SSH – Възможности и приложения

представени от

Васил Колев

vasil@ludost.net http://vasil.ludost.net/

Георги Чорбаджийски

georgi@unixsol.org http://georgi.unixsol.org/

План за презентацията

- Какво е SSH?
- Как протича една SSHv2 сесия?
- Методи за идентификация
- SSH конзола
- Локално пренасочване на портове (*Local port forwarding*)
- Отдалечено пренасочване на портове (*Remote port forwarding*)
- Динамично пренасочване на портове (*Dynamic port forwarding*)
- Прехвърляне на X11 протокола (*X11 forwarding*)
- Прехвърляне на идентификация (*Agent forwarding*)
- Допълнителни програми

Основни възможности на SSH

- SSH протокола
  - Сигурна идентификация и на двете страни на връзката (RSA, DSA)
  - Възможности за криптиране на връзката (Blowfish, 3DES, DES)
  - Множество методи за идентификация (пароли, публични ключове, hostbased)
  - Отдалечен интерактивен достъп (`ssh host`)
  - Отдалечено изпълнение на команди (`ssh host 'command'`)
  - Трансфер на файлове (`scp, sftp`)
  - Прехвърляне на TCP връзки (`ssh -L, ssh -R, ssh -D`)
  - Прехвърляне на X11 протокола (`ssh -X`)
  - Прехвърляне на идентификация (`ssh-agent, ssh-add`)
  - Компресия на данните (`ssh -C`)
- Сигурен заместител на rlogin, rcp, rexec, telnet и ftp
- Дефакто стандарт за отдалечена администрация

История и реализации

- Версии на протокола
  - Първата версия на протокола (SSHv1) е от 1995 и е замислена като заместител на r-suite, използването и не е препоръчително.
  - Втората версия (SSHv2) е подготвена като IETF стандарт през 1997 и целта и е да поправи пропуските във версия 1 и да направи протокола модулен. Не е съвместима с версия 1.
- Основни SSH имплементации
  - SSH.com – комерсиална имплементация на SSHv1 и v2
  - lsh – свободна имплементация на SSHv2
  - OpenSSH – свободна имплементация на SSHv1 и v2, използва се много широко във всички UNIX подобни операционни системи

Как протича една SSHv2 сесия

- При първоначално свързване се договаря версията на протокола.
- Сървърът изпраща своят ключ, а клиента проверява дали му е известен и дали съвпада.
- Договарят се алгоритми за криптиране и хеширане.
- Договаря се сесиен ключ, използвайки Diffie-Hellman key exchange алгоритъма.
- От тук нататък сесията се криптира.
- Договарят се механизми за идентификация и клиентът се идентифицира.
- Договаря се потребителската сесия. Може да бъде:
  - интерактивна
  - пренасочване на портове
  - изпълнение на команда
  - прехвърляне на файлове

Основни методи за идентификация

- Чрез използване на пароли
  - клиентът въвежда парола, която се проверява от сървъра
- Идентификация чрез публичен ключ
  - На база публичният ключ на клиента, описан от него в `~/.ssh/authorized_keys` файла
- Host based идентификация
  - Подобна на r-suite

Идентификация с пароли

- Предимства
  - съвместимост
  - лесна настройка
- Недостатъци
  - всеки път трябва да се въвежда паролата, не е възможно автоматизиране
  - компрометирането на паролата е фатално
- Предварителна подготовка
- Демонстрация
  - използване на идентификация с пароли

Идентификация с публични ключове

- Предимства
  - по-сигурен метод за идентификация от паролите
  - възможно е автоматизиране на идентификацията (`ssh-agent`)
  - възможно е частният ключ да се взима от външен носител (SSH_ASKPASS)
- Недостатъци
  - различен метод от идентификацията с пароли
  - изисква предварителна подготовка от страна на потребителя
- Предварителна подготовка
  - Клиентът генерира двойка ключове – публичен и частен и защитава частният с парола (`ssh-keygen -t dsa`)
  - Публичният ключ (`id_dsa.pub`) се добавя на сървъра в home директорията на потребителя във файла `~/.ssh/authorized_keys`
- Демонстрация
  - генериране на ключове
  - използване на ключове
  - използване на `ssh-agent`

Host based идентификация

- Предимства
  - много подобна на идентификацията на r-suite
- Недостатъци
  - твърде подобен на r-suite
  - изисква предварителна подготовка от страна на администратора
- Използването и не се препоръчва

SSH конзола

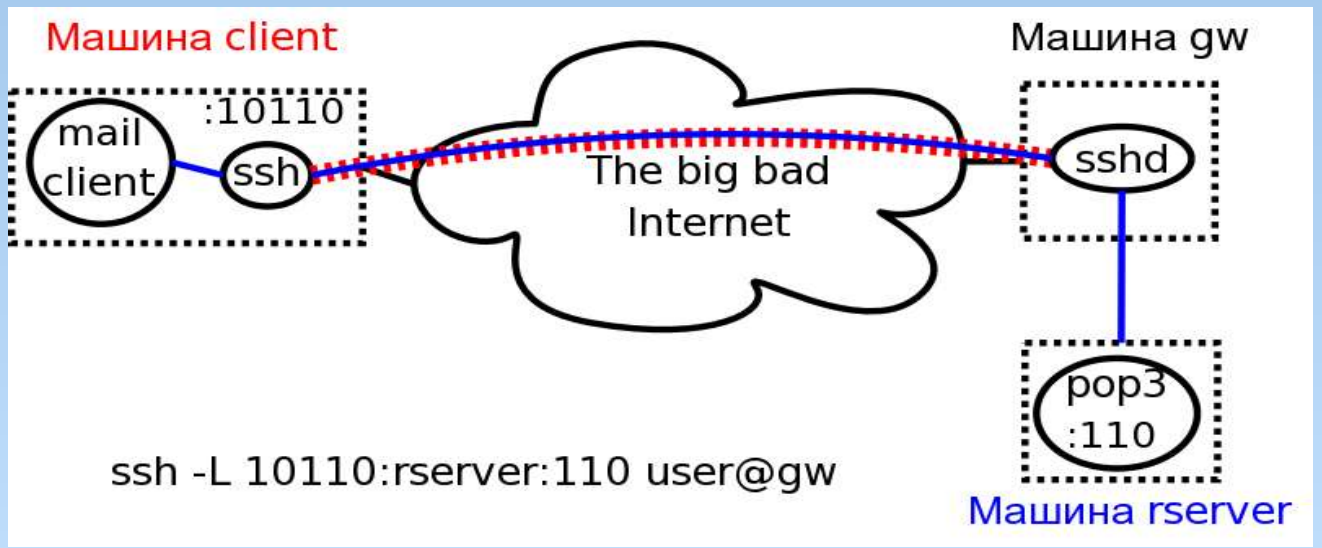
- Може да бъде извикана по всяко време, когато има интерактивна сесия с комбинацията от клавиши: ~?
- Дава следните възможности:
  - показване на списъка с пренасочени портове;
  - команден ред, от който могат да се настройва пренасочване на портове;
  - изискване за нов сесиен ключ;
  - поставяне на SSH на заден план;
  - моментално затваряне на връзката;

Локално пренасочване на портове

- Дава възможност на потребителя да ползва отдалечена услуга по сигурен начин.
- Как работи? (`ssh -L`)
  - ssh слуша на порт на клиентската машина

```
client> ssh -L 10110:rserver:110 user@gw
```

  - При свързване към порт `10110` на `client`, ssh прехвърля връзката до `rserver:110` през криптирания канал между `client` и gw.
- Демонстрация

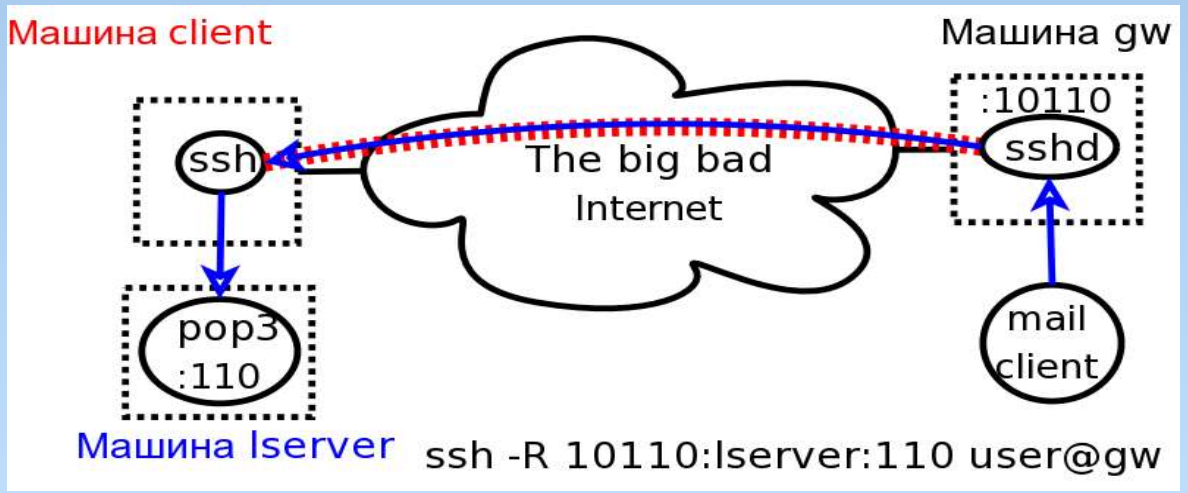


Отдалечено пренасочване на портове

- Дава възможност на отдалечени потребители да се свързват към ресурс в локалната мрежа, използвайки криптиран канал
- Как работи? (`ssh -R`)
  - sshd слуша на порт на отдалечената машина

```
client> ssh -R 10110:lserver:110 user@gw
```

  - При свързване към порт `10110` на `gw`, ssh прехвърля връзката до `lserver:110` през криптирания канал от `gw` до `client`.
- Демонстрация



Динамично пренасочване на портове

- Дава възможност на всяка програма, която поддържа SOCKS4/5 да излиза сигурно през друга машина
- Как работи? (`ssh -D`)
  - ssh слуша на порт на клиентската машина

```
client> ssh -D 1080 user@gw
```

  - На порт 1080 има слушащ SOCKS4/5 сървър. Заявките на всяка програма, настроена да ползва `client:1080` за SOCKS сървър, ще се тунелират през криптирания канал от `client` до `gw` и ще излизат през `gw`.
- Демонстрация

Прехвърляне на X11 протокола

- Дава възможност сигурно да се използват графични приложения от отдалечена машина.
- Как работи? (`ssh -X`)
  - влиза се на отдалечената машина

    ```
    client> ssh -X user@rserver
    ```

  - SSH настройва $DISPLAY променливата на `rserver`. По този начин всяко стартирано X приложение ще се вижда на локалният дисплей на клиента.
  - За да работи е необходимо на `rserver` да е инсталирана програмата `xauth`
- Демонстрация

Прехвърляне на идентификация

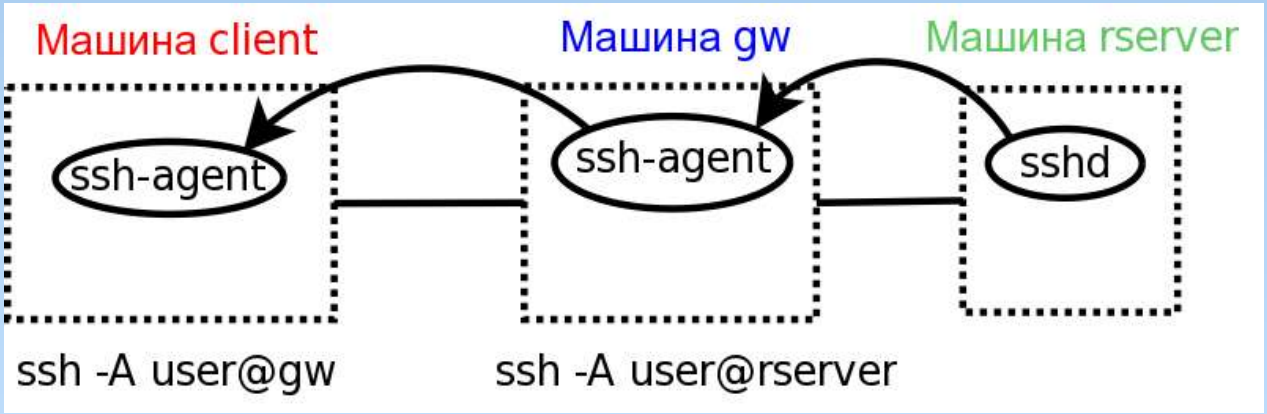
- Дава възможност да се използват ключовете въведени в `ssh-agent`, когато се прехвърляте към трета машина
- Как работи? (`ssh -A`)
  - влиза се на отдалечената машина

```
client> ssh -A user@gw
```

  - SSH активира ssh-agent на gw. При опит за логване на `rserver` от gw, чрез `ssh-agent` ще бъде поискана аутентикация от client. Не се прехвърлят ключове.

```
gw> ssh -A user@rserver
```

- Демонстрация



Допълнителни възможности

- Компресия на данните (`ssh -C`)
- Трансфер на файлове
  - `scp`
  - `sftp`
- Помощни програми за работа с ключове
  - `ssh-keyscan`
  - `ssh-copy-id`
- Други програми
  - `scponly` (http://www.sublimation.org/scponly/)

Благодарим за вниманието!

Ако имате въпроси, заповядайте!

Тази презентация е направена само със свободен софтуер. Използван е OpenOffice Impress, от пакета OpenOffice, който можете да си свалите от http://www.openoffice.org/