{jcomments on}SSH, OPENSSH-SERVER, SCP, ключи шифрования.

SSH (англ. Secure SHell — «безопасная оболочка») — сетевой протокол сеансового уровня, позволяющий производить удалённое управление операционной системой и туннелирование TCP-соединений (например, для передачи файлов). Сходен по функциональности с протоколами Telnet и rlogin, но, в отличие от них, шифрует весь трафик, включая и передаваемые пароли.

OpenSSH (открытая безопасная оболочка) — набор программ, предоставляющих шифрование сеансов связи по компьютерным сетям с использованием протокола SSH.

В установке никаких сложностей нет:

sudo apt-get install openssh-server

и можно подключаться к серверу:

ssh 10.10.10.4

Вводим логин и пароль учетной записи, имеющейся на сервере, и мы внутри.

для отсоединения:

exit

scp - утилита для передачи файлов по ssh.

Копируем файл на сервер 10.10.10.4 через 22ой порт:

scp -P 22 /home/virtdiver/test.txt it@10.10.10.4:/home/it/test.txt

Копируем файл с сервера 10.10.10.4 через 22ой порт:

scp -P 22 it@10.10.10.4:/home/it/test.txt /home/virtdiver/test.txt

копируем папку test на сервер 10.10.10.4 через 22ой порт:

scp -P 22 -r /home/virtdiver/test it@10.10.10.4:/home/it

Копируем папку test с сервера 10.10.10.4 через 22ой порт:

scp -P 22 -r it@10.10.10.4:/home/it/test /home/virtdiver

Если возникнет необходимость передавать файл, содержащий пробелы в названии, то путь нужно поместить в кавычки, и экранировать пробелы слешем (). В примере ниже передается файл с именем "filename with space.txt" (За эту информацию спасибо **zbl**):

scp -P 22 'it@10.10.10.4:/home/it/filename with space.txt' '/home/virtdiver/filename with space.txt'

Генерация SSH ключей

В первую очередь, нужно создать пару ключей, если их ещё нет. По умолчанию ключи хранятся в домашней директории пользователя, /home/имя пользователя/.ssh/id_rsa и id_rsa.pub. Ключи должны генериться на клиенте именно в той учетке, из которой будете заходить на сервер по ssh. Меняем пользователя на нужного (если необходимо):

sudo su postgres

Генерируем ключи:

ssh-keygen

путь можно оставить по умолчанию, просто жмем Enter;

вводим пароль, если нужно, если не нужно, жмем Enter.

Пара ключей готова. Мне авторизация при помощи ключей нужна была для автоматического слива бэкапа базы данных с двух серверов на третий посредством scp, для специально созданного пользователя postgres, поэтому пароль я оставил пустым.

Далее нужно скопировать ключ на ssh-сервер:

scp ~/.ssh/id_rsa.pub postgres@10.10.10.4:.ssh/authorized_keys2

Проверяем:

ssh 10.10.10.4

Наблюдаем, что пароль не запросился.

Для других ключи генерятся аналогично, но ключ теперь копировать на сервер не надо, необходимо дописать существующий на ssh-сервере, данными из сгенерированного:

cat ~/.ssh/id_rsa.pub | ssh postgres@10.10.10.4 "cat >> .ssh/authorized_keys2"

В итоге имеем доступ без пароля на 10.10.10.4 с авторизацией, использующей ключи шифрования, для пользователя postgres с обоих серверов.

А теперь о том, из-за чего я убил целый час. Если сгенерировать ключик под windows с помощью PuTTYgen и аналогично добавить его на сервер, то через putty с этим ключем мы зайдем на сервак, а с линукса с этим же ключем - нет, так как у putty и openssh разные форматы. Если файл ключа создан в putty и нужно сконвертировать в формат openssh то ставим:

sudo apt-get install putty-tools

И конвертируем:

puttygen -O private-openssh /home/virtdiver/putty/priv_key.ppk -o /home/virtdiver/putty/priv_key

где priv_key.ppk - файл формата putty , priv_key - получаемый файл ключа в формате openssh.

И обратная процедура, из формата openssh в формат putty:

puttygen /home/virtdiver/putty/priv_key -o /home/virtdiver/putty/priv_key.ppk