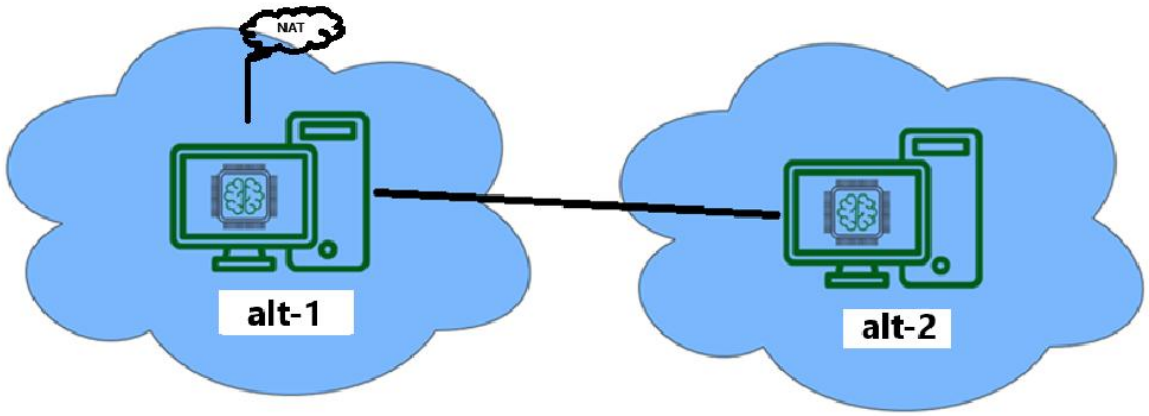
# Лабораторная работа №13
# Удаленный доступ

**Цель работы:**

Изучить процесс установки и настройки механизмов удалённого доступа в ОС Альт.

**Оборудование, ПО:**

- Alt1; Alt2
- Справочная литература или доступ в сеть интернет.




| Машина | IP | Маска | Шлюз | DNS |
|---|---|---|---|---|
| alt-1.test.ru<br>(Альт<br>Рабочая станция 10.4) | NAT<br>192.168.10.10 | -<br>24 | -<br>- | |
| alt-2.test.ru<br>(Альт<br>Рабочая станция 10.4) | 192.168.10.20 | 24 | 192.168.10.10 | - |

**Контрольные вопросы:**

- Что такое SSH и для чего он используется?
- Где находится конфигурационный файл SSH-сервера в Alt Linux?
- Почему рекомендуется отключать аутентификацию по паролю?
- Как ограничить доступ к SSH по IP-адресу?
- Как работает аутентификация в ssh по ключам?

**Контрольные задания:**

**После выполнения основной части лабораторной работы**

1. Запретите парольную аутентификацию, оставив только аутентификацию по ключам. Докажите это.
2. Настройте аутентификацию по ключам для всех пользователей в системе, а не только для пользователя user.
3. Настройте двухфакторную аутентификацию.

**Используемые источники:**

Официальная документация операционной системы Альт Рабочая Станция 10.4 https://docs.altlinux.org/ru-RU/alt-workstation/10.4/html/alt-workstation/index.html

SSH — сетевой протокол, используемый для удалённого управления операционными системами и передачи файлов. Аббревиатура расшифровывается как Secure Shell. Ключевая особенность заключается в том, что SSH шифрует трафик, делая подключения безопасными.

Реализован пакетом openssh (как серверная, так и клиентская части) и устанавливается по умолчанию.

Команда ssh позволяет установить безопасное подключение к удаленной системе, пройти аутентификацию от имени определенного пользователя и получить интерактивный сеанс командной оболочки в удаленной системе для этого пользователя. Кроме того, с помощью команды ssh можно выполнять отдельные команды в удаленной системе без запуска интерактивной оболочки.

Для подключения к удалённой системе используется команда ssh. В базовом виде команда имеет следующую форму:

**$ ssh host**

где host — IP-адрес или имя узла, к которому осуществляется подключение.

Эта команда предполагает, что имя пользователя на удалённой системе совпадает с именем текущего пользователя в локальной системе. Если это не так, можно указать имя пользователя на удалённой системе, используя следующий синтаксис:

**$ ssh user@host**

Для подключения к серверу, потребуется ввести пароль.

Чтобы завершить сеанс ssh и вернуться в сеанс локальной оболочки, следует ввести команду:

**$ exit**

В ОС Альт сервер и клиент openssh установлены по умолчанию.

| |
|---|
| От имени суперпользователя проверьте статус демона sshd, а далее запустите и добавьте в автозапуск его командой:<br><br>**systemctl enable --now sshd.service** |
| |

*Файл конфигурации сервера*

/etc/openssh/sshd_config — файл конфигурации сервера OpenSSH. Для применения изменений, внесённых в этот файл, необходимо перезапустить сервер.

Стандартная конфигурация сервера OpenSSH работает хорошо. Однако вы можете внести некоторые изменения для повышения безопасности системы. Например, можно запретить удаленный вход напрямую в учетную запись root, а также запретить аутентификацию на основе пароля (заменив ее аутентификацией на основе закрытого ключа SSH).

| Посмотрите содержимое данного конфигурационного файла |
| --- |
| |

В рамках примера настройки ssh-ссервера на ВМ alt-1.test.ru сменим ее рабочий порт на 2222.

```
# This is the sshd server system-wide configuration file.  See
# sshd_config(5) for more information.

# This sshd was compiled with PATH=/bin:/usr/bin:/usr/local/bin

# The strategy used for options in the default sshd_config shipped with
# OpenSSH is to specify options with their default value where
# possible, but leave them commented.  Uncommented options override the
# default value.

Port 2222
#AddressFamily any
#ListenAddress 0.0.0.0
#ListenAddress ::

#HostKey /etc/openssh/ssh_host_rsa_key
#HostKey /etc/openssh/ssh_host_ecdsa_key
#HostKey /etc/openssh/ssh_host_ed25519_key

# Ciphers and keying
```

| Смените порт. Перезапустите сервер командой |
| --- |
| |

| |
|---|
| **systemctl restart sshd.service**<br><br>Докажите, что сервер работает на новом порту. |
| |

Опишите подробно следующие параметры, использующиеся в параметрах ssh-сервера.

| | |
|---|---|
| AddressFamily | |
| ListenAddress | |
| PermitRootLogin | |
| StrictModes | |
| MaxSessions | |
| MaxAuthTries | |
| PubkeyAuthentication | |
| GSSAPI | |
| KerberosAuthentication | |

После перезапуска подключитесь с машины alt-2.test.ru на машину alt-1.test.ru при помощи команды
ssh user@192.168.10.10

| |
|---|
| Приведите результат работы команды |
| |

ssh user@192.168.10.10 -p 2222

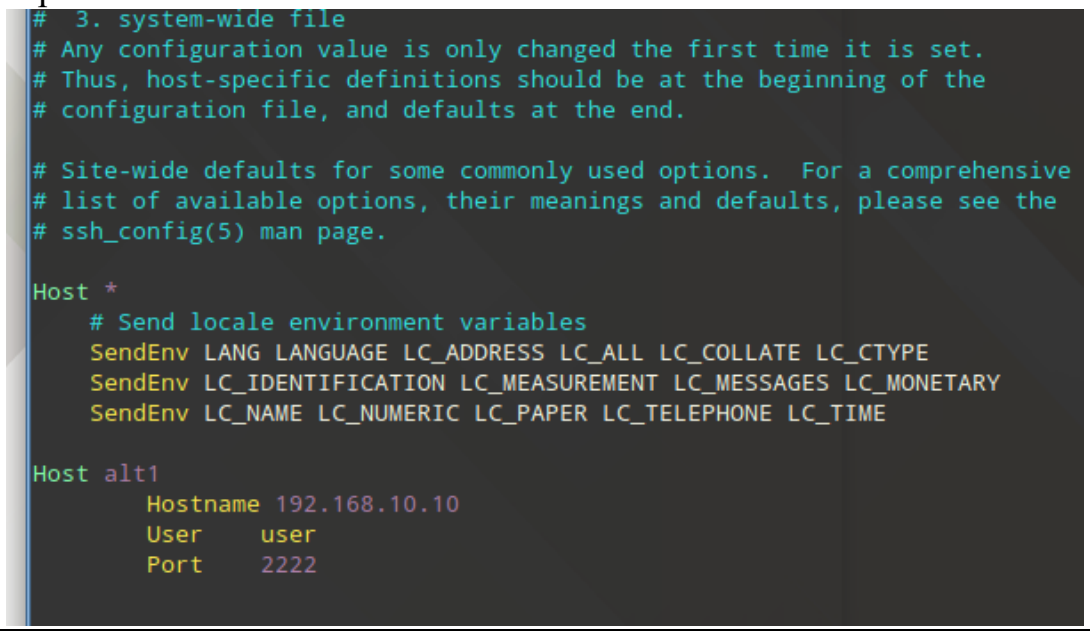
| |
|---|
| Приведите результат работы команды |
| |

Для настройки ssh-клиента используется файл /etc/openssh/ssh_config

| |
|---|
| Посмотрите содержимое данного конфигурационного файла |
| |

Файл ssh_config содержит глобальные настройки для SSH-клиента в системе Alt Linux. Он определяет параметры подключения по умолчанию для всех пользователей.

Создадим свое собственное подключение alt1 со следующими параметрами:




| |
|---|
| На машине **alt2** создайте новое подключение. Проверьте работоспособность командой:<br><br>**ssh alt1** |
| |

Опишите подробно следующие параметры, использующиеся в параметрах ssh-клиента.

| | |
|---|---|
| Port | |
| User | |
| IdentityFile | |
| Protocol | |
| StrictHostKeyChecking | |
| ForwardX11 | |
| Compression | |

## Ключи хостов SSH

SSH защищает связь, используя шифрование с открытым ключом. Когда клиент SSH подключается к SSH-серверу, сервер отправляет копию открытого ключа клиенту до входа клиента в систему. Это необходимо для настройки безопасного шифрования канала связи и выполнения аутентификации между клиентом и сервером.

Когда пользователь выполняет команду **ssh** для подключения к серверу SSH, команда проверяет наличие копии открытого ключа для этого сервера в локальных файлах известных хостов. Системный администратор может заранее настроить его в файле /etc/openssh/ssh_known_hosts, или у пользователя может быть файл ~/.ssh/known_hosts с этим ключом в домашнем каталоге.

Если у клиента есть копия ключа, команда **ssh** будет сравнивать ключ из файлов известных хостов для этого сервера с полученным ключом. Если ключи не совпадают, клиент предполагает, что сетевой трафик к серверу перехватывается или сервер был скомпрометирован, и запрашивает у пользователя подтверждение, продолжать подключение или нет.

## Аутентификация на основе ключей SSH

Вы можете настроить SSH-сервер на прохождение аутентификации без пароля с помощью ключей. Такая аутентификация основана на схеме «закрытый-открытый ключ».

Необходимо создать пару связанных файлов криптографических ключей. Один из них — это закрытый ключ, другой — открытый ключ. Файл закрытого ключа используется в качестве учетных данных аутентификации и, как пароль, должен храниться в секрете и быть защищен. Открытый ключ копируется в системы, к которым хочет подключиться пользователь, и используется для проверки закрытого ключа. Открытый ключ не обязательно держать в секрете.

Вы помещаете копию открытого ключа в свою учетную запись на сервере. Когда вы пытаетесь войти в систему, SSH-сервер может использовать открытый ключ, чтобы отправить запрос, на который можно правильно ответить закрытым ключом. В результате ваш клиент **ssh** может автоматически проходить аутентификацию на сервере с помощью уникальной копии закрытого ключа. Это позволяет вам безопасно входить в системы без интерактивного ввода пароля.

**Настроим аутентификацию по ключам. На ВМ alt2 создадим ключи.**

Чтобы создать закрытый ключ и соответствующий открытый ключ для аутентификации, используйте команду **ssh-keygen**. По умолчанию закрытые и открытые ключи хранятся в файлах ~/.ssh/id_rsa и ~/.ssh/id_rsa.pub соответственно.

| |
|---|
| Выполните команду<br><br>**ssh-keygen** |
| |

Если вы не укажете парольную фразу, когда команда **ssh-keygen** предложит сделать это, сформированный закрытый ключ не будет защищен. В этом случае любой пользователь с вашим файлом закрытого ключа сможет использовать его для аутентификации. Если вы установите парольную фразу, вам потребуется вводить ее при прохождении аутентификации с помощью закрытого ключа. (Поэтому вы будете использовать на удаленном хосте парольную фразу закрытого ключа, а не пароль для прохождения аутентификации.)

**Передача открытого ключа**

Для использования аутентификации на основе ключей необходимо скопировать открытый ключ в целевую систему. Команда **ssh-copy-id** копирует открытый ключ из пары ключей SSH в целевую систему. Если вы не укажете путь к файлу открытого ключа при выполнении команды **ssh-copy-id**, будет использоваться файл /home/[пользователь]/.ssh/id_rsa.pub по умолчанию.

| |
|---|
| Выполните команду<br><br>**ssh-copy-id user@192.168.10.10** |

| |
|---|
| |

Объясните, почему данная команда не работает. P.S. обратите внимание на ключ **-p**

После успешной передачи открытого ключа в удаленную систему можно будет входить в эту систему через SSH, проходя аутентификацию по соответствующему закрытому ключу. Если вы не укажете путь к файлу закрытого ключа при выполнении команды **ssh**, будет использоваться файл /home/*user*/.ssh/id_rsa по умолчанию.

| |
|---|
| На машине alt1 покажите переданный открытый ключ |
| |

## xRDP

XRDP (X Remote Desktop Protocol) — это открытая реализация сервера удалённого рабочего стола, использующая протокол RDP (Remote Desktop Protocol), первоначально разработанный Microsoft. Основное назначение XRDP:
Обеспечение графического доступа к Linux-системам
Поддержка стандартных RDP-клиентов
Интеграция с различными дисплейными серверами (X11, Wayland)

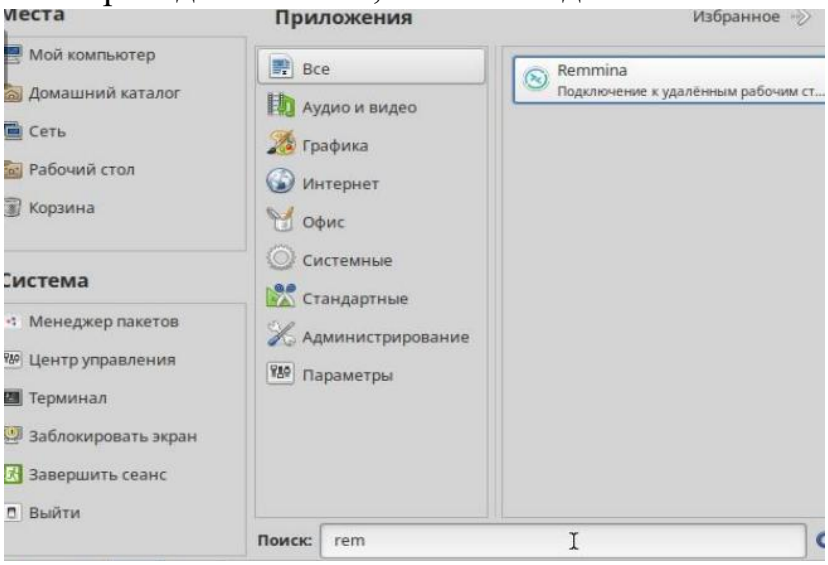
## На alt2

1. Устанавливаем xrdp на сервер –
   **apt-get install xrdp**
2. Включаем и добавляем в автозапуск –
   systemctl enable xrdp
3. Создадим пользователя для подключения –
   **useradd test**
4. Поставим ему пароль –
   **passwd test** (ставим 123)
5. Настраиваем права доступа. Для доступа к терминальному сеансу – включить в группу tsusers –
   **gpasswd -a test tsusers**
6. Для проброса папки включить в группу fuse –
   **gpasswd -a test fuse**
7. Для проброса USB-устройств необходимо установить пакет
   **xrdp-usb- session**
8. Просмотрим настройки сервера –
   **vim /etc/xrdp/sesman.ini**

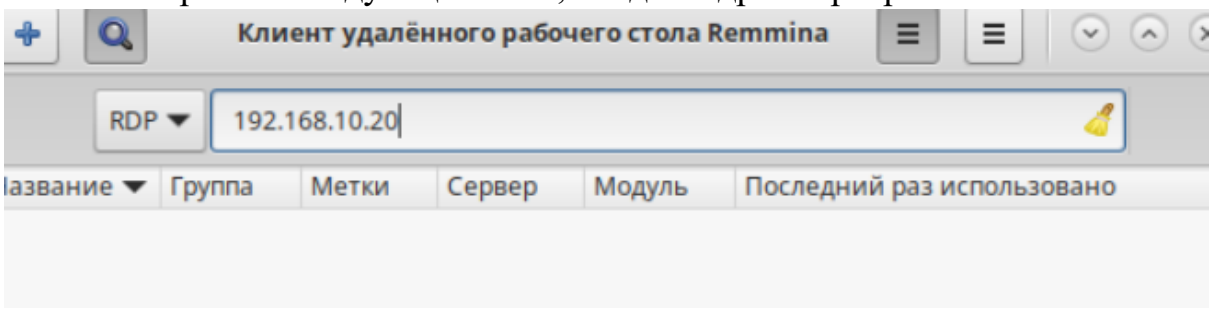
Настройки по умолчанию:
AllowRootLogin=true — авторизация Root;
MaxLoginRetry=4 — максимальное количество попыток подключения;
TerminalServerUsers=tsusers — группа, в которую необходимо добавить пользователей для организации доступа к серверу;
MaxSessions=50 — максимальное количество подключений к серверу;
KillDisconnected=false — разрыв сеанса при отключении пользователя;
FuseMountName=Mount_FOLDER — название монтируемой папки.
9. По умолчанию используется порт 3389, который можно изменить в /etc/xrdp/xrdp.ini
10. Вкючаем xrdp командой: **systemctl start xrdp**

**На alt1**

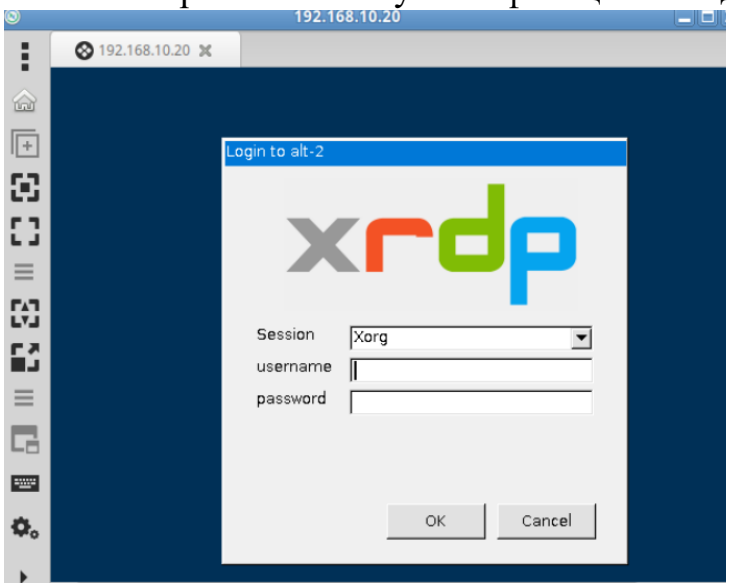
11. Настроим клиента. Установим Remina –
**apt-get install remmina remmina-plugins-rdp**
12. Переходим в Меню, затем находим Remina:




13. Нас встречает следующее окно, вводим адрес сервера:




14. Нас встречает окно аутентификации. Вводим логин и пароль (test 123)




14. Если все хорошо, то у нас откроется окно с удаленной машиной (если не получилось, то попробуйте перезапустить сервер)

**VNC**

VNC (Virtual Network Computing) – система удалённого доступа к графическому рабочему столу, использующая протокол RFB (Remote Frame Buffer). Основные особенности:
Кроссплатформенность (поддержка Linux, Windows, macOS)
Передача растрового изображения экрана
Работа на уровне фреймбуфера (не зависит от оконной системы)

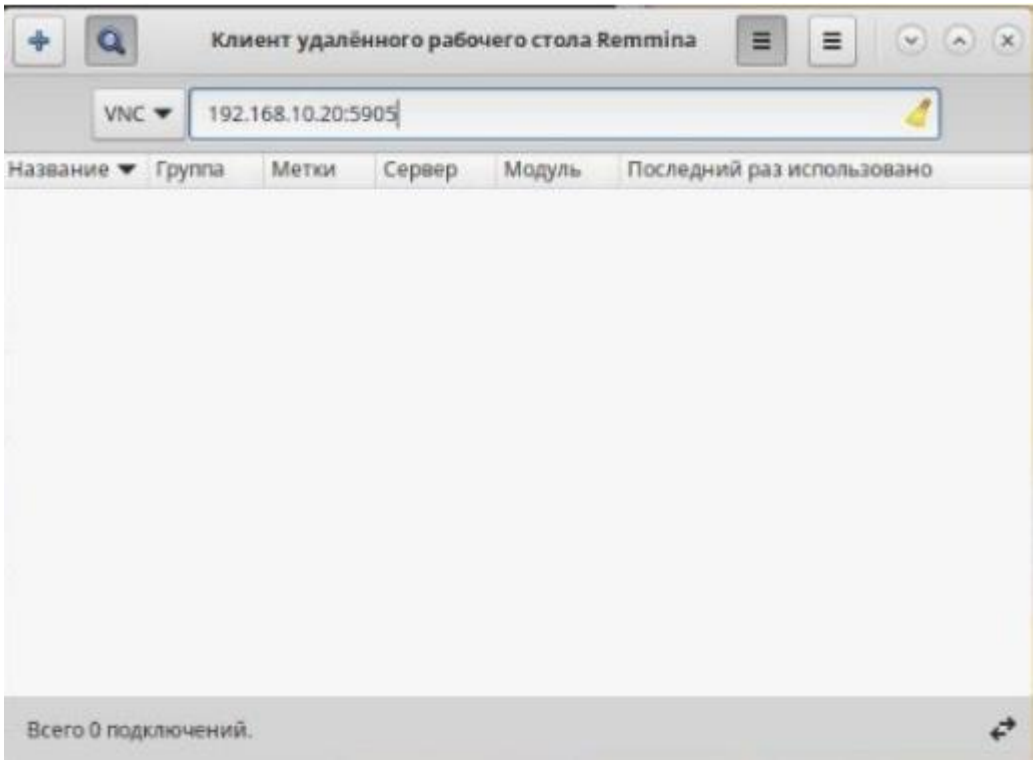
1. Установим x11vnc на сервер –
   **apt-get install x11vnc**
2. Вводим в строку терминала следующее – **x11vnc**
3. На клиенте устанавливаем remmina – apt-get install remmina remmina-plugins-vnc
4. Перейдем к настройкам сервера. Можно поставить пароль на сервер следующей командой (ставим 123) – **x11vnc –storepasswd**

```
x11vnc --storepasswd
Enter VNC password:
Verify password:
Write password to /root/.vnc/passwd?  [y]/n y
Password written to: /root/.vnc/passwd
```

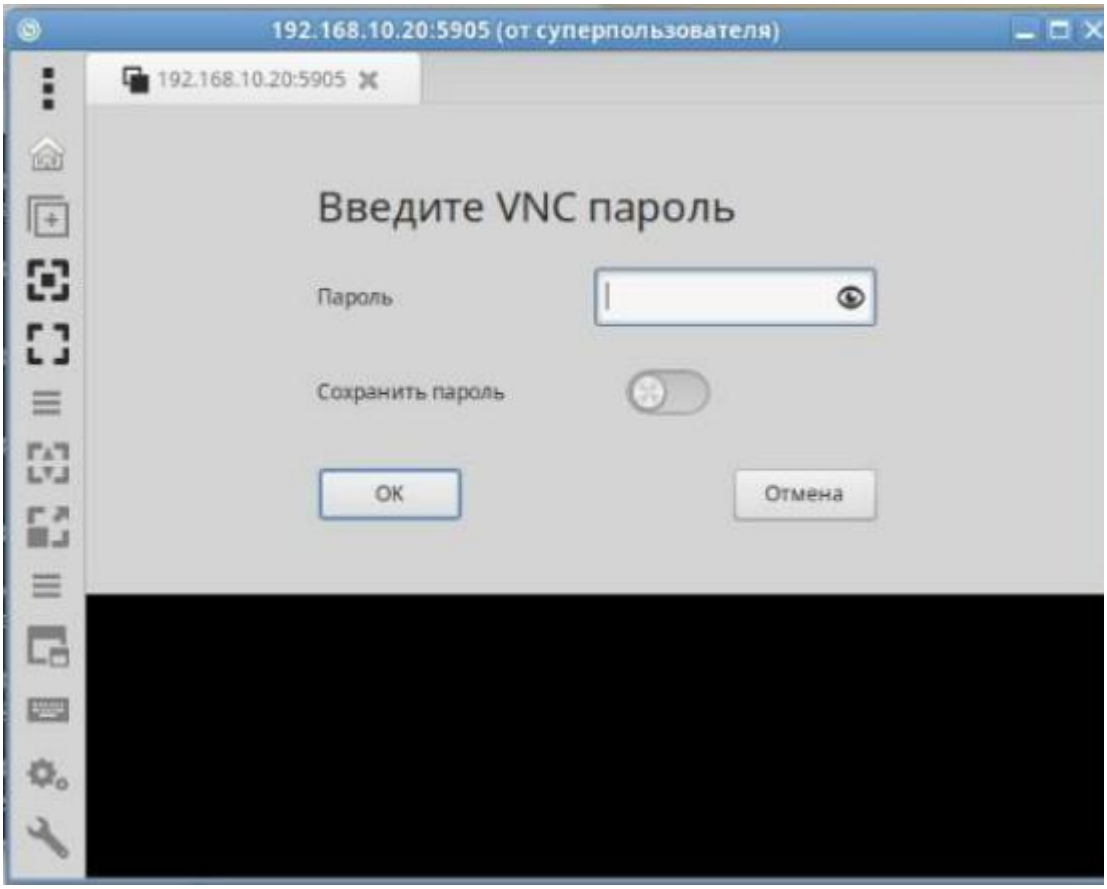
5. Откроем соединения на сервере –

**x11vnc -rfbauth .vnc/passwd -rfbport 5905**

6. Попробуем подключиться к серверу средствами remmina:



7. Вводим пароль 123




| |
|---|
| Покажите пример настройки x11vnc через графику |
| |