OEF BEST PRACTICES

1. apachetest

Doel: Benchmark test voor apache webserver oproepen
Ref: man ab
man curl
Nodig: sudo apt-get install apache2 apache2-utils curl

Start de apache webserver op met: `sudo service apache2 start`
Gebruik volgende code in je script om de webserver te testen:

```
ab -n 100 -kc 10 ${url}
```

Let op: De url moet voor ab eindigen op een slash !

Schrijf het script *apachetest.sh* met volgende functies:
a) Voer een check uit op het bestaan van het programma ab (apache benchmark). Als dit niet geïnstalleerd is geef je de melding:
"Het programma ab is nodig. Installeren kan met sudo apt-get install apache2-utils" en het programma eindigt met exit code 1

b) Kijk na of de gebruiker een url heeft meegegeven
Als dit niet het geval is gebruik je 127.0.0.1 als url

```
Usage: ./apachetest [url]
```

c) Test eerst zelf uit met het programma `curl` of de meegegeven url bereikbaar is. Is dit niet zo, geef dan de foutmelding: De url is niet bereikbaar en eindig je met exit code 2

2. Grote bestanden

Doel: Zoeken naar grote bestanden
Ref: man find
man stat

Schrijf het script *findbig.sh* dat voldoet aan de best practices en volgende functies uitvoert:
a) Kijk met een if structuur na met welke opties jou programma wordt opgeroepen. Volgende opties moet je implementeren:
--help (enkel tonen wanneer deze optie ingegeven wordt)
--version (enkel tonen wanneer deze optie ingegeven wordt)

b) Als eerste argument en tweede argument
$1 het pad waarin je gaat zoeken (default /)
$2 de grootte van het bestand in MB (default 10MB)

Dit is het commando dat je moet oproepen:

```
find $path -type f -size "+${size}M" -exec stat -c '%s %n' {}
\;
```

c) Je script toont enkel een help wanneer --help als eerste argument werd gegeven

d) Je script toont enkel zijn versie wanneer --version als eerste argument wordt ingegeven:

```
./findbig.sh --version
findbig.sh version 0.1
```

e) Je script geen fouten wanneer $1 of $2 niet zijn ingegeven, maar gebruikt de default waarden (/ en 10 MB)

3. Systeemprogramma's

Doel: Weergeven programma's die vanuit sbin zijn opgestart. De gegevens van alle programma's staan in de directory /proc/PID/cmdline. (met PID= nummer van het proces)

Maak het script *cmdline.sh* dat voldoet aan de "best practices"*:*
a) Overloop met een for lus en het commando
`` `ls /proc/*/cmdline` `` alle cmdline bestanden op het systeem
b) Kijk na of de inhoud van het bestand niet leeg is
c) Toon enkel de inhoudslijnen die "`sbin`" bevatten
d) Kijk eerst na of het gevonden bestand nog bestaat voor je punt c uitvoert.

Voorbeeld output:

```
/proc/1143/cmdline: /usr/sbin/modem-manager
/proc/1188/cmdline: /usr/sbin/cupsd-F
...
```

4. Backup

Doel: Schrijven van een backup script
Ref: man find, man tar, man date

© 2019 jan.celis@kdg.be CC-BY-NA

Schrijf het script *backup.sh* dat voldoet aan de best practices met volgende functies:
a) Het script zoekt naar alle sh bestanden die recent gewijzigd zijn in alle subdirectories van /home
Het script bundelt de gevonden bestanden in het bestand:
Backup_2013_02_10_10_44.tar.gz

b) Het script zoekt default in de /home directory. Wanneer iemand als eerste argument een andere directory ingeeft, gebruik je dit als directory.

c) Kijk na of de ingegeven directory bestaat. Wanneer de gegeven directory niet bestaat eindig je met de foutmelding: "Opstarten met een directory als eerste argument: bv. sudo ./`basename $0` /home"

d) Definieer en gebruik de variabele
errorlog="/var/log/error_`basename $0`.log"
Hierin komen alle foutmeldingen (STDERR) van het script

e) Definieer en gebruik de variabele
backuplog="/var/log/backup_`basename $0`.log"
Hierin komen alle meldingen van STDOUT (zodat er niets op het scherm verschijnt.

f) Kijk na of het script als root wordt opgestart. Zoniet eindig je met de foutmelding: "Opstarten als root gebruiker: bv. sudo .\`basename $0`

5. Cron taak maken
Doel: Opstarten van een proces als regelmatig achtergrondproces
Ref: man 5 crontab

Schrijf het script maakcron.sh. Dit script gaat een crontaak aanmaken die elke minuut een bepaald commando uitvoert. Hiervoor maakt het script in de directory /etc/cron.d een bestand aan met de naam
/etc/cron.d/backupcron
met volgende inhoud:

```
# Minute Hour  Day_of_Month Month  Day of Week  User  Command
# (0-59) (0-23) (1-31)      (1-12)  (0-6)
```

© 2019 jan.celis@kdg.be CC-BY-NA

```
* * * * * root ${commando}
```

Het script kijkt volgende zaken na en eindigt anders met een passende foutmelding:
a) Het moet opgestart worden als root
b) Het moet een eerste argument $1 krijgen
c) Het eerste argument moet een uitvoerbaar bestand zijn (r en x)

Test uit met het backup script backup.sh
Opmerking: Eventuele cron fouten kan je terug vinden in de log: /var/log/syslog

d) Schrijf een --remove optie: bv:
`sudo ./maakcron.sh --remove /home/jancelis/backup.sh`
verwijdert het bestand /etc/cron.d/backup.sh.cron
of toont een foutboodschap wanneer dit niet aanwezig is.

e) Wanneer het eerste argument -h of --help is komt de volgende melding:

```
Usage: sudo ./maakcron.sh [ --remove] command
       --remove   Remove cron file
       --help     Display this help message
```

6. Openstaande poorten
Doel: Vinden welk proces een bepaalde poort gebruikt

Nodig: sudo apt-get install netcat

Schrijf in je shell script *netcat.sh* aan het begin volgende code:

```
# Dit start met netcat een luisterende poort
# (optie kleine L)op poort 13
nc -l 13
```

a) Gebruik een for of while loop om uit te zoeken welke processen er poorten 1 tot 100 hebben openstaan.
Met behulp van het commando `lsof` kan je je antwoord bekomen.
Voorbeeld met poort 8008:

```
lsof -nPi tcp:8008 | grep -i "listen"
# -n geen vertaling van ip adres/naar naam (domain lookup)
# -P geen vertaling van poortnummer naar poortnaam
# -i IP protocol
```

b) Voer je script uit als root (gewone gebruikers zien enkel poorten > 1023)
Eindig met een foutmelding wanneer het script niet als root wordt

© 2019 jan.celis@kdg.be CC-BY-NA

uitgevoerd

c) Zorg dat je script voldoet aan de best practices

d) Zorg voor een foutmelding wanneer netcat niet geïnstalleerd is op het systeem .

7. autologin

Doel: Script schrijven dat automatisch met ssh verbindt met "expect"
Ref: http://smacak.wordpress.com/2010/08/15/
man adduser
Nodig: sudo apt-get install openssh-server expect

a) Start de openssh server op met sudo service ssh restart

b) Schrijf het script *autologin.sh* dat voldoet aan de "Best Practices".
Het script krijgt als argumenten de hostname, username en paswoord.
Voorbeeld:

```
./autologin.sh  127.0.0.1 test supersecret
```

c) Geef een "Usage: ./autologin.sh hostname username password" wanneer er geen 3 argumenten ingegeven worden

d) Kijk na of ssh geïnstalleerd is, zoniet geef je een foutmelding en sluit je af met exit 1

e) Kijk na of de lokale ssh service wel reageert, zoniet geef je een foutmelding en sluit je af met exit 1. Testen of de lokale ssh service reageert kan met het programma netcat als volgt:

```
nc -w 1 127.0.0.1 22 2>/dev/null >/dev/null
```

f) Maak met het programma adduser een testgebruiker aan

g) Laat *autologin.sh* het volgende expect script (*expect.sh*) schrijven. Zorg ook dat de variabele gebruiker en paswoord gedeclareerd zijn.
Laat autologin het script ./expect.sh opstarten om automatisch in te loggen

```
#!/usr/bin/expect -f
set timeout 20
spawn ssh -o "StrictHostKeyChecking no" $username@127.0.0.1
expect "*assword:*"
send $paswoord\r
interact
```

© 2019 jan.celis@kdg.be CC-BY-NA

```
exit
```

8. apachetest2

Doel: Variant vraag 1 apachetest
Nodig: sudo apt-get install openssh-server expect

Schrijf een variant van het eerste script.
Wanneer ab niet geïnstalleerd is
a) Vraag je om apache2-utils te installeren (y/n)
b) Indien y installeer je apache2-utils:
Wanneer de installatie niet geslaagd is eindig je met de foutmelding:
Installatie apache2-utils mislukt en exit code 3
Lukt de installatie wel, dan voer je de rest van het programma uit.
c) Indien n eindig je met de foutmelding dat ab nodig is en exit code1.

© 2019 jan.celis@kdg.be CC-BY-NA