


POLITECHNIKA WROCŁAWSKA  Wydział Informatyki i Telekomunikacji	Wydział: Informatyki i Telekomunikacji Kierunek: Cyberbezpieczeństwo Rok Akademicki: 2024/2025 Rok studiów, semestr: I, 2 Grupa: 2 Termin: <i>poniedziałek, godz. 15.15</i>
Programowanie skryptowe – Laboratorium 4	
Prowadzący: mgr inż. Karolina Pfajfer Data wykonania ćwiczenia: 24.03.2025 Data oddania sprawozdania: 30.03.2025	Autor: <i>Adam Dąbrowski, 283832</i>

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
- b) Instalacja klienta pocztowego np. Thunderbird
- c) Dodanie nowego użytkownika
- d) Utworzenie katalogów dla nowego pracownika (Documents, Pictures, Videos)
- e) Wyświetlenie informacji o wersji systemu, adresu ip, adresu mac

W funkcji głównej skrypt powinien zapytać o nazwę użytkownika i przekazać tą informację do właściwej funkcji. Każdy etap skryptu powinien być opatrzony komentarzem.

```
GNU nano 8.3          nowy_komp.sh *
#!/bin/bash

#sprawdzenie aktualizacji i ich instalacja

check_upd() {
echo "_____ "
echo "Sprawdzanie i aktualizacja..."
#musze to zahaszować, ponieważ za każdym razem aktualizuje mi się system
sudo apt update
#sudo apt upgrade -y
}

#Instalacja thunderbird
install_T() {
echo "_____ "
echo "Instalacja Thunderbird"
sudo apt install -y thunderbird
}

#dodawanie użytkownika

dod_user() {
echo "_____ "
echo "Dodawanie użytkownika"
local uzytk=$1
#pytanie o nazwę użytkownika
#read -p "Podaj nazwę nowego użytkownika" username

sudo adduser $uzytk
}

#dodawanie kat dla nowego pracownika

dod_kat() {
echo "_____ "
echo "Tworzenie katalogów"
sciezka_uzyt="/home/$nazwa"
sudo mkdir -p "$sciezka_uzyt/Documents" "$sciezka_uzyt/Pictures" "$sciezka_uzyt/Videos"
}

#wypisuje wszystkie informacje takie jak wersja, IP i MAC
wys_inf() {
echo "_____ "
echo "Wersja systemu to"
```

```

#wypisuje wszystkie informacje takie jak wersja, IP i MAC
wys_inf() {
echo "_____ "
echo "Wersja systemu to"
lsb_release -a | awk 'NR ≥ 1'
echo "Adres IP= $(hostname -I | awk '{print $1}')"
echo "Adres MAC= $(ip link show | awk '/ether/ {print $2}')"
}

#funkcja wykonujaca wszystkie pozostale funkcje
funkcja_glowna() {
read -p "podaj nazwe uzytkownika" nazwa
local name=$nazwa
cat << AAA
1) check_upd
2) install_T
3) dod_user "$name"
4) dod_kat
5) wys_inf
AAA
read -p "Jesli chcesz wykonac pojedyncza funkcje kliknij liczbe od 1 do 5 jesli wszystkie 6 " numer

case $numer in
1)
check_upd ;;
2)
install_T ;;
3)
dod_user "$name" ;;
4)
dod_kat ;;
5)
wys_inf ;;
6)
check_upd; install_T; dod_user "$name"; dod_kat; wys_inf ;;
*)
echo "Nie poprawny numer"
esac

echo "Koniec"
}

funkcja_glowna

```

```

(userkali@hostkali)-[~]
$ ./nowy_komp.sh
podaj nazwe uzytkownika Adam
1) check_upd
2) install_T
3) dod_user "Adam"
4) dod_kat
5) wys_inf
Jesli chcesz wykonac pojedyncza funkcje kliknij liczbe od 1 do 5 jesli wszystkie 6 6

Sprawdzanie i aktualizacja...
Hit:1 http://http.kali.org/kali kali-rolling InRelease
9 packages can be upgraded. Run 'apt list --upgradable' to see them.

Instalacja Thunderbird
thunderbird is already the newest version (1:128.8.0esr-1).
The following packages were automatically installed and are no longer required:
  crackmapexec          libgtksourceview-3.0-1      libtag1v5-vanilla
  firebird3.0-common    libgtksourceview-3.0-common libtagc0
  firebird3.0-common-doc libgtksourceviewmm-3.0-0v5  libunwind-19
  libbfio1              libgumbo2                  libwebrtc-audio-processing1
  libc++1-19            libhdf5-103-1t64           libx265-209
  libc++abi1-19         libhdf5-hl-100t64          openjdk-23-jre
  libcapstone4          libjxl0.9                  openjdk-23-jre-headless
  libconfig++9v5        libldap-2.5-0              python3-appdirs
  libconfig9            libmbedcrypto7t64          python3-ntlm-auth
  libdirectfb-1.7-7t64  libmsgpack-0-1             python3-setproctitle
  libegl-dev            libnetcdf19t64             python3.12
  libflac12t64          libpaper1                  python3.12-dev
  libfmt9               libpoppler140              python3.12-minimal
  libgdal35             libpython3.12-dev          python3.12-venv
  libgeos3.13.0         libpython3.12-minimal      ruby-zeitwerk
  libgl1-mesa-dev       libpython3.12-stdlib       ruby3.1
  libglapi-mesa         libpython3.12t64          ruby3.1-dev
  libgles-dev           libqt5sensors5            ruby3.1-doc
  libgles1              libqt5webkit5             strongswan
  libglvnd-core-dev    libsuperlu6
  libglvnd-dev          libtag1v5
Use 'sudo apt autoremove' to remove them.

Summary:
  Upgrading: 0, Installing: 0, Removing: 0, Not Upgrading: 9

Dodawanie uzytkownika
New password:
Retype new password:
passwd: password updated successfully
Changing the user information for Adam
Enter the new value, or press ENTER for the default
  Full Name []:
  Room Number []:
  Work Phone []:

```

```
Dodawanie użytkownika
New password:
Retype new password:
passwd: password updated successfully
Changing the user information for Adam
Enter the new value, or press ENTER for the default
    Full Name []:
    Room Number []:
    Work Phone []:
    Home Phone []:
    Other []:
Is the information correct? [Y/n] y

Tworzenie katalogow

Wersja systemu to
Distributor ID: Kali
Description:    Kali GNU/Linux Rolling
Release:        2025.1
Codename:       kali-rolling
Adres IP= 10.0.2.15
Adres MAC= 08:00:27:b7:25:c7
Koniec

(userkali@hostkali)-[~]
$ sudo ls /home/Adam
Documents  Pictures  Videos
```

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

- a) Obliczy sumę kontrolną wskazanego pliku (MD5 lub SHA256).
- b) Wyśle zapytanie do API VirusTotal za pomocą polecenia curl.
- c) Zinterpretuje odpowiedź API i wyświetli informację, czy plik jest bezpieczny, czy nie. Sprawdź, czy wyniki są zgodne z oczekiwaniami - pobierz plik EICAR oraz utwórz nowy plik testowy. Każdy etap skryptu powinien być opatrzony komentarzem.


```
GNU nano 8.3 reputacja_plikow.sh
#!/bin/bash

read -p "Podaj nazwe pliku do sprawdzenia " plik
#tworzy sume kontrolna wskazanego pliku za pomoca SHA-256
hash=$(sha256sum $plik | awk '{print $1}')
echo "suma kontrolna hash: $hash"
#wyslanie zapytania do api virustotal
odp=$(curl -s -X GET "https://www.virustotal.com/vtapi/v2/file/report?\
apikey=83df948511c8fd060677f4035f11ed6e79335c990ef852592638e1956f7aa0d46\
resource=$hash")
#zczytuje odpowiedz od virustotal
positives=$(echo $odp | jq '.positives')
#echo $positives
#rozstrzyga czy plik jest bezpieczny czy nie
if [ $positives = 0 ] || [ $positives = "null" ]
then
echo "$plik jest bezpieczny"
else
echo "$plik jest niebezpieczny"
fi
```

```
(userkali@hostkali)-[~]
└─$ ./reputacja_plikow.sh
Podaj nazwe pliku do sprawdzenia plik.txt
suma kontrolna hash: a0884bf78edc0adae3ae60c3d215256408de892586b5806808876ad31f6eed47
plik.txt jest bezpieczny

(userkali@hostkali)-[~]
└─$ ./reputacja_plikow.sh
Podaj nazwe pliku do sprawdzenia pusty.txt
suma kontrolna hash: e3b0c44298fc1c149afbf4c8996fb92427ae41e4649b934ca495991b7852b855
pusty.txt jest bezpieczny

(userkali@hostkali)-[~]
└─$ ./reputacja_plikow.sh
Podaj nazwe pliku do sprawdzenia wirus.txt
suma kontrolna hash: 131f95c51cc819465fa1797f6ccacf9d494aaaaff46fa3eac73ae63ffbdf8267
wirus.txt jest niebezpieczny

(userkali@hostkali)-[~]
└─$ ./reputacja_plikow.sh
Podaj nazwe pliku do sprawdzenia system_info.txt
suma kontrolna hash: 940562f6c15453776b47687e068752f60b0eef1974e5c8690214cc6a2d8b00ab
system_info.txt jest bezpieczny
```

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.

```
GNU nano 8.3 openweather.sh
#!/bin/bash
# $1 - miasto
# klucz api z openWeather
api_key="af952005b87dee13260461981578d62c"
# zapytanie do strony openWeather które od razu filtruje i wypisuje jedynie potrzebne informacje
pogoda=$(curl -s "https://api.openweathermap.org/data/2.5/forecast?q=$1&appid=$api_key&units=metric" \
| jq -r '.list[] | select(.dt_txt | contains("12:00:00")) | "{(.dt_txt) - {(main.temp)}°C - {(main.humidity)}%"')
# przekazanie do pliku pdf informacji o pogodzie i mieście
echo -e "$1\n" > "prognozaMiasta$1.pdf"
echo -e "$pogoda" >> "prognozaMiasta$1.pdf"
```

```
(userkali@hostkali)-[~]
$ ./openweather.sh Wrocław
```

```
(userkali@hostkali)-[~]
$ cat prognozaMiastaWrocław.pdf
Wrocław

2025-03-31 12:00:00 - 9.33°C - 71%
2025-04-01 12:00:00 - 9.35°C - 72%
2025-04-02 12:00:00 - 13.16°C - 57%
2025-04-03 12:00:00 - 13.09°C - 62%
2025-04-04 12:00:00 - 15.84°C - 49%
```