

# Analyse

Version 25.09.2004

## Diplomarbeit I00 (2004)

# MuSeGa



## Mobile User Secure Gateway

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



# Abstract

Dieses Dokument dient als Grundlage für die Entwicklung von MuSeGa. Aus ihm resultiert das zu verwendende Betriebssystem sowie die zu verwendenden Softwarekomponenten

## Versionskontrolle

Version	Datum	Kommentar	Genehmigt
0.1	26.07.2004	Start	
0.2	28.07.2004		
0.3	29.07.2004		
0.4	02.08.2004	pptp debian Teil fertig	
0.5	12.08.2004	Erste vollständige Version	
0.6	25.09.2004	Kleine Ergänzungen, Anpassungen	Lu

## Inhaltsverzeichnis

1 Allgemeines.....	5
1.1 Sinn und Zweck.....	5
1.2 Leserkreis.....	5
2 In Frage kommende Betriebssysteme abklären.....	5
2.1 Linux.....	5
2.2 BSD.....	6
3 Betriebssysteme installieren.....	6
3.1 Debian stable (3.0).....	6
3.1.1 Update auf 3.1 „testing“.....	6
3.2 OpenBSD 3.5.....	6
4 Zu verwendende Softwarepakete abklären.....	7
4.1 Für Debian 3.1 Sarge.....	7
4.1.1 Firewall.....	7
4.1.2 RADIUS Client.....	7
4.1.3 VLAN.....	7
4.1.4 Admin via cmd und Web-GUI.....	7
4.1.5 NAT/NAPT.....	8
4.1.6 DHCP-Server.....	8
4.1.7 DNS-Forwarding.....	8
4.1.8 Statistik.....	8
4.1.9 PPTP Endpunkt.....	9
4.1.10 SMTP Catch and Forward.....	11
4.1.11 Bandbreiten Management.....	11
4.1.12 IDS.....	12
4.1.13 IPSec Endpunkt.....	12
4.2 Für OpenBSD 3.5.....	12
4.2.1 Firewall.....	12
4.2.2 RADIUS Client.....	13
4.2.3 VLAN.....	13
4.2.4 Admin via cmd und Web-GUI.....	13
4.2.5 NAT/NAPT.....	13
4.2.6 DHCP-Server.....	13
4.2.7 DNS-Forwarding.....	13
4.2.8 Statistik.....	14
4.2.9 PPTP Endpunkt.....	14
4.2.10 SMTP Catch and Forward.....	15
4.2.11 Bandbreiten Management.....	15
4.2.12 IDS.....	15
4.2.13 IPSec Endpunkt.....	15
5 Bestimmen des Betriebssystem.....	16
5.1 Gegenüberstellung Debian 3.1 – OpenBSD 3.5.....	16
5.2 Entscheidung für Debian 3.1 (Sarge).....	16
6 Weitere Komponenten.....	17
6.1 Datenbank.....	17

---

6.2 Datenbank Zugriff.....	17
7 Quellen.....	17
8 Anhang.....	18
8.1 Beschreibung der Linux Distributionen (Quelle DistroWatch.org).....	18
8.1.1 Mandrakelinux.....	18
8.1.2 Fedora.....	19
8.1.3 Debian.....	20
8.1.4 Gentoo.....	20
8.1.5 SUSE.....	21
8.1.6 Slackware.....	22
8.2 Beschreibung der BSD Varianten.....	23
8.2.1 FreeBSD.....	23
8.2.2 NetBSD.....	23
8.2.3 OpenBSD.....	23



## 1 Allgemeines

### 1.1 Sinn und Zweck

Anhand dieses Dokumentes soll entschieden werden, welches Betriebssystem für den Mobile User Secure Gateway am besten in Frage kommt. Es soll die Vor- und Nachteile der einzelnen Betriebssysteme, so wie deren Features aufzeigen. Weiter soll es Aufschluss darüber geben, für welche Funktionalität man am besten welche Software verwendet sowie in welchem Umfang man einzelnen Funktionalitäten implementieren kann.

### 1.2 Leserkreis

Dieses Dokument richtet sich an den Experten sowie an die Betreuer der Projektes. Um dieses Dokument verstehen zu können, sind gute Kenntnisse aus dem UNIX/Linux Bereich unabdingbar.

## 2 In Frage kommende Betriebssysteme abklären

Da MuSeGa als OpenSource Projekt entwickelt werden soll, fallen alle kommerziellen Betriebssysteme weg. Übrig bleiben noch diverse Linux Distributionen sowie die drei BSD Varianten.

### 2.1 Linux

Linux ist sicherlich der Favorit als Basis für den Mobile User Secure Gateway. Hier eine kleine Auswahl möglichen Distributionen:

- Mandrake
- Fedora
- debian
- gentoo
- SUSE
- Slackware

Genauere Beschreibungen der einzelnen Distributionen befinden sich im Anhang.

Da es zeitlich nicht möglich ist, alle Distributionen auszuprobieren, musste ich mich für eine entscheiden. Ich kenne alle oben aufgeführten Distributionen recht gut und aus folgenden Gründen habe ich mich für debian entschieden:

- Debian hat sich bei mir im Security- und Webservices-Bereich seit Jahren bewährt
- Es wird von sehr vielen Leuten eingesetzt und man bekommt darum fast bei jedem Problem Hilfe aus der Community.
- Das Softwarepaketverwaltungssystem ist vorbildlich und funktioniert sehr gut.
- Debian Systeme können dank APT sehr einfach auf dem neusten Stand gehalten werden. (APT = Debian Paketverwaltungssystem)
- Debian verfügt über sehr lange Releasezyklen. Dadurch muss man nicht alle paar



Monate das System neu installieren, was bei einem produktiv eingesetzten Gateway sehr wichtig ist.

- Am 15. September soll das nächste Release (Sarge, 3.1) erscheinen.

## 2.2 BSD

Auch die drei BSD Varianten kämen als Basis für MuSeGa durchaus in Frage. Dieses sind:

- FreeBSD
- NetBSD
- OpenBSD

Auch hier musste ich mich für eine BSD Variante entscheiden und aus folgenden Gründen habe ich mich für OpenBSD entschieden.

- OpenBSD gilt als eines der sichersten Betriebssysteme überhaupt.
- Ich kenne OpenBSD schon recht gut, verwende es als Router und Firewall.
- Es bringt von sich aus schon einige Sicherheitsfunktionen mit, die man bei anderen Systemen erst einbauen muss.

## 3 Betriebssysteme installieren

### 3.1 Debian stable (3.0)

Die Installation von Debian stable (3.0) verlief reibungslos und war schnell abgeschlossen. Auch die Updates sowie das neuste Kernelimage waren schnell installiert. Dies nicht zuletzt weil ich all dies früher schon zig mal gemacht hatte.

#### 3.1.1 Update auf 3.1 „testing“

Da Debian stable (3.0) schon über zwei Jahre (released am 19. Juli 2002) auf dem Buckel hat und die Version 3.1 laut debian.org schon am 15. September 2004 erscheinen soll, habe ich mich entschieden gleich die Version 3.1 zu verwenden. Diese ist zwar noch nicht offiziell freigegeben als stable, aber sie läuft dennoch sehr stabil.

Das Updaten gestaltet sich sehr einfach: apt.conf auf „testing“ anpassen und apt-get dist-upgrade ausführen.

### 3.2 OpenBSD 3.5

Auch die Installation von OpenBSD ging sehr schnell und ohne Probleme. Nach ca. 10 – 15 Minuten hat man ein fertig installiertes Betriebssystem. Einzig das Auspacken der Sourcen sowie der Pakete dauert anschliessend etwas länger. Da man bei OpenBSD die Security-Updates direkt in den Sourcecode einbringt, muss man anschliessend den Kernel neu kompilieren. Dies kann je nach Prozessor eine weile dauern.



## 4 Zu verwendende Softwarepakete abklären

### 4.1 Für Debian 3.1 Sarge

#### 4.1.1 Firewall

Unter Linux stellt sich eigentlich die Frage nach der Wahl der Firewallsoftware nicht. In der Kernelserie 2.4 gibt es einen integrierten Paketfilter mit dem Namen „iptables“. Dieser bietet eine grosse Fülle von Funktionen (NAT, NAPT) und ist eigentlich ein Standard unter Linux. Er wird auch in der Kernelserie 2.6 weitergeführt.

#### 4.1.2 RADIUS Client

Um vom Gateway aus mit dem RADIUS-Server zu sprechen, bietet sich die Verwendung eines RADIUS-Clients in Perl an. Ein Perl Modul mit dieser Funktionalität existiert bereits:

Authen:::Radius - provide simple Radius client facilities

<http://search.cpan.org/~manowar/RadiusPerl-0.11/Radius.pm>

Perl ist ein Bestandteil von Linux, daher muss sonst nichts installiert werden.

RADIUS-Attribute die mitgegeben werden können:

<http://www.freeradius.org/rfc/attributes.html>

RADIUS Authentication Library:

<http://www.helsinki.fi/~vviitane/hupnet/files/radauth.pm>

#### 4.1.3 VLAN

VLANs werden seit Kernel 2.4.14 unterstützt. Debian stable (3.0) verwendet Kernel 2.4.18, jedoch ist die VLAN Funktionalität nicht per default aktiviert. Also musste ich erstmal einen neuen Kernel mit VLAN Unterstützung kompilieren: CONFIG\_VLAN\_8021Q

```
musega:~# vconfig add eth0 2
Added VLAN with VID == 2 to IF -:eth0:-
```

Die VLAN Unterstützung gemäss 802.1Q ist also nun betriebsbereit.

Weitere Informationen zu VLAN unter Linux befinden sich unter folgendem Link:

VLAN on Linux howto:

<http://www.linuxjournal.com/article.php?sid=7268>

#### 4.1.4 Admin via cmd und Web-GUI

MuSeGa soll via GUI von überall her administriert werden können. Darum läuft dieses auf einem Webserver. Als Webserver empfiehlt sich hier der altbewährte Apache Version 1.3.x im SSL (TLS) Modus. Um nun das GUI auf dem Webserver abzubilden empfiehlt sich eine Script/Websprache. Hier ist auf den ersten Blick PHP geeignet. PHP ist aber eine reine Websprache und darum von der Kommandozeile her nicht

erreichbar. Perl hingegen ist eine sehr mächtige und bewährte Scriptsprache, die via CGI auch problemlos ein HTML GUI zur Verfügung stellen kann. Perl 5.6.1 ist Bestandteil von Debian stable (3.0). Falls diese Version nicht ausreichen sollte, kann man problemlos auf 5.8.x updaten (apt-get install perl/testing).

### 4.1.5 NAT/NAPT

Die NAT/NAPT Funktionalität bringt unter Linux bereits iptables mit sich. Unten ein kleines NAT-Beispiel:

```
$IPTABLES -t nat -A PREROUTING -d 212.254.178.231 -j DNAT --to-destination 192.168.101.101
```

Hier werden alle Pakete mit der Destination 212.254.178.231 zu der privaten (internen) Adresse 192.168.101.101 weitergeleitet.

Auch NAPT funktioniert einwandfrei:

```
$IPTABLES -t nat -A PREROUTING -p tcp -d 193.5.86.83 --destination-port 80 -j DNAT \
--to-destination 193.5.83.10:443
```

Hier werden zum Beispiel alle Pakete mit Destination 193.5.86.83 und dem Zielport 80 (http) zur Adresse 193.5.83.10 und dem Zielport 443 (https) weitergeleitet.

### 4.1.6 DHCP-Server

Unter Linux ist eigentlich seit langem der DHCP Server von ISC Standard, so auch unter Debain. Eine einfache Konfiguration sieht zum Beispiel so aus:

```
subnet 192.168.120.0 netmask 255.255.255.0 {
  range 192.168.120.10 192.168.120.240;
  option domain-name-servers ns2.luke.ch, ns1.luke.ch;
  option domain-name "luke.lab";
  option routers 192.168.120.1;
  option subnet-mask 255.255.255.0;
  option broadcast-address 192.168.120.255;
  default-lease-time 302400;
  max-lease-time 604800;
}
```

Weiter Informationen befinden sich auf der ISC DHCP Homepage:

<http://www.isc.org/index.pl?sw/dhcp/>

### 4.1.7 DNS-Forwarding

Das DNS-Forwarding kann unter Linux mit Hilfe von iptables gelöst werden. Folgende Regel leitet alle eingehenden UDP Pakete auf Port 53 vom Netzwerk 10.0.0.0/8 an einen internen DNS-Server weiter:

```
$IPTABLES -t nat -A PREROUTING -s 10.0.0.0/8 -p udp -d any -dport 53 -j DNAT -to-destination
ns1.demonet.ch
```

### 4.1.8 Statistik

Die Statistiken bestehen eigentlich alle aus Countern, welche an diversen Orten zusammengesucht werden. Welche Counter alle gesammelt werden, kann aus dem Design Dokument entnommen werden.

Um diese Counter grafisch darzustellen und auf längere Sicht ressourcensparend abzulegen, bietet sich das RRD Tool von Tobi Oetiker an:

<http://people.ee.ethz.ch/~oetiker/webtools/rrdtool/>



Der Datenbestand hat so eine fixe Grösse und man hat trotzdem Statistiken die über längere Zeit zurückreichen. RRD-Tool ist in Perl geschrieben und daher ein guter Kandidat für dieses Projekt.

### 4.1.9 PPTP Endpunkt

PoPToP scheint genau das zu sein was ich suche:

„*Poptop is the PPTP server solution for Linux. Ports also exist for Solaris 2.6, OpenBSD, FreeBSD and others.* „

PoPToP Homepage

<http://www.poptop.org/>

Der ppp Daemon von Debian 3.0 unterstützt aber noch kein MPPE. Also muss man dieses Paket aus dem testing Release installieren, was einige weitere Pakete nach sich zieht:

```
apt-get install ppp/testing
Sorry, but the following packages have unmet dependencies:
  ppp: Depends: libc6 (>= 2.3.2.ds1-4) but 2.2.5-11.5 is to be installed
        Depends: libpam0g (>= 0.76) but 0.72-35 is to be installed
        Depends: libpcap0.7 but it is not going to be installed
        Depends: libssl0.9.7 but it is not going to be installed
        Depends: libpam-runtime (>= 0.76-13.1) but 0.72-35 is to be installed
E: Sorry, broken packages
musega:~/soft# apt-get install ppp/testing libc6/testing libpam0g/testing libpcap0.7/testing
libssl0.9.7/testing libpam-runtime/testing
```

Debian Package: pptpd (1.2.1-1)

<http://packages.debian.org/testing/net/pptpd.html>

Kernel 2.6.7 mit MPPE Patch patchen und kompilieren. Folgende Links waren sehr nützlich dabei:

MPPE/MPPC kernel module for Linux

<http://www.polbox.com/h/hs001/>

PPTP Debian Howto

<http://pptpclient.sourceforge.net/howto-debian.phtml#mppe>

Nun die wichtigsten Konfigfiles:

/etc/pptpd.conf

```
option /etc/ppp/pptpd-options
localip 10.0.0.1
remoteip 10.20.0.10-100
```

/etc/ppp/pptpd-options

```
# Authentication

# Name of the local system for authentication purposes
# (must match the second field in /etc/ppp/chap-secrets entries)
name pptpd

# Usually it is not necessary to have the PPTP Server authenticate in this way.
```



```
noauth

# Encryption

# Debian: on systems with a kernel built with the package
# kernel-patch-mppe >= 2.4.2 and using ppp >= 2.4.2, ...
# {{{
refuse-pap
refuse-chap
refuse-mschap
# Require the peer to authenticate itself using MS-CHAPv2 [Microsoft
# Challenge Handshake Authentication Protocol, Version 2] authentication.
require-mschap-v2
# Require MPPE 128-bit encryption
# (note that MPPE requires the use of MSCHAP-V2 during authentication)
require-mppe-128
# }}}

# Network and Routing

ms-dns 192.168.112.112
ms-dns 192.168.101.200

# Add an entry to this system's ARP [Address Resolution Protocol]
# table with the IP address of the peer and the Ethernet address of this
# system. This will have the effect of making the peer appear to other
# systems to be on the local ethernet.
# (you do not need this if your PPTP server is responsible for routing
# packets to the clients -- James Cameron)
proxyarp

# Debian: do not replace the default route
nodefaultroute

# Logging

# Enable connection debugging facilities.
# (see your syslog configuration for where pppd sends to)
debug

# Miscellaneous

# Create a UUCP-style lock file for the pseudo-tty to ensure exclusive
# access.
lock

# Disable BSD-Compress compression
nobsdcomp
```

### /etc/ppp/chap.secrets

```
# Secrets for authentication using CHAP
# client      server  secret          IP addresses
test          *      password        *
```

### PPTP Client for UNIX/Linux

<http://pptpclient.sourceforge.net/>

### Diagnosis Howto

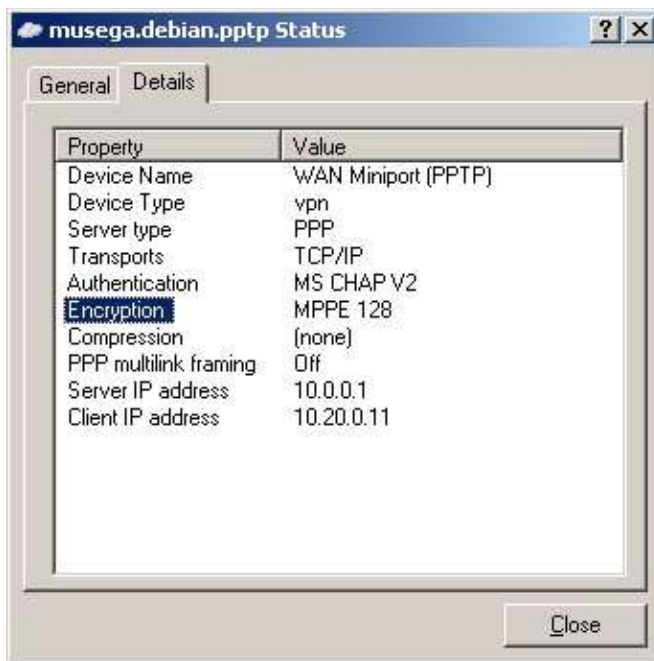
<http://pptpclient.sourceforge.net/howto-diagnosis.phtml>



Poptop Artikel von Ralf Spenneberg

[http://www.spenneberg.com/linux-magazin/037-039\\_pptp\\_neu.pdf](http://www.spenneberg.com/linux-magazin/037-039_pptp_neu.pdf)

Am Schluss scheint endlich alles zu funktionieren:



#### 4.1.10 SMTP Catch and Forward

Um allen ausgehenden Mailverkehr auf einen internen Mailgateway umzulenken, genügt folgendes iptables Kommando:

```
$IPTABLES -t nat -A PREROUTING -p tcp -s 10.0.0.0/8 --source-port 1024:65535 --destination-port 25 -j DNAT --to-destination mailgateway
```

Hier wird der Traffic aus dem Netz 10.0.0.0/8 mit Sourceport zwischen 1024 und 65535 sowie Destinationport 25 (SMTP) auf den Server mailgateway umgeleitet.

#### 4.1.11 Bandbreiten Management

Das Bandbreiten Management kann unter Linux mit Hilfe von Queues erfolgen. Mit Hilfe des „tc“ Kommandos vom Paket „iproute2“ werden also verschiedene Queues erstellt, denen dann der vorher mittels iptables markierte Traffic zugeordnet wird:

```
##### queues erstellen #####  
RATEUP=640  
  
## root queue  
/sbin/tc qdisc add dev eth0 root handle 1:0 htb default 13 r2q 1  
  
## queue mit höchster Priorität für ACKs  
/sbin/tc class add dev eth0 parent 1:1 classid 1:10 htb rate 64kbit ceil ${RATEUP}kbit prio 1
```



```
## telnet, ssh, vpn, all udp
/sbin/tc class add dev eth0 parent 1:1 classid 1:11 htb rate 158kbit ceil ${RATEUP}kbit prio 2

## http, https
/sbin/tc class add dev eth0 parent 1:1 classid 1:12 htb rate 128kbit ceil ${RATEUP}kbit prio 3

## normal traffic
/sbin/tc class add dev eth0 parent 1:1 classid 1:13 htb rate 128kbit ceil ${RATEUP}kbit prio 4

##### Traffic markieren #####

# small packets (probably just ACKs: TCP ACK Packets are 60Byte)
iptables -t mangle -A MYSHAPER-OUT -o eth0 -p tcp -m length --length :64 -j MARK --set-mark 10

# icmp
iptables -t mangle -A MYSHAPER-OUT -o eth0 -p icmp -j MARK --set-mark 10

# vpn, ipsec
iptables -t mangle -A MYSHAPER-OUT -o eth0 -p 50 -j MARK --set-mark 11

# telnet, ssh
iptables -t mangle -A MYSHAPER-OUT -o eth0 -p tcp --dport 22:23 -j MARK --set-mark 11

# all udp (online gaming..)
iptables -t mangle -A MYSHAPER-OUT -o eth0 -p udp -j MARK --set-mark 11

# http, https from webservers
iptables -t mangle -A MYSHAPER-OUT -o eth0 -s 192.168.101.0/24 -p tcp -m multiport --sport
80,443 -j MARK --set-mark 12

##### Traffic den queues zuweisen #####

/sbin/tc filter add dev eth0 parent 1:0 prio 0 protocol ip handle 10 fw flowid 1:10
/sbin/tc filter add dev eth0 parent 1:0 prio 0 protocol ip handle 11 fw flowid 1:11
/sbin/tc filter add dev eth0 parent 1:0 prio 0 protocol ip handle 12 fw flowid 1:12
/sbin/tc filter add dev eth0 parent 1:0 prio 0 protocol ip handle 13 fw flowid 1:13
/sbin/tc filter add dev eth0 parent 1:0 prio 0 protocol ip handle 14 fw flowid 1:14
/sbin/tc filter add dev eth0 parent 1:0 prio 0 protocol ip handle 15 fw flowid 1:15
```

### 4.1.12 IDS

Als IDS bietet sich Snort an: <http://www.snort.org>

Snort habe ich bereits in einer früheren Arbeit eingehend bearbeitet. Das dazugehörige Dokument befindet sich hier:

<http://i00-2.luke.ch/faecher/iin/files/praktikum/iin-gruppenarbeit-snort.pdf>

### 4.1.13 IPsec Endpunkt

Zum Thema IPsec mit Linux habe ich bereits vor kurzem eine Arbeit gemacht. Die Arbeit findet sich unter folgendem Link:

<http://i00-2.luke.ch/faecher/isec/files/ipsec-kernel-2.6.pdf>

## 4.2 Für OpenBSD 3.5

### 4.2.1 Firewall

OpenBSD ist das ideale Betriebssystem für einen Firewall. Seit der Version 3.0 verfügt OpenBSD über eine neue Paketfilter Implementation mit dem Namen pf. Ursprünglich wurde pf von Daniel Hartmeier (einem Schweizer) im Jahre 2001 entwickelt. Das



Produkt war so gut das es schon bald in den OpenBSD Sourcetree aufgenommen wurde. Heute ist pf unter OpenBSD ein Standard. Die Portierung nach NetBSD und FreeBSD ist auch schon fast abgeschlossen.

### 4.2.2 RADIUS Client

Da Perl ebenfalls problemlos unter OpenBSD läuft, kann man auch hier das Perlmodul als RADIUS-Client verwenden (siehe Punkt 4.1).

### 4.2.3 VLAN

OpenBSD 3.5 unterstützt VLANs out-of-the-box. Hier ein Auszug aus den Manual-Pages:

```
VLAN(4)                                OpenBSD Programmer's Manual          VLAN(4)
NAME
    vlan - IEEE 802.1Q encapsulation/decapsulation pseudo-device
SYNOPSIS
    pseudo-device vlan [count]
DESCRIPTION
    The vlan Ethernet interface allows construction of virtual LANs when used
    in conjunction with IEEE 802.1Q-compliant Ethernet devices.

    A vlan interface can be created at runtime using the ifconfig vlanN
    create command or by setting up a hostname.if(5) configuration file for
    netstart(8).
```

### 4.2.4 Admin via cmd und Web-GUI

Der Apache Webserver in der Version 1.3.x steht unter OpenBSD 3.5 ebenfalls zur Verfügung. Auch Perl ist in der Version 5.8.2 als Bestandteil der Basisinstallation vorhanden und somit der geeignete Kandidat.

### 4.2.5 NAT/NAPT

Auch in OpenBSD wird die NAT/NAPT Funktionalität bereits vom eigenen Paketfilter (pf) unterstützt:

```
binat on fxp0 from 192.168.1.66 to any -> 20.0.5.5
```

Hier wird zum Beispiel allen Verkehr auf die Adresse 20.0.5.5 an die interne Adresse 192.168.1.66 weitergeleitet. Umgekehrt wird der Verkehr der von 192.168.1.66 kommt mit der Absenderadresse 20.0.5.5 versehen.

### 4.2.6 DHCP-Server

Auch unter OpenBSD wird praktisch nur der DHCP Daemon von ISC eingesetzt. Die Konfiguration sieht genau gleich aus wie unter Linux.

### 4.2.7 DNS-Forwarding

Unter OpenBSD kann man die DNS-Forwarding Funktionalität mit Hilfe des Paketfilter (pf) sicherstellen:

```
rdr on fxp1 proto udp from 10.0.0.0/8 to $EXT_INT port 53 -> {172.16.1.10, 172.16.2.10}
```



In diesem Beispiel werden alle DNS Queries (UDP Port 53) im round-robin Verfahren an zwei interne DNS-Server weitergeleitet.

### 4.2.8 Statistik

Statistiken werden ebenfalls mit dem RRD-Tool implementiert. Siehe Punkt 4.1 ->

### 4.2.9 PPTP Endpunkt

PoPToP läuft auch unter OpenBSD. Siehe Punkt 4.1 ->

Die Installation gestaltet sich wie folgt:

```
cd /usr/ports/net/poptop
make install
```

Anschliessend müssen ein paar Konfigurationsdateien angepasst werden:

#### /etc/pptpd.conf

```
option /etc/ppp/options.pptpd
debug
localip 10.0.0.1
remoteip 10.20.0.100-200
```

#### /etc/ppp/options.pptpd

```
noauth
proxyarp
+MSChap-V2
mppe-128
mppe-stateless
```

#### /etc/ppp/ppp.conf

```
default:
  set log Phase Chat LCP IPCP CCP tun command
  set timeout 0
  set dial
  set login
  set mppe * stateless
  set device localhost:pptp

pptp:
  disable pap
  disable chap
  enable MSChapV2
  disable deflate predl
  deny deflate predl

  accept mppe
  enable proxy
  accept dns

  set dns 192.168.112.112 192.168.101.200

# Server (local) IP address, Range for Clients, and Netmask
# Use the same IP addresses you specified in /etc/pppd.conf :
set ifaddr 10.0.0.1 10.20.0.100-10.20.0.200 255.255.255.0
```

#### /etc/ppp/ppp.secret

```
# Authname Authkey Peer's IP address Label Callback
test test * * *
```

Den Kernel braucht man nicht neu zu übersetzen und auch der ppp Daemon von



OpenBSD unterstützt bereits MPPE.



#### 4.2.10 SMTP Catch and Forward

Um unter OpenBSD allen Mailverkehr vom Netz 10.0.0.0/8 auf einen internen Mailgateway umzuleiten, genügt folgende Zeile im pf.conf File:

```
rdp on fxpl proto tcp from 10.0.0.0/8 to any port 25 -> mailgateway port 25
```

#### 4.2.11 Bandbreiten Management

OpenBSD bietet ebenfalls Bandbreiten Management via Queues an. Dabei stehen verschiedene Arten von Queues zur Auswahl (Priority Queues, Class-Based Queues). Der Traffic wird ebenfalls wie bei Linux mittels Paketfilterregeln markiert, und dann so den einzelnen Queues zugewiesen:

```
altq on fxpl priq bandwidth 100Mb queue { ssh, other }  
queue ssh priority 15  
queue other priority 8 priq(default)
```

Hier wird ssh Traffic prioritär vor allem anderen Traffic behandelt.

#### 4.2.12 IDS

Snort wird auch von OpenBSD unterstützt. Siehe Punkt 4.1

#### 4.2.13 IPSec Endpunkt

OpenBSD beinhaltet bereits die IPSec Funktionalität. Diese wird aber aus Zeitgründen nicht weiter erläutert.



## 5 Bestimmen des Betriebssystem

### 5.1 Gegenüberstellung Debian 3.1 – OpenBSD 3.5

0 = neutral  
+ = gut  
++ = sehr gut

	Debian 3.1 (Sarge)	OpenBSD 3.5
Security	+	++
Performance	++	+
Paketfilter	++	++
Updates Verfügbarkeit	++	++
Updates Handhabung	++	+
Softwareverwaltung	++	0
Software Verfügbarkeit	++	+
Software Aktualität	+	++
Community Support	++	0
SMP Support	++	0(*)
Total	18/20	11/20

(\* SMP wird erst mit Version 3.6 unterstützt, die am 1. November 2004 erscheinen wird.)

### 5.2 Entscheidung für Debian 3.1 (Sarge)

Zum jetzigen Zeitpunkt scheint mir Debian die bessere Wahl zu sein als Betriebssystem für MuSeGa. OpenBSD bietet in der Version 3.5 noch keinen SMP Support. Dies ist sicherlich ein Nachteil, da auf dem Gateway viele rechenintensiven Operationen ausgeführt werden sollen (Verschlüsselung, Intrusion Detection, Statistiken, etc. ). Weiter ist Debian mit seinem ausgeklügelten Paketsystem einfacher auf dem neusten Stand zu halten als OpenBSD. Bei OpenBSD muss man bei jedem Patch den Sourcecode patchen und die Software/Kernel neu übersetzen.

Meine Wahl fällt also auf Debian 3.1 Codename Sarge. Die Entscheidung war aber knapper als dies die oben stehende Tabelle vermuten lässt. Bis in einem Jahr würde die Wahl vielleicht anders aussehen.



## 6 Weitere Komponenten

### 6.1 Datenbank

Für die Benutzer- und Regelnverwaltung empfiehlt sich eine Datenbank als Datenstruktur. MySQL sollte alle Anforderungen erfüllen und ist auch sehr schnell installiert:

```
apt-get install mysql-server
```

### 6.2 Datenbank Zugriff

Um nun auf diese Datenbank zuzugreifen gibt es bereits vorgefertigte Perl Module:

dbi  
dbd-mysql

Auch diese sind schnell installiert:

```
apt-get install libdbi-perl/testing libdbd-mysql-perl/testing
```

## 7 Quellen

- Distrowatch Homepage (26.07.2004)  
<http://www.distrowatch.org>
- **Absolute OpenBSD** von Michael W. Lucas (2003)  
ISBN 1-886411-99-9
- **Secure Architectures with OpenBSD** von Brandon Palmer & Jose Nazario (2004)  
ISBN 0-321-19366-0
- **Building Firewalls with OpenBSD and PF** (Second Edition) von Jacek Artymiak (2003)  
ISBN 83-916651-1-9
- **Das Firewall Buch** von Wolfgang Barth (2001)  
ISBN 3-934678-40-8



## 8 Anhang

### 8.1 Beschreibung der Linux Distributionen

(Quelle DistroWatch.org)

#### 8.1.1 Mandrakelinux



**Mandrakelinux** Mandrakelinux, started by Gaël Duval, is a distribution that has experienced enormous rise in popularity since its [first release](#) in July 1998. The developers took the Red Hat distribution, changed the default desktop to KDE and added an easy-to-use installer, breaking the myth that Linux is hard to install. Mandrake's hardware detection features and disk partitioning utilities are considered by many to be the best in the industry and many users found themselves running Mandrake where other distributions failed to provide the required usability.

Mandrakelinux has since matured to become a popular distribution among those new to Linux and among home users looking for an alternative operating system. The Mandrake development is completely open and transparent with new packages appearing in the so-called "cooker" directory on a daily basis. When a new release is entering a beta stage, a cooker snapshot is accepted as the first beta. The beta testing process used to be short and intensive, but starting with version 9.0, it has become longer and more thorough. The beta mailing lists are extremely busy, but you are still likely to receive a very fast response to any bug or concern that you report.

The result of this type of development is a cutting edge release - a highly up-to-date Linux distribution. As a trade-off, the users are likely to notice more bugs and perhaps less stability than with other distributions. Many people find this trade-off acceptable on their desktops - they get the very latest software and the occasional application crash is something they can live with.

**Pros:** User-friendly, graphical configuration utilities, enormous community support, NTFS partition resizing.

**Cons:** Some releases are buggy, the releases are initially made available to MandrakeClub members only.

**Software package management:** RPM

**Free download:** FTP installation available immediately after release, ISO images only after a delay lasting several weeks

[The Mandrakelinux page...](#)



### 8.1.2 Fedora

**Fedora**

For many, the name Red Hat epitomises Linux, as it is probably the best-known Linux company in the world. Founded in 1995 by Bob Young and Marc Ewing, Red Hat, Inc. has only recently started showing signs of profitability, due to services and its Red Hat Enterprise Linux product line. However, Red Hat Linux 9 was the last version in the Red Hat Linux product line, which was replaced by Fedora Core in late 2003. While Fedora is officially sponsored by Red Hat, it is developed with community participation, has a short life-span and serves mainly as a testing base for Red Hat Enterprise Linux.

What is so special about Red Hat Linux and Fedora Core? It is a curious mix of conservative and leading-edge packages put together on top of many knowledge-intensive utilities developed in-house. The packages are not the most up-to-date; once a new beta version is announced, the package versions are frozen, except for security updates. The result is a well-tested and stable distribution, the beta program and bug reporting facility are open to the public and there are several mailing lists. Red Hat Linux has become a dominant Linux distribution on servers around the world.

One other reason for Red Hat's success is the variety of popular services the company offers. The software packages are easy to update via Red Hat Network, a free repository of software and valuable information. A vast range of support services and enterprise Linux products are available from the company and, while not always cheap, you are virtually assured of an excellent support by highly skilled support personnel. The company has even developed a certification program to further popularise its distribution - the RHCE (Red Hat Certified Engineer) training and examinations are now available in most parts of the world. All these factors have contributed to the fact that Red Hat is now a recognised brand name in the IT industry.

**Pros:** Widely used, excellent community support, lots of innovation.

**Cons:** Limited product life-span of the free edition, poor multimedia support, concerns over the Red Hat to Fedora transition

**Software package management:** RPM

**Free download:** Yes

[The Fedora Project page...](#)



### 8.1.3 Debian

**debian**

Debian GNU/Linux, started by Ian Murdock in 1993, is a completely non-commercial project; perhaps the purest form of the ideals that started the free software movement. Hundreds of volunteer developers from all over the world contribute to the project, which is well managed and strict, assuring a quality distribution known as Debian.

At any time during the development process, there are three branches in the main directory tree - "stable", "testing" and "unstable" (also known as "sid"). When a new version of a package appears, it is placed in the unstable branch for first testing. If it passes, the package moves to the testing branch, which undergoes rigorous testing lasting many months. This branch is only declared stable after a very thorough testing. As a result of this, the distribution is possibly the most stable and reliable, albeit not the most up-to-date. While the stable branch is perfect for use on mission critical servers, many users prefer to run the more up-to-date testing or unstable branches on their personal computers.

Debian's other main claim to fame is the reputation for being hard to install, unless the user has an intimate knowledge about the computer's hardware. Compensating this failing is "apt-get", a convenient installer for Debian packages. Many Debian users joke that their installer is so bad, because they only need it once - as soon as Debian is up and running, all future updates of any scale can be accomplished via the apt-get utility.

**Pros:** 100% free, excellent web site and community resources, well-tested, painless software installation with apt-get.

**Cons:** Archaic installer, the stable version tends to be out-dated.

**Software package management:** DEB

**Free download:** Yes

[The Debian/GNU Linux page...](#)

### 8.1.4 Gentoo

**gentoo linux**

Gentoo Linux was created by Daniel Robbins, a former Stampede Linux and FreeBSD developer. It was the author's exposure to FreeBSD and its autobuild feature called "ports", which inspired him to incorporate ports into Gentoo under the

name of "portage". A detailed account of these beginnings of Gentoo can be found in this three-part series called [Making the distribution](#). Gentoo's first stable release was announced in March 2002.

Gentoo Linux is a source-based distribution. While the installation media provide various levels of pre-compiled binary packages to get a basic Linux system up and running, the idea behind Gentoo is to compile all source packages on the user's computer. The main advantage of this is that all software is highly optimised for the computer architecture it is built on. Also, updating the installed software to newer version is a matter of typing a simple command. Many Gentoo users enjoy the fact the software packages kept in a central repository are usually kept highly up-to-date and available within days (sometimes even within hours) since their release by the upstream developers. On the other hand, installing Gentoo and turning it into a full-blown distribution with the latest graphical desktops, multimedia and development tools is tedious and long - count on several days even on a computer with a fast processor.

**Pros:** Painless installation of individual software packages, highly up-to-date, superb documentation, the "geek feeling" of building a distribution tailored to user's needs.

**Cons:** Long and tedious system installation, occasional instability and risk of breakdown.

**Software package management:** SRC

**Free download:** Yes

[The Gentoo Linux page...](#)

### 8.1.5 SUSE



SUSE is another company with desktop focus, although a range of less visible enterprise class products are also available. The distribution has received positive reviews for its installer and YaST configuration tools, developed by SUSE's own developers. The documentation, which comes with the boxed product, has repeatedly been labelled as the most complete, thorough and usable by far. The distribution has achieved substantial market share in Europe and North America, but it is not marketed in Asia and other parts of the world. SUSE was acquired by Novell in late 2003.

SUSE's development takes place completely behind closed doors and no public betas are provided for testing. The company has a policy of not making the software available for free download until 1 - 2 months after the boxed versions are in stores. Even so, SUSE does not provide easily installable ISO images of SUSE LINUX, relying on sales of boxed sets to deliver the product to the majority of their users.

**Pros:** Professional attention to detail, easy-to-use YaST configuration tools.

**Cons:** Only available in parts of the world from software resellers or via FTP install,



includes proprietary components, which prevents re-distribution.

**Software package management:** RPM

**Free download:** Historically, SUSE did not provide ISO images for download, but this has changed starting with version 9.1, the Personal edition of which appeared on SUSE's FTP server about 2 months after the official release. The Professional edition of SUSE LINUX is available for installation via FTP, usually about 1 - 2 months after the official release. The FTP installation is not difficult, but requires fast Internet connection.

[The SUSE LINUX page...](#)

### 8.1.6 Slackware



**slackware** Slackware Linux, created by Patrick Volkerding in 1992, is the oldest surviving Linux distribution. It offers no bells and whistles, sticking with a text-based installer and no graphical configuration tools. Where other distributions tried hard to develop easy-to-use front ends for many common utilities, Slackware offers no hand-holding and everything is still done through configuration files. Because of this, Slackware is only recommended to those novice users who intend to spend some time on learning about Linux.

Nevertheless, Slackware has a magic appeal to many users. It is extremely stable and secure - very suitable for server deployment. Experienced Linux administrators find that the distribution is less buggy as it uses most packages in their pristine forms and without too many in-house enhancements which have a potential to introduce new bugs. Releases are infrequent (about once a year), although up-to-date packages are always available for download after the official release. Slackware is a fine distribution for those who are interested in deeper knowledge of Linux internals.

Perhaps the best characteristic of this distribution I have heard is this: if you need help with your Linux box, find a Slackware user. A Slackware user is more likely to fix the problem than a user familiar with any other distribution.

**Pros:** Highly stable and bug-free, strong adherence to UNIX principles.

**Cons:** All configuration is done by editing text files, limited hardware auto-detection.

**Software package management:** TGZ

**Free download:** Yes

[The Slackware Linux page...](#)

## 8.2 Beschreibung der BSD Varianten

### 8.2.1 FreeBSD



FreeBSD is a UN\*X-like operating system for the i386, IA-64, PC-98, Alpha/AXP, and UltraSPARC platforms based on U.C. Berkeley's "4.4BSD-Lite" release, with some "4.4BSD-Lite2" enhancements. It is also based indirectly on William Jolitz's port of U.C. Berkeley's "Net/2" to the i386, known as "386BSD", though very little of the 386BSD code remains. FreeBSD is used by companies, Internet Service Providers, researchers, computer professionals, students and home users all over the world in their work, education and recreation.

<http://www.freebsd.org/>

### 8.2.2 NetBSD



NetBSD is a free, secure, and highly portable UNIX-like Open Source operating system available for many platforms, from 64-bit AlphaServers and desktop systems to handheld and embedded devices. Its clean design and advanced features make it excellent in both production and research environments, and it is user-supported with complete source. Many applications are easily available through The NetBSD Packages Collection.

<http://www.netbsd.org/>

### 8.2.3 OpenBSD



The OpenBSD project produces a FREE, multi-platform 4.4BSD-based UNIX-like operating system. Our efforts emphasize portability, standardization, correctness, proactive security and integrated cryptography. OpenBSD supports binary emulation of most programs from SVR4 (Solaris), FreeBSD, Linux, BSD/OS, SunOS and HP-UX. OpenBSD is freely available from our FTP sites, and also available in an inexpensive 3-CD set.

<http://www.openbsd.org/>