Benjis Dokumentation

Benjamin Ludwig

Contents

Chapter 1

Monitoring

1.1 Icinga Zeitprofile

es wird eine Zeitperiode definiert, in der Alarmiert werden soll.
Diese Periode ist dann mit 'check_period' auf den einzelnen Host
oder Service anzuwenden.
Im Beispiel soll immer alarmiert werden, AUSER von 05:00-06:25
jeden Tag.

Alarmierung für bestimmten Zeitpunkt abschalten:

```
define timeperiod {

     timeperiod_name 24x7_backup
     alias           immer-frueh
     sunday  00:00-05:00,06:25-24:00
     monday  00:00-05:00,06:25-24:00
     tuesday 00:00-05:00,06:25-24:00
     wednesday       00:00-05:00,06:25-24:00
     thursday        00:00-05:00,06:25-24:00
     friday  00:00-05:00,06:25-24:00
     saturday        00:00-05:00,06:25-24:00
}
```

Chapter 2

sonstige Hacks

2.1 Unter Ubuntu jffs2-images mounten

```
sudo apt-get install mtd-tools
sudo modprobe -v mtd
sudo modprobe -v jffs2
sudo modprobe -v mtdram total_size=256000 erase_size=256
sudo modprobe -v mtdchar
sudo modprobe -v mtdblock
sudo dd if=<deinImage.img> of=/dev/mtd0
sudo mount -t jffs2 /dev/mtdblock0 <deinPfadWoEsHinSoll>
```

2.2 Sed spielerei die Erste

```
Achtung mit den Hochkommas!
Zeile an bestimmter Position einfügen(hier zeile 12) und dazu
noch huebsch mit Tabulatoren formatieren:
sed ‘12i\\tTEXT\t\t\tMEHRTEXT’ <Datei>
```

2.3 Tunnel bauen

```
#!/bin/bash
#build the tunnel to remote_ip via host
ssh -N -L <local_port>:<remote_ip>:<remote_port user@host &
#connect to host, via local port
ssh -p <local_port> <user>@localhost
#tunnel a remote port to another machine while using an existing tunnel
ssh -p <local_port> root@localhost -L localhost:8080:192.168.1.1:80

#scp durch bestehenden Tunnel
scp -P <local_port> <datei> root@localhost:<remote_pfad>
#oder vom remote host holen
scp -P <local_port> root@localhost:<remote_pfad> <lokaler_pfad>
```

2.4 expect-scripts

```
#!/usr/bin/expect

if {$argc != 1} {
    send_user "\tusage: $argv0 <ip-address>\n"
    exit
}

set IPADDRESS [lindex $argv 0]

# security: write password to root only readable file in e.g. /root/authfiles
# so you may use this password here by:
#
#set PASSWORD_DIR   /root/authfiles
#set PASSWORD_FILE  "pwd-${IPADDRESS}"
#set status [catch { exec cat ${PASSWORD_DIR}${PASSWORD_FILE} } PASSWORD]
#
# alternatively set password simply here
set PASSWORD "<password>"

spawn /usr/bin/ssh admin@${IPADDRESS}

while (1) {
    expect {
        "password:" {
            send "${PASSWORD}\n"
            break
        }
        # this is useful, if ssh connects first time to IPADDRESS
        "connecting (yes/no)?" { send "yes\n" }
    }
}
expect "ES-2024PWR#" { send "show hardware-monitor c\n" }
expect "ES-2024PWR#" { send "exit\n" }
```

2.5 rsync-magic

```
logger -t Backup "begin incremental backup of <Directory>"
# incremental backup of /etc/apache2/*
rsync -chavz P --stats /etc/apache2 \
<user>@<server>:<path_on_remote_host>
logger -t Backup "incremental backup done"
```

2.6 Mounten unter Linux

```
place a credentials file at a place of your choise. in that case
> /etc/backup-creds
put username and password in it as below.

cat /etc/backup-creds
username=<Domain>/<Password>
password=<password of $username>

Mount manually with:
mount -t cifs -o rw,nobrl,nosuid,nodev,credentials=</path_to_credentials file> \
<//backup-server/backup_path </local_mount_point/<local_backup_path/>

or put it in /etc/fstab for mounting it on bootstrap:
<//backup-server/backup_path  </local_mount_point/<local_backup_path/>  \
cifs    noauto,credentials=/etc/backup-creds    0       0
```