Эффективный SSH

Отдел R&D

Маркетинговая группа Текарт

26 мая 2009 г.

Что такое SSH

- Протокол, обеспечивающий шифрование сетевого трафика
- Две несовместимые версии: SSH-1 (1995) и SSH-2 (2000)
- Замена telnet, ftp, rlogin, rsh, rcp
- Существует практически для всех платформ
- OpenSSH – наиболее распространенная реализация

Архитектура клиент-сервер

Университет Текарт

Аутентификация

Атаки, связанные с подменой идентичности

MIDM: man in the middle attack

IP spoofing — атакующий отсылает свои пакеты, симулируя, что они пришли с другого компьютера

DNS spoofing — атакующий фальсифицирует записи DNS

...

Доверяй, но проверяй: нельзя доверять сети, необходимо проверять идентичность хостов

Университет Текарт

Проверка идентичности хоста

- При установке сервера генерируются ключи: `/etc/ssh/ssh_host_key`, `/etc/ssh/ssh_host_key.pub`
- Ключи используются при установке соединения
- Идентификатор сервера: хеш его ключа хоста, добавляется в `known_hosts`:

```
The authenticity of host 'ssh1.techart.ru (194.186.243.194)' can't be established.
DSA key fingerprint is 7f:a8:b2:84:2a:c2:34:af:d4:d3:4a:4b:2e:eb:55:50.
Are you sure you want to continue connecting (yes/no)?
```

- Смена хоста $\Rightarrow$ смена отпечатка $\Rightarrow$ предупреждение

Получение отпечатка на сервере после логина:

```
ssh-keygen -l -f /etc/ssh/ssh_host_rsa_key.pub
1024  7f:a8:b2:84:2a:c2:34:af:d4:d3:4a:4b:2e:eb:55:50
```

Хранение идентификаторов хостов

- `/etc/ssh/ssh_known_hosts` заполняется администратором
- `~/.ssh/known_hosts` заполняется автоматически

Формат: HOSTNAMES | BITS EXPONENT MODULUS | COMMENT

HOST PUBLIC KEY

HOSTNAMES

- список шаблонов имен хостов через запятую (* и ?);
- можно запрещать отдельные хосты с помощью !;
- ия хоста может храниться в виде хеш-строки для повышения безопасности.

Методы аутентификации

Password	Парольная аутентификация
Host Based	Аутентификация по адресу клиента с проверкой подлинности
Public Key	Аутентификация с использованием публичного ключа

Другие методы: Kerberos, LDAP и т.д.

Парольная аутентификация

Меры предосторожности:

- пароль пересылается в зашифрованном виде;
- выполняется проверка идентичности хоста;

Недостатки:

- Пароль должен быть достаточно сложным, для противодействия подбору;
- Его необходимо вводить при каждом подключении;
- После попадания на сервер теоретически может быть перехвачен, если сервер скомпрометирован.

Автоматический ввод пароля, например, sshpass – плохо.

Аутентификация по адресу клиента

- Клиентские имя хоста и учетная запись должны быть в списке доверенных хостов:

  `/etc/hosts.equiv`
  `/etc/ssh/shosts.equiv`
  `~/.rhosts`
  `~/.shosts`

- Сервер обязан проверить идентичность клиента по его ключу:

  `/etc/ssh/ssh_known_hosts`
  `~/.ssh/known_hosts`

Аутентификация с использованием публичного ключа

1. Клиент посылает серверу запрос на соединение;
2. Сервер идентифицирует хост клиента;
3. Сервер кодирует случайную строку с помощью public key;
4. Клиент декодирует строку с помощью private key;
5. Клиент комбинирует ее с идентификатором сессии, хеширует и посылает на сервер;
6. Сервер проводит аналогичные вычисления;
7. Если результат совпал – аутентификация прошла успешно.

Private не покидает клиента, public не нуждается в защите.

Генерация ключей

```
$ ssh-keygen -t {dsa,rsa}
```

Результат:

```
~/.ssh/id_dsa  ~/.ssh/id_dsa.pub
~/.ssh/id_rsa  ~/.ssh/id_rsa.pub
```

Дополнительная мера безопасности: парольная фраза.

- снижает риск в случае похищения личного ключа;
- ввод можно автоматизировать (ssh-agent);
- использовать пустую фразу не рекомендуется.

Установка публичного ключа на сервер

- Заходим с паролем, добавляем ключ в файл .ssh/authorized_keys
- То же самое, но автоматически: ssh-copy-id.

```
ssh-copy-id [-i [identity_file]] [user@]machine
```

Настройка параметров клиентских сессий

authorized_keys – не просто список ключей, но и возможность настройки параметров подключения.

Формат: `OPTIONS` `KEY TYPE` `BASE64 PUBLIC KEY` `COMMENT`

`OPTIONS`

command	принудительно запускать указанную команду;
from	явное указание списка хостов, шаблоны;
environment	установка значений переменных среды;
permitopen	перенаправление только для портов из списка;
...	

Можно один хост – несколько ключей, разные настройки.

Принудительное выполнение команды в authorized_keys

- Указанная пользователем команда и shell не выполняются.
- Можно использовать различные ключи для различных команд:

```
command="/usr/bin/uptime",no-port-forwarding...
command="/etc/init.d/apache restart",no-port...
```

- В собственных скриптах, оригинальная вызванная команда доступна как переменная `$SSH_ORIGINAL_COMMAND`.

Практическое использование

SSH Agent

Если private key не защищен с помощью парольной фразы, его могут украсть и использовать с другой машины.

Неудобства:

- при каждом подключении надо заново вводить фразу;
- нельзя автоматически запускать процессы.

Выход: `ssh-agent`

- Запускается как демон;
- Ключи добавляются с помощью `ssh-add`;
- Незашифрованные ключи – только в памяти;
- Дочерние процессы тоже используют агент.

Использование SSH Agent

Варианты запуска:

- для текущего shell: `eval ‘ssh-agent‘`;
- отдельный shell: `ssh-agent bash`;
- любая программа, например оконный менеджер или десктопное окружение;
- скорее всего, уже запущен в вашей десктопной системе.

`ssh-add` – добавление, удаление и просмотр списка ключей

```
$ ssh-add
Enter passphrase for /home/max/.ssh/id_rsa:
Identity added: /home/max/.ssh/id_rsa
```

Стандартные возможности

- Удаленный shell:

```
$ ssh max@ssh1.techart.ru
```

- Удаленное выполнение команд:

```
$ ssh max@ssh1.techart.ru 'ls -l'
```

- Безопасное копирование:

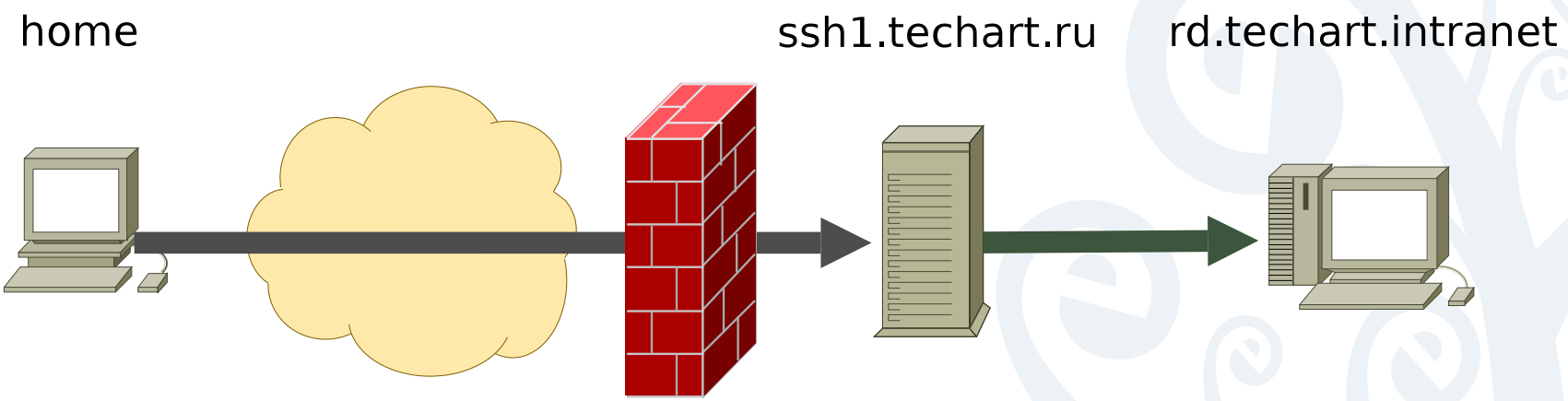
```
$ ssh scp max@ssh1.techart.ru:backup.tgz .
```

- sftp – безопасный аналог ftp:

```
$ sftp max@ssh1.techart.ru
sftp> ls
```

ProxyCommand - работа через прокси

Задача: получить доступ к машине за файрволом.



Решение: `~/.ssh/config`

```
Host rd.techart.intranet
ProxyCommand=ssh ssh1.techart.ru /usr/bin/nc %h 22
```

Университет Текарт

Туннели

Port Forwarding

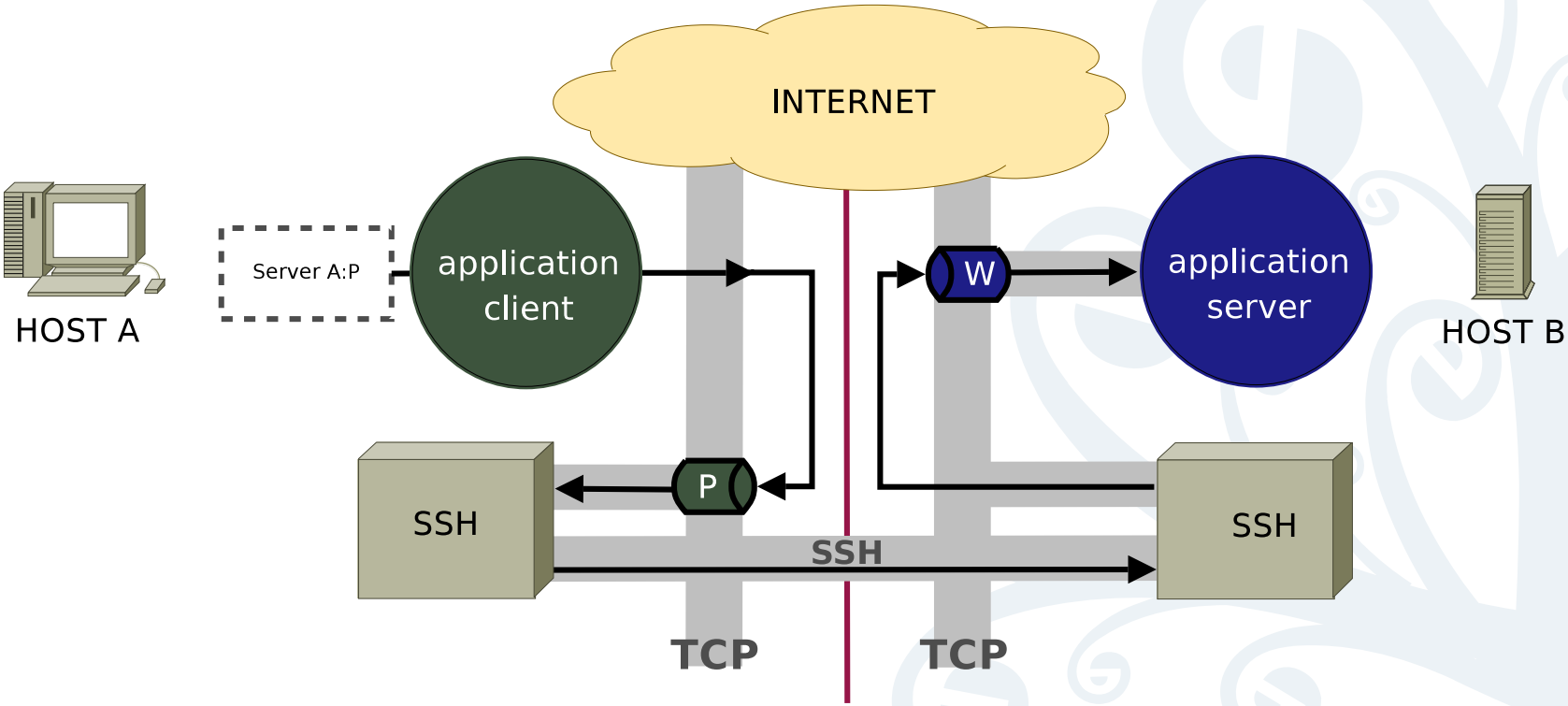
SSH позволяет строить защищенные туннели для различных сетевых протоколов.

Зачем:

- Доступ по протоколам SMTP, IMAP, LDAP через файрвол, запрещающий непосредственный доступ.
- Использование SMTP-сервера провайдера даже за пределами его сети;
- Шифрование данных для предотвращения их перехвата;

Работает только для TCP, с UDP-сервисами так работать нельзя.

Port Forwarding: схема работы




```
host-a$ ssh -L P:locahost:W host-b
```

Университет Текарт

Local vs Remote Forwarding

Local клиентское приложение на стороне SSH-клиента, сервер – на стороне SSH-сервера.

```
ssh -L <localport>:<dest-host>:<dest-port>
```

Remote клиентское приложение на стороне SSH-сервера, сервер – на стороне SSH-клиента.

```
ssh -R  <remoteport>:<dest-host>:<dest-port>
```

Local Forward: пример

Задача: забирать из дома почту с корпоративного pop3-сервера, закрытого файрволом.

Командная строка:

```
$ ssh -L 10010:pop3.techart.intranet:110 ssh1.techart.ru
```

Или `~/.ssh/config`:

```
HostName pop3.techart.intranet
  HostName ssh1.techart.ru
  LocalForward 10010:pop3.techart.intranet:110
```

Адрес сервера в почтовом клиенте: `localhost:10010`

Remote Forward

Задача: получить с работы доступ к домашнему компьютеру, у которого даже нет статического IP-адреса.

На домашней машине запускаем и оставляем:

```
$ ssh -R 10022:localhost:22 ssh1.techart.ru
```

Приходим на работу и на своей рабочей машине:

```
$ ssh -p 10022 ssh1.techart.ru
```

Мы получили удаленный shell на домашней машине.

Что читать

- OpenSSH
- SSH, The Secure Shell: The Definitive Guide, 2nd ed.
- Pro OpenSSH
- Getting Started With SSH
- Practical SSH Encryption, Tunneling and Automation
- SSH Links

Спасибо за внимание