Лабораторная 1.

[**Transmission Control Protocol**](https://ru.wikipedia.org/wiki/Transmission_Control_Protocol)**(TCP) и**[**Internet Protocol**](https://ru.wikipedia.org/wiki/Internet_Protocol)**(IP),**

**Понятие IP-адреса.** IP-адрес (Internet Protocol Address) – уникальный адрес для обозначения устройств, например персональных ПК, мобильных телефонов, серверов, служащий для идентификации устройства в сети и общения с другими устройствами в сети. Каждое устройство, подключенное к сети Интернет, должно иметь уникальный IP-адрес внутри сети. Другими словами, IP-адрес аналогичен домашнему адресу и телефонному номеру, с помощью которых можно однозначно определить человека. **В рамках данного пособия будет изучен IPv4. Он использует 32-битное число для представления IP-адреса.**

**IP (Internet Protocol)** – протокол сетевого уровня стека TCP/IP; согласно протоколу IP у каждого узла сети есть IP-адрес (адреса), что обеспечивает возможность пересылки пакетов между любыми узлами сети. Пересылка в глобальной сети (между сетями) выполняется маршрутизаторами на сетевом уровне, а в локальном сегменте сети доставка пакета до адресата выполняется на канальном уровне, по физическому адресу.

**Маска подсети.** Сетевые устройства логически объединяются в подсети. Каждому сетевому устройству назначаются IP-адрес и маска подсети (netmask). **Маска подсети в IPv4 состоит из 32 бит, непрерывной последовательности единиц (1), за которой следует непрерывная последовательность нулей (0). В маске подсети не может стоять единица после нуля.** Маска подсети служит 10 для разделения IP-адресов на подсети и определения того, какие из них будут принадлежать каждой подсети.

IP-адрес делится с помощью маски на адрес подсети (префикс сети) и адрес узла. Рассмотрим пример:

Поэтому ее можно записывать приставкой к IP-адресу, например: 192.168.0.1/24. В данном случае 24 – это количество единиц в маске. Такая запись называется CIDR и может принимать значения от 1 до 32, по количеству единиц в 32-битном значении маски.

Сетевой префикс (адрес сети) вычисляется побитовой операцией AND между IP-адресом и маской. В приведенном примере это первые три четверти записи IP-адреса, поэтому префикс подсети – 192.168.0.0. Адресом узла является оставшаяся часть, которая соответствует нулям маски – 0.0.0.1.

Принадлежность IP-адресов одной подсети. Для того, чтобы выяснить, принадлежат ли два IP-адреса одной подсети, необходимо выполнить следующее. Во-первых, их маски должны быть одинаковы. Во-вторых, нужно перевести IP-адреса и маску в двоичный вид:

**Широковещательный IP-адрес.** Широковещательный IP-адрес (broadcast) нужен для того, чтобы отправить данные всем узлам подсети. Например, протокол ARP, который связывает MAC-адрес устройства и IP-адрес, использует для этого широковещательный IP-адрес. Для того, чтобы получить широковещательный IP-адрес, необходимо «перевернуть» значения маски и применить побитовое OR между IP-адресом и маской:

Отправляя пакет с указанным IP-адресом, система обеспечит его получение всеми узлами, находящимися в данной подсети.

**Сетевой уровень (межсетевой уровень)** модели TCP/IP служит для образования единой транспортной системы, объединяющей несколько сетей, причем эти сети могут использовать совершенно различные принципы передачи сообщений между конечными узлами и обладать произвольной топологией.

**Канальный уровень** предназначен для обеспечения взаимодействия сетей на физическом уровне и контроля за ошибками передачи данных. Связь между канальным и сетевым уровнем обеспечивается протоколами преобразования между сетевыми и физическими адресами – ARP и InARP.

**Каждый компьютер в сети TCP/IP имеет адреса двух типов: физический (или локальный) и IP-адрес**. Физический адрес узла определяется технологией построения отдельной сети, в которую входит данный узел. Для узлов, входящих в локальные сети, – это МАС-адрес сетевого адаптера или порта маршрутизатора. Такие адреса назначаются производителями оборудования и являются уникальными адресами, поскольку управляются централизовано. Для IPv4-сетей МАС-адрес имеет формат 6 байтов: старшие 3 байта – идентификатор фирмы производителя, а младшие 3 байта назначаются уникальным образом самим производителем.

Адрес IPv4 состоит из 4 байт. Этот адрес используется на сетевом уровне. Он назначается администратором во время конфигурирования компьютеров и маршрутизаторов. IP-Адрес состоит из двух частей: номера сети и номера узла. Номер сети определяет конкретную физическую сеть, а номер узла определяет конкретную рабочую станцию, сервер и пр., включенную в сеть. Подсеть – это физический сегмент TCP/IP сети, в котором используются IP-адреса с общим номером сети

В классовой адресации номер сети и номер узла определяются по принадлежности IP-адреса одному из классов адресов: A, B, C, D или E. Класс определяется значениями первых битов адреса. Например, если первые два бита адреса равны 10, то сеть относится к классу В. В сетях класса В под адрес сети и под адрес узла отводится по 16 бит. В бесклассовой адресации номер сети, к которой принадлежит узел с заданным IP-адресом, определяется другим способом: вместе с IP-адресом предоставляется и маска подсети

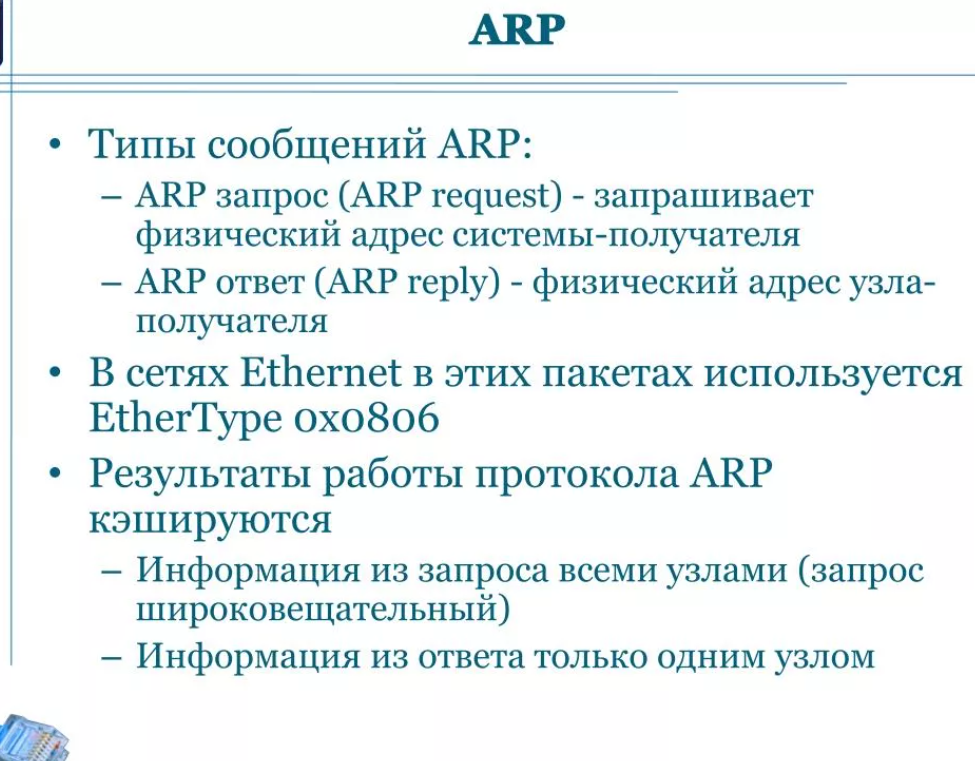
**Отображение физических адресов на IP-адреса происходит с помощью протокола ARP. Функционирование ARP происходит различным образом – в зависимости от того**, какой протокол канального уровня работает в данной сети. В локальных сетях протокол ARP использует широковещательные кадры протокола канального уровня для поиска в сети узла с заданным IP-адресом. Узел, которому нужно выполнить отображение IP-адреса на локальный адрес, формирует ARP-запрос, вкладывает его в кадр протокола канального уровня с указанием в нем известного IP-адреса и осуществляет его широковещательную рассылку по сети. Все узлы локальной сети получают ARP-запрос и сравнивают указанный там IP-адрес с собственным. В случае их совпадения узел формирует ARP-ответ, в котором указывает свои IP- и локальный адреса. ARP-запросы и ответы используют один и тот же формат пакета. Так как локальные адреса могут в различных типах сетей иметь различную длину, то формат пакета протокола ARP зависит от типа сети. Для того чтобы не перегружать сеть запросами, ARP использует таблицу отображения (так называемый ARP-кэш). Эта таблица содержит три поля – IP-адрес, соответствующий ему MAC-адрес и тип. Тип может быть статическим или динамическим. Запись в таблице имеет динамический тип, если она внесена в таблицу путем широковещательного запроса ARP. Такие записи имеют время устаревания (обычно 180 или 360 с), после истечения которого они из таблицы удаляются.

Для объединения подсетей в единую сеть в простейшем случае используется маршрутизация по умолчанию. Она организуется посредством шлюзов. **Шлюзом будем называть узел внутри подсети, который предоставляет доступ в другую подсеть.** Чаще всего в виде шлюза выступает маршрутизатор. Схема такой маршрутизации выглядит следующим образом: задан адрес шлюза по умолчанию. При попытке отправки пакета в сеть, узел проверяет совпадение подсети назначения пакета с подсетью узла. Если подсети разные, то пакет отправляется на шлюз. В простейшем случае шлюз сравнивает сеть IP-адреса назначения с номерами сетей на своих интерфейсах и в случае их совпадения направляет пакет в узел назначения через этот сетевой интерфейс. В противном случае он отправляет пакет в узел, указанный в качестве шлюза по умолчанию на самом шлюзе. Если такового нет, то пакет теряется.

Для проверки соединений и корректного функционирования сети обычно используется протокол **ICMP.** ICMP (Internet Control Message Protocol) – протокол управляющих сообщений интернета. ICMP – протокол сетевого уровня и работает поверх протокола IP. Он предназначен для обмена информацией об ошибках между маршрутизаторами (шлюзами) сети и узлом-источником пакета. С помощью специальных пакетов этот протокол сообщает о невозможности доставки пакета, превышении времени жизни, об аномальных значениях параметров, изменении маршрута пересылки, о состоянии системы и т. п. В простейшем случае, для проверки работоспособности сети используются два сообщения ICMP: Echo-запрос (Echo request) и Echo-ответ (Echo reply). Когда на узел приходит сообщение ICMP типа «Echo-запрос», он отправляет сообщение «Echo-ответ» на тот узел, с которого пришел запрос. Пример реализации такого обмена представлен в утилите ping, входящей в состав почти любой сетевой ОС

**Протокол определения адреса (ARP)** — это протокол или процедура, которая соединяет постоянно изменяющийся адрес интернет-протокола (IP) с фиксированным физическим адресом оборудования, также известным как адрес контроля доступа к медиаинформации (MAC), в локальной сети (LAN).

**ARP** ([англ.](https://ru.wikipedia.org/wiki/%D0%90%D0%BD%D0%B3%D0%BB%D0%B8%D0%B9%D1%81%D0%BA%D0%B8%D0%B9_%D1%8F%D0%B7%D1%8B%D0%BA) *Address Resolution Protocol*



**Маршрутизатор (router)** - это сетевой компьютер связывающий участки локальной сети, который обрабатывает полученные данные по заданным правилам администратора и опираясь на таблицу маршрутизации определяет путь для пересылки данных.

Коммутатор (он же Ethernet Switch или просто «свитч») – это отдельный узел, служащий для объединения нескольких устройств в локальную сеть.

Основная функция *повторителя*(repeater), как это следует из его названия, – повторение сигналов, поступающих на его порт. Повторитель улучшает электрические характеристики сигналов и их синхронность, и за счет этого появляется возможность увеличивать общую длину кабеля между самыми удаленными в сети узлами.

Концентратор – устройство, у которого суммарная пропускная способность входных каналов выше пропускной способности выходного канала. Так как потоки входных данных в концентраторе больше выходного потока, то главной его задачей является концентрация данных. При этом возможны ситуации, когда число блоков данных, поступающее на входы концентратора, превышает его возможности. Тогда концентратор ликвидирует часть этих блоков.

**ICMP** (англ. Internet Control Message Protocol — протокол межсетевых управляющих сообщений) — сетевой протокол, входящий в стек протоколов TCP/IP. В основном ICMP используется для передачи сообщений об ошибках и других исключительных ситуациях, возникших при передаче данных, например, запрашиваемая услуга недоступна или хост, или маршрутизатор не отвечают. Также на ICMP возлагаются некоторые сервисные функции (services).

**Повторитель** (репи́тер, от [англ.](https://ru.wikipedia.org/wiki/%D0%90%D0%BD%D0%B3%D0%BB%D0%B8%D0%B9%D1%81%D0%BA%D0%B8%D0%B9_%D1%8F%D0%B7%D1%8B%D0%BA) *repeater*) — [сетевое оборудование](https://ru.wikipedia.org/wiki/%D0%A1%D0%B5%D1%82%D0%B5%D0%B2%D0%BE%D0%B5_%D0%BE%D0%B1%D0%BE%D1%80%D1%83%D0%B4%D0%BE%D0%B2%D0%B0%D0%BD%D0%B8%D0%B5), предназначенное для увеличения расстояния сетевого соединения и его расширения за пределы одного сегмента или для организации двух ветвей, путём повторения электрического сигнала «один в один». Бывают однопортовые повторители и многопортовые. В терминах [модели OSI](https://ru.wikipedia.org/wiki/%D0%9C%D0%BE%D0%B4%D0%B5%D0%BB%D1%8C_OSI) работает на [физическом уровне](https://ru.wikipedia.org/wiki/%D0%A4%D0%B8%D0%B7%D0%B8%D1%87%D0%B5%D1%81%D0%BA%D0%B8%D0%B9_%D1%81%D0%BB%D0%BE%D0%B9).

Концентратор (hub) – это сетевое устройство, предназначенное для объединения устройств сети в сегменты. Основной принцип его работы заключается в трансляции пакетов, поступающих на один из его портов на все другие порты. Таким образом, пакет, поступивший в сеть, будет отправлен всем остальным устройствам сети, т.е. будет осуществляться широковещательная передача.

МАС адрес (media access control address) – уникальный идентификатор, назначенный сетевому адаптеру, применяется в сетях стандартов IEEE 802, в основном Ethernet, Wi-Fi и Bluetooth.

[**Сетевая**](https://ru.wikipedia.org/wiki/%D0%9A%D0%BE%D0%BC%D0%BF%D1%8C%D1%8E%D1%82%D0%B5%D1%80%D0%BD%D0%B0%D1%8F_%D1%81%D0%B5%D1%82%D1%8C)**модель OSI** (The **Open Systems Interconnection** model**)** — сетевая модель стека (магазина) сетевых протоколов OSI/ISO. Посредством данной модели различные сетевые устройства могут взаимодействовать друг с другом. Модель определяет различные уровни взаимодействия систем. Каждый уровень выполняет определённые функции при таком взаимодействии.

Основное различие между IP-адресом и Межсетевой экран-адресом состоит в том, что **IP-адрес — это логический адрес, характеризующий местоположение вашего устройства в сети Интернет**, тогда как **MAC-адрес — это физический адрес, идентифицирующий фактическое устройство.**

Маска подсети — битовая маска для определения по IP-адресу адреса подсети и адреса узла (хоста, компьютера, устройства) этой подсети. В отличие от IP-адреса маска подсети не является частью IP-пакета.

**Широковещательный канал**, широковещание ([англ.](https://ru.wikipedia.org/wiki/%D0%90%D0%BD%D0%B3%D0%BB%D0%B8%D0%B9%D1%81%D0%BA%D0%B8%D0%B9_%D1%8F%D0%B7%D1%8B%D0%BA) *broadcasting*) — метод [передачи данных](https://ru.wikipedia.org/wiki/%D0%9F%D0%B5%D1%80%D0%B5%D0%B4%D0%B0%D1%87%D0%B0_%D0%B4%D0%B0%D0%BD%D0%BD%D1%8B%D1%85) в [компьютерных сетях](https://ru.wikipedia.org/wiki/%D0%9A%D0%BE%D0%BC%D0%BF%D1%8C%D1%8E%D1%82%D0%B5%D1%80%D0%BD%D0%B0%D1%8F_%D1%81%D0%B5%D1%82%D1%8C), при котором [поток данных](https://ru.wikipedia.org/wiki/%D0%9F%D0%BE%D1%82%D0%BE%D0%BA_%D0%B4%D0%B0%D0%BD%D0%BD%D1%8B%D1%85) (каждый переданный [пакет](https://ru.wikipedia.org/wiki/%D0%9F%D0%B0%D0%BA%D0%B5%D1%82_(%D1%81%D0%B5%D1%82%D0%B5%D0%B2%D1%8B%D0%B5_%D1%82%D0%B5%D1%85%D0%BD%D0%BE%D0%BB%D0%BE%D0%B3%D0%B8%D0%B8)) в случае пакетной передачи) предназначен для приёма всеми участниками сети.

В [TCP/IP](https://ru.wikipedia.org/wiki/TCP/IP) широковещание (*broadcast*) возможно только в пределах одного сегмента сети ([L2](https://ru.wikipedia.org/wiki/%D0%9A%D0%B0%D0%BD%D0%B0%D0%BB%D1%8C%D0%BD%D1%8B%D0%B9_%D1%83%D1%80%D0%BE%D0%B2%D0%B5%D0%BD%D1%8C) или [L3](https://ru.wikipedia.org/wiki/%D0%A1%D0%B5%D1%82%D0%B5%D0%B2%D0%BE%D0%B9_%D1%83%D1%80%D0%BE%D0%B2%D0%B5%D0%BD%D1%8C)). Однако пакеты данных могут быть посланы из-за пределов сегмента, в который будет осуществлено широковещание (например, передача пакета на [широковещательный](https://ru.wikipedia.org/wiki/%D0%A8%D0%B8%D1%80%D0%BE%D0%BA%D0%BE%D0%B2%D0%B5%D1%89%D0%B0%D1%82%D0%B5%D0%BB%D1%8C%D0%BD%D1%8B%D0%B9_%D0%B0%D0%B4%D1%80%D0%B5%D1%81) [IP-адрес](https://ru.wikipedia.org/wiki/IP-%D0%B0%D0%B4%D1%80%D0%B5%D1%81) через [маршрутизатор](https://ru.wikipedia.org/wiki/%D0%9C%D0%B0%D1%80%D1%88%D1%80%D1%83%D1%82%D0%B8%D0%B7%D0%B0%D1%82%D0%BE%D1%80) из-за пределов сети). Нагрузка на сеть в случае широковещания не отличается от обычной передачи данных одному адресату, поскольку пакеты данных не размножаются (в отличие от *[unicast](https://ru.wikipedia.org/wiki/Unicast" \o "Unicast)*).

Примером широковещания является определение [MAC-адреса](https://ru.wikipedia.org/wiki/MAC-%D0%B0%D0%B4%D1%80%D0%B5%D1%81), соответствующего определённому IP-адресу (например, с помощью протокола [ARP](https://ru.wikipedia.org/wiki/ARP)). В этом случае отправляется широковещательный пакет с запросом, который достигает все подключенные к данному [**L2-домену**](https://ru.wikipedia.org/wiki/%D0%A8%D0%B8%D1%80%D0%BE%D0%BA%D0%BE%D0%B2%D0%B5%D1%89%D0%B0%D1%82%D0%B5%D0%BB%D1%8C%D0%BD%D1%8B%D0%B9_%D0%B4%D0%BE%D0%BC%D0%B5%D0%BD) сети устройства. Устройство с искомым IP-адресом отправляет в ответ пакет, содержащий требуемый MAC-адрес.

Широковещательным IP адресом является последний адрес в подсети. Если сеть состоит из одного адреса /32, то она не имеет широковещательного адреса.

Лабораторная 2

Публичный (белый) IP-адрес - - тот адрес, под которым компьютер или другое устройство видно и работает в Интернете. И из любой точки земли, где он есть, этот IP будет доступен.

Частный (серый) IP-адрес используется во внутренних локальных сетях. С ним компьютер будет доступен с любого другого из этой сети, но из Интернета его видно не будет.

3. МАРШРУТИЗАЦИЯ

**Маршрутизация – важнейший процесс в IP-сетях. Чтобы некоторая машина могла найти в сети другую, должен иметься механизм описания того,**

**как пакеты должны передаваться от одной машины к другой. Такой механизм и называется маршрутизацией.** Более строгое определение звучит так:

**Маршрутизация – процесс определения маршрута следования пакетов данных в компьютерных сетях**. Маршрутизация выполняется специальными

программными или аппаратными средствами – маршрутизаторами.

**Маршрутизатор – сетевое устройство, используемое в компьютерных**

**сетях передачи данных, которое на основании информации о топологии сети**

**(таблицы маршрутизации) и определенных правил принимает решения о пересылке пакетов сетевого уровня их получателю**.

Маршрутизация, осуществляемая IP, – это процесс поиска в таблице маршрутизации, определение интерфейса, куда будет послан пакет. Существует два типа маршрутизации:

статическая и динамическая.

**Статическая маршрутизация осуществляется на основе таблиц маршрутизации, задаваемых администратором. Динамическая маршрутизация осуществляется с помощью протоколов маршрутизации. Протокол маршрутизации – это сетевой протокол, используемый маршрутизаторами для определения возможных маршрутов следования данных в составной компьютерной**

**сети**.

Применение протокола маршрутизации позволяет избежать ручного

ввода всех допустимых маршрутов, что, в свою очередь, снижает количество

ошибок, обеспечивает согласованность действий всех маршрутизаторов в

сети и облегчает труд администраторов.

**3.1. Принцип статической маршрутизации**

Далее рассматривается статическая маршрутизация в IP-сетях. Частично

эта тема уже затрагивалась при рассмотрении простейшей маршрутизации на

основе шлюзов по умолчанию. Таблица маршрутизации содержит информацию, на основе которой маршрутизатор принимает решение о дальнейшей

пересылке пакетов. Как только IP-пакет попадает на маршрутизатор (или на

любой узел сети, поддерживающий сетевой уровень модели TCP/IP), выполняется последовательность действий:

1. Маршрутизатор определяет, является ли узел-получатель пакета локальным (т.е. находится ли он в той же подсети, что и маршрутизатор). Маршрутизатор имеет несколько интерфейсов и может одновременно находиться

в нескольких подсетях. Если получатель находится в той же подсети, то пакет направляется ему напрямую.

2. Если узел-получатель находится в другой подсети, то просматривается таблица маршрутизации на предмет пути к этому узлу. При просмотре

сеть узла-получателя сравнивается с записями о сетях в таблице маршрутизации и при обнаружении совпадения пакет направляется маршрутизатору,

указанному в соответствующей записи.

3. Если маршрута в таблице маршрутизации не найдено, то пакет отправляется шлюзу по умолчанию.

4. Если запись о шлюзе по умолчанию отсутствует, то пакет уничтожается.

Для указания шлюза по умолчанию в таблице существует специальная

запись default, которая указывает, на какой узел должен быть направлен пакет, если сеть его назначения не определена. Как правило, в локальной сети

присутствует один шлюз, используемый по умолчанию.

**3.1.1. Таблицы маршрутизации**

**Таблица — маршрутизации-это база данных, которая отслеживает пути, как карта, и использует их, чтобы определить, в какую сторону перенаправлять трафик.**

Чтобы по адресу сети назначения можно было бы выбрать маршрут

дальнейшей пересылки пакета, каждый узел, поддерживающий сетевой уровень, анализирует специальную информационную структуру, называемую

таблицей маршрутизации.

27

Простейшая таблица маршрутизации включает в себя информацию об

узле (или о сети) назначения, о маске подсети для узла (сети) назначения, об

интерфейсе, через который следует направить пакет, и о шлюзе – удаленном

узле, которому будет передан пакет.

Рассмотрим на примере, как может выглядеть таблица маршрутизации.

Пусть задана конфигурация сети, показанная на рис. 3.1.

Рис. 3.1

Записи о статических маршрутах маршрутизатора R2 приведены в таблице 3.1.

Таблица 3.1

Destination Gateway Genmask Iface

default 11.0.0.1 255.0.0.0 eth0

14.0.0.0 12.0.0.2 255.0.0.0 eth1

В первой колонке перечисляются номера подсетей, во второй – какому

маршрутизатору следует перенаправить пакет для отправки в заданную подсеть. Третья колонка задает маску подсети назначения, в четвертой колонке

указывается, через какой интерфейс следует направить пакет.

**3.2. Протоколы динамической маршрутизации**

Протоколы динамической маршрутизации делятся на протоколы **внутридоменной маршрутизации – IGP (Interior Gateway Protocol) – и протоколы**

**междоменной маршрутизации – EGP (Exterior Gateway Protocol).**

IGP-протоколы используются для передачи информации о маршрутах в пределах автономной системы (домена маршрутизации); пример автономной системы –сеть одной компании.

EGP-протоколы используются для соединения автономных систем между собой.

**IGP-протоколы подразделяются на дистанционно-векторные протоколы (RIP, EIGRP) и протоколы состояния каналов связи (OSPF, IS-IS).** Первые основаны на алгоритме DVA (distance vector algorithm), вторые – на ал-

28

горитме LSA (link state algorithm). Основные различия между этими двумя

видами следующие:

1) Тип информации, которой обмениваются роутеры: таблицы маршрутизации в DVA-протоколах и таблицы топологии в LSA-протоколах.

2) Процесс выбора лучшего маршрута.

3) Количество информации о сети, хранящееся в каждом роутере: при

DVA роутер знает только своих соседей, при LSA – имеет представление обо

всей сети.

Из EGP-протоколов в настоящее время активно используется один –

BGP. Предназначенный для оперирования большими объемами данных, он

обеспечивает связность всего Интернета, но может применяться и внутри

домена.

Протоколы динамической маршрутизации не просто автоматизируют

настройку маршрутов в сети, но позволяют выполнять быстрое автоматическое их перестроение при возникновении проблем на участках сети. Еще

одна полезная функция – балансировка трафика.

Подробнее об алгоритмах динамической маршрутизации можно прочитать в [10], о протоколах RIP, OSPF и BGP – в [16].

*Маршрутизатор — это устройство, которое создает локальную сеть, принимает данные от интернет-провайдера и передает их конечному потребителю по кабелю или при помощи беспроводной технологии Wi-Fi.*

Как работает динамическая маршрутизация:

* + 1. = прямое подключение к сети

Правило выборов приоритета:

1)подхождение Ip адреса по маске

2)Чем больше маска, тем приоритет больше

3)если маски одинаковые, то используем метрику. Чем ниже метрика – выше приоритет. Метрика – сколько трудозатрат на передачу пакета через интерфейс

# Типы записей в таблице маршрутизации

[⇐ Предыдущая](https://infopedia.su/10x644c.html)**Стр 7 из 7**

Каждая запись в таблице маршрутизации считается маршрутом и может иметь один из следующих типов.

1. Маршрут к сети.

Такой маршрут ведет к сети, входящей в объединенную сеть и имеющей указанный код (идентификатор).

2. Маршрут к узлу.

Такой маршрут ведет к конкретному узлу в объединенной сети, обладающему указанным адресом (кодом сети и кодом узла). Маршруты к узлу обычно используются для создания пользовательских маршрутов к отдельным узлам с целью оптимизации или контроля сетевого трафика.

3. Маршрут по умолчанию.

Маршрут по умолчанию используется, если в таблице маршрутизации не были найдены подходящие маршруты. Например, если маршрутизатор или узел не могут найти нужный маршрут к сети или маршрут к узлу, то используется маршрут по умолчанию. Маршрут по умолчанию упрощает настройку узлов. Вместо того чтобы настраивать на узлах маршруты ко всем сетям объединенной сети, используется один маршрут по умолчанию для перенаправления всех пакетов с адресами сетей, не обнаруженными в таблице маршрутизации.

Лабораторная 34

ifconfig означает “конфигурация интерфейса”. Это позволяет нам просматривать и настраивать параметры сетевого интерфейса.

Интерфейс - это точка соединения между компьютером и сетью.

Настройка компьютера для установки виртуальных машин.

Введем следующие понятия:

**1. Хост – компьютер, на котором устанавливаются виртуальные машины.**

**2. Виртуальная машина – программная система, эмулирующая аппаратное обеспечение некоторой платформы (гостевая платформа) и исполняющая программы для гостевой платформы, установленная на хосте.**

При выполнении лабораторных работ будет использоваться программный продукт Oracle VirtualBox. Допускается использовать другую систему для

запуска виртуальных машин, но она должна отвечать следующим основным

требованиям:

1) поддерживаются типы подключений интерфейсов: NAT, внутренняя

сеть (internal network);

2) поддерживается возможность клонирования виртуальной машины.

По умолчанию в лабораторных работах рассматриваются ОС с разрядностью x64, однако если хостовый компьютер недостаточно мощный или количество оперативной памяти меньше 3 Гб, тогда можно использовать ОС с разрядностью x86 – на выполнении лабораторных работ это не отразится.

На хосте необходимо включить поддержку виртуализации. Выполнить это необходимо в BIOS, причем в каждой версии BIOS настройка своя. На процессорах Intel при входе в BIOS необходимо зайти в пункт настройки «CPU Configuration». Этот раздел может находиться в «Advanced» либо в «Integrated Peripherals». Внутри настроек CPU должна быть строчка, похожая на «Intel 6 Virtualization Technology». Ее необходимо активировать либо выбрать «Enable».

На процессорах AMD в BIOS необходимо выполнить настройки в «Advanced» в «CPU Configuration». Там необходимо настроить пункт «SVM Mode», включив его («Enabled»). После этого необходимо сохранить изменения и перезагрузить компьютер. При установке виртуальной машины разрядности x64 может появиться ошибка 0x80004005 (либо другая) с информацией о проблемах, связанных с виртуализацией, это означает, что были выполнены не все настройки в BIOS.

В таком случае необходимо найти соответствующие пункты, связанные с использованием виртуальных машин. Для поиска можно, зная версию BIOS, найти в Интернете информацию о процессе настройки виртуализации. Если настроить данный функционал не получится, допускается использовать ОС Ubuntu Server x86. Важно: на компьютерах в компьютерном классе необходимые настройки BIOS выполнены в полном объеме.

В ОС Ubuntu LTS 18.04 и более новых версиях для работы с сетью используется network-manager, лабораторные работы будут выполняться с использованием ifconfig и /etc/network/interfaces. Данные утилиты используются, например, в ОС Ubuntu 16 и Ubuntu 14. Лабораторные работы подготовлены на базе ОС Ubuntu LTS 16.04, однако может быть использована и более старая версия ОС.

Лабораторные работы выполняются в терминальном режиме, поэтому достаточно устанавливать только серверную версию ОС. Рекомендуемый объем оперативной памяти – 512 Мб, но если объема оперативной памяти на хосте недостаточно, то после установки виртуальной машины можно установить 256 Мб.

В процессе создания виртуальной машины необходимо настроить количество используемых центральных процессоров, достаточно выбрать один на каждую виртуальную машину. Также необходимо настроить виртуальный жесткий диск. Рекомендуется выбрать VDI (Virtual Disk Image), динамический жесткий диск с объемом 10 Гб и выше.

В качестве типа подключения сетевого интерфейса для начальной настройки достаточно выбрать NAT или сетевой мост. Это позволит виртуальной машине иметь доступ в Интернет для настройки и скачивания необходимых приложений.

Дополнительно для удобства доступа к терминалу виртуальной машины при наличии технической возможности следует использовать тип подключения «сетевой мост». Это позволит виртуальной машине получить IP-адрес из подсети, которая является общей с хостом. Установив на хост программу Putty, можно подключиться по ssh к виртуальной машине, указав ее IP-адрес в настройках подключения Putty.

**Узнать IP-адрес можно, выполнив на машине команду ifconfig.**

Если у вас нет возможности получить IP-адрес из той же подсети, то можно настроить два сетевых интерфейса: первый с NAT для выхода в Интернет, второй – виртуальный адаптер хоста (VirtualBox Host-Only Ethernet Adapter). Благодаря ему будет использоваться виртуальная локальная сеть между хостом и виртуальной машиной и она будет доступна по IP-адресу из

этой подсети к подключению по ssh.

**OpenSSH** - это мощный набор инструментов для удаленного управления и передачи данных между сетевыми компьютерами. Вы также узнаете о некоторых параметрах конфигурации, возможных с серверным приложением OpenSSH, и о том, как их изменить в вашей системе Ubuntu.

OpenSSH - это свободно доступная версия семейства инструментов протокола Secure Shell (SSH) для удаленного управления или передачи файлов между компьютерами.

**SSH (от англ. secure shell ― безопасная оболочка) ― это защищённый сетевой протокол для удалённого управления сервером через интернет.**

Настройка ОС Ubuntu после установки. После установки ОС Ubuntu следует настроить необходимый функционал.

Нужно снизить время ожидания получения сетевых настроек системой в период загрузки ОС. При неверных настройках или их отсутствии (по умолчанию) ОС будет запускаться несколько минут. Чтобы уменьшить время запуска, необходимо выполнить в терминале следующие команды:

sudo mkdir -p /etc/systemd/system/networking.service.d/

sudo bash -c 'echo -e "[Service]\nTimeoutStartSec=20sec" >

/etc/systemd/system/networking.service.d/timeout.conf'

sudo systemctl daemon-reload

Для доступа в систему по ssh (например, используя Putty) необходимо установить на виртуальной машине ssh-сервер:

sudo apt-get update

sudo apt-get install openssh-server

Также стоит отметить, что для применения настроек сети необходимо перезагружать виртуальную машину, так как перезагрузка сетевого сервиса поможет не во всех случаях. Выполнить это можно командой sudo reboot, после чего через 20…30 с установить новое ssh-подключение с хоста на виртуальную машину.

После выполнения перечисленных действий виртуальная машина будет готова к использованию в следующих лабораторных работах. Для их выполнения можно осуществить клонирование настроенной виртуальной машины.

Для этого необходимо выключить виртуальную машину и в меню VirtualBox

найти пункт «Клонировать». В процессе клонирования необходимо задать новое имя, в политике MAC-адресов рекомендуется выбрать «Сгенерировать новые MAC-адреса для всех сетевых адаптеров». На следующем шаге нужно выбрать «полное клонирование». Такие настройки создадут отдельную независимую копию виртуальной машины. В результате появится новая машина, которая будет полностью идентична исходной виртуальной машине и может быть использована.

**Понятие IP-адреса**.

**IP-адрес (Internet Protocol Address)** – уникальный адрес для обозначения устройств, например персональных ПК, мобильных телефонов, серверов, служащий для идентификации устройства в сети и общения с другими устройствами в сети.

Каждое устройство, подключенное к сети Интернет, должно иметь уникальный IP-адрес внутри сети. Другими словами, IP-адрес аналогичен домашнему адресу и телефонному номеру, с помощью которых можно однозначно определить человека.

В рамках данного пособия будет изучен IPv4. Он использует 32-битное число для представления IP-адреса.

32 бита дают возможность получить около 4 млрд уникальных адресов, поэтому нет возможности выделить всем отдельные IP-адреса внутри одной сети. Для решения данной проблемы служит трансляция IP-адресов (NAT, Masquerade). Также существует версия протокола IPv6, которая может предоставить значительно большее количество уникальных адресов. Обычно IP-адрес представляется в десятичном виде из четырех наборов чисел, каждый из которых хранит в себе восьмибитное число. Так как число восьмибитное, то в десятичной системе данное значение может быть от 0 до 255.

Маска подсети. Сетевые устройства логически объединяются в подсети. Каждому сетевому устройству назначаются IP-адрес и маска подсети (netmask). Маска подсети в IPv4 состоит из 32 бит, непрерывной последовательности единиц (1), за которой следует непрерывная последовательность нулей (0). В маске подсети не может стоять единица после нуля. Маска подсети служит для разделения IP-адресов на подсети и определения того, какие из них будут принадлежать каждой подсети.

IP-адрес делится с помощью маски на адрес подсети (префикс сети) и адрес узла. Рассмотрим пример: IP-адрес: 192.168.0.1 Маска: 255.255.255.0

CIDR - **Classless Inter-Domain Routing это гибкий метод IP-адресации в компьютерных сетях, который позволяет экономить адресное пространство, данный метод снимает ограничение классовых сетей, которое заключалось в том, что конкретный IP-адрес принадлежал конкретному классу, а это означало, что этому IP-адресу соответствовала маска фиксированной длины, строго закрепленная за классом.**

TCP/IP - это набор протоколов, который задает стандарты связи между компьютерами и содержит подробные соглашения о маршрутизации и межсетевом взаимодействии.

Поэтому ее можно записывать приставкой к IP-адресу, например,: 192.168.0.1/24. В данном случае 24 – это количество единиц в маске. Такая запись называется CIDR и может принимать значения от 1 до 32, по количеству единиц в 32-битном значении маски.

Для вычисления адреса сети по адресу узла необходимо перевести IP-адрес и маску в двоичную систему, например: 192.168.0.1 ◊ 11000000.10101000.00000000.00000001 255.255.255.0 ◊ 11111111.11111111.11111111.00000000

**Сетевой префикс (адрес сети) вычисляется побитовой операцией AND между IP-адресом и маской.**

В приведенном примере это первые три четверти записи IP-адреса, поэтому префикс подсети – 192.168.0.0. Адресом узла является оставшаяся часть, которая соответствует нулям маски – 0.0.0.1. Принадлежность IP-адресов одной подсети.

Для того, чтобы выяснить, **принадлежат ли два IP-адреса одной подсети**, необходимо выполнить следующее. Во-первых, их маски должны быть одинаковы. Во-вторых, нужно перевести IP-адреса и маску в двоичный вид: 253.45.78.14 ◊ 11111101.00101101.01 | 001110.00001110 253.45.126.14 ◊ 11111101.00101101.01 | 111110.00001110 255.255.192.0 ◊ 11111111.11111111.11 | 000000.00000000

Применив к каждому IP-адресу и маске побитовое AND, мы получаем адреса подсети для каждого IP-адреса. Если они совпадают, это значит, что эти два IP-адреса принадлежат одной подсети (как в варианте выше).

В случае, если в результате адреса подсети разные – IP-адреса принадлежат разным подсетям: 156.14.32.78 ◊ 10011100.00001110.00100000.01 | 001110 156.14.32.130 ◊ 10011100.00001110.00100000.10 | 000010 255.255.155.192 ◊ 11111111.11111111.11111111.11 | 000000

Если настроить IP-адреса на два ПК и физически соединить их (коммутатор, напрямую кабелем, объединить в одну сеть в VirtualBox логически и др.), то проверить принадлежность их к общей сети можно с помощью выполнения следующей команды в терминале: ping 11 Здесь в качестве подставляется IP-адрес второго ПК. Если ответ приходит – они в одной подсети, если нет – в разных.

Пример успешного ответа: 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=45 time=4.55 ms 64 bytes from 8.8.8.8: icmp\_seq=2 ttl=45 time=4.57 ms 64 bytes from 8.8.8.8: icmp\_seq=3 ttl=45 time=4.57 ms Пример неуспешного ответа: PING 192.168.254.30 (192.168.254.30) 56(84) bytesofdata. From 192.168.76.1 icmp\_seq=1 Destination Host Unreachable From 192.168.76.1 icmp\_seq=2 Destination Host Unreachable From 192.168.76.1 icmp\_seq=3 Destination Host Unreachable Также система может выдать сообщение «Network is unreachable». Это произойдет в случае, если местонахождение узла неизвестно.

Широковещательный IP-адрес.

**Широковещательный IP-адрес (broadcast**) нужен для того, чтобы отправить данные всем узлам подсети. Например, протокол ARP, который связывает MAC-адрес устройства и IP-адрес, использует для этого широковещательный IP-адрес. Для того, чтобы получить широковещательный IP-адрес, необходимо «перевернуть» значения маски и применить побитовое OR между IP-адресом и маской:

Отправляя пакет с указанным IP-адресом, система обеспечит его получение всеми узлами, находящимися в данной подсети.

**Зарезервированные IP-адреса**. Есть пул IP-адресов, которые назначать сетевому устройству нельзя: 0.0.0.0/8 (если первый октет «0»), 127.0.0.0/8 (интерфейс loopback), 224.0.0.0/4 (служебная подсеть для групповой рассылки), 240.0.0.0/4 (резервные IP-адреса) и 255.255.255.255 (широковещательный IPадрес). Если настроить IP-адрес из этих подсетей, то сетевое устройство может работать неверно или вообще быть недоступным.

Файл /etc/network/interfaces является основным файлом настроек сетевых интерфейсов

Лабораторная 5

**iptables** — утилита командной строки, является стандартным интерфейсом управления работой межсетевого экрана (брандмауэра) netfilter для ядер Linux, начиная с версии 2.4. Для использования утилиты iptables требуются привилегии суперпользователя (root).

**Межсетево́й экра́н, сетево́й экра́н** — программный или программно-аппаратный элемент компьютерной сети, осуществляющий контроль и фильтрацию проходящего через него сетевого трафика в соответствии с заданными правилами.

Необходимость использования механизмов переадресации.

На рис. 3.1 представлена общая схема сети, в которой используются механизмы переадресации. Применительно к цели данной лабораторной работы основным элементом является частная сеть, включающая компьютер 1 и маршрутизатор 2, а вспомогательными – DNS-сервер google 8.8.8.8 (на рисунке отмечен номером 3) и частная сеть № 2 c компьютером 4. Маршрутизатор 2 включает в себя частную подсеть № 1 и выход в сеть Интернет. Он (при правильной настройке) является своего рода «проводником» в Интернет для частной сети № 1. Данная лабораторная посвящена процессу настройки и работы данного маршрутизатора. Существует несколько вариантов подключения компьютера 1 к сети Интернет. 15 eth1: 10.238.22.14 Интернет 2 3 8.8.8.8 3 Частная сеть № 2 eth0: 192.168.1.2 4 eth0: 192.168.0.2 eth0: 192.168.0.1 1 Рис. 3.1. Общая схема сети

**Механизм Маскарад.**

**Маскарад** - замена адреса на адрес машины, выполняющей маскарад.

Данный механизм настраивается на маршрутиаторе 2 и работает следующим образом: всем пакетам при прохождении через eth1 маршрутизатора 2 присваивается новый IP-адрес отправителя, который подключен к внешней сети (в примере это eth1: 10.238.22.14).

Следует отметить, что частная сеть не видна из сети Интернет, как и сеть Интернет не видна из частной сети № 1.

При использовании «маскарадинга» (masquerade) узлы частной сети могут общаться с внешней сетью (например, Интернет) благодаря подмене IP-адресов.

В указанном примере IP-адрес 10.238.22.14 доступен в сети Интернет. Это позволяет запрашивать с него любую информацию, находящуюся в сети Интернет. Маскарадинг позволяет компьютерам частной сети передать запрос и получить ответ из сети Интернет. Рис. 3.2 отражает данный процесс при обращении ПК из сети № 1 к DNS-серверу 8.8.8.8. 16

Данный процесс состоит из восьми шагов. На шаге 1 создается пакет от ПК 192.168.0.2 к 8.8.8.8. Далее пакет присылается на интерфейс eth0 маршрутизатора (шаг 2). На маршрутизаторе настроен Masquerade, меняющий IP-адрес отправителя на свой внешний интерфейс (в примере 10.238.22.14) при прохождении пакета во внешнюю сеть (шаг 3). На шаге 4 измененный пакет доставляется на узел с IP-адресом 8.8.8.8. На шаге 5 пакет обрабатывается и узел отвечает, указывая адрес доставки 10.238.22.14. Данный адрес доступен в сети Интернет, поэтому пакет на шаге 6 доходит до маршрутизатора. На шаге 7 с помощью Masquerade пакет видоизменяется обратно, чтобы дойти до конечного узла, который выполнил запрос (192.168.0.2). На шаге 8 измененный пакет приходит к начальному узлу, который отправил запрос на шаге 1. Далее представлен пример настройки Masquerade с помощью iptables. В качестве маршрутизатора может выступать любой узел, подключенный к частной сети № 1 и сети Интернет в ОС Ubuntu:

iptables -t nat -A POSTROUTING -o enp0s8 -j MASQUERADE

Данное правило меняет всем пакетам, проходящим через интерфейс enp0s8, IP-адрес источника на IP-адрес интерфейса enp0s8.

**Механизм NAT.**

**NAT (от англ. Network Address Translation — «преобразование сетевых адресов») — это механизм в сетях TCP/IP, позволяющий преобразовывать IP-адреса транзитных пакетов.**

NAT похож на Masquerade.

Разница в том, что IP-адрес отправителя подменяется не на IP-адрес устройства, через который он проходит, а на специальный IP-адрес, заданный в процессе настройки данного устройства.

В зависимости от необходимости используются различные типы NAT.

С технической точки зрения они сводятся к настройке **sNAT (source NAT – замена IP-адреса источника)** и dNAT **(destination NAT – замена IP-адреса назначения)**.

**sNAT** позволяет внутренним узлам частной сети общаться с внешними сетями (например, сетью Интернет).

**dNAT** позволяет из внешней сети успешно отправить данные на узел внутренней сети.

Следует отметить, что, подключая компьютер к сети Интернет через WiFi или другой маршрутизатор, используют одну из перечисленных выше технологий для организации доступа в сеть Интернет с домашних компьютеров.

При этом зачастую это уже преднастроено в маршрутизаторах, и ручная настройка не нужна, так как используется Masquerade. Если же требуется, чтобы компьютер имел конкретный IP-адрес, то необходимо дополнительно настроить маршрутизатор (sNAT, dNAT).

Рассмотрим настройку sNAT и dNAT.

Для этого на узле, который имеет выход во внешнюю сеть, необходимо настроить **вторичный IP-адрес.** Он будет связующим звеном между частной и внешней сетями.

Сначала настроим на узле маршрутизации **sNAT**: 17 iptables -t nat -A POSTROUTING -s 192.168.0.2/32 -o enp0s3 -j sNAT--to-source 10.144.2.100 Данное правило означает, что в цепочке NAT после обработки пакета для всех пакетов, IP-адрес источника которых равен 192.168.0.2 (т. е. они были отправлены с этого узла), будет происходить его смена на 10.144.2.100 (IP-адрес, доступный во внешней сети). Благодаря этому правилу пакет, отправленный из частной сети, сможет дойти до необходимого узла во внешней сети и получить ответ.

Чтобы из внешней сети (например, сети Интернет) можно было получить доступ к узлу в частной сети, необходимо настроить **dNAT**: iptables -t nat -A PREROUTING -d 10.144.2.100 -j DNAT --to-destination 192.168.0.2 Данное правило означает, что если из внешней («публичной») сети пакет будет отправлен на 10.144.2.100, то при прохождении через узел, на котором это правило настроено, произойдет подмена IP-адреса назначения, и пакет дойдет до требуемого узла в частной сети с IP-адресом 192.168.0.2.

**Цепочки**

**Таблица nat содержит следующие цепочки:**

•PREROUTING — в эту цепочку пакеты попадают до принятия решения о маршрутизации. По сути, термин «решение о маршрутизации» подразумевает деление трафика на входящий (предназначенный самому хосту) и транзитный (идущий через этот хост на другие хосты). Именно на данном этапе нужно проводить операции проброса (DNAT, REDIRECT, NETMAP).

•OUTPUT — через эту цепочку проходят пакеты, сгенерированные процессами самого хоста. На данном этапе при необходимости можно повторить операции проброса, так локально сгенерированные пакеты не проходят цепочку PREROUTING и не обрабатываются ее правилами. См. пример для действия DNAT ниже.

•POSTROUTING — через эту цепочку проходят все исходящие пакеты, поэтому именно в ней целесообразно проводить операции маскарадинга (SNAT и MASQUERADE).

**Что в таблице указано nat**

и типо можно ли в nat на один адрес назначить несколько адессов

адрес источник и подстановка, могу быть также порты

**Типы трансляции сетевых адресов.**

Процесс сопоставления IP-адресов при трансляции обычно проходит по двум принципам: «один к одному»; «многие к одному».

При трансляции **«один к одному»** происходит сопоставление одного внутреннего IP к одному внешнему.

Например, локальный адрес компьютера 1 на рис. 3.1 (192.168.0.2) будет сопоставляться с глобальным адресом NATмаршрутизатора (10.238.22.14). Если в сети № 1 появится еще один компьютер (например, компьютер 5 с адресом 192.168.0.5), которому нужен доступ в Интернет, то на NAT-маршрутизаторе потребуется еще один глобальный адрес (например, 10.238.22.15).

Трансляция адресов «один к одному» может быть:

• статической, определяемой администратором;

• динамической, когда у NAT-маршрутизатора есть пул глобальных адресов, автоматически используемых для сопоставления локальным адресам по мере необходимости.

При трансляции **«многие к одному»** происходит сопоставление не только IP-адресов, но и портов, либо другой информации, когда протокол не использует портов для отправки сообщений.

Например, пусть компьютеры 1 (192.168.0.2) и 5 (192.168.0.5) обращаются к одному и тому же сервису в Интернете (8.8.8.8), а компьютер 1 – еще и к узлу с глобальным адресом 10.239.12.222 и портом 77. На NATмаршрутизаторе может быть создана таблица. 18

* Локальный адрес-порт
* Глобальный адрес-порт
* NAT-маршрутизатора
* Глобальный адрес-порт узла в Интернете

192.168.0.2:15003 10.238.22.14:1101 8.8.8.8:53 192.168.0.5:15003 10.238.22.14:1102 8.8.8.8:53 192.168.0.2:8877 10.238.22.14:1103 10.239.12.222:77

Как видно из таблицы, при таком принципе работы NAT сопоставляет и транслирует не только IP-адрес, но и порт.

Это делается для того, чтобы избежать случаев, когда с двух сетевых устройств одновременно отправляются пакеты с использованием одного порта (см. первую и вторую строки таблицы).

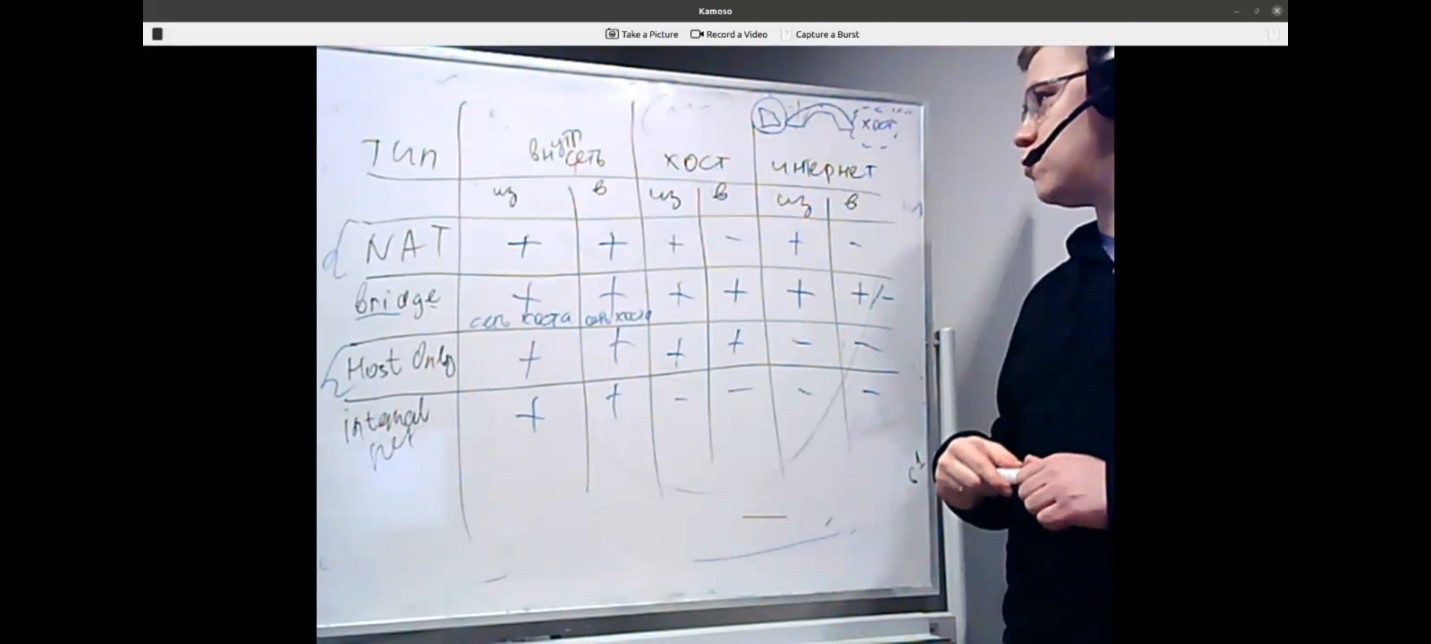
Подобная реализация позволяет динамически обеспечить общение с нескольких IP-адресов, используя один внешний IP-адрес. Такой принцип использует Masquerade. Также с помощью принципа «многие к одному» можно настроить доступ извне локальной сети к двум ее сервисам по одному IPадресу, настроив трансляцию по специально заданным вручную портам. Это позволит лучше использовать пространство белых IP-адресов и не покупать дополнительные.

Отслеживание смены IP-адресов в заголовках (Masquerade). Для наглядной демонстрации работы данных механизмов рассмотрим примеры настроек: root@ub-nat:/home/user# tcpdump -p icmp –i eth0 17:09:10.828249 IP 192.168.0.2> 8.8.8.8: ICMP echo request, id 2407, seq 1, length 64 17:09:10.906169 IP 8.8.8.8 >192.168.0.2: ICMP echo reply, id 2407, seq 1, length 64 При просмотре трафика, проходящего через интерфейс eth0 маршрутизатора 2 (рис. 3.2), мы видим в заголовке пакета IP-адрес источника в частной сети. Тот же самый трафик, прошедший через интерфейс eth1 маршрутизатора 2, имеет следующий вид: root@ub-nat:/home/user# tcpdump -p icmp -i enp0s3 17:09:10.828251 IP 10.238.22.14 > 8.8.8.8: ICMP echo request, id 2418, seq 1, length 64 17:13:20.906157 IP 8.8.8.8 > 10.238.22.14: ICMP echo reply, id 2418, seq 1, length 64 В примере в заголовке пакета изменен IP-адрес отправителя на IP-адрес внешнего интерфейса маршрутизатора 2. Узел 8.8.8.8, получив пакет, отправляет ответ на IP-адрес доступного узла (10.238.22.14), который далее транслирует его в частную сеть. Перед началом выполнения лабораторной работы стоит отметить, что реализация технологии NAT на практике у вендоров сетевых устройств может быть различной. Нет одного конкретного алгоритма, который реализуется на 19 практике. В данной лабораторной работе представлены основные принципы технологии трансляции IP-адресов, однако на практике могут быть различные нюансы, например добавление ACL (контроль доступа) в процесс трансляции либо урезание некоторого функционала. Построение инфраструктуры для выполнения работы. В условиях наличия только одного компьютера реализация данной задачи сводится, например, к построению подобной инфраструктуры в Oracle VirtualBox (рис. 3.3). Гипервизор VirtualBox Сеть Интернет Сеть Intranet1 10.0.1.0/24 Виртуальный NAT: 10.0.2.2 Сеть Intranet2 10.0.0.0/24 Eth0: 10.0.1.2 Eth0: 10.0.0.3 ub1 ub2 ub-nat 8.8.8.8 Eth0: 10.0.1.1 Eth1: 10.0.0.1 Eth2: 10.0.2.15 NAT-сеть Рис. 3.3. Топология построения инфраструктуры в VirtualBox Перед настройкой сетевых интерфейсов для ускорения загрузки виртуальных машин следует выполнить следующее: sudo mkdir -p /etc/systemd/system/networking.service.d/ sudo bash -c 'echo -e "[Service]\nTimeoutStartSec=20sec" > /etc/systemd/system/networking.service.d/timeout.conf' sudo systemctl daemon-reload

Для создания данной топологии необходимо использовать следующие типы подключения интерфейсов в VirtualBox: • Внутренняя сеть (Intranet1 и Intranet2). Внутренняя сеть, согласно руководству VirtualBox, является «программной сетью, которая может быть видима для выборочно установленных виртуальных машин, но не для приложений, работающих на хосте или на удаленных машинах, расположенных извне». 20 Такая сеть представляет собой набор из хоста и нескольких виртуальных машин. Но ни одно из вышеперечисленных устройств не имеет выхода через физический сетевой адаптер – он полностью программный, используемый VirtualBox в качестве сетевого маршрутизатора. В целом получается частная локальная сеть только для гостевых операционных систем без доступа в Интернет.

**• Трансляция сетевых адресов (NAT).** Протокол NAT позволяет гостевой операционной системе выходить в Интернет, используя при этом частный IP, который недоступен со стороны внешней сети или же для всех машин локальной физической сети. Такая сетевая настройка позволяет посещать web-страницы, скачивать файлы, просматривать электронную почту. И все это – используя гостевую операционную систему. Однако извне невозможно напрямую соединиться с такой системой, если она использует NAT.

Можно провести аналогию с настройкой механизма sNAT, представленного ранее. В качестве маршрутизатора будет выступать виртуальная машина «ub-nat», которая будет иметь выход в сеть Интернет посредством NAT-сети, а также подключена к двум внутренним сетям Intranet1 и Intranet2. Для обеспечения возможности переадресации трафика между интерфейсами внутри «ub-nat» необходимо включить данную опцию в sysctl. Для этого необходимо в файле /etc/sysctl.conf задать следующую переменную: net.ipv4.ip\_forward = 1 После этого следует перезагрузить «ub-nat», чтобы применить настройки. Однако отличие построенной инфраструктуры в VirtualBox (рис. 3.3) от общего примера (рис. 3.2) состоит в том, что внешняя частная сеть № 2 в VirtualBox является внутренней и «маршрутизатор ub-nat» напрямую подключен к данной сети – маршрут до частной сети № 2 ему «известен». В общем случае местоположение сети Intranet2 неизвестно. Для того, чтобы воссоздать подобные условия на одной из машин ub1 или ub2, необходимо закрыть прямой доступ в соседнюю внутреннюю сеть, например, для ub1: root@ub1:/home/user# iptables –A OUTPUT -d 10.0.0.0/24 –j DROP До применения этого правила можно проверить доступность с ub1 до ub2 с помощью команды ping: root@ub1:/home/user# ping 10.0.0.3 PING 10.0.0.3 (10.0.0.3) 56(84) bytes of data. 64 bytes from 10.0.0.3: icmp\_seq=1 ttl=64 time=0.022 ms 64 bytes from 10.0.0.3: icmp\_seq=2 ttl=64 time=0.015 ms После закрытия доступа к подсети 10.0.0.0/24 для ub1 команда ping успешно выполняться не будет: 21 root@ub1:/home/user# ping 10.0.0.3 PING 10.0.0.3 (10.0.0.3) 56(84) bytes of data. --- 10.0.0.3 ping statistics --- 6 packets transmitted, 0 received, 100% packet loss, time 5039ms Далее по ходу выполнения работы понадобится выполнять команды iptables. Для того, чтобы после перезагрузки/выключения машин настройки iptables не стерлись, следует выполнить следующие действия: 1. Для сохранения правил, настроенных в текущей сессии, выполните команду: # iptables-save > /root/firewall.rules 2. Для загрузки их после включения компьютера необходимо добавить следующую строку в файл /etc/network/interfaces: pre-up iptables-restore < /root/firewall.rules Это позволит применить правила после перезагрузки/выключения компьютера. Эти действия помогут воссоздать общую картину взаимодействия узлов (ub1, ub2) с NAT-маршрутизатором (ub-nat). Для выполнения лабораторной работы настройте виртуальные машины, как указано на рис. 3.3. Для усложнения задачи можно сменить IP-адреса узлов на любые, подходящие к условию. Для удобства работы с виртуальными машинами рекомендуется добавить к каждой машине еще один интерфейс – виртуальный адаптер хоста (Hostonly). При подключении типа «Виртуальный адаптер хоста» гостевые ОС могут взаимодействовать между собой, а также с хостом. Но все это только внутри самой виртуальной машины VirtualBox. В этом режиме адаптер хоста использует свое собственное, специально для этого предназначенное устройство, которое называется vboxnet0. Также VirtualBox создает подсеть и назначает IP-адреса сетевым картам гостевых операционных систем. Гостевые ОС не могут взаимодействовать с устройствами, находящимися во внешней сети, так как они не подключены к ней через физический интерфейс. Данное подключение позволит поднять ssh-сессию с хостового ПК к каждому из узлов (для Windows можно использовать Putty), тем самым упростив доступ к управлению и настройке виртуальных машин.



Лабораторная 6

СОЗДАНИЕ ВИРТУАЛЬНЫХ ЛОКАЛЬНЫХ СЕТЕЙ VLAN

Логическое разграничение компьютерной сети на виртуальные подсети. Локальные сети в наше время состоят из большого количества узлов:

компьютеров, маршрутизаторов, коммутаторов, оргтехники. Чем больше количество узлов в одной подсети, тем большее количество сетевых процессов в них происходит, например, широковещательных запросов, передачи данных с

узла на узел, отправки запросов на узлы-сервисы и т. д. Далее рассмотрен пример локальной сети предприятия (рис. 4.1).

Сеть предприятия является общей для каждого узла сети. Ethernet на рисунке – абстракция коммутаторов и каналов связи между узлами. При данном

построении без дополнительных настроек каждый узел виден внутри сети.

Руководство, бухгалтерия и остальные (разработчики, сервисы и т. д.) имеют

доступ друг к другу. С технической точки зрения все в порядке, у каждого есть

доступ к сети Интернет и к любым ресурсам компании. Однако с точки зрения

организации рабочего процесса данная настройка не очень удачна, например

любой узел имеет доступ к ресурсам бухгалтерии, финансовым документам,

персональным данным.

**Данная проблема может быть решена различными способами. Вот несколько стандартных вариантов решения:**

**1. Настроить правила доступа к каждому узлу на всех узлах сети.**

**2. Физически разделить общую сеть на несколько сетей, после чего**

**настроить управляемый доступ с помощью маршрутизатора.**

**3. Настроить виртуальные сети (VLAN).**

Первый вариант очень трудоемок – на каждом узле необходимо настроить права доступа (ACL – access control list). После первичной настройки обслуживание данной системы также затрудняется изменениями внутри сети:

при добавлении нового узла доступ к нему необходимо настроить на каждом

элементе сети; при изменении политики доступа к какому-то узлу необходимо

перенастроить правила доступа на всех устройствах. Также сильно повышается риск человеческой ошибки при настройке подобного решения, поэтому в

больших сетях (уже начиная с 10 компьютеров) он обычно не используется.

*Второй и третий варианты подходят для решения задачи в общем виде,*

*однако имеют свои достоинства и недостатки. Например, при физическом разделении сети необходимо проложить каналы связи, установить коммутаторы*

*до каждого узла, который будет принадлежать конкретной подсети (рис. 4.2).*

*В случае с общей сетью все компьютеры подключены к любому коммутатору в любом порядке. Для того чтобы физически разделить логические*

группы компьютеров (А, Б, В), необходимо добавить в сеть большее количество коммутаторов (их число зависит от расположения конечных узлов каждой

из групп), обеспечить физическое подключение от конечного узла к требуемой

подсети. На рисунке представлен простейший пример из трех групп, компьютеры которых находятся недалеко друг от друга. Чем больше компьютеров

27

в сети и чем дальше они находятся друг от друга, тем большее количество

сетевых элементов необходимо добавить к сети: установить коммутаторы и

проложить новые линии связи и т. д. Подобное решение обычно используют,

когда компания не слишком крупная либо необходимо обеспечить реальное

физическое разделение между узлами разных групп.

**Для снижения издержек можно использовать третий вариант – настроить**

**виртуальные подсети в общей сети, что позволит логически разделить подсети. Если нет необходимости физического разделения подсетей, данный**

**вариант позволит сэкономить на покупке нового оборудования и прокладки**

**линий связи между узлами (рис. 4.3).**

Данный вариант включает только логическую настройку узлов. Таким

образом можно обособить между собой подсети А, Б, В (руководство,

бухгалтерия, остальные) без дополнительных финансовых затрат на закупку

оборудования.

**Виртуальной локальной сетью VLAN называется логическая группа**

**узлов сети, трафик которой, в том числе и широковещательный, полностью**

**изолирован от других узлов сети на канальном уровне. Это означает, что передача кадров между разными виртуальными сетями на основании МАСадреса невозможна независимо от типа адреса – индивидуального, группового**

**или широковещательного. В то же время внутри виртуальной сети кадры передаются по технологии коммутации, т. е. только на тот порт, который связан**

**с МАС-адресом назначения кадра**.

**VLAN обладают следующими преимуществами:**

1. Гибкость внедрения – VLAN являются эффективным способом группировки сетевых пользователей в виртуальные рабочие группы независимо от

их физического размещения в сети.

2. Ограничивают распространение широковещательного трафика, что

увеличивает полосу пропускания, доступную для пользователя.

3. **Позволяют повысить безопасность сети,** определив с помощью фильтров, настроенных на коммутаторе или маршрутизаторе, политику взаимодействия пользователей из разных виртуальных сетей.

Далее рассмотрена реализация данного решения. **Она может происходить**

**на уровне сетевых узлов (коммутаторов, маршрутизаторов) или на уровне**

**каждого ПК.** В обоих вариантах происходит следующее: пакеты дополнительно маркируются информацией о VLAN. **Эта информация называется тегом и состоит из идентификатора протокола тега, приоритета, идентификатора**

**канонического формата и самого идентификатора VLAN.** Каждой такой

виртуальной подсети выделяется номер, чтобы сетевой узел мог обеспечить

обмен данными между узлами одного VLAN и не перенаправлял данные на

узлы с другими номерами.

**VLAN ID — это идентификационный 12-битный набор цифр, благодаря которому можно создавать многоуровневые виртуальные сети, обходя любые физические препятствия, как, например, географическое положение**

**Для новой подсети можно выбрать любое значение**

**между 1 и 4094. VLAN ID 0 применяется в общей сети, которая не использует**

**VLAN, однако в условиях стандартов, использующих VLAN (например,**

**IEEE 802.1q, IEEE 802.1p), данным, исходящим из общей сети, назначается**

**тег по умолчанию.**

VLAN на основе сетевых узлов. При настройке VLAN на основе коммутаторов и маршрутизаторов ничего не настраивается на компьютерах и других

узлах, не отвечающих за сетевое окружение (оргтехника и т. д.). На коммутаторах создаются настройки для групп портов. Им назначаются различные

VLAN ID в зависимости от того, компьютер какой логической сети к ним подключается. При проходе данных от компьютера к коммутатору пакеты тегируются (рис. 4.4).

Получая в результате пакет с тегом, следующие узлы разбирают его, анализируют и, если он соответствует настройкам на порте (VLAN ID), отбрасывают его либо отправляют нужному узлу далее. Благодаря этим настройкам

работает логическое разграничение сети.

VLAN на основе сетевых карт узлов (ПК, оргтехника и др.). Настройку

виртуальной подсети можно выполнить и на узлах компьютеров на сетевых

картах. В этом случае на узле добавляется дополнительный виртуальный

интерфейс, который подключается к физической сети через существующий

сетевой интерфейс. Далее приведен пример настройки ОС Ubuntu.

Необходимо установить утилиту vlan:

sudo apt-get install vlan

В случае, когда физический сетевой интерфейс имеет имя eth0, в файле

/etc/network/interfaces необходимо добавить следующие строки:

auto eth0.1001

iface eth0.1001 inet static

address 1.0.0.1

netmask 255.255.255.0

vlan\_raw\_device eth0

Для применения настроек необходимо перезагрузить ПК. Эти действия

создадут виртуальный сетевой интерфейс с VLAN ID 1001 с IP-адресом

1.0.0.1, подключенный к сетевому интерфейсу eth0. После настройки данный

ПК сможет общаться с другими компьютерами, у которых настроены такой же

VLAN ID и IP-адрес из такой же подсети.

30

Данный вариант создания виртуальных подсетей является более сложным, так как необходимо выполнять настройки VLAN на каждом узле. Если

коммутаторы не поддерживают стандарты, использующие VLAN, тогда

настройку VLAN можно выполнить только за счет изменения конфигураций

всех компьютеров сети.

**VLANS или Virtual LANS — это технология, позволяющая разделить домашнюю сеть на сегменты с помощью недорогих коммутаторов.**

Как правило, коммутатор отправляет широковещательный трафик на все подключенные порты и позволяет устройствам, подключенным к любому порту, взаимодействовать с любым другим устройством.

**VLAN были созданы для уменьшения объема широковещательного трафика в сети.**

*VLAN-ы же помогают снизить сетевой трафик формированием нескольких широковещательных доменов, разбивая большую сеть на несколько меньших независимых сегментов с небольшим количеством широковещательных запросов, посылаемых к каждому устройству всей сети в целом*.

*VLAN (аббр. от англ. Virtual Local Area Network) — виртуальная локальная компьютерная сеть.* Представляет собой группу хостов с общим набором требований[источник не указан 719 дней], которые взаимодействуют так, как если бы они были подключены к широковещательному домену независимо от их физического местонахождения. VLAN имеет те же свойства, что и физическая локальная сеть, но позволяет конечным членам группироваться вместе, даже если они не находятся в одной физической сети. Такая реорганизация может быть сделана на основе программного обеспечения вместо физического перемещения устройств.

Обозначение членства в VLAN

Для этого существуют следующие решения:

по порту (англ. port-based, 802.1Q): порту коммутатора вручную назначается одна VLAN. В случае, если одному порту должны соответствовать несколько VLAN (например, если соединение VLAN проходит через несколько сетевых коммутаторов), то этот порт должен быть членом транка. Только одна VLAN может получать все кадры, не отнесённые ни к одной VLAN (в терминологии 3Com, Planet, D-Link, Zyxel, HP — untagged, в терминологии Cisco, Juniper, Eltex — native VLAN). Сетевой коммутатор будет добавлять метки данной VLAN ко всем принятым кадрам не имеющим никаких меток. VLAN, построенные на базе портов, имеют некоторые ограничения.

по MAC-адресу (MAC-based): членство в VLANe основывается на MAC-адресе рабочей станции. В таком случае сетевой коммутатор имеет таблицу MAC-адресов всех устройств вместе с VLANами, к которым они принадлежат.

по протоколу (Protocol-based): данные 3-4 уровня в заголовке инкапсулированного в кадр пакета используются чтобы определить членство в VLANe. Например, IP-машины могут быть переведены в первую VLAN, а AppleTalk-машины во вторую. Основной недостаток этого метода в том, что он нарушает независимость уровней, поэтому, например, переход с IPv4 на IPv6 приведет к нарушению работоспособности сети.

методом аутентификации (англ. authentication based): устройства могут быть автоматически перемещены в VLAN основываясь на данных аутентификации пользователя или устройства при использовании протокола 802.1X.

широковещательный домен - это совокупность всех портов коммутаторов соединенных в один сегмент.

Такие порты коммутатора называются портами доступа или нетегированными портами (access port, untagged). Обычно они используются для подключения конечных узлов сети, которые не должны ничего знать об иных VLAN и работать в собственном сегменте.

Принято считать, что достаточно изолировать трафик внутри отдельного VLAN и становится абсолютно невозможно ни просмотреть, ни, тем более, модифицировать его другим участникам сети, которые не имеют прямого доступа к этому VLAN'у.

**номер 1 — это стандартный VLAN, который изначально есть на каждом коммутаторе.**

**На каких узлах может быть настроен VLAN?**

Лабораторная 7

СЕТЕВЫЕ ЭКРАНЫ. IPTABLES

5.1. Цель и задачи

Целью работы является изучение принципов работы с сетевыми экранами. Необходимо решить следующие задачи:

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

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

5.2. Основные теоретические сведения

**Сетевой экран – это программный или аппаратный элемент компьютерной сети, осуществляющий контроль и фильтрацию проходящего через него**

**сетевого трафика в соответствии с заданными правилами. Это необходимо для**

**того, чтобы блокировать вредоносный трафик, направленный на различные**

**сервисы, для обеспечения их бесперебойной работы и своевременного ответа**

**на легитимные запросы.**

*Существует два основных подхода к настройке сетевых экранов:*

*1. Все, что не запрещено, разрешено.*

*2. Все, что явно не разрешено, запрещено.*

В зависимости от требований безопасности и причин установки межсетевого экрана выбирается один из этих подходов. Например, если необходимо

запретить доступ части узлов к другим, проще использовать первый подход

(рис. 5.1). На маршрутизаторе можно запретить трафик, проходящий из подсети А в подсеть Б. После этих настроек узлы подсети А будут иметь доступ к

ресурсам сети Интернет, но не смогут обмениваться данными с подсетью Б.

Если требуется максимально закрыть доступ к узлу извне, то используется второй подход. В его основе лежит запрет по умолчанию любого трафика

с дальнейшим разрешением доступа с конкретных узлов, подсетей и др., а

также разрешением доступа к ним.

**Типы сетевых экранов по их местоположению**. Существует два основных варианта расположения сетевого экрана: host-based и network-based.

34

И н т е р н е т

Подсеть А Подсеть Б

Рис. 5.1. Схема доступа к сети Интернет и подсети Б

**Host-based сетевой экран** – это экран, который располагается на самой системе, где необходимо фильтровать трафик. Network-based-экран устанавливаетсяна промежуточных узлах между защищаемыми ресурсами и другими узлами.

*Host-based сетевые экраны позволяют гибко и удобно настраивать правила на конкретном компьютере, упрощая тем самым процесс настройки и обслуживания системы фильтрации. Однако для защиты большого количества*

сетевых узлов их необходимо устанавливать на каждом узле, что может

усложнить процесс администрирования системы. Кроме того, в корпоративной сети, обеспечивая защиту только на самих узлах, линии связи (коммутаторы и др.) остаются без защиты. Это может повлечь за собой отсутствие

доступа к узлам при атаках на узлы связи.

Данная проблема может быть решена с помощью network-based сетевых

экранов. Они устанавливаются на промежуточных узлах между защищаемыми

узлами и внешним миром. Их можно настроить на фильтрацию всего трафика,

проходящего через них во внутреннюю сеть и обратно, тем самым обеспечив

централизованное администрирование сетевого экрана всех узлов внутренней

сети. Однако следует отметить, что из-за дополнительных работ с проходящим

трафиком длительность коммутации через подобные маршрутизаторы увеличивается, а нагрузка на них возрастает, при большом объеме циркулирующих

данных это стоит учитывать. То же самое происходит и с защитой host-based –

она использует ресурсы защищаемого узла. Внутренние узлы, по какой-то

35

причине начавшие генерировать вредоносный трафик на соседние узлы, в

случае использования защиты network-based на концевом маршрутизаторе

заблокированы не будут.

*Есть еще один вариант использования сетевого экрана как промежуточной точки между внешней сетью и защищаемой – proxy-сервер.* Трафик перед

получением его защищаемым узлом перенаправляется на специальный узел.

Этот узел является сетевым экраном и обеспечивает анализ данных и фильтрацию вредоносного трафика. Обычно подобное перенаправление устанавливается на концевом маршрутизаторе внутренней защищаемой сети, а сам сетевой

экран может находиться внутри защищаемой сети и вне ее. Внешние узлы

proxy-сервера обычно предоставляются сторонними компаниями, предлагающими «очистку данных». Такой подход позволяет снизить нагрузку на администрирование и настройку сетевого экрана, однако конфиденциальные данные при передаче их через внешний proxy-сервер могут быть перехвачены.

Если подобную возможность нельзя допускать, можно установить внутренний

proxy-сервер (рис. 5.2).

Рис. 5.2. Схемы расположения proxy-сервера

**Типы сетевых экранов по глубине анализа трафика.** Рассмотрим два

наиболее часто используемых типа сетевых экранов.

Первый анализирует только заголовки пакетов и может работать на транспортном, сетевом и канальном уровнях иерархии DoD (TCP/IP). В данном варианте анализ входящих и исходящих пакетов осуществляется на основе информации, содержащейся в следующих полях TCP- и IP-заголовков пакетов:

IP-адрес отправителя; IP-адрес получателя; порт отправителя; порт получателя. В зависимости от отслеживания активных соединений подобные сетевые

экраны делятся на два типа: stateless (простая фильтрация) и stateful (фильтрация с учетом контекста). Экраны stateless фильтруют потоки данных на основе

36

статических данных о пакете, stateful включают в себя анализ логики работы

протоколов и приложений.

К преимуществам фильтрации на основе анализа заголовков относятся

невысокая стоимость подобных экранов, гибкость в задании правил фильтрации и небольшая задержка при прохождении трафика за счет простых правил

анализа потоков данных. Однако данные экраны не могут анализировать фрагментированные пакеты и не отслеживают взаимосвязи между пакетами в силу

отсутствия анализа данных в пакете, относящихся к уровню представления.

**Второй тип межсетевого экрана может анализировать данные в пакете и**

**работает на прикладном уровне иерархии DoD (TCP/IP**). Благодаря работе на

прикладном уровне можно организовать большое число проверок, которые будут использовать особенности работы протоколов прикладного уровня.

Например, можно добавить проверку взлома известных «дыр» в программном

обеспечении и протоколах. Но подобный функционал увеличивает время анализа данных, что сказывается на производительности. Обычно такие сетевые

экраны являются host-based-экранами или proxy-серверами, так как работают

под управлением более сложных ОС и требуют бóльших ресурсов по сравнению с экранами, работающими только с заголовками пакетов.

В данной лабораторной работе используется первый вариант сетевого

экрана Linux – Netfilter. Для его настройки будет использован iptables.

Утилита управления сетевым экраном iptables. **iptables – это утилита**

**командной строки, управляющая межсетевым экраном. С ее помощью создаются правила, которые управляют фильтрацией и перенаправлением пакетов.**

Работа с iptables возможна только от имени суперпользователя. Есть несколько

основных базовых понятий для работы с iptables: правило, таблица и цепочка.

Правило состоит из критерия, действия и счетчика. Если пакет соответствует критерию (требованию), то к нему принимается действие и этот пакет

учитывается счетчиком. Критерий анализирует свойство пакета, например,

условия совпадения источника назначения с заданным в условии. Действие

определяет, что необходимо сделать с пакетом, если он попадает под критерий. В случае срабатывания правила счетчик учитывает этот пакет и также

учитывает суммарный объем пакетов, подходящих под этот критерий, в байтах.

Таблица – совокупность базовых и пользовательских цепочек, объединенных общим функциональным назначением. Имена таблиц (как и модулей

критериев) записываются в нижнем регистре, так как не могут конфликтовать

с именами пользовательских цепочек.

37

**Цепочкой** является упорядоченная последовательность правил. Правила

в цепочке срабатывают в порядке их указания, поэтому при их настройке важен порядок. Существует пять базовых цепочек:

1. PREROUTING – для изначальной обработки входящих пакетов.

2. INPUT – для входящих пакетов, адресованных непосредственно локальному процессу (клиенту или серверу).

3. FORWARD – для входящих пакетов, перенаправленных на выход (заметьте, что перенаправляемые пакеты проходят сначала цепь PREROUTING,

затем FORWARD и POSTROUTING).

4. OUTPUT – для пакетов, генерируемых локальными процессами.

5. POSTROUTING – для окончательной обработки исходящих пакетов.

В случае необходимости можно создавать дополнительные пользовательские цепочки. Также существуют приложения, добавляющие свои цепочки в

iptables, например, docker или Open vSwitch.

**Цепочки организованы в четыре таблицы:**

1. raw – просматривается до передачи пакета системе определения состояний. Используется редко, например, для маркировки пакетов, которые НЕ

должны обрабатываться системой определения состояний. Для этого в правиле указывается действие NOTRACK. Содержит цепочки PREROUTING и

OUTPUT.

2. mangle – содержит правила модификации (обычно заголовка) IP‐

пакетов. Среди прочего поддерживает действия TTL (Time to Live), TOS (Type

of Service) и MARK (для изменения полей TTL и TOS и для изменения маркеров пакета). Содержит все пять стандартных цепочек.

3**. nat** – просматривает только пакеты, создающие новое соединение (согласно системе определения состояний). Поддерживает действия DNAT,

SNAT, MASQUERADE, REDIRECT. Содержит цепочки PREROUTING,

OUTPUT и POSTROUTING. Данная таблица используется в лаб. работе № 3.

4. filter – основная таблица, используется по умолчанию, если название

таблицы не указано. Содержит цепочки INPUT, FORWARD и OUTPUT.

На рис. 5.3 представлена упрощенная диаграмма прохождения таблиц и

цепочек.

Сначала пакет проходит через цепочку PREROUTING. В случае, если пакет должен быть передан другому узлу, далее используется цепочка

FORWARD и после нее – POSTROUTING. В случае, когда пакет отправлялся

38

текущей системе, он попадает в цепочку INPUT. После этого он обрабатывается соответствующим приложением. В случае генерации ответа или запроса

из самой системы к пакету сначала применяется цепочка OUTPUT, а после –

POSTROUTING. В случае, если на каком-то из правил данный пакет отбрасывается, к нему применяются два варианта: DROP или REJECT. Действие DROP

просто останавливает дальнейшую работу с пакетом. REJECT генерирует

ответ отправителю о блокировке пакета.

Сеть

PREROUTING

mangle nat

INPUT

mangle filter

Входная

маршрутизация

Обработка

локальным

приложением

OUTPUT

mangle nat filter

POSTROUTING

mangle nat

FORWARD

mangle filter

Сеть

Адрес назначения принадлежит этому узлу

Адрес назначения принадлежит

другому узлу

Рис. 5.3. Диаграмма процесса анализа пакетов в iptables

Далее представлены примеры использования iptables. Данная команда

запрещает весь входящий трафик:

iptables -A INPUT -j DROP

*Ключ –A означает, что правило добавляется в конец таблицы, в данном*

*примере INPUT. Так как правила к пакетам применяются по порядку, то это*

*необходимо учитывать. Чтобы вставить правило в конкретное место в цепочке, необходимо использовать ключ –I:*

iptables -I INPUT 3 –s 192.168.0.32 -j DROP

Данная команда вставляет правило на 3-ю строчку, и оно блокирует все

пакеты, у которых IP-адрес источника равен 192.168.0.32.

В следующем примере блокируются все пакеты, кроме ssh-соединений:

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

iptables -A INPUT -j DROP

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

39

все пакеты. *Важен порядок применения правил. Второе правило добавляется*

*в конце таблицы, поэтому первое правило применяется сначала*.

Вывести существующие правила можно следующей командой:

iptables –nvL

Примеры, касающиеся перенаправления трафика для NAT, рассмотрены

в лаб. работе № 3. Если необходимо заблокировать исходящие пакеты, то используется таблица OUTPUT:

iptables -A OUTPUT -d 10.10.10.10 -j DROP

Это правило блокирует все пакеты, у которых IP-адрес назначения

10.10.10.10. Для отладки работы iptables можно настроить логирование:

iptables -A INPUT -I enp0s3 -s 1.0.0.0/24 -j LOG --log-prefix

"Logging info”

iptables -A INPUT -i enp0s3 -s 1.0.0.0/24 -j DROP

Первое правило сначала записывает событие, означающее, что на интерфейс

enp0s3 пришел входящий пакет с IP-адресом источника 1.0.0.0/24. Второе правило блокирует его. Посмотреть логи можно в этом файле: /var/log/messages.

Также iptables может работать с MAC-адресами:

iptables -A INPUT -p tcp –d port 22 -m mac --mac-source

00:04:BA:67:04:37 -j ACCEPT

Данное правило разрешит прием пакетов для ssh-соединения от узла с

указанным MAC-адресом.

Кроме добавления правил в цепочку их можно также удалять поштучно

либо целиком очистить все правила. Если нужно удалить конкретное правило,

это можно выполнить, указав его порядковый номер либо полностью указав

спецификацию правила:

iptables –D INPUT 3

iptables –D INPUT –s 1.1.1.1 –j DROP

Первое правило удаляет из INPUT третье правило, а второе удаляет правило, которое отбрасывает пакеты, приходящие от IP-адреса 1.1.1.1.

Следующая команда удаляет все правила:

iptables –F

Чтобы удалить правила для конкретной цепочки для конкретной таблицы,

необходимо дополнительно указать нужную информацию:

iptables –t nat –F PREROUTING

Данный пример стирает все правила из цепочки PREROUTING для таблицы nat.

Сетевой экран – это система защиты компьютерной сети, которая ограничивает прохождение входящего, исходящего и внутрисетевого трафика.

Это программное обеспечение или программно-аппаратный модуль, который принимает решение – пропустить или заблокировать проходящий пакет данных. Его основная функция – блокировать вредоносную активность и предотвращать несанкционированные действия пользователей как в частной сети, так и за ее пределами.

Тип А — межсетевые экраны уровня сети. Устанавливаются на периметре системы или между составляющими одной физически сегментированной сети. Такие экраны бывают только программно-аппаратными.

Тип Б — межсетевые экраны уровня логических границ сети. Они также используется на периметре системы или между несколькими сегментами, однако в этом случае сеть может быть поделена логически, а не физически. Такие экраны могут быть как программно-аппаратными, так и программными.

Тип В — межсетевые экраны уровня узла. Такой экран применяется на узле информационной системы. Бывают только программными, устанавливаются на мобильном или стационарном оборудовании отдельно взятого узла системы.

Тип Г — межсетевые экраны, используемые на серверах, обслуживающих веб-приложения и службы. Отличие этих экранов заключается в том, что они ведут работу с информационными потоками по протоколу передачи гипертекста от сервера к серверу. Эти МЭ бывают как программными, так и программно-аппаратными.

Тип Д — межсетевые экраны используемые в системах управления технологическими и производственными процессами. Они работают с промышленными протоколами передачи информации (Industrial Ethernet, HART, CAN, Profibus и др). Бывают как программные, так и программно-аппаратные.

Таким образом, применение host-based значительно более масштабируемое решение, чем использование классических файерволов.

Таким образом, host-based файерволы обладают целым рядом преимуществ:

Возможность гибко настраивать политики трафика используя метаданные;

Решение проблемы микросегментации в неоднородной среде;

Масштабируемость решения, возможность подстраиваться под динамично развивающуюся архитектуру виртуальной сети;

Доступность: host-based файерволы значительно дешевле, чем network-based

Однако, нельзя сказать, что при использовании host-based файероволов не возникает абсолютно никаких проблем. Эти устройства потребляют значительную часть полезной нагрузки, что особенно сильно проявляется, если потоки трафика значительно вырастают. Именно на этом этапе мы платим за масштабируемость нашего решения. Более того, host-based значительно уступает обычным сетевым файерволам по быстродействию. Эта проблема на сегодняшний день является наиболее актуальной и играет важную роль, когда речь касается выбора устройств для защиты сети.

Iptables — это межсетевой экран для операционных систем Linux. Успешно применять этот инструмент могут не только продвинутые пользователи Linux, но и новички.

Каковы недостатки прокси-серверов?

# 1: Кража личных данных

Кэш-память прокси-сервера может хранить много конфиденциальных данных, таких как имена пользователей и пароли.

Хотя вначале это может показаться неопасным, поскольку прокси-серверы могут блокировать получение этой информации внешними серверами, это может быть и плачевно, если лицо, связанное с прокси-сервером, не отвечает за сохраненную информацию.

В этой ситуации человек со злым умыслом может легко получить конфиденциальные данные, которые могут скомпрометировать вашу личность в Интернете.

Именно по этой причине вы должны использовать авторитетного интернет-провайдера для покупки прокси-серверов.

# 2: Несовместимость

Хотя использование прокси-серверов может быть весьма выгодным, существуют проблемы, которые могут возникнуть, если прокси-сервер несовместим с локальной сетью серверов.

Прокси-серверы являются независимыми системами, которые работают в соответствии с их собственными конфигурациями; следовательно, вы можете приобрести прокси-сервер, который подходит для локального сервера, или настроить один в соответствии с другим.

Это еще одна причина, почему вы должны нанять надежного интернет-провайдера, предлагающего удобные прокси-серверы.

# 3: Стоимость

Цена на прокси может достигать различных значений.

Как правило, бесплатные прокси очень медленные.

Вернемся к iptables, функции NAT реализованы в одноименной таблице nat (все таблицы обозначаются строчными буквами), которая содержит цепочки PREROUTING, POSTROUTING и OUTPUT. В первые две цепочки попадает весьВ таблице nat доступно два основных действия:

SNAT - Source Network Address Translation - изменение адреса и порта источника пакета, доступен в цепочке POSTROUTING

DNAT - Destination Network Address Translation - изменение адреса и порта назначения пакета, доступен в цепочках PREROUTING и OUTPUT трафик узла, в последнюю только собственный исходящий.

В качестве продолжения нашего цикла о брандмауэре iptables мы рассмотрим таблицу nat, в которой происходит преобразование сетевых адресов. С этой таблицей, также, как и с filter вы будете часто встречаться, выполняя такие привычные и повседневные задачи, как выход в интернет, проброс портов или перенаправление трафика. При этом действие механизма NAT более сложное и требует более глубокого понимания происходящих процессов, поэтому советуем уделить внимание в первую очередь теоретической части.

DROP и REJECT отклоняют соединение, но REJECT еще дополнительно отправляет инициатору ICMP пакет, с сообщением, что попытка соединения отклонена. DROP - просто отбрасывает пришедший пакет, не отправляя никакого сообщения в ответ. В случае DoS-атаки, REJECT будет создавать избыточный трафик своими ответами, так что в случае отклонения соединения лучше всего использовать DROP.

Какую цепочку лучше использовать, чтобы заблокировать доступ с ПК

на ресурс во внешней сети?

Помимо набора правил и индивидуальных настроек, защиту корпоративных сетей организуют при помощи разных типов межсетевых экранов, аппаратных и программных. Их отличает способ блокировок-фильтрации, управления функционалом.

Межсетевые экраны делят на две группы — аппаратные и программные.

правила в фаерволе Iptables читаются сверху-вниз