# System operacyjny Linux

Paweł Rajba

pawel.rajba@continet.pl
http://kursy24.eu/

# Zawartość modułu 12

- Protokół SSH
  - Do czego to?
  - Oprogramowanie
  - Uruchamianie serwera SSH
  - Logowanie, nawiązywanie połączenia
  - Konfiguracja klienta, konfiguracja serwera
  - Klucze, zarządzanie kluczami
  - Przekazywanie portów
  - Bezpieczna wymiana plików

 

# Do czego to?

- Zwiększa bezpieczeństwo
- Pozwala zdalnie zarządzać systemem
- Umożliwia na bezpieczne kopiowanie plików
- Umożliwia tworzenie bezpiecznych tuneli

 

# Oprogramowanie

- Klienty
  - OpenSSH
  - PuTTY
  - freeSSHd
  - WinSSHD
  - F-Secure SSH
- Serwery
  - OpenSSH
  - WinSSHD
  - freeSSHd

 

# Uruchamianie serwera SSH

- Zarządzanie demonem SSH
  - /etc/init.d/sshd start|stop|restart|status

 

# Logowanie

- Do połączenia będziemy używać programu ssh
  - Składnia
    - ssh [-xX] [-l login] [-p port] [uzytkownik@]host [polecenie]
  - Opcje
    - -x — wyłącza przekierowanie X-ów,
    - -X — włącza przekierowanie X-ów,
    - -l login — podajemy nazwę użytkownika,
    - -p port — podajemy port, na który będziemy się łączyć.

 

# Logowanie

- Kilka uwag
  - Przekierowanie X-ów można też włączyć wprowadzając wpis "ForwardX11 yes" do pliku ~/.ssh/config
  - Podając polecenie zamiast uruchamiać powłokę spowodujemy wykonanie tego polecenia na zdalnym hoście
  - Przykłady
    - ssh -l pawel serwer
    - ssh pawel@serwer
    - ssh -X -l pawel serwer
    - ssh pawel@serwer /home/pawel/backup.sh

 

# Nazwiązanie połączenia

- Serwer wysyła:
  - klucz hosta (generowany raz na początku; 1024b)
  - klucz serwera (regenerowany co ok. godzinę; 768b)
- Klient
  - sprawdza klucz hosta
  - generuje liczbę 256 bitową, szyfruję ją kluczem serwera i wysyła do serwera

 

# Nazwiązanie połączenia

- Serwer
  - odszyfrowuje wysłaną liczbę (swoim kluczem prywatnym), która staje się kluczem sesji
  - wysyła klientowi listę obsługiwanych protokołów symetrycznych: 3DES, IDEA, ... oraz inne szczegóły
- Klient
  - konfrontuje parametry serwera z własnymi możliwościami i rozpoczyna transmisję danych

 

# Nazwiązanie połączenia

- Przy pierwszym logowaniu:
  - Otrzymujemy komunikat:
    pawel@cassiopeia:~> ssh 150.20.1.7
    The authenticity of host '150.20.1.7 (150.20.1.7)' can't be established.
    DSA key fingerprint is d0:be:40:46:8e:2d:66:12:ff:72:36:7e:97:45:9d:1b.
    Are you sure you want to continue connecting (yes/no)?
  - Odpowiadamy yes i dostajemy następnie
    Warning: Permanently added '150.20.1.7' (DSA) to the list of known hosts.
    pawel@150.20.1.7's password:
  - Podajemy hasło i zostajemy zalogowani
- Przy kolejnych logowaniach
  - klucz jest znajdowany w pliku ~/.ssh/known_hosts i jesteśmy od razu pytani o hasło

 

# Konfiguracja klienta

- Pliki konfiguracyjne
  - ,,dla wszystkich" – /etc/ssh/ssh_config
    - ten plik musi być do czytania dla wszystkich
  - ,,dla użytkownika" – $HOME/.ssh/config
    - ten plik powinien mieć prawa rw tylko dla właściciela
- Opcje konfiguracji
  - Host, Hostname, User,
  - Port, ForwardX11,
  - Compression, CompressionLevel,
  - Protocol

 

# Konfiguracja klienta

- Opcje konfiguracji
  - Host – otwiera sekcję konfiguracji dla hosta/grupy hostów; przy definicji można używać znaków specjalnych * i ?
  - Hostname – rzeczywista nazwa/adres hosta
  - User – użytkownik, na którego będzie logowanie
  - Port – port połączenia (domyślnie 22)
  - ForwardX11 – czy przekierowywać X-y (domyślnie no)
  - Compression – czy włączona kompresja (domyślnie no),
  - CompressionLevel – poziom kompresji 1-9 (domyślnie 6)
  - Protocol – wybór protokołów (domyślnie "2,1")

 

# Konfiguracja klienta

- Przykład konfiguracji
  - ```
    Host ii
      Hostname 150.10.20.30
      User pawel
      Port 22
      ForwardX11 yes
      Compression yes
      CompressionLevel 9
      Protocol 2,1
    Host ath
      Hostname 210.110.110.10
      User prajba
      Port 1022
      ForwardX11 no
    ```

 
    ```

# Konfiguracja serwera

- Plik konfiguracyjny
  - /etc/ssh/sshd_config
- Opcje konfiguracji
  - Port 22
  - Protocol 2
  - PermitRootLogin no
  - X11Forwarding no
  - AllowUsers, AllowGroups
  - DenyUsers, DenyGroups

 

# Konfiguracja serwera

- Można sterować dostępem poprzez odpowiednią edycję plików
  - /etc/hosts.allow
  - /etc/hosts.deny
  - /etc/nologin
  - /etc/shosts.equiv

 

# Klucze

- klucz hosta
  - pozwala na identyfikację hosta i zabezpieczenie przed niektórymi atakami
  - generowany przy pierwszym uruchomieniu sshd
- klucze użytkowników
  - generowane na żądanie
  - umożliwiają bezhasłowe logowanie

 

# Klucze

- klucze sesji
  - ustalane przy nawiązywaniu połączenia
  - wymieniane co pewien czas
  - wykorzystywane do szyfrowania algorytmem symetrycznym

 

# Zarządzanie kluczami

- ssh-keygen (1) – tworzy klucze ssh
  - Składnia
    - ssh-keygen [-q] [-b bity] -t typ [-N hasło] [-C komentarz] [-f plik]
  - Opcje
    - -q – wersja ,,cicha"
    - -b – określa liczbę bitów (domyślnie 1024)
    - -t typ – rodzaj kluczy: rsa1 (ver. 1), rsa i dsa (ver. 2)
    - -N hasło – określa hasło
    - -C komentarz – komentarz (zwykle user@host)
    - -f plik – określa, gdzie będzie zapisany klucz prywatny; klucz publiczny to plik.pub

 

# Zarządzanie kluczami

- ssh-keygen (2) – zmienia hasło klucza
  - Składnia
    - ssh-keygen -p [-P stare] [-N nowe] [-f plik]
  - Opcje
    - -P stare – określa aktualne hasło
    - -N nowe – określa hasło, które ma obowiązywać po zmianie
    - -f plik – określa plik klucza, dla którego chcemy zmienić hasło

 

# Zarządzanie kluczami

- ssh-keygen (3) – na podstawie klucz prywatnego generuje klucz publiczny
  - Składnia
    - ssh-keygen -y [-f plik]
  - Opcje
    - -f plik – określa lokalizację pliku z kluczem prywatnym

 

# Zarządzanie kluczami

- Pliki kluczy
  - $HOME/.ssh/identity
    - klucz prywatny rsa (ssh w wersji 1)
  - $HOME/.ssh/identity.pub
    - klucz publiczny rsa (ssh w wersji 1)
    - należy dopisać jego zawartość do pliku $HOME/.ssh/authorized_keys na hostach, na które chcemy się logować przy użyciu autentykacji RSA

 

# Zarządzanie kluczami

- Pliki kluczy c.d.
  - $HOME/.ssh/id_dsa
    - klucz prywatny dsa (ssh w wersji 2)
  - $HOME/.ssh/id_dsa.pub
    - klucz publiczny dsa (ssh w wersji 2)
    - również można kopiować do pliku authorized_keys
  - $HOME/.ssh/id_rsa
    - klucz prywatny rsa (ssh w wersji 2)
  - $HOME/.ssh/id_rsa.pub
    - klucz publiczny rsa (ssh w wersji 2)
    - również można kopiować do pliku authorized_keys

 

# Przykład 1

- Logowanie bez hasła
  - Tworzymy parę kluczy poleceniem
    ```
    # ssh-keygen -t rsa -N '' -C '' -f ~/.ssh/id_rsa
    ```
  - Klucz publiczny dopisujemy do pliku authorized_keys poleceniem
    ```
    # scp ~/.ssh/id_rsa.pub \
    login@komputer:.ssh/authorized_keys
    ```
  - Możemy się już logować na komputer bez hasła

 

# Przykład 2

- Logowanie superbezpieczne
  - Tworzymy parę kluczy poleceniem (hasło nie powinno by puste)
    ```
    # ssh-keygen -t dsa
    ```
  - Klucz publiczny dopisujemy do pliku authorized_keys poleceniem
    ```
    # scp ~/.ssh/id_rsa.pub \
    login@komputer:.ssh/authorized_keys
    ```
  - Ustawiamy opcje w pliku /etc/ssh/sshd_config
    - PermitRootLogin no
    - PubkeyAuthentication yes
    - AuthorizedKeysFile .ssh/authorized_keys
    - ChallengeResponseAuthentication no

 

# Przykład 2

- Logowanie superbezpieczne c.d.
  - Restartujemy usługę ssh
    - /etc/init.d/sshd restart
  - Od teraz logowanie jest tylko poprzez klucze
- Ale jak się zalogować spod Windows?
  - Trzeba skopiować klucz prywatny do Windows
  - Poprzez program puttygen zaimportować klucz openssh (load, podajemy hasło i save private key)
  - Przy definicji połączenia wybrać Connection | SSH | Auth i wskazać zaimportowany klucz prywatny, i sru

 

# Przekazywanie portów

- Lokalny port jest przekazywany na zdalny komputer
- Tworzy bezpieczny tunel pomiędzy naszym komputerem a komputerem zdalnym
- Ma znaczenie, gdy nie możemy ufać np. bezprzewodowej sieci lokalnej
- Przykład

```
# ssh -f -N  -L110:mail.serwer.pl:110
             -L25:mail.serwer.pl:25
             user@mail.serwer.pl
```

 

# Bezpieczna wymiana plików

- scp – bezpieczne kopiowanie plików
  - Składnia
    - scp [-12pqrv] [-l limit] [-P port] [[user@]host1:]plik1 [[user@]host2:]plik2
  - Opcje
    - -1, -2 — wymusza użycie protokołu w wersji 1 lub 2
    - -p — przy kopiowaniu zachowuje czas modyfikacji, czas dostępu i tryb pliku,
    - -q — wyłącza pasek postępu,

 

# Bezpieczna wymiana plików

- scp c.d.
  - Opcje c.d.
    - -r — rekurencyjnie kopiuje podkatalogi,
    - -v — dodatkowe komunikaty,
    - -l limit — określa limit łącza w Kbit/s,
    - -P port — określa port.
  - Przykłady

```
scp dane.txt serwer.dom.pl:/home/pawel
scp -r /home/pawel/* serwer.dom.pl:/home/pawel
scp serwer.dom.pl:/home/pawel/backup.tgz .
scp -r serwer.dom.pl:/home/pawel/dane.* katalog
```

 

# Bezpieczna wymiana plików

- sftp – bezpieczne ftp
  - Składnia
    - sftp [[user@]host[:plik [plik]]]
      kopiuje plik
    - sftp [[user@]host[:katalog[/]]]
      przechodzi w tryb interaktywny do katalogu katalog
    - sftp -b batchfile [user@]host
      wykonuje listę poleceń

 

# Bezpieczna wymiana plików

- sftp c.d.
  - Uwaga:
    - Przy wykonywaniu listy poleceń, jeżeli dowolne z poleceń get, put, rename, ln, rm, mkdir, chdir, ls, lchdir, chmod, chown, chgrp, lpwd, lmkdir zakończy się błędem, wykonywanie listy poleceń zostaje przerwane
    - Aby nie przerywać wykonywania listy poleceń, należy przed poleceniem umieścić znak "-" np.
      **`-rm -r katalog/*`**

 

# Bezpieczna wymiana plików

- sftp c.d.
  - Lista poleceń
    - bye – kończy sesję
    - cd path – przechodzi do path na zdalnym komputerze
    - chgrp grp path – zmienia grupę na grp dla path; grp musi być numerycznym GID
    - chmod mode path – zmienia tryb na mode dla path
    - chown own path – zmienia właściciela na own dla path; own musi być numerycznym UID
    - exit – kończy sesję
    - get remote-path [local-path] – pobiera zdalny plik
    - help – drukuje dostępne komendy lub info o poleceniu

 

# Bezpieczna wymiana plików

- sftp c.d.
  - Lista poleceń c.d.
    - lcd path – zmienia ścieżkę lokalną na path
    - lls [opcje] [path] – listuje zawartość katalogu na lokalnym hoście
    - lmkdir path – tworzy katalog na lokalnym hoście
    - ln plik nazwa – tworzy link symboliczny
    - lpwd – wyświetla katalog lokalny
    - ls [opcje] [path] – listuje zawartość katalogu na zdalnym hoście
    - lumask umask – ustawia maskę na lokalnym hoście
    - mkdir path – tworzy katalog na zdalnym hoście

 

# Bezpieczna wymiana plików

- sftp c.d.
  - Lista poleceń c.d.
    - progress – włącza pojawianie się paska progresu
    - put local-path [remote-path] – kopiuje plik na zdalny host
    - pwd – wyświetla bieżący katalog na zdalnym hoście
    - quit – kończy sesję
    - rename old new – zamienia nazwę
    - rm path – usuwa pliki
    - rmdir path – usuwa katalogi
    - symlink plik nazwa – tworzy link symboliczny
    - version – drukuje wersję protokołu sftp

 