Лабораторная работа № 10

Тема: Основы сетевой безопасности.

Цель: Изучение основных принципов и инструментов сетевой безопасности.

Netfilter — встроенный в ядро Linux сетевой фильтр. Для управления netfilter служит утилита iptables. Основа iptables — таблицы, в которых содержатся цепочки с правилами. Основная работа происходит с двумя таблицами:

1. Таблица filter. В этой таблице происходит фильтрация входящего и исходящего трафика, а также транзитный трафик.

2. Таблица nat. Необходима для трансляции адресов и портов.

Таблица "filter" является одной из основных таблиц фильтрации пакетов в утилитах iptables и nftables в Linux. Она используется для управления сетевым трафиком на уровне IP-пакетов (уровень 3 в стеке сетевых протоколов). В таблице "filter" можно создавать различные записи (правила), чтобы фильтровать и манипулировать сетевым трафиком.

Вот некоторые из наиболее распространенных типов записей (правил) в таблице "filter":

1. **INPUT**: Это правила, применяемые к входящим пакетам, которые направляются к самому Linux-хосту.
2. **OUTPUT**: Это правила, применяемые к исходящим пакетам, которые отправляются с Linux-хоста.
3. **FORWARD**: Это правила, применяемые к пакетам, которые перенаправляются через Linux-хост (например, в качестве шлюза или моста).
4. **ACCEPT**: Это действие, указывающее, что пакет должен быть принят и разрешен для прохождения.
5. **DROP**: Это действие, указывающее, что пакет должен быть отброшен и не разрешен для прохождения.
6. **REJECT**: Это действие, указывающее, что пакет должен быть отклонен и отправлен обратно отправителю с соответствующим сообщением об ошибке.

Таблица "nat" является одной из таблиц фильтрации пакетов в утилитах iptables и nftables в Linux. Она используется для управления сетевой трансляцией адресов (Network Address Translation, NAT) и перенаправлением портов. В таблице "nat" можно создавать различные записи (правила), чтобы изменять и манипулировать сетевым трафиком на уровне IP-пакетов (уровень 3 в стеке сетевых протоколов).

Вот некоторые из наиболее распространенных типов записей (правил) в таблице "nat":

7. **PREROUTING**: Это правила, применяемые к пакетам входящего трафика до маршрутизации. Они могут использоваться для изменения адреса назначения или порта пакетов.
8. **POSTROUTING**: Это правила, применяемые к пакетам исходящего трафика после маршрутизации. Они могут использоваться для изменения адреса источника или порта пакетов.
9. **OUTPUT**: Это правила, применяемые к исходящим пакетам, которые отправляются с Linux-хоста. Они могут использоваться для изменения адреса источника или порта пакетов.
10. **MASQUERADE**: Это действие, которое позволяет скрыть внутренние IP-адреса за внешним IP-адресом маршрутизатора. Оно используется, когда вы хотите, чтобы внутренние устройства имели доступ в Интернет через общедоступный IP-адрес.
11. **DNAT** (Destination NAT): Это правило преобразования целевого IP-адреса и порта входящих пакетов. Оно используется для перенаправления входящего трафика на другой IP-адрес и порт в локальной сети.

12. **SNAT** (Source NAT): Это правило преобразования исходного IP-адреса и порта исходящих пакетов. Оно используется для изменения исходного IP-адреса и порта исходящего трафика.
13. **REDIRECT**: Это действие, которое перенаправляет пакеты на другой порт на том же хосте. Оно используется, например, для перенаправления входящего трафика на специфический порт на локальной машине.

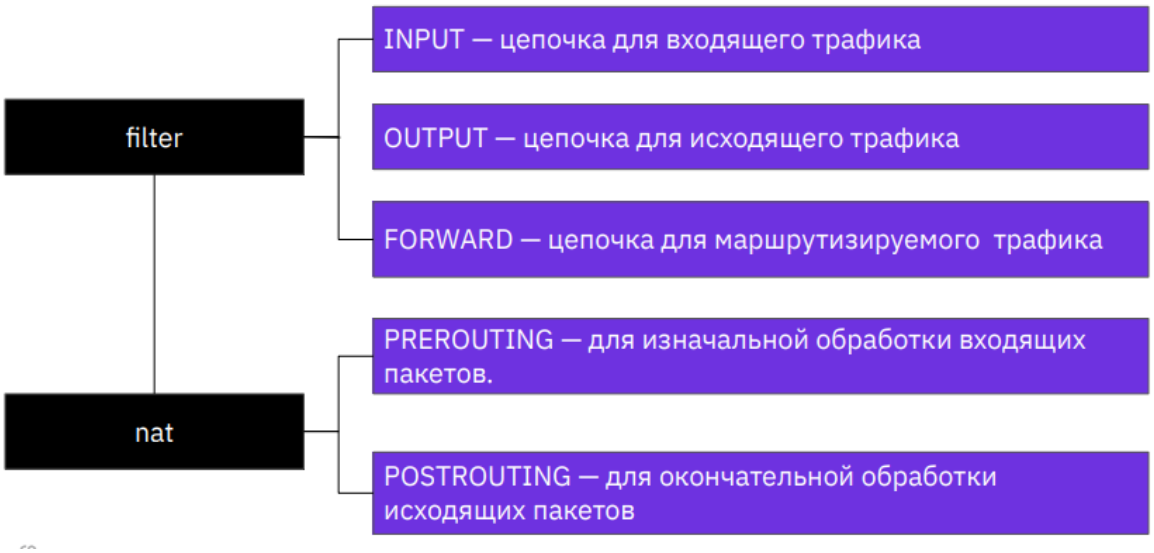
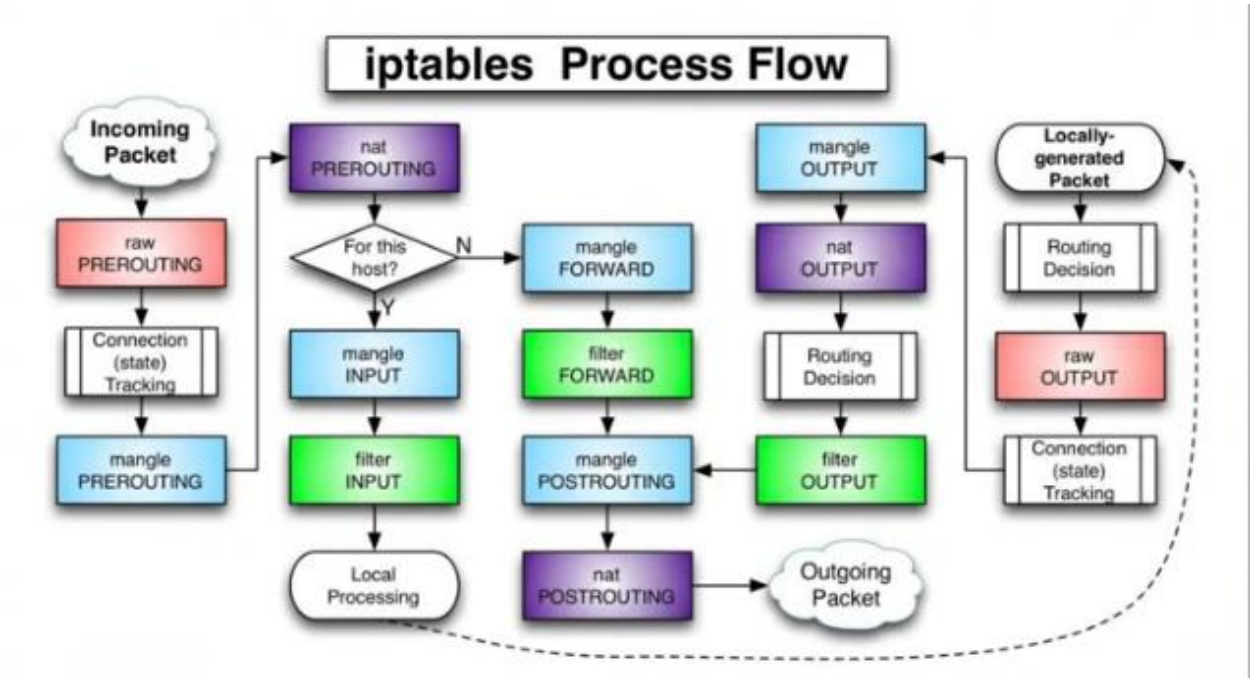


Схема работы netfilter



Утилита iptables

Просмотр портов. Которые слушают приложения:

```
svv@server22:~$ sudo netstat -ntlp
[sudo] password for svv:
Active Internet connections (only servers)
Proto Recv-Q Send-Q Local Address           Foreign Address         State       PID/Program name
tcp        0      0 0.0.0.0:8080            0.0.0.0:*               LISTEN      680/nginx: master p
tcp        0      0 127.0.0.53:53           0.0.0.0:*               LISTEN      593/systemd-resolve
tcp        0      0 0.0.0.0:22              0.0.0.0:*               LISTEN      663/sshd: /usr/sbin
tcp6       0      0 :::8080                 :::*                    LISTEN      680/nginx: master p
tcp6       0      0 :::22                   :::*                    LISTEN      663/sshd: /usr/sbin
svv@server22:~$
```

Просмотр существующих правил, все разрешено:

```
svv@server22:~$ sudo 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
svv@server22:~$
```

Добавим правило.
-A INPUT: Это опция команды iptables, указывающая, что правило должно быть добавлено в цепочку "INPUT". Цепочка "INPUT" применяется к входящим пакетам, которые направляются к самому Linux-хосту.
-p tcp: Это опция команды iptables, которая указывает на протокол TCP. Она ограничивает применение правила только к TCP-пакетам.
--dport=22: Это опция команды iptables, которая указывает на порт назначения (destination port). В данном случае, порт 22, который является стандартным портом для протокола SSH (Secure Shell).
-j ACCEPT: Это опция команды iptables, которая указывает на действие, которое должно быть выполнено для соответствующих пакетов. В данном случае, она указывает на принятие (ACCEPT) пакетов, которые соответствуют заданным условиям (протоколу TCP и порту 22).
Таким образом, данная команда добавляет правило в цепочку "INPUT", которое позволяет принимать входящие TCP-пакеты, направленные на порт 22.

```
svv@server22:~$ sudo iptables -A INPUT -p tcp --dport=22 -j ACCEPT
svv@server22:~$ sudo iptables -L
Chain INPUT (policy ACCEPT)
target     prot opt source               destination
ACCEPT     tcp  --  anywhere             anywhere             tcp dpt:ssh

Chain FORWARD (policy ACCEPT)
target     prot opt source               destination

Chain OUTPUT (policy ACCEPT)
target     prot opt source               destination
svv@server22:~$
```

Новое правило появилось в таблице:

```
svv@server22:~$ sudo iptables -L -n -v
Chain INPUT (policy ACCEPT 0 packets, 0 bytes)
 pkts bytes target     prot opt in     out     source               destination
   44  2448 ACCEPT     tcp  --  *      *       0.0.0.0/0            0.0.0.0/0           tcp dpt:22

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
svv@server22:~$
```

Еще одно правило, разрешим все подключения к localhost:

```
svv@server22:~$ sudo iptables -A INPUT -i lo -j ACCEPT
svv@server22:~$ sudo iptables -L -nv
Chain INPUT (policy ACCEPT 0 packets, 0 bytes)
 pkts bytes target     prot opt in     out     source               destination
  256 14544 ACCEPT     tcp  --  *      *       0.0.0.0/0            0.0.0.0/0           tcp dpt:22
   48  3264 ACCEPT     all  --  lo     *       0.0.0.0/0            0.0.0.0/0
```

Разрешим все подключения по служебному сетевому протоколу icmp:

```
svv@server22:~$ sudo iptables -A INPUT -p icmp -j ACCEPT
svv@server22:~$ sudo iptables -L -nv
Chain INPUT (policy ACCEPT 0 packets, 0 bytes)
 pkts bytes target     prot opt in     out     source               destination
  256 14544 ACCEPT     tcp  --  *      *       0.0.0.0/0            0.0.0.0/0           tcp dpt:22
   48  3264 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
```

Также разрешим HTTP и HTTPS подключения (80 и 443 порты):

```
svv@server22:~$ sudo iptables -A INPUT -p tcp --dport=80 -j ACCEPT
svv@server22:~$ sudo iptables -A INPUT -p tcp --dport=443 -j ACCEPT
svv@server22:~$ sudo iptables -L -nv
Chain INPUT (policy ACCEPT 0 packets, 0 bytes)
 pkts bytes target     prot opt in     out     source               destination
  360 20840 ACCEPT     tcp  --  *      *       0.0.0.0/0            0.0.0.0/0           tcp dpt:22
   80  5584 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 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 0 packets, 0 bytes)
 pkts bytes target     prot opt in     out     source               destination
```

Запретим все подключения, кроме указанных (политика по умолчанию – DROP)

```
svv@server22:~$ sudo iptables -P INPUT DROP
svv@server22:~$ sudo iptables -L -nv
Chain INPUT (policy DROP 0 packets, 0 bytes)
 pkts bytes target     prot opt in     out     source               destination
  508 29952 ACCEPT     tcp  --  *      *       0.0.0.0/0            0.0.0.0/0           tcp dpt:22
  136  9392 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 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 0 packets, 0 bytes)
 pkts bytes target     prot opt in     out     source               destination
svv@server22:~$
```

Разрешим входящие пакеты, которые являются частью уже установленного соединения:

```
svv@server22:~$ sudo iptables -A INPUT -m state --state ESTABLISHED,RELATED -j ACCEPT
svv@server22:~$ sudo iptables -L -nv
Chain INPUT (policy DROP 66 packets, 4036 bytes)
 pkts bytes target     prot opt in     out     source               destination
  162  9456 ACCEPT     tcp  --  *      *       0.0.0.0/0            0.0.0.0/0           tcp dpt:22
   52  3744 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 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
    1    64 ACCEPT     all  --  *      *       0.0.0.0/0            0.0.0.0/0           state RELATED,ESTABLISHED

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

Chain OUTPUT (policy ACCEPT 184 packets, 13872 bytes)
 pkts bytes target     prot opt in     out     source               destination
```

Таблица nat

```
svv@server22:~$ sudo iptables -t nat -L -nv
Chain PREROUTING (policy ACCEPT 0 packets, 0 bytes)
 pkts bytes target     prot opt in     out     source               destination

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
```

Настройка проброса портов. Допустим наш веб-сервер “слушает” порт 8080, а клиенты обращаются на стандартный 80-й порт:

```
svv@server22:~$ sudo iptables -t nat -I PREROUTING -p tcp --dport 80 -j REDIRECT --to-port 8080
svv@server22:~$ sudo iptables -t nat -L
Chain PREROUTING (policy ACCEPT)
target     prot opt source               destination
REDIRECT   tcp  --  anywhere             anywhere             tcp dpt:http redir ports 8080

Chain INPUT (policy ACCEPT)
target     prot opt source               destination

Chain OUTPUT (policy ACCEPT)
target     prot opt source               destination

Chain POSTROUTING (policy ACCEPT)
target     prot opt source               destination
svv@server22:~$
```

Сохранение настроек в файл:

```
svv@server22:~$ sudo iptables-save > ipt.rules
svv@server22:~$ nano ipt.rules
```

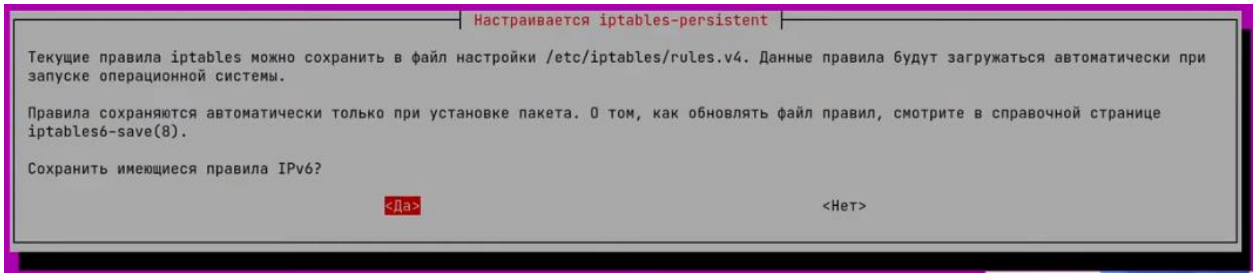
```
  GNU nano 6.2                                        ipt.rules                                        I
 1 # Generated by iptables-save v1.8.7 on Fri Dec 15 03:17:17 2023
 2 *filter
 3 :INPUT DROP [288:17828]
 4 :FORWARD ACCEPT [0:0]
 5 :OUTPUT ACCEPT [0:0]
 6 -A INPUT -p tcp -m tcp --dport 22 -j ACCEPT
 7 -A INPUT -i lo -j ACCEPT
 8 -A INPUT -p icmp -j ACCEPT
 9 -A INPUT -p tcp -m tcp --dport 80 -j ACCEPT
10 -A INPUT -p tcp -m tcp --dport 443 -j ACCEPT
11 COMMIT
12 # Completed on Fri Dec 15 03:17:17 2023
13
```

Чтение настроек из файла:

```
svv@server22:~$ sudo iptables-restore < ipt.rules
```

Можно поручить сохранение и применение настроек iptables специальным утилитам:

```
svv@server22:~$ sudo apt install iptables-persistent netfilter-persistent
```




```
svv@server22:~$ sudo netfilter-persistent save
run-parts: executing /usr/share/netfilter-persistent/plugins.d/15-ip4tables save
run-parts: executing /usr/share/netfilter-persistent/plugins.d/25-ip6tables save
svv@server22:~$ sudo netfilter-persistent start
run-parts: executing /usr/share/netfilter-persistent/plugins.d/15-ip4tables start
run-parts: executing /usr/share/netfilter-persistent/plugins.d/25-ip6tables start
svv@server22:~$
```

Задание:

1. Настроить iptables: разрешить подключения только на 22-й и 80-й порты.
2. Настроить iptables: разрешить пакеты, которые являются частью уже установленного соединения или связаны с таким соединением.
3. Изменить политику на DROP.
4. Настроить проброс портов локально с порта 80 на порт 8000.
5. Проверить, что сервер имеет выход в Интернет, доступен по ssh и веб-сервер.