**Практическая работа №6**

Для выполнения данной практической работы необходимо подключиться к лабораторному стенду. Адреса для подключения и пароль выдаст преподаватель во время пары.

Для подключения необходимо использовать VNC-клиент. Скачать его можно на сайте: https://www.realvnc.com/en/connect/download/viewer/ Необходимо выбрать вариант **«Standalone EXE x64»**, и нажать на кнопку «Download VNC Viewer» (рисунок 1).

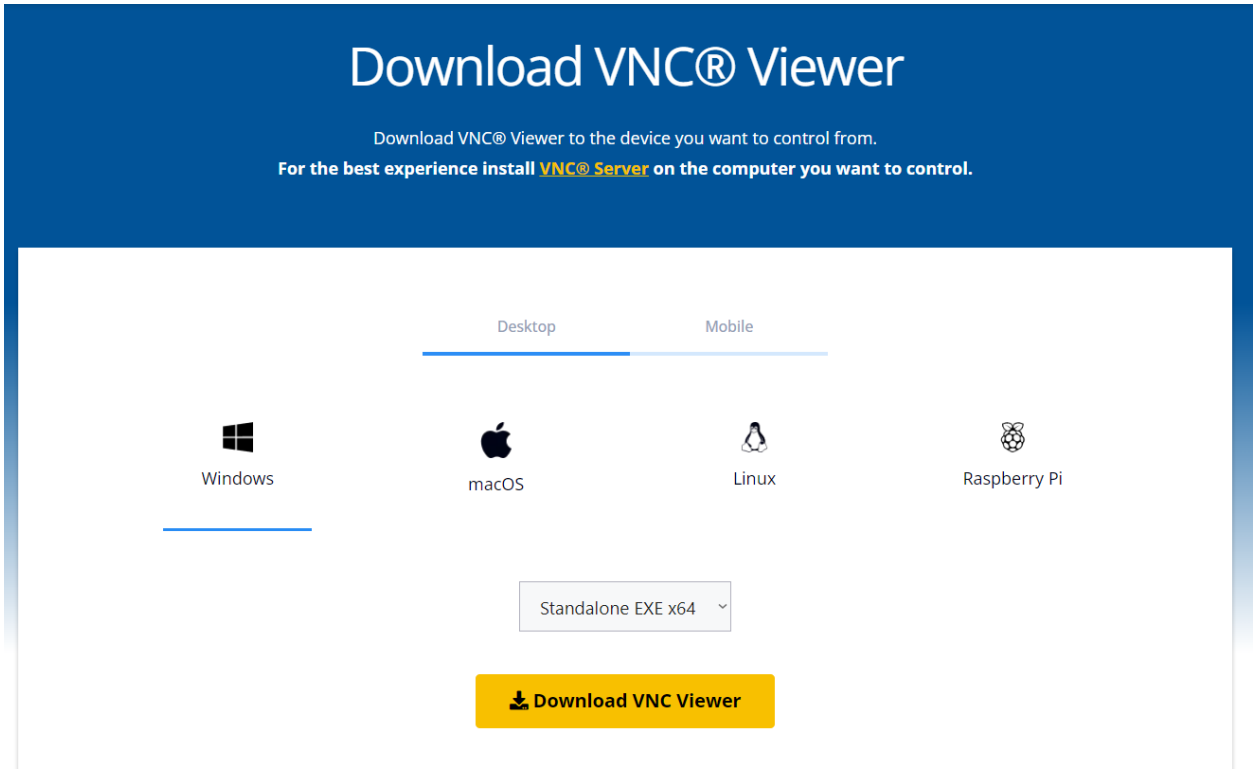


*Рисунок 1. Скачивание VNC клиента*

Для подключения к **ВМ с ОС Windows** необходимо использовать порт подключения, который начинается с цифры 6. Пароль в ВМ: 12345

Для подключения к **ВМ с ОС AstraLinux** необходимо использовать порт подключения, который начинается с цифры 7. Пароль в ВМ: iamlordofnowhere

Одним из наиболее распространенных применений ОС Linux является работа в качестве сетевых сервисов — веб-серверов, файловых серверов, маршрутизаторов, сетевых экранов, анализаторов трафика, сетевых трансляторов, точек доступа и т.д. Это возможно благодаря развитым средствам и возможностям конфигурирования в ядре ОС Linux и обширной и гибкой системе сетевых служб.

Одной из наиболее востребованных функций в сетевой инфраструктуре является работа в качестве маршрутизатора (устройства и серверы, выполняющие данную функцию, часто также называют роутерами). Во многих случаях это интегрированное устройство выполняет также и другие функции — межсетевого экрана (файерволл, брандмауэр) и транслятора адресов (NAT, актуально для IPv4).

Вообще говоря, настольные версии Linux (куда как раз относится Astra, а также Ubuntu, Fedora и многие другие) не являются оптимальными для данной задачи — они излишне перегружены лишними пакетами, в основном, графического интерфейса, который потребляет много ресурсов, а на сервере или сетевом устройстве обычно бесполезен. Тем не менее, механизмы работы настольных версий Linux и серверных/встроенных не отличаются (на самом деле, отличаются, но в данном случае эти отличия не имеют принципиального значения). Как именно следует настраивать ту или иную возможность в большей степени зависит от дистрибутива (а конкретнее, от включенных туда программных средств).

Чтобы успешно выполнять свою функцию маршрутизатора, узел с ОС Linux должен решать три задачи для проходящих пакетов:

1) Перенаправление пакетов (с одного сетевого интерфейса на другой)
2) Трансляция сетевых адресов
3) Фильтрация пакетов

В простейшем случае п.3 не требуется для работоспособности, но необходим для безопасности (вопреки распространенному мнению, трансляция адресов не обеспечивает сетевой безопасности и легко преодолевается путем посылки специально сформированных пакетов).

Для решения задачи №1 в ядре Linux необходимо включить соответствующую опцию и осуществить настройку маршрутов (чтобы пакет было, куда перенаправлять). Конфигурация маршрутов во многих случаях осуществляется автоматически, например, с помощью протокола DHCP (он позволяет автоматически конфигурировать как маршрут (шлюз) по-умолчанию, так и присылать конкретные маршруты до определенных узлов и / или подсетей через соответствующие опции). Поэтому нам необходимо включить перенаправление пакетов командой

```
echo 1 > /proc/sys/net/ipv4/ip_forward
```

**или**

```
sysctl -w net.ipv4.ip_forward=1
```

Для постоянной работы маршрутизатором эту опцию следует сделать перманентной. Для этого необходимо отредактировать файл /etc/sysctl.conf или создать файл в папке /etc/sysctl.d/ , включив туда строку

```
net.ipv4.ip_forward=1
```

Существуют и другие механизмы включения перенаправления пакетов (например, через параметры сети system-network, если используется данный вариант конфигурации), но они зависят от дистрибутива.

Для начала работы нормального маршрутизатора (типа тех, что стоят у провайдеров) данной опции в сочетании с таблицами маршрутизации достаточно. В случае домашнего устройства требуется еще как минимум настройка сетевой трансляции адресов. Для ее настройки необходимо разобраться, какой интерфейс является внутренним (локальная сеть), а какой внешним (сеть провайдера). Суть сетевой трансляции (в данном случае) в том, что все компьютеры в локальной сети представляются как один узел с одним сетевым адресом (маршрутизатор). Это позволяет экономить ценный ресурс адресного пространства IPv4.

Существует множество вариантов трансляции адресов (с перенаправлением портов, без оного, трансляция адреса источника, трансляция адреса назначения и пр.). В нашем случае будет использоваться (а для домашних маршрутизаторов используется в 100% случаев) динамическая трансляция адреса источника — маскарадинг. Для включения маскарадинга следует выполнить команду

iptables -t nat -A POSTROUTING -o [выходной_интерфейс] -j MASQUERADE

например,

```
iptables -t nat -A POSTROUTING -o eth0 -j MASQUERADE
```

Для перманентного сохранения конфигурации сетевой трансляции требуется обратиться к руководству вашего дистрибутива. В большинстве случаев данная задача решается через настройку iptables (за межсетевой экран и за сетевую трансляцию в Linux отвечает одна подсистема — netfilter) посредством включенного в дистрибутив инструмента (ufw, firewalld и другие). В Astra Linux используется файервол ufw.

Для начала проверим работу файервола ufw выполнив команду

```
ufw status
```

Помните, что эту, и многие другие команды системного администрирования в ОС Linux следует запускать от имени

суперпользователя — приписав sudo в начале команды или выполнив перед началом административных действий команду sudo su.

Команда ufw status выдаст неутешительный результат — файервол отключен. Включим его командой

```
ufw enable
```

Теперь необходимо разрешить транзитные соединения командой

```
ufw default allow routed
```

**ИЛИ** отредактировав файл /etc/default/ufw установить параметру DEFAULT_FORWARD_POLICY значение ACCEPT:

```
DEFAULT_FORWARD_POLICY="ACCEPT"
```

К сожалению, ufw является межсетевым экраном для настольных ОС, и не имеет встроенной поддержки маскарадинга (в отличие от, например, firewalld). Однако, в него включен механизм хуков, позволяющий дополнять правила файервола любыми, определенными в семантике iptables (firewalld обладает аналогичными функциями, практически любой межсетевой экран позволяет задавать правила iptables вручную, поскольку работает именно через прослойку iptables. Исключением из данного правила является лишь NFT (NFTABLES), которые представляют собой концептуально новую модель работы netfilter).

Воспользуемся этим механизмом для создания необходимого нам правила. Для этого отредактируем файл /etc/ufw/before.rules включив в него следующие строки:

```
*nat
:POSTROUTING ACCEPT [0:0]
#Forwardtraffic from eth1 through eth0.
-A POSTROUTING -s 192.168.1.0/24 -o eth0 -j MASQUERADE
#don't delete the 'COMMIT' line or these nat table rules won't
#be processed
COMMIT
```

Обратите внимание, что **СТРОКИ НУЖНО ВКЛЮЧАТЬ ЛИБО СТРОГО В НАЧАЛО ФАЙЛА, ЛИБО СТРОГО В КОНЕЦ** — оказавшись посередине, они разрушат настройки внутренних механизмов ufw.

После этого необходимо перезагрузить файервол командами

```
ufw disable
ufw enable
```

**или** же перезагрузив всю систему целиком.

Проверим наличие нашего правила командой

```
iptables -t nat -L
```

**или**

```
iptables-save
```

(Последняя при этом выведет вообще все настроенные в системе правила)

Теперь осуществите настройку сетевого адаптера **eth1** в соответствии с конфигурацией (как настраивали в практической работе 6) со следующими параметрами:

Адрес 192.168.1.1

Маска сети /24

Шлюз отсутствует

Перезапустите eth1 через ifdown и ifup

Этой конфигурации достаточно, чтобы доступ к сети из ОС Windows заработал при грамотной ручной настройке. Для автоматической настройки необходимо запустить в ОС Linux службу автоматической конфигурации сети. Наиболее универсальной из таких служб является ISC DHCPD (на домашних системах часто используют dnsmasq, являющийся более предпочтительным для систем с ограниченными ресурсами). Установим dhcpd командой

```
apt install isc-dhcp-server
```

и настроим его, отредактировав файл /etc/dhcp/dhcpd.conf (пример содержимого можно удалить)

```
subnet 192.168.1.0 netmask 255.255.255.0 {
  range 192.168.1.50 192.168.1.240;
  option routers 192.168.1.1;
  option broadcast-address 192.168.1.255;
  option domain-name-servers 192.168.1.1;
  authoritative;
}
```

Запустим dhcpd командой

```
systemctl enable --now isc-dhcp-server
```

Необходимо подождать пару минут. После этого узел с ОС Windows автоматически получит сетевой адрес из заданного диапазона.

Проверим работоспособность сети, выполнив в ОС Windows команду (для этого надо подключиться к ВМ с ОС Windows)

```
ping -n 10 8.8.8.8
```

ОС должна получать ответ от сервера. Однако при проверке работы Интернета, например, через браузер, обнаружатся проблемы. Да и привычные команды проверки сети

ping ya.ru

работать не будут. Это возможно исправить ручной конфигурацией сетевого стека ОС Windows, но мы пойдем другим путём. Для полностью автоматической работы сети не хватает последнего элемента — сервера службы доменных имен. На его роль мы возьмем ISC BIND, наверное, наиболее функциональный вариант из возможных. На основе именно BIND работает большая часть корневых узлов системы DNS всей глобальной сети. Большая часть его возможностей в нашем случае останется невостребованной, но сложные нестандартные конфигурации — конек BIND. Для систем с ограниченными ресурсами чаще используется dnsmasq, требующий меньше ресурсов, но поддерживающий лишь кэширующий режим (режима мастера (авторитетного) и форвардинга там нет). Установим ISC BIND командой

```
apt install bind9
```

Сам исполняемый файл демона, зовется, как ни странно named (у слова bind уже есть другой смысл). По умолчанию он уже сконфигурирован для работы в кеширующем режиме с рекурсивной обработкой запросов начиная с корневых серверов системы DNS. Все, что нам необходимо, это запустить демон командой

```
systemctl enable --now bind9
```

Не забываем, что для работы сетевых сервисов необходимо разрешить их порты (или профили) в файерволе, для ufw это делается командой

```
ufw allow Bind9
```

После этого на ВМ с ОС Windows должен появиться доступ в глобальную сеть без какой-либо дополнительной настройки (возможно, после перезагрузки или отключения/включения сетевого адаптера).

Заполните файл отчета «Шаблон для практической 7». Прикрепите его в СДО с названием «ПР6_Фамилия_Группа», где в названии будет указана ваша фамилия и группа.

Данный отчет должен содержать скриншоты выполнения работы (замените скриншотом слово <..скриншот..> в соответствующем пункте).

На **ВСЕХ** скриншотах, которые вы делаете, должно быть видно ваше ФИО и группу (для этого откройте блокнот и запишите их там), текущую дату и время и номер ВМ.

**Ответьте на теоретические вопросы:**

1) Что такое маскарадинг? чем отличаются цели MASQUERADE и SNAT в iptables?

2) В конфигурационных файлах большинства роутеров присутствует строка

```
-A FORWARD -p tcp -m tcp --tcp-flags SYN,RST SYN -j TCPMSS --clamp-mss-to-pmtu
```

Что она делает? Почему она необходима? Почему в данном случае все работает без нее?