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
\chapter{sonstige Hacks}
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
```

2.7 PDF Einschränkungen entfernen

```
Entfert Drucksperren, editier und extrahier-einschränkungen auf PDFs.

1.  Install QPDF:
    > sudo aptitude install qpdf
2.  Remove restrictions:
    > qpdf --decrypt input.pdf output.pdf
3.  To do this with many PDFs use the following one-liner:
    > for file in *.pdf; do qpdf --decrypt $file ${file/.pdf/_rescued.pdf}; done
```

2.8 Ubuntu XFCE extended Screen

```
1.  Install arandr:
    > sudo apt-get install arandr
2.  arandr von der comando-zeile aus starten. Ein GUI geht auf und dann die
    Bildschirme zurecht rücken wie man es braucht.
```

2.9 Config Routing add/del

```
Routen Setzen um Gateway im Entsprechenden Netz zu erreichen:
sudo route add -net 10.0.2.0/24 eth0
sudo route del -net 10.0.2.0 netmask 255.255.255.0 dev eth0

IP-Forwarding zwichen 2 Interfacen in Linux aktivieren
sysctl -w net.ipv4.ip_forward=1
echo 1 >/proc/sys/net/ipv4/ip_forward
```

Chapter 3

Datenbanken

3.1 Postgres DB - HBA config

für Postgresql gibt es eine Datei /etc/postgresql/<VERSION>/main/pg_hba.conf
die als Art "Firewall" Funktion für die Datenbank funktioniert.

Standardmässig besagt diese das Verbindungen ausschliesslich von Lokal auf die Datenbank gemacht werden dürfen.

um dies zu Ändern muss die entsprechende IP oder das Netz angegeben werden:

```
# Database administrative login by UNIX sockets
local   all         postgres                          trust

# TYPE  DATABASE    USER        CIDR-ADDRESS          METHOD

# "local" is for Unix domain socket connections only
#local  all         all                               ident
local   all         all                               ident
# IPv4 local connections:
#host   all         all         127.0.0.1/32          md5
host    all         all         127.0.0.1/32          md5
host    all         all         192.168.0.1/24        trust
# IPv6 local connections:
#host   all         all         ::1/128               md5
host    all         all         ::1/128               md5
host    all         all         192.168.0.1/24        md5
```

3.2 Postgres Tunnel für pgadmin

pgadmin wird verwendet um eine GUI Oberfläche für Postgresql Datenbanken zu haben. Da wegen der oben bereits erwähnten Firewall meist nur von lokal aus verbunden werden kann, benötigt es einen Tunnel um eine Verbindung herzustellen.

der Tunnel wird wie gewöhnlich über SSH gestartet:

ssh -L <LokalerPort>:localhost:<5432(standard bei psql)> username@remote_ip

3.3 Datenbank Passwort to md5

UPDATE <TABLE> SET <ATTRIBUTE>=md5('pass') WHERE <ATTRIBUTE>='<VALUE>';