



Berner Fachhochschule
Hochschule für Technik und Informatik HTI

Design 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 resultieren das ganze Systemdesign (Datenstrukturen, GUIs, etc.)

Versionskontrolle

Version	Datum	Kommentar	Genehmigt
0.1	03.08.2004	Erster Draft	
0.2	12.08.2004	überarbeitet	
0.3	25.09.2004	überarbeitet	



Inhaltsverzeichnis

- 1 Allgemeines..... 4
 - 1.1 Sinn und Zweck..... 4
 - 1.2 Leserkreis..... 4
- 2 Prozessabläufe..... 4
 - 2.1 Benutzeranmeldung..... 4
 - 2.2 Benutzer Abmeldung..... 4
 - 2.2.1 Manuell..... 4
 - 2.2.2 Automatisch..... 5
- 3 Datenbank Layout..... 5
 - 3.1 Haupttabellen..... 5
 - 3.1.1 Tabelle user..... 5
 - 3.1.2 Tabelle group..... 6
 - 3.1.3 Tabelle fw_rule..... 7
 - 3.1.4 Tabelle fw_nat_rule..... 8
 - 3.2 Firewall-Objekt Tabellen..... 8
 - 3.2.1 Tabelle fw_object_addressranges..... 8
 - 3.2.2 Tabelle fw_object_groups..... 9
 - 3.2.3 Tabelle fw_object_hosts..... 10
 - 3.2.4 Tabelle fw_object_networks..... 10
 - 3.2.5 Tabelle fw_service_groups..... 11
 - 3.2.6 Tabelle fw_service_icmp..... 11
 - 3.2.7 Tabelle fw_service_ip..... 12
 - 3.2.8 Tabelle fw_service_tcp..... 12
 - 3.2.9 Tabelle fw_service_udp..... 13
 - 3.2.10 Tabelle fw_time_timerange..... 14
 - 3.3 Weitere Tabellen..... 15
 - 3.3.1 Tabelle mac_blacklist..... 15
 - 3.3.2 Tabelle mac_user_history..... 16
- 4 Weitere Datenstrukturen..... 17
 - 4.1 User Statistiken..... 17
 - 4.2 Allgemeine Statistiken..... 17
- 5 Perl Programmierung..... 18
 - 5.1 Site Design mit CGI..... 18
 - 5.2 SQL Datenbank und Performance..... 18
 - 5.3 Perldoc..... 18
 - 5.4 Perl Libraries (PM)..... 18
- 6 Tests..... 18
- 7 Nützliche Links..... 19
- 8 Quellen..... 19



1 Allgemeines

1.1 Sinn und Zweck

Dieses Dokument dient als Basis für die Implementation von MuSeGa. Es veranschaulicht interne Abläufe sowie die verwendeten Datenstrukturen. Weiter gibt es nützliche Hinweise die bei der Implementation beachtet werden sollten.

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 Prozessabläufe

2.1 Benutzeranmeldung

1. Der Benutzer verbindet sich mit dem wireless Netzwerk
2. Der Notebook sendet einen DHCP Request aus
3. Der MuSeGa verarbeitet den DHCP Request und teilt dem Client die nötigen Daten mit
4. Der Benutzer startet seinen Webbrowser
5. Die http/https Anfrage wird auf den MuSeGa umgeleitet und der Login-Screen erscheint
6. Der Benutzer tippt seinen Accountnamen und sein Passwort ein
7. Der MuSeGa startet mit den Benutzerangaben eine Anfrage an den RADIUS-Server
8. Sind die Angaben korrekt, kriegt er eine positive Rückmeldung sowie weitere Attribute
9. Nun wird geprüft, ob der Benutzer schon einmal an MuSeGa angemeldet war
Ist dies nicht der Fall, wird der Benutzer lokal registriert
10. Anhand der Attribute vom RADIUS-Server wird der Benutzer einer bestimmten Gruppe zugeordnet
11. Nun wird der Benutzername mit der IP und MAC Adresse des Clients assoziiert.
12. Die Firewallregeln der Gruppe werden nun auf die IP- und MAC-Adresse des Clients angepasst und angewendet.
13. Der Benutzer kann von nun an alle Dienste benutzen, die für ihn freigegeben wurden.

2.2 Benutzer Abmeldung

2.2.1 Manuell

1. Der Benutzer begibt sich auf die MuSeGa Website und klickt dort auf logout
2. Alle zur IP Adresse gehörenden Verbindungen werden geschlossen
3. Alle Firewallregeln zu dieser IP Adresse werden gelöscht

2.2.2 Automatisch

1. Wenn eine definierte Zeit kein Verkehr generiert wird, wird der Benutzer automatisch ausgeloggt
2. -> geht weiter bei Punkt2 (Siehe 3.2.1)

3 Datenbank Layout

3.1 Haupttabellen

MuSeGa besteht aus rund vier Haupttabellen. In den „User“ und „Group“ Tabellen werden die Benutzer- und Gruppen-Eigenschaften gespeichert. In den Tabellen „fw_rule“ und „fw_nat_rule“ werden die Firewall-Regeln sowie die NAT-Regeln gespeichert.

3.1.1 Tabelle user

In dieser Tabelle werden alle Eigenschaften die einem Benutzer zugeordnet werden können abgespeichert.

Feldname	Beschreibung
id	Primärschlüssel
username	Benutzername
group	Gruppen, zu denen der Benutzer gehört. (via RADIUS)
mac_address	MAC Adresse mit der der Benutzer zur Zeit online ist.
ip_address	Momentan zugeteilte IP-Adresse des Benutzers
online	Offline, Online
last_login	Zeitpunkt, als der Benutzer das letzte mal einlogte
nb_login	
locked	True/False: Gibt an ob der Account gesperrt ist.
total_packet	Anzahl IP Pakete total
total_byte	Anzahl Bytes total
local_auth	Gibt an ob ein Anmelden via lokale DB erlaubt ist.
local_pw	Passwort für lokales Anmelden
nmap_result	Resultat eines nmap Checks auf den Client
client_os	OS des Clients (von nmap)
client_open_port	Offene Ports auf dem Client
description	

#

```
# Table structure for table `user`
#
CREATE TABLE `user` (
  `id` int(10) unsigned NOT NULL auto_increment,
  `username` varchar(30) NOT NULL default '',
  `group` varchar(30) NOT NULL default '',
  `mac_address` varchar(17) NOT NULL default '',
  `ip_address` varchar(45) NOT NULL default '',
  `online` char(3) NOT NULL default '',
  `last_login` datetime NOT NULL default '0000-00-00 00:00:00',
  `nb_login` int(10) unsigned NOT NULL default '0',
  `locked` char(3) NOT NULL default '',
  `total_packet` bigint(20) unsigned NOT NULL default '0',
  `total_byte` bigint(20) unsigned NOT NULL default '0',
  `local_auth` char(3) NOT NULL default 'no',
  `local_pw` varchar(30) NOT NULL default '',
  `nmap_result` text NOT NULL,
  `client_os` varchar(30) NOT NULL default '',
  `client_open_port` text NOT NULL,
  `description` varchar(100) NOT NULL default '',
  PRIMARY KEY (`id`)
) TYPE=MyISAM AUTO_INCREMENT=2 ;
```

3.1.2 Tabelle group

Alle Gruppeneigenschaften befinden sich in dieser Tabelle.

Feldname	Beschreibung
id	Primärschlüssel
name	Name der Gruppe
class_string	Wert, welcher mit dem Wert vom RADIUS-Server verglichen wird, um die Gruppenzuteilung vorzunehmen
bandwidth	Maximale Bandbreite die den Mitgliedern dieser Gruppe zur Verfügung steht
description	Beschreibung

```
# -----
#
# Table structure for table `group`
#
CREATE TABLE `group` (
  `id` int(10) unsigned NOT NULL auto_increment,
  `name` varchar(30) NOT NULL default '',
  `class_string` varchar(30) NOT NULL default '',
  `bandwidth` int(11) NOT NULL default '-1',
  `description` varchar(50) NOT NULL default '',
  `unused` text NOT NULL,
  PRIMARY KEY (`id`)
) TYPE=MyISAM AUTO_INCREMENT=3 ;
```



3.1.3 Tabelle fw_rule

Diese Tabelle beinhaltet also alle Statischen und Dynamischen Firewall Regeln, welche vom Administrator definiert wurden.

Feldname	Beschreibung
id	Primärschlüssel
number	Nummer der Regel (wichtig für Reihenfolge)
source	Liste von Source-Objekten
destination	Liste von Destination-Objekten
service	Liste von Service-Objekten
action	Was soll mit betroffenen Paketen gemacht werden: Werte: ACCEPT, REJECT, DROP
time	Liste von Zeit-Objekten
log_prefix	Falls das Paket geloggt wird; Prefix der dann ebenfalls im Log steht
log_level	Falls das Paket geloggt wird; Level des Loggings: Notice, Debug, Critical etc.
stateful	Gibt an, ob es sich um eine stateful Regel handelt (per Default ja)
description	Beschreibung
enabled	0 oder 1, gibt an ob die Regel aktiv ist

Anmerkung zu den Firewall Regeln:

Unter Source stehen drei verschiedene Typen zur Verfügung:

1. Host Objekt (IP Adresse) -> statische Regel
2. User Objekt (also ein Link auf eine IP-Adresse -> dynamische Regel)
3. Group Objekt (via User-Tabelle bezieht sich die Regel wieder auf eine IP-Adresse -> dynamische Regel)

```
# -----
#
# Table structure for table `fw_rule`
#

CREATE TABLE `fw_rule` (
  `id` int(10) unsigned NOT NULL auto_increment,
  `number` smallint(5) unsigned NOT NULL default '0',
  `source` text NOT NULL,
  `destination` text NOT NULL,
  `service` text NOT NULL,
  `action` varchar(6) NOT NULL default 'drop',
  `time` text NOT NULL,
  `log_prefix` varchar(30) NOT NULL default '',
  `log_level` varchar(30) NOT NULL default '',
  `stateful` tinyint(1) unsigned NOT NULL default '1',
  `description` varchar(200) NOT NULL default '',
  `enabled` tinyint(1) unsigned NOT NULL default '1',
  PRIMARY KEY (`id`)
) TYPE=MyISAM AUTO_INCREMENT=62 ;
```

3.1.4 Tabelle fw_nat_rule

Diese Tabelle enthält alle vom Administrator definierten NAT Regeln

Feldname	Beschreibung
id	Primärschlüssel
number	Nummer der Regel (wichtig für Reihenfolge)
original_source	Ursprungsadresse
original_destination	Ursprungsziel
original_service	Ursprungsservice
translated_source	Übersetzte Source
translated_destination	Übersetzte Destination
translated_service	Übersetzter Service
description	Beschreibung
enabled	0 oder 1, gibt an ob die Regel aktiv ist

```
# -----
#
# Table structure for table `fw_nat_rule`
#
CREATE TABLE `fw_nat_rule` (
  `id` int(10) unsigned NOT NULL auto_increment,
  `number` smallint(5) unsigned NOT NULL default '0',
  `original_source` text NOT NULL,
  `original_destination` text NOT NULL,
  `original_service` text NOT NULL,
  `translated_source` text NOT NULL,
  `translated_destination` text NOT NULL,
  `translated_service` text NOT NULL,
  `description` varchar(200) NOT NULL default '',
  `enabled` tinyint(1) unsigned NOT NULL default '1',
  PRIMARY KEY (`id`)
) TYPE=MyISAM AUTO_INCREMENT=9 ;
```

3.2 Firewall-Objekt Tabellen

Die Tabellen fw_rule und fw_nat_rule setzen sich aus Objekten zusammen. Diese Objekte (zum Beispiel TCP-Service ftp) werden wiederum in Tabellen gehalten.

3.2.1 Tabelle fw_object_addressranges

Eine Zeile dieser Tabelle stellt immer ein Addressrange-Objekt dar.

Feldname	Beschreibung
Id	Primärschlüssel

Feldname	Beschreibung
name	Name des Objektes
start_address	Start des Adressbereichs
stop_address	Stop des Adressbereichs
description	Beschreibung

```
# -----
#
# Table structure for table `fw_object_addressranges`
#
CREATE TABLE `fw_object_addressranges` (
  `id` int(10) unsigned NOT NULL auto_increment,
  `name` varchar(50) NOT NULL default '',
  `start_address` varchar(45) NOT NULL default '',
  `stop_address` varchar(45) NOT NULL default '',
  `description` varchar(200) NOT NULL default '',
  PRIMARY KEY (`id`)
) TYPE=MyISAM AUTO_INCREMENT=2 ;
```

3.2.2 Tabelle fw_object_groups

Um die Regeln übersichtlicher zu gestalten, ist es auch möglich Objekte einer Gruppe zuzuordnen, und dann nur diese Gruppe in die Regel einzufügen. Eine Zeile in dieser Tabelle stellt also ein solches Gruppenobjekt dar.

Feldname	Beschreibung
id	Primärschlüssel
name	Name des Objektes
member	Liste der Mitglieder in dieser Gruppe
description	Beschreibung

```
# -----
#
# Table structure for table `fw_object_groups`
#
CREATE TABLE `fw_object_groups` (
  `id` int(10) unsigned NOT NULL auto_increment,
  `name` varchar(50) NOT NULL default '',
  `member` text NOT NULL,
  `description` varchar(200) NOT NULL default '',
  PRIMARY KEY (`id`)
) TYPE=MyISAM AUTO_INCREMENT=3 ;
```



3.2.3 Tabelle fw_object_hosts

Eine Zeile in dieser Tabelle stellt ein normales Host-Objekt dar. Dabei handelt es sich um ein Gerät mit einer IP-Adresse und einer MAC-Adresse.

Feldname	Beschreibung
id	Primärschlüssel
name	Name des Objektes
ip_address	IP-Adresse des Objektes
mac	MAC-Adresse des Objektes
description	Beschreibung

```
# -----
#
# Table structure for table `fw_object_hosts`
#
CREATE TABLE `fw_object_hosts` (
  `id` int(10) unsigned NOT NULL auto_increment,
  `name` varchar(50) NOT NULL default '',
  `ip_address` varchar(45) NOT NULL default '',
  `mac` varchar(17) NOT NULL default '',
  `description` varchar(200) NOT NULL default '',
  PRIMARY KEY (`id`)
) TYPE=MyISAM AUTO_INCREMENT=6 ;
```

3.2.4 Tabelle fw_object_networks

Eine Zeile in dieser Tabelle repräsentiert also ein Network-Objekt

Feldname	Beschreibung
id	Primärschlüssel
name	Name des Objektes
address	IP-Adresse des Subnetzes
netmask	Netzwerkmaske des Subnetzes
description	Beschreibung

```
# -----
#
# Table structure for table `fw_object_networks`
#
CREATE TABLE `fw_object_networks` (
  `id` int(10) unsigned NOT NULL auto_increment,
  `name` varchar(50) NOT NULL default '',
  `address` varchar(45) NOT NULL default '',
  `netmask` varchar(15) NOT NULL default '',
  `description` varchar(200) NOT NULL default '',
  PRIMARY KEY (`id`)
)
```

) TYPE=MyISAM AUTO_INCREMENT=3 ;

3.2.5 Tabelle fw_service_groups

Um die Regeln übersichtlicher zu gestalten, ist es auch möglich Service-Objekte einer Gruppe zuzuordnen, und dann nur diese Gruppe in die Regel einzufügen. Eine Zeile in dieser Tabelle stellt also ein solches Gruppenobjekt dar.

Feldname	Beschreibung
id	Primärschlüssel
name	Name des Objektes
member	Liste der Gruppenmitglieder
description	Beschreibung

```
# -----
#
# Table structure for table `fw_service_groups`
#
CREATE TABLE `fw_service_groups` (
  `id` int(10) unsigned NOT NULL default '0',
  `name` varchar(50) NOT NULL default '',
  `member` text NOT NULL,
  `description` varchar(200) NOT NULL default ''
) TYPE=MyISAM;
```

3.2.6 Tabelle fw_service_icmp

Eine Zeile dieser Tabelle entspricht einem ICMP Service.

Feldname	Beschreibung
id	Primärschlüssel
name	Name des Objektes
type	ICMP Typ
type_desc	Beschreibung des ICMP Typs
code	ICMP Code
code_desc	Beschreibung des ICMP Codes
description	Beschreibung

```
# -----
#
# Table structure for table `fw_service_icmp`
#
CREATE TABLE `fw_service_icmp` (
  `id` int(10) unsigned NOT NULL auto_increment,
```

```

`name` varchar(50) NOT NULL default '',
`type` tinyint(4) NOT NULL default '-1',
`type_desc` varchar(50) NOT NULL default '',
`code` tinyint(3) unsigned NOT NULL default '0',
`code_desc` varchar(50) NOT NULL default '',
`description` varchar(200) NOT NULL default '',
PRIMARY KEY (`id`)
) TYPE=MyISAM AUTO_INCREMENT=3 ;

```

3.2.7 Tabelle fw_service_ip

Eine Zeile dieser Tabelle entspricht einem IP Objekt.

Feldname	Beschreibung
id	Primärschlüssel
name	Name des Objektes
protocol	Protokollnummer (/etc/protocols)
option	IP Optionen
description	Beschreibung

```

# -----
#
# Table structure for table `fw_service_ip`
#
CREATE TABLE `fw_service_ip` (
  `id` int(10) unsigned NOT NULL auto_increment,
  `name` varchar(50) NOT NULL default '',
  `protocol` tinyint(3) unsigned NOT NULL default '0',
  `option` varchar(100) NOT NULL default '',
  `description` varchar(200) NOT NULL default '',
  PRIMARY KEY (`id`)
) TYPE=MyISAM AUTO_INCREMENT=15 ;

```

3.2.8 Tabelle fw_service_tcp

Eine Zeile in dieser Tabelle entspricht einem TCP-Objekt

Feldname	Beschreibung
id	Primärschlüssel
name	Name des Objektes
source_port_start	Sourceport-Range Start
source_port_end	Sourceport-Range Stop
destination_port_start	Destinationport-Range Start
destination_port_end	Destinationport-Range-Stop



Feldname	Beschreibung
examine_flag_urg	Überprüfe dieses TCP Flag
examine_flag_ack	Überprüfe dieses TCP Flag
examine_flag_psh	Überprüfe dieses TCP Flag
examine_flag_rst	Überprüfe dieses TCP Flag
examine_flag_syn	Überprüfe dieses TCP Flag
examint_flat_fin	Überprüfe dieses TCP Flag
flag_urg	Dieses TCP Flag muss gesetzt sein
flag_ack	Dieses TCP Flag muss gesetzt sein
flag_psh	Dieses TCP Flag muss gesetzt sein
flag_rst	Dieses TCP Flag muss gesetzt sein
flag_syn	Dieses TCP Flag muss gesetzt sein
flag_fin	Dieses TCP Flag muss gesetzt sein
description	Beschreibung

```
# -----
#
# Table structure for table `fw_service_tcp`
#

CREATE TABLE `fw_service_tcp` (
  `id` int(10) unsigned NOT NULL auto_increment,
  `name` varchar(50) NOT NULL default '',
  `source_port_start` smallint(5) unsigned NOT NULL default '1024',
  `source_port_end` smallint(5) unsigned NOT NULL default '65535',
  `destination_port_start` smallint(5) unsigned NOT NULL default '0',
  `destination_port_end` smallint(5) unsigned NOT NULL default '0',
  `examine_flag_urg` smallint(1) unsigned NOT NULL default '0',
  `examine_flag_ack` smallint(1) unsigned NOT NULL default '0',
  `examine_flag_psh` smallint(1) unsigned NOT NULL default '0',
  `examine_flag_rst` smallint(1) unsigned NOT NULL default '0',
  `examine_flag_syn` smallint(1) unsigned NOT NULL default '0',
  `examine_flag_fin` smallint(1) unsigned NOT NULL default '0',
  `flag_urg` smallint(1) unsigned NOT NULL default '0',
  `flag_ack` smallint(1) unsigned NOT NULL default '0',
  `flag_psh` smallint(1) unsigned NOT NULL default '0',
  `flag_rst` smallint(1) unsigned NOT NULL default '0',
  `flag_syn` smallint(1) unsigned NOT NULL default '0',
  `flag_fin` smallint(1) unsigned NOT NULL default '0',
  `description` varchar(200) NOT NULL default '',
  PRIMARY KEY (`id`)
) TYPE=MyISAM AUTO_INCREMENT=5 ;
```

3.2.9 Tabelle fw_service_udp

Eine Zeile dieser Tabelle entspricht einem UDP-Objekt



Feldname	Beschreibung
id	Primärschlüssel
name	Name des Objektes
source_port_start	Sourceport-Range Start
source_port_end	Sourceport-Range Stop
destination_port_start	Destinationport-Range Start
destination_port_end	Destinationport-Range-Stop
description	Beschreibung

```
# -----
#
# Table structure for table `fw_service_udp`
#
CREATE TABLE `fw_service_udp` (
  `id` int(10) unsigned NOT NULL auto_increment,
  `name` varchar(50) NOT NULL default '',
  `source_port_start` smallint(5) unsigned NOT NULL default '0',
  `source_port_end` smallint(5) unsigned NOT NULL default '0',
  `destination_port_start` smallint(5) unsigned NOT NULL default '0',
  `destination_port_end` smallint(5) unsigned NOT NULL default '0',
  `description` varchar(200) NOT NULL default '',
  PRIMARY KEY (`id`)
) TYPE=MyISAM AUTO_INCREMENT=2 ;
```

3.2.10 Tabelle fw_time_timerange

Eine Zeile dieser Tabelle entspricht einem Timerange-Objekt.

Feldname	Beschreibung
id	Primärschlüssel
name	Name des Objektes
start_minute	Start Minute
start_hour	Start Stunde
start_day	Start Tag
start_month	Start Monat
start_year	Start Jahr
stop_minute	Stop Minute
stop_hour	Stop Stunde
stop_day	Stop Tag
stop_month	Stop Monat
stop_year	Stop Jahr

Feldname	Beschreibung
description	Beschreibung
mon	An diesem Wochentag
tue	An diesem Wochentag
wed	An diesem Wochentag
thu	An diesem Wochentag
fri	An diesem Wochentag
sat	An diesem Wochentag
sun	An diesem Wochentag

```
# -----
#
# Table structure for table `fw_time_timerange`
#
CREATE TABLE `fw_time_timerange` (
  `id` int(10) unsigned NOT NULL auto_increment,
  `name` varchar(50) NOT NULL default '',
  `start_minute` tinyint(2) NOT NULL default '-1',
  `start_hour` tinyint(2) NOT NULL default '-1',
  `start_day` tinyint(2) NOT NULL default '-1',
  `start_month` tinyint(2) NOT NULL default '-1',
  `start_year` smallint(4) NOT NULL default '-1',
  `stop_minute` tinyint(2) NOT NULL default '-1',
  `stop_hour` tinyint(2) NOT NULL default '-1',
  `stop_day` tinyint(2) NOT NULL default '-1',
  `stop_month` tinyint(2) NOT NULL default '-1',
  `stop_year` smallint(4) NOT NULL default '-1',
  `description` varchar(200) NOT NULL default '',
  `mon` tinyint(2) NOT NULL default '-1',
  `tue` tinyint(2) NOT NULL default '-1',
  `wed` tinyint(2) NOT NULL default '-1',
  `thu` tinyint(2) NOT NULL default '-1',
  `fri` tinyint(2) NOT NULL default '-1',
  `sat` tinyint(2) NOT NULL default '-1',
  `sun` tinyint(2) NOT NULL default '-1',
  PRIMARY KEY (`id`)
) TYPE=MyISAM AUTO_INCREMENT=3;
```

3.3 Weitere Tabellen

Weiter werden noch ein paar weitere Tabellen dazukommen.

3.3.1 Tabelle mac_blacklist

Verwendung:

In dieser Tabelle werden alle gesperrten MAC-Adressen eingetragen. Die Netzwerkkarten mit diesen MAC-Adressen können sich dann nicht mehr einloggen bis der Eintrag wieder aus der Datenbank gelöscht wird.

Feldname	Beschreibung
mac	MAC Adresse
date	Datum der Sperrung
reason	Grund der Sperrung
how_long	Zeit wann die Sperrung automatisch wieder aufgehoben wird.

```
# -----
#
# Table structure for table `mac_blacklist`
#
CREATE TABLE `mac_blacklist` (
  `id` int(10) unsigned NOT NULL auto_increment,
  `mac` varchar(17) NOT NULL default '',
  `date` datetime NOT NULL default '0000-00-00 00:00:00',
  `reason` text NOT NULL,
  `how_long` datetime NOT NULL default '0000-00-00 00:00:00',
  `unused` varchar(50) NOT NULL default '',
  PRIMARY KEY (`id`)
) TYPE=MyISAM AUTO_INCREMENT=2 ;
```

3.3.2 Tabelle mac_user_history

In dieser Tabelle werden bei jeder Benutzeranmeldung folgende Daten hineingeschrieben:

Feldname	Beschreibung
id	Primärschlüssel
mac_address	MAC Adresse des Clients
username	Benutzername des Clients
ip_address	IP-Adresse des Clients
first_time	Datum der ersten Anmeldung
last_time	Datum der letzten Anmeldung

Bei der Anmeldung wird überprüft, ob es schon einen Eintrag mit der aktuellen MAC-Adresse und dem aktuellen Benutzernamen gibt. Ist dies der Fall, werden lediglich die Felder „ip_address“ und „last_time“ aktualisiert. Anderenfalls wird ein neuer Eintrag erstellt.

Damit wäre es zum Beispiel möglich, eine gestohlene Netzwerkkarte oder ein gestohlenen Notebook wieder ausfindig zu machen, falls sich der Dieb wieder am selben Netzwerk anmeldet.



```
# -----  
#  
# Table structure for table `mac_user_history`  
#  
  
CREATE TABLE `mac_user_history` (  
  `id` bigint(20) unsigned NOT NULL auto_increment,  
  `mac_address` varchar(17) NOT NULL default '',  
  `username` varchar(30) NOT NULL default '',  
  `ip_address` varchar(45) NOT NULL default '',  
  `first_time` datetime NOT NULL default '0000-00-00 00:00:00',  
  `last_time` datetime NOT NULL default '0000-00-00 00:00:00',  
  PRIMARY KEY (`id`)  
) TYPE=MyISAM AUTO_INCREMENT=1 ;
```

4 Weitere Datenstrukturen

4.1 User Statistiken

Folgende Werte werden zu Statistik Zwecken pro Benutzer gespeichert. Da sich diese Zähler dauernd ändern, werden sie nicht in der Datenbank gespeichert. Stattdessen werden sie in einem Hashtable gehalten und nach Bedarf kann eine RRD-Grafik erstellt werden.

Name	Beschreibung
session_packet	Anzahl IP Pakete in dieser Session
session_byte	Anzahl Bytes in dieser Session
packets_sec	Pakete pro Sekunde (Durchschnitt aus letzter Minute)
bytes_sec	Bytes pro Sekunde (Durchschnitt aus letzter Minute)
nb_target	Anzahl Destination IP-Adressen in den letzten xxx Minuten (Also Nummer der Verbindungen)

4.2 Allgemeine Statistiken

Vom gesamten Verkehr, der den Gateway passiert werden folgende Werte gespeichert:

- Source-IP-Address
- Destination-IP-Address
- Protokoll
- Destination-Port
- Total-Packets
- Total-Bytes



Diese Werte können dann für verschiedene Statistiken verwendet werden.

5 Perl Programmierung

5.1 Site Design mit CGI

Ablauf:

1. Get und Post Variablen auswerten
2. Anhand der Variablen Datenbank Updates durchführen
3. Datenbank Queries durchführen
4. Anhand der Resultate die Website generieren

5.2 SQL Datenbank und Performance

Die Datenbankzugriffe sollten aus Performancegründen auf ein Minimum reduziert werden. Es empfiehlt sich also die Verwendung von Hashtabellen um schneller auf die Daten zugreifen zu können. Sind die Daten im Hashtable aktualisiert, können sie alle auf einmal mit der Datenbank abgeglichen werden

5.3 Perldoc

Alle Funktionen werden direkt im Code beschrieben und anschliessend kann daraus automatisch die HTML-Dokumentation der Funktionen generiert werden

5.4 Perl Libraries (PM)

Um den Überblick im Code zu behalten, werden alle Gruppen von Funktionen in eigene Bibliotheken ausgelagert

6 Tests

Bereits in der Designphase muss ich sie Frage gestellt werden, wie die Software am Schluss getestet wird. Zusammen mit den Betreuern sind wir zu folgendem Schluss gekommen:

Die Software wird mit einem Fieldtest getestet:

Dazu wird eine Schulklasse der HTI (I-01) von Hansjürg Wenger für ca. vier Lektionen zur Verfügung stehen. Der genaue Termin muss noch abgemacht werden.

Für diese Tests muss vorher ein genaues Testszenario definiert werden. Getestet werden eigentlich alle implementierten Features.

- ≥ 10 Benutzer mit unterschiedlichen Gruppenzugehörigkeiten einloggen
- Einloggen via PPTP
- Einloggen zu SWITCHMobile (ipsec pass through)
- Traffic erzeugen
- Load-Messungen machen



- Filter testen (class attribut)
- Internet Zugriff
- Homedir Zugriff

Das Testszenario wird in einem eigenen Dokument ausführlich beschrieben. Es kann unter <http://musega.ch> heruntergeladen werden.

7 Nützliche Links

- MySQL Data Types:
http://dev.mysql.com/doc/mysql/en/Numeric_type_overview.html
- Netfilter Erweiterungen HOWTO
<http://www.netfilter.org/documentation/HOWTO/de/netfilter-extensions-HOWTO.html>

8 Quellen

- Perl Dokumentation (25.9.2004)
<http://www.perldoc.com/>
- MySQL Manual (25.9.2004)
<http://dev.mysql.com/doc/mysql/en/>
- Linux 2.4 Packet Filtering HOWTO (25.9.2004)
<http://www.netfilter.org/documentation/HOWTO/de/packet-filtering-HOWTO.html>
- The Hidden Treasures of IPTables (25.9.2004)
<http://www.lowth.com/howto/iptables-treasures.php>
- Iptables Tutorial 1.1.19 (25.9.2004)
<http://iptables-tutorial.frozentux.net/iptables-tutorial.html>
- Bifrost - Firewalling made easy (25.9.2004)
<http://bifrost.heimdalls.com/demo.html>
- Netfilter Erweiterungen HOWTO (25.9.2004)
<http://www.netfilter.org/documentation/HOWTO/de/netfilter-extensions-HOWTO.html>
- Debian GNU/Linux Anwenderhandbuch (25.9.2004)
<http://www.openoffice.de/linux/buch/index.html>