3. Logowanie (notatki)

Plik ten zawiera przydatne komendy wykorzystywane w kursie

Wylogowywanie i wyłączanie maszyny:

```
exit #wyloguj
logout # wyloguj
sudo poweroff # wyłącz maszynę
sudo reboot # zrestartuj maszynę
```

Pliki profilowe: profile i passwd:

```
more /etc/profile
more /etc/passwd
grep root /etc/passwd
```

Komunikaty logowania

```
more /etc/issue
more /etc/issue.net
more /etc/motd
```

login

```
sudo login
touch ~/.hushlogin

echo $HOME
echo $SHELL
echo $PATH
echo $LOGNAME
echo $MAIL
echo $TERM

sudo login -p

vim /etc/login.defs
egrep "^[^#]" /etc/login.defs
```

powiązane pliki

```
/var/run/utmp # lista bieżących sesji pracy
/var/log/wtmp # lista poprzednich sesji pracy
/etc/passwd # informacje o kontach użytkowników
/etc/shadow # zakodowane hasła i informacje o ich ważności
/etc/motd # plik 'wiadomości dnia'
/etc/nologin # zapobiega logowaniu innych niż root
/etc/ttytype # lista typów terminali
$HOME/.hushlogin # zapobiega wyświetlaniu wiadomości dnia
```

Przeglądania plików utmp:

```
who # używa pliku /var/run/utmp
last # używa pliku /var/log/wtmp
sudo lastb # /var/log/btmp
```

Przeglądania plików utmp c.d.:

```
utmpdump /var/run/utmp
utmpdump /var/log/wtmp
sudo utmpdump /var/log/btmp

sudo nano /etc/gdm3/custom.conf
```

Automatyczne logowanie | Jak wyłączyć ?

```
sudo nano /etc/gdm3/custom.conf
---
# AutomaticLoginEnable = true
# Automatic Login = [użytkownik1 ]
```

Jak wyświetlić dostępnych użytkowników na VPS

```
less /etc/passwd
cut -d : -f 1 /etc/passwd

cat /etc/group
cut -d : -f 1 /etc/group

w
# prosty sposób na wyświetlenie wszystkich aktualnie zalogowanych użytkowników, ich czasu zalogowania i aktualnie używanego polecenia
who
```

Lab - video 1

```
sudo su root
whoami

for l in {1..10}; do useradd student$l; done
cat /etc/passwd
cat /etc/shadow

for l in {1..10}; do echo student$l:nowehaslo | chpasswd; done
cat /etc/shadow
```

Stworzyliśmy użytkowników i nadaliśmy im hasła. Zalogujmy się na ich konta

```
su student1
whoami
exit
```

```
su student5
whoami
exit
clear
```

Teraz na różne sposoby zablokujmy użytkownikom dostęp do konta

```
usermod -s /usr/srbin/nologin student1
cat /etc/passwd | grep student1

usermod -s /bin/false student2
cat /etc/passwd | grep student2

su student1
# This account is currently not available.
su student2
whoami

vi /etc/nologin.txt # Jakis komunikat dla studenta 1

cat /etc/shadow
passwd -l student3
su student3
whoami
su student3

passwd -u student3
whoami

usermod -L student4
su student4
clear

touch /etc/nologin
CTRL+ALT+F3
# (F3-F6) przełączanie do terminala tekstowego tty - teletypewriter w celu otwarcia
nowej sesji i próby zalogowania na innego użytkownika
CTRL+ALT+F1
# (F1-F2) przełączanie z terminala tekstowego tty spowrotem do graficznego GUI
vi /etc/nologin # Prace serwisowe 2021-24-01

# Usuńmy pliki nologin, tak by umożliwić dalsze logowanie do kont użytkowników
rm /etc/nologin.txt
rm /etc/nologin
```

Więcej o tty

Dzienniki logów

```
# Debian / Ubuntu
tail -100 /var/log/auth.log
```

```
# Redhat / CentOS
tail -100 /var/log/secure

grep -c 'Apr 23.*Failed password' /var/log/auth.log
grep 'student' /var/log/auth.log

sudo less /var/log/auth.log

last
lastlog
```

Zdalny dostęp - telnet

```
# telnet
telnet localhost
```

Zdalny dostęp – rsh

```
# rsh
rsh moj-serwer.com ls /home/student
```

Unikaj niezaszyfrowanej komunikacji

```
# Redhat / CentOS
yum erase xinetd ypserv tftp-server telnet-server rsh-server

# Debian / Ubuntu
sudo apt-get --purge remove xinetd nis yp-tools tftpd atftpd tftpd-hpa telnetd rsh-
server rsh-redone-server
```

Jak SSH uwierzytelnia użytkowników

```
~/.ssh/authorized_keys
```

Lab - video 2

INSTALUJEMY I URUCHAMIAMY SSH SERVER

```
# Otwieramy terminal CTRL+ALT+TAB
sudo apt update
sudo apt install openssh-server
sudo systemctl status ssh
sudo ufw allow ssh

netstat -antp
netstat -antp | grep 22
```

Tworzymy nowego użytkownika **karmaz**. Analogicznie możesz stworzyć drugiego, nowego użytkownika np. **student**

```
# Komendy przydatne w celu stworzenia użytkownika karmaz
sudo useradd karmaz -m # tworzy użytkownika karmaz wraz z jego folderem domowym
echo karmaz:nowehaslo | sudo chpasswd # ustala hasło użytkownika "nowehaslo"

# Następnie zmieniamy shell użytkownika karmaz:
sudo chsh -s /bin/bash karmaz # zmieniamy shell z /bin/sh na /bin/bash
sudo usermod karmaz -s /bin/bash
# alternatywnie mogliśmy dodać do useradd -s /bin/bash

# Możemy również dodać go do grupy sudo (czyli umożliwić wykonywanie komend na prawach root'a)
groups karmaz
usermod -aG sudo karmaz
groups karmaz

# Logujemy się na nowo stworzonego użytkownika karmaz
su karmaz # lub CTRL+ALT+F6 i następnie uwierzytelniamy się: karmaz:nowehaslo
```

Ostatecznie logujemy się przez SSH z konta **student** na konto **karmaz** podając hasło

```
student@KURS:~$
ssh karmaz@localhost

karmaz@KURS:~$
sudo systemctl status ssh
sudo systemctl stop ssh
sudo systemctl start ssh
exit
```

```
karmaz@KURS:~$
ip a
# zczytujemy ip  maszyny, w tym przypadku 192.168.242.132

student@KURS:~$
ssh karmaz@192.168.242.132 # tutaj należy zamienić ip własnej maszyny w sieci lokalnej
exit
```

GENERUJEMY KLUCZ SSH

Teraz wygenerujemy parę kluczy private-public dla użytkownika **student**, tak by móc bezpieczenie komunikować się pomiędzy kontami

```
student@KURS:~$
cd && ssh-keygen
# Możesz to zrobić sam, zgodnie z tym tutorialem
# https://docs.github.com/en/github/authenticating-to-github/generating-a-new-ssh-key-and-adding-it-to-the-ssh-agent
exit
```

Mocniejszy klucz (i nadpisanie pary kluczy), zmiana i usuwanie hasła do klucza SSH

```
 ssh-keygen -b 4096
 ssh-keygen -p # zmiana passphrase
```

```
ssh-keygen -l # odcisk palca
```

UDOSTĘPNIAMY NASZ KLUCZ PUBLICZNY

Korzystając z utility ssh-copy-id przekażemy (innej) maszynie nasz klucz publiczny. Po raz ostatni logujemy się hasłem na konto użytkownika przez SSH, w celu przekazania naszego klucza publicznego

```
ssh-copy-id student@localhost
ssh-copy-id karmaz@localhost
```

Od teraz możemy zalogować się z konta np. **student** na konto **karmaz** bez podawania hasła, lecz z wykorzystaniem pary kluczy

```
student@KURS:~$
ssh student@localhost
student@KURS:~$
whoami #student
exit

student@KURS:~$
ssh karmaz@localhost
karmaz@KURS:~$
whoami # karmaz
exit
```

Co stało się na serwerze SSH

```
student@KURS:~$
ls .ssh/
cat .ssh/authorized_keys
cat .ssh/id_rsa.pub
```

Alternatywnie do ssh-copy-id, możemy przesłać nasz klucz stosując następującą komendę:

```
cat ~/.ssh/id_rsa.pub | ssh student@localhost "mkdir -p ~/.ssh && cat >>
~/.ssh/authorized_keys"
cat ~/.ssh/id_rsa.pub | ssh karmaz@localhost "mkdir -p ~/.ssh && cat >>
~/.ssh/authorized_keys"

student@KURS:~$
cat .ssh/authorized_keys

# echo >>  authorized_keys (dopisanie do pliku) - używamy tej opcji
# echo > authorized_keys  (nadpisanie pliku)

ssh localhost
```

WYKONYWANIE ZDALNYCH KOMEND I ZAMYKANIE POŁĄCZEŃ PRZEZ SSH

```
student@KURS:~$
ssh karmaz@localhost whoami
# karmaz
```

SSH NA INNYM PORCIE, NIŻ 22, NP. 4242

```
ssh -p 4242 username@localhost
clear
```

SSH CONFIG

```
cd .ssh/
clear
ls
ls -lah
vi config
#HOST remote_alias
#    HostName localhost
#    Port 4242
#    User karmaz
# w celu zapisania wprowadź :wq
clear
```

AGENT SSH, LOGOWANIE BEZ WPROWADZANIA HASŁA (PASSPHRASE)

```
cd .ssh/
ssh-agent
eval $(ssh-agent)

ssh add

ssh -A karmaz@localhost
```

WYŁĄCZENIE UWIERZYTELNIANIA HASŁEM

```
student@KURS:~$
ssh karmaz@localhost

karmaz@KURS:~$
clear
ls -lah /etc/ssh/sshd_config

sudo vi /etc/ssh/sshd_config
# PasswordAuthentication no
PasswordAuthentication no
# Odkomentowujemy i zapisujemy plik

sudo service ssh restart

ssh karmaz@localhost
# karmaz@localhost: Permission denied (publickey).

ssh student@localhost
student@KURS:~$
exit
```

ZMIANA PORTU SSH NA ZDALNYM SERWERZE

```
ssh student@localhost
student@KURS:~$
sudo vi /etc/ssh/sshd_config
# Port 22
Port 4242
# Odkomentowujemy i zapisujemy plik

sudo service ssh restart # Ubuntu & Debian
sudo service sshd restart # CentOS & Fedora

ssh student@localhost
# ssh: connect to host localhost port 22: Connection refused
netstat -antp
clear

ssh -p 4242 student@localhost
clear
exit
```

OGRANICZENIE LOGOWANIA PRZEZ SSH

```
ssh -p 4242 student@localhost
student@KURS:~$
sudo vi /etc/ssh/sshd_config
# PermitRootLogin prohibit-password
PermitRootLogin no
# Odkomentowujemy i zapisujemy plik

sudo service ssh restart # Ubuntu & Debian
```

Umożliwienie rootowi wykonywanie tylko wybranych komend (np. kopie zapasowe lub "ls -lah")

```
student@KURS:~$
sudo vi /root/.ssh/authorized_keys
# command="ls -lah" ssh-rsa AAAA***= root@KURS
sudo vi /etc/ssh/sshd_config
# PermitRootLogin prohibit-password
PermitRootLogin forced-commands-only
# i zapisujemy plik

sudo service ssh restart
```