**МИНОБРНАУКИ РОССИИ**

**Санкт-Петербургский государственный**

**электротехнический университет**

**«ЛЭТИ» им. В.И. Ульянова (Ленина)**

**Кафедра МО ЭВМ**

отчет

**по лабораторной работе №7**

**по дисциплине «Сети и телекоммуникации»**

**Тема: Сетевые экраны. IPTABLES.**

|  |  |  |
| --- | --- | --- |
| Студентка гр. 1304 |  | Чернякова В.А. |
| Преподаватель |  | Ефремов М.А. |

Санкт-Петербург

2023

**Цель работы.**

Целью работы является изучение принципов работы с сетевыми экранами.

**Задание.**

1. Создать три виртуальные машины (лаб. работа № 1).

2. Научиться блокировать и разрешать прием и отправку пакетов с помощью iptables, настраивать логирование событий.

**Выполнение работы.**

В соответствии с заданием, были развёрнуты три виртуальные машины U1, U2 и U3 со следующими сетевыми конфигурациями:

|  |  |  |  |
| --- | --- | --- | --- |
| *Устройство* | U1 | U2 | U3 |
| *IP-адрес* | 15.0.3.14 | 15.0.3.2 | 15.0.3.112 |
| *Маска подсети* | 255.255.255.0 | | |

По умолчанию у устройств свободный доступ друг к другу:

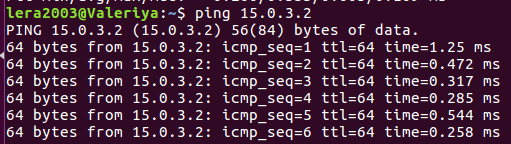


Рисунок 1 – доступ с U1 на U2.

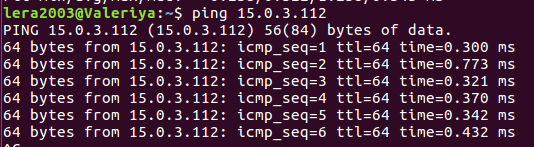


Рисунок 2 – доступ с U1 на U3.

1. Заблокировать доступ по IP-адресу ПК Ub1 к Ub3.

Заблокируем доступ по IP-адресу с 15.0.3.14 на 15.0.3.112. Для этого воспользуемся следующей командой:

sudo iptables –A OUTPUT –d 15.0.3.112 –j DROP

-A – добавление правила в конец списка.

OUTPUT – исходящие пакеты.

-d – адрес назначения.

-j – указание допустимых действий.

DROP – запрещает пакет.

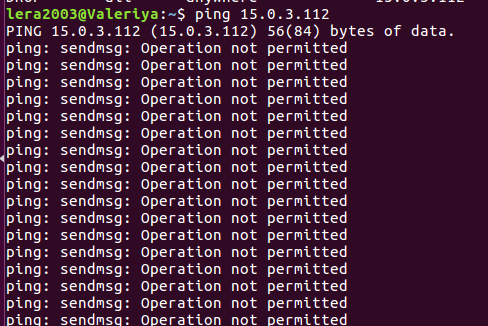


Рисунок 3 – доступ с U1 на U3.

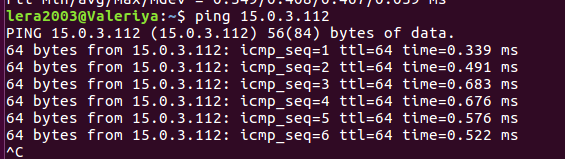


Рисунок 4 – доступ с U2 на U3.

1. Заблокировать доступ по 21-му порту на Ub1.

Для того, чтобы заблокировать доступ по 21 порту на U1

воспользуемся следующей командой:

sudo iptables -A INPUT -p tcp --dport 21 -j REJECT

INPUT – входящие пакеты.

-p – вручную установить протокол (в данном случае TCP).

--dport – порт назначения.

REJECT – запрещает отправку пакета с отправкой сообщения источнику.

Проверим доступность U1 по ssh.

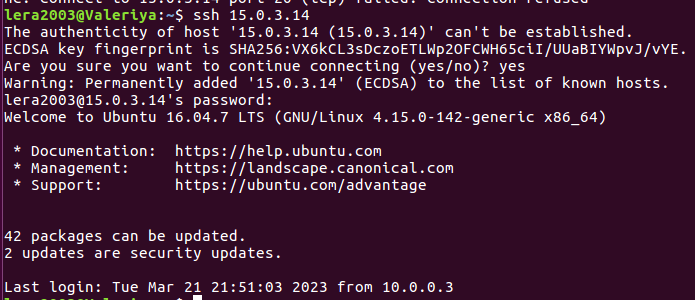


Рисунок 5 – доступ по ssh на U1.

Для проверки недоступности по 21-му порту на U1 воспользуемся утилитой Ubuntu nc. Команда nc (netcat) служит для передачи и получения данных посредством протоколов TCP и UDP.

Использовались следующие ключи:

-v – подробный режим. Используется при сканировании портов;

-z – отключить отправку данных. Используется при сканировании портов.



Рисунок 6 – недоступность по 21-му порту U1.

1. Разрешить доступ только по ssh на Ub2.

Введем команды, блокирующие все пакеты, кроме ssh-соединений:

iptables -A INPUT -p tcp --dport 22 -j ACCEPT

iptables -A INPUT -j DROP

ACCEPT – разрешение пакетов.

Так как ssh-сервер слушает 22-й порт, то первая команда разрешает для TCP-протокола (ключ –p) 22-й порт назначения, а второе правило блокирует все пакеты.

Важен порядок применения правил. Второе правило добавляется в конце таблицы, поэтому первое правило применяется сначала.

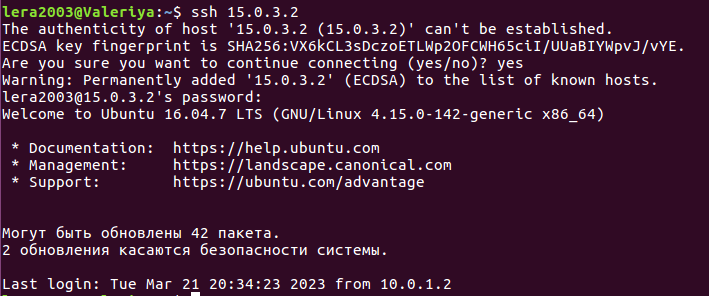


Рисунок 7 – доступ к U2 по ssh.

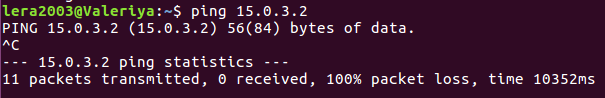


Рисунок 8 – доступ с U1 на U2.

1. Запретить ICMP-запросы на IP-адрес 8.8.8.8 двумя способами.

В начале создадим правило в цепочке INPUT, для запрета принятия пакетов. Команда будет выглядеть так:

sudo iptables –A INPUT –p icmp –s 8.8.8.8 –j REJECT

-s – адрес источника.

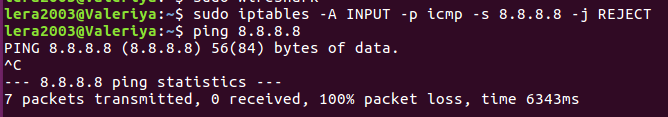


Рисунок 9 – доступ с U3 к 8.8.8.8 после применённого INPUT правила.

Рассмотрим с помощью Wireshark работу хоста при отправке данного запроса.

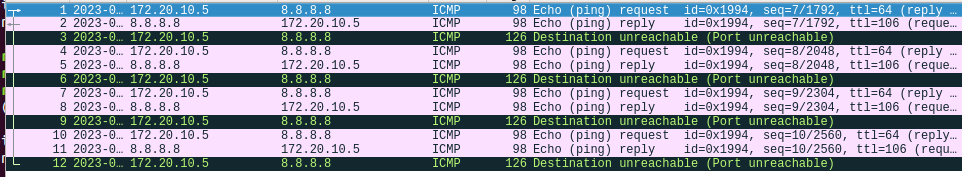


Рисунок 10 – работа хоста при отправке пакетов с U3 на 8.8.8.8 после применённого INPUT правила.

Теперь создадим правило в цепочке OUTPUT, для запрета отправки пакетов. Команда будет выглядеть следующим образом:

sudo iptables –A OUTPUT –p icmp –d 8.8.8.8 –j REJECT

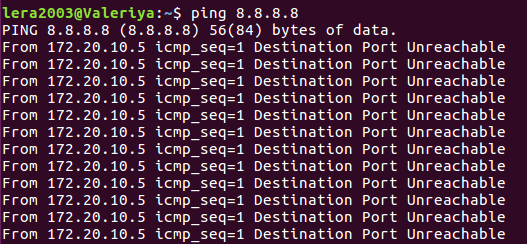


Рисунок 11 – доступ с U3 к 8.8.8.8 после применённого OUTPUT правила.

Рассмотрим с помощью Wireshark работу хоста при отправке данного запроса.

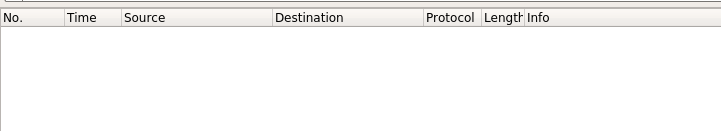


Рисунок 12 – работа хоста при отправке пакетов с U3 на 8.8.8.8 после применённого OUTPUT правила.

Таким образом, можно сделать вывод, что способ запрета ICMP запросов через цепочку правил в OUTPUT более выгодный. В таком случае пакеты вообще не выходят (как видно, консоль в Wireshark в таком случае пуста), общий трафик в подсети не увеличивается.

1. Полностью запретить доступ к Ub3. Разрешить доступ по ICMP протоколу.

Для того, чтобы полностью запретить доступ к U3 и разрешить доступ по ICMP введем команды в следующем порядке:

sudo iptables –A INPUT –p icmp –j ACCEPT

sudo iptables –A INPUT –j REJECT

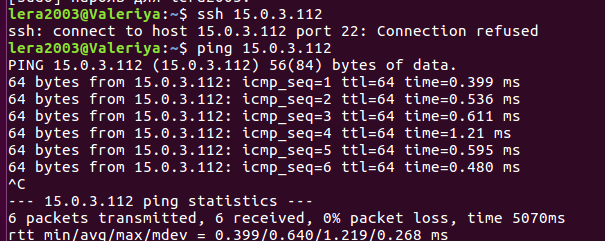


Рисунок 13 – попытка ssh и icmp подключения к U3 с U1.

1. Запретить подключение к Ub1 по порту 80. Настроить логирование попыток подключения по 80-му порту.

С помощью следующей команды запретим по 80 порту подключение к U1. Также настроим логирование попыток подключения по данному порту.

sudo iptables –A INPUT –p tcp -–dport 80 –j LOG –-log-prefix “Package received at port 80”

sudo iptables –A INPUT –p tcp –-dport 80 –j REJECT

Первое правило сначала записывает событие, означающее, что происходит подключение по 80 порту. Второе правило блокирует его. Посмотреть логи можно в этом файле: /var/log/kern.log.

telnet – команда для проверок, связанных с TCP-протоколом.

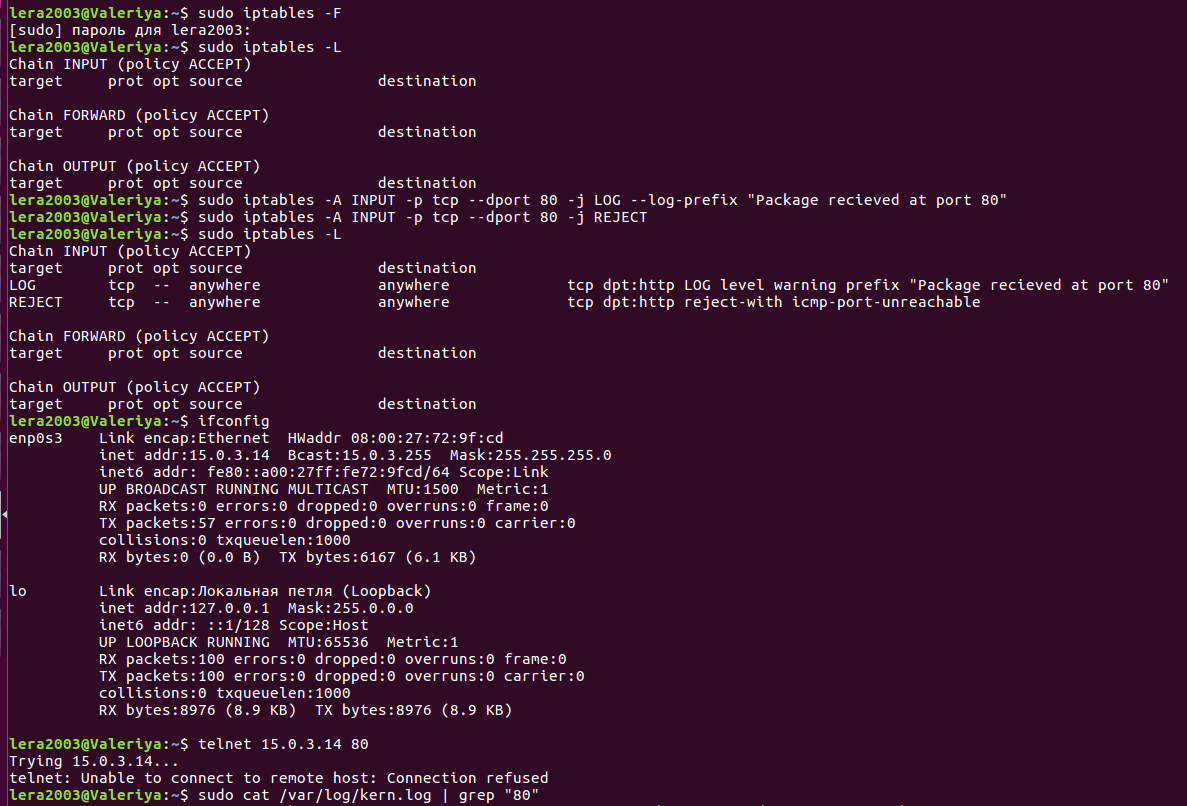


Рисунок 14 – подключение к U1 через 80 порт.

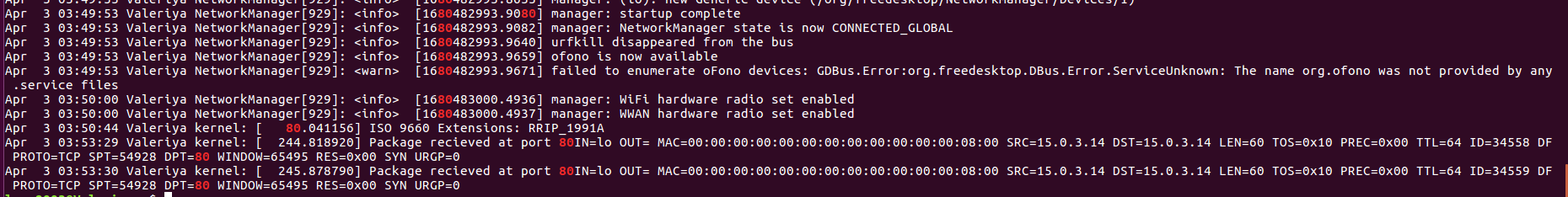


Рисунок 15 – содержимое файла /var/log/kern.log

1. Заблокировать доступ по 80-му порту к Ub3 с Ub1 по его MACадресу.

Заблокируем доступ по 80-му порту к U3 c U1 по его MACадресу.

MACадрес U3 - 08:00:27:aa:bb:cc.

Введем следующую команду, чтобы осуществить раннее описанные действия:

sudo iptables –A INPUT –p tcp -–sport 80 –m mac –-mac-source 08:00:27:aa:bb:cc –j REJECT

Данное правило запретит прием пакетов для 80 порта от узла с указанным MAC-адресом.



Рисунок 16 – подключение по 80 порту с U3 к U1.

Изменим MACадрес U3 и повторим эксперимент.

Измененный MACадрес U3 – 08:00:27:ab:b8:eb.



Рисунок 17 – подключение по 80 порту с U3 к U1 после изменения MACадреса.

Таким образом, запросы блокируются с конкретного MAC. IP адрес роли не играет.

1. Полностью закрыть доступ к Ub1. Разрешить доступ для Ub3 к Ub1, используя диапазон портов 20–79.

Полностью закроем доступ к U1, лишь разрешив доступ для U3 к U1 по диапазону портов 20-79. Для этого введем команду:

sudo iptables –A INPUT –p tcp –s 15.0.3.112 --dport 20:79 –j ACCEPT

sudo iptables –A INPUT –j REJECT

Проверим корректность работы данного правила.

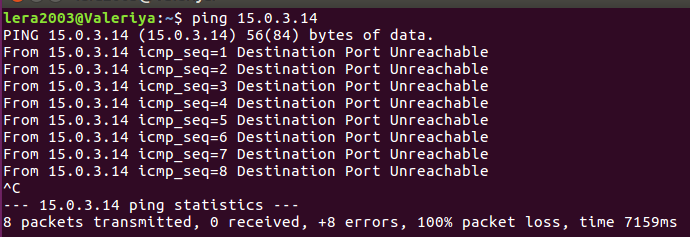


Рисунок 18 – попытка отправки запроса с U2 на U1.

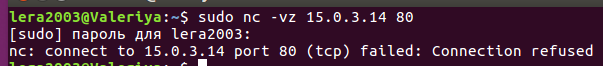


Рисунок 19 – недоступность для U3 к U1 по 80 порту.



Рисунок 20 –доступ для U3 к U1 по 22 порту(ssh).

1. Разрешить только одно ssh-подключение к Ub3.

Для того чтобы разрешить только одно ssh подключение введем следующую команду:

sudo iptables –A INPUT –p tcp -–syn –-dport 22 –m connlimit

--connlimit-above 1 –-connlimit-mask 0 –j -REJECT

--syn – первый пакет, который отправляется при установке нового соединения.

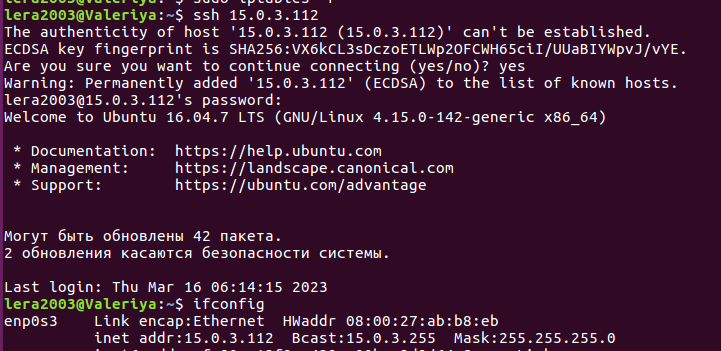
-m – подключает указанный модуль.

сonnlimit – позволяет ограничить количество параллельных подключений к серверу.

--connlimit-above 1 – совпадение, если количество соединений превышает один.

--conlimit-mask 0 – группировка хостов, используя длину префикса.

Для проверки корректности работы правила подключимся с помощью ssh соединения к U1 с U3, и не выходя из сессии попробуем сделать тоже самое с U2.



21 рисунок – подключение по ssh с U3 к U1.



22 рисунок – параллельное подключение по ssh с U2 на U1.

Таким образом, правило, описанное выше разрешает одновременное подключение только одно подключение по ssh.

**Выводы.**

В ходе работы были изучены принципы работы с сетевыми экранами. Были развёрнуты три виртуальные машины, и с помощью утилиты iptables был выполнен ряд задач по фильтрации сетевого трафика.