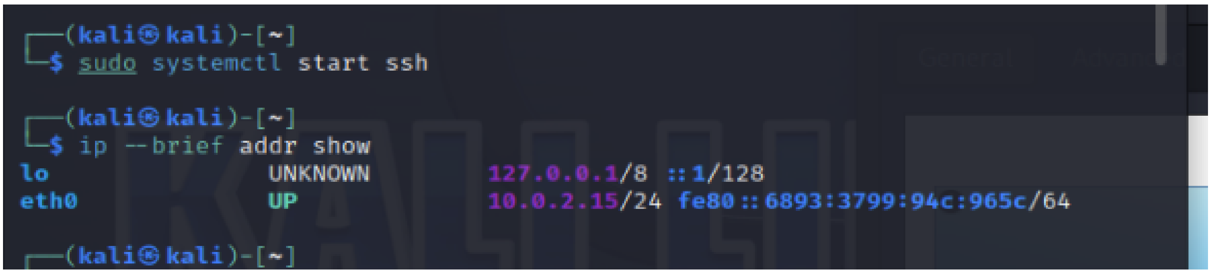
zad 1

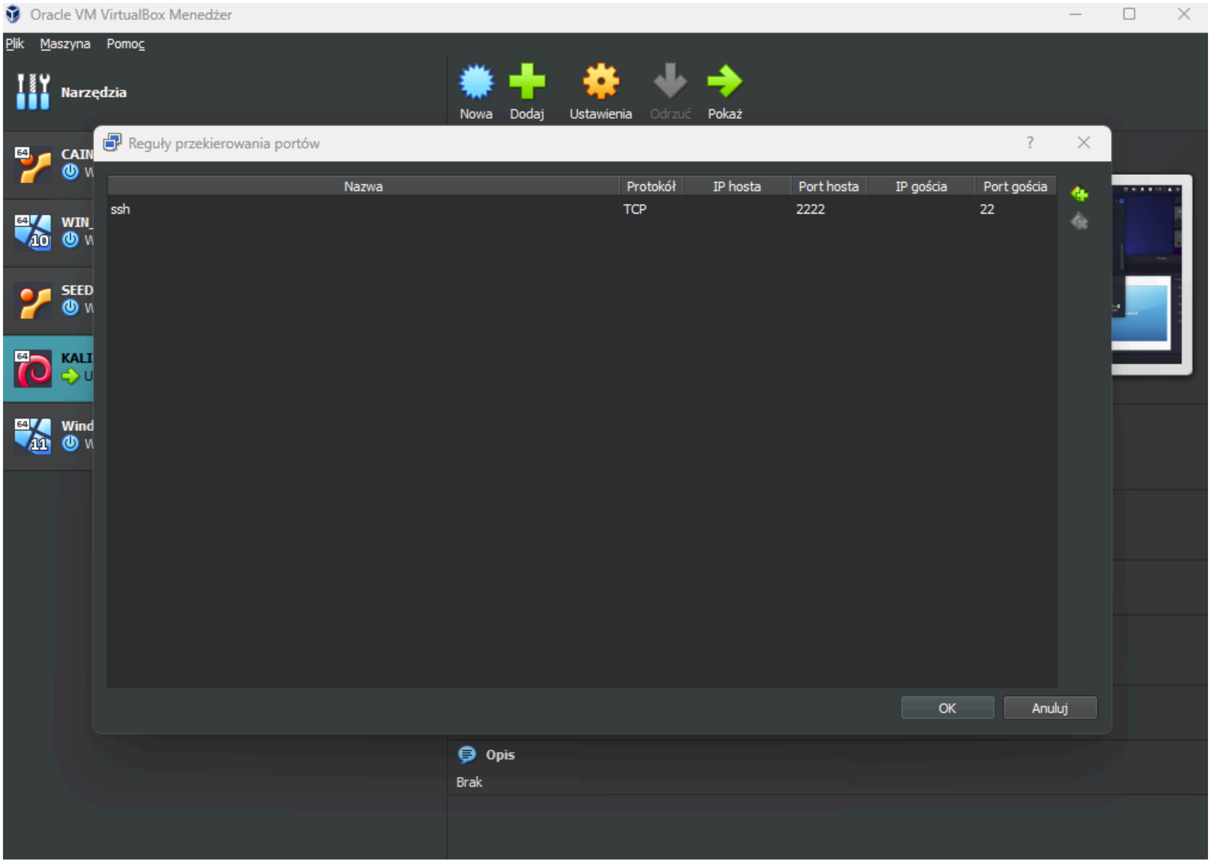
https://averagelinuxuser.com/ssh-into-virtualbox/

Na maszynie wirtualnej wystartowałem ssh.



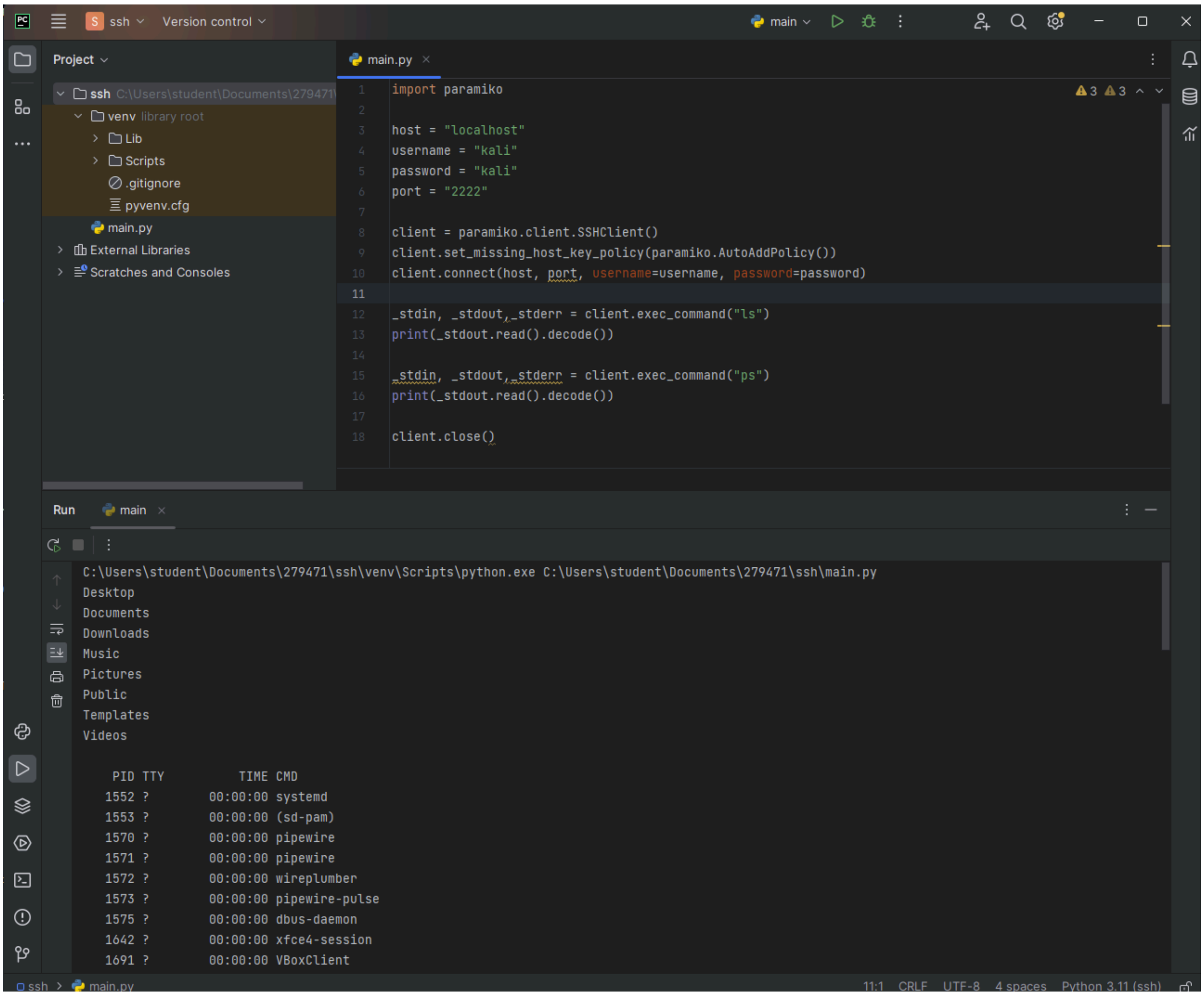
```
┌──(kali㉿kali)-[~]
└─$ sudo systemctl start ssh

┌──(kali㉿kali)-[~]
└─$ ip --brief addr show
lo               UNKNOWN        127.0.0.1/8 ::1/128
eth0             UP             10.0.2.15/24 fe80::6893:3799:94c:965c/64

┌──(kali㉿kali)-[~]
```

W ustawieniach maszyny dodałem przekierowanie portów



Nazwa	Protokół	IP hosta	Port hosta	IP gościa	Port gościa
ssh	TCP		2222		22

Skrypt pythonowy i output:



```python
import paramiko

host = "localhost"
username = "kali"
password = "kali"
port = "2222"

client = paramiko.client.SSHClient()
client.set_missing_host_key_policy(paramiko.AutoAddPolicy())
client.connect(host, port, username=username, password=password)

_stdin, _stdout,_stderr = client.exec_command("ls")
print(_stdout.read().decode())

_stdin, _stdout,_stderr = client.exec_command("ps")
print(_stdout.read().decode())

client.close()
```

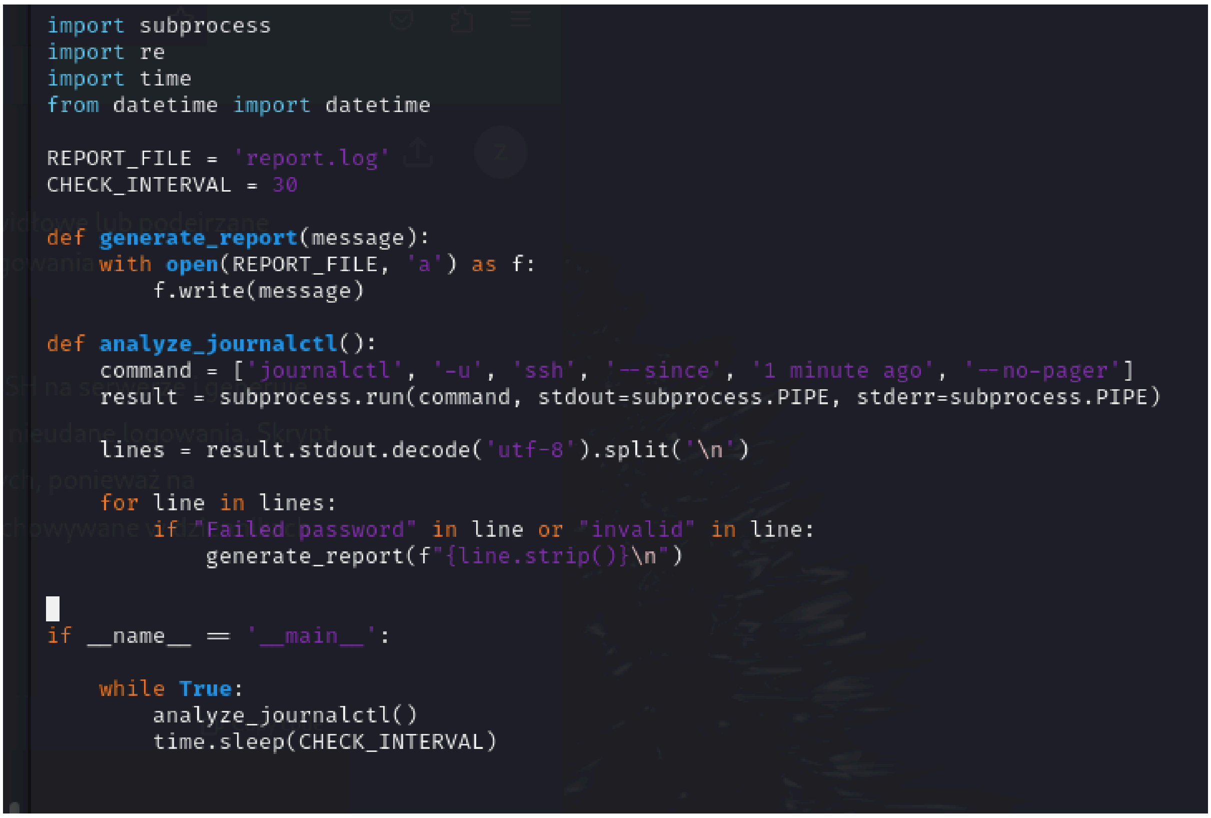
```
C:\Users\student\Documents\279471\ssh\venv\Scripts\python.exe C:\Users\student\Documents\279471\ssh\main.py
Desktop
Documents
Downloads
Music
Pictures
Public
Templates
Videos

    PID TTY          TIME CMD
   1552 ?        00:00:00 systemd
   1553 ?        00:00:00 (sd-pam)
   1570 ?        00:00:00 pipewire
   1571 ?        00:00:00 pipewire
   1572 ?        00:00:00 wireplumber
   1573 ?        00:00:00 pipewire-pulse
   1575 ?        00:00:00 dbus-daemon
   1642 ?        00:00:00 xfce4-session
   1691 ?        00:00:00 VBoxClient
```

Zrobiłem połączenie z Windowsa z normalnej maszyny na maszynę wirtualną Kali.

zad 2

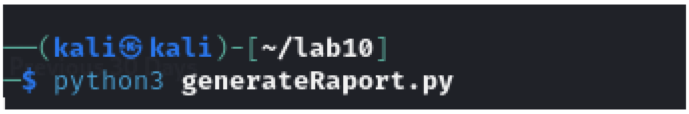
Sprawdzam logi za pomocą journalctl.

Skrypt:

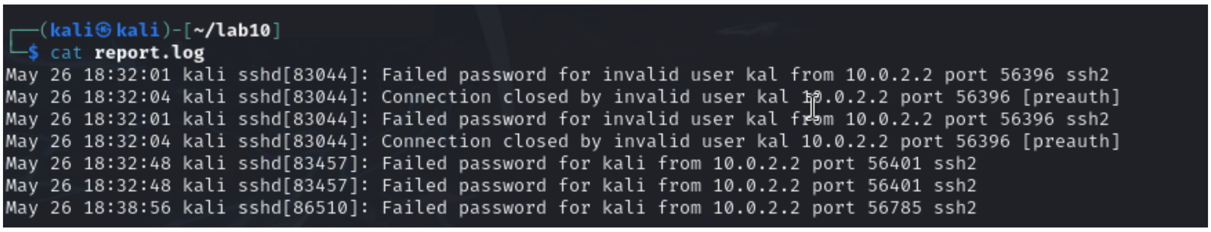


```python
import subprocess
import re
import time
from datetime import datetime

REPORT_FILE = 'report.log'
CHECK_INTERVAL = 30

def generate_report(message):
    with open(REPORT_FILE, 'a') as f:
        f.write(message)

def analyze_journalctl():
    command = ['journalctl', '-u', 'ssh', '--since', '1 minute ago', '--no-pager']
    result = subprocess.run(command, stdout=subprocess.PIPE, stderr=subprocess.PIPE)

    lines = result.stdout.decode('utf-8').split('\n')

    for line in lines:
        if "Failed password" in line or "invalid" in line:
            generate_report(f"{line.strip()}\n")


if __name__ == '__main__':

    while True:
        analyze_journalctl()
        time.sleep(CHECK_INTERVAL)
```

wykonanie:

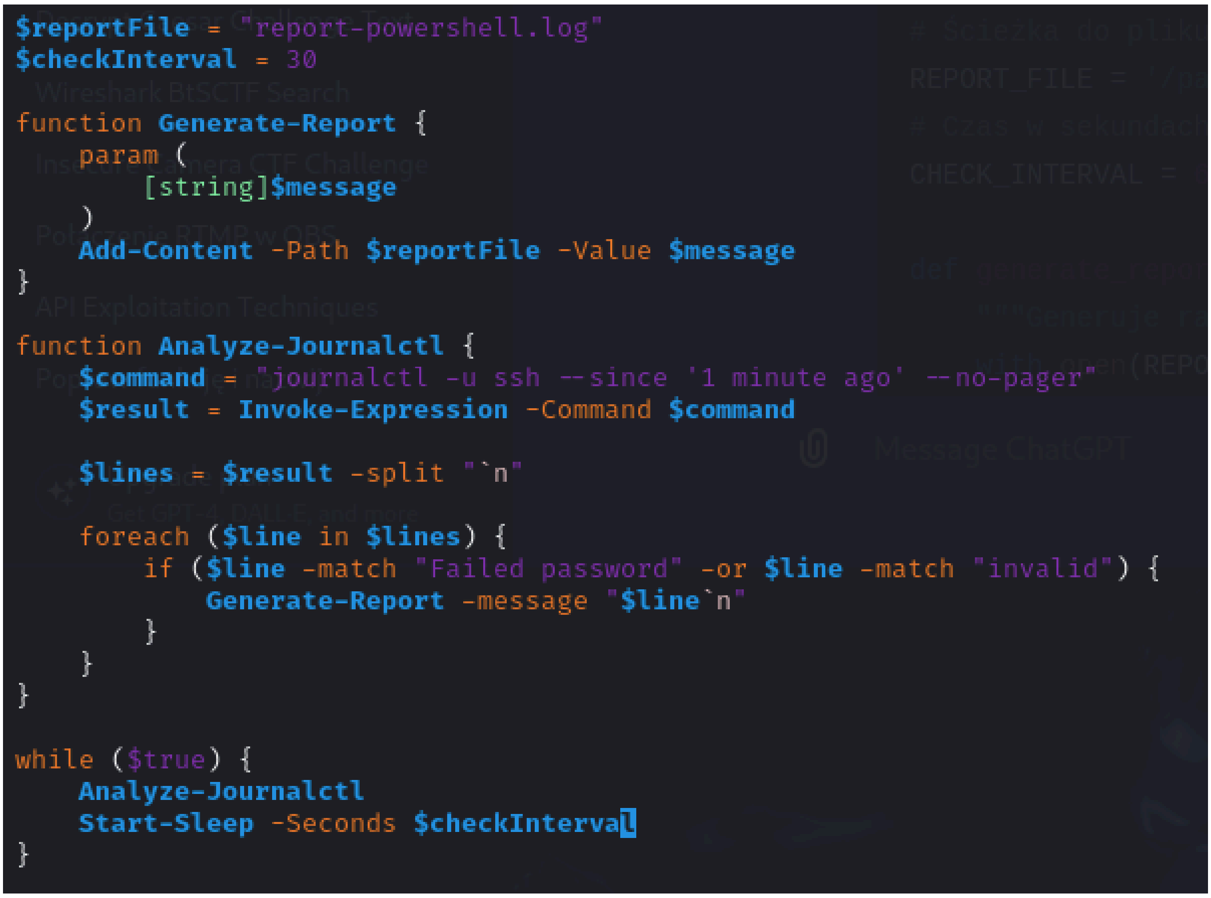


```
┌──(kali㉿kali)-[~/lab10]
└─$ python3 generateRaport.py
```

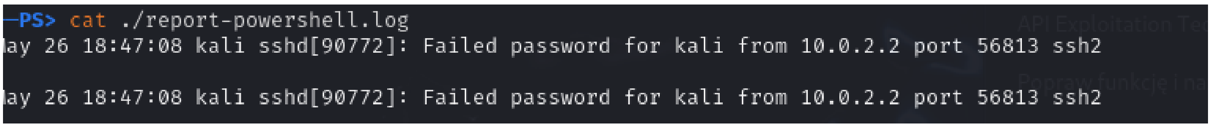


```
┌──(kali㉿kali)-[~/lab10]
└─$ cat report.log
May 26 18:32:01 kali sshd[83044]: Failed password for invalid user kal from 10.0.2.2 port 56396 ssh2
May 26 18:32:04 kali sshd[83044]: Connection closed by invalid user kal 10.0.2.2 port 56396 [preauth]
May 26 18:32:01 kali sshd[83044]: Failed password for invalid user kal from 10.0.2.2 port 56396 ssh2
May 26 18:32:04 kali sshd[83044]: Connection closed by invalid user kal 10.0.2.2 port 56396 [preauth]
May 26 18:32:48 kali sshd[83457]: Failed password for kali from 10.0.2.2 port 56401 ssh2
May 26 18:32:48 kali sshd[83457]: Failed password for kali from 10.0.2.2 port 56401 ssh2
May 26 18:38:56 kali sshd[86510]: Failed password for kali from 10.0.2.2 port 56785 ssh2
```

Skrypt wykonany w Powershellu:



```
$reportFile = "report-powershell.log"
$checkInterval = 30

function Generate-Report {
    param (
        [string]$message
    )
    Add-Content -Path $reportFile -Value $message
}

function Analyze-Journalctl {
    $command = "journalctl -u ssh --since '1 minute ago' --no-pager"
    $result = Invoke-Expression -Command $command

    $lines = $result -split "`n"

    foreach ($line in $lines) {
        if ($line -match "Failed password" -or $line -match "invalid") {
            Generate-Report -message "$line`n"
        }
    }
}

while ($true) {
    Analyze-Journalctl
    Start-Sleep -Seconds $checkInterval
}
```

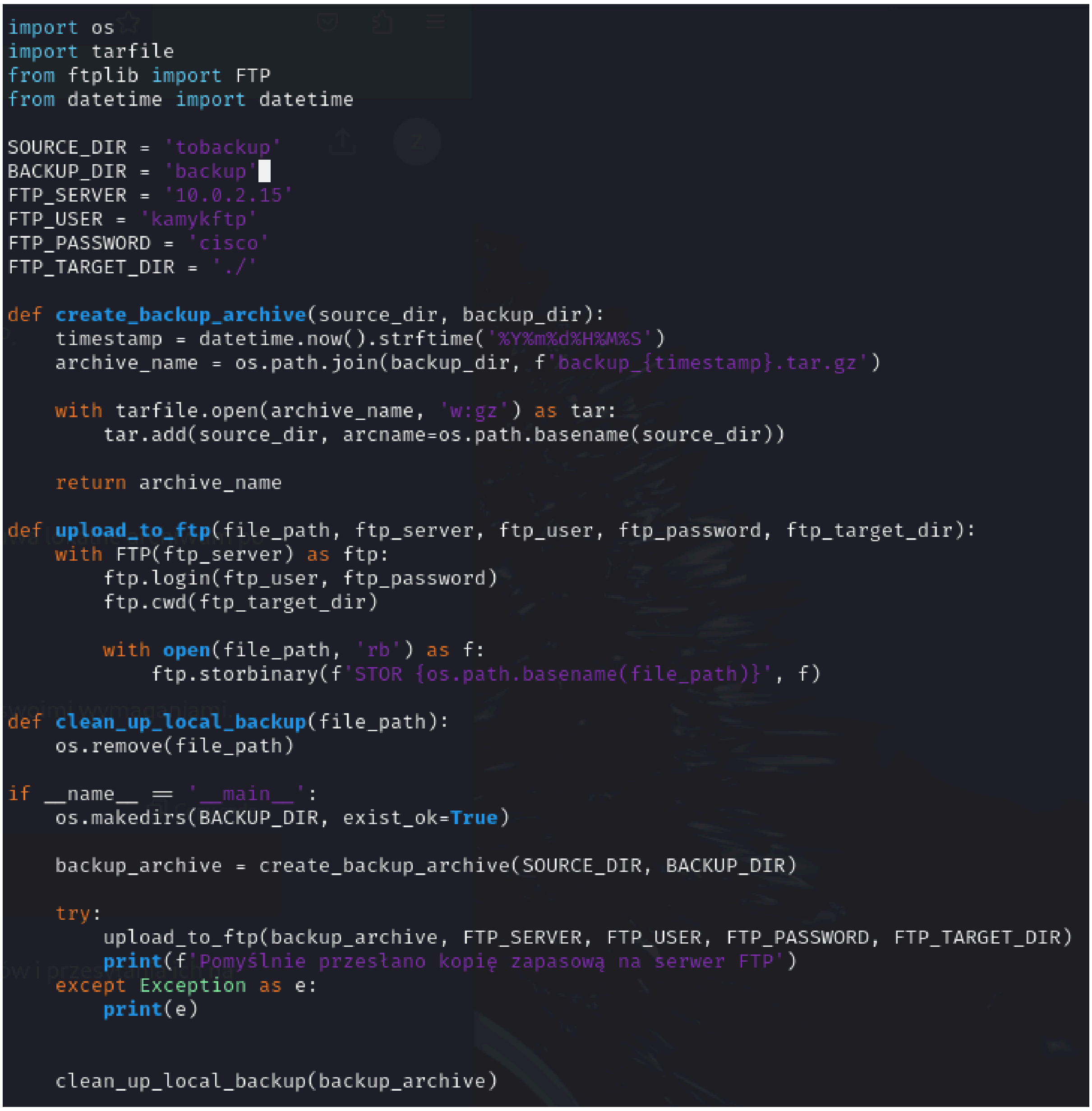


```
─PS> cat ./report-powershell.log
lay 26 18:47:08 kali sshd[90772]: Failed password for kali from 10.0.2.2 port 56813 ssh2

lay 26 18:47:08 kali sshd[90772]: Failed password for kali from 10.0.2.2 port 56813 ssh2
```

zad3

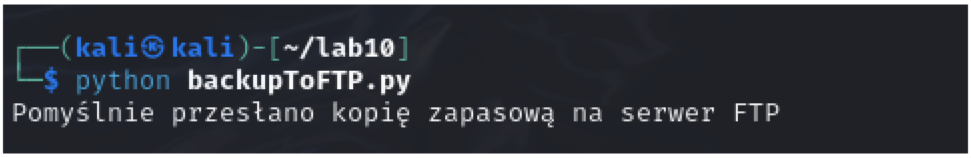
Postawiłem FTP za pomocą vsftpd. Posługiwałem się tym poradnikiem: https://www.geeksforgeeks.org/how-to-setup-and-configure-an-ftp-server-in-linux-2/

Skrypt:



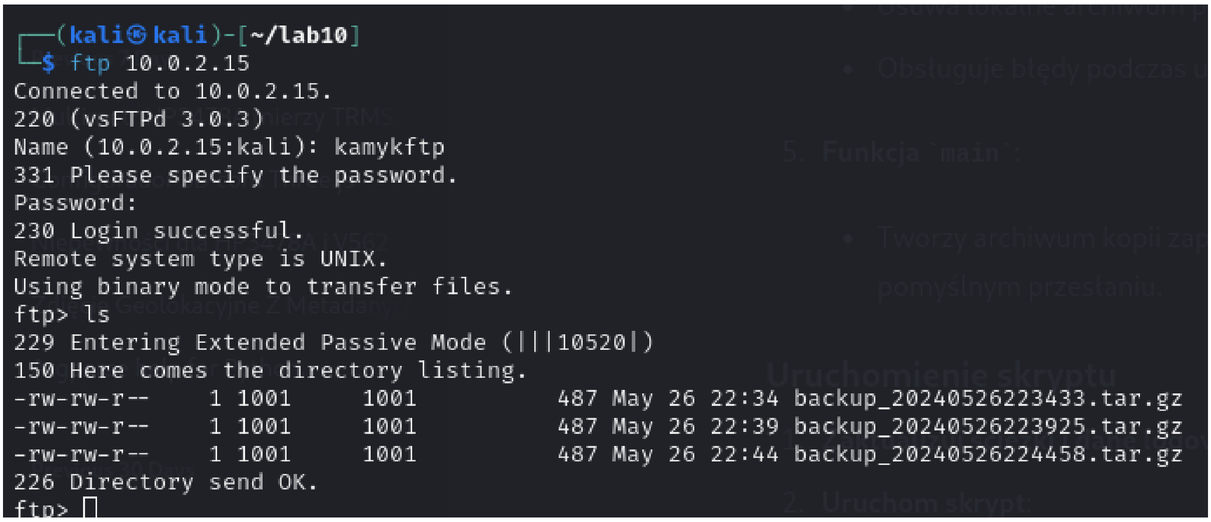
```python
import os
import tarfile
from ftplib import FTP
from datetime import datetime

SOURCE_DIR = 'tobackup'
BACKUP_DIR = 'backup'
FTP_SERVER = '10.0.2.15'
FTP_USER = 'kamykftp'
FTP_PASSWORD = 'cisco'
FTP_TARGET_DIR = './'

def create_backup_archive(source_dir, backup_dir):
    timestamp = datetime.now().strftime('%Y%m%d%H%M%S')
    archive_name = os.path.join(backup_dir, f'backup_{timestamp}.tar.gz')

    with tarfile.open(archive_name, 'w:gz') as tar:
        tar.add(source_dir, arcname=os.path.basename(source_dir))

    return archive_name

def upload_to_ftp(file_path, ftp_server, ftp_user, ftp_password, ftp_target_dir):
    with FTP(ftp_server) as ftp:
        ftp.login(ftp_user, ftp_password)
        ftp.cwd(ftp_target_dir)

        with open(file_path, 'rb') as f:
            ftp.storbinary(f'STOR {os.path.basename(file_path)}', f)

def clean_up_local_backup(file_path):
    os.remove(file_path)

if __name__ == '__main__':
    os.makedirs(BACKUP_DIR, exist_ok=True)

    backup_archive = create_backup_archive(SOURCE_DIR, BACKUP_DIR)

    try:
        upload_to_ftp(backup_archive, FTP_SERVER, FTP_USER, FTP_PASSWORD, FTP_TARGET_DIR)
        print(f'Pomyślnie przesłano kopię zapasową na serwer FTP')
    except Exception as e:
        print(e)


    clean_up_local_backup(backup_archive)
```

wykonanie:



```
┌──(kali㉿kali)-[~/lab10]
└─$ python backupToFTP.py
Pomyślnie przesłano kopię zapasową na serwer FTP
```

serwer ftp:



```
┌──(kali㉿kali)-[~/lab10]
└─$ ftp 10.0.2.15
Connected to 10.0.2.15.
220 (vsFTPd 3.0.3)
Name (10.0.2.15:kali): kamykftp
331 Please specify the password.
Password:
230 Login successful.
Remote system type is UNIX.
Using binary mode to transfer files.
ftp> ls
229 Entering Extended Passive Mode (|||10520|)
150 Here comes the directory listing.
-rw-rw-r--    1 1001     1001          487 May 26 22:34 backup_20240526223433.tar.gz
-rw-rw-r--    1 1001     1001          487 May 26 22:39 backup_20240526223925.tar.gz
-rw-rw-r--    1 1001     1001          487 May 26 22:44 backup_20240526224458.tar.gz
226 Directory send OK.
ftp> 
```