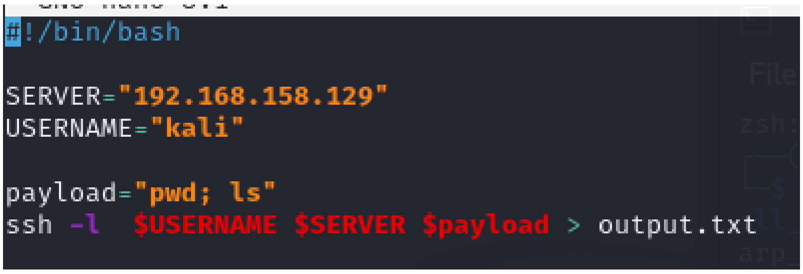
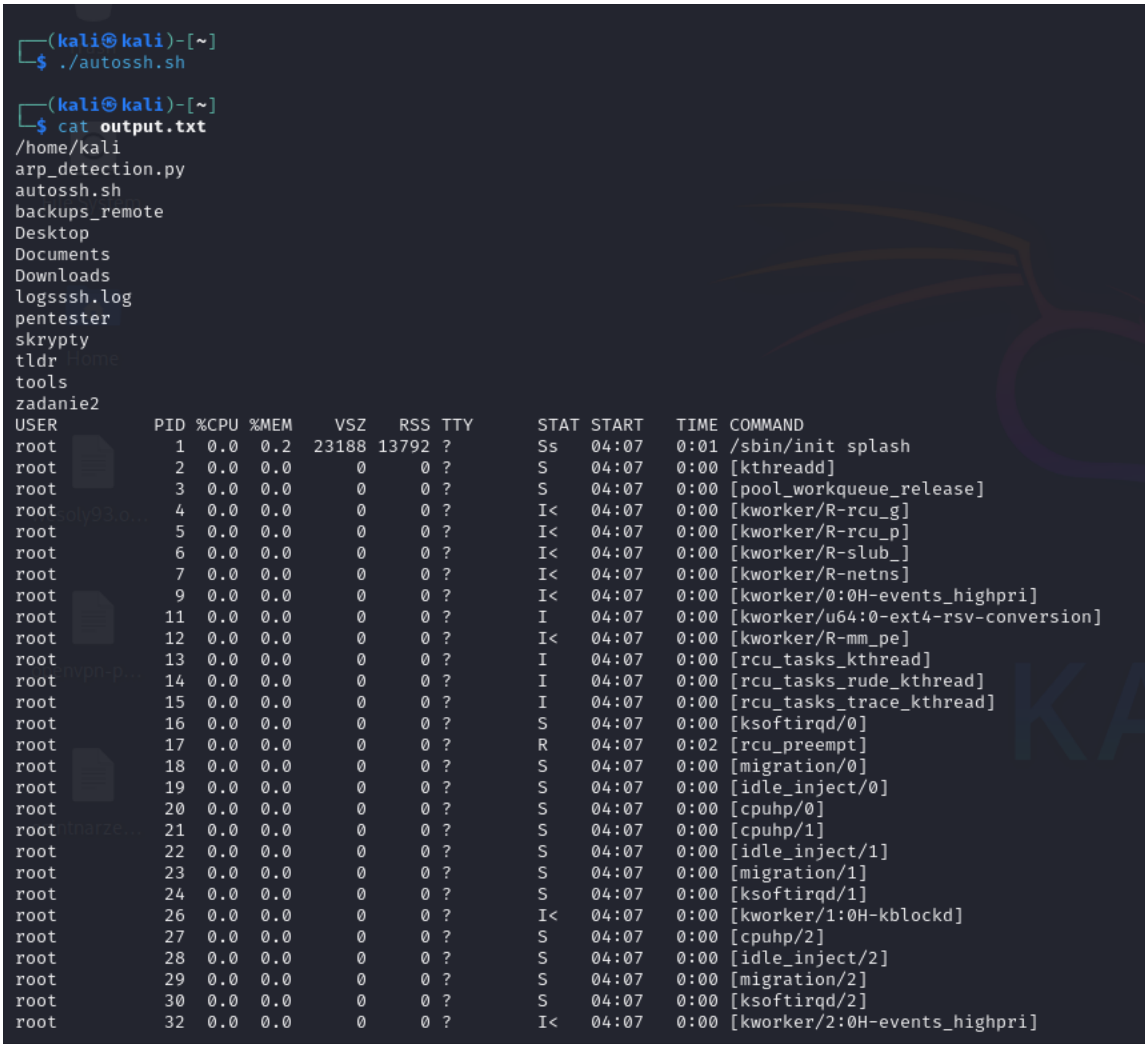
1. Napisz skrypt, który automatycznie loguje się na zdalny serwer przy użyciu SSH, wykonuje kilka poleceń (np. listowanie plików, procesów) i zapisuje wynik do pliku lokalnego

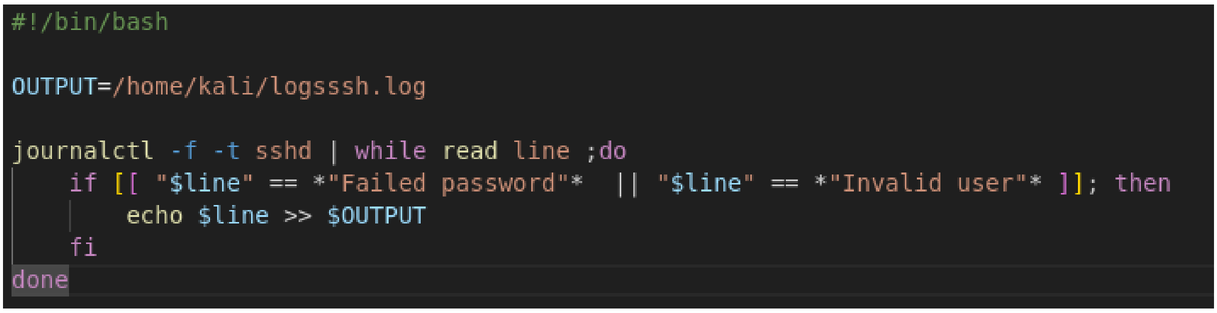


```
#!/bin/bash

SERVER="192.168.158.129"
USERNAME="kali"

payload="pwd; ls"
ssh -l  $USERNAME $SERVER $payload > output.txt
```

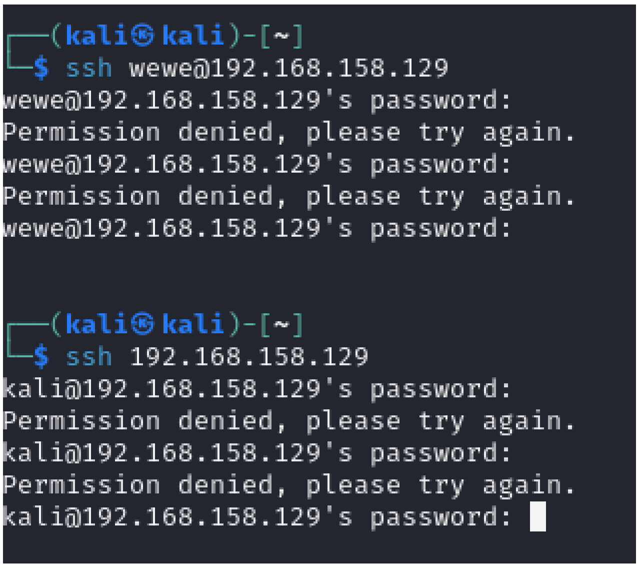


```
┌──(kali㉿kali)-[~]
└─$ ./autossh.sh

┌──(kali㉿kali)-[~]
└─$ cat output.txt
/home/kali
arp_detection.py
autossh.sh
backups_remote
Desktop
Documents
Downloads
logsssh.log
pentester
skrypty
tldr
tools
zadanie2
USER         PID %CPU %MEM    VSZ   RSS TTY      STAT START   TIME COMMAND
root           1  0.0  0.2  23188 13792 ?        Ss   04:07   0:01 /sbin/init splash
root           2  0.0  0.0      0     0 ?        S    04:07   0:00 [kthreadd]
root           3  0.0  0.0      0     0 ?        S    04:07   0:00 [pool_workqueue_release]
root           4  0.0  0.0      0     0 ?        I<   04:07   0:00 [kworker/R-rcu_g]
root           5  0.0  0.0      0     0 ?        I<   04:07   0:00 [kworker/R-rcu_p]
root           6  0.0  0.0      0     0 ?        I<   04:07   0:00 [kworker/R-slub_]
root           7  0.0  0.0      0     0 ?        I<   04:07   0:00 [kworker/R-netns]
root           9  0.0  0.0      0     0 ?        I<   04:07   0:00 [kworker/0:0H-events_highpri]
root          11  0.0  0.0      0     0 ?        I    04:07   0:00 [kworker/u64:0-ext4-rsv-conversion]
root          12  0.0  0.0      0     0 ?        I<   04:07   0:00 [kworker/R-mm_pe]
root          13  0.0  0.0      0     0 ?        I    04:07   0:00 [rcu_tasks_kthread]
root          14  0.0  0.0      0     0 ?        I    04:07   0:00 [rcu_tasks_rude_kthread]
root          15  0.0  0.0      0     0 ?        I    04:07   0:00 [rcu_tasks_trace_kthread]
root          16  0.0  0.0      0     0 ?        S    04:07   0:00 [ksoftirqd/0]
root          17  0.0  0.0      0     0 ?        R    04:07   0:02 [rcu_preempt]
root          18  0.0  0.0      0     0 ?        S    04:07   0:00 [migration/0]
root          19  0.0  0.0      0     0 ?        S    04:07   0:00 [idle_inject/0]
root          20  0.0  0.0      0     0 ?        S    04:07   0:00 [cpuhp/0]
root          21  0.0  0.0      0     0 ?        S    04:07   0:00 [cpuhp/1]
root          22  0.0  0.0      0     0 ?        S    04:07   0:00 [idle_inject/1]
root          23  0.0  0.0      0     0 ?        S    04:07   0:00 [migration/1]
root          24  0.0  0.0      0     0 ?        S    04:07   0:00 [ksoftirqd/1]
root          26  0.0  0.0      0     0 ?        I<   04:07   0:00 [kworker/1:0H-kblockd]
root          27  0.0  0.0      0     0 ?        S    04:07   0:00 [cpuhp/2]
root          28  0.0  0.0      0     0 ?        S    04:07   0:00 [idle_inject/2]
root          29  0.0  0.0      0     0 ?        S    04:07   0:00 [migration/2]
root          30  0.0  0.0      0     0 ?        S    04:07   0:00 [ksoftirqd/2]
root          32  0.0  0.0      0     0 ?        I<   04:07   0:00 [kworker/2:0H-events_highpri]
```

2. Napisz skrypt, który monitoruje logi połączeń SSH na serwerze i generuje raport, gdy wykryje nieprawidłowe lub podejrzane aktywności, na przykład nieudane logowania

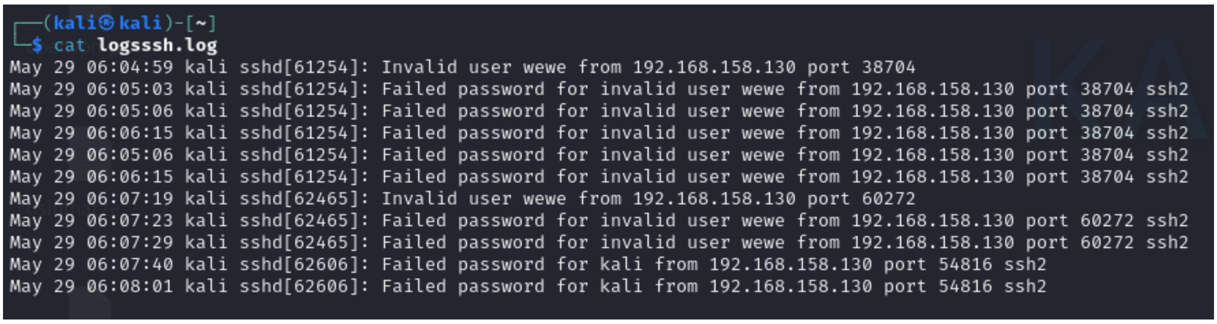


```
#!/bin/bash

OUTPUT=/home/kali/logsssh.log

journalctl -f -t sshd | while read line ;do
    if [[ "$line" == *"Failed password"*  || "$line" == *"Invalid user"* ]]; then
        echo $line >> $OUTPUT
    fi
done
```



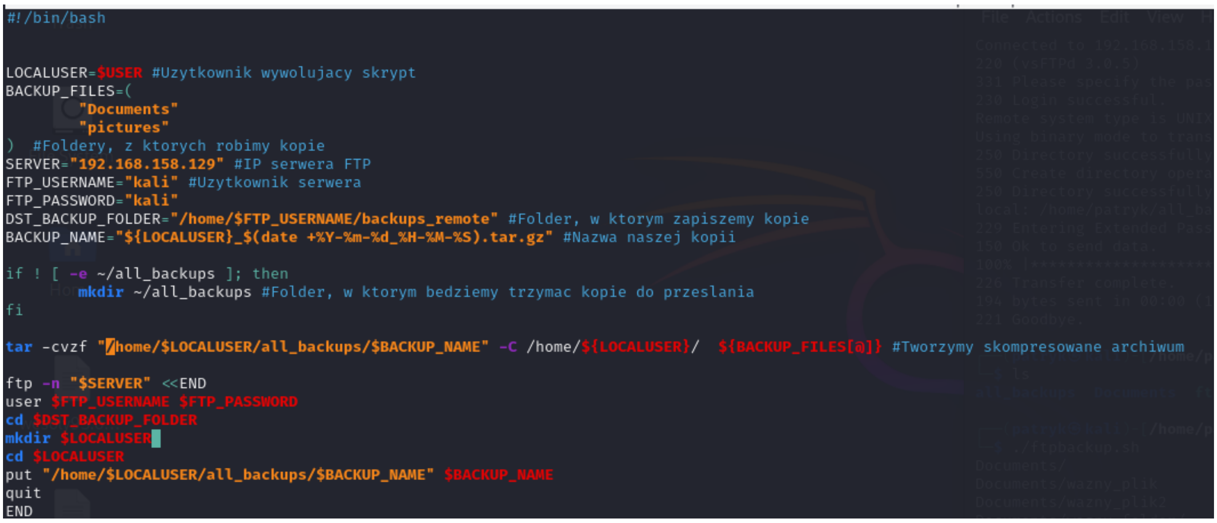
```
┌──(kali㉿kali)-[~]
└─$ ssh wewe@192.168.158.129
wewe@192.168.158.129's password:
Permission denied, please try again.
wewe@192.168.158.129's password:
Permission denied, please try again.
wewe@192.168.158.129's password:


┌──(kali㉿kali)-[~]
└─$ ssh 192.168.158.129
kali@192.168.158.129's password:
Permission denied, please try again.
kali@192.168.158.129's password:
Permission denied, please try again.
kali@192.168.158.129's password:
```

Wynik działania skryptu



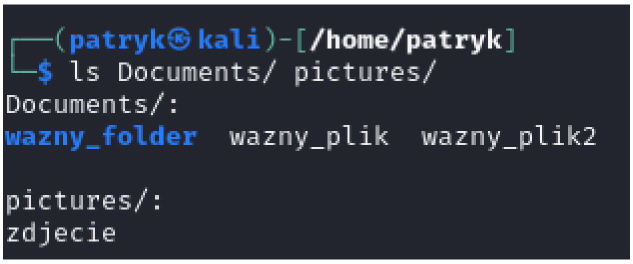
```
┌──(kali㉿kali)-[~]
└─$ cat logsssh.log
May 29 06:04:59 kali sshd[61254]: Invalid user wewe from 192.168.158.130 port 38704
May 29 06:05:03 kali sshd[61254]: Failed password for invalid user wewe from 192.168.158.130 port 38704 ssh2
May 29 06:05:06 kali sshd[61254]: Failed password for invalid user wewe from 192.168.158.130 port 38704 ssh2
May 29 06:06:15 kali sshd[61254]: Failed password for invalid user wewe from 192.168.158.130 port 38704 ssh2
May 29 06:05:06 kali sshd[61254]: Failed password for invalid user wewe from 192.168.158.130 port 38704 ssh2
May 29 06:06:15 kali sshd[61254]: Failed password for invalid user wewe from 192.168.158.130 port 38704 ssh2
May 29 06:07:19 kali sshd[62465]: Invalid user wewe from 192.168.158.130 port 60272
May 29 06:07:23 kali sshd[62465]: Failed password for invalid user wewe from 192.168.158.130 port 60272 ssh2
May 29 06:07:29 kali sshd[62465]: Failed password for invalid user wewe from 192.168.158.130 port 60272 ssh2
May 29 06:07:40 kali sshd[62606]: Failed password for kali from 192.168.158.130 port 54816 ssh2
May 29 06:08:01 kali sshd[62606]: Failed password for kali from 192.168.158.130 port 54816 ssh2
```

3. Napisz skrypt, który wykonuje kopię zapasową określonych plików na lokalnym komputerze i przesyła je na zdalny serwer FTP w określonym katalogu. Dodaj funkcję archiwizacji i kompresji przed wysłaniem plików.

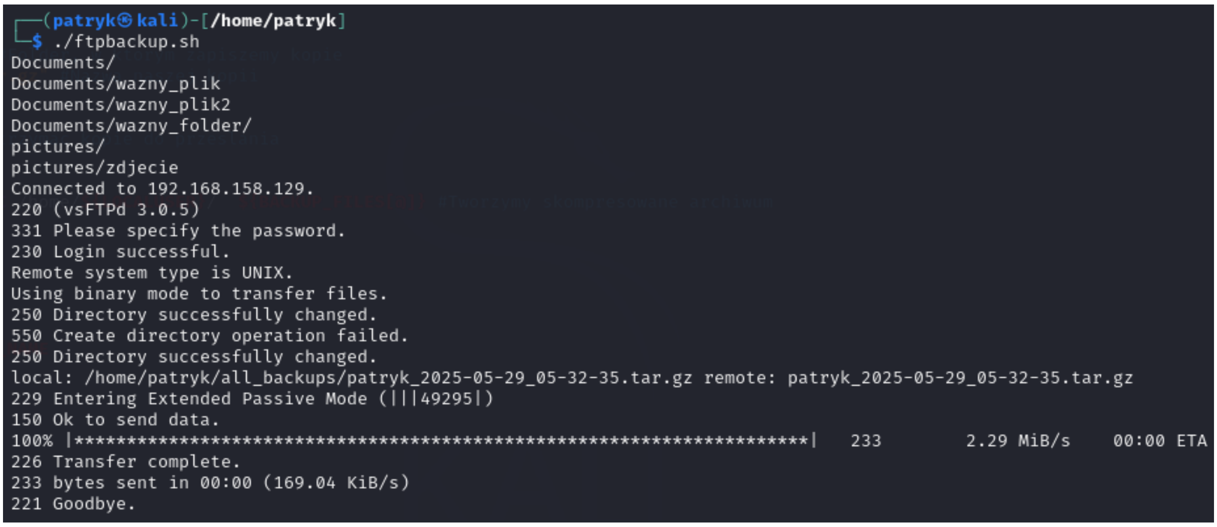


```
#!/bin/bash


LOCALUSER=$USER #Uzytkownik wywolujacy skrypt
BACKUP_FILES=(
        "Documents"
        "pictures"
)  #Foldery, z ktorych robimy kopie
SERVER="192.168.158.129" #IP serwera FTP
FTP_USERNAME="kali" #Uzytkownik serwera
FTP_PASSWORD="kali"
DST_BACKUP_FOLDER="/home/$FTP_USERNAME/backups_remote" #Folder, w ktorym zapiszemy kopie
BACKUP_NAME="${LOCALUSER}_$(date +%Y-%m-%d_%H-%M-%S).tar.gz" #Nazwa naszej kopii

if ! [ -e ~/all_backups ]; then
        mkdir ~/all_backups #Folder, w ktorym bedziemy trzymac kopie do przeslania
fi

tar -cvzf "/home/$LOCALUSER/all_backups/$BACKUP_NAME" -C /home/${LOCALUSER}/  ${BACKUP_FILES[@]} #Tworzymy skompresowane archiwum

ftp -n "$SERVER" <<END
user $FTP_USERNAME $FTP_PASSWORD
cd $DST_BACKUP_FOLDER
mkdir $LOCALUSER
cd $LOCALUSER
put "/home/$LOCALUSER/all_backups/$BACKUP_NAME" $BACKUP_NAME
quit
END
```

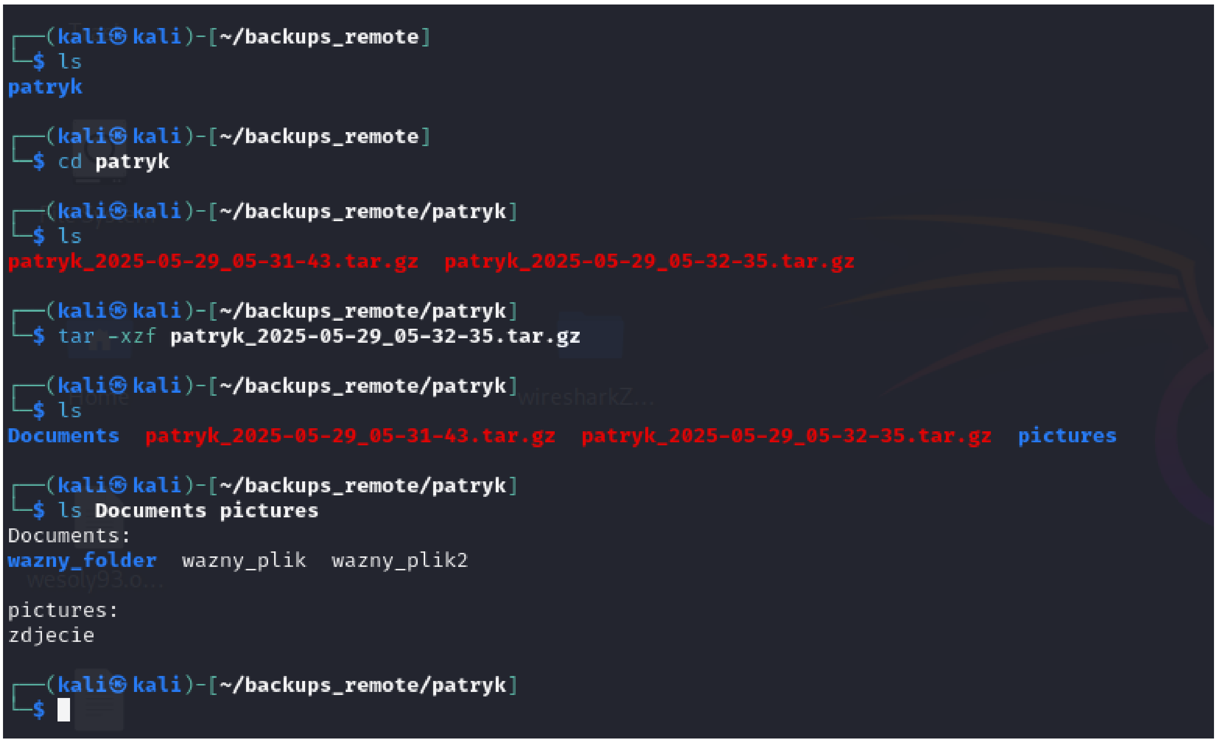
Struktura katalogów, które przesyłamy



```
┌──(patryk㉿kali)-[/home/patryk]
└─$ ls Documents/ pictures/
Documents/:
wazny_folder  wazny_plik  wazny_plik2

pictures/:
zdjecie
```



```
┌──(patryk㉿kali)-[/home/patryk]
└─$ ./ftpbackup.sh
Documents/
Documents/wazny_plik
Documents/wazny_plik2
Documents/wazny_folder/
pictures/
pictures/zdjecie
Connected to 192.168.158.129.
220 (vsFTPd 3.0.5)
331 Please specify the password.
230 Login successful.
Remote system type is UNIX.
Using binary mode to transfer files.
250 Directory successfully changed.
550 Create directory operation failed.
250 Directory successfully changed.
local: /home/patryk/all_backups/patryk_2025-05-29_05-32-35.tar.gz remote: patryk_2025-05-29_05-32-35.tar.gz
229 Entering Extended Passive Mode (|||49295|)
150 Ok to send data.
100% |***********************************************************************|   233        2.29 MiB/s    00:00 ETA
226 Transfer complete.
233 bytes sent in 00:00 (169.04 KiB/s)
221 Goodbye.
```



```
┌──(kali㉿kali)-[~/backups_remote]
└─$ ls
patryk

┌──(kali㉿kali)-[~/backups_remote]
└─$ cd patryk

┌──(kali㉿kali)-[~/backups_remote/patryk]
└─$ ls
patryk_2025-05-29_05-31-43.tar.gz  patryk_2025-05-29_05-32-35.tar.gz

┌──(kali㉿kali)-[~/backups_remote/patryk]
└─$ tar -xzf patryk_2025-05-29_05-32-35.tar.gz

┌──(kali㉿kali)-[~/backups_remote/patryk]
└─$ ls
Documents  patryk_2025-05-29_05-31-43.tar.gz  patryk_2025-05-29_05-32-35.tar.gz  pictures

┌──(kali㉿kali)-[~/backups_remote/patryk]
└─$ ls Documents pictures
Documents:
wazny_folder  wazny_plik  wazny_plik2

pictures:
zdjecie

┌──(kali㉿kali)-[~/backups_remote/patryk]
└─$ 
```