**МИНОБРНАУКИ РОССИИ**

**Санкт-Петербургский государственный**

**электротехнический университет**

**«ЛЭТИ» им. В.И. Ульянова (Ленина)**

**Кафедра МО ЭВМ**

отчет

**по лабораторной работе №4**

**по дисциплине «Сети и телекоммуникации»**

Тема:**«Применение межсетевого экрана Linux»**

|  |  |  |
| --- | --- | --- |
| Студент гр. 8382 |  | Мирончик П.Д. |
| Преподаватель |  | Лавров А.А. |

Санкт-Петербург

2020

**Цель работы**

Научиться создавать, удалять и изменять правила межсетевого экрана iptables (настройка блокировки трафика, разрешения принятия трафика, логгирования приходящих пакетов).

# Задание

0. На всех машинах запустить скрипт toscrath.sh. Проверить, что таб­лицы ядра пусты.

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

2. Заблокировать доступ по порту X на Ub1.

3. Заблокировать доступ к порту X на Ub3 от UbR. Проверить возмож­ность доступа с Ub1.

4. Полностью запретить доступ к Ub3. Разрешить доступ к порту X.

5. С помощью правила по умолчанию обеспечить блокировку всех вхо­дящих и исходящих пакетов узла Ub3, исключая пакеты управления сетью (протокол ICMP). Убедиться, что Ub3 принимает и отвечает на запросы ко­манды ping, но не отвечает на запросы протокола TCP.

6. Запретить подключение к Ub1 по порту X. Настроить логгирование попыток подключения по порту X.

7. Заблокировать доступ по порту X к Ub3 с Ub1 по его MAC-адресу.

8. Полностью закрыть доступ к Ub1. Разрешить доступ для Ub3 к Ub1, используя диапазон портов X.

9. Разрешить только одно ssh подключение к UbR.

Значение X своё для каждого задания в каждом варианте.

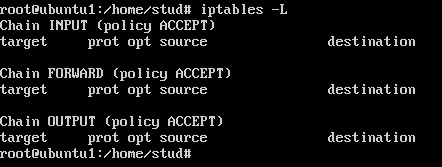
Каждое задание в отчете оформить отдельным подразделом, указав вы­полненные команды (их вывод, если есть), представить результаты проверки успешности выполнения задач. После выполнения каждого задания выпол­нять скрипт delAllTables.sh.

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Вар.\Задание | 2  X= | 3  X= | 4  X= | 6  X= | 7  X= | 8  X= |
| 13 | 33 | 83 | 33 | 83 | 23 | 23-83 |

# Ход работы

**0. На всех машинах запустить скрипт toscratch.sh. Проверить, что таблицы ядра пусты.**

После выполнения скрипта и перезагрузки машины убедимся, что таблицы пусты:

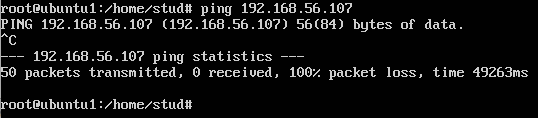


**1. Заблокировать доступ по IP-адресу Ub1 к Ub3.**

Машина Ub1 имеет IP адрес 192.169.56.106, соответственно на Ub3 выполним команду

*iptables –t filter –A INPUT –s 192.168.56.106 –j DROP*

Результат можно проверить, например, при помощи *ping*:



**2. Заблокировать доступ по порту X на Ub1.**

Выполним на машине Ub1

*iptables –t filter –A INPUT –p tcp –dport 33 -j DROP*

*iptables –t filter –A INPUT –p udp –dport 33 -j DROP*

Можно проверить результат при помощи

*telnet 192.168.56.107 33*

отправленного с машины UbR – мы не получим ответа от Ub3, т.к. пакет будет отброшен.

**3. Заблокировать доступ к порту X на Ub3 от UbR. Проверить возмож­ность доступа с Ub1.**

Выполним на машине Ub3

*iptables –t filter –A INPUT –s 192.168.56.108 –p tcp –dport 83 –j DROP*

*iptables –t filter –A INPUT –s 192.168.56.108 –p udp –dport 83 –j DROP*

*iptables –t filter –A INPUT –s 192.168.56.109 –p tcp –dport 83 –j DROP*

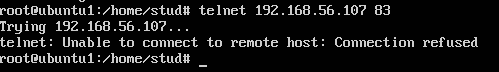
*iptables –t filter –A INPUT –s 192.168.56.109 –p udp –dport 83 –j DROP*

Проверим, что UbR не имеет доступа через

UbR: *telnet 192.168.56.107 83*

и удостоверимся, что ub1 имеет доступ:

Ub1: *telnet 192.168.56.107 83*

****

**4. Полностью запретить доступ к Ub3. Разрешить доступ к порту X.**

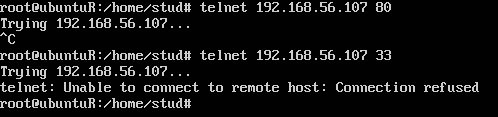
Выполним на ub3

*iptables –P INPUT DROP*

*iptables –t filter –A INPUT –p tcp –dport 33 –j ACCEPT*

*iptables –t filter –A INPUT –p udp –dport 33 –j ACCEPT*

и выполним проверку:



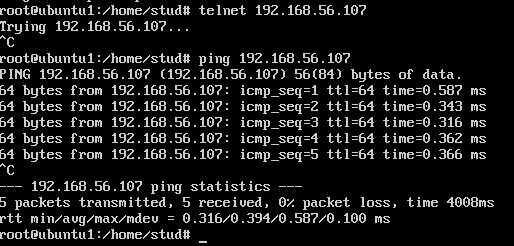
**5. С помощью правила по умолчанию обеспечить блокировку всех входящих и исходящих пакетов узла Ub3, исключая пакеты управления сетью (протокол ICMP). Убедиться, что Ub3 принимает и отвечает на запросы команды ping, но не отвечает на запросы протокола TCP.**

выполним на Ub3

*iptables –P INPUT DROP*

*iptables –t filter –A INPUT –p icmp –j ACCEPT*

И проверим, например, с машины Ub1



**6. Запретить подключение к Ub1 по порту X. Настроить логгирование попыток подключения по порту X.**

Выполним на машине Ub1

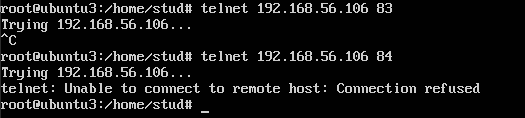
*iptables -t filter -A INPUT -p tcp --dport 83 -j LOG*

*iptables -t filter -A INPUT -p udp --dport 83 -j LOG*

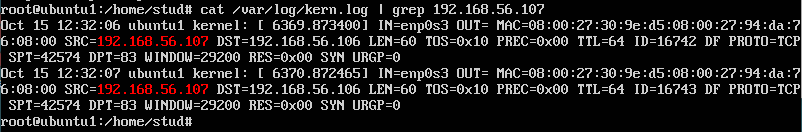
*iptables –t filter –A INPUT –p tcp --dport 83 -j DROP*

*iptables –t filter –A INPUT –p udp --dport 83 -j DROP*

попробуем подключиться к Ub1



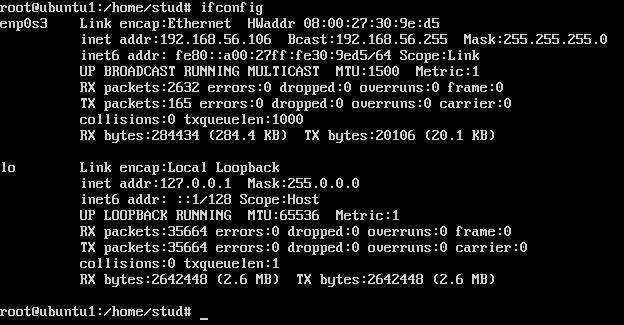
и просмотрим лог



**7. Заблокировать доступ по порту X к Ub3 с Ub1 по его MAC-адресу.**

Выполним на Ub1

*ifconfig*



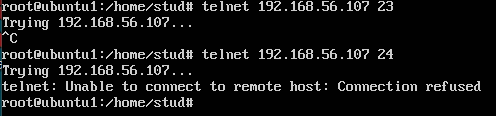
и найдем отсюда mac-адрес Ub1.

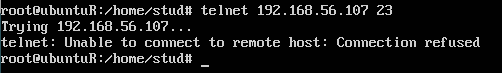
Выполним на Ub3

*iptables -t filter -A INPUT -p tcp --dport 23 -m mac --mac-source 08:00:27:30:9e:d5 -j DROP*

*iptables -t filter -A INPUT -p udp --dport 23 -m mac --mac-source 08:00:27:30:9e:d5 -j DROP*

И проверим результат





**8. Полностью закрыть доступ к Ub1. Разрешить доступ для Ub3 к Ub1, используя диапазон портов X.**

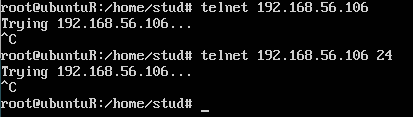
Выполним на Ub1

*iptables -P INPUT DROP*

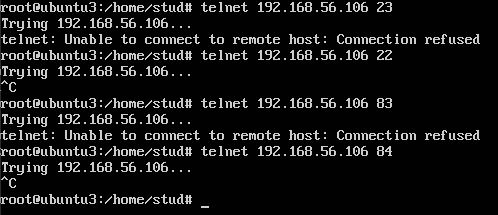
*iptables -t filter -A INPUT -p tcp --dport 23:83 -s 192.168.56.107 -j ACCEPT*

*iptables -t filter -A INPUT -p udp --dport 23:83 -s 192.168.56.107 -j ACCEPT*

Выполним запрос с UbR

**

И с Ub3



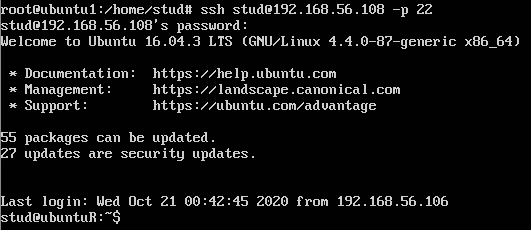
**9. Разрешить только одно ssh подключение к UbR.**

Выполним на UbR

iptables -t filter -A INPUT -p tcp --dport 22 -m connlimit --connlimit-upto 1 --connlimit-mask 0 –j ACCEPT

iptables –t filter –A INPUT –p tcp –dport 22 –j DROP

И проверим результат работы. Подключимся к UbR с Ub1



и попробуем подключится с Ub2



# Вывод

В данной лабораторной работе были изучены способы настройки межсетевого экрана iptables. Разобраны основные параметры iptables, позволяющие изменять правила, блокировать определенный протокол, выбирать определенный порт, источник, в том числе и по mac адресу. Исследована возможность ограничения количества одновременных подключений к одному устройству.

**Идз**

**1. Учет состояний соединений (state) в iptables.**

В Iptables существуют такие типа состояния (доступно, если модуль ‘state‘ загружен с помощью ‘-m state‘):

*NEW* — Все пакеты устанавливающие новое соединение (Например, запрос на установление соединения)

*ESTABLISHED* — Все пакеты принадлежащие установленному соединению (Например, GET ответ web-сервера)

*RELATED* — Пакеты, не принадлежащие установленному соединению (то есть те пакеты, которые являются частью новых соединений, которые было инициированы уже установленным ESTABLISHED соединением), но связанные с ним. (Например — FTP в активном режиме использует разные соединения для передачи данных. Эти соединения связанны.)

*INVALID* — Пакеты, которые не могут быть по тем или иным причинам идентифицированны. Например ICMP ошибки не принадлежащие существующим соединениям

**2. Предложите способ учета объема трафика сетевого взаимодействия с некоторым узлом, идентифицируемым по IP-адресу, на основе iptables и напишите соответствующие правила, необходимые для реализации подобной возможности.**

*iptables –t filter –A INPUT –s 192.168.56.106 –j LOG --log-level 4 --log-prefix “Target found: ”*

Таким образом, мы сохраняем в лог информацию о полученных пакетах от адреса 192.168.56.106. Осталось только посмотреть записи в логах:

*cat /var/log/kern.log | grep “Target found: “ | wc -l*

Эта команда выведет количество полученных от нужного адреса пакетов. Можно ввести

*cat /var/log/kern.log | grep “Target found: “*

для подробной информации о пакетах.