**МИНОБРНАУКИ РОССИИ**

**Санкт-Петербургский государственный**

**электротехнический университет**

**«ЛЭТИ» им. В.И. Ульянова (Ленина)**

**Кафедра МОЭВМ**

отчет

**по лабораторной работе №3**

**по дисциплине «Сети ЭВМ и телекоммуникации»**

Тема: **Использование межсетевого**

**экрана ОС Linux**

|  |  |  |
| --- | --- | --- |
| Студент гр. 6304 |  | Виноградов К.А. |
| Преподаватель |  | Фирсов М.А. |

Санкт-Петербург

2018

**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 21 порту на Ub1.

3. Заблокировать доступ к порту 80 на Ub3 от UbR. Проверить возмож­ность доступа с Ub1.

4. Полностью запретить доступ к Ub3. Разрешить доступ к порту 21.

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 80. Настроить логгирование попыток подключения по 80 порту.

7. Заблокировать доступ по 20 порту к Ub3 с Ub1 по его MAC-адресу.

8. Полностью закрыть доступ к Ub1. Разрешить доступ для Ub3 к Ub1, используя диапазон портов 20-79.

9. Разрешить только одно ssh подключение к UbR.

**Основные теоретические сведения.**

В состав ядра ОС Linux входит подсистема фильтрации пакетов. Данная подсистема в своей работе использует универсальную систему идентифика­ции пакетов iptables. В данной работе рассматривается использование одно­именной административной утилиты для настройки межсетевого экрана на базе ОС Linux.

При получении каждого пакета сетевая подсистема сравнивает данные о нем с информацией в таблицах ядра. На основании этого сравнения сетевая подсистема принимает решение о действии, которое необходимо выполнить над данным пакетом.

Есть 4 таблицы: raw, mangle, nat и filter. Каждая таблица обработки паке­тов содержит несколько предопределенных цепочек, а также может содер­жать цепочки, определенные пользователем. Всего есть 5 предопределенных цепочек:

● PREROUTING – для всех пакетов, приходящих извне;

● INPUT – для пакетов, пришедших извне и предназначенных для ло­кальной системы;

● FORWARD – для маршрутизируемых пакетов, т.е. пакетов, пришед­ших извне и предназначенных для другой системы;

● OUTPUT – для пакетов, созданных в данной системе;

● POSTROUTING – для всех исходящих пакетов.

Каждая цепочка состоит из набора правил, а каждое правило определяет действие над пакетом, которое будет выполняться в случае, если пакет соот­ветствует условиям, заданным в правиле. Такое действие будем называть «цель». «Целью» может быть как переход к любой определенной пользовате­лем цепочке, так и одно из предопределенных действий.

Для каждого правила в таблице фильтрации можно задать в качестве «цели»:

● перевод пакета в пользовательскую цепочку;

● принятие пакета (ACCEPT);

● «выброс» пакета (DROP);

● отказ от получения пакета (REJECT);

● передача пакета служебной программе (QUEUE);

При принятии пакета он передается далее – либо в пользовательский процесс, либо по сети в соответствии с правилами маршрутизации. При вы­бросе пакета он просто теряется. Отказ от получения пакета вызывает от­правление сообщения источнику пакета о том, что пакет не может быть при­нят. Возможен выбор сообщаемой причины непринятия пакета. Передача па­кета служебной программе используется для какой-либо специальной обра­ботки пакетов.

**Ход работы.**

1. Скрипт запущен на всех машинах.

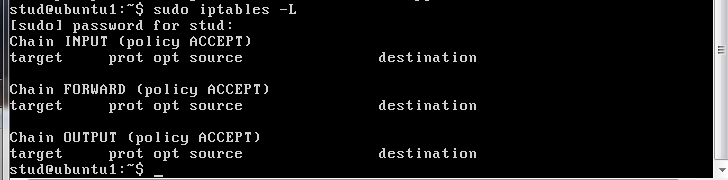


Рисунок 1 – Проверка таблиц

1. Для начала проверим доступ от Ub1 к Ub3.

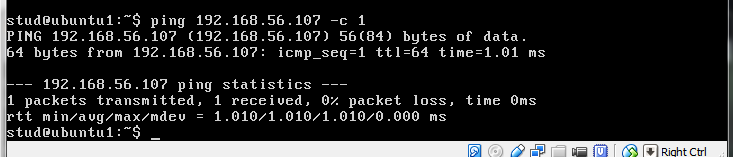


Рисунок 2 – Проверка доступа Ub1 к Ub3

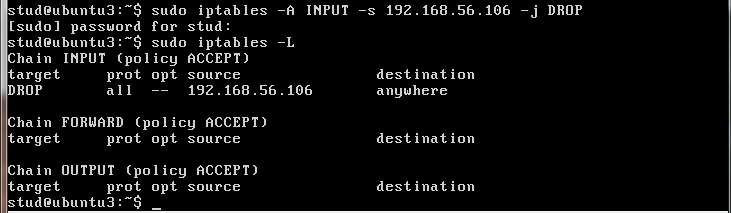


Рисунок 3 – Блокировка Ub1 в Ub3

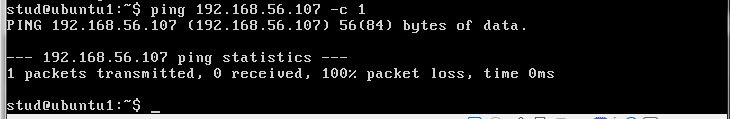


Рисунок 4 – Проверка доступа после блокировки

1. Для начала проверим доступ к Ub1 по 21 порту.



Рисунок 5 – Настройка портов на прослушивание на Ub1



Рисунок 6 – Проверка доступа с Ub3 по 21 порту



Рисунок 7 – Проверка доступа с UbR по 21 порту

Далее закроем доступ к 21 порту на Ub1 с помощью команды *sudo iptables –A INPUT –p tcp --dport 21 –j REJECT*, и снова проверим доступ к Ub1 по 21 порту.

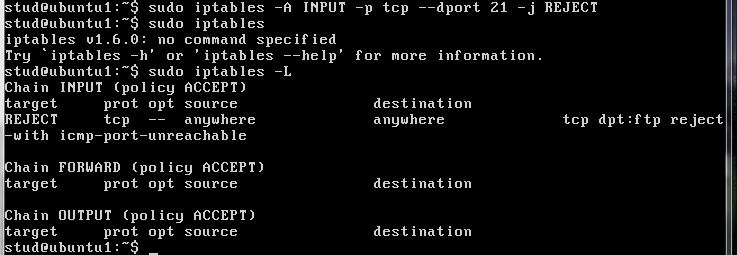


Рисунок 8 – Запрет доступа на Ub1 по 21 порту

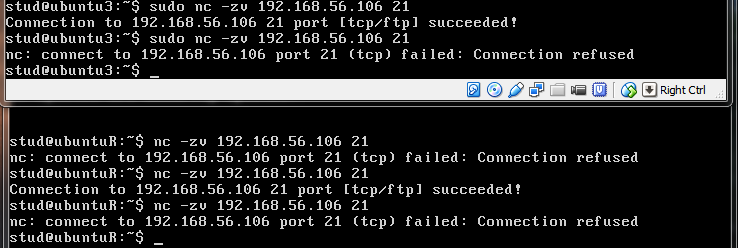


Рисунок 9 - Проверка доступа с Ub3 и UbR по 21 порту

1. Для начала проверим доступ к Ub3 с UbR по 80 порту.

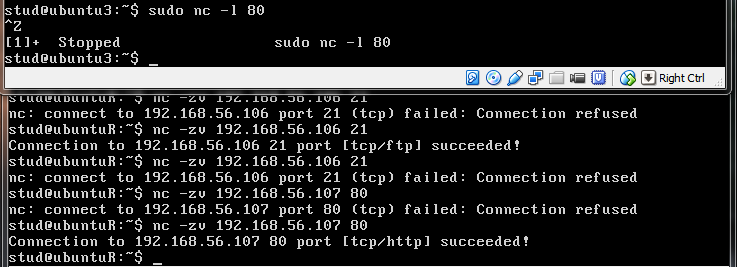


Рисунок 10 – Настройка порта на слушание и проверка подключения UbR к Ub3 по 80 порту

Далее заблокируем доступ к 80 порту на Ub3 по ip адресу UbR с помощью команды *sudo iptables –A INPUT –p tcp --dport 80 –s 192.168.56.108 –j REJECT*, и снова проверим доступ к Ub3 по 80 порту.

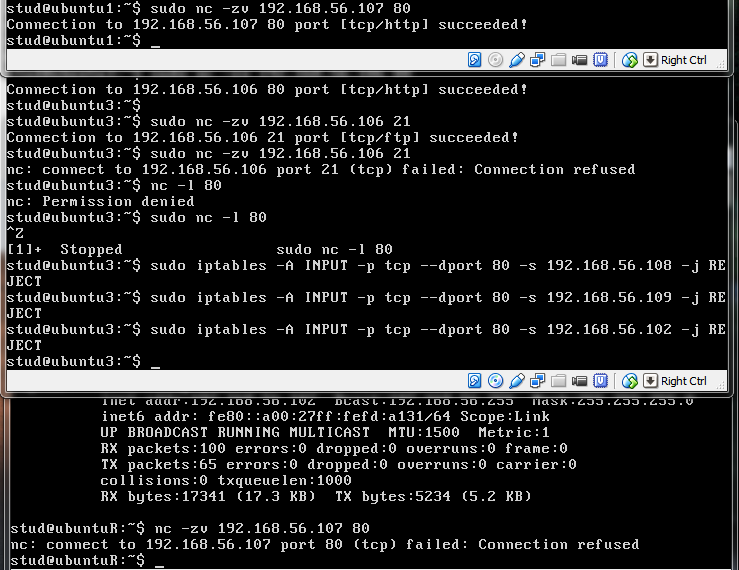


Рисунок 11 – Запрет UbR по IP и проверка

1. Полностью закроем доступ к Ub3, кроме доступа по 21 порту.

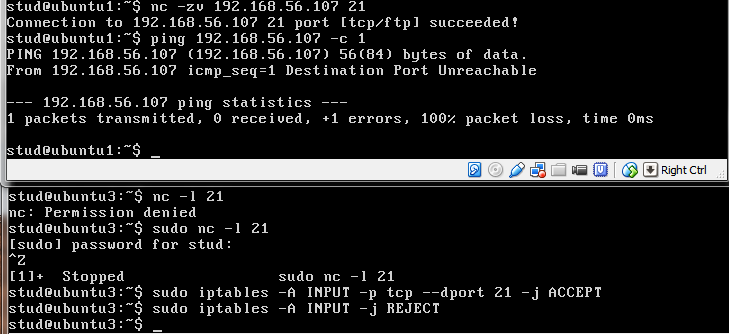


Рисунок 12 – Команды для изменения доступа и проверка

Правило, принимающее пакеты на 21 порт, расположено до того, которое отклоняет все входящие пакеты, поэтому у Ub3 есть доступ к 21 порту.

1. Для начала проверим доступ к Ub3. Далее для цепочек INPUT, OUTPUT сначала по умолчанию установим цель DROP, затем добавим правило для проверки ICMP с целью ACCEPT.

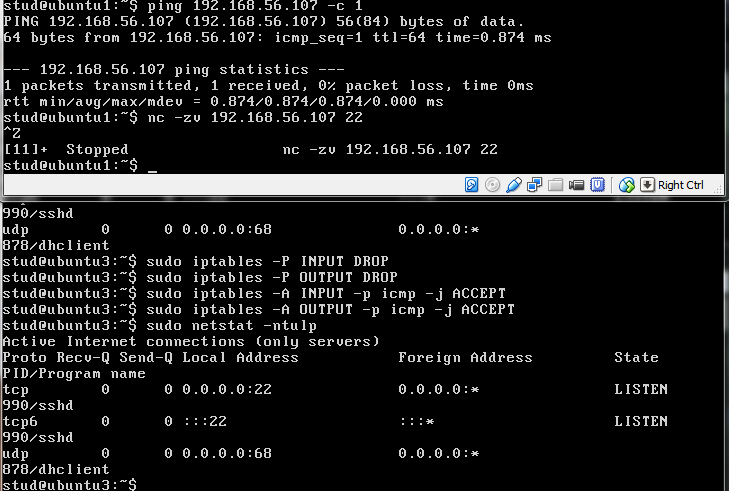


Рисунок 13 – Проверка до команд, ввод команд, проверка после ввода команд

Таким образом, Ub3 доступна для протокола ICMP, но недоступна для TCP.

1. Для начала проверим доступ к Ub1 по 80 порту.

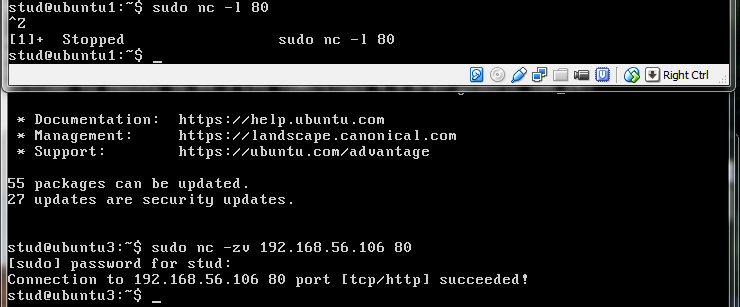


Рисунок 14 – Проверка доступа к Ub1 по 80 порту

Далее запретим подключение к Ub1 по порту 80 и настроим логгирование.

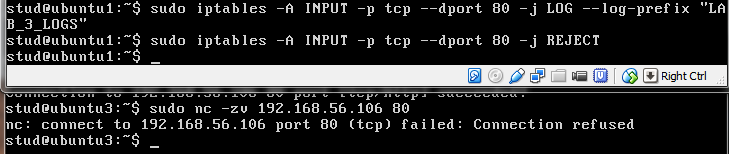


Рисунок 15 – Ввод команд и проверка

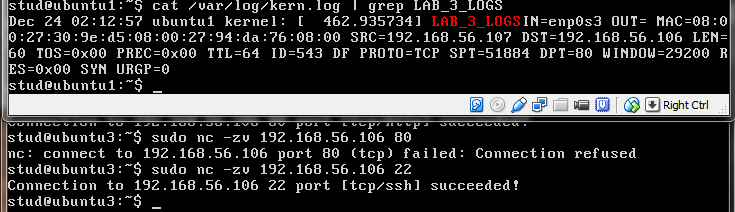


Рисунок 16 – Логгирование и подключение по другому порту

При этом по другому порту доступ остался.

1. Для начала проверим доступ к Ub3 по 20 порту.

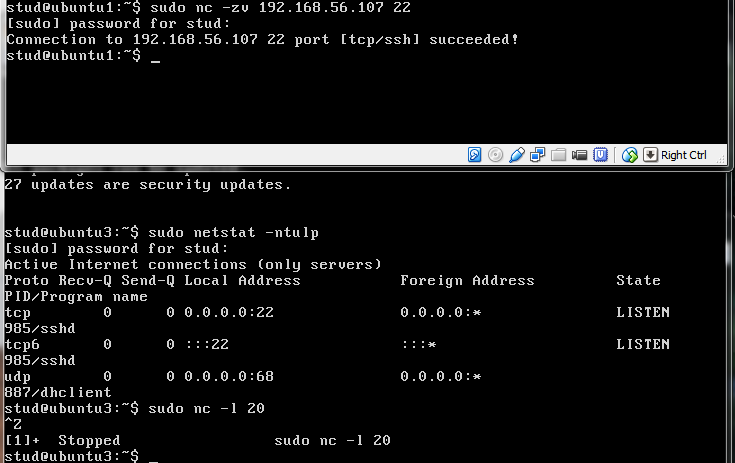


Рисунок 17 - Проверка доступа на Ub3 по порту 20

Далее заблокируем доступ по 20 порту к Ub3 с по его MAC-адресу с помощью команды *sudo iptables –A INPUT –p tcp --dport 20 –m mac --mac-source 08:00:27:30:9e:d5 –j REJECT*.

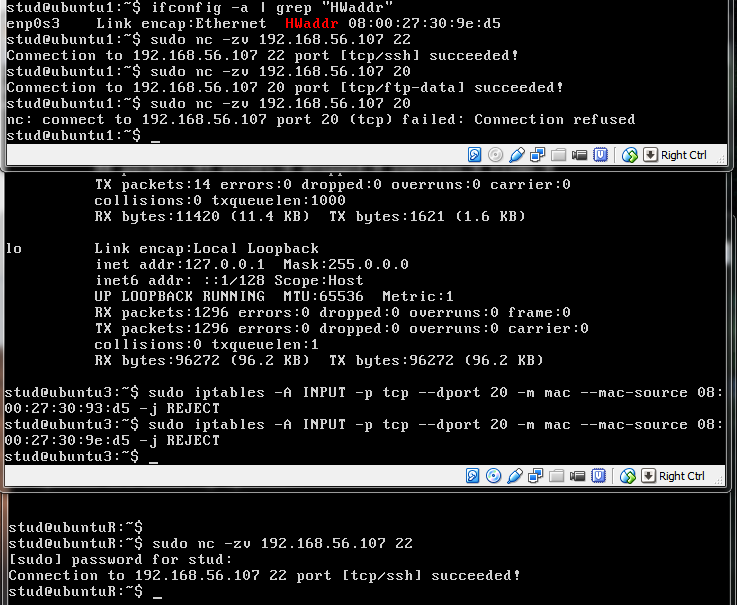


Рисунок 18 – Блокировка, проверка с Ub1 по порту 20 и с UbR по порту 22

1. Для начала проверим доступ к Ub1.

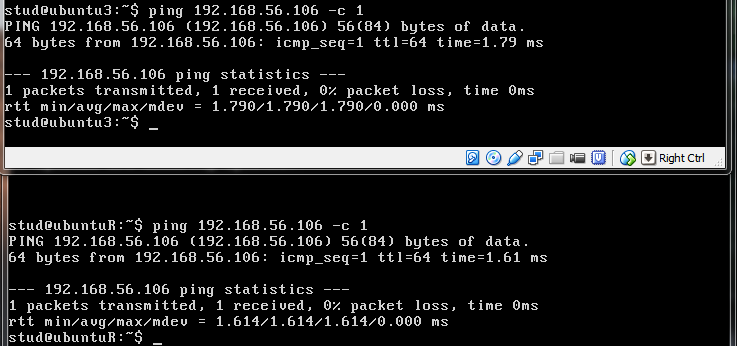


Рисунок 19 - Проверка доступности Ub1 с Ub3 и UbR

Далее закроем доступ к Ub1, открыв его при этом для портов из диапазона 20-79 для Ub3.

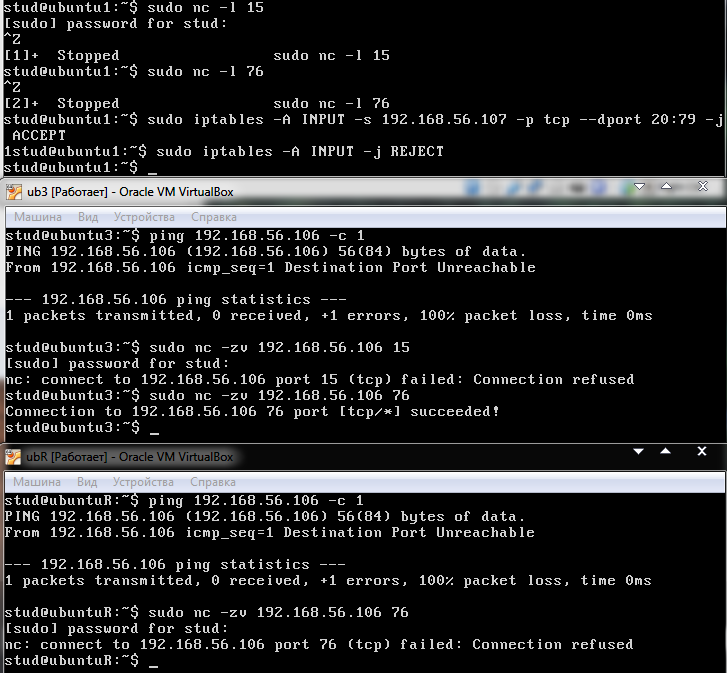


Рисунок 20 – Введённые команды и проверка после них

В результате доступ с помощью ping закрыт для Ub3 и UbR, но при этом доступ к портам 20-79 открыт для Ub3, и закрыт для UbR.

1. Для начала проверим возможность одновременного подключения с Ub1 и Ub3.

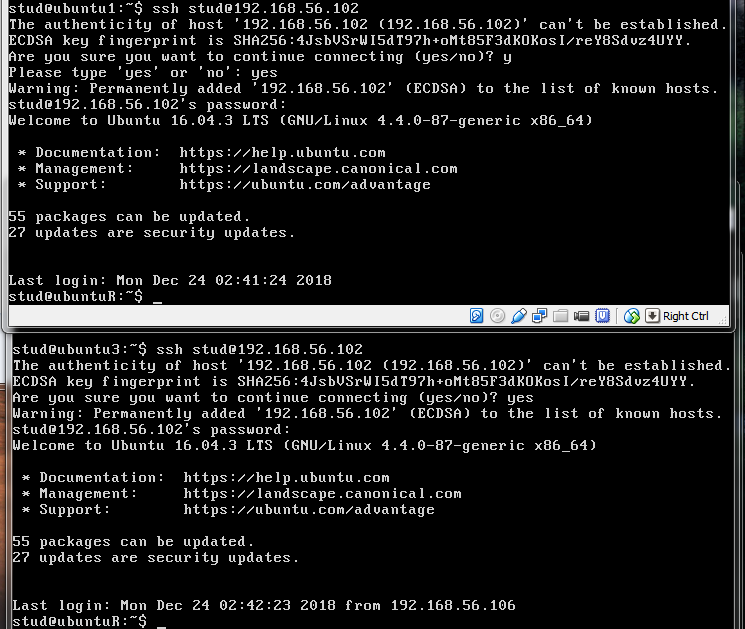


Рисунок 21 - одновременное ssh подключение Ub3 и UbR

Далее ограничим число подключений до 1 по порту 22 с ip адресов с маской /24.

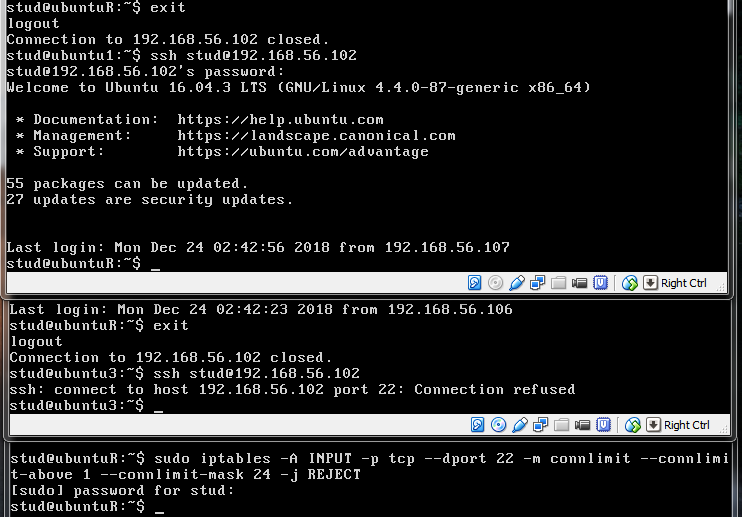


Рисунок 22 – Настройка, выход, переподключение

Таким образом, одновременное подключение невозможно.

**Вывод.**

В результате выполнения данной лабораторной работы были изучены правила межсетевого экрана iptables, а также работа по добавлению, удалению и изменению их, в том числе настройка блокировки трафика, разрешения принятия тра­фика, логгирования приходящих пакетов.

Запет подключений по дефолту