



Berner Fachhochschule
Hochschule für Technik und Informatik HTI

Installationshandbuch

Version 10.01.2005

Diplomarbeit I00 (2004)

MuSeGa



Mobile User Secure Gateway

Experte: Andreas Dürsteler (Swisscom)
Betreuer: Hansjürg Wenger (HTI)
Gerhard Hassenstein (HTI)
Diplomand: Lukas Reusser (Swisscom)



Abstract

Diese Anleitung beschreibt wie man vorgehen muss, um MuSeGa in seinem vollen Funktionsumfang auf einem System zu installieren.

Versionskontrolle

Version	Datum	Kommentar	Genehmigt
0.1	08.01.2005	Erster Draft	
0.2	10.01.2005	Alles fertig!	Lu



Inhaltsverzeichnis

1 Allgemeines.....	4
1.1 Sinn und Zweck.....	4
1.2 Leserkreis.....	4
1.3 Umfang.....	4
2 Voraussetzungen.....	5
2.1 Hardware.....	5
2.2 Software.....	5
3 Installation von Debian 3.0.....	5
3.1 Booten von CD.....	5
3.2 Harddisk Partitionierung.....	5
3.3 Partitionen initialisieren.....	6
3.4 Netzwerk konfigurieren.....	6
3.5 Basissystem installieren.....	6
3.6 Konfiguration nach Reboot.....	6
4 Konfiguration von Debian 3.0.....	7
4.1 Einloggen als root.....	7
4.2 APT konfigurieren.....	7
4.3 System Hardening.....	8
5 Debian Update auf Version 3.1.....	9
6 Selbst kompilierte Software.....	10
7 Benötigte Software.....	15
7.1 Apache.....	15
7.1.1 Installation.....	15
7.1.2 Konfiguration.....	15
7.2 PPTPD.....	17
7.2.1 Konfiguration.....	17
7.3 RADIUS-Client.....	19
7.4 MySQL.....	20
7.4.1 phpmyadmin.....	20
7.5 Perl Module.....	20
7.6 phpsysinfo.....	20
7.7 tcpdump.....	21
7.8 lynx.....	21
7.9 dhcpd.....	21
8 MuSeGa Software.....	21
8.1 Installation.....	21
8.2 Berechtigungen Überprüfen.....	22
8.3 MuSeGa Starten.....	24
9 Installierte Pakete überprüfen.....	25
10 Weiteres Vorgehen.....	25
11 Glossar.....	25



1 Allgemeines

1.1 Sinn und Zweck

Sinn dieses Dokumentes ist es, den zukünftigen Administrator des Mobile User Secure Gateways durch die Installation zu führen, bis er ein lauffähiges System hat.

1.2 Leserkreis

Dieses Dokument richtet sich an einen erfahrenen UNIX/Linux Administrator der sich zum Ziel gesetzt hat, MuSeGa auf einem Rechner zu installieren.

1.3 Umfang

Der Umfang beschränkt sich auf die reine Installation von MuSeGa. Technische Details entnehmen Sie aus der technischen Dokumentation. Wie Sie MuSeGa später konfigurieren, entnehmen Sie aus dem Betriebshandbuch.



2 Voraussetzungen

2.1 Hardware

Sie brauchen einen Rechner mit mindestens zwei Netzwerkkarten um MuSeGa betreiben zu können. Je nach der Geschwindigkeit ihrer Netzwerkkarten brauchen sie auch eine schnellere CPU. Haben Sie zwei Gigabit-Karten, sollten sie sehr schnelle CPUs (>2GHz) wählen.

Das Referenzsystem ist ein **IBM eServer xSeries 335** mit folgender Ausstattung: Dual Xeon 2.8GHz, 1GB Memory, 2x 36GB 15'000rpm Disk, Dual Gigabit Ethernet, Hardware RAID-Controller

2.2 Software

MuSeGa basiert auf Debian Linux 3.1 (Codename Sarge). Sie sollten also über eine Debian Installations-CD, sowie über einen schnellen Internetzugang verfügen.

3 Installation von Debian 3.0

3.1 Booten von CD

Um von der CD zu starten, müssen Sie einfach die erste CD von Debian einlegen. Beim darauf erscheinenden Prompt sollten Sie unbedingt **bf24** eingeben, damit von Anfang an ein Kernel der Serie 2.4 verwendet wird. Andernfalls wird ein Kernel der Version 2.2 verwendet.

3.2 Harddisk Partitionierung

Nach kurzer Zeit befinden Sie sich im Installationsmenü von Debian. Der oberste Punkt ist immer derjenige, welcher meistens als nächster ausgeführt werden sollte. Zuerst müssen Sie also die Harddisk für Debian partitionieren. Die Partitionierungstabelle sollte für MuSeGa ungefähr wie folgt aussehen.

```
musega:~# df -kh
Filesystem      Size  Used Avail Use% Mounted on
/dev/sda2       3.8G  233M  3.3G   7% /
tmpfs           506M    0  506M   0% /dev/shm
/dev/sda1        61M   40M   19M  68% /boot
/dev/sda3       3.8G  1.2G  2.5G  32% /usr
/dev/sda5       471M   8.1M  439M   2% /tmp
/dev/sda7        19G  2.8G   15G  16% /var
```

Bei unserem Referenzsystem stehen insgesamt 36GB zur Verfügung. Die /var Partition sollten Sie möglichst gross (>20-30GB) wählen, da alle Logfiles sowie alle Statistiken in diesem Verzeichnis abgelegt werden.



3.3 Partitionen initialisieren

Die erstellten Partitionen müssen nun initialisiert werden. Als erstes sollte Sie / dev/sda2 wählen, weil diese die Rootpartition ist und als erstes initialisiert werden muss. Verwenden Sie das Journaling-Filesystem ext3 für alle Partitionen.

Weiter geht es mit „Install kernel and drivers modules“ und „configure device driver modules“. Hier sollten Sie den Treiber für die verwendete Netzwerkkarte auswählen. An der HTI werden zum Beispiel Karten von Intel (EtherExpress) verwendet. Dafür lädt man das Modul eeepro100.

3.4 Netzwerk konfigurieren

Als Hostnamen verwenden Sie zum Beispiel musega. Am besten konfigurieren Sie erst einmal nur das Interface eth0. Dabei handelt es sich um das interne Interface. Verwenden Sie also eine gültige Adresse aus ihrem Netzbereich (zum Beispiel 172.17.35.105).

3.5 Basissystem installieren

Haben Sie das Netzwerk korrekt konfiguriert, wählen Sie den Punkt „Install the base system“. Anschliessend installieren Sie mit dem Punkt „Make the system bootable“ noch einen Bootloader. Die Basisinstallation ist nun abgeschlossen und Sie müssen das System neu starten.

3.6 Konfiguration nach Reboot

Nach dem Neustart setzen Sie die Zeitzone auf „europe/zurich“ und aktivieren aus Sicherheitsgründen die Funktionen „shadow passwords“ und „md5 passwords“. Anschliessend werden Sie aufgefordert ein gutes Root-Passwort zu setzen. Ebenfalls sollte auch gleich ein normaler Benutzer (ohne root Privilegien) angelegt werden. Die Frage zum entfernen der pcmcia Pakete können Sie getrost mit ja beantworten, wenn Sie keine pcmcia Geräte haben.

Um Software nachinstallieren zu können und das System immer auf dem aktuellen Stand zu halten, werden Sie anschliessend aufgefordert so genannte APT-Sourcen anzugeben. Dabei handelt es sich um URLs welche auf die benötigte Software und Updates zeigen. Der Switch-Mirror ist normalerweise sehr schnell: ftp://mirror.switch.ch. Die bereits enthaltenen Links auf die Security-Updates können Sie übernehmen.

In tasksel könnten Sie noch weitere zu installierende Gruppen von Paketen angeben. Wählen Sie aber nothing und finish. Auch dselect sollten Sie überspringen. Damit wäre es möglich noch einzelne Pakete zu installieren. Beantwortet Sie nun die folgende Frage nach der Installation der ausgewählten Software mit „Y“, werden die Pakete heruntergeladen und installiert. Bei ssh werden Sie noch gefragt ob Sie ssh Version 1 erlauben wollen. Dort sollten Sie ssh2 only anwählen. Auch die Mail-Installation stellt noch ein paar Fragen. Am einfachsten wählen Sie local delivery only und geben den Namen des vorhin installierten Benutzers an, um alle erzeugten Emails zu erhalten. Haben Sie all diese Konfiguration hinter sich, gelangen Sie zum Login-Screen.



4 Konfiguration von Debian 3.0

4.1 Einloggen als root

Nun ist Debian Linux 3.0 eigentlich fertig installiert und bereit zur Arbeit. Es sind nun jedoch noch einige Konfigurationen manuell auszuführen.

Falls Sie über einen Proxy-Server mit dem Internet verbunden sind, sollten Sie die `http_proxy` Variable wie folgt setzen:

```
export http_proxy=http://proxy.foo.bar:8080
```

Im Falle der HTI ist dies jedoch nicht nötig.

4.2 APT konfigurieren

Als erstes sollten Sie nun `/etc/apt/apt.conf` editieren und folgende Zeilen einfügen:

```
APT::Default-Release "3.0*";  
APT::Cache-Limit "25165824";
```

Die erste Zeile weist apt an, nur Pakete aus dem Stable-Release 3.0 zu installieren. Die zweite stellt sicher, dass apt auch noch korrekt funktioniert, wenn viele Quellen verwendet werden.

Eine weitere Datei die ebenfalls sofort editiert werden sollte ist `/etc/apt/sources.list`:

```
# stable  
deb ftp://mirror.switch.ch/mirror/debian/ stable main non-free contrib  
deb-src ftp://mirror.switch.ch/mirror/debian/ stable main non-free contrib  
deb http://non-us.debian.org/debian-non-US stable/non-US main contrib non-free  
deb-src http://non-us.debian.org/debian-non-US stable/non-US main contrib non-free  
  
# testing  
deb ftp://mirror.switch.ch/mirror/debian/ testing main non-free contrib  
deb-src ftp://mirror.switch.ch/mirror/debian/ testing main non-free contrib  
  
# security updates  
deb http://security.debian.org/ stable/updates main contrib non-free  
deb http://security.debian.org/ testing/updates main contrib non-free
```

Diese Datei enthält alle Quellen in denen apt nach Software sucht. Im ersten Teil haben wir alle URLs für Debian Stable angegeben und im unteren Teil alle Quellen für Debian Testing. Testing ist die neue Version von Debian Linux. Sie trägt den Codenamen Sarge und die Versionsnummer 3.1

Haben Sie diese Dateien editiert, müssen Sie `apt-get update` ausführen. Dabei werden die URLs in `sources.list` durchgegangen und die entsprechenden Inhaltsverzeichnisse herunter geladen.



Mit dem Befehl „*apt-get upgrade*“ können Sie dann alle Pakete aktualisieren, welche nicht mehr auf dem neusten Stand sind.

4.3 System Hardening

Ein minimales System Hardening sollten Sie jetzt auf jeden Fall durchführen. Mit *netstat* können Sie herausfinden, auf welchen Ports ein Dienst läuft:

```
musega:~# netstat -na
Active Internet connections (servers and established)
Proto Recv-Q Send-Q Local Address           Foreign Address         State
tcp      0      0 0.0.0.0:80              0.0.0.0:*               LISTEN
tcp      0      0 0.0.0.0:10000          0.0.0.0:*               LISTEN
tcp      0      0 0.0.0.0:22             0.0.0.0:*               LISTEN
tcp      0      0 0.0.0.0:443            0.0.0.0:*               LISTEN
tcp      0      0 10.10.0.1:1723         0.0.0.0:*               LISTEN
tcp      0      224 192.168.112.5:22      192.168.112.10:41644    ESTABLISHED
```

Für MuSeGa brauchen Sie ausser Secureshell (SSH) erstmal gar nichts. Also setzen Sie vor jede Zeile in */etc/inetd.conf* einen Gartenhag (#). Dann stoppen Sie alle nicht gebrauchte Dienste:

```
/etc/init.d/inetd stop
/etc/init.d/portmap stop
/etc/init.d/lpd Stop
```

Die gestoppten Dienste sollen natürlich auch beim nächsten Systemstart nicht wieder gestartet werden:

```
rm /etc/rcS.d/S41portmap
cd /etc/rc3.d
rm S14ppp S19nfs-common S20exim S20inetd S20lpd S20nfs-kernel-server
```

Weiter gibt es auf dem System noch eine Reihe Software die Sie bedenkenlos entfernen können:

```
musega:~# apt-get remove lpr mailx ppp pppoe pppconfig pppoeconf nfs-common
nfs-kernel-server portmap vacation
```

Nun sollten Sie das System neu starten um sich davon zu überzeugen, dass wirklich nur noch die gewünschten Dienste starten:

```
musega:~# shutdown -r now
```

Ein *netstat -na* sollte jetzt wirklich nur noch den Port 22 (ssh) zeigen.

Nun wäre übrigens auch ein guter Zeitpunkt um ein kleines Backup des ganzen



Systems zu machen.

5 Debian Update auf Version 3.1

MuSeGa läuft zwar auch unter Debian 3.0, da wir aber etwas mit der Zeit gehen möchten und Debian 3.1 sowieso bald (sicher in 2005) offiziell freigegeben wird, verwende ich die Version 3.1. Um also jetzt ein Upgrade auf die Version 3.1 durchzuführen, editieren Sie die Datei `/etc/apt/apt.conf` und ändern den Default-Release auf `testing`:

```
APT:Default-Release "testing";
```

Anschliessend führen Sie folgende Befehle aus:

```
musega:~# apt-get update  
musega:~# apt-get dist-upgrade
```

Dies wird nun eine Weile dauern. Auf einem Pentium IV mit 3.2GHz, 2GB RAM und einem 100MBit Internetzugang dauert das ganze zwischen 20 und 30 Minuten. Unter VMWare wohlgemerkt.

```
musega:~# apt-get dist-upgrade  
Reading Package Lists... Done  
Building Dependency Tree... Done  
Calculating Upgrade... Done  
The following packages will be REMOVED:  
  console-tools-libs  
The following NEW packages will be installed:  
  aptitude coreutils cpp-3.3 debconf-i18n dictionaries-common dselect e2fslibs g++-  
  3.3 gcc-3.3 gcc-3.3-base initscripts libacl1 libattr1 libblkid1 libbz2-1.0  
  libcomerr2 libconsole libdb1-compat libdb3-util libdb4.0 libdb4.2 libdevmapper1.00  
  libdns11 libgcrypt1 libgdbm3 libgnutls7 libidn11 libisc7 liblocale-gettext-perl  
  liblzo1 libmagic1 libncursesw5 libnewt0.51 libpng10-0 libsasl2 libsigc++-1.2-5c102  
  libss2 libssl0.9.7 libstdc++5 libstdc++5-3.3-dev libtasn1-0 libtext-charwidth-perl  
  libtext-iconv-perl libtext-wrapi18n-perl libtextwrap1 libuuid1 linux-kernel-headers  
  python2.3 slangla-utf8 sysv-rc wamerican  
151 packages upgraded, 51 newly installed, 1 to remove and 0 not upgraded.  
Need to get 91.0MB/91.0MB of archives. After unpacking 102MB will be used.  
Do you want to continue? [Y/n] Y
```

Während der Installation werden noch ein paar Fragen gestellt. Mit etwas Erfahrung können Sie diese aber problemlos beantworten.

Anschliessend sollten Sie das System neu starten um zu sehen ob noch alles normal funktioniert und startet. Ist alles gut gelaufen, können Sie sich daran machen, einen Kernel der Serie 2.6 zu erstellen. Lesen Sie dazu das nächste Kapitel.



6 Selbst kompilierte Software

MuSeGa verwendet einige spezielle Funktionalitäten, welche noch keinen Einzug in den Linux Kernel gefunden haben. Darum müssen Sie sich nun die aktuellen Kernel-Sourcen besorgen, diese mit den nötigen Patches versehen und anschliessend den Code neu kompilieren. Mit dem Kernel alleine ist es aber nicht getan. Weiter müssen Sie sich die Sourcen von pppd und pptpd besorgen und ein paar Änderungen vornehmen. Zu guter Letzt müssen Sie auch noch iptables neu kompilieren.

Im Folgenden werde ich ihnen detailliert erläutern wie Sie dabei vorzugehen haben:

Ins Verzeichnis /usr/src wechseln.

```
musega:~# cd /usr/src
```

Dann brauchen wir noch etwas Software für später:

```
musega:/usr/src# apt-get install make gcc kernel-package dpkg-dev
```

Herausfinden, welches die aktuellen Kernel Sourcen sind:

```
musega:/usr/src# apt-cache search kernel-source-2.6.  
kernel-patch-debian-2.6.8 - Debian patches to Linux 2.6.8  
kernel-source-2.6.8 - Linux kernel source for version 2.6.8 with Debian  
patches
```

Aktuelle Version (2.6.8) herunterladen und entpacken

```
musega:/usr/src# apt-get install kernel-source-2.6.8  
musega:/usr/src# bunzip2 -c kernel-source-2.6.8.tar.bz2 | tar x  
musega:/usr/src# ln -s kernel-source-2.6.8 linux
```

MPPE/MPPC kernel module for Linux herunterladen und Kernel patchen

Weitere Informationen befinden sich hier: <http://www.polbox.com/h/hs001/>

```
musega:/usr/src# wget http://www.polbox.com/h/hs001/linux-2.6.8-mppe-mppc-  
1.1.patch.gz  
musega:/usr/src# gzip -d linux-2.6.8-mppe-mppc-1.1.patch.gz  
musega:/usr/src# patch -p0 -i linux-2.6.8-mppe-mppc-1.1.patch
```

pppd Sourcen herunterladen, mppe-mppe Patch herunterladen und patchen



```
musega:/usr/src# wget ftp://ftp.samba.org/pub/ppp/ppp-2.4.3.tar.gz
musega:/usr/src# wget http://www.polbox.com/h/hs001/ppp-2.4.3-mppe-mppc-1.1.patch.gz
musega:/usr/src# tar zxvf ppp-2.4.3.tar.gz
musega:/usr/src# gzip -d ppp-2.4.3-mppe-mppc-1.1.patch.gz
musega:/usr/src# patch -p0 -i ppp-2.4.3-mppe-mppc-1.1.patch
```

Nun muss noch das RADIUS-Plugin gepatcht werden. Dieser Fehler hat mich einige Stunden gekostet!

In Zeile 705 in der Datei /usr/src/ppp-2.4.3/pppd/plugins/radius/radius.c muss folgender Code eingefügt/ersetzt werden:

```
/*if (mppe_enc_policy && mppe_enc_keys) {*/
    if (mppe_enc_keys) {
        mppe_keys_set = 1;
        if (!mppe_enc_policy) {
            /* MS-MPPE-Encryption-Policy = 'require_encryption' */
            mppe_enc_policy = 0x00000002;
        }
        /* Set/modify allowed encryption types. */
        if (mppe_enc_types)
            set_mppe_enc_types(mppe_enc_policy, mppe_enc_types);
    }
}
```

Das Problem ist, dass das Plugin mit einem Fehler abbricht, wenn es keine Encryption-Policy vom RADIUS-Server erhält. Dies ist leider beim RADIUS-Server der HTI der Fall. Mit diesem Patch setzte ich nun diese Policy auf „require_encryption“ wenn der RADIUS-Server nichts von sich aus mit gibt.

Nun können Sie pppd kompilieren und installieren

```
musega:/usr/src# cd ppp-2.4.3
musega:/usr/src/ppp-2.4.3# ./configure
musega:/usr/src/ppp-2.4.3# make
musega:/usr/src/ppp-2.4.3# make install
```

pptpd von www.poptop.org herunterladen, kompilieren und installieren

```
musega:/usr/src# tar zxvf pptpd-1.2.1.tar.gz
musega:/usr/src# cd pptpd-1.2.1
musega:/usr/src/pptpd-1.2.1# ./configure
musega:/usr/src/pptpd-1.2.1# make
musega:/usr/src/pptpd-1.2.1# make install
```

patch-o-matic von netfilter.org herunterladen und entpacken



```
musega:/usr/src# wget http://www.netfilter.org/files/patch-o-matic-ng-20040621.tar.bz2
musega:/usr/src# tar xvfj patch-o-matic-ng-20040621.tar.bz2
```

iptables Sourcen herunterladen und entpacken

```
musega:/usr/src# apt-get source iptables
musega:/usr/src# tar xvfj iptables-1.2.11.tar.bz2
musega:/usr/src# ln -s iptables-1.2.11/ iptables
```

Umgebung setzen und patch-o-matic starten

```
musega:/usr/src# cd patch-o-matic-ng-20040621#
musega:/usr/src/patch-o-matic-ng-20040621# export KERNEL_DIR=/usr/src/linux
musega:/usr/src/patch-o-matic-ng-20040621# export
IPTABLES_DIR=/usr/src/iptables

# Folgende Zeile umgeht einen kleinen Bug:
echo "IPTABLES_VERSION:=1.2.11" >> /usr/src/iptables/Makefile

musega:/usr/src/patch-o-matic-ng-20040621# ./runme base
```

Nachdem `./runme base` gestartet wurde, werden Ihnen verschiedene iptables Features aufgelistet, und Sie werden gefragt, ob Ihr Kernel dieses Feature unterstützen soll. Lesen Sie sorgfältig und wählen Sie bei folgenden Features YES:

connlimit
dstlimit
iprange
mport
quota
time

Vor der Konfiguration des Kernels müssen Sie noch ein paar Libraries installieren:

```
musega:/usr/src/linux# apt-get install libncurses5-dev libncurses5 libc6-dev
```

Konfiguration des Kernels starten:

```
musega:/usr/src/linux# make menuconfig
```

Bei der Konfiguration sind Sie ziemlich frei. Ich ziehe es jeweils vor, auf einem Server alle nicht gebrauchten Dinge weg zulassen. Auf jeden Fall muss der Kernel ihre



aktuelle Hardware unterstützen. Weiter sollten Sie alle iptables relevanten Dinge aktivieren.

Folgende Punkte haben wir mit den Kernel-Patches eingefügt. Sie müssen auf jeden Fall aktiviert werden!

```
#
# ATM drivers
#
CONFIG_PPP_MPPE_MPPC=m

#
# IP: Netfilter Configuration
#
CONFIG_IP_NF_MATCH_IPRANGE=m
CONFIG_IP_NF_MATCH_CONNLIMIT=m
CONFIG_IP_NF_MATCH_DSTLIMIT=m
CONFIG_IP_NF_MATCH_MPORT=m
CONFIG_IP_NF_MATCH_QUOTA=m
CONFIG_IP_NF_MATCH_TIME=m
```

Sind Sie mit der Konfiguration fertig, speichern Sie diese und verlassen sie menuconfig.

Nun startet man aber nicht wie bei vielen anderen Distributionen make, sondern Sie erstellen gleich ein Debian-Paket des neuen Kernels. Dazu geben sie folgenden Befehl ein:

```
musega:/usr/src/linux# make-kpkg --append-to-version -mppe-ipt --
revision=luk.0 --initrd kernel_image
```

Dieser Befehl kompiliert unseren Kernel und legt ihn in einer .deb-Datei ab. Um diesen Kernel nun zu installieren verwenden Sie folgenden Befehl:

```
musega:/usr/src# dpkg -i kernel-image-2.6.8-mppe-ipt_luk.0_i386.deb
```

Nun sollten Sie noch die Datei /etc/lilo.conf anpassen:

```
prompt
delay=100
timeout=100
```

Diese drei Zeilen müssen unbedingt vorhanden sein, da Sie sonst beim Booten keine Chance haben einen Kernel auszuwählen. Nach dem Editieren unbedingt „lilo -v“ ausführen.



```
musega:/usr/src# lilo -v
LILO version 22.6.1, Copyright (C) 1992-1998 Werner Almesberger
Development beyond version 21 Copyright (C) 1999-2004 John Coffman
Released 17-Nov-2004, and compiled at 20:30:43 on Dec 9 2004
Debian GNU/Linux

Reading boot sector from /dev/sda
Using MENU secondary loader
Calling map_insert_data

Boot image: /vmlinuz -> boot/vmlinuz-2.6.8-mppe-ipt-04
Mapping RAM disk /initrd.img -> boot/initrd.img-2.6.8-mppe-ipt-04
Added Linux *

Boot image: /vmlinuz.old -> boot/vmlinuz-2.6.8-grsec-mppe-ipt-03
Mapping RAM disk /initrd.img.old -> boot/initrd.img-2.6.8-grsec-mppe-ipt-03
Added LinuxOLD

Writing boot sector.
/boot/boot.0800 exists - no boot sector backup copy made.
```

Nun muss nur noch iptables neu kompiliert werden. Dafür gehen Sie wie folgt vor:

```
musega:/usr/src/linux# apt-get install devscripts
musega:/usr/src/linux# apt-get install html2text
musega:/usr/src/linux# apt-get install linuxdoc-tools
musega:/usr/src/linux# apt-get install debhelper debconf-utils
```

Diese Tools werden gebraucht, um ein solches Debian Package, wie wir es machen wollen herzustellen.

Patch-o-matic hat die iptables Sourcen schon korrekt angepasst. Wechseln sie also ins iptables Verzeichnis und führen Sie folgende Befehle aus:

```
musega:/usr/src/iptables/debian# debchange -i
musega:/usr/src/iptables# debian/rules binary
```

Das entstandene Softwarepackage installieren Sie nun wie folgt:

```
musega:/usr/src# dpkg -i iptables_1.2.11-9_i386.deb
```

Das war es soweit. Hat alles sauber geklappt können Sie ihr System nun neu starten.

Überprüfen Sie nun mit folgenden Befehlen ob sie jetzt überall über die richtige Softwareversion verfügen:



```
musega:~# uname -a
Linux musega 2.6.8-mppe-ipt-04 #1 SMP Fri Dec 24 11:28:17 CET 2004 i686
GNU/Linux

musega:~# iptables -V
iptables v1.2.11

musega:~# which pppd
/usr/local/sbin/pppd
musega:~# pppd --help
pppd version 2.4.3

/usr/local/sbin/pptpd
musega:~# pptpd -v
Poptop v1.2.1
```

7 Benötigte Software

7.1 Apache

MuSeGa ist komplett in Perl geschrieben und verwendet für die Benutzerschnittstellen dynamische generierte HTML Seiten. Um diese Seiten darzustellen wird der Apache Webserver mit CGI verwendet.

7.1.1 Installation

Um eine sichere Verbindung vom Client zum Webserver zu garantieren, verwenden wir die mit TLS gesicherte Version von Apache. Die Installation starten Sie wie folgt:

```
musega:~# apt-get install apache-ssl
```

Danach haben Sie bereits ein lauffähige Apache Installation auf ihrem System.

7.1.2 Konfiguration

Die Konfigurationsdatei `/etc/apache-ssl/httpd.conf` muss nun ziemlich stark angepasst werden. In folgendem Ausschnitt werden nur die Differenzen zur Standardkonfiguration angezeigt:

```
Timeout 10
KeepAlive Off
KeepAliveTimeout 10
ServerAdmin admin@hti.bfh.ch
```



```
ServerName musega.hti.bfh.ch

DocumentRoot /var/www/musega/user
<Directory /var/www/musega/user/>
    Options Indexes Includes FollowSymLinks MultiViews
    AllowOverride None
    Order allow,deny
    Allow from all
</Directory>

ServerSignature Off

ServerTokens Prod

#Alias /icons/ /usr/share/apache/icons/

#ScriptAlias /cgi-bin/ /usr/lib/cgi-bin/

<Directory /var/www/musega/admin/>
    AllowOverride None
    Options ExecCGI FollowSymLinks
    Order allow,deny
    Allow from all
    # Protect directory
    AuthType Basic
    AuthName "MuSeGa Admin Interface (unauthorized access prohibited!)"
    AuthUserFile /etc/musega/htpasswd
    Require user admin root
</Directory>

<Directory /var/www/musega/user/>
    AllowOverride None
    Options ExecCGI
    Order allow,deny
    Allow from all
</Directory>

Alias /images/ /var/www/musega/images/

<Directory /var/www/musega/images>
    AllowOverride None
    Order allow,deny
    Allow from all
</Directory>

# If the perl module is installed, this will be enabled.
#<IfModule mod_perl.c>
# Alias /perl/ /var/www/perl/
# <Location /perl>
#     SetHandler perl-script
#     PerlHandler Apache::Registry
#     Options +ExecCGI
# </Location>
#</IfModule>

#Alias /doc/ /usr/share/doc/

#<Location /doc>
# order deny,allow
# deny from all
# allow from 127.0.0.0/255.0.0.0
# Options Indexes FollowSymLinks MultiViews
#</Location>

Listen 80
Listen 443
Listen 10000

<VirtualHost *:80>
    SSLDisable
    DocumentRoot /var/www/musega/user
    #ServerName musega.hti.bfh.ch
    ServerName 10.10.0.1

    RewriteEngine on
    RewriteCond %{REQUEST_METHOD} ^ (TRACE|TRACK)
```



```

RewriteRule .* [F]
RewriteCond %{SERVER_PORT} 80$
RewriteCond %{REQUEST_URI} !/login\.pl.*
RewriteCond %{REQUEST_URI} !/favicon\.ico
RewriteCond %{REQUEST_URI} !/images/.
#RewriteRule ^/(.*) http://%{SERVER_NAME}/login.pl?_requested_url=http://%{HTTP_HOST}/$1 [R=307,L]
RewriteRule ^/(.*) https://%{SERVER_NAME}/login.pl?_requested_url=http://%{HTTP_HOST}/$1 [R=307,L]
</VirtualHost>
<VirtualHost *:443>
DocumentRoot /var/www/musega/user
#ServerName musega.hti.bfh.ch
ServerName 10.10.0.1
SSLCertificateFile /etc/apache-ssl/apache.pem
#ErrorDocument 404 /login.pl

RewriteEngine on
RewriteCond %{REQUEST_METHOD} ^(TRACE|TRACK)
RewriteRule .* [F]
RewriteCond %{SERVER_PORT} 443$
RewriteCond %{REQUEST_URI} !/login\.pl.*
RewriteCond %{REQUEST_URI} !/favicon\.ico
RewriteCond %{REQUEST_URI} !/images/.
RewriteRule ^/(.*) https://%{SERVER_NAME}/login.pl?_requested_url=https://%{HTTP_HOST}/$1 [R=307,L]
</VirtualHost>
<VirtualHost *:10000>
DocumentRoot /var/www/musega/admin
ServerName musega.hti.bfh.ch
SSLCertificateFile /etc/apache-ssl/apache.pem
#RewriteEngine on
#RewriteCond %{REQUEST_METHOD} ^(TRACE|TRACK)
#RewriteRule .* [F]
</VirtualHost>
```

Diese Anpassungen sind nötig damit Apache auf Port 80, 443 sowie 10000 auf eingehende Verbindungen hört und jeweils die richtigen Seiten aufruft.

Aus Sicherheitsgründen sollten Sie alle Standard-Aliases (icons, doc, etc.) aus der Konfiguration löschen, da diese einem Angreifen bereits erste Angriffspunkte bieten.

7.2 PPTPD

Den pptp Daemon haben wir ja bereits installiert, nun müssen Sie ihn noch konfigurieren:

7.2.1 Konfiguration

Sie müssen zwei Konfigurationsdateien anpassen.

`/etc/pptpd.conf`

```
ppp /usr/local/sbin/pppd
option /etc/ppp/pptpd-options
debug
logwtmp
localip 192.168.112.5
remoteip 192.168.112.150-210
```

Hier müssen Sie die beiden Parameter localip und remoteip noch ihrem Netzwerk anpassen. Bei localip nehmen Sie die Adresse von eth0, bei remoteip müssen Sie einen nicht benutzten Adressbereich aus ihrem internen Netzwerk nehmen.



In der Datei pppd-options müssen Sie lediglich die DNS Server (ms-dns) anpassen. Alle anderen Optionen sollten Sie so übernehmen.

`/etc/ppp/pptpd-options`

```
name musega
domain hta-be.bfh.ch
refuse-pap
refuse-chap
refuse-mschap
require-mschap-v2
mppc
mppe required,stateless
auth
-pap
-chap
-mschap
+mschap-v2
mtu 1450
mru 1450
lcp-echo-failure 30
lcp-echo-interval 5
ipcp-accept-local
ipcp-accept-remote
ms-dns 192.168.112.112
ms-dns 192.168.101.101
nodefaultroute
nobsdcomp
plugin radius.so
plugin radattr.so
```

Damit der pppd nun bei jedem Systemstart gestartet wird, brauchen Sie noch ein Startup-Script:

```
#!/bin/sh
# Copyright Rene Mayrhofer, Gibraltar, 1999
# This script is distributed under the GPL

PATH=/bin:/usr/bin:/sbin:/usr/sbin
DAEMON=/usr/local/sbin/pptpd
ARGS="-l 10.10.0.1"
PIDFILE=/var/run/pptpd.pid
FLAGS="defaults 50"

case "$1" in
  start)
    echo -n "Starting PPTP Daemon: "
    start-stop-daemon --start --quiet --pidfile $PIDFILE --exec $DAEMON \
      -- $ARGS < /dev/null > /dev/null
    echo "pptpd."
    ;;
  stop)
    echo -n "Stopping PPTP: "
    start-stop-daemon --stop --quiet --pidfile $PIDFILE --exec $DAEMON
    echo "pptpd."
    ;;
  force-reload|restart)
```



```
echo "Restarting PPTP: "  
sh $0 stop  
sh $0 start  
;;  
status)  
  if [ ! -r $PIDFILE ]; then  
    # no pid file, process doesn't seem to be running correctly  
    exit 3  
  fi  
  PID=`cat $PIDFILE | sed 's/ //g`  
  EXE=/proc/$PID/exe  
  if [ -x "$EXE" ] &&  
    [ "`ls -l \"$EXE\" | cut -d'>' -f2,2 | cut -d' ' -f2,2`" = \  
      "$DAEMON" ]; then  
    # ok, process seems to be running  
    exit 0  
  elif [ -r $PIDFILE ]; then  
    # process not running, but pidfile exists  
    exit 1  
  else  
    # no lock file to check for, so simply return the stopped status  
    exit 3  
  fi  
  ;;  
*)  
  echo "Usage: /etc/init.d/pptpd {start|stop|restart|force-reload}"  
  exit 1  
  ;;  
esac  
exit 0
```

Mit einem Link im Runlevel 2 sorgen Sie dafür, das dieses Script auch bei jedem Systemstart aufgerufen wird:

```
musega:/etc/rc2.d# ln -s ../init.d/pptpd ./S20pptpd
```

7.3 RADIUS-Client

Das RADIUS-Plugin des ppp Daemons, setzt einen installierten RADIUS-Client voraus. Sie müssen also folgende Pakete installieren:

```
musega:~# apt-get install libradius1 radiusclient1
```

In dieser Datei müssen Sie dann das Shared-Secret ihres RADIUS-Servers konfigurieren:

/etc/radiusclient/servers

```
192.168.200.100 seeehrschwierig
```

Die Dictionary Datei muss anschliessend noch mit der von MuSeGa mitgelieferten ersetzt werden: **/etc/radiusclient/dictionary**
Dies ist nötig da im Standard-Dictionary nicht alle verwendeten Attribute aufgeführt



sind.

7.4 MySQL

MuSeGa verwendet für die Speicherung verschiedener Daten eine SQL-Datenbank. MySQL installieren Sie mit folgendem Befehl:

```
musega:~# apt-get install mysql-common mysql-server
```

7.4.1 phpmyadmin

phpmyadmin ist ein Web-GUI für MySQL Datenbanken. Es ist schnell installiert und einfach zu bedienen. Zudem lassen sich Imports sowie Exports sehr bequem durchführen. Folgende Zeile installiert phpmyadmin:

```
musega:~# apt-get install phpmyadmin
```

Damit es später vom Administrations-Interface aus erreichbar ist, müssen Sie später noch folgenden Link hinzufügen:

```
musega:/var/www/musega/admin# ln -s /usr/share/phpmyadmin phpmyadmin
```

7.5 Perl Module

MuSeGa ist komplett in Perl realisiert. Einige verwendete Module müssen noch nachinstalliert werden:

```
musega:~# apt-get install  
libauthen-radius-perl  
libdbd-mysql-perl  
libdbi-perl  
libipc-shareable-perl  
libipc-sharelite-perl  
liblocale-gettext-perl  
libnet-daemon-perl  
libnet-pcap-perl  
libplrpc-perl  
librrdp-perl  
librrds-perl  
perl-suid
```

7.6 phpsysinfo

Um phpsysinfo zu installieren geben Sie folgenden Befehl ein:

Benötigte Software



```
musega:~# apt-get install phpsysinfo
```

Anschliessend müssen Sie noch einen Link erstellen:

```
musega:/var/www/musega/admin# ln -s /usr/share/phpsysinfo phpsysinfo
```

7.7 tcpdump

Tcpdump wird als Input für den Statistik-Daemon benutzt. Um es zu installieren geben Sie folgenden Befehl ein:

```
musega:~# apt-get install tcpdump
```

7.8 lynx

Lynx ist ein Textbrowser und er wird wie folgt installiert:

```
musega:~# apt-get install lynx-ssl
```

7.9 dhcpd

Damit MuSeGa den Clients IP-Adressen zuteilen kann, müssen Sie den DHCP-Server installieren:

```
musega:~# apt-get install dhcp
```

Anschliessend müssen Sie noch das Interface anpassen, auf dem der Daemon IP-Adressen vergeben soll. Tragen Sie also in der Datei /etc/init.d/dhcp das Interface eth1 ein. Dies gewährleistet das MuSeGa nur auf dem WLAN-Interface Adressen zuteilt.

8 MuSeGa Software

8.1 Installation

Um nun die MuSeGa Software selbst zu installieren, begeben Sie sich ins Rootverzeichnis ihres Servers und entpacken Sie die Datei musega-soft-0.9.tgz.

Stoppen Sie aber ihren MySQL Server bevor Sie das tun.

```
Musega:/# tar -xvzf /root/musega-soft-0.9.tgz
```

Alle Dateien sollten sich jetzt bereits am richtigen Platz befinden. Einzig die Dateisystemberechtigungen müssen Sie jetzt noch überprüfen und gegebenenfalls anpassen.



8.2 Berechtigungen Überprüfen

Es folgt eine Liste von allen Verzeichnissen mit ihren Berechtigungen. Spezialfälle sind gelb hervorgehoben:

/var/www/musega

```
drwxr-xr-x 4 root root 4096 Jan 8 21:33 admin
drwxr-xr-x 2 root root 4096 Jan 2 23:42 images
drwxr-xr-x 3 root root 4096 Jan 5 22:41 lib
drwxr-xr-x 2 root root 4096 Jan 4 22:06 user
```

/var/www/musega/admin

```
-rwsr-xr-x 1 root root 53139 Jan 6 22:44 admin.pl
-rwxr--r-- 1 root root 3128 Nov 19 00:05 favicon.ico
-rw-r--r-- 1 root root 13 Nov 13 13:27 index.html
-rwxr-xr-x 1 root root 12053 Dec 31 15:09 musega_logind.pl
-rwxr-xr-x 1 root root 20912 Jan 4 23:14 musega_rrdd.pl
-rwxr-xr-x 1 root root 6134 Jan 4 23:24 musega_statsd.pl
-rwxr-xr-x 1 root root 11063 Jan 4 23:09 musegactrl.pl
lrwxrwxrwx 1 root root 21 Nov 14 19:28 phpmyadmin -> /usr/share/phpmyadmin
lrwxrwxrwx 1 root root 21 Jan 1 23:22 phpsysinfo -> /usr/share/phpsysinfo
drwxr-xr-x 3 root root 4096 Jan 5 22:36 rrd
drwxrwsr-x 2 root www-data 4096 Jan 6 22:34 stats
-rwxr-xr-x 1 root root 6844 Dec 27 13:17 switchmobile_acl_gen.pl
```

/var/www/musega/lib

```
-rw-r--r-- 1 root root 30435 Jan 7 21:10 musega_admin.pm
-rw-r--r-- 1 root root 237637 Jan 5 22:26 musega_db.pm
-rw-r--r-- 1 root root 5737 Dec 8 22:11 musega_db_access.pm
-rw-r--r-- 1 root root 2630 Dec 30 18:17 musega_log.pm
-rw-r--r-- 1 root root 30222 Jan 5 22:57 musega_login.pm
-rw-r--r-- 1 root root 2190 Oct 31 23:57 musega_properties.pm
-rw-r--r-- 1 root root 10958 Oct 31 19:09 musega_rfc_data.pm
-rw-r--r-- 1 root root 65116 Dec 30 18:18 musega_rule_compiler.pm
-rw-r--r-- 1 root root 9291 Jan 4 01:35 musega_stats.pm
-rw-r--r-- 1 root root 6078 Jan 5 18:38 musega_system_commands.pm
```

/var/www/musega/user

```
-rwxr--r-- 1 root root 3128 Nov 29 23:41 favicon.ico
-rw-r--r-- 1 root root 12 Nov 13 13:27 index.html
-rwxr-xr-x 1 root root 8622 Jan 4 22:04 login.pl
```

/var/lib

```
drwxr-x--- 2 www-data root 4096 Jan 6 22:17 musega
```

/var/lib/mysql/musega

```
drwx----- 2 mysql mysql 4096 Jan 5 20:37 .
drwxr-xr-x 5 mysql mysql 4096 Jan 8 15:54 ..
```



```

-rw-rw---- 1 mysql mysql      348 Dec 26 15:52 conf_dhcpd_range.MYD
-rw-rw---- 1 mysql mysql     2048 Dec 26 23:43 conf_dhcpd_range.MYI
-rw-rw---- 1 mysql mysql     9034 Nov 19 00:11 conf_dhcpd_range.frm
-rw-rw---- 1 mysql mysql       52 Jan  6 12:16 conf_dhcpd_static.MYD
-rw-rw---- 1 mysql mysql     2048 Jan  7 00:09 conf_dhcpd_static.MYI
-rw-rw---- 1 mysql mysql     8702 Nov 19 00:12 conf_dhcpd_static.frm
-rw-rw---- 1 mysql mysql      436 Dec 26 15:53 conf_interfaces.MYD
-rw-rw---- 1 mysql mysql     2048 Dec 26 23:43 conf_interfaces.MYI
-rw-rw---- 1 mysql mysql     8924 Nov 19 00:12 conf_interfaces.frm
-rw-rw---- 1 mysql mysql    12456 Jan  6 17:48 conf_iptables.MYD
-rw-rw---- 1 mysql mysql     2048 Jan  7 00:09 conf_iptables.MYI
-rw-rw---- 1 mysql mysql     9458 Jan  5 20:37 conf_iptables.frm
-rw-rw---- 1 mysql mysql       36 Dec 26 15:54 conf_radius.MYD
-rw-rw---- 1 mysql mysql     2048 Dec 26 23:43 conf_radius.MYI
-rw-rw---- 1 mysql mysql     8636 Nov 19 00:12 conf_radius.frm
-rw-rw---- 1 mysql mysql       0 Jan  7 21:04 dhcp_leases.MYD
-rw-rw---- 1 mysql mysql     1024 Jan  7 21:04 dhcp_leases.MYI
-rw-rw---- 1 mysql mysql     8692 Nov 19 00:12 dhcp_leases.frm
-rw-rw---- 1 mysql mysql  2725620 Jan  6 22:05 dubiously_clients.MYD
-rw-rw---- 1 mysql mysql     2048 Jan  7 00:09 dubiously_clients.MYI
-rw-rw---- 1 mysql mysql     8682 Jan  3 12:06 dubiously_clients.frm
-rw-rw---- 1 mysql mysql       0 Jan  7 21:04 fw_active_redirect.MYD
-rw-rw---- 1 mysql mysql     1024 Jan  7 21:04 fw_active_redirect.MYI
-rw-rw---- 1 mysql mysql     8652 Dec  8 20:16 fw_active_redirect.frm
-rw-rw---- 1 mysql mysql       0 Jan  6 21:16 fw_compiled_mangle_rule.MYD
-rw-rw---- 1 mysql mysql     1024 Jan  6 21:16 fw_compiled_mangle_rule.MYI
-rw-rw---- 1 mysql mysql     8720 Dec 29 17:50 fw_compiled_mangle_rule.frm
-rw-rw---- 1 mysql mysql     1072 Jan  6 18:34 fw_compiled_nat_rule.MYD
-rw-rw---- 1 mysql mysql     2048 Jan  7 00:09 fw_compiled_nat_rule.MYI
-rw-rw---- 1 mysql mysql     8720 Nov 19 00:12 fw_compiled_nat_rule.frm
-rw-rw---- 1 mysql mysql     5900 Jan  6 16:46 fw_compiled_rule.MYD
-rw-rw---- 1 mysql mysql     2048 Jan  7 00:09 fw_compiled_rule.MYI
-rw-rw---- 1 mysql mysql     8754 Nov 19 00:14 fw_compiled_rule.frm
-rw-rw---- 1 mysql mysql       0 Jan  7 21:04 fw_dynamic_active_rule.MYD
-rw-rw---- 1 mysql mysql     1024 Jan  7 21:04 fw_dynamic_active_rule.MYI
-rw-rw---- 1 mysql mysql     8632 Nov 19 00:14 fw_dynamic_active_rule.frm
-rw-rw---- 1 mysql mysql      140 Jan  6 21:16 fw_mangle.MYD
-rw-rw---- 1 mysql mysql     2048 Jan  7 00:09 fw_mangle.MYI
-rw-rw---- 1 mysql mysql     8806 Dec 29 12:00 fw_mangle.frm
-rw-rw---- 1 mysql mysql      308 Dec 26 22:58 fw_nat_rule.MYD
-rw-rw---- 1 mysql mysql     2048 Dec 26 23:43 fw_nat_rule.MYI
-rw-rw---- 1 mysql mysql     8958 Nov 19 00:14 fw_nat_rule.frm
-rw-rw---- 1 mysql mysql      84 Nov 19 00:14 fw_object_addressranges.MYD
-rw-rw---- 1 mysql mysql     2048 Nov 19 00:45 fw_object_addressranges.MYI
-rw-rw---- 1 mysql mysql     8694 Nov 19 00:14 fw_object_addressranges.frm
-rw-rw---- 1 mysql mysql      204 Dec 28 22:45 fw_object_groups.MYD
-rw-rw---- 1 mysql mysql     2048 Dec 28 23:16 fw_object_groups.MYI
-rw-rw---- 1 mysql mysql     8640 Nov 19 00:14 fw_object_groups.frm
-rw-rw---- 1 mysql mysql      712 Dec 29 11:43 fw_object_hosts.MYD
-rw-rw---- 1 mysql mysql     2048 Dec 30 00:07 fw_object_hosts.MYI
-rw-rw---- 1 mysql mysql     8670 Nov 19 00:14 fw_object_hosts.frm
-rw-rw---- 1 mysql mysql      252 Jan  6 13:42 fw_object_networks.MYD
-rw-rw---- 1 mysql mysql     2048 Jan  7 00:09 fw_object_networks.MYI
-rw-rw---- 1 mysql mysql     8672 Nov 19 00:14 fw_object_networks.frm
-rw-rw---- 1 mysql mysql     2076 Jan  6 16:02 fw_rule.MYD
-rw-rw---- 1 mysql mysql     2048 Jan  7 00:09 fw_rule.MYI
-rw-rw---- 1 mysql mysql     8896 Nov 19 00:14 fw_rule.frm
-rw-rw---- 1 mysql mysql      80 Nov 19 00:14 fw_service_groups.MYD
-rw-rw---- 1 mysql mysql     2048 Nov 19 00:45 fw_service_groups.MYI
-rw-rw---- 1 mysql mysql     8640 Nov 19 00:14 fw_service_groups.frm
-rw-rw---- 1 mysql mysql      104 Nov 19 00:14 fw_service_icmp.MYD
-rw-rw---- 1 mysql mysql     2048 Nov 19 00:45 fw_service_icmp.MYI
-rw-rw---- 1 mysql mysql     8728 Nov 19 00:14 fw_service_icmp.frm
-rw-rw---- 1 mysql mysql      596 Dec 11 00:27 fw_service_ip.MYD
-rw-rw---- 1 mysql mysql     2048 Dec 11 00:48 fw_service_ip.MYI
-rw-rw---- 1 mysql mysql     8672 Nov 19 00:14 fw_service_ip.frm
    
```



```
-rw-rw---- 1 mysql mysql      312 Dec 29 11:43 fw_service_tcp.MYD
-rw-rw---- 1 mysql mysql     2048 Dec 30 00:07 fw_service_tcp.MYI
-rw-rw---- 1 mysql mysql     9304 Nov 19 00:14 fw_service_tcp.frm
-rw-rw---- 1 mysql mysql      184 Dec  8 16:09 fw_service_udp.MYD
-rw-rw---- 1 mysql mysql     2048 Dec  9 00:04 fw_service_udp.MYI
-rw-rw---- 1 mysql mysql     8824 Nov 19 00:14 fw_service_udp.frm
-rw-rw---- 1 mysql mysql      104 Nov 19 00:14 fw_time_timerange.MYD
-rw-rw---- 1 mysql mysql     2048 Nov 19 00:45 fw_time_timerange.MYI
-rw-rw---- 1 mysql mysql     9172 Nov 19 00:14 fw_time_timerange.frm
-rw-rw---- 1 mysql mysql      372 Dec 30 14:48 fw_traffic_queue.MYD
-rw-rw---- 1 mysql mysql     2048 Dec 31 00:23 fw_traffic_queue.MYI
-rw-rw---- 1 mysql mysql     8798 Dec 28 17:42 fw_traffic_queue.frm
-rw-rw---- 1 mysql mysql      196 Jan  5 12:13 group.MYD
-rw-rw---- 1 mysql mysql     2048 Jan  5 23:23 group.MYI
-rw-rw---- 1 mysql mysql     8680 Jan  5 12:12 group.frm
-rw-rw---- 1 mysql mysql      96 Jan  4 20:54 mac_blacklist.MYD
-rw-rw---- 1 mysql mysql     2048 Jan  5 00:07 mac_blacklist.MYI
-rw-rw---- 1 mysql mysql     8656 Jan  1 18:17 mac_blacklist.frm
-rw-rw---- 1 mysql mysql      156 Jan  6 22:11 mac_user_history.MYD
-rw-rw---- 1 mysql mysql     2048 Jan  7 00:09 mac_user_history.MYI
-rw-rw---- 1 mysql mysql     8754 Dec 31 17:55 mac_user_history.frm
-rw-rw---- 1 mysql mysql     2588 Jan  5 17:04 switchmobile_acls.MYD
-rw-rw---- 1 mysql mysql     2048 Jan  5 23:23 switchmobile_acls.MYI
-rw-rw---- 1 mysql mysql     8710 Dec 26 16:27 switchmobile_acls.frm
-rw-rw---- 1 mysql mysql    124164 Jan  6 22:13 traffic_stats_services.MYD
-rw-rw---- 1 mysql mysql    22528 Jan  7 00:09 traffic_stats_services.MYI
-rw-rw---- 1 mysql mysql     8984 Jan  3 20:27 traffic_stats_services.frm
-rw-rw---- 1 mysql mysql      40 Jan  6 22:13 traffic_throughput.MYD
-rw-rw---- 1 mysql mysql     1024 Jan  7 00:09 traffic_throughput.MYI
-rw-rw---- 1 mysql mysql     8752 Jan  3 20:02 traffic_throughput.frm
-rw-rw---- 1 mysql mysql     1864 Jan  6 22:17 user.MYD
-rw-rw---- 1 mysql mysql     2048 Jan  7 00:09 user.MYI
-rw-rw---- 1 mysql mysql     9318 Jan  4 20:49 user.frm
```

Sind alle Berechtigungen richtig, ist die Installation abgeschlossen.

8.3 MuSeGa Starten

Wechseln Sie nun ins Verzeichnis `/var/www/musega/admin` und geben sie folgenden Befehl ein:

```
musega:/var/www/musega/admin# ./musegactrl.pl start
Starting MuSeGa Engine..
musega:/var/www/musega/admin# rrd daemon started.. (pid 1386)
stats daemon started.. (pid 1387)
login daemon started.. (pid 1391)
```

Ist anstelle des oben gezeigten Textes eine Fehlermeldung erschienen, liegt das wahrscheinlich daran, dass entweder eine Berechtigung noch nicht stimmt oder irgend ein Modul oder ein Softwarepaket fehlt. Um sicher zu gehen dass Sie kein Paket vergessen haben, können Sie ein `apt-get install --simulate` mit der Softwareliste des Referenzsystems machen (siehe nächster Punkt).



9 Installierte Pakete überprüfen

Wollen Sie überprüfen ob Sie alle Pakete installiert haben die MuSeGa braucht, Verwenden Sie den Befehl *apt-get install --simulate* mit folgender Liste:

```
musega:~#apt-get install --simulate adduser apache-common apache-ssl apache-utils
apt apt-utils aptitude ash at base-config base-files base-passwd bash binutils
bsdmainutils bsduutils bzip2 console-common console-data console-tools coreutils cpio
cpp cpp-2.95 cpp-3.3 cramfsprogs cron dash debconf debconf-english debconf-utils
debhelper debianutils defoma devscripts dhcp dhcp-client dialog diff dpkg dpkg-dev
dselect e2fslibs e2fsprogs emacs20 emacsen-common ethereal-common exim fdutils file
fileutils findutils fontconfig gcc gcc-2.95 gcc-3.3 gcc-3.3-base gettext gettext-
base gettext-el gradm2 grep grep-dctrl groff-base gzip host hostname html2text
ifupdown info initrd-tools initscripts intltool-debian ipchains iproute iptables
kernel-source-2.6.8 klogd less libacl1 libadns1 libapache-mod-php4 libatml libatml-
dev libattr1 libauthen-radius-perl libblkid1 libbz2-1.0 libc6 libc6-dev libcap1
libcomerr2 libconsole libdb1-compat libdb2 libdb3 libdb3-util libdb4.2 libdbd-mysql-
perl libdbi-perl libdevmapper1.00 libexpat1 libfontconfig1 libfreetype6 libgcc1
libgcrypt11 libgd2-noxpm libgdbm3 libgdbmg1 libgl1.2 libgl1.2-0 libgnutls11
libgpg-error0 libice6 libident libipc-shareable-perl libipc-sharelite-perl libjpeg62
libldap2 liblocale-gettext-perl liblockfile1 liblz1 libmagic1 libmm11
libmysqlclient10 libmysqlclient12 libncurses5 libncurses5-dev libncursesw5 libnet-
daemon-perl libnet-pcap-perl libnewt0 libopencdk8 libpam-modules libpam-runtime
libpam0g libpam0g-dev libpcap0 libpcap0.7 libpcap0.7-dev libpcap0.8 libpcre3
libperl5.8 libplrpc-perl libpng12-0 libpopt0 libradius1 libreadline4 librrd0
librrdp-perl librrds-perl libsas12 libsas17 libsigc++-1.2-5c102 libsm6 libsp1 libss2
libssl-dev libssl0.9.6 libssl0.9.7 libstdc++2.10-glibc2.2 libstdc++5 libtasn1-2
libuuid1 libwrap0 libx11-6 libxaw7 libxext6 libxft1 libxi6 libxmu6 libxmuu1 libxp6
libxpm4 libxrandr2 libxrender1 libxt6 libxtrap6 libxtst6 lilo linux-kernel-headers
linuxdoc-tools linuxlogo login logrotate lynx-ssl mailx make makedev man-db manpages
mawk mbr mime-support modconf module-init-tools modutils mount mysql-client mysql-
common mysql-server ncurses-base ncurses-bin net-tools netbase netkit-inetd netkit-
ping nload ntpdate nvi openssl passwd patch pciutils perl perl-base perl-modules
perl-suid php4 php4-common php4-mysql phpmyadmin phpsysinfo po-debconf procs psmisc
radiusclient1 rrdtool sed setserial sgml-base sgml-data shellutils slang1 slangla-
utf8 sp ssh ssl-cert strace sysklogd syslinux sysv-rc sysvinit tar tasksel tcpd
tcpdump telnet tethereal textutils ttf-bitstream-vera ucf util-linux wget wwwconfig-
common xfree86-common xlibs xlibs-data xml-core zlib
```

10 Weiteres Vorgehen

Die Installation ist nun abgeschlossen. Konsultieren Sie das Betriebshandbuch für die weitere Konfiguration. Möchten Sie mehr technische Details wissen, lesen Sie die technische Dokumentation.

11 Glossar

Der Glossar wird aufgrund der einfacheren Erweiterung in einem separaten Dokument geführt. Er ist unter <http://musega.ch> zu finden.