Руководство по установке и настройке SSH на сервере с Ubuntu

SSH (Secure Shell) представляет собой сетевой протокол, спроектированный для обеспечения безопасной связи между клиентом и сервером. Каждая отправляемая команда и полученная информация шифруются, что гарантирует безопасное управление удаленным хостом, передачу файлов и решение других задач.

Для успешной настройки SSH соединения вам следует выполнить следующие шаги:

1. Установить компоненты SSH-сервера на серверной машине. Для этого используйте пакет openssh-server, который обеспечивает функциональность сервера SSH.
2. Убедитесь, что на клиентской машине присутствует клиентский компонент SSH. В большинстве дистрибутивов Linux и BSD он уже предустановлен, однако для операционной системы Windows может потребоваться установка дополнительных утилит. Один из наиболее распространенных вариантов для Windows – это PuTTY, который предоставляет возможность установки SSH-соединения с удаленным хостом.

Активация SSH

Изначально, для безопасности, удаленный доступ к серверу через защищенный сетевой протокол отключен по умолчанию. Тем не менее, в Ubuntu настройка SSH выполняется легко и просто. Для начала, откройте консоль на сервере, на котором требуется активировать SSH в Ubuntu.

Далее, выполните обновление пакетного менеджера для обеспечения актуальности системы.

```
apt update
```

Для установки необходимого программного обеспечения, выполните следующее:

```
sudo apt install openssh-server
```

Обратите внимание: для выполнения обеих операций необходимы права суперпользователя, которые можно получить, используя sudo. По умолчанию, в Ubuntu, OpenSSH запускается автоматически после установки. Однако, если вам нужно проверить его текущий статус, вы можете воспользоваться командой:

```
sudo systemctl status ssh
```

При выполнении проверки статуса службы OpenSSH в выводе должно отображаться, что служба запущена и настроена на автозапуск при загрузке системы:

```
ssh.service - OpenBSD Secure Shell server
    Loaded: loaded (/lib/systemd/system/ssh.service; enabled; vendor preset: enabled)
    Active: active (running) since Mon 2022-03-21 12:34:00 CEST; 1m ago
...
```

Это означает, что команда “Install SSH Ubuntu” выполнилась успешно. Чтобы вернуться в командную строку, нажмите клавишу “q”.

Если служба не активирована, вы можете запустить ее вручную, воспользовавшись следующей командой:

```
sudo systemctl enable --now ssh
```

В Ubuntu встроен инструмент управления брандмауэром, известный как UFW. Если брандмауэр активен в вашей системе, не забудьте разрешить доступ к порту SSH:

```
sudo ufw allow ssh
```

Теперь у вас есть возможность установить SSH-соединение с вашей системой Ubuntu с помощью любого удаленного компьютера.

Формирование пары ключей

Для усиления безопасности соединения и обеспечения удобной аутентификации, используется ключевая пара, состоящая из открытого и закрытого ключей. Открытый ключ хранится на хосте, в то время как закрытый ключ — на компьютере пользователя.

Давайте рассмотрим процесс создания ключей на разных операционных системах. Начнем с Ubuntu.

Для генерации новой пары ключей RSA с длиной 2048 бит вам потребуется открыть терминал и выполнить следующую команду:

```
ssh-keygen -t rsa
```

При создании пары ключей возникнет вопрос о месте их сохранения. Если вы просто нажмете Enter, ключи будут автоматически сохранены в подкаталоге .ssh в вашей домашней папке. Вы также можете выбрать другой путь для сохранения, но наилучшей практикой является использование каталога по умолчанию, так как это упрощает управление ключами в будущем.

Если на клиентском компьютере уже существует пара ключей, вам будет предложено перезаписать их. Решение о перезаписи остается на ваше усмотрение, но будьте осторожны. Если вы решите перезаписать ключи, то прежняя пара ключей будет удалена и больше не сможет быть использована для доступа к серверу. Чтобы избежать конфликта, лучше всего присваивать каждой новой паре уникальное имя, хотя каталог для хранения можно оставить неизменным.

Также вам будет предложено задать парольную фразу, что добавит дополнительный уровень безопасности, предотвращая доступ неавторизованных пользователей к вашему хосту. Если вы не желаете устанавливать парольную фразу, просто нажмите Enter.

Для убедительности в создании ключей можно выполнить следующую команду:

```
ls -l ~/.ssh/id_*.pub
```

В терминале будет отображен список сгенерированных ключей. Аналогично можно создать ключевую пару на macOS.

Если вы используете Windows, более простым вариантом будет загрузить PuTTY, который включает в себя утилиту PuTTYgen. С помощью этой утилиты можно создать пару ключей простым перемещением мыши. В PuTTYgen также можно указать папку для сохранения ключей и добавить парольную фразу для усиления безопасности.

Внесение ключа на сервер

Закрытый ключ остается на компьютере и никому не передается. В то время как открытую часть ключа необходимо перенести на сервер. Если у вас есть доступ к хосту с использованием пароля, то можно передать открытый ключ с помощью команды ssh-copy-id. Пример такой команды:

```
ssh-copy-id ubuntu@192.168.1.10
```

Вместо “ubuntu” введите ваше имя пользователя, а вместо “192.168.1.10” укажите IP-адрес сервера. После этого вам будет предложено ввести пароль, и после успешной аутентификации открытая часть ключа будет передана на хост.

Для установления подключения с использованием ключей выполните следующую команду:

```
ssh ubuntu@192.168.18.76
```

Замените “ubuntu” на ваше имя пользователя и ‘192.168.1.10’ на IP-адрес вашего сервера. Если вы не настроили парольную фразу, вы сможете войти в систему без дополнительной аутентификации. Система безопасности проведет проверку открытой и закрытой части ключа, и при совпадении установит соединение.

Настройка конфигурации

Для настройки Ubuntu Server используется файл /etc/ssh/sshd_config. Однако, перед внесением в него изменений, рекомендуется создать резервную копию. Это предотвратит потерю данных и сэкономит время, если случится ошибка.

Для создания резервной копии выполните следующую команду:

```
sudo cp /etc/ssh/sshd_config /etc/ssh/sshd_config.factory-defaults
```

В файле /etc/ssh/sshd_config.factory-defaults будут сохранены стандартные параметры. Вы будете вносить изменения и редактировать файл /etc/ssh/sshd_config.

Выключение возможности аутентификации с использованием пароля

Использование аутентификации по паролю на SSH Server Ubuntu может быть удобным, однако создание длинных и сложных паролей может привести к их небезопасному хранению. Для повышения уровня безопасности аутентификации рекомендуется использовать ключи шифрования вместо паролей. В таком случае, пароль может стать излишним. В целях увеличения безопасности, рекомендуется его отключить.

При этом, имейте в виду следующее:

1. Отключение аутентификации по паролю может повлечь за собой риск блокировки вашего доступа к серверу.
2. Существует риск блокировки доступа, если вы потеряете закрытый ключ или испортите файл ~/.authorized_keys.
3. При блокировке доступа, вы можете потерять возможность доступа к данным и приложениям на сервере.
4. Рекомендуется отключать аутентификацию по паролю только если вы хорошо знакомы с аутентификацией с использованием ключей и полностью осознаете потенциальные последствия блокировки доступа.

Для отключения аутентификации по паролю выполните следующие шаги:

1. Подключитесь к серверу с правами root.
2. Отредактируйте файл sshd_config, изменяя параметр PasswordAuthentication с ‘Yes’ на ‘No’.

3. После внесения изменений, перезапустите службу SSH с помощью следующей команды, выполненной от имени пользователя root:

```
sudo systemctl restart sshd
```

После выполнения этого действия, вы больше не сможете использовать пароли для аутентификации. Для подключения останется только возможность использовать SSH-ключи в Linux.

Отмена привилегий root

Для повышения безопасности вашей удаленной системы Ubuntu, рассмотрите вариант отключения доступа пользователя root через SSH. В Ubuntu, установленной на удаленном хосте, отредактируйте файл конфигурации и настройте соответствующий параметр, чтобы исключить доступ root-пользователя через SSH.

Откройте файл конфигурации SSH в Ubuntu для внесения изменений.

```
sudo vi /etc/ssh/sshd_config
```

Измените параметр PermitRootLogin на ‘No’.

Также имеется возможность разрешить аутентификацию пользователя root при использовании любого другого разрешенного механизма, за исключением пароля. Для этого установите параметру PermitRootLogin значение ‘prohibit-password’.

С такой настройкой вы сможете войти в систему как пользователь root с использованием закрытого ключа. Главное условие – перед перезапуском службы SSH-сервера необходимо скопировать открытый ключ в систему.

Для применения обновленной конфигурации permitrootlogin в SSHD перезапустите службу:

```
sudo systemctl restart sshd
```

Модификация порта по умолчанию

Исходно Open Server Ubuntu использует порт 22 по умолчанию. Для усиления безопасности, рекомендуется установить альтернативный порт. Часто рекомендуется выбирать значения из верхнего диапазона, например, от 50000 до 65000, и лучше всего использовать порты, в которых все цифры различны, например, 56789.

Необходимо открыть файл конфигурации:

```
sudo vi /etc/ssh/sshd_config
```

Раскомментируйте строку “Port 22”, заменив 22 на другой номер, например 56789. Сохраните внесенные изменения и закройте файл.

Чтобы применить эту конфигурацию, выполните перезапуск службы:

```
sudo systemctl restart sshd
```

После успешного перезапуска, проверьте, что теперь соединение осуществляется через новый порт:

```
ssh -p 56789 user@ip_server
```

Не забывайте, что после каждой настройки требуется выполнить перезапуск службы. В противном случае, SSH-подключение в Linux будет работать согласно предыдущей конфигурации.

Настройка создания туннелей

Туннелирование представляет собой метод передачи нешифрованного трафика или информации по зашифрованному каналу. Этот процесс может быть применен не только для передачи файлов, но также для обеспечения доступа к службам внутренней сети через брандмауэры и для создания виртуальных частных сетей (VPN).

Существуют три разновидности туннелирования:

- Локальное
- Удаленное
- Динамическое

Для настройки определенных видов туннелирования потребуется внести изменения в конфигурационный файл SSH.

Локальная переадресация

Этот процесс представляет собой перенаправление порта с клиентской машины на удаленную машину, а затем соединение переадресуется на другой порт целевой машины.

SSH-клиент проверяет наличие соединения на указанном порту. Когда он получает запрос на установление соединения, он пересылает его на указанный порт на удаленном хосте. После этого хост устанавливает соединение с другой целевой машиной через настроенный порт.

Обычно локальное перенаправление используется для удаленного доступа к службам из внутренней сети. Например, это может использоваться для доступа к базе данных или удаленного обмена файлами.

Для настройки локального перенаправления используется аргумент -L. Пример команды:

```
ssh networkadmin@server.example -L 8080:server1.example:3000
```

Теперь запустите веб-браузер на вашем локальном компьютере. Вместо того, чтобы обращаться к удаленному приложению по адресу server.example:3000, вы можете воспользоваться localhost:8080 для доступа к нему.

Удаленная переадресация

Возможность удаленного перенаправления позволяет устанавливать соединение с локальным компьютером из удаленного и работать с его ресурсами. По умолчанию SSH не включает функцию удаленного перенаправления портов, поэтому для использования этой функции вам необходимо внести дополнительные настройки в файл конфигурации SSH и выполнить дополнительную настройку сервера Ubuntu.

Для начала, откройте файл конфигурации:

```
sudo vi /etc/ssh/sshd_config
```

Установите значение ‘Yes’ для параметра GatewayPorts. Сохраните внесенные изменения и выполните перезапуск службы:

```
sudo systemctl restart sshd
```

Для настройки перенаправления, воспользуйтесь опцией -R. Пример команды:

```
ssh -R 8080:127.0.0.1:3000 -N -f user@remote.host
```

После выполнения данной команды, сервер будет прослушивать порт 8080 и перенаправлять всю сетевую активность на порт 3000, который открыт на вашем локальном компьютере. Удаленное перенаправление, как правило, используется для предоставления внешнему пользователю доступа к внутренним службам.

Динамическая переадресация

Локальное и удаленное перенаправление позволяют устанавливать туннели и обмениваться данными через один порт, в то время как динамическое перенаправление позволяет вам работать с несколькими портами одновременно.

Динамическое туннелирование создает на вашем локальном компьютере сокс-сервер, действующий как сервер SOCKS. Обычно он слушает порт 1080 по умолчанию. Когда удаленный хост подключается к этому порту, трафик пересылается сначала на удаленную

машину, а затем на целевой динамический порт.

Для настройки динамического туннелирования используется параметр SSH -D. Вот пример соответствующей команды:

```
ssh -D 9090 -N -f user@remote.host
```

После настройки туннелирования вы можете сконфигурировать ваше приложение для его использования, например, настроить прокси-сервер в вашем браузере. Важно помнить, что перенаправление трафика должно быть настроено отдельно для каждого приложения, которое вы хотите использовать через туннель.

Отключение SSH

Для деактивации сервера OpenSSH, приостановите работу службы SSH, выполнив следующую команду:

```
sudo systemctl disable --now ssh
```

Для повторного запуска службы, выполните следующую команду:

```
sudo systemctl enable --now ssh
```

Команда “Enable SSH” в Ubuntu не переустанавливает программное обеспечение, поэтому вам не придется выполнять повторную настройку. Она просто активирует ранее установленную и настроенную службу.

Заключение

В данной статье мы освоили основы применения SSH на Ubuntu. Теперь у вас есть знания о том, как установить необходимое программное обеспечение для настройки безопасного соединения, настроить его, настроить туннелирование и даже отключить службу, если она не требуется.

Умение подключаться к Ubuntu через SSH – это весьма распространенная навык, который может оказаться полезным вам, будь то в области разработки, администрирования или для личных нужд. Например, это может пригодиться для создания безопасного соединения между разными устройствами в локальной сети.

Revision #1
Created 2023-10-24 17:32:32 UTC by odiljonov
Updated 2023-11-25 17:58:32 UTC by odiljonov