

# 1. Napisz skrypt w bash, który pomoże administratorowi i przyspieszy tworzenie nowego komputera w organizacji. Podziel zadania na odpowiednie funkcje:

a) Sprawdzenie aktualizacji systemu i ich instalacja

```
#Aktualizacja repozytorium i instalacja aktualizacji
function updates(){
    apt -y update
    apt -y upgrade
}
```

Rysunek 1.Kod skryptu

```
└─$ sudo ./skrypt.sh
[sudo] password for kali:
Get:1 https://packages.microsoft.com/repos/code stable InRelease [3,590 B]
Get:2 http://kali.download/kali kali-rolling InRelease [41.5 kB]
Get:3 https://packages.microsoft.com/repos/code stable/main amd64 Packages [17.7 kB]
Get:4 https://packages.microsoft.com/repos/code stable/main armhf Packages [17.9 kB]
Get:5 https://packages.microsoft.com/repos/code stable/main arm64 Packages [17.9 kB]
Get:6 http://kali.download/kali kali-rolling/main amd64 Packages [20.7 MB]
Get:7 http://kali.download/kali kali-rolling/main amd64 Contents (deb) [49.9 MB]
Fetched 70.8 MB in 8s (9,182 kB/s)
2072 packages can be upgraded. Run 'apt list --upgradable' to see them.
The following packages were automatically installed and are no longer required:
firebird3.0-common libgail18t64 libjsoncpp25 perl-modules-5.38
firebird3.0-common-doc libgeos3.12.2 libmbcncrypto7t64 pipewire-module-xrdp
fonts-liberation2 libgfpapi0 libbmx1 python3-appdirs
freerdp2-x11 libgfrpc0 libmsgraph-0-1 python3-hatch-vcs
golang-1.23-go libgfdx0 libpaper1 python3-hatchling
golang-1.23-src libgl1-mesa-dev libperl5.38t64 python3-lib2to3
hydra-gtk libglapi-mesa libpipewire-0.3-modules-xrdp python3-pathspect
ibverbs-providers libgles-dev libplacebo338 python3-pluggy
libassuan0 libgles1 libplist3 python3-setproctitle
libavfilter9 libglusterfs0 libpostproc57 python3-setuptools-scm
libbfi0 libglvnd-core-dev libpython3.11-dev python3-trove-classifiers
libboost-iostreams1.83.0 libglvnd-dev librados2 python3.11
libboost-thread1.83.0 libgspell-1-2 librdmacm1t64 python3.11-dev
libcapstone4 libgtk2.0-0t64 libtag1v5 python3.11-minimal
libcephfs2 libgtk2.0-bin libtag1v5-vanilla ruby-zeitwerk
libconfig++9v5 libgtk2.0-common libtagc0 ruby3.1
libconfig9 libgtksourceview-3.0-1 libusbmuxd6 ruby3.1-dev
libdirectfb-1.7-7t64 libgtksourceview-3.0-common libwebrtc-audio-processing1 ruby3.1-doc
libegl-dev libgtksourceviewmm-3.0-0v5 libwinpr2-2t64 rwho
libflac12t64 libgumbo2 libzip4t64 rwhod
libfmt9 libibverbs1 openjdk-17-jre samba-vfs-modules
libfreerdp-client2-2t64 libimobiledevice6 openjdk-17-jre-headless strongswan
libfreerdp2-2t64 libiniparser1 openjdk-23-jre
libgail-common libjim0.82t64 openjdk-23-jre-headless

Use 'sudo apt autoremove' to remove them.

Upgrading:
7zip libphoto2-l10n login.defs
accountsservice libphoto2-port12t64 logsave
```

Rysunek 2.Wynik skryptu

b) Instalacja klienta pocztowego np. Thunderbird

```
#Instalacja programu thunderbird
function thunderbird_install(){
    apt -y install thunderbird
}
```

Rysunek 3.Kod skryptu

```
(kali@kali) [~/Documents/Programowanie_skryptowe/tab4]
$ sudo ./skrypt.sh
The following packages were automatically installed and are no longer required:
  ibverbs-providers libgfapi0 libibverbs1 librdmacm1t64 python3.11-dev
  libboost-iostreams1.83.0 libgfrpc0 libpipewire-0.3-modules-xrdp pipewire-module-xrdp python3.11-minimal
  libboost-thread1.83.0 libgfrpc0 libpython3.11-dev python3-lib2to3 samba-vfs-modules
  libcephfs2 libglusterfs0 librados2 python3.11
Use 'sudo apt autoremove' to remove them.

Installing:
  thunderbird

Installing dependencies:
  libbotan-2-19 libotr5t64 librnnp0 libsexpp0 libtspi1

Suggested packages:
  libotr5-bin

Summary:
  Upgrading: 0, Installing: 6, Removing: 0, Not Upgrading: 2073
  Download size: 65.3 MB
  Space needed: 260 MB / 54.2 GB available

Get:1 http://kali.download/kali kali-rolling/main amd64 libtspi1 amd64 0.3.15-0.4 [165 kB]
Get:2 http://http.kali.org/kali kali-rolling/main amd64 libbotan-2-19 amd64 2.19.5+dfsg-3 [1,739 kB]
Get:3 http://kali.download/kali kali-rolling/main amd64 libotr5t64 amd64 4.1.1-6 [84.5 kB]
Get:4 http://http.kali.org/kali kali-rolling/main amd64 libsexpp0 amd64 0.8.7-4+b1 [24.5 kB]
Get:5 http://http.kali.org/kali kali-rolling/main amd64 librnnp0 amd64 0.17.1-1+b1 [336 kB]
Get:6 http://kali.download/kali kali-rolling/main amd64 thunderbird amd64 1:128.8.0esr-1 [62.9 MB]
20% [6 thunderbird 8,270 B/62.9 MB 0%]^C
```

Rysunek 4.Wynik skryptu

### c) Dodanie nowego użytkownika

```
#Tworzenie uzytkownika wraz z katalogiem domowym i podkatalogami
function add_user(){
    useradd -m $1 -g kali -s /bin/bash
    passwd -d $1
}
```

Rysunek 5.Kod skryptu

```
(kali@kali)-[~/Documents/Programowanie_skryptowe/lab4]
$ sudo ./skrypt.sh
Podaj nazwe uzytkownika: Patryk
passwd: password changed.
/home/kali/Documents/Programowanie_skryptowe/lab4

(kali@kali)-[~/Documents/Programowanie_skryptowe/lab4]
$ su Patryk
(Patryk@kali)-[/home/kali/Documents/Programowanie_skryptowe/lab4]
$ cd ~

(Patryk@kali)-[~]
$ pwd
/home/Patryk

(Patryk@kali)-[~]
$
```

Rysunek 6. Wynik skryptu

- d) Utworzenie katalogów dla nowego pracownika (Documents, Pictures, Videos)

```
#Tworzenie katalogow w katalogu domowym
function create_dirs(){
    cd /home/$1
    mkdir Documents Pictures Videos
    cd -
}

create_dirs $username
```

Rysunek 7. Kod skryptu

```
(Patryk@kali)-[~]
$ pwd
/home/Patryk

(Patryk@kali)-[~]
$ ls
Documents Pictures Videos

(Patryk@kali)-[~]
$
```

Rysunek 8. Wynik skryptu

- e) Wyświetlenie informacji o wersji systemu, adresu ip, adresu mac

```
#Wypisywanie informacji o systemie
function info(){
    #Pobranie informacji o używanym interfejsie i ip
    ipmac=$( ip route get 1.1.1.1 | awk '{print $(NF-2),$(NF-4)}' 2>/dev/null)
    echo "Wersja: $( uname -r )"
    echo "IP: " "${ipmac[0]}"
    echo "MAC: " "${ifconfig "${ipmac[1]}" | awk '/ether/ {print $2}'}"
}
```

Rysunek 9. Kod skryptu

```
(kali㉿kali)-[~/Documents/Programowanie_skryptowe/lab4]
$ ./skrypt.sh
Wersja: 6.8.11-amd64
IP: 192.168.131.128
MAC: 00:0c:29:fd:63:4a
```

Rysunek 10. Wynik skryptu

2. Załóż konto na VirusTotal i napisz automatyzację w bashu sprawdzającą reputację plików, która:

```

#input od uzytkownika
read -p "Podaj nazwe/sciezke do pliku: " file

#Obliczanie sumy kontrolnej
sum=$(sha256sum $file )
echo "Suma kontrolna(sha256) $sum"

#Wyslanie pliku na serwer do analizy
UPLOAD=$(curl --request POST \
  --url https://www.virustotal.com/api/v3/files \
  --header 'accept: application/json' \
  --header 'content-type: multipart/form-data' \
  --header 'x-apikey: f5269e38dd5a80f21f792c7ff31afd13fcc58aceba453f260abe388bd380850f' \
  --form file=@"$file")

#Pobranie id naszego pliku
ID=$(echo "$UPLOAD" | jq -r '.data.id')

#Oczekiwanie az skan sie wykona
echo "Oczekiwanie na wynik analizy..."

#Pobranie informacji o przeprowadzanej analizie
ANALIZA=$(curl --request GET \
  --url https://www.virustotal.com/api/v3/analyses/$ID \
  --header 'accept: application/json' \
  --header 'x-apikey: f5269e38dd5a80f21f792c7ff31afd13fcc58aceba453f260abe388bd380850f')
STATUS=$(echo $ANALIZA | jq -r '.data.attributes.status')

#Oczekiwanie, az analiza dobiegnie konca
while [ "$STATUS" = "queued" ]; do
  ANALIZA=$(curl -s --request GET \
    --url https://www.virustotal.com/api/v3/analyses/$ID \
    --header 'accept: application/json' \
    --header 'x-apikey: f5269e38dd5a80f21f792c7ff31afd13fcc58aceba453f260abe388bd380850f')
  STATUS=$(echo $ANALIZA | jq -r '.data.attributes.status')
  sleep 30
done

#Wydobycie informacji o ilosci wykrytych zagrozen przez rozne AV
WYKRYTE_ZAGROZENIA=$(echo "$ANALIZA" | jq '.data.attributes.stats.malicious')

#Wypisanie czy plik jest zlosliwy/bezpieczny
if [ "$WYKRYTE_ZAGROZENIA" -gt 0 ]; then
  echo "Plik jest zlosliwy! Wykryte zagrozenia: $WYKRYTE_ZAGROZENIA"
else
  echo "Plik jest bezpieczny."
fi

```

Rysunek 11.Kod skryptu

Wynik dla pliku EICAR:

```
(kali㉿kali)-[~/Documents/Programowanie_skryptowe/lab4]
$ ./zapytania.sh
Podaj nazwe/sciezke do pliku: eicar.xls
Suma kontrolna(sha256) a5199c09459358a485bddb85488a65b74f09fbc409d8affe022ae45470898fde eicar.xls
% Total    % Received % Xferd  Average Speed   Time    Time     Time  Current
           Dload  Upload  Total   Spent    Left     Speed
100 26039 100    226 100 25813    389   44459 --:--:-- --:--:-- --:--:-- 44817
Oczekiwanie na wynik analizy...
% Total    % Received % Xferd  Average Speed   Time    Time     Time  Current
           Dload  Upload  Total   Spent    Left     Speed
100   777 100    777    0    0    2233     0 --:--:-- --:--:-- --:--:-- 2239
Plik jest zlosliwy! Wykryte zagrozenia: 35
```

Rysunek 12. Wynik skryptu

Wynik dla zwykłego pliku:

```
(kali㉿kali)-[~/Documents/Programowanie_skryptowe/lab4]
$ touch tekstowy.txt

(kali㉿kali)-[~/Documents/Programowanie_skryptowe/lab4]
$ ./zapytania.sh
Podaj nazwe/sciezke do pliku: tekstowy.txt
Suma kontrolna(sha256) e3b0c44298fc1c149afb4c8996fb92427ae41e4649b934ca495991b7852b855 tekstowy.txt
% Total    % Received % Xferd  Average Speed   Time    Time     Time  Current
           Dload  Upload  Total   Spent    Left     Speed
100   428 100    226 100   202    272    243 --:--:-- --:--:-- --:--:-- 515
Oczekiwanie na wynik analizy...
% Total    % Received % Xferd  Average Speed   Time    Time     Time  Current
           Dload  Upload  Total   Spent    Left     Speed
100   773 100    773    0    0    2199     0 --:--:-- --:--:-- --:--:-- 2196
% Total    % Received % Xferd  Average Speed   Time    Time     Time  Current
           Dload  Upload  Total   Spent    Left     Speed
100   773 100    773    0    0    1677     0 --:--:~ --:~:~ --:~:~ 1680
% Total    % Received % Xferd  Average Speed   Time    Time     Time  Current
           Dload  Upload  Total   Spent    Left     Speed
100   773 100    773    0    0    2378     0 --:~:~ --:~:~ --:~:~ 2385
% Total    % Received % Xferd  Average Speed   Time    Time     Time  Current
           Dload  Upload  Total   Spent    Left     Speed
100   773 100    773    0    0    2080     0 --:~:~ --:~:~ --:~:~ 2077
% Total    % Received % Xferd  Average Speed   Time    Time     Time  Current
           Dload  Upload  Total   Spent    Left     Speed
100   773 100    773    0    0    2353     0 --:~:~ --:~:~ --:~:~ 2356
% Total    % Received % Xferd  Average Speed   Time    Time     Time  Current
           Dload  Upload  Total   Spent    Left     Speed
100   773 100    773    0    0    2113     0 --:~:~ --:~:~ --:~:~ 2112
% Total    % Received % Xferd  Average Speed   Time    Time     Time  Current
           Dload  Upload  Total   Spent    Left     Speed
100 13623 100 13623    0    0   42639     0 --:~:~ --:~:~ --:~:~ 42705
Plik jest bezpieczny.
```

Rysunek 13. Wynik skryptu

Zadanie dodatkowe

```
#!/bin/bash

#Pobieramy dane za pomoca api o pogodzie w danym miescie
DANE=$(curl -s -X GET "http://api.weatherapi.com/v1/forecast.json?key=79e2781f1c6848d0919154502252703&q=$1&days=5&aqi=no&alerts=no")

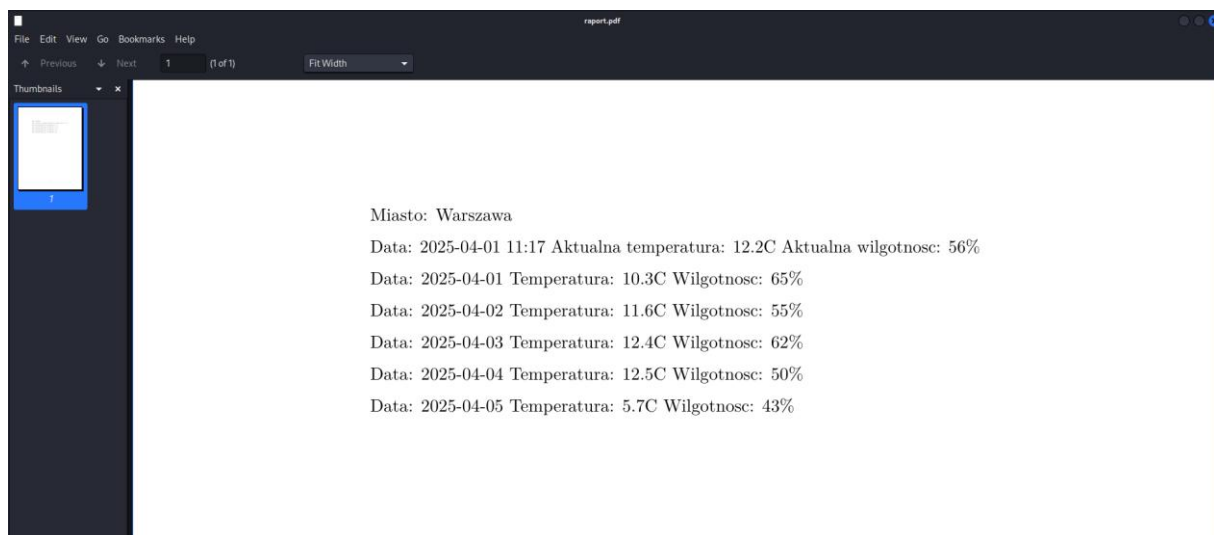
#za pomoca narzedzia jq pobieramy wartosci pliku json:

#Pobieramy dzisiejsza date
DATA=$(echo $DANE | jq -r '.location.localtime')
#Pobieramy Aktualna temperatura
TEMP=$(echo $DANE | jq -r '.current.temp_c')
#Pobieramy Aktualna wilgotnosc
HUMIDITY=$(echo $DANE | jq -r '.current.humidity')
#Przekierowujemy dane do pliku raport.txt | -e umozliwia uzycie znakow specjalnych
echo -e "Miasto: $1\n\n Data: $DATA\n\t Aktualna temperatura: ${TEMP}C\n\t Aktualna wilgotnosc: ${HUMIDITY}%\n" > raport.txt

#Za pomoca petli przechodzimy przez kolejne dni i pobieramy informacje
for i in {0..4}; do
    #Data
    DATA=$(echo "$DANE" | jq -r ".forecast.forecastday[$i].date")
    #Srednia temperatura w ciagu dnia
    TEMP=$(echo "$DANE" | jq -r ".forecast.forecastday[$i].day.avgtemp_c")
    #Srednia wilgotnosc w ciagu dnia
    HUMIDITY=$(echo "$DANE" | jq -r ".forecast.forecastday[$i].day.avghumidity")
    #Przekierowujemy dane do pliku
    echo -e "Data: $DATA\n\t Temperatura: ${TEMP}C\n\t Wilgotnosc: ${HUMIDITY}%\n" >> raport.txt
done

#Za pomoca narzedzia pandoc tworzymy raport
pandoc raport.txt -o raport.pdf
rm raport.txt
```

Rysunek 14.Kod skryptu



Rysunek 15.Wynik skryptu