

## OEF LOOPS

### 1. Bestaat

Doel: Nakijken of een commando uitvoerbaar is alvorens uit te voeren

Schrijf het script *bestaat.sh* met volgende functies:

- a) Kijkt na of de executable `"/bin/df"` uitvoerbaar is
- b) Als het uitvoerbaar is wordt het commando uitgevoerd met de optie `-h`
- c) Als het niet uitvoerbaar is eindigt het script met de melding:  
"Het bestand `/bin/df` is niet uitvoerbaar"

### 2. Bestaat versie 2

Doel: Nakijken of een commando uitvoerbaar is alvorens uit te voeren

De opzet van het script *bestaatv2.sh* is hetzelfde als dat van *bestaat.sh*. Dit keer kijken we volgende zaken specifieker na:

- a) Kijk na of het bestand `/bin/df` bestaat. Is dit niet zo dan geef je de melding `"/bin/df bestaat niet"` en je eindigt met exit 1
- b) Kijk dan na of `/bin/df` leesbaar én uitvoerbaar is (`r` én `x`). Is dit niet zo dan geef je de melding `"/bin/df is niet uitvoerbaar"` en je eindigt met exit 2

### 3. Apacheconfig

Doel: Vermits apache verschillende configuratiebestanden heeft, is het soms moeilijk om te vinden in welk bestand welke parameter staat. Dit script zoekt in een aantal apache configuratiebestanden naar een keyword dat wordt opgegeven als eerste argument.

Nodig: `sudo apt-get install apache2`

Schrijf het script *checkapache.sh* met volgende functies:

- a) Het script declareert volgende variabelen:  
`adir="/etc/apache2"`  
`confs="$adir/apache2.conf $adir/ports.conf"`
- b) Het script kijkt na of het eerste argument `$1` werd ingegeven. Is dit leeg, dan toont het script `"geen argument ingegeven"` en stopt met exit 1.
- c) Het script gebruikt een for lus om één voor één de bestandsnamen uit de

variabele \$confs te doorlopen

d) Het script kijkt na of het bestand bestaat (onbestaande bestanden geven geen foutmelding en het script loopt gewoon verder)

e) Het script zoekt in elk bestand uit \$confs of dit het meegegeven argument \$1 bevat (Hint: `grep -iHn "$1" "bestandsnaam"`)

Voorbeeldoutput:

```
./checkapache.sh LogLevel
# LogLevel: Control the number of messages logged to the error_log.
LogLevel warn
...
```

#### 4. Gewone gebruiker

Doel: Zoeken naar alle "gewone" gebruikers op het systeem

Nodig: Minstens één gewone gebruiker (normaal je default user)

Schrijf het script *forif.sh* dat volgende acties doet:

- a) Alle lijnen van het bestand `/etc/passwd` doorlopen met een `for` lus
- b) Uit de lijnen het derde veld filteren (hint: dit kan bv met `| cut -d: -f3`)
- c) Als de nummer die je vind `>= 1000` toon je de gebruikersnaam als volgt:

Gewone gebruikers op het systeem:

jancelis

kdguntu

- d) Vang de fouten in de `if` op (bekijk hiervoor 3.3 input validatie)

#### 5. Inlezen paswoord

Doel: Nakijken van een input string

Schrijf het script *geefpaswoord.sh*.

Hierin declareer je de variabele `paswoord` als volgt:

```
paswoord="supersecret"
```

Lees het paswoord in in de variabele 'mogelijkpaswoord' met:

```
read -s -p "Paswoord:" mogelijkpaswoord
```

Gebruik een `if` structuur om te kijken of het paswoord klopt.

Als het klopt geef je de melding "Toegelaten", anders geef je de melding "VERBODEN!"