SSH

Данная статья применима к:

- ОС ОН Орёл 2.12;
- ОС СН Смоленск 1.6;
- ОС СН Ленинград 8.1.

Установка пакета

Пакет SSH входит в дистрибутивы ОС ОН Орёл 2.12 и ОС СН Смоленск 1.6, но по умолчанию устанавливается только клиент.

Установку сервера можно выполнить при инсталляции системы, отметив соответствующий пункт в диалоге выбора программного обеспечения, либо после установки системы с помощью графического менеджера пакетов или из командной строки:

```
apt install ssh
```

После установки с помощью графического менеджера пакетов или из командной строки сервис запускается автоматически.

В ОС ОН Astra Linux Орёл начиная с версии 2.12.12 сервис SSH, установленный при инсталляции операционной системы, запускается после перезагрузки автоматически.

В более ранних версиях после установки при инсталляции ОС сервис ssh будет отключен, и его нужно будет включить и запустить отдельно командами:

```
systemctl enable ssh
systemctl start ssh
```

Настройка пакета

Проверить состояние сервиса:

```
systemctl status ssh
```

Конфигурация сервера хранится в файле /etc/ssh/sshd_config.

Чтобы изменения вступили в силу необходимо перезапустить сервис:

```
systemctl restart sshd
```

Простейшие меры безопасности

Из соображений повышения безопасности рекомендуется изменить некоторые значения по умолчанию:

- **Port** - номер порта, который слушает сервис (22) - на любое другое
- **MaxAuthTries** - количество попыток подключения (6) - уменьшить, например, до 3
- **LoginGraceTime** - время, дающееся для подключения (2m) - уменьшить, например до 30s

Дополнительно, можно ограничить IP-адреса, с которых возможно подключение, например:

- в файле /etc/hosts.deny запретить все подключения:

/etc/hosts.deny: list of hosts that are _not_ allowed to access the system.

See the manual pages hosts_access(5) and hosts_options(5).
#
Example: ALL: some.host.name, .some.domain
ALL EXCEPT in.fingerd: other.host.name, .other.domain
#
If you're going to protect the portmapper use the name "rpcbind" for the
daemon name. See rpcbind(8) and rpc.mountd(8) for further information.
#
The PARANOID wildcard matches any host whose name does not match its
address.
#
You may wish to enable this to ensure any programs that don't
validate looked up hostnames still leave understandable logs. In past
versions of Debian this has been the default.
ALL: PARANOID

ALL

- а в файле /etc/hosts.allow разрешить только необходимые:

/etc/hosts.allow: list of hosts that are allowed to access the system.
See the manual pages hosts_access(5) and hosts_options(5).
#
Example: ALL: LOCAL @some_netgroup
ALL: .foobar.edu EXCEPT terminalserver.foobar.edu
#
If you're going to protect the portmapper use the name "rpcbind" for the
daemon name. See rpcbind(8) and rpc.mountd(8) for further information.
#

SSHD: 192.168.1.1
SSHD: 192.168.0.0/16

Авторизация по ключам

См. раздел "В помощь администратору"

Настройка клиента

Конфигурация клиента хранится в файле /etc/ssh/ssh_config.

Клиент работоспособен сразу после установки ОС, и настроек не требует.

Настройка выбора алгоритмов защитного преобразования.

Проверить список поддерживаемых алгоритмов защитного преобразования (параметр cipher) и выработки имитовставки (параметр mac) можно командами:

```
ssh -Q cipher
ssh -Q mac
```

В поставляемый в составе дистрибутивов ОС ОН Орёл 2.12 и ОС СН Смоленск 1.6 пакет ssh встроены следующие алгоритмы защитного преобразования:

3des-cbc
blowfish-cbc
cast128-cbc
arcfour
arcfour128
arcfour256
aes128-cbc
aes192-cbc
aes256-cbc
rijndael-cbc@lysator.liu.se
aes128-ctr
aes192-ctr
aes256-ctr
grasshopper-cbc (ГОСТ Р 34.13–2015 "Кузнечик")
grasshopper-ctr (ГОСТ Р 34.13–2015 "Кузнечик")
aes128-gcm@openssh.com
aes256-gcm@openssh.com
chacha20-poly1305@openssh.com

В поставляемый в составе дистрибутивов ОС ОН Орёл 2.12 и ОС СН Смоленск 1.6 пакет ssh встроены следующие алгоритмы выработки имитовставки:

hmac-sha1
hmac-sha1-96
hmac-sha2-256
hmac-sha2-512
hmac-md5
hmac-md5-96
hmac-ripemd160
hmac-ripemd160@openssh.com
umac-64@openssh.com
umac-128@openssh.com
hmac-sha1-etm@openssh.com
hmac-sha1-96-etm@openssh.com
hmac-sha2-256-etm@openssh.com
hmac-sha2-512-etm@openssh.com
hmac-md5-etm@openssh.com
hmac-md5-96-etm@openssh.com
hmac-ripemd160-etm@openssh.com
umac-64-etm@openssh.com
umac-128-etm@openssh.com
hmac-gost2012-256-etm (ГОСТ Р 34.11-2012)

При этом в список алгоритмов защитного преобразования (параметр конфигурации Ciphers) и выработки имитовставок (параметр конфигурации MACs), допустимых к использованию, по умолчанию включены следующие алгоритмы (перечислены в порядке убывания приоритетов применения):

```
# Ciphers grasshopper-ctr,aes128-ctr,aes192-ctr,aes256-ctr,arcfour256,arcfour128,aes128-cbc,3des-cbc
# MACs hmac-gost2012-256-etm,hmac-md5,hmac-sha1,umac-64@openssh.com,hmac-ripemd160
```

Выше приведены строчки из конфигурационного файла клиента /etc/ssh/ssh_config, аналогичные строчки имеются в конфигурационном файле сервера /etc /ssh/sshd_config.
Если возникает необходимость изменить набор допустимых алгоритмов, или изменить их приоритеты, следует раскомментировать эту строчку, и указать нужные алгоритмы в нужном порядка приоритета.

Например, для приоритетного выбора более простых, а значит, более быстрых алгоритмов можно использовать в файле /etc/ssh/ssh_config следующую конфигурацию:

```
Ciphers aes128-ctr,aes192-ctr,aes256-ctr,arcfour256,arcfour128,aes128-cbc,3des-cbc,grasshopper-ctr
MACs hmac-md5,hmac-sha1,umac-64@openssh.com,hmac-ripemd160,hmac-gost2012-256-etm
```

Монтирование удалённой файловой системы по ssh

SSHFS (Secure Shell FileSystem) - программа, позволяющая монтировать удаленную файловую систему и взаимодействовать с удаленными файловыми ресурсами как с обычными файлами.

Для монтирования **SSHFS** использует **SSH File Transfer Protocol (SFTP)** — безопасный протокол передачи данных, обеспечивающий полный доступ к файловым ресурсам через протокол **Secure Shell**.

От сервера, предоставляющего удалённый ресурс для монтирования, требуется только работающий сервис SSH.

При стандартной установке ОС СН Орёл 2.12 и ОС ОН Смоленск 1.6 пакет sshfs устанавливается автоматически, при необходимости может быть установлен с помощью графического менеджера пакетов или из командной строки:

```
apt install sshfs
```

Для выполнения монтирования без запроса пароля должна быть настроена авторизация по ключам (см. выше).

Команда монтирования:

```
sshfs -o allow_other,IdentityFile=~local_user/.ssh/id_rsa
remote_user@server:/home/remote_user /mnt
```

Пояснения к команде:

- **-o** - ключ, указывающий, что далее следуют параметры подключения/монтирования;
- **allow_other** - разрешить доступ к примонтированному ресурсу непривилегированным пользователям (необязательный параметр);
- **IdentityFile=~local_user/.ssh/id_rsa** - файл с ключом доступа (необязательный параметр, если его нет - будет запрошен пароль);
- **remote_user@server:/home/remote_user** - указание на ресурс, который будем монтировать;
  (в данном случае на сервере server от имени пользователя remote_user монтируем домашний каталог этого пользователя /home/remote_user)
- **/mnt** - локальная точка монтирования

Автоматическое монтирование может быть задано в файле /etc/fstab:

```
sshfs#remote_user@server:/home/remote_user/ /mnt fuse defaults,allow_other,IdentityFile=~local_user/.ssh/id_rsa 0 0
```

Для выполнения автоматического монтирования без запроса пароля должна быть настроена авторизация по ключам (см. выше).

Демонтировать ресурс можно обычной командой umount с указанием точки монтирования:

```
umount /mnt
```

В случае ошибок монтирования для демонтирования может понадобиться отдельная команда:

```
fusermount -u /mnt
```

Использование алгоритмов защитного преобразования данных ГОСТ

Сервер и клиент ssh, входящие в состав дистрибутивов ОС ОН Орёл 2.12 и ОС СН Смоленск 1.6, имеют встроенную поддержку работы с алгоритмами защитного преобразования ГОСТ,
причем, если такие алгоритмы поддерживаются и клиентом и сервером, то они используются по умолчанию.
Некоторые подробности про эти алгоритмы можно прочитать в описании библиотеки libgost-astra.

Проверка используемого алгоритма защитного преобразования

Проверку используемого при подключении алгоритма защитного преобразования можно производить как на стороне клиента, так и на стороне сервера.
В обоих случаях для этого нужно включить вывод отладочной информации.

На стороне сервера:
в файле конфигурации сервера /etc/ssh/sshd_config раскомментировать строчки SyslogFacility и LogLevel, заменить уровень отладки INFO на DEBUG

```
# Logging
SyslogFacility AUTH
~~LogLevel INFO~~
LogLevel DEBUG
```

После внесения изменений перезапустить сервер:

```
systemctl restart ssh
```

Отладочная информация сервера ssh выводится в файл /var/log/auth.log, и отслеживать сообщения о методах защитного преобразования, используемых при подключении клиентов, можно командой:

```
tail -f | grep cipher /var/log/auth.log
```

На стороне клиента:
При выполнении подключения использовать опцию -v для вывода отладочной информации:

```
ssh -v ServerSSH
```

И в первом, и во втором случае используемый для подключения метод защитного преобразования будет отображен в виде строчек такого вида:

debug1: kex: server->client cipher: grasshopper-ctr MAC: hmac-gost2012-256-etm compression: none
debug1: kex: client->server cipher: grasshopper-ctr MAC: hmac-gost2012-256-etm compression: none