Практическая работа №7

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).

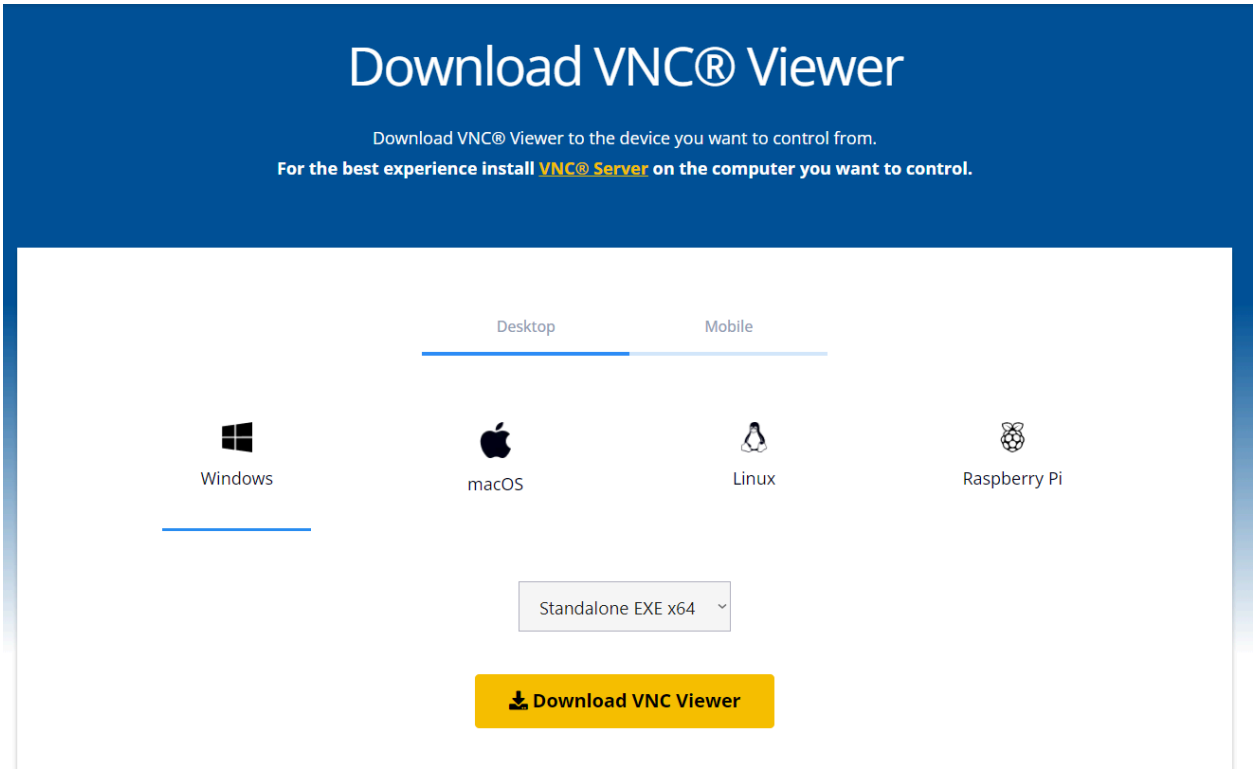

Рисунок 1. Скачивание VNC клиента

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

Задание 1)

В **ВМ с ОС AstraLinux** сконфигурируйте службу удаленного доступа SSH и проверьте её работу, подключившись к виртуальной машине с помощью подходящего ssh клиента (Putty для Windows, можно использовать ssh клиент, входящий в поставку современных версий Windows).

0. Станьте суперпользователем

```
sudo su
```

1. Откройте конфигурационный файл

```
nano /etc/ssh/sshd_config
```

2. Разрешите:

а) прослушивать порт 22: Port 22

б) прослушивать IP-адрес eth0 10.0.99.15: ListenAddress 10.0.99.15

в) подключаться с помощью ключей: PubkeyAuthentication yes

3. Перезагрузите службу удаленного доступа ssh

```
service sshd restart
```

Подключение выполняется следующим образом:

На своей машине (**не в ВМ**) запускаем клиент Putty и в строке HostName указываем адрес подключения, а в строке Port — порт.

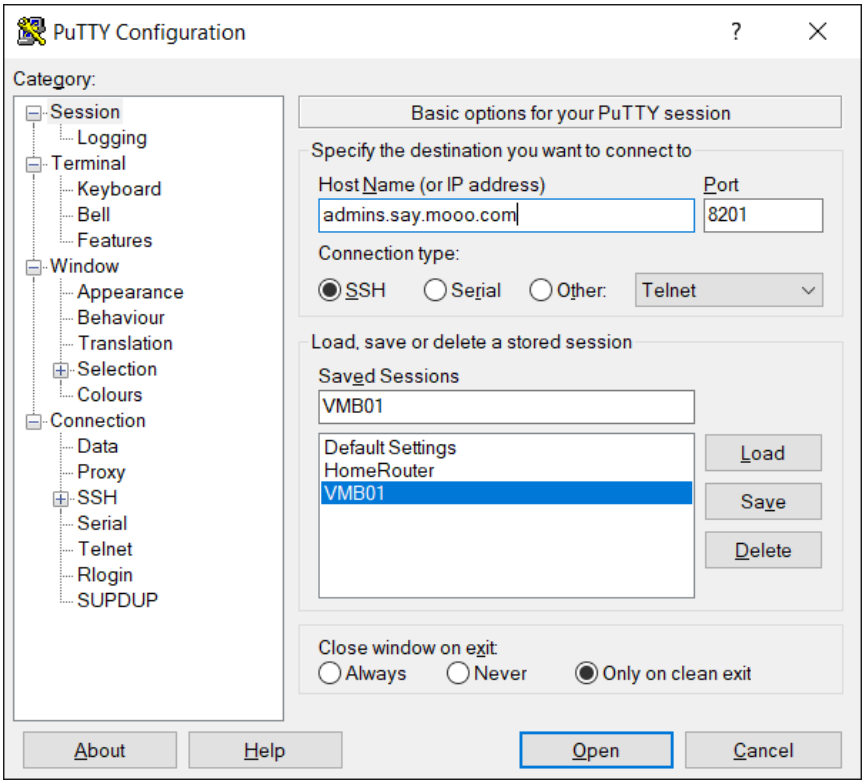

Рисунок 2. Начальный экран PuTTY

Взять значения порта можно из таблицы

Виртуальная машина	Адрес подключения	Порт
A01	admins.say.mooo.com	8101
A02	admins.say.mooo.com	8102
A03	admins.say.mooo.com	8103
A04	admins.say.mooo.com	8104
A05	admins.say.mooo.com	8105
A06	admins.say.mooo.com	8106
A07	admins.say.mooo.com	8107
A08	admins.say.mooo.com	8108

Виртуальная машина	Адрес подключения	Порт
B01	admins.say.mooo.com	8201
B02	admins.say.mooo.com	8202
B03	admins.say.mooo.com	8203
B04	admins.say.mooo.com	8204
B05	admins.say.mooo.com	8205
B06	admins.say.mooo.com	8206
B07	admins.say.mooo.com	8207
B08	admins.say.mooo.com	8208

A09	admins.say.mooo.com	8109
A10	admins.say.mooo.com	8110
A11	admins.say.mooo.com	8111
A12	admins.say.mooo.com	8112
A13	admins.say.mooo.com	8113
A14	admins.say.mooo.com	8114
A15	admins.say.mooo.com	8115
A16	admins.say.mooo.com	8116
A17	admins.say.mooo.com	8117
A18	admins.say.mooo.com	8118
A19	admins.say.mooo.com	8119
A20	admins.say.mooo.com	8120
A21	admins.say.mooo.com	8121
A22	admins.say.mooo.com	8122
A23	admins.say.mooo.com	8123
A24	admins.say.mooo.com	8124
A25	admins.say.mooo.com	8125
A26	admins.say.mooo.com	8126
A27	admins.say.mooo.com	8127
A28	admins.say.mooo.com	8128
A29	admins.say.mooo.com	8129
A30	admins.say.mooo.com	8130
A31	admins.say.mooo.com	8131
A32	admins.say.mooo.com	8132
A33	admins.say.mooo.com	8133
A34	admins.say.mooo.com	8134
A35	admins.say.mooo.com	8135
A36	admins.say.mooo.com	8136
A37	admins.say.mooo.com	8137
A38	admins.say.mooo.com	8138
A39	admins.say.mooo.com	8139
A40	admins.say.mooo.com	8140

B09	admins.say.mooo.com	8209
B10	admins.say.mooo.com	8210
B11	admins.say.mooo.com	8211
B12	admins.say.mooo.com	8212
B13	admins.say.mooo.com	8213
B14	admins.say.mooo.com	8214
B15	admins.say.mooo.com	8215
B16	admins.say.mooo.com	8216
B17	admins.say.mooo.com	8217
B18	admins.say.mooo.com	8218
B19	admins.say.mooo.com	8219
B20	admins.say.mooo.com	8220
B21	admins.say.mooo.com	8221
B22	admins.say.mooo.com	8222
B23	admins.say.mooo.com	8223
B24	admins.say.mooo.com	8224
B25	admins.say.mooo.com	8225
B26	admins.say.mooo.com	8226
B27	admins.say.mooo.com	8227
B28	admins.say.mooo.com	8228
B29	admins.say.mooo.com	8229
B30	admins.say.mooo.com	8230
B31	admins.say.mooo.com	8231
B32	admins.say.mooo.com	8232
B33	admins.say.mooo.com	8233
B34	admins.say.mooo.com	8234
B35	admins.say.mooo.com	8235
B36	admins.say.mooo.com	8236
B37	admins.say.mooo.com	8237
B38	admins.say.mooo.com	8238
B39	admins.say.mooo.com	8239
B40	admins.say.mooo.com	8240

При этом также желательно сменить некоторые параметры:
Шрифт (ставим, который нравится, рекомендуется Consolas).

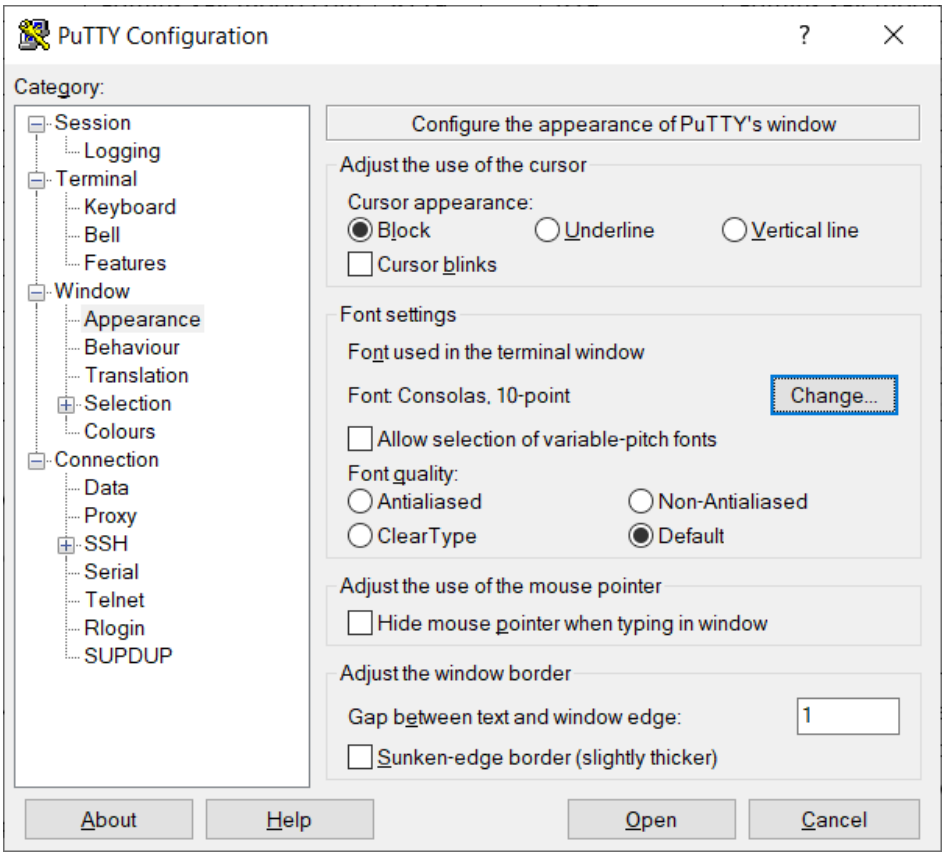

Рисунок 3. Изменение шрифта

Тип терминала (нужно установить linux вместо xterm).

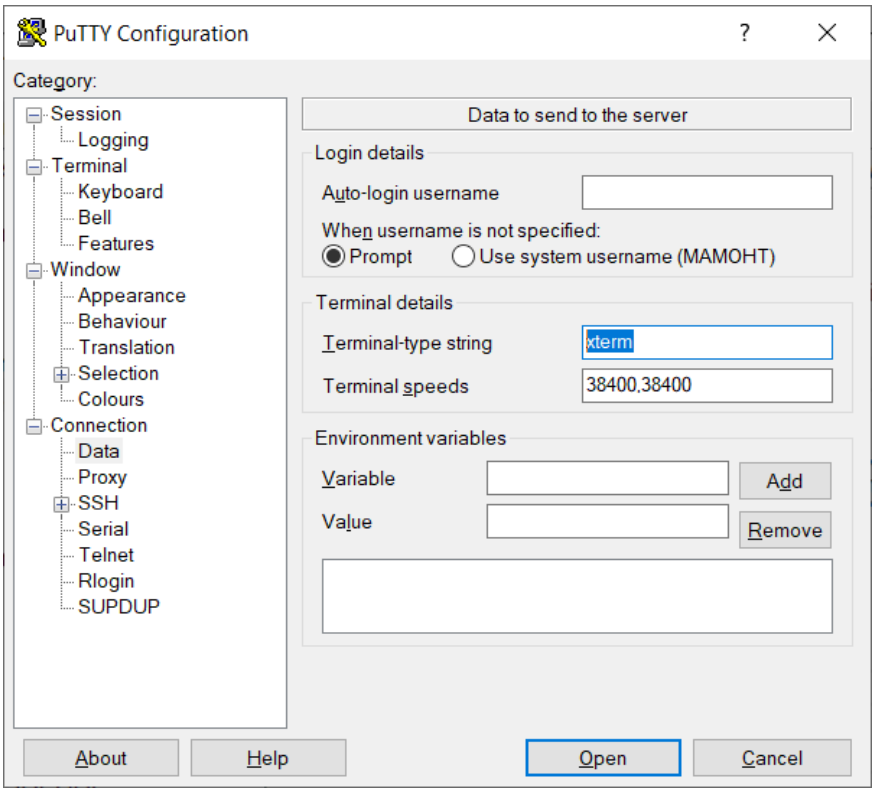

Рисунок 4. Тип терминала

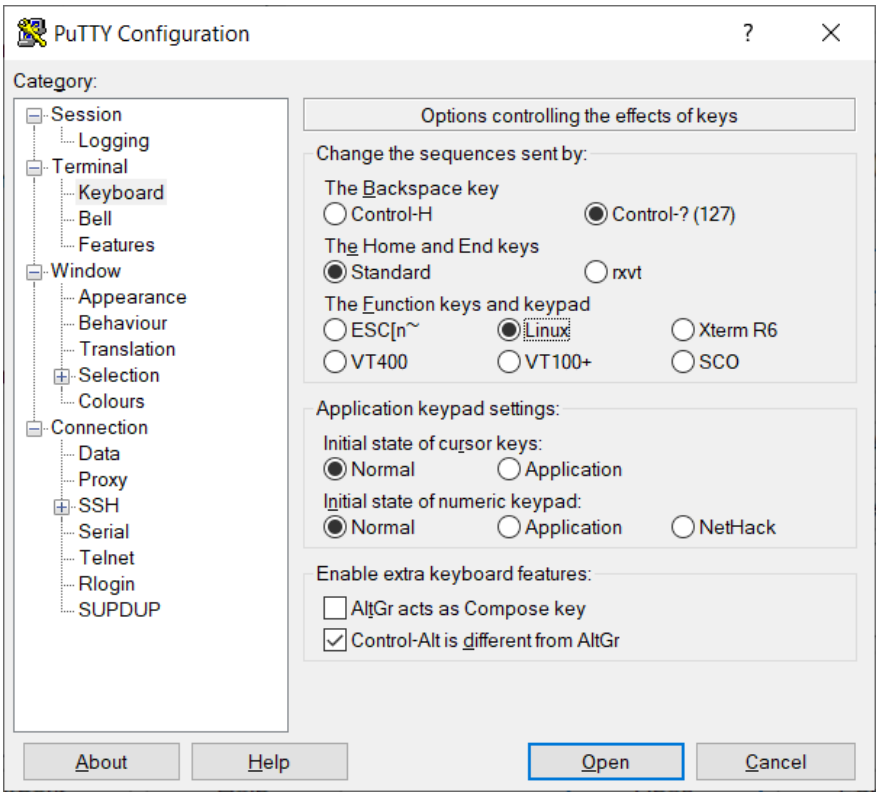

Рисунок 5. Изменение типа терминала

И отключить особую обработку клавиш цифровой клавиатуры.

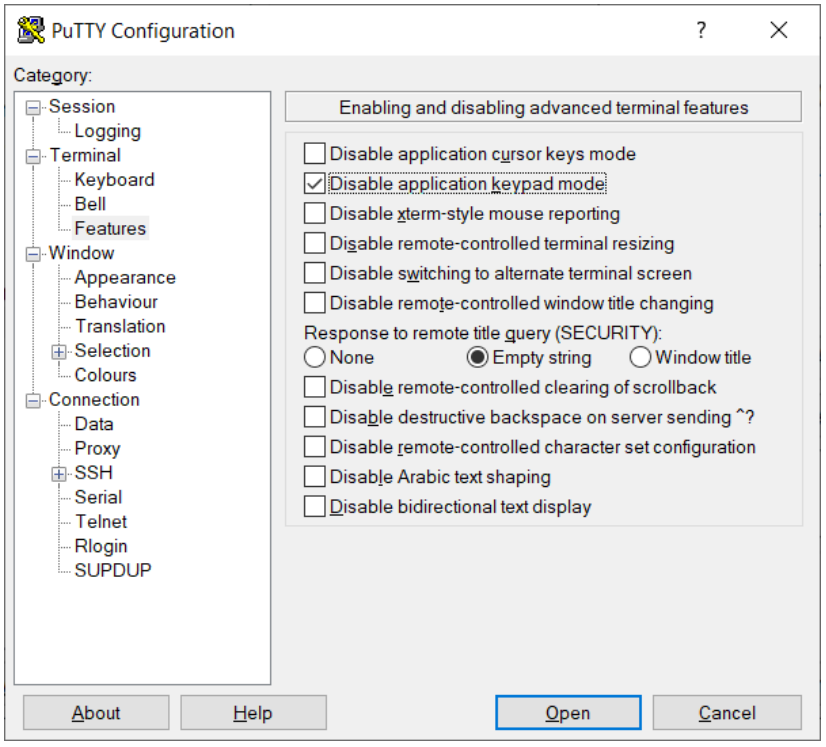

Рисунок 6. Отключение обработки клавиш клавиатуры

При первом подключении к серверу отобразится окно проверки цифровой подписи сервера.

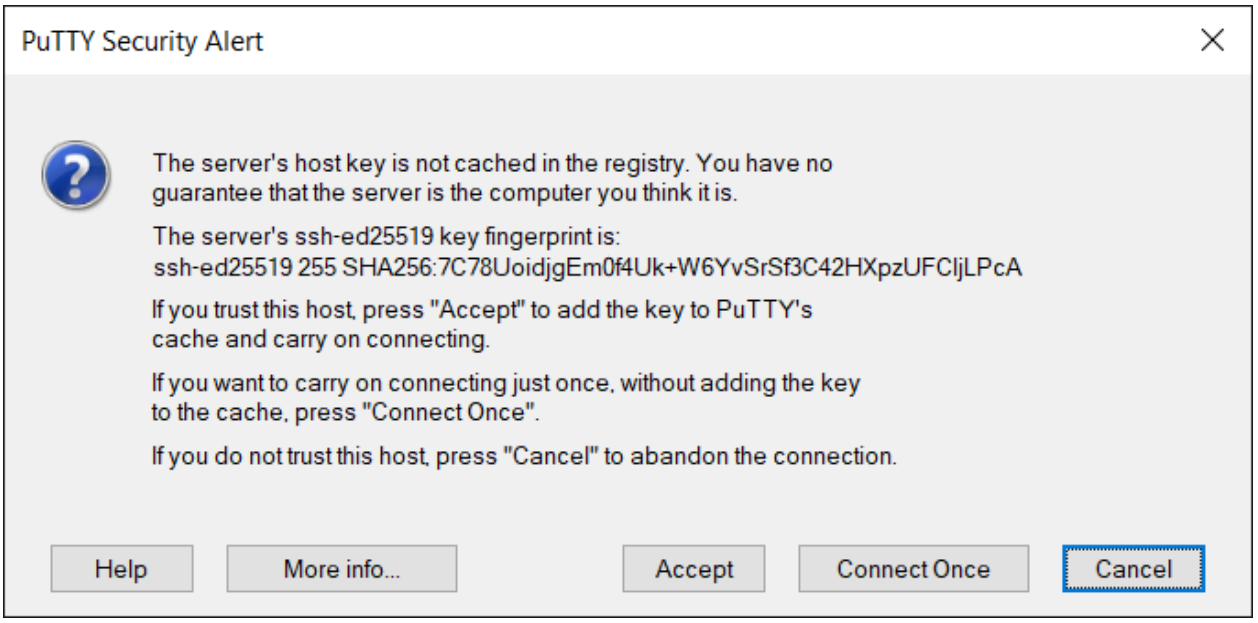

Рисунок 7. Окно проверки цифровой подписи сервера

Либо принимаем (Accept), либо соглашаемся на однократное подключение (Connect Once), в таком случае при следующем подключении окно будет показано повторно.

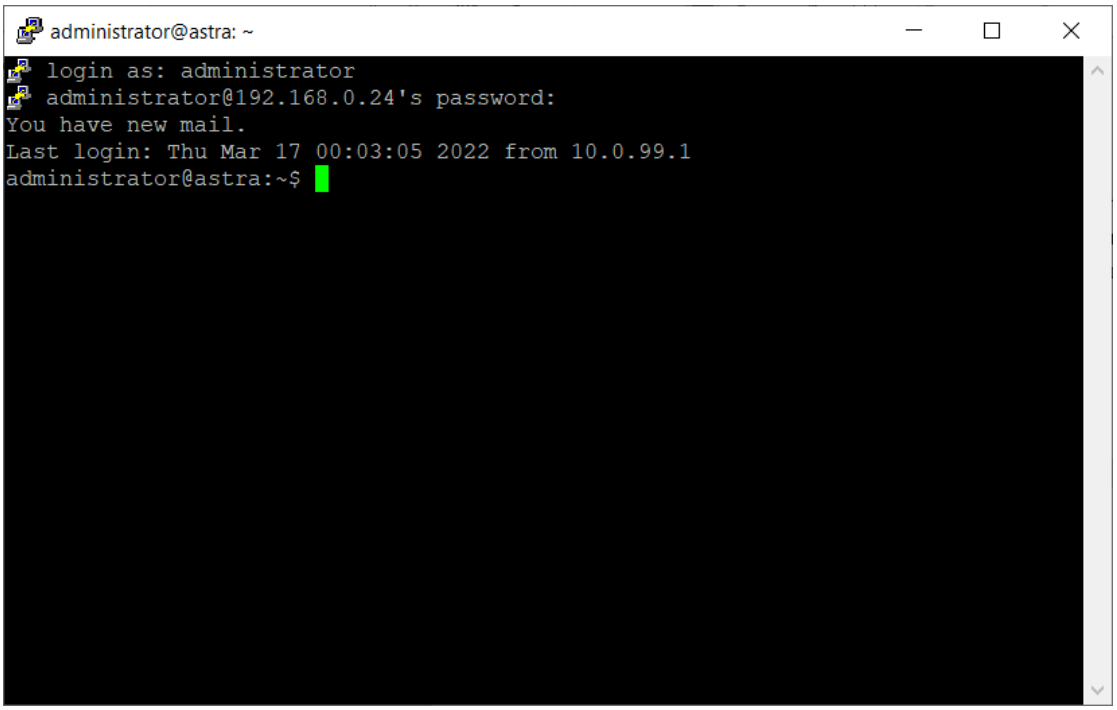

Рисунок 8. Окно PuTTY

Вводим логин и пароль, которые задали при установке.

Сгенерируйте себе ключевую пару (открытый и закрытый ключи) для SSH-доступа. Донастройте SSH-сервер (PubkeyAuthentication yes если не сделали вначале) и настройте SSH-клиент для аутентификации с помощью созданных ключей. Проверьте возможность доступа с использованием

ключей (без ввода пароля). Запретите на сервере вход по паролю, оставив только доступ по ключам. Проверьте полученную конфигурацию.

Для генерации ключей воспользуемся программой PuttyGen.

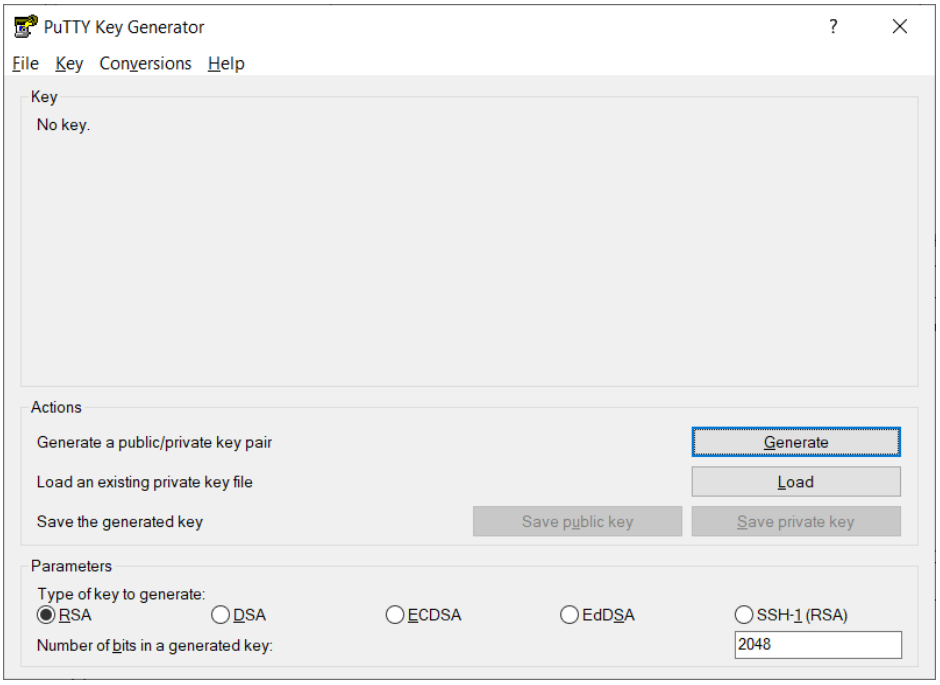

Рисунок 9. Приложение PuTTY Key Generator

Снизу выбирается тип ключа (EdDSA в нашем случае) и параметры (ED25519).

Нажатием кнопки Generate создаем ключевую пару.

Случайное шевеление мышкой в данном случае, как ни странно, заметно ускоряет процесс ☺.

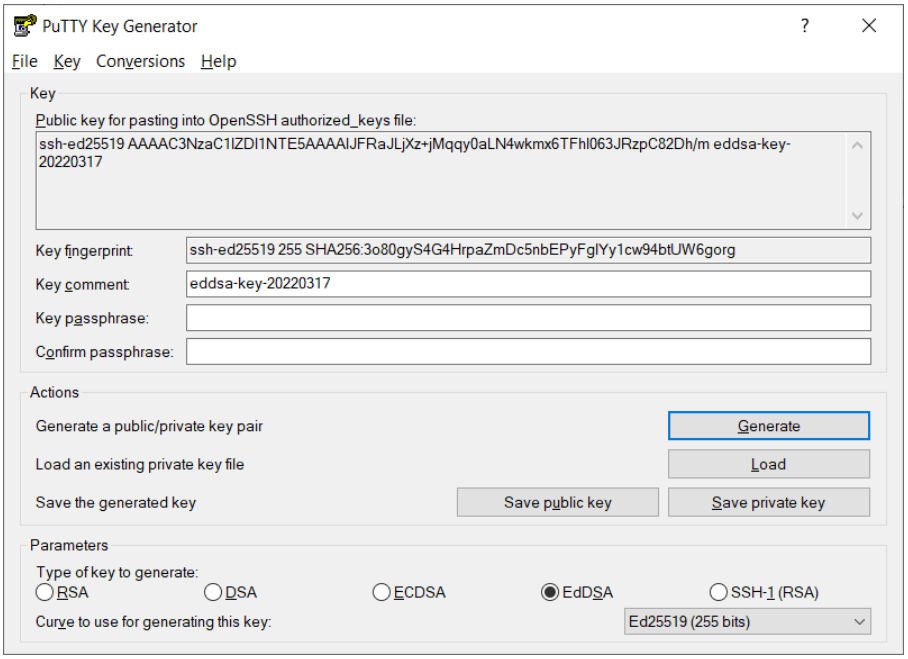

Рисунок 10. Результат работы PuTTY Key Generator

После генерации сохраняем ключ кнопкой Save private key

Ключ из верхней части окна копируем в буфер обмена и в открытом терминале выполняем команды

```
mkdir ~/.ssh
echo [вставить_ключ_сюда] >> ~/.ssh/authorized_keys
```

Аналогичные действия можно произвести и с помощью любого текстового редактора внутри виртуальной машины.

Закрываем сессию вводом команды logout

Подключаемся снова, указав адрес подключения, порт, имя пользователя и ключевой файл для аутентификации.

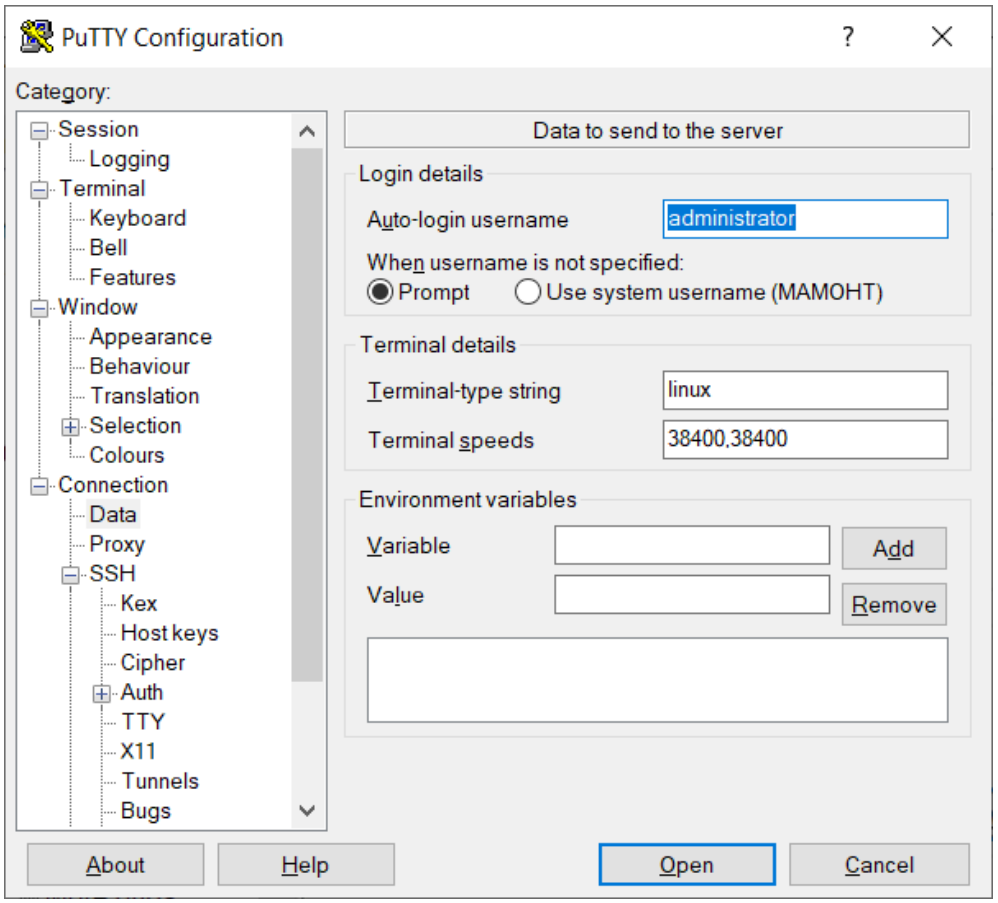

Рисунок 11. Ввод имени пользователя в PuTTY

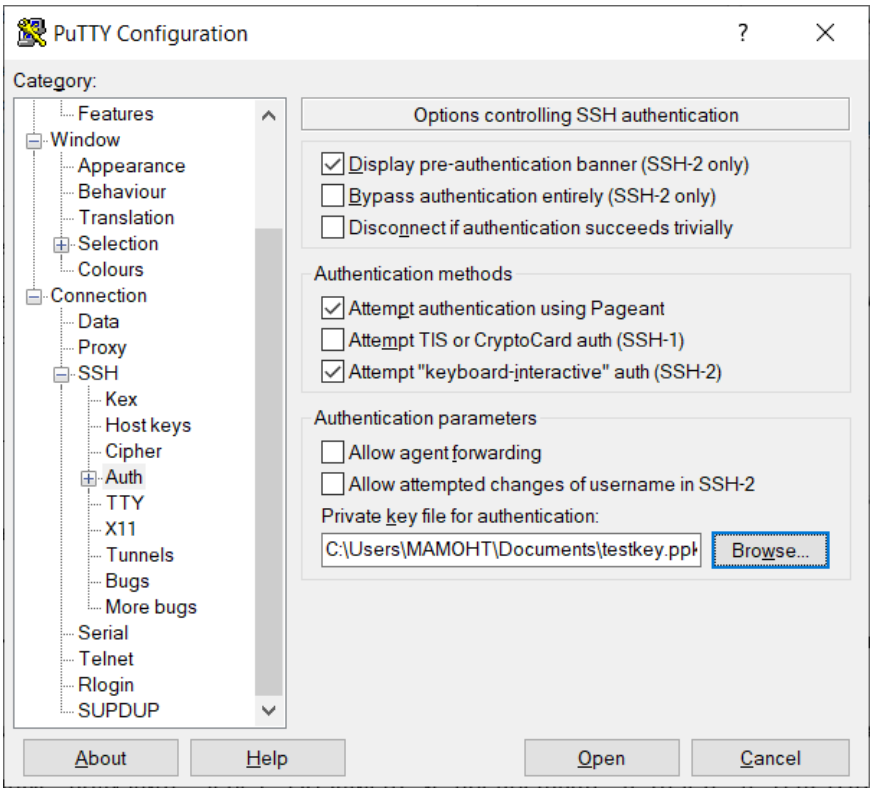

Рисунок 12. Выбор ключевого файла для аутентификации

Вход в систему должен произойти без запроса пароля. Если сервер сообщает об ошибке, и запрашивает пароль, значит, вы где-то ошиблись в процессе настройки.

Проведите базовую инвентаризацию компьютера, включая модель и возможности CPU, модель, серийный номер и производителя материнской платы, mac и ip адреса всех сетевых адаптеров.

Все данные получайте через SSH-клиент и предоставьте в отчете в ТЕКСТОВОМ виде (НЕ скриншоты). Копирование/вставка в ssh клиенте поддерживаются.

ПОДСКАЗКА:

```
cat /proc/cpuinfo
dmidecode --type baseboard
ip addr list
```

Задание 2)

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 255.255.255.0

Шлюз отсутствует

Перезапустите eth1 через ifdown eth1 и ifup eth1 (они не должны выдавать ошибок, если выдают - перезагрузите ВМ или попробуйте сделать ifdown eth1 и ifup eth1 еще несколько раз)

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 файлы /etc/default/isc-dhcp-server и /etc/dhcp/dhcpd.conf

1) Файл /etc/default/isc-dhcp-server

```
INTERFACESv4="eth1"
```

2) 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;
}
```

Разрешим работу протокола DHCP через файервол командами

```
ufw allow bootpc
```

```
ufw allow bootps
```

Запустим dhcpd командой

```
service isc-dhcp-server start
ИЛИ
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).

```
ping ya.ru
```

Разрешим подключение по SSH: для работы сетевых сервисов необходимо разрешить их порты (или профили) в файерволе, для ufw это делается командой

```
ufw allow ssh
```

Снова подключимся через PuTTY, проверим что всё корректно работает.

Заполните файл отчета «Шаблон для практической 7». Прикрепите его в СДО с названием «ПР7_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.

Вопросы для самоконтроля

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?