SSH

Emil Tomczyk

SKNI

25-04-2022
v1.1

Spis treści

Co to jest SSH

SSH to protokół używany do zdalnego łączenia się z różnymi systemami. Umożliwia on nawiązywanie połączeń, tworzenie tuneli, przesyłanie plików a wszystko to przez bezpieczne, szyfrowane łącze.

Czego używaliśmy przed SSH

Czego używaliśmy przed SSH

- rlogin

Czego używaliśmy przed SSH

- rlogin
- telefon

Czego używaliśmy przed SSH

- rlogin
- telefon
- wycieczka do serwerowni

Skąd się wzięło SSH

SSH narodziło się na uniwersytecie w Helsinkach w 1995. Jego powstanie było związane z niedawnym atakiem na sieć uniwersytecką, ktoś podsłuchiwał połączenia, przez co uzyskał dostęp do haseł systemowych które były używane w programach `rlogin` czy `ftp`.

Skąd się wzięło SSH

SSH narodziło się na uniwersytecie w Helsinkach w 1995. Jego powstanie było związane z niedawnym atakiem na sieć uniwersytecką, ktoś podsłuchiwał połączenia, przez co uzyskał dostęp do haseł systemowych które były używane w programach `rlogin` czy `ftp`.
Kolejne wersje programu były coraz bardziej zamykane, w roku 2000 liczba użytkowników osiągnęła około 2 miliony.

Skąd się wzięło SSH

SSH narodziło się na uniwersytecie w Helsinkach w 1995. Jego powstanie było związane z niedawnym atakiem na sieć uniwersytecką, ktoś podsłuchiwał połączenia, przez co uzyskał dostęp do haseł systemowych które były używane w programach `rlogin` czy `ftp`.
Kolejne wersje programu były coraz bardziej zamykane, w roku 2000 liczba użytkowników osiągnęła około 2 miliony.
Aby stworzyć wolną implementację `SSH` powstał projekt `OSSH`. Z tego pakietu wyewoluował projekt `OpenSSH` działający w ramach `OpenBSD`. 1 grudnia 1999 roku pakiet `OpenSSH` został udostępniony w wersji 2.6 systemy `OpenBSD` oraz sportowany na inne systemy, takie jak `GNU/Linux` czy `FreeBSD`

Dlaczego używamy SSH

Dlaczego używamy SSH

- bezpieczne

Dlaczego używamy SSH

- bezpieczne
- wygodne

Dlaczego używamy SSH

- bezpieczne
- wygodne
- skuteczne

Podstawowe użycie SSH

Podstawowy sposób użycia SSH to polecenie: `ssh <host>`

Podstawowe użycie SSH

Podstawowy sposób użycia SSH to polecenie: `ssh <host>`
Komenda ta, przy typowym użyciu (przejdziemy do tego potem), połączy się z określonym hostem. Polecenie zaloguje się z nazwą użytkownika bieżącego użytkownika.

Podstawowe użycie SSH

Podstawowy sposób użycia SSH to polecenie: `ssh <host>`
Komenda ta, przy typowym użyciu (przejdziemy do tego potem), połączy się z określonym hostem. Polecenie zaloguje się z nazwą użytkownika bieżącego użytkownika.
Bez dodatkowej konfiguracji polecenie poprosi o hasło do konta z którym chcemy się połączyć.

Flagi klienta SSH

Na ten moment najważniejsze są dla nas następujące flagi:

Flagi klienta SSH

Na ten moment najważniejsze są dla nas następujące flagi:

- `-l`
  Flaga ta pozwala określać nazwę użytkownika. Domyślnie zostanie użyta nazwa bieżącego użytkownika.

Flagi klienta SSH

Na ten moment najważniejsze są dla nas następujące flagi:

- `-l`
  Flaga ta pozwala określać nazwę użytkownika. Domyślnie zostanie użyta nazwa bieżącego użytkownika.
- `-p`
  Flaga ta określa port do którego chcemy się połączyć. Domyślnie użyty będzie port 22.

Flagi klienta SSH

Na ten moment najważniejsze są dla nas następujące flagi:

- `-l`
  Flaga ta pozwala określać nazwę użytkownika. Domyślnie zostanie użyta nazwa bieżącego użytkownika.
- `-p`
  Flaga ta określa port do którego chcemy się połączyć. Domyślnie użyty będzie port 22.
- `-i`
  Flaga ta określa plik z kluczem **prywatnym** który zostanie użyty do połączenia. Domyślnie jest to plik `~/.ssh/id_rsa`.

Klucze SSH

Klucze SSH umożliwiają łączenie się ze zdalnym hostem bez użycia hasła.

Klucze SSH

Klucze SSH umożliwiają łączenie się ze zdalnym hostem bez użycia hasła.
Serwer może zostać tak skonfigurowany by odrzucał połączenia bez poprawnego klucza.

Utworzenie klucza SSH

Najpierw, jeśli tego nie zrobiliśmy wcześniej, musimy wygenerować swój klucz. Możemy to zrobić poleceniem `ssh-keygen`

Utworzenie klucza SSH

Najpierw, jeśli tego nie zrobiliśmy wcześniej, musimy wygenerować swój klucz. Możemy to zrobić poleceniem `ssh-keygen`
Program zapyta nas o ścieżkę (możemy zostawić domyślną) i hasło do klucza (możemy zostawić puste).

Utworzenie klucza SSH

Najpierw, jeśli tego nie zrobiliśmy wcześniej, musimy wygenerować swój klucz. Możemy to zrobić poleceniem `ssh-keygen`
Program zapyta nas o ścieżkę (możemy zostawić domyślną) i hasło do klucza (możemy zostawić puste).
Zostaną wygenerowane dwa pliki, domyślnie będą się one znajdować w katalogu `.ssh` oraz nosić nazwy `id_rsa` oraz `id_rsa.pub`
Plik z roszerzeniem `pub` jest naszym kluczem publicznym który wysyłamy na inne maszyny, plik bez roszerzenia jest kluczem prywatnym który powinniśmy trzymać w bezpiecznym miejscu.

Użycie klucza SSH

Aby użyć naszego klucza musimy wrzucić go na zdalną maszynę. Na początek użyjemy dosyć topornego sposobu. Później zobaczymy inne możliwości.

Użycie klucza SSH

Aby użyć naszego klucza musimy wrzucić go na zdalną maszynę. Na początek użyjemy dosyć topornego sposobu. Później zobaczymy inne możliwości.
Najpierw wyświetlimy plik z kluczem publicznym (`cat .ssh/id_rsa.pub`), po czym go skopiujemy zawartość do pliku `.ssh/authorized_keys` na zdalnym komputerze. Jeśli katalog `.ssh` na drugiej maszynie nie istnieje to możemy go utworzyć następującymi poleceniami:

Użycie klucza SSH

Aby użyć naszego klucza musimy wrzucić go na zdalną maszynę. Na początek użyjemy dosyć topornego sposobu. Później zobaczymy inne możliwości.
Najpierw wyświetlimy plik z kluczem publicznym (`cat .ssh/id_rsa.pub`), po czym go skopiujemy zawartość do pliku `.ssh/authorized_keys` na zdalnym komputerze. Jeśli katalog `.ssh` na drugiej maszynie nie istnieje to możemy go utworzyć następującymi poleceniami:

```
mkdir ~/.ssh
```

Użycie klucza SSH

Aby użyć naszego klucza musimy wrzucić go na zdalną maszynę. Na początek użyjemy dosyć topornego sposobu. Później zobaczymy inne możliwości.
Najpierw wyświetlimy plik z kluczem publicznym (`cat .ssh/id_rsa.pub`), po czym go skopiujemy zawartość do pliku `.ssh/authorized_keys` na zdalnym komputerze. Jeśli katalog `.ssh` na drugiej maszynie nie istnieje to możemy go utworzyć następującymi poleceniami:

```
mkdir ~/.ssh
chmod 0700 ~/.ssh
```

Użycie klucza SSH

Po wgraniu klucza SSH, powinniśmy móc zalogować się tak jak wcześniej. Różnica będzie polegać na tym, że zdalny system nie poprosi nas o hasło.

Kopiowanie przy użyciu scp

scp jest programem bardzo podobnym w użyciu do cp. Umożliwia on przesył pliku za pomocą SSH na inną maszynę.
Przykładowa składnia jest następująca:

Kopiowanie przy użyciu scp

`scp` jest programem bardzo podobnym w użyciu do cp. Umożliwia on przesył pliku za pomocą SSH na inną maszynę.
Przykładowa składnia jest następująca:

```
scp plik_lokalny user@zdalna.maszyna:plik_zdalny
```

Kopiowanie przy użyciu scp

`scp` jest programem bardzo podobnym w użyciu do `cp`. Umożliwia on przesył pliku za pomocą SSH na inną maszynę.
Przykładowa składnia jest następująca:
`scp plik_lokalny user@zdalna.maszyna:plik_zdalny`
Polecenie to prześle plik `plik_lokalny` na komputer `zdalna.maszyna`. Zostanie on tam utworzony pod nazwą `plik_zdalny` w katalogu domowym użytkownika `user`.

Fakty scp

Użycie `scp` może być początkowo uciążliwe przez kilka różnic w użyciu w stosunku do `ssh` oraz `cp`, a także inne, specyficzne cechy. Najważniejsze to:

Fakty scp

Użycie `scp` może być początkowo uciążliwe przez kilka różnic w użyciu w stosunku do `ssh` oraz `cp`, a także inne, specyficzne cechy. Najważniejsze to:

- Flaga ustawiająca port to -P

Fakty scp

Użycie `scp` może być początkowo uciążliwe przez kilka różnic w użyciu w stosunku do `ssh` oraz `cp`, a także inne, specyficzne cechy. Najważniejsze to:

- Flaga ustawiająca port to -P
- Flagi muszą być podane przed plikami

Fakty scp

Użycie `scp` może być początkowo uciążliwe przez kilka różnic w użyciu w stosunku do `ssh` oraz `cp`, a także inne, specyficzne cechy. Najważniejsze to:

- Flaga ustawiająca port to -P
- Flagi muszą być podane przed plikami
- Jeśli kopiujemy na zdalną maszynę i nie podamy ścieżki, to plik zostanie zapisany w katalogu domowym

Fakty scp

Użycie `scp` może być początkowo uciążliwe przez kilka różnic w użyciu w stosunku do `ssh` oraz `cp`, a także inne, specyficzne cechy. Najważniejsze to:

- Flaga ustawiająca port to -P
- Flagi muszą być podane przed plikami
- Jeśli kopiujemy na zdalną maszynę i nie podamy ścieżki, to plik zostanie zapisany w katalogu domowym
- Musimy pamiętać o podaniu dwukropka, nawet jeśli nie chcemy określać ścieżki

Przykłady użycia scp

Przykłady użycia scp

- `scp -P 2273 plik.tar.gz alice@host.local:`

Przykłady użycia scp

- `scp -P 2273 plik.tar.gz alice@host.local:`
- `scp alice@serwer:plik123 .`

Przykłady użycia scp

- `scp -P 2273 plik.tar.gz alice@host.local:`
- `scp alice@serwer:plik123 .`
- `scp -i .ssh/klucz plik root@remote:/usr/share/`

Przykłady użycia scp

- `scp -P 2273 plik.tar.gz alice@host.local:`
- `scp alice@serwer:plik123 .`
- `scp -i .ssh/klucz plik root@remote:/usr/share/`
- `scp plik plik2`

SSH config

Aby nie męczyć się za każdym razem z podawaniem portu, nazwy użytkownika, klucza i innych parametrów możemy je określić w pliku `.ssh/config`.

SSH config

Przykładowa struktura tego pliku jest następująca:

SSH config

Przykładowa struktura tego pliku jest następująca:

```
Host <nazwa hosta>
    HostName <pelna nazwa hosta>
    Port <port>
    User <uzytkownik>
    IdentityFile <klucz prywatny SSH>
```

sshfs

`sshfs` jest to polecenie wchodzące w skład pakietu `fuse`. Pozwala ono podmontować jako użytkownik katalog będący na innym systemie. Z perspektywy użytkownika jest to zwykły katalog jak każdy inny w systemie.

Użycie sshfs

`sshfs` można wywołać następująco:

```
sshfs user@host:<ścieżka> <ścieżka lokalna> [opcje]
```

Aby odmontować taki katalog wydajemy polecenie:

```
umount <katalog>
```

SFTP

`sshfs` którego właśnie użyliśmy używa pod spodem `SFTP` - programu wchodzącego w skład pakietu `OpenSSH` który służy do bezpiecznego transferu plików. Program ten działa podobnie do `FTP`.

SFTP

`sshfs` którego właśnie użyliśmy używa pod spodem `SFTP` - programu wchodzącego w skład pakietu `OpenSSH` który służy do bezpiecznego transferu plików. Program ten działa podobnie do `FTP`.
Możemy użyć tego programu także w graficznych menadżerach plików, takich jak Thunar, Nautilus, PCManFM czy Dolphin.

SFTP

Aby to zrobić wystarczy wpisać w pasek adresu następującą treść:

SFTP

Aby to zrobić wystarczy wpisać w pasek adresu następującą treść:
`sftp://[nazwa_użytkownika@]hostname[:port][sciezka]`

SFTP

Aby to zrobić wystarczy wpisać w pasek adresu następującą treść:
`sftp://[nazwa_użytkownika@]hostname[:port][sciezka]`
Przykładowo:

- `sftp://yamo.skni.umsc.pl`

SFTP

Aby to zrobić wystarczy wpisać w pasek adresu następującą treść:
`sftp://[nazwa_użytkownika@]hostname[:port][sciezka]`
Przykładowo:

- `sftp://yamo.skni.umsc.pl`
- `sftp://user@yamo.skni.umsc.pl`

SFTP

Aby to zrobić wystarczy wpisać w pasek adresu następującą treść:
`sftp://[nazwa_użytkownika@]hostname[:port][sciezka]`
Przykładowo:

- `sftp://yamo.skni.umsc.pl`
- `sftp://user@yamo.skni.umsc.pl`
- `sftp://user@yamo.skni.umsc.pl:2795`

SFTP

Aby to zrobić wystarczy wpisać w pasek adresu następującą treść:
`sftp://[nazwa_użytkownika@]hostname[:port][sciezka]`
Przykładowo:

- `sftp://yamo.skni.umsc.pl`
- `sftp://user@yamo.skni.umsc.pl`
- `sftp://user@yamo.skni.umsc.pl:2795`
- `sftp://user@yamo.skni.umsc.pl:2795/home/user`

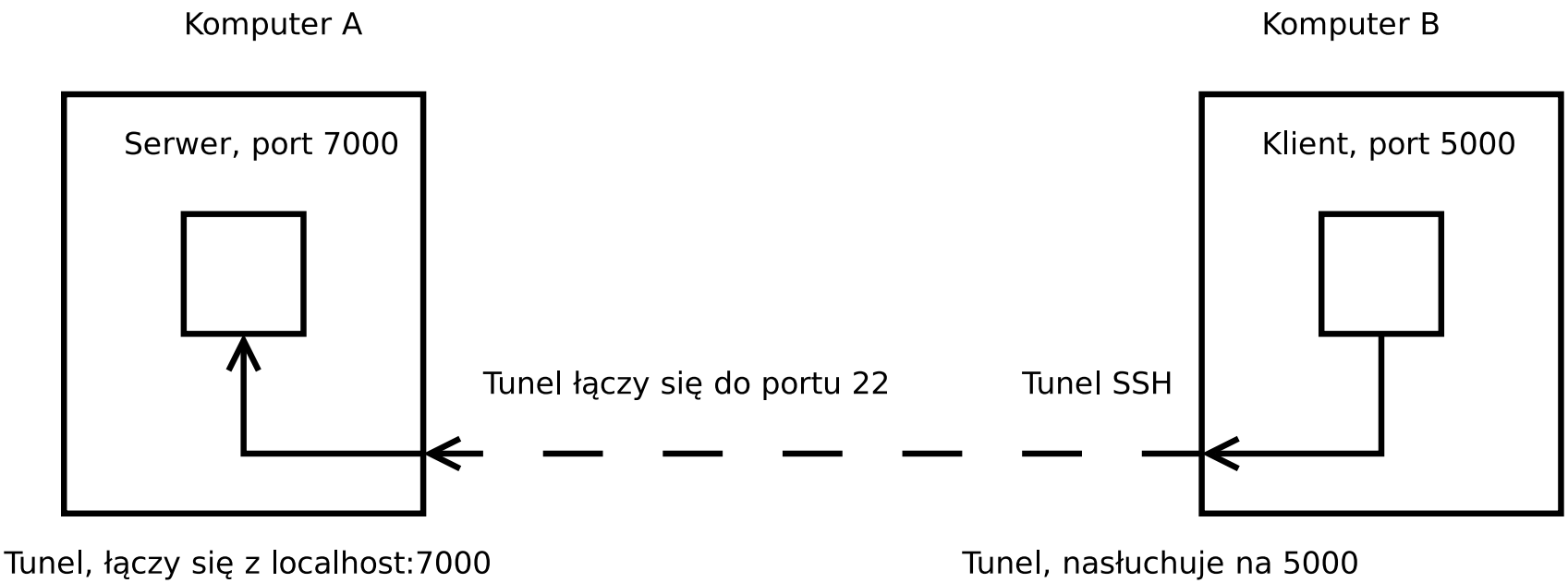
SFTP

Jeśli nie podamy nazwy użytkownika, program powinien nas zapytać o nią oraz hasło. Jeśli nazwa użytkownika była podana w ścieżce, a w systemie jest działający klucz SSH, to zostanie on uzyty.
Działanie tej metody w programach jest zależne od ich implementacji oraz dodatkowych pakietów w systemie.
Przykłady były testowane w Thunarze i PCManFM.

Tunel SSH

Jednym z dodatkowych zastosowań SSH jest możliwość utworzenia tunelu SSH. Umożliwia on bezpieczne przekierowanie gniazd sieciowych przez sieć. Przydają się one także przy przebijaniu się przez NAT, zapory sieciowe i inne przeszkody.

tunel SSH



Rysunek: ssh -L 5000:localhost:7000 user@server

Przekierowanie X11

Jedną z dodatkowych funkcjonalności SSH jest przekierowanie protokołu `X11`.

Przekierowanie X11

Jedną z dodatkowych funkcjonalności SSH jest przekierowanie protokołu `X11`.
Co to oznacza?

Przekierowanie X11

Jedną z dodatkowych funkcjonalności SSH jest przekierowanie protokołu `X11`.
Co to oznacza?
Dodając flagę `-X` możemy uruchomić tryb w którym możliwe będzie uruchomienie programów graficznych, które mimo tego, że wykonują się po stronie serwera, to wyświetlają się na naszym ekranie.

Przekierowanie X11

Jedną z dodatkowych funkcjonalności SSH jest przekierowanie protokołu `X11`.
Co to oznacza?
Dodając flagę `-X` możemy uruchomić tryb w którym możliwe będzie uruchomienie programów graficznych, które mimo tego, że wykonują się po stronie serwera, to wyświetlają się na naszym ekranie.
Oczywiście musi być to włączone w konfiguracji demona SSH a także lokalnie musimy mieć uruchomiony serwer `X11`.

Przykład X11

Przykładowo:
Wydając polecenie `ssh user@serwer -X` uruchomimy sesję SSH. Jeśli wpiszemy nazwę programu graficznego, np. `emacs`, to zamiast błędu lub uruchomienia wersji konsolowej otworzy nam się okno programu.

Przekierowanie X11

Przekierowanie takie ma swoje wady i zalety.

Zalety przekierowania X11

Zalety:

Zalety przekierowania X11

Zalety:

- Przezroczystość dla protokołu X11

Zalety przekierowania X11

Zalety:

- Przezroczystość dla protokołu X11
- Możliwość uruchamiania programów graficznych

Zalety przekierowania X11

Zalety:

- Przezroczystość dla protokołu X11
- Możliwość uruchamiania programów graficznych
- Umożliwia bezpieczne używanie protokołu X11 w niebezpiecznych sieciach lub za NATem

Wady przekierowania X11

Wady:

Wady przekierowania X11

Wady:

- Stabilność (niektóre programy lubią się wysypać)

Wady przekierowania X11

Wady:

- Stabilność (niektóre programy lubią się wysypać)
- Duży narzut protokołu w specyficznych przypadkach

Podsumowanie przekierowania X11

Podsumowując, na szybkim łączu i małym opóźnieniu protokół ten jest całkiem używalny. W przypadku problemów warto użyć `VNC` albo `Spice`.

Logowanie z użyciem SSH

Zaloguj się do systemu za pomocą SSH używając danych podanych na kartce. Nazwa hosta to `yamo.skni.umcs.pl`, a port to 2795.

Przygotowanie klucza SSH

Wygeneruj i wrzuć plik z kluczem na serwer tak, by nie trzeba było podawać hasła przy logowaniu.

Przygotowanie klucza SSH

Wygeneruj i wrzuć plik z kluczem na serwer tak, by nie trzeba było podawać hasła przy logowaniu.
Podpowiedź: do przekopiowania możesz użyć następującego polecenia

```
cat plik | ssh user@host "cat - > plik"
```

Kopiowanie plików z użyciem SCP

W katalogu domowym na zdalnej maszynie znajduje się kilka plików. Przekopiuj te pliki na swój komputer, spróbuj też kilka plików ze swojej maszyny przerzucić na serwer.

SSH config

Utwórz plik `~/.ssh/config` z taką zawartością, byś nie musiał podawać nazwy użytkownika oraz portu przy używaniu SSH.

sshfs

Podobnie jak przy SCP zamontuj zdalny katalog i przerzuć do niego kilka plików.

SFTP

Zamontuj w używanym przez siebie programie do obsługi plików katalog domowy na zdalnej maszynie i przejrzyj znajdujące się tam pliki.

tunel SSH

Na komputerze o adresie `yamo.skni.umcs.pl` działa demon SSH na porcie 2795. Na serwerze tym działa też prosty demon który odpowiada na połączenia na porcie 27070. Stwórz tunel SSH do tego gniazda i odczytaj wiadomość. Niech tunel po naszej stronie nasłuchuje na porcie 5000

X11

Na zdalnej maszynie są dostępne graficzne programy. Spróbuj jakiś uruchomić, na przykład `firefox` lub `gimp`.