

Część Praktyczna

- 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

```
update_system() {  
    echo "Sprawdzanie i instalacja aktualizacji"  
    sudo apt update && sudo apt upgrade -y  
    sudo apt autoremove -y  
}
```

b) Instalacja klienta pocztowego np. Thunderbird

```
install_thunderbird() {  
    echo "Instalacja Thunderbird"  
    sudo apt install -y thunderbird  
}
```

c) Dodanie nowego użytkownika

```
add_new_user() {  
    username=$1  
    sudo useradd $username  
    echo "Nowy uzytkownik $username dodany"  
}
```

```
Podaj nazwe uzytkownika: Dawid  
[sudo] password for kali:  
Nowy uzytkownik Dawid dodany
```

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

KONTAKT

Adrian Florek

adrian.florek@pwr.edu.pl

PROGRAMOWANIE SKRYPTOWE

```
add_new_directions() {  
    username=$1  
    userhome=/home/$username  
    sudo mkdir $userhome  
    sudo mkdir /home/$username/Pictures  
    sudo mkdir /home/$username/Documents  
    sudo mkdir /home/$username/Videos  
    echo "Katalogi utworzone"
```

```
Podaj nazwe uzytkownika: Dawid  
Katalogi utworzone
```

```
(kali㉿kali)-[/home/Dawid]  
$ ls  
Documents Pictures Videos
```

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

```
infos() {  
    echo "Wersj linuxa na $komputer to $(uname -r)"  
    echo "Adres IP: "  
    hostname -I | awk '{print $1}'  
    echo "Adres Mac: "  
    ip link show | awk '/ether/ {print $2}'  
}
```

```
Wersj linuxa na to 6.11.2-amd64  
Adres IP:  
10.0.2.15  
Adres Mac:  
08:00:27:6e:13:6e
```

W funkcji głównej skrypt powinien zapytać o nazwę użytkownika i przekazać tą informację do właściwej funkcji.

Każdy etap skryptu powinien być opatrzony komentarzem.

KONTAKT

Adrian Florek

adrian.florek@pwr.edu.pl

PROGRAMOWANIE SKRYPTOWE

```
while true; do
    echo "Menu konfiguracji nowego komputera"
    echo "1) Aktualizacja systemu"
    echo "2) Instalacja Thunderbird"
    echo "3) Dodanie nowego użytkownika"
    echo "4) Utworzenie katalogów użytkownika"
    echo "5) Wyświetlenie informacji o systemie"
    echo "6) Wyjście"
    read -p "Wybierz opcję: " wybor

    case $wybor in
        1) update_system ;;
        2) install_thunderbird ;;
        3) read -p "Podaj nazwe uzytkownika: " username
           add_new_user $username;;
        4) read -p "Podaj nazwe uzytkownika: " username
           add_new_directions $username;;
        5) infos ;;
        6) echo "Koniec."; exit 0 ;;
        *) echo "Niepoprawna opcja." ;;
    esac
done
```

```
Menu konfiguracji nowego komputera
1) Aktualizacja systemu
2) Instalacja Thunderbird
3) Dodanie nowego użytkownika
4) Utworzenie katalogów użytkownika
5) Wyświetlenie informacji o systemie
6) Wyjście
Wybierz opcję: █
```

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

- a) Obliczy sumę kontrolną wskazanego pliku (MD5 lub SHA256).

KONTAKT

Adrian Florek

adrian.florek@pwr.edu.pl

PROGRAMOWANIE SKRYPTOWE

```
# Funkcja do obliczania sumy kontrolnej pliku (MD5 lub SHA256)
calculate_checksum() {
    file="$1"
    algorithm="$2" # Możesz podać "md5" lub "sha256"

    if [[ "$algorithm" == "md5" ]]; then
        checksum=$(md5sum "$file" | awk '{print $1}')
    elif [[ "$algorithm" == "sha256" ]]; then
        checksum=$(sha256sum "$file" | awk '{print $1}')
    else
        echo "Nieobsługiwany algorytm sumy kontrolnej"
        exit 1
    fi

    echo "$checksum"
}
```

Podaj ścieżkę pliku, który chcesz zeskanować: /home/kali/Downloads/NewEmptyFile.txt
Podaj algorytm sumy kontrolnej (md5 lub sha256): md5
Suma kontrolna wynosi: 80d4df5d89bad6454ea1491edaa80b8b

b) Wyśle zapytanie do API VirusTotal za pomocą polecenia curl.

```
# Funkcja do wysyłania zapytania do API VirusTotal
file_upload() {
    checksum=$1
    response=$(curl --silent --request GET \
        --url "https://www.virustotal.com/api/v3/files/$checksum" \
        --header 'accept: application/json' \
        --header 'x-apikey: ea92c7d9db127314acfb1a4865104c8535491cabab8c7ec5155ceb5dc79b4d5')
    echo "$response"
}
```

c) Zinterpretuje odpowiedź API i wyświetli informację, czy plik jest bezpieczny, czy nie.

KONTAKT

Adrian Florek

adrian.florek@pwr.edu.pl

PROGRAMOWANIE SKRYPTOWE

```
# Główna funkcja skryptu
read -p "Podaj ścieżkę pliku, który chcesz zeskanować: " path
read -p "Podaj algorytm sumy kontrolnej (md5 lub sha256): " algorithm

# Sprawdzanie, czy plik istnieje
if [[ ! -f "$path" ]]; then
    echo "Plik nie istnieje!"
    exit 1
fi

# Obliczanie sumy kontrolnej
checksum=$(calculate_checksum "$path" "$algorithm")
echo "Suma kontrolna wynosi: $checksum"
# Wysyłanie zapytania do API VirusTotal
response=$(file_upload "$checksum")

# Liczba bezpiecznych i niebezpiecznych testów
liczba_bezpiecznych_testow=$(echo "$response" | grep -o '"harmless": [0-9]*' | awk '{print $2}' | tr -d '[:space:]')
liczba_niebezpiecznych_testow=$(echo "$response" | grep -o '"malicious": [0-9]*' | awk '{print $2}' | tr -d '[:space:]')

# Wyświetlanie wyników
echo "Jest dokładnie $liczba_bezpiecznych_testow bezpiecznych testów"
echo "Jest dokładnie $liczba_niebezpiecznych_testow niebezpiecznych testów"

# Sprawdzanie, czy bezpieczne
if [[ "$liczba_niebezpiecznych_testow" -gt 0 ]]; then
    echo "Plik jest potencjalnie niebezpieczny."
else
    echo "Plik jest bezpieczny."
fi
```

```
Podaj ścieżkę pliku, który chcesz zeskanować: /home/kali/Downloads/eicar_com.zip
Podaj algorytm sumy kontrolnej (md5 lub sha256): md5
Suma kontrolna wynosi: 6ce6f415d8475545be5ba114f208b0ff
Jest dokładnie 1580 bezpiecznych testów
Jest dokładnie 4262 niebezpiecznych testów
Plik jest potencjalnie niebezpieczny.

Podaj ścieżkę pliku, który chcesz zeskanować: /home/kali/Downloads/NewEmptyFile.txt
Podaj algorytm sumy kontrolnej (md5 lub sha256): md5
Suma kontrolna wynosi: 80d4df5d89bad6454ea1491edaa80b8b
Jest dokładnie 00 bezpiecznych testów
Jest dokładnie 00 niebezpiecznych testów
Plik jest bezpieczny.
```

Sprawdź, czy wyniki są zgodne z oczekiwaniami - pobierz plik EICAR oraz utwórz nowy plik testowy.

Każdy etap skryptu powinien być opatrzony komentarzem.

KONTAKT

Adrian Florek

adrian.florek@pwr.edu.pl

PROGRAMOWANIE SKRYPTOWE

Zadanie dodatkowe:

Napisz skrypt bash, który automatyzuje pobieranie danych o pogodzie z serwisu API i generuje raport w formacie PDF.

Wymagania:

- Skrypt musi pobierać dane o pogodzie dla określonego miasta (przykładowo z serwisu API OpenWeatherMap - <https://openweathermap.org/>).
- Skrypt musi generować raport w formacie PDF, zawierający następujące informacje:
 - o Nazwa miasta
 - o Aktualna temperatura i wilgotność
 - o Prognoza pogody na 5 dni
- Skrypt musi być uruchamiany z wiersza poleceń i przyjmować jako argument nazwę miasta.
- Skrypt musi być dobrze udokumentowany, z komentarzami wyjaśniającymi jego działanie.