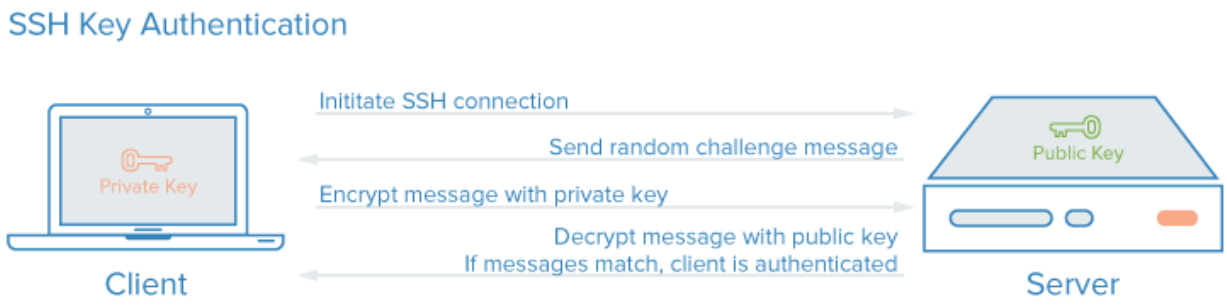
Как работают ключи SSH?

SSH сервер может выполнять аутентификацию пользователей с помощью различных алгоритмов. Самый популярный - это аутентификация по паролю. Он достаточно прост, но не очень безопасный. Пароли передаются по безопасному каналу, но они недостаточно сложны для противостояния попыткам перебора. Вычислительная мощность современных систем в сочетании со специальными скриптами делают перебор очень простым. Конечно, существуют другие способы дополнительной безопасности, например, fail2ban, но аутентификация по ключу SSH более надежна.

Каждая пара ключей состоит из открытого и закрытого ключа. Секретный ключ сохраняется на стороне клиента и не должен быть доступен кому-либо еще. Утечка ключа позволит злоумышленнику войти на сервер, если не была настроена дополнительная аутентификация по паролю.

Открытый ключ используется для шифрования сообщений, которые можно расшифровать только закрытым ключом. Это свойство и используется для аутентификации с помощью пары ключей. Открытый ключ загружается на удаленный сервер, к которому необходимо получить доступ. Его нужно добавить в специальный файл ~/.ssh/authorized_keys.

Когда клиент попытается выполнить проверку подлинности через этот ключ, сервер отправит сообщение, зашифрованное с помощью открытого ключа, если клиент сможет его расшифровать и вернуть правильный ответ - аутентификация пройдена.

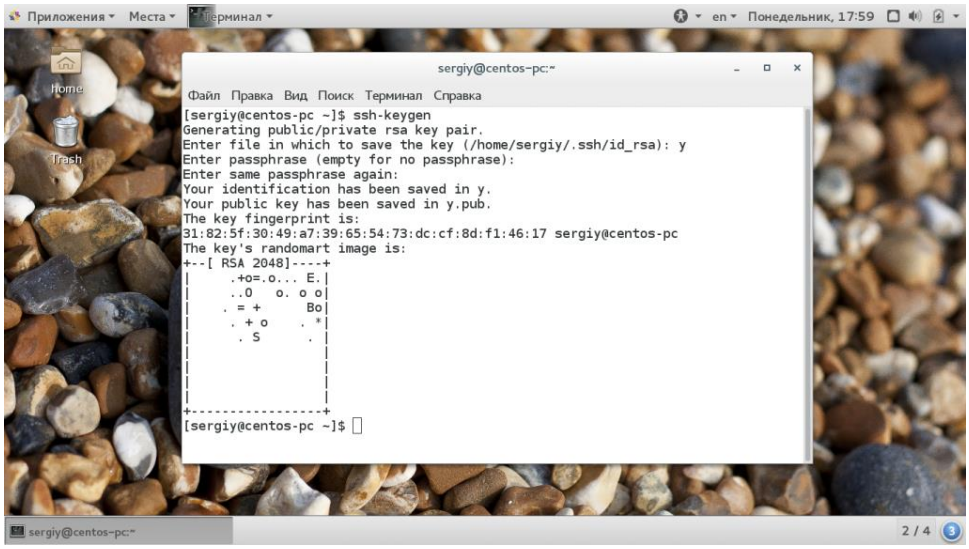


Как создать ключи SSH?

Сначала необходимо создать ключи ssh для аутентификации на локальном сервере. Для этого существует специальная утилита ssh-keygen, которая входит в набор утилит OpenSSH. По умолчанию она создает пару 2048 битных RSA ключей, которая подойдет не только для SSH, но и для большинства других ситуаций.

И так, генерация ключей ssh выполняется командой:

```
ssh-keygen
```



Утилита предложит вам выбрать расположение ключей. По умолчанию ключи располагаются в папке ~/.ssh/. Лучше ничего не менять, чтобы все работало по умолчанию и ключи автоматически подхватывались. Секретный ключ будет называться id_rsa, а публичный id_rsa.pub.

Затем утилита предложит ввести пароль для дополнительного шифрования ключа на диске. Его можно не указывать, если не хотите. Использование дополнительного шифрования имеет только один минус - необходимость вводить пароль, и несколько преимуществ:

- Пароль никогда не попадет в сеть, он используется только на локальной машине для расшифровки ключа. Это значит что перебор по паролю больше невозможен.
- Секретный ключ хранится в закрытом каталоге и у клиента ssh нет к нему доступа пока вы не введете пароль;
- Если злоумышленник хочет взломать аутентификацию по ключу SSH, ему понадобится доступ к вашей системе. И даже тогда ключевая фраза может стать серьезной помехой на его пути.

Но все же, это необязательное дополнение и если не хотите, то вы можете просто нажать Enter. Тогда доступ по ключу ssh будет выполняться автоматически и вам не нужно будет что-либо вводить.

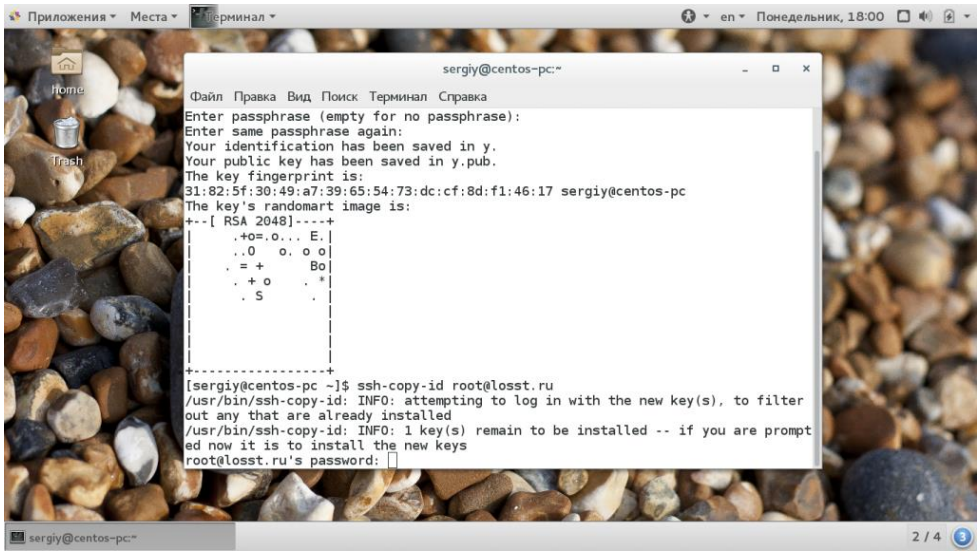
Теперь у вас есть открытый и закрытый ключи SSH и вы можете использовать их для проверки подлинности. Дальше нам осталось разместить открытый ключ на удаленном сервере.

Загрузка ключа на сервер

Когда генерация ключей завершена, нам осталось только загрузить ключ на сервер. Для загрузки ключа можно использовать несколько способов. В некоторых случаях вы можете указать ключ в панели управления сервером, например, cPanel или любой другой. Но мы такой способ рассматривать не будем. Мы рассмотрим ручные способы.

Самый простой способ скопировать ключ на удаленный сервер - это использовать утилиту ssh-copy-id. Она тоже входит в пакет программ OpenSSH. Но для работы этого метода вам нужно иметь пароль доступа к серверу по SSH. Синтаксис команды:

```
ssh-copy-id username@remote_host
```



При первом подключении к серверу система может его не распознать, поэтому вам нужно ввести yes. Затем введите ваш пароль пользователя на удаленном сервере. Утилита подключится к удаленному серверу, а затем использует содержимое ключа id.rsa.pub для загрузки его на сервер в файл ~/.ssh/authorized_keys. Дальше вы можете выполнять аутентификацию с помощью этого ключа.

Если такой способ по какой-либо причине для вас не работает, вы можете скопировать ключ по ssh вручную. Мы создадим каталог ~/.ssh, а затем поместим наш ключ в файл authorized_keys с помощью символа >>, это позволит не перезаписывать существующие ключи:

```
cat ~/.ssh/id_rsa.pub | ssh username@remote_host "mkdir -p ~/.ssh && cat >> ~/.ssh/authorized_keys"
```

Здесь вам тоже нужно набрать yes, если вы подключаетесь к новому серверу, а затем ввести пароль. Теперь вы можете использовать созданный ключ для аутентификации на сервере:

```
ssh username@remote_host
```

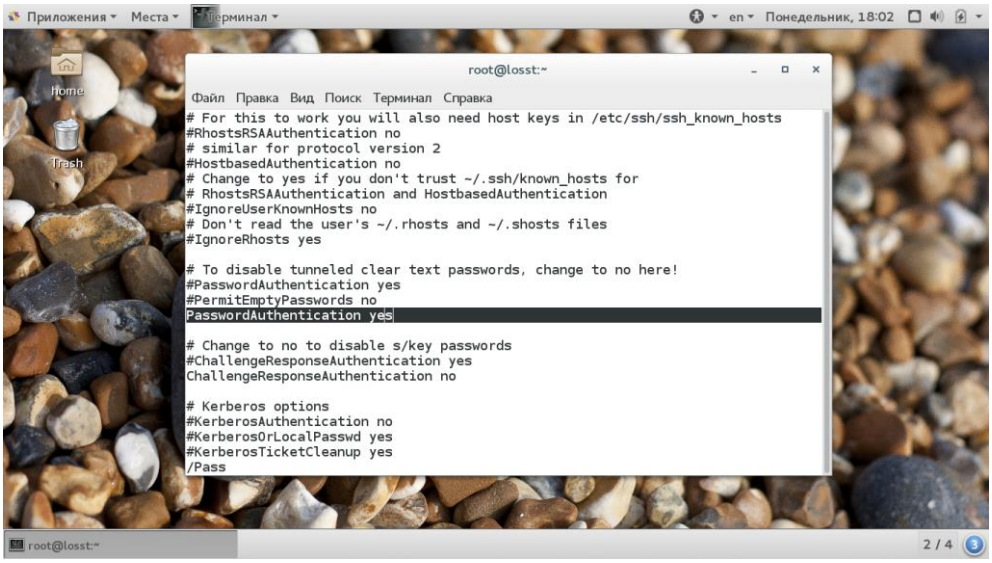
Если вы не захотели создать ssh ключ с доступом по паролю, то вы сразу же будете авторизованы, что очень удобно. Иначе, сначала вам придется ввести фразу-пароль для расшифровки ключа.

Отключение проверки пароля

Если пароль больше не будет использоваться, то для увеличения безопасности системы лучше его вовсе отключить. Но убедитесь, что ключ надежно сохранен и вы его не потеряете, потому что по паролю вы больше не войдете. Авторизуйтесь на сервере, затем откройте конфигурационный файл /etc/ssh/sshd_config и найдите там директиву PasswordAuthenticatin. Нужно установить ее значение в No:

sudo vi /etc/ssh/sshd_config

PasswordAuthentication no



Теперь сохраните файл и перезапустите службу ssh:

sudo service ssh restart

Дальше будет возможно только подключение по ключу ssh, пароль не будет приниматься.