled; vendor preset: disabled)  
   Active: active (running) since Sat 2022-10-15 06:45:51 PDT; 23min ago  
     Docs: man:knockd(1)  
  Main PID: 16040 (knockd)  
    Tasks: 1 (limit: 8166)  
  Memory: 688.0K  
     CPU: 62ms  
   CGroup: /system.slice/knockd.service  
           └─16040 /usr/sbin/knockd -i wlan0  
  
Oct 15 06:59:09 kali knockd[16040]: 192.168.2.250: openSSH: Stage 1  
Oct 15 06:59:10 kali knockd[16040]: 192.168.2.250: openSSH: Stage 2  
Oct 15 06:59:10 kali knockd[16040]: 192.168.2.250: openSSH: Stage 3  
Oct 15 06:59:10 kali knockd[16040]: 192.168.2.250: openSSH: OPEN SESAME  
Oct 15 06:59:10 kali knockd[19417]: openSSH: running command: /sbin/iptables -I INPUT -s 192.168.2.250 -p tcp >  
Oct 15 06:59:16 kali knockd[16040]: 192.168.2.250: closeSSH: Stage 1  
Oct 15 06:59:16 kali knockd[16040]: 192.168.2.250: closeSSH: Stage 2  
Oct 15 06:59:17 kali knockd[16040]: 192.168.2.250: closeSSH: Stage 3  
Oct 15 06:59:17 kali knockd[16040]: 192.168.2.250: closeSSH: OPEN SESAME  
Oct 15 06:59:17 kali knockd[19448]: closeSSH: running command: /sbin/iptables -D INPUT -s 192.168.2.250 -p tcp>  
~  
~  
~  
~  
~  
~  
~
```

Log file for knocking process and connections

b)TCP prevents anyone from sending loads of request at multiple ports very quickly to avoid congestion hence it prevents attackers from brute forcing the knocking sequence but the udp protocol provides no such limitation and hence it can be used to brute force the knocking sequence by sending many requests at multiple ports at once.

TCP protocol also ensures security of the connection and all the communications done over tcp once the connection is established can be ensured to be more secure than the UDP protocol as UDP protocol is the bare bones algorithm which provides no such security and confidentiality guarantee of the communication.

c)

The default choice of ports of knockd is not safe as these ports are known to everyone and the entire reason for knocking a port would fail if anyone can access the ports after knowing the correct sequence.

The reason to knock before allowing a certain ports is to prevent people without the correct sequence from using the ports.

If the port sequence details is leaked or left unchanged to the default value then anyone can access the ports and cause harm and hence there would be a huge threat if the knocking sequence is left at the default port sequence.