## Продвинутая работа с Linux

### Цель работы

Продемонстрировать умение работать с командной строкой Linux с удаленной машиной по протоколу SSH.

### Задания для выполнения

1. Убедитесь, что на виртуальной машине установлен и запущен SSH-сервер.
2. В программе Putty (или любом другом SSH-клиенте), запущенной на хост-машине подключаемся по SSH на нужный IP к виртуальной машине и далее работаем с системой только через SSH
3. Отображаем содержимое /etc/apt/sources.list, используя команду cat
4. С её же помощью и символа “>” создаем файл в домашней директории пользователя.
5. Отображаем список процессов.
6. Запускаем мониторинг ресурсов системы.
7. Обновляем кэш менеджера пакетов apt-get.
8. Установите удаленно какой-либо пакет из репозитория на ваш выбор.
9. Выполните копирование файла по протоколу SSH с одного компьютера на другой.
10. Перезагрузить удаленный сервер через SSH.

### Методические указания

SSH — защищенный протокол для удаленного доступа к компьютерам. Через SSH можно выполнять операции в командной строке компьютера, который физически находится в другом месте.

Иными словами, SSH — это дистанционная командная строка. Визуально вы работаете на своем компьютере, но в реальности — на другом.

Что значит «протокол»?

Протокол — это набор соглашений, правил, по которым разные программы могут обмениваться информацией. SSH — это набор правил, который известен и вашему компьютеру, и физически отдаленному компьютеру.

Пример: вы вводите команду удаления файла, и эта команда передается на другой компьютер и выполняется там. Ответ (или сообщение об ошибке) возвращается и показывается на вашем компьютере.

Что значит «защищенный»?

Вся информация передается в зашифрованном виде. Подобно тому, как некоторые сайты работают по HTTPS, шифруя информацию. Например, ваш онлайн-банкинг обязательно должен работать по защищенному соединению. В таком случае даже если всю информацию перехватывает злоумышленник, он не сможет расшифровать её.

Зачем это нужно?

Не всегда есть возможность физически находиться у компьютера, с которым нужно работать. Например, если вы хотите создать свой сайт, то он будет размещен на компьютере хостинг-провайдера. Этот компьютер может находиться на другом конце света. Вам нужен способ запускать команды на этом компьютере не выходя из своего дома.

#### Как подключаться по SSH?

Для подключения к удаленной машине по SSH нужен клиент — специальная программа. В \*nix-подобных системах (Linux, macOS) клиент обычно установлен в системе по умолчанию, и достаточно открыть терминал. В Windows нужно скачать сторонний клиент, например, Putty.

Для подключения нужно указать адрес сервера и, опционально, имя пользователя и порт. Вот как выглядит команда при использовании консольного клиента (в терминале):

ssh username@remote\_host -p port

Например, для подключения к серверу 52.307.149.244 в аккаунт ivan нужно ввести:

ssh ivan@52.307.149.244

Если не указывать порт, то будет использован порт SSH по умолчанию — 22. Используемый порт задается при настройке SSH-сервера, программы, которая запущена на удаленном компьютере и ожидает подключения извне.

#### Fingerprint

При первом подключении появится сообщение:

The authenticity of host '52.307.149.244 (52.307.149.244)' can't be established.

ECDSA key fingerprint is fd:fd:d4:f9:77:fe:73:84:e1:55:00:ad:d6:6d:22:fe.

Are you sure you want to continue connecting (yes/no)? yes

Введите yes в первый раз.

Это нужно для повышения безопасности. При настройке SSH-сервера создается уникальная комбинация символов — fingerprint («отпечатки пальцев»). Ваш компьютер запоминает эту комбинацию и сверяет ее при каждом новом соединении. Если кто-то переустановит SSH-сервер, или всю операционную систему, или вообще заменит удаленный компьютер, сохранив его адрес, то при следующем соединении вы узнаете об этом, потому что изменится fingerprint.

Если fingerprint не меняется, то такое сообщение не будет появляться.

#### Подключение по паролю

Простейший вариант — подключение по паролю. После ввода команды ssh система запросит пароль:

ivan@52.307.149.244's password:

Пароль придется вводить каждый раз.

Подключение по ключу, без пароля

Для удобного подключения по SSH (и многим другим сервисам) без ввода пароля можно использовать ключи.

Нужно создать пару ключей: приватный (закрытый) ключ и публичный (открытый) ключ. Приватный ключ нужно хранить и никогда никому не показывать. Публичный ключ можно показывать всем и распространять свободно.

Эти ключи связаны друг с другом таким образом, что зашифровав информацию одним ключом, расшифровать ее можно только другим. Например, если ваш друг зашифрует письмо вашим публичным ключом, то прочитать его сможете только вы, потому что для этого нужен ваш приватный ключ. И наоборот: если вы зашифруете что-то своим приватным ключом, то расшифровать его можно только вашим публичным ключом. Так как публичный ключ доступен всем, любой может расшифровать это сообщение. Но он может быть уверен, что сообщение пришло именно от вас. В этом заключается идея цифровой подписи.

#### Генерация ключей

Создадим пару ключей:

ssh-keygen

Программа запустится и спросит, куда сохранять ключи:

Generating public/private rsa key pair.

Enter file in which to save the key (/home/demo/.ssh/id\_rsa):

Нажмите Enter для сохранения в стандартное место — директорию .ssh/id\_rsa в вашей домашней директории.

Программа запросит passphrase. Это вроде пароля для ключа. Можно просто нажать Enter и пропустить этот шаг. Или ввести passphrase — тогда его нужно будет вводить каждый раз, когда используется ключ.

Enter passphrase (empty for no passphrase):

Enter same passphrase again:

Ключи созданы:

Your identification has been saved in /home/demo/.ssh/id\_rsa.

Your public key has been saved in /home/demo/.ssh/id\_rsa.pub.

The key fingerprint is:

8c:e9:7c:fa:bf:c4:e5:9c:c9:b8:60:1f:fe:1c:d3:8a root@here

The key's randomart image is:

+--[ RSA 2048]----+

| |

| |

| |

| + |

| o S . |

| o . \* + |

| o + = O . |

| + = = + |

| ....Eo+ |

+-----------------+

Теперь у вас есть два файла:

~/.ssh/id\_rsa — приватный ключ. Никогда никому и никуда не передавайте его!

~/.ssh/id\_rsa.pub — публичный ключ. Спокойно распространяйте его.

В Windows можно использовать ssh-gen в подсистеме Ubuntu for Windows или в командной строке Git for Windows. Или создавать ключи графической утилитой вроде PuTTYgen.

#### Загрузка публичного ключа на сервер

Нужно добавить публичный ключ на сервер в файл ~/.ssh/authorized\_keys. Самый простой способ — запустить на локальной машине команду для копирования ключа:

ssh-copy-id -i /home/demo/.ssh/id\_rsa ivan@52.307.149.244

Другой способ — подключиться по паролю, открыть в редакторе файл ~/.ssh/authorized\_keys и добавить в конец текст из вашего файла ~/.ssh/id\_rsa.pub.

Теперь при подключении пароль запрашиваться не будет1.

После включения соединений по ключу рекомендуется отключить подключение по паролю.

#### ssh-agent

При работе с ключами возможны две неудобные ситуации:

1. Если при созданиия ключа вы указали passphrase (пароль для ключа), то вам придется вводить пароль при каждом подключении.
2. Если у вас есть несколько ключей для разных целей, то при соединении по ssh придется указывать нужный ключ вручную

ssh-agent решает эти проблемы. Этот агент аутентификации (authentication agent) работает на фоне в \*nix-системах. В зависимости от системы, вам, возможно, придется установить и настроить его автозапуск самостоятельно.

Если добавить ключ к агенту, то:

1. для него больше не будет спрашиваться passphrase
2. не нужно будет вводить ключ вручную — он будет автоматически использован при соответствующем подключении

ssh-add /home/demo/.ssh/id\_rsa добавит ключ id\_rsa в запущенный в системе агент. Если у него есть passphrase, то агент попросит ввести его.

Если запустить ssh-add без аргументов, то будут добавлены ключи ~/.ssh/id\_rsa, ~/.ssh/id\_dsa, ~/.ssh/id\_ecdsa, ~/.ssh/id\_ed25519 и ~/.ssh/identity.

Список добавленных в агент ключей можно посмотреть командой ssh-add -L:

→ ssh-add -L

ssh-rsa AAAAB3NzaC1yc2EAAAADAQABAAABAQC91r/+5WEQHcxVMrxpP9jKuONjlrnEHshfG3v/ab2NKDSljdskODOIsdhaaoDoiSADhAaoDISHasoiDiASisjadOHISDdKJDASHSidshIHDSIHDIAsdjasAs7XG/drBhi16zQ2e8VcLD7bVQS1Cpo0O1tP+93YQBvcIE02RltqVKYo7BlgCaJzpdowK8fHSzpfCYsEFjdjosOjfdsjdjkAJOKkKKHJHhaIiAiaihsiIoqkpqdmlnvnuuUSCaAS8aDhajiadiiAahhakKAKDHAKurmD08jnX9HfH/d15pLK/Glo1Su6iEOU3bW8k92QlY54pPFLKiNRPFuUryE5md7T /Users/demo/.ssh/some\_key.pem

ssh-rsa AAAAB3NzaC1yc2EAAAABIwAAAQEAsLC9WpSZ/9YpQ2z1FTSsORcP+ohzCdVjYaoc3C0fRnUbkp4SnvMHFTHNFFod0FhV0cQcOLvBsZAK/0tUPIXeDDFvYD70r5i0AsQbqA0k7gK3b3MP7tmnPxMHd607TI+1FMO54Yig0vnpZOgKmgCsxWq6tckwyLB91BlPiGxLBZiu5yPDIguEQCSnAwkF0vjqrNGsoHB4+fkj0USfjiifsjihf39hifSIHiJFHSijshfj39jfsjisfiisfiissr893IFsifijfsjSOIiAShadfhssU0q0JpjaDEWcMmYXmuz3xSnbhkueGLBXMU2zXDFDWCDSHq9/oRr29UAfVaHAMw== /Users/demo/.ssh/id\_rsa

ssh-agent привязан к сессии. Поэтому, например, если перезагрузить компьютер, то ключи нужно будет добавлять в агент заново.

#### Форвардинг (проброс) ключей

Если вы подключились к удаленному серверу X, и с него хотите подключиться к другому серверу Y, например, чтобы сделать git pull с GitHub’а, то придется держать копию ваших ключей на сервере X.

Утилита ssh с флагом -A позволяет «пробросить» ключи с подключаемой машины в удаленную:

ssh -A ivan@52.307.149.244

Ключи, добавленные к агенту аутентификации (ssh-agent) станут доступными на удаленном сервере. При этом файлы-ключи физически не будут находиться на сервере.

### Контрольные вопросы

1. Для чего нужен протокол SSH?
2. Почему этот протокол называют защищенным?
3. Какая информация нужна для того, чтобы подключиться к удаленной машине?
4. Из чего состоит пара ключей?
5. Каковы рекомендуемые действия по безопасному использованию SSH?
6. Как установить и настроить пакета OpenSSH в Linux и Windows?

### Дополнительные задания

1. Выполните настройку Вашего и виртуального компьютеров так, чтобы возможно было входить с одного на другой по открытому ключу, без явного запроса пароля, по протоколу SSH.
2. Создайте файл в домашней директории локальной машины. Скопируйте его на удаленную машину при помощи утилиты scp. Удалите исходный файл. Скопируйте файл обратно.
3. (\*) Проанализировать сравнительную скорость ssh-соединения, выполнив следующие замеры:
4. Выполнить передачу файла через FTP с помощью утилиты wget. Проанализировать проходящие пакеты с помощью утилиты tcpdump(-ХХ). Запомнить скорость передачи.
5. Выполнить передачу через ssh. Проанализировать проходящие пакеты с помощью утилиты tcpdump(-ХХ). Запомнить скорость передачи. Сравнить передаваемые пакеты и скорости передачи данных.
6. Включить сжатие ssh и повторить замер скорости. В каждом тесте анализировать результаты для файла состоящего из нулей и для файла состоящего из случайных последовательностей (dd if= /dev/urandom of=file ds=1M count=10), для текстового конфигурационного файла или бинарного файла.