Ujjwal Shah

Firewall and Intrusion Detection Systems
Project

This is an individual project, please work out your own solutions and results.

Pre-work: Use the virtual network environment we have setup and make sure to change the VM host names to [YourName]-Attacker, [YourName]-Router, and [YourName]-Victim. **Your grade will be ZERO if the screenshots of the terminal commands don't show your names.** Show screenshots for each of the following tasks.

1. The Router's INPUT table only accepts traffic from the three subnets: 192.168.0.0/24 , 10.0.3.0/24, and 10.0.2.0/24 on the corresponding interfaces. Any spoofed traffic should be dropped. Use a good example and a counter example to show the rules work.

   The rules :

   The good example:

```
┌──(root💀 ujjwal-victim)-[/home/kali]
└─# ping 192.168.0.200
PING 192.168.0.200 (192.168.0.200) 56(84) bytes of data.
64 bytes from 192.168.0.200: icmp_seq=1 ttl=64 time=0.399 ms
64 bytes from 192.168.0.200: icmp_seq=2 ttl=64 time=0.541 ms
64 bytes from 192.168.0.200: icmp_seq=3 ttl=64 time=0.516 ms
64 bytes from 192.168.0.200: icmp_seq=4 ttl=64 time=0.369 ms
64 bytes from 192.168.0.200: icmp_seq=5 ttl=64 time=0.434 ms
64 bytes from 192.168.0.200: icmp_seq=6 ttl=64 time=0.518 ms
^C
--- 192.168.0.200 ping statistics ---
6 packets transmitted, 6 received, 0% packet loss, time 5221ms
rtt min/avg/max/mdev = 0.369/0.462/0.541/0.065 ms
```

```
┌──(root💀 ujjwal-attacker)-[/home/kali]
└─# ping 10.0.3.20
PING 10.0.3.20 (10.0.3.20) 56(84) bytes of data.
64 bytes from 10.0.3.20: icmp_seq=1 ttl=64 time=0.449 ms
64 bytes from 10.0.3.20: icmp_seq=2 ttl=64 time=0.475 ms
64 bytes from 10.0.3.20: icmp_seq=3 ttl=64 time=0.465 ms
64 bytes from 10.0.3.20: icmp_seq=4 ttl=64 time=0.460 ms
64 bytes from 10.0.3.20: icmp_seq=5 ttl=64 time=0.522 ms
^C
--- 10.0.3.20 ping statistics ---
5 packets transmitted, 5 received, 0% packet loss, time 4100ms
rtt min/avg/max/mdev = 0.449/0.474/0.522/0.025 ms
```

```
┌──(root💀 ujjwal-router)-[/home/kali]
└─# tcpdump -i eth1
tcpdump: verbose output suppressed, use -v or -vv for full protocol decode
listening on eth1, link-type EN10MB (Ethernet), capture size 262144 bytes
09:26:41.126096 IP 10.0.3.10 > 10.0.3.20: ICMP echo request, id 11646, seq 1, length 64
09:26:41.126165 IP 10.0.3.20 > 10.0.3.10: ICMP echo reply, id 11646, seq 1, length 64
09:26:42.153227 IP 10.0.3.10 > 10.0.3.20: ICMP echo request, id 11646, seq 2, length 64
09:26:42.153252 IP 10.0.3.20 > 10.0.3.10: ICMP echo reply, id 11646, seq 2, length 64
09:26:43.178150 IP 10.0.3.10 > 10.0.3.20: ICMP echo request, id 11646, seq 3, length 64
09:26:43.178177 IP 10.0.3.20 > 10.0.3.10: ICMP echo reply, id 11646, seq 3, length 64
09:26:44.201232 IP 10.0.3.10 > 10.0.3.20: ICMP echo request, id 11646, seq 4, length 64
09:26:44.201256 IP 10.0.3.20 > 10.0.3.10: ICMP echo reply, id 11646, seq 4, length 64
09:26:45.225645 IP 10.0.3.10 > 10.0.3.20: ICMP echo request, id 11646, seq 5, length 64
09:26:45.225679 IP 10.0.3.20 > 10.0.3.10: ICMP echo reply, id 11646, seq 5, length 64
09:26:46.217513 ARP, Request who-has 10.0.3.20 tell 10.0.3.10, length 46
09:26:46.217527 ARP, Reply 10.0.3.20 is-at 08:00:27:f9:56:c0 (oui Unknown), length 28
09:26:46.281033 ARP, Request who-has 10.0.3.10 tell 10.0.3.20, length 28
09:26:46.281364 ARP, Reply 10.0.3.10 is-at 08:00:27:a3:53:2a (oui Unknown), length 46
^C
14 packets captured
14 packets received by filter
0 packets dropped by kernel
```

Counter Example:

```
┌──(root💀 ujjwal-router)-[/home/kali]
└─# tcpdump -i eth1
tcpdump: verbose output suppressed, use -v or -vv for full protocol decode
listening on eth1, link-type EN10MB (Ethernet), capture size 262144 bytes
^C09:36:34.833586 IP 96.212.90.255.2084 > 10.0.3.20.0: Flags [S], seq 337226137, win 512, length 0

1 packet captured
219835 packets received by filter
219803 packets dropped by kernel
```

```
┌──(root💀 ujjwal-attacker)-[/home/kali]
└─# hping3 --rand-source -S --flood -V 10.0.3.20
using eth1, addr: 10.0.3.10, MTU: 1500
HPING 10.0.3.20 (eth1 10.0.3.20): S set, 40 headers + 0 data bytes
hping in flood mode, no replies will be shown
^C
--- 10.0.3.20 hping statistic ---
219832 packets transmitted, 0 packets received, 100% packet loss
round-trip min/avg/max = 0.0/0.0/0.0 ms
```

2. The Router's INPUT table only accepts two types of traffic: SSH and ping (echo-request), other types of traffic should be dropped. Use a good example and a counter example to show the rules works.

Rules:

Ujjwal Shah

```
┌──(root💀 ujjwal-router)-[/home/kali]
└─# iptables -A INPUT -j ACCEPT -p TCP --destination-port 22

┌──(root💀 ujjwal-router)-[/home/kali]
└─# iptables -A INPUT -j ACCEPT -p ICMP

┌──(root💀 ujjwal-router)-[/home/kali]
└─# iptables -P INPUT DROP

┌──(root💀 ujjwal-router)-[/home/kali]
└─# iptables -L
Chain INPUT (policy DROP)
target     prot opt source               destination
ACCEPT     tcp  --  anywhere             anywhere             tcp dpt:ssh
ACCEPT     icmp --  anywhere             anywhere             icmp echo-request

Chain FORWARD (policy ACCEPT)
target     prot opt source               destination

Chain OUTPUT (policy ACCEPT)
target     prot opt source               destination
```

Examples:

```
┌──(root💀 ujjwal-victim)-[/home/kali]
└─# ssh kali@192.168.0.200
kali@192.168.0.200's password:
Linux ujjwal-router 5.9.0-kali1-amd64 #1 SMP Debian 5.9.1-1kali2 (2020-10-29) x86_64

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 Apr 24 09:25:30 2021 from 192.168.0.100
┌(Message from Kali developers)

  We have kept /usr/bin/python pointing to Python 2 for backwards
  compatibility. Learn how to change this and avoid this message:
  ⇒ https://www.kali.org/docs/general-use/python3-transition/

└(Run "touch ~/.hushlogin" to hide this message)
┌──(kali㉿ ujjwal-router)-[~]
└─$
```

```
┌──(root㉿ ujjwal-router)-[/home/kali]
└─# tcpdump -i eth1
tcpdump: verbose output suppressed, use -v or -vv for full protocol decode
listening on eth1, link-type EN10MB (Ethernet), capture size 262144 bytes
15:29:58.895387 IP 10.0.3.10 > 10.0.3.20: ICMP echo request, id 49310, seq 1, length 64
15:29:58.895419 IP 10.0.3.20 > 10.0.3.10: ICMP echo reply, id 49310, seq 1, length 64
15:29:59.913894 IP 10.0.3.10 > 10.0.3.20: ICMP echo request, id 49310, seq 2, length 64
15:29:59.913919 IP 10.0.3.20 > 10.0.3.10: ICMP echo reply, id 49310, seq 2, length 64
15:30:00.940190 IP 10.0.3.10 > 10.0.3.20: ICMP echo request, id 49310, seq 3, length 64
15:30:00.940215 IP 10.0.3.20 > 10.0.3.10: ICMP echo reply, id 49310, seq 3, length 64
15:30:01.978932 IP 10.0.3.10 > 10.0.3.20: ICMP echo request, id 49310, seq 4, length 64
15:30:01.978957 IP 10.0.3.20 > 10.0.3.10: ICMP echo reply, id 49310, seq 4, length 64
15:30:03.012228 IP 10.0.3.10 > 10.0.3.20: ICMP echo request, id 49310, seq 5, length 64
15:30:03.012265 IP 10.0.3.20 > 10.0.3.10: ICMP echo reply, id 49310, seq 5, length 64
15:30:04.038903 IP 10.0.3.10 > 10.0.3.20: ICMP echo request, id 49310, seq 6, length 64
15:30:04.038927 IP 10.0.3.20 > 10.0.3.10: ICMP echo reply, id 49310, seq 6, length 64
15:30:04.068557 ARP, Request who-has 10.0.3.20 tell 10.0.3.10, length 46
15:30:04.068571 ARP, Reply 10.0.3.20 is-at 08:00:27:f9:56:c0 (oui Unknown), length 28
15:30:04.099957 ARP, Request who-has 10.0.3.10 tell 10.0.3.20, length 28
15:30:04.100381 ARP, Reply 10.0.3.10 is-at 08:00:27:a3:53:2a (oui Unknown), length 46
^C
16 packets captured
16 packets received by filter
0 packets dropped by kernel
```

```
┌──(root㉿ ujjwal-router)-[/home/kali]
└─# tcpdump -i eth2
tcpdump: verbose output suppressed, use -v or -vv for full protocol decode
listening on eth2, link-type EN10MB (Ethernet), capture size 262144 bytes
15:33:35.274151 IP 192.168.0.100 > 192.168.0.200: ICMP echo request, id 25367, seq 1, length 64
15:33:35.274204 IP 192.168.0.200 > 192.168.0.100: ICMP echo reply, id 25367, seq 1, length 64
15:33:36.372581 IP 192.168.0.100 > 192.168.0.200: ICMP echo request, id 25367, seq 2, length 64
15:33:36.372609 IP 192.168.0.200 > 192.168.0.100: ICMP echo reply, id 25367, seq 2, length 64
15:33:37.378642 IP 192.168.0.100 > 192.168.0.200: ICMP echo request, id 25367, seq 3, length 64
15:33:37.378716 IP 192.168.0.200 > 192.168.0.100: ICMP echo reply, id 25367, seq 3, length 64
15:33:38.467399 IP 192.168.0.100 > 192.168.0.200: ICMP echo request, id 25367, seq 4, length 64
15:33:38.467423 IP 192.168.0.200 > 192.168.0.100: ICMP echo reply, id 25367, seq 4, length 64
15:33:40.380085 ARP, Request who-has 192.168.0.200 tell 192.168.0.100, length 46
15:33:40.380097 ARP, Reply 192.168.0.200 is-at 08:00:27:f1:3c:2a (oui Unknown), length 28
15:33:40.426793 ARP, Request who-has 192.168.0.100 tell 192.168.0.200, length 28
15:33:40.427251 ARP, Reply 192.168.0.100 is-at 08:00:27:2d:b9:2a (oui Unknown), length 46
^C
12 packets captured
12 packets received by filter
0 packets dropped by kernel
```

```
┌──(root㉿ ujjwal-router)-[/home/kali]
└─# service apache2 status
● apache2.service - The Apache HTTP Server
     Loaded: loaded (/lib/systemd/system/apache2.service; disabled; vendor preset: disabled)
     Active: active (running) since Sat 2021-04-24 15:37:12 EDT; 37s ago
       Docs: https://httpd.apache.org/docs/2.4/
    Process: 1565 ExecStart=/usr/sbin/apachectl start (code=exited, status=0/SUCCESS)
   Main PID: 1576 (apache2)
      Tasks: 6 (limit: 2308)
     Memory: 18.0M
     CGroup: /system.slice/apache2.service
             ├─1576 /usr/sbin/apache2 -k start
             ├─1578 /usr/sbin/apache2 -k start
             ├─1579 /usr/sbin/apache2 -k start
             ├─1580 /usr/sbin/apache2 -k start
             ├─1581 /usr/sbin/apache2 -k start
             └─1582 /usr/sbin/apache2 -k start
```

```
┌──(root㉿ ujjwal-victim)-[/home/kali]
└─# nmap 192.168.0.200
Starting Nmap 7.91 ( https://nmap.org ) at 2021-04-24 15:38 EDT
Nmap scan report for 192.168.0.200
Host is up (0.00042s latency).
Not shown: 999 filtered ports
PORT   STATE SERVICE
22/tcp open  ssh
MAC Address: 08:00:27:F1:3C:2A (Oracle VirtualBox virtual NIC)
```

3. The Router's OUTPUT table should set state tracking rules that drop all INVALID traffic and always accept ESTABLISHED and RELATED traffic.

   Rules:

```
┌──(root💀 ujjwal-router)-[/home/kali]
└─# iptables -A OUTPUT -m conntrack --ctstate INVALID -j DROP

┌──(root💀 ujjwal-router)-[/home/kali]
└─# iptables -A OUTPUT -m conntrack --ctstate ESTABLISHED,RELATED -j ACCEPT

┌──(root💀 ujjwal-router)-[/home/kali]
└─# iptables -L
Chain INPUT (policy ACCEPT)
target     prot opt source               destination

Chain FORWARD (policy ACCEPT)
target     prot opt source               destination

Chain OUTPUT (policy ACCEPT)
target     prot opt source               destination
DROP       all  --  anywhere             anywhere             ctstate INVALID
ACCEPT     all  --  anywhere             anywhere             ctstate RELATED,ESTABLISHED
```

Example:

```
┌──(root💀 ujjwal-router)-[/var/log]
└─# ping 10.0.3.10
PING 10.0.3.10 (10.0.3.10) 56(84) bytes of data.
64 bytes from 10.0.3.10: icmp_seq=1 ttl=64 time=0.453 ms
64 bytes from 10.0.3.10: icmp_seq=2 ttl=64 time=0.507 ms
64 bytes from 10.0.3.10: icmp_seq=3 ttl=64 time=0.514 ms
64 bytes from 10.0.3.10: icmp_seq=4 ttl=64 time=0.497 ms
64 bytes from 10.0.3.10: icmp_seq=5 ttl=64 time=0.466 ms
64 bytes from 10.0.3.10: icmp_seq=6 ttl=64 time=0.432 ms
64 bytes from 10.0.3.10: icmp_seq=7 ttl=64 time=0.845 ms
64 bytes from 10.0.3.10: icmp_seq=8 ttl=64 time=0.462 ms
64 bytes from 10.0.3.10: icmp_seq=9 ttl=64 time=0.469 ms
64 bytes from 10.0.3.10: icmp_seq=10 ttl=64 time=0.399 ms
64 bytes from 10.0.3.10: icmp_seq=11 ttl=64 time=0.437 ms
64 bytes from 10.0.3.10: icmp_seq=12 ttl=64 time=0.433 ms
64 bytes from 10.0.3.10: icmp_seq=13 ttl=64 time=0.417 ms
64 bytes from 10.0.3.10: icmp_seq=14 ttl=64 time=0.534 ms
^C
--- 10.0.3.10 ping statistics ---
14 packets transmitted, 14 received, 0% packet loss, time 13302ms
rtt min/avg/max/mdev = 0.399/0.490/0.845/0.105 ms

┌──(root💀 ujjwal-router)-[/etc/apache2]
└─# conntrack -L
icmp     1 29 src=10.0.3.20 dst=10.0.3.10 type=8 code=0 id=43648 src=10.0.3.10 dst=10.0.3.20 type=0 code=0 id=43648 mark=0 use=1
```

4. The Router's OUTPUT table should only allow outgoing NEW traffic for the following services: a. SSH (22)-TCP, b. HTTP (80)-TCP, c. HTTPS(443)-TCP, d. DNS (53)-UDP, and e. ping (echo-request) messages. Show an example.

Rules:

```
┌──(root💀 ujjwal-router)-[/home/kali]
└─# iptables -A OUTPUT -p tcp --dport 22 --syn -m conntrack --ctstate NEW -j ACCEPT

┌──(root💀 ujjwal-router)-[/home/kali]
└─# iptables -A OUTPUT -p tcp --dport 80 --syn -m conntrack --ctstate NEW -j ACCEPT

┌──(root💀 ujjwal-router)-[/home/kali]
└─# iptables -A OUTPUT -p tcp --dport 443 --syn -m conntrack --ctstate NEW -j ACCEPT

┌──(root💀 ujjwal-router)-[/home/kali]
└─# iptables -A OUTPUT -p udp --dport 53 --syn -m conntrack --ctstate NEW -j ACCEPT
iptables v1.8.5 (nf_tables): unknown option "--syn"
Try `iptables -h' or 'iptables --help' for more information.

┌──(root💀 ujjwal-router)-[/home/kali]
└─# iptables -A OUTPUT -p udp --dport 53 -m conntrack --ctstate NEW -j ACCEPT

┌──(root💀 ujjwal-router)-[/home/kali]
└─# iptables -A OUTPUT -p icmp --icmp-type echo-request -j ACCEPT
```

Ujjwal Shah

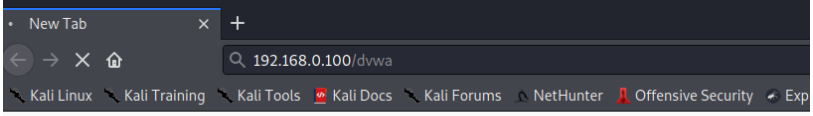
```
┌──(root💀ujjwal-router)-[/home/kali]
└─# iptables -P OUTPUT DROP

┌──(root💀ujjwal-router)-[/home/kali]
└─# iptables -L
Chain INPUT (policy ACCEPT)
target     prot opt source               destination

Chain FORWARD (policy ACCEPT)
target     prot opt source               destination

Chain OUTPUT (policy DROP)
target     prot opt source               destination
ACCEPT     tcp  --  anywhere             anywhere             tcp dpt:ssh flags:FIN,SYN,RST,ACK/SYN ctstate NEW
ACCEPT     tcp  --  anywhere             anywhere             tcp dpt:http flags:FIN,SYN,RST,ACK/SYN ctstate NEW
ACCEPT     tcp  --  anywhere             anywhere             tcp dpt:https flags:FIN,SYN,RST,ACK/SYN ctstate NEW
ACCEPT     udp  --  anywhere             anywhere             udp dpt:domain ctstate NEW
ACCEPT     icmp --  anywhere             anywhere             icmp echo-request
```

Examples:

```
┌──(root💀ujjwal-router)-[/home/kali]
└─# conntrack -L
icmp     1 22 src=10.0.3.20 dst=10.0.3.10 type=8 code=0 id=40409 src=10.0.3.10 dst=10.0.3.20 type=0 code=0 id=40409 mark=0 use=1
icmp     1 19 src=192.168.0.100 dst=192.168.0.200 type=8 code=0 id=24641 src=192.168.0.200 dst=192.168.0.100 type=0 code=0 id=24641 mark=0 use=1
conntrack v1.4.6 (conntrack-tools): 2 flow entries have been shown.

┌──(root💀ujjwal-router)-[/home/kali]
└─# tcpdump -i eth2
tcpdump: verbose output suppressed, use -v or -vv for full protocol decode
listening on eth2, link-type EN10MB (Ethernet), capture size 262144 bytes
12:44:28.279749 IP 192.168.0.100 > 192.168.0.200: ICMP echo request, id 62039, seq 1, length 64
12:44:29.300839 IP 192.168.0.100 > 192.168.0.200: ICMP echo request, id 62039, seq 2, length 64
12:44:30.408557 IP 192.168.0.100 > 192.168.0.200: ICMP echo request, id 62039, seq 3, length 64
12:44:31.495839 IP 192.168.0.100 > 192.168.0.200: ICMP echo request, id 62039, seq 4, length 64
12:44:32.518302 IP 192.168.0.100 > 192.168.0.200: ICMP echo request, id 62039, seq 5, length 64
12:44:33.349623 ARP, Request who-has 192.168.0.200 tell 192.168.0.100, length 46
12:44:33.349645 ARP, Reply 192.168.0.200 is-at 08:00:27:f1:3c:2a (oui Unknown), length 28
12:44:33.541469 IP 192.168.0.100 > 192.168.0.200: ICMP echo request, id 62039, seq 6, length 64
^C
8 packets captured
8 packets received by filter
0 packets dropped by kernel
```

```
┌──(kali㉿ujjwal-victim)-[~]
└─$ ping 192.168.0.200
PING 192.168.0.200 (192.168.0.200) 56(84) bytes of data.
^C
--- 192.168.0.200 ping statistics ---
6 packets transmitted, 0 received, 100% packet loss, time 5262ms
```

```
┌──(root💀ujjwal-router)-[/home/kali]
└─# ping 10.0.3.10
PING 10.0.3.10 (10.0.3.10) 56(84) bytes of data.
64 bytes from 10.0.3.10: icmp_seq=1 ttl=64 time=0.802 ms
64 bytes from 10.0.3.10: icmp_seq=2 ttl=64 time=0.344 ms
64 bytes from 10.0.3.10: icmp_seq=3 ttl=64 time=0.424 ms
64 bytes from 10.0.3.10: icmp_seq=4 ttl=64 time=0.360 ms
64 bytes from 10.0.3.10: icmp_seq=5 ttl=64 time=0.464 ms
```



```
┌──(root💀 ujjwal-router)-[/home/kali]
└─# conntrack -L
tcp      6 1 LAST_ACK src=192.168.0.200 dst=192.168.0.100 sport=37540 dport=22 src=192.168.0.100 dst=192.168.0.200 sport=22 dport=37540 [ASSURED] mark=0 use=1
conntrack v1.4.6 (conntrack-tools): 1 flow entries have been shown.

┌──(root💀 ujjwal-router)-[/home/kali]
└─# conntrack -L
conntrack v1.4.6 (conntrack-tools): 0 flow entries have been shown.

┌──(root💀 ujjwal-router)-[/home/kali]
└─# conntrack -L
tcp      6 27 LAST_ACK src=10.0.2.15 dst=13.225.210.18 sport=38168 dport=443 src=13.225.210.18 dst=10.0.2.15 sport=443 dport=38168 [ASSURED] mark=0 use=1
udp      17 16 src=10.0.2.15 dst=192.168.1.1 sport=50276 dport=53 src=192.168.1.1 dst=10.0.2.15 sport=53 dport=50276 mark=0 use=1
tcp      6 26 LAST_ACK src=10.0.2.15 dst=35.244.181.201 sport=53322 dport=443 src=35.244.181.201 dst=10.0.2.15 sport=443 dport=53322 [ASSURED] mark=0 use=1
tcp      6 298 ESTABLISHED src=192.168.0.200 dst=192.168.0.100 sport=58262 dport=80 src=192.168.0.100 dst=192.168.0.200 sport=80 dport=58262 [ASSURED] mark=0 use=1
tcp      6 117 TIME_WAIT src=10.0.2.15 dst=34.216.80.151 sport=38952 dport=443 src=34.216.80.151 dst=10.0.2.15 sport=443 dport=38952 [ASSURED] mark=0 use=1
tcp      6 299 ESTABLISHED src=10.0.2.15 dst=52.35.236.53 sport=57476 dport=443 src=52.35.236.53 dst=10.0.2.15 sport=443 dport=57476 [ASSURED] mark=0 use=1
tcp      6 117 TIME_WAIT src=10.0.2.15 dst=52.10.171.58 sport=53642 dport=443 src=52.10.171.58 dst=10.0.2.15 sport=443 dport=53642 [ASSURED] mark=0 use=1
tcp      6 109 TIME_WAIT src=10.0.2.15 dst=52.35.236.53 sport=57464 dport=443 src=52.35.236.53 dst=10.0.2.15 sport=443 dport=57464 [ASSURED] mark=0 use=1
tcp      6 293 ESTABLISHED src=10.0.2.15 dst=13.225.210.18 sport=38196 dport=443 src=13.225.210.18 dst=10.0.2.15 sport=443 dport=38196 [ASSURED] mark=0 use=1
tcp      6 27 LAST_ACK src=10.0.2.15 dst=13.225.210.18 sport=38186 dport=443 src=13.225.210.18 dst=10.0.2.15 sport=443 dport=38186 [ASSURED] mark=0 use=1
tcp      6 105 FIN_WAIT src=192.168.0.200 dst=192.168.0.100 sport=58252 dport=80 src=192.168.0.100 dst=192.168.0.200 sport=80 dport=58252 [ASSURED] mark=0 use=1
tcp      6 27 LAST_ACK src=10.0.2.15 dst=13.225.210.18 sport=38170 dport=443 src=13.225.210.18 dst=10.0.2.15 sport=443 dport=38170 [ASSURED] mark=0 use=1
tcp      6 293 ESTABLISHED src=10.0.2.15 dst=13.225.210.18 sport=38198 dport=443 src=13.225.210.18 dst=10.0.2.15 sport=443 dport=38198 [ASSURED] mark=0 use=1
tcp      6 119 TIME_WAIT src=10.0.2.15 dst=172.217.9.234 sport=58584 dport=443 src=172.217.9.234 dst=10.0.2.15 sport=443 dport=58584 [ASSURED] mark=0 use=1
tcp      6 27 LAST_ACK src=10.0.2.15 dst=13.225.210.18 sport=38172 dport=443 src=13.225.210.18 dst=10.0.2.15 sport=443 dport=38172 [ASSURED] mark=0 use=1
tcp      6 28 LAST_ACK src=10.0.2.15 dst=54.192.100.61 sport=36114 dport=443 src=54.192.100.61 dst=10.0.2.15 sport=443 dport=36114 [ASSURED] mark=0 use=1
conntrack v1.4.6 (conntrack-tools): 16 flow entries have been shown.
```

```
┌──(kali㉿ ujjwal-victim)-[~]
└─$ ssh kali@192.169.0.200
^C

┌──(kali㉿ ujjwal-victim)-[~]
└─$ ssh kali@10.0.3.20
^C
```

```
┌──(root💀 ujjwal-router)-[/home/kali]
└─# conntrack -L
tcp      6 112 TIME_WAIT src=10.0.2.15 dst=13.225.210.18 sport=38168 dport=443 src=13.225.210.18 dst=10.0.2.15 sport=443 dport=38168 [ASSURED] mark=0 use=1
tcp      6 111 TIME_WAIT src=10.0.2.15 dst=35.244.181.201 sport=53322 dport=443 src=35.244.181.201 dst=10.0.2.15 sport=443 dport=53322 [ASSURED] mark=0 use=1
tcp      6 110 TIME_WAIT src=10.0.2.15 dst=34.216.80.151 sport=38952 dport=443 src=34.216.80.151 dst=10.0.2.15 sport=443 dport=38952 [ASSURED] mark=0 use=1
tcp      6 110 TIME_WAIT src=10.0.2.15 dst=52.10.171.58 sport=53642 dport=443 src=52.10.171.58 dst=10.0.2.15 sport=443 dport=53642 [ASSURED] mark=0 use=1
tcp      6 110 TIME_WAIT src=10.0.2.15 dst=13.225.210.18 sport=38196 dport=443 src=13.225.210.18 dst=10.0.2.15 sport=443 dport=38196 [ASSURED] mark=0 use=1
tcp      6 111 TIME_WAIT src=10.0.2.15 dst=13.225.210.18 sport=38186 dport=443 src=13.225.210.18 dst=10.0.2.15 sport=443 dport=38186 [ASSURED] mark=0 use=1
tcp      6 112 TIME_WAIT src=10.0.2.15 dst=13.225.210.18 sport=38170 dport=443 src=13.225.210.18 dst=10.0.2.15 sport=443 dport=38170 [ASSURED] mark=0 use=1
tcp      6 110 TIME_WAIT src=10.0.2.15 dst=13.225.210.18 sport=38198 dport=443 src=13.225.210.18 dst=10.0.2.15 sport=443 dport=38198 [ASSURED] mark=0 use=1
tcp      6 111 TIME_WAIT src=10.0.2.15 dst=172.217.9.234 sport=58584 dport=443 src=172.217.9.234 dst=10.0.2.15 sport=443 dport=58584 [ASSURED] mark=0 use=1
tcp      6 112 TIME_WAIT src=10.0.2.15 dst=13.225.210.18 sport=38172 dport=443 src=13.225.210.18 dst=10.0.2.15 sport=443 dport=38172 [ASSURED] mark=0 use=1
tcp      6 112 TIME_WAIT src=10.0.2.15 dst=54.192.100.61 sport=36114 dport=443 src=54.192.100.61 dst=10.0.2.15 sport=443 dport=36114 [ASSURED] mark=0 use=1
conntrack v1.4.6 (conntrack-tools): 11 flow entries have been shown.
```

5. The Router's FORWARD table should set state tracking rules that drop all INVALID traffic and always accept ESTABLISHED and RELATED traffic. Show an example.

Rules:

```
┌──(root💀 ujjwal-router)-[/home/kali]
└─# iptables -A FORWARD -m conntrack --ctstate INVALID -j DROP

┌──(root💀 ujjwal-router)-[/home/kali]
└─# iptables -A FORWARD -m conntrack --ctstate ESTABLISHED,RELATED -j ACCEPT

┌──(root💀 ujjwal-router)-[/home/kali]
└─# iptables -L
Chain INPUT (policy ACCEPT)
target     prot opt source               destination

Chain FORWARD (policy ACCEPT)
target     prot opt source               destination
DROP       all  --  anywhere             anywhere             ctstate INVALID
ACCEPT     all  --  anywhere             anywhere             ctstate RELATED,ESTABLISHED

Chain OUTPUT (policy ACCEPT)
target     prot opt source               destination
```

Example:

```
┌──(root💀 ujjwal-router)-[/var/log]
└─# conntrack -L
udp      17 26 src=10.0.2.15 dst=192.168.1.1 sport=41997 dport=53 src=192.168.1.1 dst=10.0.2.15 sport=53 dport=41997 mark=0 use=1
tcp      6 431996 ESTABLISHED src=10.0.3.10 dst=192.168.0.100 sport=47962 dport=80 src=192.168.0.100 dst=10.0.3.10 sport=80 dport=47962 [ASSURED]
udp      17 26 src=10.0.2.15 dst=192.168.1.1 sport=41004 dport=53 src=192.168.1.1 dst=10.0.2.15 sport=53 dport=41004 mark=0 use=1
conntrack v1.4.6 (conntrack-tools): 3 flow entries have been shown.
```

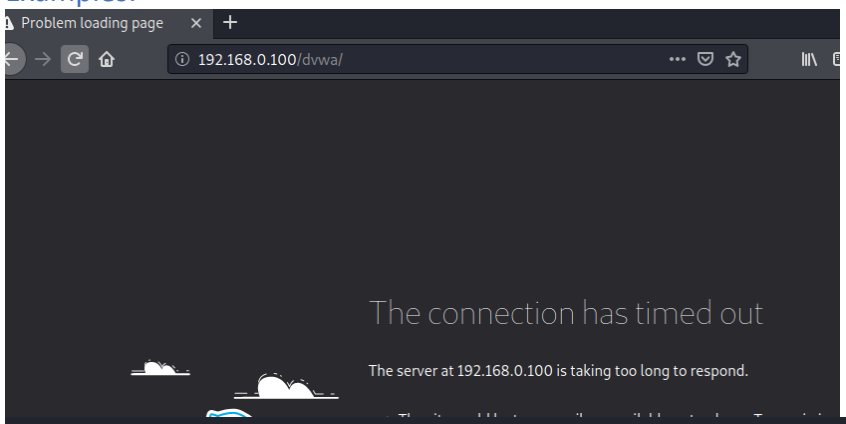
6. The Router's FORWARD table should **only allow NEW traffic from the Attacker machine** for the following services: a. SSH (22)-TCP, b. HTTP (80)-TCP, c. HTTPS(443)-TCP, d. DNS (53)-UDP. Show an example.

Rules:

```
┌──(root💀 ujjwal-router)-[/home/kali]
└─# iptables -A FORWARD -p tcp --dport 80 --syn -m conntrack --ctstate NEW -s 10.0.3.10 -j ACCEPT

┌──(root💀 ujjwal-router)-[/home/kali]
└─# iptables -A FORWARD -p tcp --dport 443 --syn -m conntrack --ctstate NEW -s 10.0.3.10 -j ACCEPT

┌──(root💀 ujjwal-router)-[/home/kali]
└─# iptables -A FORWARD -p udp --dport 53 --syn -m conntrack --ctstate NEW -s 10.0.3.10 -j ACCEPT
iptables v1.8.5 (nf_tables): unknown option "--syn"
Try `iptables -h' or 'iptables --help' for more information.

┌──(root💀 ujjwal-router)-[/home/kali]
└─# iptables -A FORWARD -p udp --dport 53 -m conntrack --ctstate NEW -s 10.0.3.10 -j ACCEPT

┌──(root💀 ujjwal-router)-[/home/kali]
└─# iptables -A FORWARD -p tcp --dport 22 --syn -m conntrack --ctstate NEW -s 10.0.3.10 -j ACCEPT
```

```
┌──(root💀 ujjwal-router)-[/home/kali]
└─# iptables -P FORWARD DROP

┌──(root💀 ujjwal-router)-[/home/kali]
└─# iptables -L
Chain INPUT (policy ACCEPT)
target     prot opt source               destination

Chain FORWARD (policy DROP)
target     prot opt source               destination
ACCEPT     tcp  --  10.0.3.10           anywhere             tcp dpt:ssh flags:FIN,SYN,RST,ACK/SYN ctstate NEW
ACCEPT     tcp  --  10.0.3.10           anywhere             tcp dpt:http flags:FIN,SYN,RST,ACK/SYN ctstate NEW
ACCEPT     tcp  --  10.0.3.10           anywhere             tcp dpt:https flags:FIN,SYN,RST,ACK/SYN ctstate NEW
ACCEPT     udp  --  10.0.3.10           anywhere             udp dpt:domain ctstate NEW

Chain OUTPUT (policy ACCEPT)
target     prot opt source               destination
```

Examples:




```
┌──(root💀 ujjwal-attacker)-[/home/kali]
└─# nmap -sS 192.168.0.100
Starting Nmap 7.91 ( https://nmap.org ) at 2021-04-27 13:36 EDT
Note: Host seems down. If it is really up, but blocking our ping probes, try -Pn
Nmap done: 1 IP address (0 hosts up) scanned in 3.55 seconds
```

```
┌──(root💀 ujjwal-router)-[/home/kali]
└─# tcpdump -i eth1
tcpdump: verbose output suppressed, use -v or -vv for full protocol decode
listening on eth1, link-type EN10MB (Ethernet), capture size 262144 bytes
13:41:36.268840 IP 10.0.3.10 > 192.168.0.100: ICMP echo request, id 44782, seq 11, length 64
13:41:37.289560 IP 10.0.3.10 > 192.168.0.100: ICMP echo request, id 44782, seq 12, length 64
13:41:38.511661 IP 10.0.3.10 > 192.168.0.100: ICMP echo request, id 44782, seq 13, length 64
13:41:39.587115 IP 10.0.3.10 > 192.168.0.100: ICMP echo request, id 44782, seq 14, length 64
13:41:40.611042 IP 10.0.3.10 > 192.168.0.100: ICMP echo request, id 44782, seq 15, length 64
13:41:41.645218 IP 10.0.3.10 > 192.168.0.100: ICMP echo request, id 44782, seq 16, length 64
13:41:42.731940 IP 10.0.3.10 > 192.168.0.100: ICMP echo request, id 44782, seq 17, length 64
13:41:43.799582 IP 10.0.3.10 > 192.168.0.100: ICMP echo request, id 44782, seq 18, length 64
13:41:44.805257 IP 10.0.3.10 > 192.168.0.100: ICMP echo request, id 44782, seq 19, length 64
13:41:45.846720 IP 10.0.3.10 > 192.168.0.100: ICMP echo request, id 44782, seq 20, length 64
13:41:46.888660 IP 10.0.3.10 > 192.168.0.100: ICMP echo request, id 44782, seq 21, length 64
13:41:47.952663 IP 10.0.3.10 > 192.168.0.100: ICMP echo request, id 44782, seq 22, length 64
13:41:49.011267 IP 10.0.3.10 > 192.168.0.100: ICMP echo request, id 44782, seq 23, length 64
```

```
┌──(root💀 ujjwal-router)-[/home/kali]
└─# conntrack -L
tcp      6 7 SYN_RECV src=10.0.3.10 dst=192.168.0.100 sport=47528 dport=80 src=192.168.0.100 dst=10.0.3.10 sport=80 dport=47528 mark=0 use=1
conntrack v1.4.6 (conntrack-tools): 1 flow entries have been shown.
```

```
┌──(root💀 ujjwal-attacker)-[/home/kali]
└─# nc -nv -u 192.168.0.100 10000
(UNKNOWN) [192.168.0.100] 10000 (?) open
```

```
tcp      6 55 SYN_RECV src=10.0.3.10 dst=192.168.0.100 sport=45774 dport=22 src=192.168.0.100 dst=10.0.3.10 sport=22 dport=45774 mark=0 use=1
conntrack v1.4.6 (conntrack-tools): 1 flow entries have been shown.
```

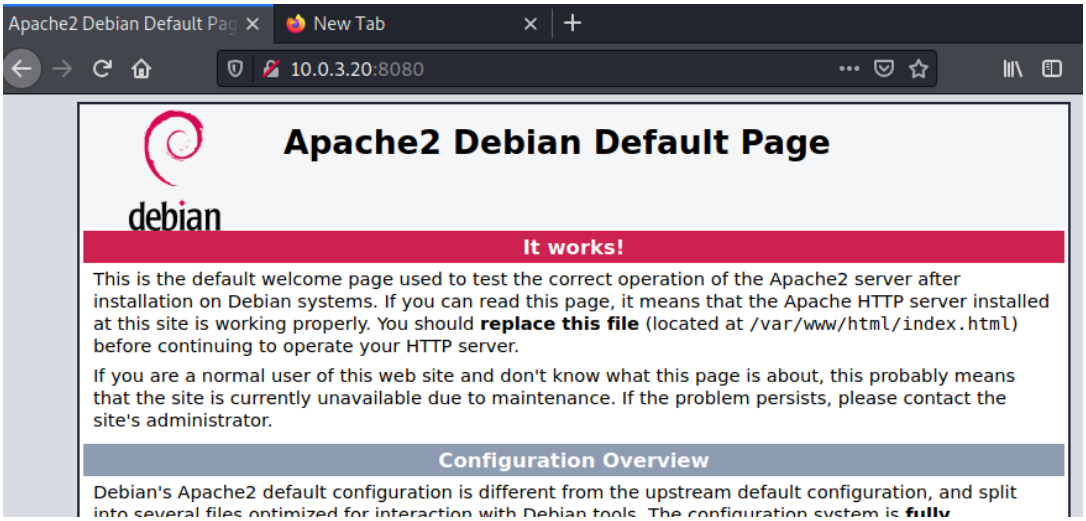
7. On the Router, setup NAT rules that map the web service on the Victim machine (192.168.0.100:80) to the Router (10.0.3.20:8080). Create your own web page and start a webserver to show it works.

```
┌──(root💀 ujjwal-router)-[/etc/apache2]
└─# iptables -t nat -L
Chain PREROUTING (policy ACCEPT)
target     prot opt source               destination
DNAT       tcp  --  anywhere             10.0.3.20            tcp dpt:http-alt to:192.168.0.100:80
DNAT       tcp  --  anywhere             10.0.3.20            tcp dpt:http to:192.168.0.100:80

Chain INPUT (policy ACCEPT)
target     prot opt source               destination

Chain POSTROUTING (policy ACCEPT)
target     prot opt source               destination

Chain OUTPUT (policy ACCEPT)
target     prot opt source               destination

┌──(root💀 ujjwal-router)-[/etc/apache2]
└─# conntrack -L
tcp      6 11 TIME_WAIT src=10.0.3.10 dst=192.168.0.100 sport=47916 dport=80 src=192.168.0.100 dst=10.0.3.10 sport=80 dport=47916 [ASSURED] mark=0 use=1
udp      17 21 src=10.0.2.15 dst=192.168.1.1 sport=57695 dport=53 src=192.168.1.1 dst=10.0.2.15 sport=53 dport=57695 mark=0 use=1
tcp      6 116 TIME_WAIT src=10.0.3.10 dst=10.0.3.20 sport=54680 dport=80 src=192.168.0.100 dst=10.0.3.10 sport=80 dport=54680 [ASSURED] mark=0 use=1
tcp      6 18 TIME_WAIT src=10.0.3.10 dst=10.0.3.20 sport=54666 dport=80 src=192.168.0.100 dst=10.0.3.10 sport=80 dport=54666 [ASSURED] mark=0 use=1
udp      17 21 src=10.0.2.15 dst=192.168.1.1 sport=56453 dport=53 src=192.168.1.1 dst=10.0.2.15 sport=53 dport=56453 mark=0 use=1
conntrack v1.4.6 (conntrack-tools): 5 flow entries have been shown.
```

Rules:

```
┌──(root💀 ujjwal-router)-[/etc/apache2]
└─# iptables -t nat -A PREROUTING -p tcp --dport 8080 -d 10.0.3.20 -j DNAT --to 192.168.0.100:80

┌──(root💀 ujjwal-router)-[/etc/apache2]
└─# iptables -t nat -A PREROUTING -p tcp --dport 80 -d 10.0.3.20 -j DNAT --to 192.168.0.100:80
```

It works, even though the server is off on router it is showing the webpage hosted on the victim machine.



8. On Router, if there is any dropped traffic by INPUT or OUTPUT tables, it should be logged by them. Set default policy to DROP for all tables on the Router. Show a logging example.

Rules:

```
┌──(root💀 ujjwal-router)-[/var/log]
└─# iptables -A OUTPUT -j LOG --log-prefix "OUTPUT DROP: "

┌──(root💀 ujjwal-router)-[/var/log]
└─# iptables -A FORWARD -j LOG --log-prefix "FORWARD DROP: "

┌──(root💀 ujjwal-router)-[/var/log]
└─# iptables -A INPUT -j LOG --log-prefix "INPUT DROP: "

┌──(root💀 ujjwal-router)-[/var/log]
└─# iptables -P FORWARD DROP

┌──(root💀 ujjwal-router)-[/var/log]
└─# iptables -P INPUT DROP

┌──(root💀 ujjwal-router)-[/var/log]
└─# iptables -P OUTPUT DROP

┌──(root💀 ujjwal-router)-[/var/log]
└─# iptables -L
Chain INPUT (policy DROP)
target     prot opt source               destination
LOG        all  --  anywhere             anywhere             LOG level warning prefix "INPUT DROP: "

Chain FORWARD (policy DROP)
target     prot opt source               destination
LOG        all  --  anywhere             anywhere             LOG level warning prefix "FORWARD DROP: "

Chain OUTPUT (policy DROP)
target     prot opt source               destination
LOG        all  --  anywhere             anywhere             LOG level warning prefix "OUTPUT DROP: "

┌──(root💀 ujjwal-router)-[/var/log]
└─#
```

Examples:

```
Apr 28 17:37:05 ujjwal-router kernel: [29135.754531] FORWARD DROP: IN=eth1 OUT=eth2 MAC=08:00:27:f9:56:c0:08:00:27:a3:53:2a:08:00 SRC=10.0.3.10 DST=192.168.0.100 LEN=84 TOS=0x00 PREC=0x00 TTL=63 ID=47113 DF PROTO=ICMP TYPE=8 CODE=0 ID=16835 SEQ=1
Apr 28 17:37:06 ujjwal-router kernel: [29136.824304] FORWARD DROP: IN=eth1 OUT=eth2 MAC=08:00:27:f9:56:c0:08:00:27:a3:53:2a:08:00 SRC=10.0.3.10 DST=192.168.0.100 LEN=84 TOS=0x00 PREC=0x00 TTL=63 ID=47312 DF PROTO=ICMP TYPE=8 CODE=0 ID=16835 SEQ=2
Apr 28 17:37:07 ujjwal-router kernel: [29137.921830] FORWARD DROP: IN=eth1 OUT=eth2 MAC=08:00:27:f9:56:c0:08:00:27:a3:53:2a:08:00 SRC=10.0.3.10 DST=192.168.0.100 LEN=84 TOS=0x00 PREC=0x00 TTL=63 ID=47553 DF PROTO=ICMP TYPE=8 CODE=0 ID=16835 SEQ=3
Apr 28 17:37:08 ujjwal-router kernel: [29138.944765] FORWARD DROP: IN=eth1 OUT=eth2 MAC=08:00:27:f9:56:c0:08:00:27:a3:53:2a:08:00 SRC=10.0.3.10 DST=192.168.0.100 LEN=84 TOS=0x00 PREC=0x00 TTL=63 ID=47748 DF PROTO=ICMP TYPE=8 CODE=0 ID=16835 SEQ=4
Apr 28 17:37:26 ujjwal-router kernel: [29156.484256] FORWARD DROP: IN=eth2 OUT=eth1 MAC=08:00:27:f1:3c:2a:08:00:27:2d:b9:2a:08:00 SRC=192.168.0.100 DST=10.0.3.10 LEN=84 TOS=0x00 PREC=0x00 TTL=63 ID=8974 DF PROTO=ICMP TYPE=8 CODE=0 ID=23558 SEQ=1
Apr 28 17:37:27 ujjwal-router kernel: [29157.507243] FORWARD DROP: IN=eth2 OUT=eth1 MAC=08:00:27:f1:3c:2a:08:00:27:2d:b9:2a:08:00 SRC=192.168.0.100 DST=10.0.3.10 LEN=84 TOS=0x00 PREC=0x00 TTL=63 ID=9120 DF PROTO=ICMP TYPE=8 CODE=0 ID=23558 SEQ=2
Apr 28 17:37:28 ujjwal-router kernel: [29158.531543] FORWARD DROP: IN=eth2 OUT=eth1 MAC=08:00:27:f1:3c:2a:08:00:27:2d:b9:2a:08:00 SRC=192.168.0.100 DST=10.0.3.10 LEN=84 TOS=0x00 PREC=0x00 TTL=63 ID=9267 DF PROTO=ICMP TYPE=8 CODE=0 ID=23558 SEQ=3
Apr 28 17:37:29 ujjwal-router kernel: [29159.555459] FORWARD DROP: IN=eth2 OUT=eth1 MAC=08:00:27:f1:3c:2a:08:00:27:2d:b9:2a:08:00 SRC=192.168.0.100 DST=10.0.3.10 LEN=84 TOS=0x00 PREC=0x00 TTL=63 ID=9453 DF PROTO=ICMP TYPE=8 CODE=0 ID=23558 SEQ=4
Apr 28 17:37:46 ujjwal-router kernel: [29176.927645] INPUT DROP: IN=eth2 OUT= MAC=08:00:27:f1:3c:2a:08:00:27:2d:b9:2a:08:00 SRC=192.168.0.100 DST=192.168.0.200 LEN=84 TOS=0x00 PREC=0x00 TTL=64 ID=30182 DF PROTO=ICMP TYPE=8 CODE=0 ID=40807 SEQ=1
Apr 28 17:37:47 ujjwal-router kernel: [29177.955789] INPUT DROP: IN=eth2 OUT= MAC=08:00:27:f1:3c:2a:08:00:27:2d:b9:2a:08:00 SRC=192.168.0.100 DST=192.168.0.200 LEN=84 TOS=0x00 PREC=0x00 TTL=64 ID=30188 DF PROTO=ICMP TYPE=8 CODE=0 ID=40807 SEQ=2
Apr 28 17:38:01 ujjwal-router kernel: [29191.528430] INPUT DROP: IN=eth1 OUT= MAC=08:00:27:f9:56:c0:08:00:27:a3:53:2a:08:00 SRC=10.0.3.10 DST=10.0.3.20 LEN=84 TOS=0x00 PREC=0x00 TTL=64 ID=4479 DF PROTO=ICMP TYPE=8 CODE=0 ID=32139 SEQ=1

Apr 28 17:38:02 ujjwal-router kernel: [29192.703228] INPUT DROP: IN=eth1 OUT= MAC=08:00:27:f9:56:c0:08:00:27:a3:53:2a:08:00 SRC=10.0.3.10 DST=10.0.3.20 LEN=84 TOS=0x00 PREC=0x00 TTL=64 ID=4527 DF PROTO=ICMP TYPE=8 CODE=0 ID=32139 SEQ=2

Apr 28 17:38:11 ujjwal-router kernel: [29201.807491] OUTPUT DROP: IN= OUT=eth1 SRC=10.0.3.20 DST=10.0.3.10 LEN=84 TOS=0x00 PREC=0x00 TTL=64 ID=10866 DF PROTO=ICMP TYPE=8 CODE=0 ID=11936 SEQ=1
Apr 28 17:38:12 ujjwal-router kernel: [29202.827249] OUTPUT DROP: IN= OUT=eth1 SRC=10.0.3.20 DST=10.0.3.10 LEN=84 TOS=0x00 PREC=0x00 TTL=64 ID=11066 DF PROTO=ICMP TYPE=8 CODE=0 ID=11936 SEQ=2
Apr 28 17:38:13 ujjwal-router kernel: [29203.851806] OUTPUT DROP: IN= OUT=eth1 SRC=10.0.3.20 DST=10.0.3.10 LEN=84 TOS=0x00 PREC=0x00 TTL=64 ID=11088 DF PROTO=ICMP TYPE=8 CODE=0 ID=11936 SEQ=3
```

```
┌──(root💀ujjwal-router)-[/var/log]
└─# ping 10.0.3.10
PING 10.0.3.10 (10.0.3.10) 56(84) bytes of data.
ping: sendmsg: Operation not permitted
ping: sendmsg: Operation not permitted
ping: sendmsg: Operation not permitted
^C
--- 10.0.3.10 ping statistics ---
3 packets transmitted, 0 received, 100% packet loss, time 2044ms
```

```
┌──(root💀ujjwal-attacker)-[/home/kali]
└─# ping 10.0.3.20
PING 10.0.3.20 (10.0.3.20) 56(84) bytes of data.
^C
--- 10.0.3.20 ping statistics ---
2 packets transmitted, 0 received, 100% packet loss, time 1175ms
```

```
┌──(root💀ujjwal-victim)-[/home/kali]
└─# ping 10.0.3.10
PING 10.0.3.10 (10.0.3.10) 56(84) bytes of data.
^C
--- 10.0.3.10 ping statistics ---
4 packets transmitted, 0 received, 100% packet loss, time 3071ms
```

9. Store all the commands in one ***iptables-YourName.sh*** file, make it executable. First flush all rules and remove all policies, then test that it works. Script:

```
root@ujjwal-ro
File  Actions  Edit  View  Help
  GNU nano 5.3                                                    ipta
#IPTABLES=/sbin/iptables
VICTIM=192.168.0.100
VICTIM_SUB=192.168.0.0/24

ATTACKER=10.0.3.10
ATTACKER_SUB=10.0.3.0/24

HOSTNET=10.0.2.2
HOSTNET_SUB=10.0.2.0/24

echo "[+] Starting iptables scripts....."
echo "[+] Flushing all rules"

$IPTABLES -F

echo "[1] Changing INPUT Table to accept only traffic from three subnets...."
$IPTABLES -A INPUT -s $VICTIM_SUB -j ACCEPT
$IPTABLES -A INPUT -s $ATTACKER_SUB -j ACCEPT
$IPTABLES -A INPUT -s $HOSTNET_SUB -j ACCEPT
$IPTABLES -P INPUT DROP
echo "[1] Completed...."

echo "[2] Changing INPUT Table to accept SSH and ICMP only...."
$IPTABLES -A INPUT -j ACCEPT -p TCP --destination-port 22
$IPTABLES -A INPUT -j ACCEPT -p ICMP --icmp-type echo-request
echo "[2] Completed...."

echo "[3] Changing OUTPUT TAble to drop all INVALID Traffic and accept ESTABLISHED and RELATED traffic...."
$IPTABLES -A OUTPUT -m conntrack --ctstate INVALID -j DROP
$IPTABLES -A OUTPUT -m conntrack --ctstate ESTABLISHED,RELATED -j ACCEPT
echo "[3] Completed...."

echo "[4] Changing OUTPUT Table to only allow NEW Traffic for SSH, HTTP, HTTPS, DNS, ping...."
$IPTABLES -A OUTPUT -p tcp --dport 22 --syn -m conntrack --ctstate NEW -j ACCEPT
$IPTABLES -A OUTPUT -p tcp --dport 80 --syn -m conntrack --ctstate NEW -j ACCEPT
$IPTABLES -A OUTPUT -p tcp --dport 443 --syn -m conntrack --ctstate NEW -j ACCEPT
$IPTABLES -A OUTPUT -p udp --dport 53 -m conntrack --ctstate NEW -j ACCEPT
$IPTABLES -A OUTPUT -j ACCEPT -p ICMP --icmp-type echo-request
echo "[4] Completed...."

echo "[5] Changing FORWARD TAble to to drop all INVALID Traffic and accept ESTABLISHED and RELATED traffic...."
$IPTABLES -A FORWARD -m conntrack --ctstate INVALID -j DROP
$IPTABLES -A FORWARD -m conntrack --ctstate ESTABLISHED,RELATED -j ACCEPT
echo "[5] Completed...."

echo "[6] Changing OUTPUT Table to only allow NEW Traffic for SSH, HTTP, HTTPS, DNS, ping from ATTACKER...."
$IPTABLES -A OUTPUT -p tcp --dport 22 --syn -m conntrack --ctstate NEW -s $ATTACKER -j ACCEPT
$IPTABLES -A OUTPUT -p tcp --dport 80 --syn -m conntrack --ctstate NEW -s $ATTACKER -j ACCEPT
$IPTABLES -A OUTPUT -p tcp --dport 443 --syn -m conntrack --ctstate NEW -s $ATTACKER -j ACCEPT
                                                                [ Read
^G Help      ^O Write Out   ^W Where Is    ^K Cut       ^T Execute   ^C Location   M-U Und
^X Exit      ^R Read File   ^\ Replace     ^U Paste     ^J Justify   ^_ Go To Line M-E Red
```

```
┌──(root💀ujjwal-router)-[/home/kali]
└─# chmod u+x iptables-ujjwal.sh
```

```
┌──(root💀ujjwal-router)-[/home/kali]
└─# ./iptables-ujjwal.sh
[+] Starting iptables scripts.....
[+] Flushing all rules
[1] Changing INPUT Table to accept only traffic from three subnets....
[1] Completed....
[2] Changing INPUT Table to accept SSH and ICMP only....
[2] Completed....
[3] Changing OUTPUT TAble to drop all INVALID Traffic and accept ESTABLISHED and RELATED traffic....
[3] Completed....
[4] Changing OUTPUT Table to only allow NEW Traffic for SSH, HTTP, HTTPS, DNS, ping....
[4] Completed....
[5] Changing FORWARD TAble to to drop all INVALID Traffic and accept ESTABLISHED and RELATED traffic....
[5] Completed....
[6] Changing OUTPUT Table to only allow NEW Traffic for SSH, HTTP, HTTPS, DNS, ping from ATTACKER....
[6] Completed....
[7] changing NAT rules to map webservice on Victim to Router....
[7] Completed....
[8] Changing all table policies to drop and LOG dropped traffic....
[8] Completed....
[+] Scripting done....
```

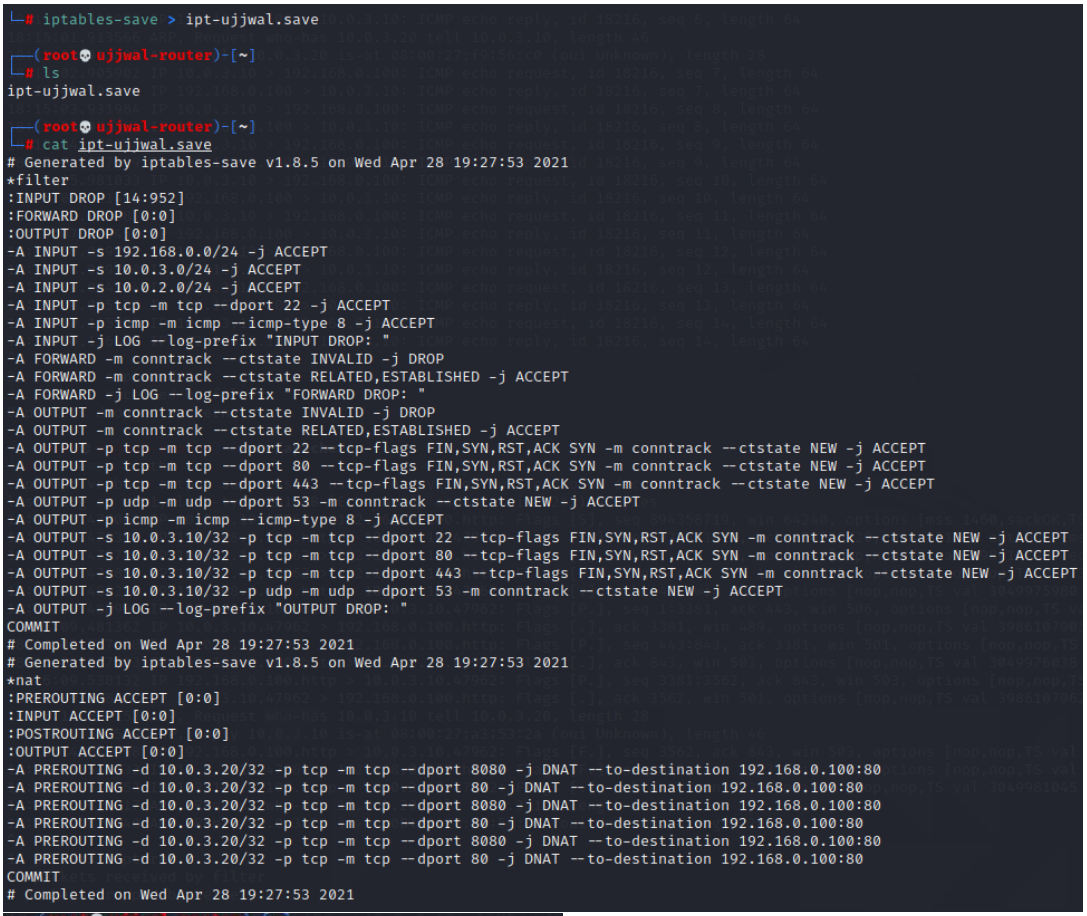
Working:

```
┌──(root💀ujjwal-router)-[/home/kali]
└─# iptables -L
Chain INPUT (policy DROP)
target     prot opt source               destination
ACCEPT     all  --  192.168.0.0/24       anywhere
ACCEPT     all  --  10.0.3.0/24          anywhere
ACCEPT     all  --  10.0.2.0/24          anywhere
ACCEPT     tcp  --  anywhere             anywhere             tcp dpt:ssh
ACCEPT     icmp --  anywhere             anywhere             icmp echo-request
LOG        all  --  anywhere             anywhere             LOG level warning prefix "INPUT DROP: "

Chain FORWARD (policy DROP)
target     prot opt source               destination
DROP       all  --  anywhere             anywhere             ctstate INVALID
ACCEPT     all  --  anywhere             anywhere             ctstate RELATED,ESTABLISHED
LOG        all  --  anywhere             anywhere             LOG level warning prefix "FORWARD DROP: "

Chain OUTPUT (policy DROP)
target     prot opt source               destination
DROP       all  --  anywhere             anywhere             ctstate INVALID
ACCEPT     all  --  anywhere             anywhere             ctstate RELATED,ESTABLISHED
ACCEPT     tcp  --  anywhere             anywhere             tcp dpt:ssh flags:FIN,SYN,RST,ACK/SYN ctstate NEW
ACCEPT     tcp  --  anywhere             anywhere             tcp dpt:http flags:FIN,SYN,RST,ACK/SYN ctstate NEW
ACCEPT     tcp  --  anywhere             anywhere             tcp dpt:https flags:FIN,SYN,RST,ACK/SYN ctstate NEW
ACCEPT     udp  --  anywhere             anywhere             udp dpt:domain ctstate NEW
ACCEPT     icmp --  anywhere             anywhere             icmp echo-request
ACCEPT     tcp  --  10.0.3.10            anywhere             tcp dpt:ssh flags:FIN,SYN,RST,ACK/SYN ctstate NEW
ACCEPT     tcp  --  10.0.3.10            anywhere             tcp dpt:http flags:FIN,SYN,RST,ACK/SYN ctstate NEW
ACCEPT     tcp  --  10.0.3.10            anywhere             tcp dpt:https flags:FIN,SYN,RST,ACK/SYN ctstate NEW
ACCEPT     udp  --  10.0.3.10            anywhere             udp dpt:domain ctstate NEW
LOG        all  --  anywhere             anywhere             LOG level warning prefix "OUTPUT DROP: "
```

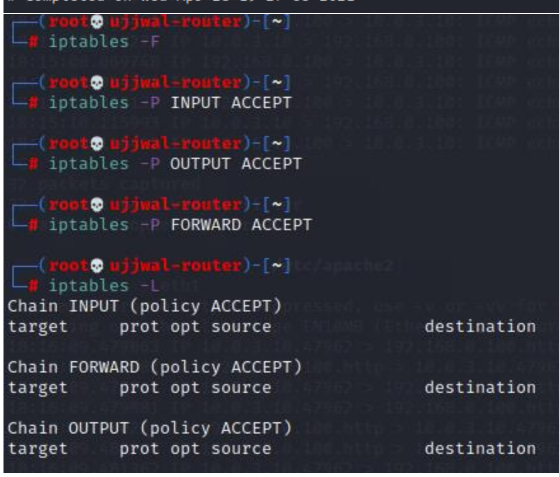
10. Generate an ***ipt-Yourname.save*** file. First flush all rules and remove all policies, then show it works with ***iptables-restore***.

    Commands:

Ujjwal Shah



```
└─# iptables-save > ipt-ujjwal.save

┌──(root💀ujjwal-router)-[~]
└─# ls
ipt-ujjwal.save

┌──(root💀ujjwal-router)-[~]
└─# cat ipt-ujjwal.save
# Generated by iptables-save v1.8.5 on Wed Apr 28 19:27:53 2021
*filter
:INPUT DROP [14:952]
:FORWARD DROP [0:0]
:OUTPUT DROP [0:0]
-A INPUT -s 192.168.0.0/24 -j ACCEPT
-A INPUT -s 10.0.3.0/24 -j ACCEPT
-A INPUT -s 10.0.2.0/24 -j ACCEPT
-A INPUT -p tcp -m tcp --dport 22 -j ACCEPT
-A INPUT -p icmp -m icmp --icmp-type 8 -j ACCEPT
-A INPUT -j LOG --log-prefix "INPUT DROP: "
-A FORWARD -m conntrack --ctstate INVALID -j DROP
-A FORWARD -m conntrack --ctstate RELATED,ESTABLISHED -j ACCEPT
-A FORWARD -j LOG --log-prefix "FORWARD DROP: "
-A OUTPUT -m conntrack --ctstate INVALID -j DROP
-A OUTPUT -m conntrack --ctstate RELATED,ESTABLISHED -j ACCEPT
-A OUTPUT -p tcp -m tcp --dport 22 --tcp-flags FIN,SYN,RST,ACK SYN -m conntrack --ctstate NEW -j ACCEPT
-A OUTPUT -p tcp -m tcp --dport 80 --tcp-flags FIN,SYN,RST,ACK SYN -m conntrack --ctstate NEW -j ACCEPT
-A OUTPUT -p tcp -m tcp --dport 443 --tcp-flags FIN,SYN,RST,ACK SYN -m conntrack --ctstate NEW -j ACCEPT
-A OUTPUT -p udp -m udp --dport 53 -m conntrack --ctstate NEW -j ACCEPT
-A OUTPUT -p icmp -m icmp --icmp-type 8 -j ACCEPT
-A OUTPUT -s 10.0.3.10/32 -p tcp -m tcp --dport 22 --tcp-flags FIN,SYN,RST,ACK SYN -m conntrack --ctstate NEW -j ACCEPT
-A OUTPUT -s 10.0.3.10/32 -p tcp -m tcp --dport 80 --tcp-flags FIN,SYN,RST,ACK SYN -m conntrack --ctstate NEW -j ACCEPT
-A OUTPUT -s 10.0.3.10/32 -p tcp -m tcp --dport 443 --tcp-flags FIN,SYN,RST,ACK SYN -m conntrack --ctstate NEW -j ACCEPT
-A OUTPUT -s 10.0.3.10/32 -p udp -m udp --dport 53 -m conntrack --ctstate NEW -j ACCEPT
-A OUTPUT -j LOG --log-prefix "OUTPUT DROP: "
COMMIT
# Completed on Wed Apr 28 19:27:53 2021
# Generated by iptables-save v1.8.5 on Wed Apr 28 19:27:53 2021
*nat
:PREROUTING ACCEPT [0:0]
:INPUT ACCEPT [0:0]
:POSTROUTING ACCEPT [0:0]
:OUTPUT ACCEPT [0:0]
-A PREROUTING -d 10.0.3.20/32 -p tcp -m tcp --dport 8080 -j DNAT --to-destination 192.168.0.100:80
-A PREROUTING -d 10.0.3.20/32 -p tcp -m tcp --dport 80 -j DNAT --to-destination 192.168.0.100:80
-A PREROUTING -d 10.0.3.20/32 -p tcp -m tcp --dport 8080 -j DNAT --to-destination 192.168.0.100:80
-A PREROUTING -d 10.0.3.20/32 -p tcp -m tcp --dport 80 -j DNAT --to-destination 192.168.0.100:80
-A PREROUTING -d 10.0.3.20/32 -p tcp -m tcp --dport 8080 -j DNAT --to-destination 192.168.0.100:80
-A PREROUTING -d 10.0.3.20/32 -p tcp -m tcp --dport 80 -j DNAT --to-destination 192.168.0.100:80
COMMIT
# Completed on Wed Apr 28 19:27:53 2021
```



```
┌──(root💀ujjwal-router)-[~]
└─# iptables -F

┌──(root💀ujjwal-router)-[~]
└─# iptables -P INPUT ACCEPT

┌──(root💀ujjwal-router)-[~]
└─# iptables -P OUTPUT ACCEPT

┌──(root💀ujjwal-router)-[~]
└─# iptables -P FORWARD ACCEPT

┌──(root💀ujjwal-router)-[~]
└─# iptables -L
Chain INPUT (policy ACCEPT)
target     prot opt source               destination

Chain FORWARD (policy ACCEPT)
target     prot opt source               destination

Chain OUTPUT (policy ACCEPT)
target     prot opt source               destination
```

Working:

```
┌──(root💀ujjwal-router)-[~]
└─# iptables-restore < ipt-ujjwal.save

┌──(root💀ujjwal-router)-[~]
└─# iptables -L
Chain INPUT (policy DROP)
target     prot opt source               destination
ACCEPT     all  --  192.168.0.0/24       anywhere
ACCEPT     all  --  10.0.3.0/24          anywhere
ACCEPT     all  --  10.0.2.0/24          anywhere
ACCEPT     tcp  --  anywhere             anywhere             tcp dpt:ssh
ACCEPT     icmp --  anywhere             anywhere             icmp echo-request
LOG        all  --  anywhere             anywhere             LOG level warning prefix "INPUT DROP: "

Chain FORWARD (policy DROP)
target     prot opt source               destination
DROP       all  --  anywhere             anywhere             ctstate INVALID
ACCEPT     all  --  anywhere             anywhere             ctstate RELATED,ESTABLISHED
LOG        all  --  anywhere             anywhere             LOG level warning prefix "FORWARD DROP: "

Chain OUTPUT (policy DROP)
target     prot opt source               destination
DROP       all  --  anywhere             anywhere             ctstate INVALID
ACCEPT     all  --  anywhere             anywhere             ctstate RELATED,ESTABLISHED
ACCEPT     tcp  --  anywhere             anywhere             tcp dpt:ssh flags:FIN,SYN,RST,ACK/SYN ctstate NEW
ACCEPT     tcp  --  anywhere             anywhere             tcp dpt:http flags:FIN,SYN,RST,ACK/SYN ctstate NEW
ACCEPT     tcp  --  anywhere             anywhere             tcp dpt:https flags:FIN,SYN,RST,ACK/SYN ctstate NEW
ACCEPT     udp  --  anywhere             anywhere             udp dpt:domain ctstate NEW
ACCEPT     icmp --  anywhere             anywhere             icmp echo-request
ACCEPT     tcp  --  10.0.3.10            anywhere             tcp dpt:ssh flags:FIN,SYN,RST,ACK/SYN ctstate NEW
ACCEPT     tcp  --  10.0.3.10            anywhere             tcp dpt:http flags:FIN,SYN,RST,ACK/SYN ctstate NEW
ACCEPT     tcp  --  10.0.3.10            anywhere             tcp dpt:https flags:FIN,SYN,RST,ACK/SYN ctstate NEW
ACCEPT     udp  --  10.0.3.10            anywhere             udp dpt:domain ctstate NEW
LOG        all  --  anywhere             anywhere             LOG level warning prefix "OUTPUT DROP: "
```

11. A 5-10 minutes presentation will be scheduled for you to demo some of project tasks. This should take place in the week before the final exam. Failure to show up will result in a project grade of ZERO.