Karl und Karlchen

Seit Juni 2011 betreibt die Computer-AG zwei verschiedene Server: Einmal Karl als Infrastrukturserver, bei dem es vor allem auf Stabilität und Zuverlässigkeit ankommt und Karlchen als Bastelkiste. In der folgenden Beschreibung geht es meistens um Karlchen.

Auf dieser Seite und auch auf den Unterseiten zu Karl und Karlchen sind keine Benutzernamen zu nennen und deswegen auch nicht zu finden!

Basisconfig

Karls Inhalt von /etc/network/interfaces

```
auto eth0
iface eth0 inet static
    address 141.10.58.147
    netmask 255.255.255.248
    network 141.10.58.144
    broadcast 141.10.58.151
    gateway 141.10.58.145
    # dns-* options are implemented by the resolvconf package, if installed
    dns-nameservers 129.143.2.4
    dns-search kvfg.info
```

Karlchens Inhalt von /etc/network/interfaces

```
auto eth0
iface eth0 inet static
    address 141.10.58.148
    netmask 255.255.255.248
    network 141.10.58.144
    broadcast 141.10.58.151
    gateway 141.10.58.145
    # dns-* options are implemented by the resolvconf package, if installed
    dns-nameservers 129.143.2.4
    dns-search kvfg.info
```

Inhalt von /etc/resolv.conf

```
search kvfg.info
nameserver 129.143.2.4
```

/etc/hosts angepasst:

```
127.0.0.1   localhost
```

```
# bei Karl
141.10.58.147   www.kvfg.info www karl
# bei Karlchen
141.10.58.148   karlchen.kvfg.info karlchen
```

.bashrc angepasst und nach /etc/skel geschrieben:

```
# ~/.bashrc: executed by bash(1) for non-login shells.

export PS1='\u@\h:\w\$ '
umask 022

# You may uncomment the following lines if you want `ls' to be colorized:
# export LS_OPTIONS='--color=auto'
# eval "`dircolors`"
alias ls='ls $LS_OPTIONS'
alias ll='ls $LS_OPTIONS -l'
alias l='ls $LS_OPTIONS -lA'
alias la='ls -la'
# Some more alias to avoid making mistakes:
alias rm='rm -i'
alias cp='cp -i'
alias mv='mv -i'
alias ..='cd ..'
alias ...='cd ../..'
```

Wie wir feststellen mussten, hat sich ein echo und ein who in der .bashrc dahingehend ausgewirkt, dass per scp keine Dateiübertragungen auf den Server mehr stattfinden konnten. Die Einträge wurden demnach wieder entfernt.

/etc/default/useradd angepasst:

```
SHELL=/bin/bash
```

Testweise neuen Nutzer hinzugefügt und diesem ein Passwort verpasst:

```
useradd -m neuerbenutzer
passwd neuerbenutzer
```

Sicherheit

sudo

Unter **Debian** sudo Umgebung installiert:

```
apt-get install sudo
```

/etc/sudoers mit

```
visudo
```

(visudo ruft nen nano auf und führt Syntaxchecks durch) angepasst und für unseren normalen Benutzer konfiguriert. Dieser hat nun sudo Rechte!

Unter **Debian**:

```
sudo adduser unserbenutzer sudo
```

Unter **Ubuntu** ist das sudo System von Haus aus aktiv und muss nicht extra installiert werden. Hier reicht es demnach neue Benutzer der Gruppe admin hinzuzufügen:

```
sudo adduser benutzername admin
```

SSH umstellen auf publickey

Im ersten Schritt wird die

```
/etc/ssh/sshd_config
```

angepasst:

```
Port 2222
PermitRootLogin no
X11Forwarding no
PrintMotd no
ChallangeResponseAuthentication no
PasswordAuthentication no
```

und dann der SSH neu gestartet:

```
/etc/init.d/ssh restart
```

Auf dem lokalen Rechner dann mit dem folgenden Befehl ein Keypaar für den Benutzer „benutzer“ erstellen und den Schlüssel mit einem **Kennwort** sichern, sollte dieser Schlüssel verloren gehen:

```
ssh-keygen -f /home/benutzer/.ssh/benutzer-karlchen
```

Den public key mit Hilfe von scp auf den Server übertragen und in dessen Server-Schlüsselbund übernehmen:

```
scp /home/benutzer/.ssh/benutzer-karlchen.pub
benutzer@karlchen.domain:/home/benutzer
```

Weiter geht es auf dem Server im Home des entsprechenden Benutzers mit:

```
cat benutzer-karlchen.pub >> /home/benutzer/.ssh/authorized_keys
```

Während der ganzen Prozedur auf jeden Fall eine Shell zum Server offenhalten, damit man sich nicht aus Versehen selbst aussperrt.

Zu erreichen ist das System nun über:

```
ssh benutzer@kvfg.info -p 2222 -i /home/benutzer/.ssh/benutzer-karlchen
```

Wer keinen Schlüssel hat und sich über SSH verbindet erhält die Meldung

```
Permission denied (publickey).
```

Jeder Benutzer auf Karlchen hat bei ausreichender Kenntnis sudo Rechte. Der Gebrauch dieser sudo Rechte wird gespeichert in:

```
/var/log/auth.log
```

Anmeldeversuche von root tauchen wie folgt auf:

```
sshd[3074]: Failed password for root from
```

fail2ban

Jetzt noch ein zusätzlicher Ausflug für die Sicherheit. Für **Debian**:

```
apt-get install fail2ban mailx
```

Unter **Ubuntu**:

```
apt-get install fail2ban mailutils
```

Das vorhandene Jail wird nun kopiert und dann angepasst.

```
cp /etc/fail2ban/jail.conf /etc/fail2ban/jail.local
```

Anpassungen vorgenommen in:

```
/etc/fail2ban/jail.local
```

Aktiviert wurden die Jails für die genutzten Dienste - also unter anderem:

- SSH
- SASL

Außerdem wurde eine IP Adresse eingerichtet, die nicht gesperrt werden kann. Welche das ist, wird hier nicht verplappert.

Dann wurde der Dienst neu gestartet:

```
/etc/init.d/fail2ban restart
```

Weitere Schritte

Weiter: Mailserver

für das Setup eines eher traditionellen **LAMP** Systems.

Weiter: Kolab

für das Setup eines **Kolab**servers auf Debian Etch.

Weiter: Subversion

für das Setup eines **SVN**servers auf Karlchen.