Урок 5. Настройка сети в Linux. Работа с IPtables

Задание:

* Настроить статическую конфигурацию (без DHCP) в Ubuntu через ip и netplan. Настроить IP, маршрут по умолчанию и DNS-сервера (1.1.1.1 и 8.8.8.8). Проверить работоспособность сети.
* Настроить правила iptables для доступности сервисов на TCP-портах 22, 80 и 443. Также сервер должен иметь возможность устанавливать подключения к серверу обновлений. Остальные подключения запретить.
* Запретить любой входящий трафик с IP 3.4.5.6.
* * Запросы на порт 8090 перенаправлять на порт 80 (на этом же сервере).
* * Разрешить подключение по SSH только из сети 192.168.0.0/24.

После загрузки задания, вы можете проверить себя самостоятельно с помощью эталонного решения

Выполнение;

Настраиваем статическую конфигурацию (без DHCP) в Ubuntu через ip и netplan.
Находим файл конфигурации Netplan введя команду: ls /etc/netplan/

```
user1@ubuntu-server:~$ ls /etc/netplan/
00-installer-config.yaml
user1@ubuntu-server:~$
```

Открываем файл конфигурации в редакторе, введя sudo nano /etc/netplan/00-installer -config.yaml и редактируем;

```
network:
  renderer: networkd
  ethernets:
    enp0s3:
      dhcp4: no
      adrresses: [192.168.1.47/24]
      gateway4: 192.168.1.1
      nameservers:
        addresses:
          - 1.1.1.1
          - 8.8.8.8
  version: 2
```

Далее сохраняем изменения при помощи команды: sudo netplan try

```
user1@ubuntu-server:~$ sudo netplan try
[sudo] password for user1:
/etc/netplan/00-installer-config.yaml:6:7: Error in network definition: unknown key 'adrresses'
      adrresses: [192.168.1.47/24]
      ^

An error occurred: the configuration could not be generated

Reverting.
Traceback (most recent call last):
  File "/usr/share/netplan/netplan/cli/commands/try_command.py", line 99, in command_try
    NetplanApply().command_apply(run_generate=True, sync=True, exit_on_error=False, state_dir=self.state)
  File "/usr/share/netplan/netplan/cli/commands/apply.py", line 131, in command_apply
    raise ConfigurationError("the configuration could not be generated")
netplan.configmanager.ConfigurationError: the configuration could not be generated

During handling of the above exception, another exception occurred:

Traceback (most recent call last):
  File "/usr/sbin/netplan", line 23, in <module>
    netplan.main()
  File "/usr/share/netplan/netplan/cli/core.py", line 50, in main
    self.run_command()
  File "/usr/share/netplan/netplan/cli/utils.py", line 247, in run_command
    self.func()
  File "/usr/share/netplan/netplan/cli/commands/try_command.py", line 81, in run
    self.run_command()
  File "/usr/share/netplan/netplan/cli/utils.py", line 247, in run_command
    self.func()
  File "/usr/share/netplan/netplan/cli/commands/try_command.py", line 113, in command_try
    self.revert()
  File "/usr/share/netplan/netplan/cli/commands/try_command.py", line 143, in revert
    NetplanApply().command_apply(run_generate=False, sync=True, exit_on_error=False, state_dir=tempdir)
  File "/usr/share/netplan/netplan/cli/commands/apply.py", line 254, in command_apply
    NetplanApply.process_sriov_config(config_manager, exit_on_error)
  File "/usr/share/netplan/netplan/cli/commands/apply.py", line 390, in process_sriov_config
    apply_sriov_config(config_manager)
  File "/usr/share/netplan/netplan/cli/sriov.py", line 411, in apply_sriov_config
    parser.load_yaml_hierarchy(rootdir)
```

Проверяем работоспособность сети

```
user1@ubuntu-server:~$ ping 1.1.1.1
PING 1.1.1.1 (1.1.1.1) 56(84) bytes of data.
64 bytes from 1.1.1.1: icmp_seq=1 ttl=58 time=20.6 ms
64 bytes from 1.1.1.1: icmp_seq=2 ttl=58 time=19.7 ms
64 bytes from 1.1.1.1: icmp_seq=3 ttl=58 time=19.6 ms
64 bytes from 1.1.1.1: icmp_seq=4 ttl=58 time=19.7 ms
64 bytes from 1.1.1.1: icmp_seq=5 ttl=58 time=19.7 ms
64 bytes from 1.1.1.1: icmp_seq=6 ttl=58 time=19.7 ms
^C
--- 1.1.1.1 ping statistics ---
6 packets transmitted, 6 received, 0% packet loss, time 5010ms
rtt min/avg/max/mdev = 19.619/19.847/20.625/0.349 ms
user1@ubuntu-server:~$ ping 8.8.8.8
PING 8.8.8.8 (8.8.8.8) 56(84) bytes of data.
64 bytes from 8.8.8.8: icmp_seq=1 ttl=58 time=28.0 ms
64 bytes from 8.8.8.8: icmp_seq=2 ttl=58 time=27.8 ms
64 bytes from 8.8.8.8: icmp_seq=3 ttl=58 time=28.1 ms
64 bytes from 8.8.8.8: icmp_seq=4 ttl=58 time=28.0 ms
64 bytes from 8.8.8.8: icmp_seq=5 ttl=58 time=28.0 ms
64 bytes from 8.8.8.8: icmp_seq=6 ttl=58 time=28.2 ms
^C
--- 8.8.8.8 ping statistics ---
6 packets transmitted, 6 received, 0% packet loss, time 5423ms
rtt min/avg/max/mdev = 27.794/28.015/28.217/0.136 ms
user1@ubuntu-server:~$
```

Настраиваем правила iptables для доступности сервисов на TCP-портах 22, 80 и 443. Для начала посмотрим текущее состояние правил, введя команду (от root): iptables –L –nv

```
user1@ubuntu-server:~$ sudo su
[sudo] password for user1:
root@ubuntu-server:/home/user1# iptables -L -nv
Chain INPUT (policy ACCEPT 0 packets, 0 bytes)
 pkts bytes target     prot opt in     out     source               destination

Chain FORWARD (policy ACCEPT 0 packets, 0 bytes)
 pkts bytes target     prot opt in     out     source               destination

Chain OUTPUT (policy ACCEPT 0 packets, 0 bytes)
 pkts bytes target     prot opt in     out     source               destination
root@ubuntu-server:/home/user1#
```

Добавляем правила для доступности сервисов на TCP-портах 22, 80 и 443 командой: iptables –A INPUT –p dport=22 –j ACCEPT и проверяем правила: iptables –L –nv

```
user1@ubuntu-server:~$ sudo su
[sudo] password for user1:
root@ubuntu-server:/home/user1# iptables -L -nv
Chain INPUT (policy ACCEPT 0 packets, 0 bytes)
 pkts bytes target     prot opt in     out     source               destination

Chain FORWARD (policy ACCEPT 0 packets, 0 bytes)
 pkts bytes target     prot opt in     out     source               destination

Chain OUTPUT (policy ACCEPT 0 packets, 0 bytes)
 pkts bytes target     prot opt in     out     source               destination
root@ubuntu-server:/home/user1# iptables -A INPUT -p tcp --dport=22 -j ACCEPT
root@ubuntu-server:/home/user1# iptables -A INPUT -p tcp --dport=80 -j ACCEPT
root@ubuntu-server:/home/user1# iptables -A INPUT -p tcp --dport=443 -j ACCEPT
root@ubuntu-server:/home/user1# iptables -L -nv
Chain INPUT (policy ACCEPT 0 packets, 0 bytes)
 pkts bytes target     prot opt in     out     source               destination
   82  5904 ACCEPT     tcp  --  *      *       0.0.0.0/0            0.0.0.0/0           tcp dpt:22
    0     0 ACCEPT     tcp  --  *      *       0.0.0.0/0            0.0.0.0/0           tcp dpt:80
    0     0 ACCEPT     tcp  --  *      *       0.0.0.0/0            0.0.0.0/0           tcp dpt:443

Chain FORWARD (policy ACCEPT 0 packets, 0 bytes)
 pkts bytes target     prot opt in     out     source               destination

Chain OUTPUT (policy ACCEPT 6 packets, 832 bytes)
 pkts bytes target     prot opt in     out     source               destination
root@ubuntu-server:/home/user1#
```

Далее разрешаем прохождение любого трафика (без ограничений) на интерфейс командой: iptables –A INPUT –i lo –j ACCEPT и проверяем правила: iptables –L –nv

```
root@ubuntu-server:/home/user1# iptables -A INPUT -i lo -j ACCEPT
root@ubuntu-server:/home/user1# iptables -L -nv
Chain INPUT (policy ACCEPT 3 packets, 486 bytes)
 pkts bytes target     prot opt in     out     source               destination
  235 17688 ACCEPT     tcp  --  *      *       0.0.0.0/0            0.0.0.0/0           tcp dpt:22
    0     0 ACCEPT     tcp  --  *      *       0.0.0.0/0            0.0.0.0/0           tcp dpt:80
    0     0 ACCEPT     tcp  --  *      *       0.0.0.0/0            0.0.0.0/0           tcp dpt:443
    0     0 ACCEPT     all  --  lo     *       0.0.0.0/0            0.0.0.0/0

Chain FORWARD (policy ACCEPT 0 packets, 0 bytes)
 pkts bytes target     prot opt in     out     source               destination

Chain OUTPUT (policy ACCEPT 5 packets, 712 bytes)
 pkts bytes target     prot opt in     out     source               destination
root@ubuntu-server:/home/user1#
```

Далее разрешаем входящие пакеты от соединений: установленные сервером с другими хостами в сети: iptables –I INPUT –m state –state ESTABLISHED,RELATED -j ACCEPT
и смотрим список правил: iptables –L –nv

```
oot@ubuntu-server:/home/user1# iptables -I INPUT -m state --state ESTABLISHED,RELATED -j ACCEPT
oot@ubuntu-server:/home/user1# iptables -L -nv
hain INPUT (policy ACCEPT 2 packets, 458 bytes)
pkts bytes target     prot opt in     out     source               destination
   6   432 ACCEPT     all  --  *      *       0.0.0.0/0            0.0.0.0/0           state RELATED,ESTABLISHED
 382 28048 ACCEPT     tcp  --  *      *       0.0.0.0/0            0.0.0.0/0           tcp dpt:22
   0     0 ACCEPT     tcp  --  *      *       0.0.0.0/0            0.0.0.0/0           tcp dpt:80
   0     0 ACCEPT     tcp  --  *      *       0.0.0.0/0            0.0.0.0/0           tcp dpt:443
   0     0 ACCEPT     all  --  lo     *       0.0.0.0/0            0.0.0.0/0

hain FORWARD (policy ACCEPT 0 packets, 0 bytes)
pkts bytes target     prot opt in     out     source               destination

hain OUTPUT (policy ACCEPT 4 packets, 672 bytes)
pkts bytes target     prot opt in     out     source               destination
oot@ubuntu-server:/home/user1#
```

Далее разрешаем трафик icmp: iptables –A INPUT –p icmp –j ACCEPT и проверяем статус: : iptables –L –nv

```
root@ubuntu-server:/home/user1# iptables -A INPUT -p icmp -j ACCEPT
root@ubuntu-server:/home/user1# iptables -L -nv
Chain INPUT (policy ACCEPT 1 packets, 229 bytes)
 pkts bytes target     prot opt in     out     source               destination
  165 13000 ACCEPT     all  --  *      *       0.0.0.0/0            0.0.0.0/0           state RELATED,ESTABLISHED
  382 28048 ACCEPT     tcp  --  *      *       0.0.0.0/0            0.0.0.0/0           tcp dpt:22
    0     0 ACCEPT     tcp  --  *      *       0.0.0.0/0            0.0.0.0/0           tcp dpt:80
    0     0 ACCEPT     tcp  --  *      *       0.0.0.0/0            0.0.0.0/0           tcp dpt:443
    0     0 ACCEPT     all  --  lo     *       0.0.0.0/0            0.0.0.0/0
    0     0 ACCEPT     icmp --  *      *       0.0.0.0/0            0.0.0.0/0

Chain FORWARD (policy ACCEPT 0 packets, 0 bytes)
 pkts bytes target     prot opt in     out     source               destination

Chain OUTPUT (policy ACCEPT 4 packets, 544 bytes)
 pkts bytes target     prot opt in     out     source               destination
root@ubuntu-server:/home/user1#
```

Далее устанавливаем политику по умолчанию (запрещаем остальные подключения): iptables –P INPUT DROP и смотрим список правил: iptables –L –nv

```
root@ubuntu-server:/home/user1# iptables -P INPUT DROP
root@ubuntu-server:/home/user1# iptables -L -nv
Chain INPUT (policy DROP 1 packets, 229 bytes)
 pkts bytes target     prot opt in     out     source               destination
  220 16928 ACCEPT     all  --  *      *       0.0.0.0/0            0.0.0.0/0           state RELATED,ESTABLISHED
  382 28048 ACCEPT     tcp  --  *      *       0.0.0.0/0            0.0.0.0/0           tcp dpt:22
    0     0 ACCEPT     tcp  --  *      *       0.0.0.0/0            0.0.0.0/0           tcp dpt:80
    0     0 ACCEPT     tcp  --  *      *       0.0.0.0/0            0.0.0.0/0           tcp dpt:443
    0     0 ACCEPT     all  --  lo     *       0.0.0.0/0            0.0.0.0/0
    0     0 ACCEPT     icmp --  *      *       0.0.0.0/0            0.0.0.0/0

Chain FORWARD (policy ACCEPT 0 packets, 0 bytes)
 pkts bytes target     prot opt in     out     source               destination

Chain OUTPUT (policy ACCEPT 4 packets, 512 bytes)
 pkts bytes target     prot opt in     out     source               destination
root@ubuntu-server:/home/user1#
```

Запретить любой входящий трафик с IP 3.4.5.6.

Вводим команду iptables –t filter –A INPUT –s 3.4.5.6 –j REJECT и смотрим список правил: iptables –L –nv:

```
root@ubuntu-server:/home/user1# iptables -t filter -A INPUT -s 3.4.5.6 -j REJECT
root@ubuntu-server:/home/user1# iptables -L -nv
Chain INPUT (policy DROP 1 packets, 229 bytes)
 pkts bytes target     prot opt in     out     source               destination
  394 32388 ACCEPT     all  --  *      *       0.0.0.0/0            0.0.0.0/0           state RELATED,ESTABLISHED
  382 28048 ACCEPT     tcp  --  *      *       0.0.0.0/0            0.0.0.0/0           tcp dpt:22
    0     0 ACCEPT     tcp  --  *      *       0.0.0.0/0            0.0.0.0/0           tcp dpt:80
    0     0 ACCEPT     tcp  --  *      *       0.0.0.0/0            0.0.0.0/0           tcp dpt:443
    0     0 ACCEPT     all  --  lo     *       0.0.0.0/0            0.0.0.0/0
    0     0 ACCEPT     icmp --  *      *       0.0.0.0/0            0.0.0.0/0
    0     0 REJECT     all  --  *      *       3.4.5.6              0.0.0.0/0           reject-with icmp-port-unre
achable

Chain FORWARD (policy ACCEPT 0 packets, 0 bytes)
 pkts bytes target     prot opt in     out     source               destination

Chain OUTPUT (policy ACCEPT 5 packets, 936 bytes)
 pkts bytes target     prot opt in     out     source               destination
root@ubuntu-server:/home/user1#
```

Перенаправляем запросы с порта 8090 на порт 80

Вводим команду:

iptables –t nat –I PREROUTING –p tcp –-dport 8090 –j REDIRECT –-to-port 80 и смотрим что получилось: iptables –L –nv –t nat

```
root@ubuntu-server:/home/user1# iptables -t nat -I PREROUTING -p tcp --dport 8090 -j REDIRECT --to-port 80
root@ubuntu-server:/home/user1# iptables -L -nv -t nat
Chain PREROUTING (policy ACCEPT 0 packets, 0 bytes)
 pkts bytes target     prot opt in     out     source               destination
    0     0 REDIRECT   tcp  --  *      *       0.0.0.0/0            0.0.0.0/0           tcp dpt:8090 redir ports 80

Chain INPUT (policy ACCEPT 0 packets, 0 bytes)
 pkts bytes target     prot opt in     out     source               destination

Chain OUTPUT (policy ACCEPT 0 packets, 0 bytes)
 pkts bytes target     prot opt in     out     source               destination

Chain POSTROUTING (policy ACCEPT 0 packets, 0 bytes)
 pkts bytes target     prot opt in     out     source               destination
root@ubuntu-server:/home/user1#
```

Разрешаем подключение по SSH только из сети 192.168.0.0/24

Создаем правило: iptables -A INPUT -p tcp --dport 22 –s 192.168.0.0/24 –j ACCEPT

и далее устанавливаем политику по умолчанию для порта 22 и вводим команду:

iptables -A INPUT -p tcp --dport 22 –j DROP