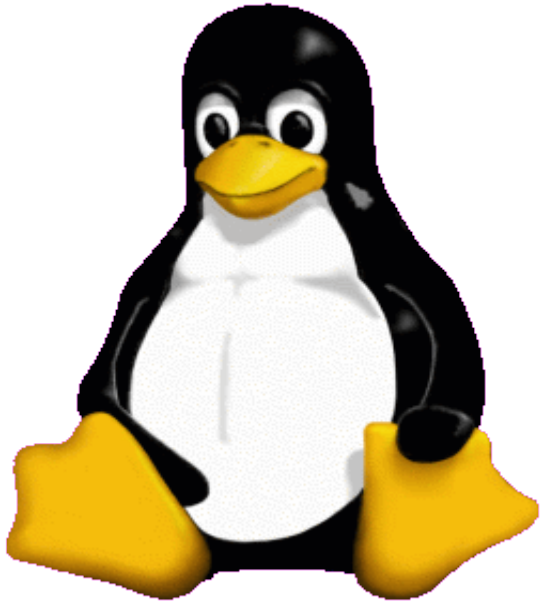
Bezpieczny serwer sieciowy Linux



Jacek Lach
jlach@star.iinf.polsl.gliwice.pl

Plan wykładu

- SSH
  - Typy autentykacji
  - Konfiguracja serwera
  - Konfiguracja klienta
  - Narzędzia towarzyszące

SSH - Secure Shell

- Udostępnia funkcjonalność r* - poleceń, zapewniając bezpieczeństwo przesyłanych danych poprzez ochronę:
  - przed podszywaniem się (IP, DNS)
  - przed podsłuchem
  - przed manipulacją przesyłanymi danymi
- Sesja nie może zostać przechwycona, sesja nie może być odtworzona poprzez odtworzenie przechwyconego ruchu

Szyfrowanie asymetryczne

- Nazywane również szyfrowaniem z kluczem publicznym. Wymagane są dwie różne, związane ze sobą wartości klucza: klucz publiczny oraz klucz prywatny.
- Obszary zastosowań algorytmów z kluczem publicznym:
  - integralność danych
  - poufność danych
  - uwierzytelnienie nadawcy

Szyfrowanie asymetryczne - klucze

- K1:
  - Klucz prywatny
  - Klucz publiczny
- K2:
  - Klucz prywatny
  - Klucz publiczny
- Wymiana kluczy publicznych - nie ma konieczności zapewnienia poufności wymiany kluczy publicznych

Szyfrowanie asymetryczne - integralność i poufność

Komunikacja: K1 -> K2

- Wiadomość oryginalna (K1)
- Szyfrowanie (K1 szyfruje wiadomość kluczem publicznym K2)
- Transfer danych
- Deszyfrowanie (K2 deszyfruje wiadomość przy użyciu swojego klucza prywatnego)
- Wiadomość oryginalna (K2)

Szyfrowanie asymetryczne - uwierzytelnienie nadawcy

Komunikacja: K1 -> K2

- Wiadomość oryginalna (K1)
- Szyfrowanie (K1 szyfruje wiadomość przy użyciu swojego klucza prywatnego)
- Transfer danych
- Deszyfrowanie (K2 deszyfruje wiadomość przy użyciu klucza publicznego K1)
- Wiadomość oryginalna (K2)

Szyfrowanie asymetryczne - podsumowanie

- Ważne: klucz prywatny musi być chroniony!
- Algorytmy generujące parę kluczy są złożone i czasochłonne
- Wymagania dotyczące wydajności ograniczają wykorzystanie szyfrowania asymetrycznego do zapewnienia poufności danych

SSH - sposoby uwierzytelniania

- Uwierzytelnienie hosta
- Uwierzytelnienie użytkownika:
  - Uwierzytelnienie zaufanego hosta (trusted host)
  - Uwierzytelnienie Rhosts i RSA
  - Uwierzytelnienie RSA
  - Uwierzytelnienie na podstawie hasła

Uwierzytelnienie hosta

- Żądanie połączenia ze strony klienta
- Serwer wysyła klucze publiczne:
  - Host Key (HK) - /etc/ssh/ssh_host_key.pub
  - Server Key (SK) - generowany co godzinę
- Klient weryfikuje klucz HK poprzez sprawdzenie plików ssh_known_hosts i known_hosts
- Aktualizacja plików lub odrzucenie połączenia (konfiguracja: StrictHostKeyChecking)
- Generacja klucza sesji (rnd() szyfrowany HK i HS)

Uwierzytelnienie użytkownika

- Uwierzytelnienie zaufanego hosta
- Opcja konfiguracji: RhostsAuthentication
- Zachowanie zgodności z rsh: dostęp bez podawania hasła dla hostów obecnych w pliku:
  - /etc/hosts.equiv /etc/shosts.equiv
  - ~/.rhosts ~/.shosts
- Zalecane wyłączenie r-poleceń oraz wyłączenie tego typu uwierzytelniania

Uwierzytelnienie użytkownika

- Uwierzytelnienie Rhosts i RSA
- Opcja konfiguracji: RhostsRSAAuthentication
- Pliki zaufanych hostów + uwierzytelnienie klienta przez serwer (klucz publiczny)
- Zalecenie wyłączenie tego typu uwierzytelniania

Uwierzytelnianie użytkownika

- Uwierzytelnianie RSA
- Opcja konfiguracji: RSAAuthentication
  - Klient wysyła klucz publiczny
  - Serwer porównuje otrzymany klucz z własną kopią oraz szyfruje nim losowe N
  - Klient deszyfruje N, wyznacza skrót N, wysyła
  - Serwer wyznacza skrót, porównuje go z otrzymanym
- Klucz prywatny może zostać zaszyfrowany - konieczność podania passphrase przy połączeniu

Połączenie - kolejność zdarzeń

- tcp_wrappers
- uwierzytelnienie hosta (f: dostęp zabroniony)
- RhostsAuthentication (s: dostęp)
- RhostsRSAAutthentication (s: dostęp)
- RSAAuthentication (s: dostęp)
- Hasło użytkownika (f: dostęp zabroniony)
- dostęp

Pierwsze połączenie

```
$ ssh 10.0.0.1

The authenticity of host '10.0.0.1 (10.0.0.1)' can't
   be established.

RSA key fingerprint is
   dd:b5:d3:47:59:fc:bd:ae:27:6e:30:94:46:08:52:d1.

Are you sure you want to continue connecting
   (yes/no)? yes

Warning: Permanently added '10.0.0.1' (RSA) to the
   list of known hosts.
```

Zmiana klucza serwera

```
$ ssh 10.0.0.1
@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@
@    WARNING: REMOTE HOST IDENTIFICATION HAS CHANGED!     @
@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@
IT IS POSSIBLE THAT SOMEONE IS DOING SOMETHING NASTY!
Someone could be eavesdropping on you right now (man-in-the-middle
   attack)!
It is also possible that the RSA host key has just been changed.
The fingerprint for the RSA key sent by the remote host is
dd:b5:d3:47:59:fc:bd:ae:27:6e:30:94:46:08:52:d1.
Please contact your system administrator.
Add correct host key in /home/userek/.ssh/known_hosts to get rid of
   this message.
Offending key in /home/userek/.ssh/known_hosts:10
RSA host key for 10.0.0.1 has changed and you have requested strict
   checking.
Host key verification failed.
```

/etc/sshd/sshd_config

- AllowHosts, DenyHosts
- AllowUsers, DenyUsers
- ListenAddress
- PasswordAuthentication
- PermitEmptyPasswords
- PermitRootLogin
- *Authentication

/etc/ssh/ssh_config

- FallBackToRsh
- PasswordAuthentication
- *Authentication
- StrictHostKeyChecking
- UseRsh

SSH - dodatki

- Secure Copy
- Zarządzanie kluczami
- Przekazywanie połączeń i proxy
- SFTP

Secure copy - scp

scp [-flags] [[user@]host1:]filename1 [[user@]host2:]filename2

- Zdalne kopiowanie, zasady uwierzytalniania użytkownika jak w ssh
- Szyfrowany kanał transmisyjny
- Użytkownik wywołujący polecenie musi autoryzować polecenie podając hasło zdalnego systemu

```
$ scp winshare.c userek@10.0.0.1:/tmp
userek@10.0.0.1's password:
winshare.c  100% |*****************************|  6658      00:00
```

Klucze użytkownika

- Generacja pary kluczy identyfikacyjnych użytkownika umożliwia logowanie bez podawania hasła.
- Klucz prywatny może być w celu zabezpieczenia dodatkowo zaszyfrowany hasłem.
- Pomoc w zarządzaniu kluczami udostępnia agent ssh.

Generacja kluczy użytkownika

```
$ ssh-keygen -t rsa
Generating public/private rsa key pair.
Enter file in which to save the key (/home/kreon/.ssh/id_rsa):
Enter passphrase (empty for no passphrase):
Enter same passphrase again:
Your identification has been saved in /home/userek/.ssh/id_rsa.
Your public key has been saved in/home/userek/.ssh/id_rsa.pub.
The key fingerprint is:
e1:28:3e:2c:ba:7d:4c:d4:e6:6d:a9:ca:0f:74:52:c2 userek@somewhere.pl
```

Klucze użytkownika

- Dystrybucja kluczy do zdalnych serwerów
- Konieczność wpisywania passphrase w celu rozszyfrowania klucza prywatnego
- Automatyzacja:
  - ssh-agent
  - ssh-add

ssh-keygen i użytkownik root

- Możliwość wygenerowania pary kluczy użytkownika
- Możliwość wygenerowania pary kluczy serwera
- Nie należy ustawiać passphrase dla prywatnego klucza serwera - uniemożliwi to odczyt klucza podczas autoryzacji połączenia

Przekazywanie

- Przekazywanie sesji Xwindow (DISPLAY, MIT-MAGIC-COOKIE)
- Przekazywanie:
  - lokalne
  - zdalne
- Blokowanie:
  - tcp_wrappers: sshdfwd-port: host
  - sshd_config: AllowTcpForwarding
  - kompilacja: --disable-client-[port | serwer]-forwardings

SFTP

- sftp-server - program podsystemu ssh umożliwiający wykorzystanie protokołu SFTP
- sftp - program klienta udostępniający bezpieczny transfer danych, oferuje funkcjonalność programu ftp; w połączeniu z uwierzytalnieniem opartym o klucze użytkownika umożliwia automatyczny transfer plików