
Wiktor Satora

Klasa 3IB

Grupa 2

27 kwiecień 2020

Ćwiczenie 47:

Konfiguracja SSH

Serwer: Ubuntu server 16.04

Klient: Ubuntu Desktop 16.04

Spis treści

1. Instalacja pakietów.....	3
2. Ustawienie i sprawdzenie działania.....	4
3. Kopiowanie za pomocą SCP.....	6
4. Kopiowanie za pomocą SFTP.....	7
5. Definicje.....	8

1. Instalacja pakietów

```
root@satora:~# apt-get install openssh-server_
```

Rysunek 1 Instalacja pakietu na serwerze

```
root@satora:~# systemctl status ssh
• ssh.service - OpenBSD Secure Shell server
  Loaded: loaded (/lib/systemd/system/ssh.service; enabled; vendor preset: enabled)
  Active: active (running) since pon 2020-04-27 13:04:54 CEST; 1min 56s ago
    Main PID: 2138 (sshd)
      CGroup: /system.slice/ssh.service
              └─2138 /usr/sbin/sshd -D

kwi 27 13:04:54 satora systemd[1]: Starting OpenBSD Secure Shell server...
kwi 27 13:04:54 satora sshd[2138]: Server listening on 0.0.0.0 port 22.
kwi 27 13:04:54 satora sshd[2138]: Server listening on :: port 22.
kwi 27 13:04:54 satora systemd[1]: Started OpenBSD Secure Shell server.
root@satora:~# _
```

Rysunek 2 Sprawdzenie działania usługi

```
user@satora:~$ ssh localhost
user@localhost's password:
Welcome to Ubuntu 16.04.6 LTS (GNU/Linux 4.4.0-177-generic x86_64)

 * Documentation:  https://help.ubuntu.com
 * Management:    https://landscape.canonical.com
 * Support:       https://ubuntu.com/advantage

 * Ubuntu 20.04 LTS is out, raising the bar on performance, security,
   and optimisation for Intel, AMD, Nvidia, ARM64 and Z15 as well as
   AWS, Azure and Google Cloud.

   https://ubuntu.com/blog/ubuntu-20-04-lts-arrives

11 pakietów może zostać zaktualizowanych.
1 aktualizacja jest aktualizacją zabezpieczeń.

New release '18.04.4 LTS' available.
Run 'do-release-upgrade' to upgrade to it.

Last login: Mon Apr 27 13:58:07 2020
user@satora:~$ _
```

Rysunek 3 Sprawdzenie logowania się lokalnie (użytkownik nie będący rootem)

```
user@satora:~$ wylogowanie
Connection to localhost closed.
user@satora:~$
```

Rysunek 4 Aby wylogować się używamy kombinacji klawiszowej Ctrl+D

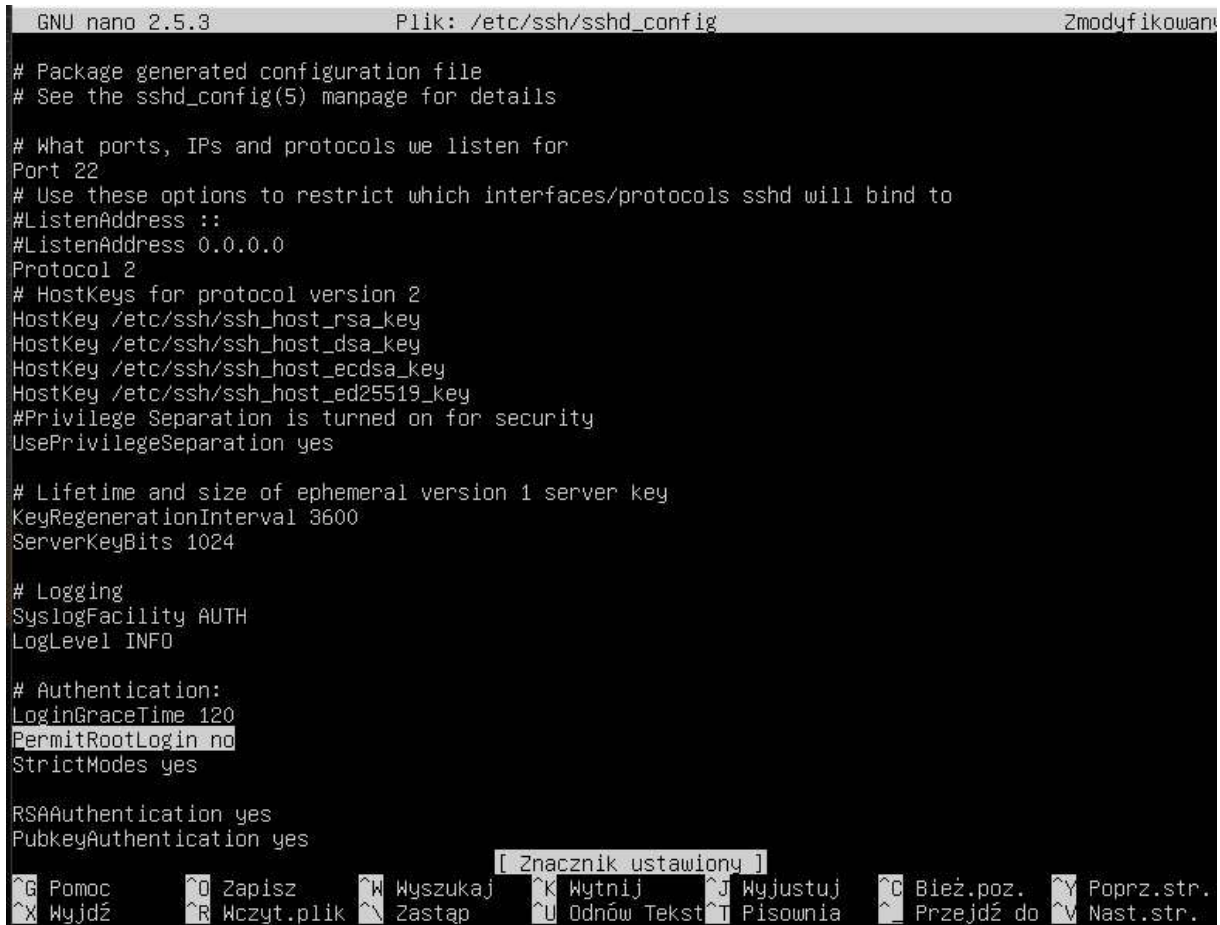
```
root@wiktork:/home/uczen# apt-get install openssh-client
```

Rysunek 5 Instalacja pakietu na kliencie

2. Ustawienie i sprawdzenie działania

```
user@satora:~$ nano /etc/ssh/sshd_config
```

Rysunek 6 Otwieramy plik



```
GNU nano 2.5.3          Plik: /etc/ssh/sshd_config          Zmodyfikowa...
# Package generated configuration file
# See the sshd_config(5) manpage for details

# What ports, IPs and protocols we listen for
Port 22
# Use these options to restrict which interfaces/protocols sshd will bind to
#ListenAddress ::
#ListenAddress 0.0.0.0
Protocol 2
# HostKeys for protocol version 2
HostKey /etc/ssh/ssh_host_rsa_key
HostKey /etc/ssh/ssh_host_dsa_key
HostKey /etc/ssh/ssh_host_ecdsa_key
HostKey /etc/ssh/ssh_host_ed25519_key
#Privilege Separation is turned on for security
UsePrivilegeSeparation yes

# Lifetime and size of ephemeral version 1 server key
KeyRegenerationInterval 3600
ServerKeyBits 1024

# Logging
SyslogFacility AUTH
LogLevel INFO

# Authentication:
LoginGraceTime 120
PermitRootLogin no
StrictModes yes

RSAAuthentication yes
PubkeyAuthentication yes

[ Znacznik ustawiony ]
^G Pomoc      ^O Zapisz     ^W Wyszukaj   ^K Wytnij    ^J Wyjustuj  ^C Bież.poz.  ^Y Poprz.str.
^X Wyjdź     ^R Wczyt.plik ^_ Zastąp    ^U Odnów Tekst ^T Pisownia  ^B Przejdź do ^V Nast.str.
```

Rysunek 7 Sprawdzamy czy root może się logować (domyślnie NIE MOŻE)

```
root@satora:/home/user# systemctl stop ssh
root@satora:/home/user# systemctl start ssh
root@satora:/home/user# systemctl status ssh
• ssh.service - OpenBSD Secure Shell server
   Loaded: loaded (/lib/systemd/system/ssh.service; enabled; vendor preset: enabled)
   Active: active (running) since pon 2020-04-27 14:11:42 CEST; 2s ago
   Process: 1898 ExecStartPre=/usr/sbin/sshd -t (code=exited, status=0/SUCCESS)
   Main PID: 1901 (sshd)
   Tasks: 1
   Memory: 716.0K
   CPU: 20ms
   CGroup: /system.slice/ssh.service
           └─1901 /usr/sbin/sshd -D

kwi 27 14:11:42 satora systemd[1]: Starting OpenBSD Secure Shell server...
kwi 27 14:11:42 satora sshd[1901]: Server listening on 0.0.0.0 port 22.
kwi 27 14:11:42 satora sshd[1901]: Server listening on :: port 22.
kwi 27 14:11:42 satora systemd[1]: Started OpenBSD Secure Shell server.
root@satora:/home/user# ssh localhost
root@localhost's password:
Permission denied, please try again.
root@localhost's password:
```

Rysunek 8 Reset usługi

```
uczen@wikt0r:~$ ssh user@satora
The authenticity of host 'satora (192.168.1.1)' can't be established.
ECDSA key fingerprint is SHA256:Ci8iDWgBcF84CKURdw6eI4ciBmM3Mj0HC/VWELFApdk.
Are you sure you want to continue connecting (yes/no)? yes
Warning: Permanently added 'satora,192.168.1.1' (ECDSA) to the list of known hosts.
user@satora's password:
Welcome to Ubuntu 16.04.6 LTS (GNU/Linux 4.4.0-177-generic x86_64)

 * Documentation:  https://help.ubuntu.com
 * Management:    https://landscape.canonical.com
 * Support:       https://ubuntu.com/advantage

 * Ubuntu 20.04 LTS is out, raising the bar on performance, security,
   and optimisation for Intel, AMD, Nvidia, ARM64 and Z15 as well as
   AWS, Azure and Google Cloud.

   https://ubuntu.com/blog/ubuntu-20-04-lts-arrives

11 pakietów może zostać zaktualizowanych.
1 aktualizacja jest aktualizacją zabezpieczeń.

New release '18.04.4 LTS' available.
Run 'do-release-upgrade' to upgrade to it.

Last login: Mon Apr 27 18:03:07 2020 from 192.168.1.6
user@satora:~$
```

Rysunek 9 Sprawdzamy logowanie zdalne z klienta (zalogowano zdalnie na serwerze)

```
user@satora:~$ pwd
/home/user
user@satora:~$ ls
user@satora:~$ touch sprawdzenie.txt
user@satora:~$ ls
sprawdzenie.txt
user@satora:~$
```

Rysunek 10 Możemy z tego miejsca utworzyć plik na serwerze

```
root@satora:/home/user# ls
sprawdzenie.txt
```

Rysunek 11 Pojawił się na Ubuntu Server

```
user@satora:~$ wylogowanie
Connection to satora closed.
uczen@wikt0r:~$
```

Rysunek 12 „Ctrl+D” kończy połączenie

3. Kopiowanie za pomocą SCP

```
uczen@wiktork:~$ scp user@satora:sprawdzenie.txt /home/uczen
user@satora's password:
sprawdzenie.txt                                100%   0   0.0KB/s   00:00
uczen@wiktork:~$
```

Rysunek 13 Kopiujemy plik z serwera na klienta

Składnia polecenia:

scp [ŹRÓDŁO] [CEL]

Ścieżkę lokalną wprowadzamy zwyczajnie, natomiast ścieżkę maszyny zdalnej poprzedzamy: (nazwa_użytkownika_zdalnego)@(nazwa_hosta_zdalnego): (dopiero_tutaj_ścieżka)

```
uczen@wiktork:~$ ls
Dokumenty      Muzyka  Pobrane  Pulpit  Szablony
examples.desktop  Obrazy  Publiczny  sprawdzenie.txt  Wideo
uczen@wiktork:~$
```

Rysunek 14 Rzeczywiście plik się skopiował

```
uczen@wiktork:~$ echo spr>sprawdzenie2.txt
```

Rysunek 15 Tworzymy plik tym razem na serwerze

```
uczen@wiktork:~$ scp sprawdzenie2.txt user@satora:/home/user
user@satora's password:
sprawdzenie2.txt                                100%   4   0.0KB/s   00:00
uczen@wiktork:~$
```

Rysunek 16 I przesyłamy w drugą stronę

```
user@satora:~$ ls
sprawdzenie2.txt  sprawdzenie.txt
user@satora:~$
```

Rysunek 17 Jak widać na Ubuntu Server przesłało się

4. Kopiowanie za pomocą SFTP

```
user@satora:~$ echo wiktora>wiktora.txt
user@satora:~$ echo satora>satora.txt
user@satora:~$ ls
satora.txt  sprawdzenie2.txt  sprawdzenie.txt  wiktora.txt
user@satora:~$ _
```

Rysunek 18 Tworzymy kilka nowych plików na serwerze

```
uczen@wiktora:~$ echo test>test.txt
uczen@wiktora:~$ ls
Dokumenty      Obrazy      Pulpit      Szablony
examples.desktop Pobrane     sprawdzenie2.txt test.txt
Muzyka         Publiczny   sprawdzenie.txt  Wideo
uczen@wiktora:~$
```

Rysunek 19 Oraz na kliencie

```
uczen@wiktora:~$ sftp user@satora
user@satora's password:
Connected to satora.
sftp> ls
satora.txt      sprawdzenie.txt      sprawdzenie2.txt      wiktora.txt
sftp> get wiktora.txt
Fetching /home/user/wiktora.txt to wiktora.txt
/home/user/wiktora.txt      100%  7  0.0KB/s  00:00
sftp> get satora.txt
Fetching /home/user/satora.txt to satora.txt
/home/user/satora.txt      100%  7  0.0KB/s  00:00
sftp> put test.txt
Uploading test.txt to /home/user/test.txt
test.txt      100%  5  0.0KB/s  00:00
sftp> ls
satora.txt      sprawdzenie.txt      sprawdzenie2.txt      test.txt
wiktora.txt
sftp> quit
uczen@wiktora:~$ ls
Dokumenty      Obrazy      Pulpit      sprawdzenie.txt  Wideo
examples.desktop Pobrane     satora.txt   Szablony         wiktora.txt
Muzyka         Publiczny   sprawdzenie2.txt test.txt
uczen@wiktora:~$
```

Rysunek 20 Używając polecenie „sftp” łączymy się z serwerem. Pobraliśmy 2 pliki (wiktora.txt oraz satora.txt), a wysłaliśmy test.txt

Składnia polecenia:

sftp (nazwa_użytkownika_zdalnego)@(nazwa_hosta_zdalnego)

Polecenie do zarządzania plikami i katalogami np. „ls”, „pwd”, „mkdir” itd. działają normalnie

Polecenie „get” pobiera plik

Polecenie „put” wysyła plik

5. Definicje

SSH (ang. secure shell) to standard protokołów komunikacyjnych używanych w sieciach komputerowych TCP/IP, w architekturze klient-serwer a od wersji 2 nawet w architekturze serwer-klient.

W ścisłym znaczeniu SSH to tylko następca protokołu Telnet, służącego do terminalowego łączenia się ze zdalnymi komputerami. SSH różni się od Telnetu tym, że transfer wszelkich danych jest zaszyfrowany oraz możliwe jest rozpoznawanie użytkownika na wiele różnych sposobów. W szerszym znaczeniu SSH to wspólna nazwa dla całej rodziny protokołów, nie tylko terminalowych, lecz także służących do przesyłania plików (SCP, SFTP), zdalnej kontroli zasobów, tunelowania i wielu innych zastosowań. Wspólną cechą wszystkich tych protokołów jest identyczna z SSH technika szyfrowania danych i rozpoznawania użytkownika. Obecnie protokoły z rodziny SSH praktycznie wyparły wszystkie inne mniej bezpieczne protokoły, takie, jak np. rlogin czy RSH.

Secure copy lub **SCP** – bezpieczny transfer plików pomiędzy lokalnym a zdalnym lub między zdalnymi komputerami, używając protokołu Secure Shell (SSH). Skrót SCP odnosi się do dwóch powiązanych ze sobą rzeczy: protokołu SSH oraz polecenia cp.

SFTP (ang. SSH File Transfer Protocol) – protokół komunikacyjny typu klient-serwer, który umożliwia przesyłanie plików poprzez sieć TCP/IP.

Przesyłając plik przy użyciu protokołu FTP uzyskujemy dobre przepływności, ale nie zyskujemy bezpieczeństwa – nasze hasła i dane nie są szyfrowane podczas przysyłania, co potencjalnie stwarza zagrożenie ich kradzieży. Znaczną poprawę bezpieczeństwa przynosi protokół SFTP, który nie wymaga obecności serwera FTP, a przesyłane dane są szyfrowane z wykorzystaniem klucza szyfrującego.

SFTP nie powinien być mylony z protokołem FTPS, który jest rozszerzeniem protokołu FTP.