

Dokumentation zur Einrichtung des Schulserver für die Freie Waldorfschule Vaihingen/Enz

Benedikt Braunger, Ales Bloss, Jonas Heinrich

Inhalt

1. Das Projekt.....4

1.0.1 - Effizientere Verwaltung durch zentralen Server.....	4
1.0.2 - Öffentlicher Internetzugang.....	5
1.0.3 - Schule und neue Medien.....	5
1.0.4 - Einsatz von Opensource-Software.....	6

2. Server.....8

2.1 Installation der Software.....	8
2.1.1 - Betriebssystem.....	8
2.1.2 - Anwendungen.....	8
2.1.3 - Netzwerk.....	9
2.2 Konfiguration.....	10
2.2.1 - Netzwerk-Adapter.....	10
2.2.2 - DHCP-Server und Hostname.....	10
2.2.3 - Bind9 DNS.....	11
2.2.4 - Apache-Server.....	13
2.2.5 - MySQL-Datenbank einrichten.....	13
2.2.6 - FTP-Server.....	14
2.2.7 - Munin - Systemüberwachung.....	15
2.2.8 - Samba-Freigaben.....	18
2.2.9 - SQUID-Proxy.....	19
2.2.10 - OpenVPN-Server.....	21
2.2.11 - IPTables2-Firewall.....	24
2.2.12 - Rsync Backup.....	26
2.2.13 - pptpd.....	26

3. Client.....28

Allgemeines.....	28
3.1 Installation der Software.....	28
3.2 Einrichten der Software.....	29
3.2.1 - Konten einrichten.....	29
3.2.1 - Konten anpassen.....	29
3.2.2 - Zeitplaner Cron konfigurieren.....	30
3.3 System absichern.....	30
3.3.1 - Bootloader Grub.....	30
3.3.2 - Epiphany Webbrowser.....	30
3.3.3 - Gnome-Desktop.....	30
3.3.4 - Zugriffsrechte.....	31
3.3.5 - X-Server absichern.....	32

3.3 Grafisches.....	32
3.3.1 - Wallpaper (1. Konzept).....	32
3.3.2 - Splash-Screen (Bootscreen).....	32
3.3.3 - GDM-Theme (Login).....	33
3.3.4 - Bildschirmschoner.....	33
3.4 Priviligierter Client (ITG-Raum).....	34
3.4.1 - Konfiguration für Windows-Systeme.....	34

03.06.2010 Vaihingen/Enz



Dokumentversion: 1.9

1. Das Projekt

Das zentrale Anliegen unseres Projektes ist es, das bestehende Computer-Netzwerk zu überarbeiten im Bezug auf Sicherheit, Stabilität und einfachere Verwaltung. Hinzu stellen wir mit unserem neuen Server die Basis und Infrastruktur für ein Internet-Café, dass es in Zukunft Oberstufenschülern ermöglichen soll, gezielt auf Lerninhalte oder verschiedene Informationen (Dienste) zurückzugreifen. Nach unserer Meinung ist es umungänglich die „neuen Medien“ und Technologien, die in unserer Gesellschaft allgemein eine immer größere Rolle spielen werden, mit in den Schulalltag zu integrieren um damit den produktiven Umgang mit Medien zu fördern und, in Rücksicht auf spätere Erwartungen des Berufsalltag, auch zu lehren.

1.0.1 - Effizientere Verwaltung durch zentralen Server



Unser ursprüngliches Ziel für dieses Projekt war es ein "Internet-Cafe" einzurichten, dabei sollte ein "transparenter Proxy"¹ für die Client-Computer nur bestimmte Seiten für Schüler zugänglich machen (Prinzip "Whitelist"²). Doch es stellte sich heraus, dass zugunsten der Sicherheit die Schüler- und Lehrernetzwerke durch eine sogenannte VPN-Verbindung³ software-basiert getrennt sein sollten. Dadurch gab es zwar viel mehr zu konfigurieren als gedacht, doch der neu eingerichtete Server ermöglichte

nun eine sichere und übersichtliche Verwaltung des Netzwerkes:

- Eingeschränkter, öffentlicher Zugang bei Standard-Konfiguration
- Getrenntes Lehrernetzwerk ohne Einschränkungen mit VPN-Konfiguration
- Internet-Zugang im Oberstufenraum
- Angepasstes "Edubuntu"-Betriebssystem⁴ für Client-Rechner
 - Artwork für Schulbetriebssystem (Modifizierter Bildschirmhintergrund, Bildschirmschoner, Bootscreen)
- Schulserver mit eigenem:
 - DHCP-Server⁵ (IP-Adressen Vergabe)
 - Squid-Proxy⁶ (eingeschränktes Netzwerk)
 - Iptables-Firewall⁷ (Schutz vor externen Angriffen auf das Netzwerk)
 - FTP⁸- und Apache-Server⁹ (schuleigenes, lokales Webportal)
 - 2 VPN-Server (Lehrer- und Beleuchternetzwerk)

1 [http://de.wikipedia.org/wiki/Proxy_\(Rechnernetz\)#Transparenter_Proxy](http://de.wikipedia.org/wiki/Proxy_(Rechnernetz)#Transparenter_Proxy)

2 <http://de.wikipedia.org/wiki/Whitelist#Software>

3 http://de.wikipedia.org/wiki/Virtual_Private_Network

4 <http://de.wikipedia.org/wiki/Edubuntu#Edubuntu>

5 <http://de.wikipedia.org/wiki/DHCP-Server>

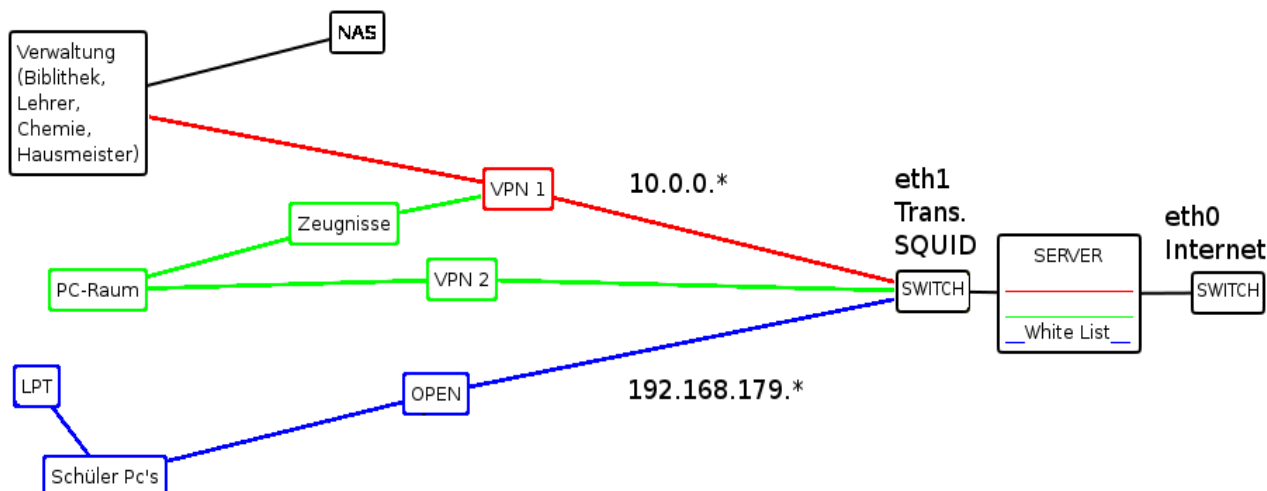
6 <http://de.wikipedia.org/wiki/Squid>

7 <http://de.wikipedia.org/wiki/Iptables>

8 http://de.wikipedia.org/wiki/File_Transfer_Protocol

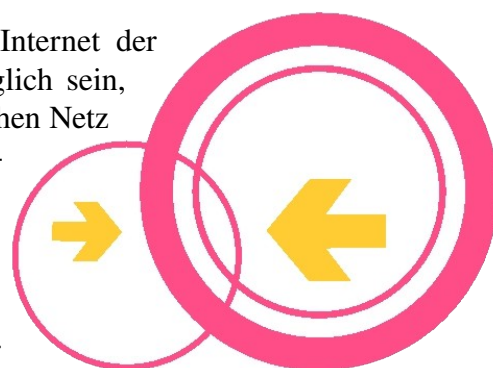
9 http://de.wikipedia.org/wiki/Apache_HTTP_Server

Alles in allem eine gute Basis für weitere Informatik-Projekte.



1.0.2 - Öffentlicher Internetzugang

Für jeden, auch nicht der Schule angehörend, sollte das Internet der Schule (gefiltert und getrennt von sensiblen Daten) zugänglich sein, über WLAN oder Internet-Terminal¹⁰. Zwar sind im öffentlichen Netz nur ausgewählte Seiten erreichbar, wie die größte Online-Enzyklopädie Wikipedia, das Wörterbuch Leo.org oder z.B. die Fahrplanauskunft bei Bahn.de, aber jeder Besucher des Geländes kann auf diese Angebote kostenlos zugreifen - im Sinne von Freifunk¹¹ (Verbreitung freier und unabhängiger Netzwerke, Demokratisierung der Kommunikationsmedien und die Förderung lokaler Sozialstrukturen). Dieser offene, weitflächiger Zugang vereinfacht im Optimalfall natürlich auch die Vernetzung aller Laptops und Desktops in unterschiedlichen Gebäuden ohne Netzkabel zu verlegen.



1.0.3 - Schule und neue Medien

Interessante Artikel, Webseiten oder Dokumente von Zuhause kann man nun an Computern im Oberstufenraum jeder Zeit ausrücken und für den Unterricht nutzen. Auch lässt sich über diesen Zugang die Recherche für Hausaufgaben vereinfachen. Schüler können dabei nicht nur selbständig lernen, wie sich wichtige Informationen im Internet finden lassen oder wie man diese zusammenträgt/nutzt, sondern auch die Arbeit mit dem Computer und dessen Software: Sei es das Anlegen einer Präsentation, die Gestaltung eines Dokumentes, Bildbearbeitung oder das Erzeugen von Tabellen mit OpenOffice¹².

Pädagogische Begründungen für die Nutzung vom Internet in Schulen

- Internet-Nutzung als elementare Kulturtechnik: Das Internet ist wichtig für Kommunikation und Kultur. Weiter sind die verschiedenen Bildungs- und Unterhaltungsangebote und der Kauf von Waren und Dienstleistungen (in diesen

¹⁰ <http://de.wikipedia.org/wiki/Kiosk-Modus>

¹¹ <http://start.freifunk.net/>

¹² <http://de.wikipedia.org/wiki/Openoffice>

Zusammenhang ist das Wissen über Datensicherheit und Datenqualität unentbehrlich) sehr wichtig.

- Steigerung der Effektivität/Effizienz: „Mit dem Einsatz von Medien kann das Ziel verfolgt werden, Lehrinhalte effektiver und effizienter zu vermitteln. [...] Aus mediendidaktischer Sicht sind besonders folgende mögliche Funktionen des Internets relevant.“ Veranschaulichung und Strukturierung, kognitive und operative Aktivitäten, Wissenskonstruktion und Kommunikation, lernmotivationale Funktion (vgl. Kerres 2000, S. 122f).
- Medienerzieherische Argumente: „Einer handlungsorientierten Medienpädagogik geht es vielmehr darum, die Fähigkeit von Menschen, sich über Medien zu artikulieren, ihre medienbezogenen Informations- und Unterhaltungsbedürfnisse und -interessen zu befriedigen und zu reflektieren sowie Medienprodukte verstehen und bewerten zu können“ (Kerres 2000, S.123).

1.0.4 - Einsatz von Opensource-Software

Als Open-Source¹³ wird Software (Debian, Ubuntu, Firefox¹⁴, Thunderbird¹⁵ ...) bezeichnet, deren Quellcode¹⁶ für jedermann zugänglich ist. Obwohl Open-Source-Software vom Prinzip her betriebssystemunabhängig ist, dürfte es wohl die rasant zunehmende Akzeptanz des freien Betriebssystems Linux¹⁷ gewesen sein, die immer mehr auf das Open-Source-Konzept aufmerksam gemacht hat.

- Freie Weiterverbreitung

Jeder darf Open-Source-Software nutzen und beliebig weiter verteilen.

- Verfügbarkeit des Quellcodes

Das Software-Paket muss den Quellcode enthalten oder angeben, an welcher frei zugänglichen Stelle dieser zu erhalten ist.

-Änderungen am Quellcode

Der Quellcode darf an eigene Bedürfnisse angepasst und in dieser veränderten Form weitergegeben werden.

Open-Source-Software wird in der Regel von engagierten Entwicklern auf freiwilliger Basis und in internationaler Zusammenarbeit erstellt. Dabei bestehen keine kommerziellen Interessen. Die Rechte der Programmierer und Nutzer werden dabei in der GNU General Public Licence (GPL)¹⁸ festgelegt.

Open-Source-Software ist zwar nicht immer kostenfrei, aber kostengünstig. In vielen Fällen ist sie kostenlos erhältlich. Durch den Einsatz von freier Software spart man somit viel Geld und vermeidet teure Lizenzen!

Dabei behält man, entgegen aller gängigen Argumente, die Kompatibilität zu Microsoft-



¹³ <http://de.wikipedia.org/wiki/Opensource>

¹⁴ <http://de.wikipedia.org/wiki/Firefox>

¹⁵ http://de.wikipedia.org/wiki/Mozilla_Thunderbird

¹⁶ <http://de.wikipedia.org/wiki/Quellcode>

¹⁷ <http://de.wikipedia.org/wiki/Linux>

¹⁸ <http://de.wikipedia.org/wiki/Gpl>

Produkte, wie z.B. Office Word¹⁹ oder Excel²⁰. Entweder durch Einsatz von Alternativen (StarOffice²¹, OpenOffice ...) oder durch "Emulation"²² von Windows-Software unter Linux ohne großen Aufwand.

19 http://de.wikipedia.org/wiki/Microsoft_Word

20 http://de.wikipedia.org/wiki/Microsoft_Excel

21 http://de.wikipedia.org/wiki/Star_Office

22 <http://de.wikipedia.org/wiki/Wine>

2. Server

Der Server hat die Aufgabe das Internet für die Schüler zu filtern (Squid-Proxy). Für Mitglieder des VPN-Netzwerkes hingegen ist das Internet frei zugänglich (VPN-Server) und der Server dient zudem als http-Server für ein schulinternes Intranet²³ (apache2). Im speziellen bedeutet dies, dass der gesamte Datenverkehr des Schulnetzwerks durch den Server geschlauft werden muss und somit dieser zum zentralen Knotenpunkt wird.

Als Betriebssystem kommt hier Debian²⁴ zum Einsatz auf dem die entsprechenden Zusatzpakete installiert werden können. Eine grafische Oberfläche²⁵ ist nicht vorgesehen.

Beim ersten Testaufbau des Servers wurde Debian 4 benutzt.

Bei einer eventuellen Neuinstallation kann auch Debian 5 „Lenny“ eingesetzt werden.



• 2.1 Installation der Software

2.1.1 - Betriebssystem

Debian 4/5²⁶ kann unter folgendem Link heruntergeladen und danach auf CD gebrannt werden.



<http://cdimage.debian.org/debian-cd/5.0.1/i386/iso-cd/debian-501-i386-netinst.iso>

Nachdem von der CD die Installation gestartet wurde, muss man Schritt für Schritt den Installationsassistenten folgen: Pakete auswählen, Zeit und Region festlegen, Partitionieren²⁷ und Administratorpasswort setzen.

2.1.2 - Anwendungen

Die Anwendungen die benötigt werden sind: *dhcp*, *apache2*, *squid*, *openvpn*, *ssh*, *php-plugin*, *mysql-server/-client* sowie *proftpd*.

Zuvor müssen jedoch noch die Paketquellen von Debian angepasst werden (hier im Beispiel zu „Etch“). Mit²⁸



```
nano /etc/apt/sources.list
```

muss die Konfiguration wie folgt geändert werden:

²³ <http://de.wikipedia.org/wiki/Intranet>

²⁴ <http://de.wikipedia.org/wiki/Debian>

²⁵ <http://de.wikipedia.org/wiki/GUI>

²⁶ <http://www.debian.org/>

²⁷ <http://www.debian.org/releases/3.0/i386/ch-partitioning.en.html>

²⁸ http://de.wikipedia.org/wiki/Terminal_%28Computer%29



```
#deb cdrom:[Debian GNU/Linux 4.0 r5 _Etch_ - Official i386 NETINST
Binary-1 20081024-15:53]/ etch contrib main
#deb cdrom:[Debian GNU/Linux 4.0 r5 _Etch_ - Official i386 NETINST
Binary-1 20081024-15:53]/ etch contrib main
```

```
deb http://security.debian.org/ etch/updates main contrib
deb-src http://security.debian.org/ etch/updates main contrib
deb http://ftp.de.debian.org/debian etch main contrib non-free
deb http://security.debian.org/ etch/updates main contrib non-free
```

Mit folgenden Befehlen kann man diese installieren:



```
$ apt-get update
```

Paketliste updaten

```
$ apt-get install squid openvpn openssl dhcp ssh Pakete herunterladen und
sshd libapr1 libaprutil1 libdbd-mysql-perl libdbi- installieren
perl libnet-daemon-perl liblprpc-perl libpq5 vim
```

(bei bedarf auch einzelnd,
nacheinander installierbar
)

Alle Anwendungen sollten sich bei korrekter Konfiguration (also wenn keine Fehler vorhanden sind) automatisch beim Systemstart als Daemon²⁹ starten. Während der Installation werden einige wenige Einstellungen abgefragt (werden später dann benötigt!), wie z.B. Benutzername und Passwort für den Zugang zur MYSQL-Datenbank³⁰.

Server per SSH-Remoteverbindung verwalten

Da ab diesen Schritt der SSH-Server³¹ schon als Hintergrunddienst auf dem Server laufen sollte, kann nun der Server von jedem PC aus verwaltet werden, der am selben Netzwerk angeschlossen ist. Unter Windows kann dafür das Programm Putty verwendet werden. Bei gängigen Linux-Systemen, ist ein SSH-Client bereits schon installiert und kann mit

`ssh root@XXX.XXX.XXX.XXX (ServerIP)` verwendet werden.

Ab dem Übernehmen der Iptables-Einstellungen in Schritt 1.2.8, kann der SSH-Port (22) nur innerhalb des VPN-Netzwerkes verwendet werden!

2.1.3 - Netzwerk

Es werden zwei Netzwerkkarten benötigt. Die eine (in diesem Fall `eth0`) ist mit dem Internet verbunden. Die andere (hier `eth1`) ist mit dem Lokalen Netzwerk verbunden, welches gefiltert werden soll es sei denn der Client ist im VPN. Auf `eth1` läuft auch der DHCP-Server. Mit OpenVPN kommt noch ein dritter Netzwerkadapter dazu. Der virtuelle tap0. Über ihn läuft

²⁹ <http://de.wikipedia.org/wiki/Daemon>

³⁰ <http://de.wikipedia.org/wiki/Mysql>

³¹ <http://de.wikipedia.org/wiki/Ssh>

Mit dem Befehl³²:

 `$ ifconfig IFC`

lassen sich die aktuellen Einstellungen der LAN-Verbindung einsehen und verändern. *IFC* hier durch Netzwerkadaptername ersetzen.

• 2.2 Konfiguration


2.2.1 - Netzwerk-Adapter

Als erstes sollten die beiden Netzwerkadapter eingestellt werden. Es ist wichtig, dass der Server eine statische IP-Adresse bekommt, damit man ihn immer unter der gleichen erreichen kann!!

Hierzu wird die Datei */etc/network/interfaces*³³ editiert³⁴:

 `$ nano /etc/network/interfaces`

Die Datei sollte folgendermaßen konfiguriert:



```
# The loopback network interface
auto lo
iface lo inet loopback

# The primary network interface Internet
allow-hotplug eth0
iface eth0 inet static
address 192.168.10.190
netmask 255.255.255.0
gateway 192.168.10.254

#The secondary network interface LAN, filtered by Squid
allow-hotplug eth1
iface eth1 inet static
address 192.168.179.1
netmask 255.255.255.0
```

Die IP-Adresse(n) bei eth0 müssen mit dem Router abgestimmt werden, die bei zwei sind frei wählbar, man muss sich allerdings im weiteren Verlauf der Konfiguration strikt an diese halten.

2.2.2 - DHCP-Server und Hostname

Der DHCP-Server verteilt IP-Adressen an Client, die keine Statische IP haben. Hier wird er benutzt um Clients automatisch in das öffentliche aber gefilterte Netzwerk zu holen.

Seine Konfigurationsdatei liegt in */etc/dhcpd.conf*³⁵. Die wurde wie folgt konfiguriert:

³² <http://www.computerhope.com/unix/uifconfi.htm>

³³ <http://www.cyberciti.biz/faq/setting-up-an-network-interfaces-file/>

³⁴ <http://www.nano-editor.org/>

³⁵ <http://www.daemon-systems.org/man/dhcpd.conf.5.html>


```

;
@      IN      NS      ns1.home.fwsv.
@      IN      NS      ns2.home.fwsv.
home.fwsv.  IN      MX      10      home.fwsv.
home.fwsv.  IN      A       192.168.179.1
ns1         IN      A       192.168.179.1
ns2         IN      A       192.168.179.2
www         IN      CNAME   home.fwsv.
mail        IN      A       192.168.179.1
ftp         IN      CNAME   home.fwsv.
home.fwsv.  IN      TXT     "v=spf1 ip4:192.168.179.1 a mx ~all"
mail        IN      TXT     "v=spf1 a -all"

```

Wie zu sehen, wurden auch „Subdomains“ konfiguriert für *mail*, *ftp* und *nameserver*. Zu beachten ist die Zeile für *SERIAL*. Diese repräsentiert das Änderungsdatum sowie die Versionsnummer: 2007011501 = (2007)(01)(15)(01) = 15.01.2007, Version 01. Mit jeder Änderung sollte diese Variable angepasst werden, damit Bind9 sie übernimmt! Der Befehl *rndc reload* kann zudem ausgeführt werden, um Bind9 zu updaten.

Auch wenn nicht immer nötig, wird aus kompatibilitätsgründen auch ein sogenannter RDNS (Reverse DNS)-Eintrag angelegt mit dem Editor *vim*: *vim /etc/bind/zones/master/192.168.179.rev*

```

$TTL 1d ;
$ORIGIN 192.168.192.IN-ADDR.ARPA.
@      IN      SOA      ns1.home.fwsv.  info.home.fwsv. (
                                2007011501
                                7200
                                120
                                2419200
                                604800
)
      IN      NS       ns1.home.fwsv.
      IN      NS       ns2.home.fwsv.
1      IN      PTR     ns1.home.fwsv.
2      IN      PTR     ns2.home.fwsv.

```

Domainnames und die Reverse-IP bei \$ORIGIN wurden dementsprechend angepasst. Die Einstellungen können mit folgendem Befehl auf ihre Vollständigkeit und Korrektheit überprüft werden. Sollte das Programm nicht mit einem „OK status“ antworten ist z.B. nochmals der Syntax zu überprüfen, da hier die meisten Fehler sich einschleichen.

```

$ cd /etc/bind/zones/master/
$ named-checkzone home.fwsv home.fwsv.db

```

Jetzt werden die „Zonen-Dateien“ in die Hauptkonfigurationsdatei von Bind9 eingetragen. Erst damit werden die Änderungen überhaupt mit eingebunden: *vim /etc/bind/named.conf.local*

```

zone "home.fwsv" {
    type master;
    file "/etc/bind/zones/master/home.fwsv.db";
};


zone "192.168.192.IN-ADDR.ARPA" {
    type master;
    file "/etc/bind/zones/master/192.168.179.rev";
};

```

Nun kann Bind9 mit folgendem Befehl neugestartet werden:


 `$ /etc/init.d/bind9 restart`

Folgende Befehle können nun an Client-Rechnern benutzt werden, um die vollständige Funktionalität von Bind9 zu überprüfen:

 `$ ping ns1.home.fwsv
$ nslookup
> ns1.home.fwsv
$ dig @192.168.179.1 home.fwsv`

2.2.4 - Apache-Server

Apache ist ein http-Server, der in unserem Fall im Lokalen Netzwerk ein kleines Intranet zur Verfügung stellen soll. Installiert wird er mit folgendem Befehl:

 `$ aptitude install apache2 php5-common php5-mysql apache2-mpm-
prefork apache2-utils apache2.2-common libapache2-mod-php5`


Standardmäßig liegen die Dateien der Intranetseite unter `/var/www/`. Dorthin kann auch die Beispiel-Intranetseite kopiert werden, zu finden bei den beiliegenden Daten der Dokumentation. Dabei sollte Wordpress von <http://www.wordpress.org> heruntergeladen und nach `/var/www/wordpress` entpackt werden. Das passende Theme liegt der Dokumentation bei.

Die Konfigurationsdateien von Apache liegen unter `/etc/apache2/`. In der Datei `apache2.conf`³⁶ liegen allgemeine Einstellungen, die eigentlich keiner Bearbeitung bedürfen. Jedoch sollte man in `/etc/apache2/conf.d/fqdn` den Servername auf `localhost` setzen mit der Zeile: `ServerName localhost`

Im Unterordner `/sites-available/` liegt die Datei `default` in der sich der Standardpfad der freigegebenen Dateien ändern lässt.

Einstellungen lassen sich direkt übernehmen in dem man `/etc/init.d/apache2 restart` ausführt und somit den Daemon neustartet

2.2.5 - MySQL-Datenbank einrichten

 `$ aptitude install mysql-client-5.0 mysql-common mysql-server mysql-
server-5.0 libmysqlclient15off`

Da z.B. der Blog des Intranets eine MySQL-Datenbank benötigt, richten wir diese über den MySQL-Client ein. Dazu wechselt man in die Eingabeaufforderung der MySQL-Console mit: `$ mysql -u root -p`

Danach wird das Passwort verlangt, dass man bei der Installation des Servers angelegt hat. In der "Shell", setzen wir die passenden "Queries" - hier ein Beispiel zur Einrichtung einer Datenbank für Wordpress (anhand der Konfiguration von `/var/www/wordpress/wp-`

36 <http://httpd.apache.org/docs/2.0/mod/core.html>

config.php):

```
mysql> CREATE DATABASE wordpress;
Query OK, 1 row affected (0.00 sec)
mysql> GRANT ALL PRIVILEGES ON wordpress.* TO
"wordpressusername"@"localhost"
-> IDENTIFIED BY "password";
Query OK, 0 rows affected (0.00 sec)
mysql> FLUSH PRIVILEGES;
Query OK, 0 rows affected (0.01 sec)
mysql> EXIT
```

2.2.6 - FTP-Server

```
$ aptitude install proftpd
```

Ein FTP ist notwendig um die Intranetseite des Servers zu editieren und zu verwalten. Das Programm ProFTP³⁷ wurde in Schritt 2.1.1 schon installiert. Mit dem Editor *nano* muss die Datei */etc/shells* aufgerufen und „*/bin/false*“ am Schluss angefügt werden.

Mit „*useradd ftpuser -p your_password -d /var/www -s /bin/false*“ wird nun ein neuer Benutzer erstellt. In der Datei */etc/proftpd/proftpd.conf* wird nun folgender text eingetragen/angepasst:

```
Include /etc/proftpd/modules.conf

# Set off to disable IPv6 support which is annoying on IPv4 only boxes.
UseIPv6 off
ServerName "home.fwsv"
ServerType standalone
DeferWelcome off
MultilineRFC2228 on
DefaultServer on
ShowSymLinks on
TimeoutNoTransfer 600
TimeoutStalled 600
TimeoutIdle 1200
DisplayLogin welcome.msg
DisplayFirstChdir .message
ListOptions "-l"
DenyFilter \*.*

# Port 21 is the standard FTP port.
Port 21
MaxInstances 8

# Set the user and group that the server normally runs at.
User nobody
Group nogroup

# Umask 022 is a good standard umask to prevent new files and dirs
# (second parm) from being group and world writable.
Umask 022 022
PersistentPasswd off
MaxClients 8
MaxClientsPerHost 8
MaxClientsPerUser 8
MaxHostsPerUser 8

# Display a message after a successful login
AccessGrantMsg "welcome !!!"
```

37 <http://www.proftpd.org/>

```
# This message is displayed for each access good or not
ServerIdent      on      "you're at home"
# Set /home/FTP-shared directory as home directory
DefaultRoot /var/www
# Lock all the users in home directory, ***** really important *****
DefaultRoot ~
MaxLoginAttempts 5
#VALID LOGINS
<Limit LOGIN>
AllowUser ftpuser
DenyALL
</Limit>
<Directory /var/www>
Umask 022 022
AllowOverwrite on
  <Limit READ WRITE DIRS CDUP MKD STOR DELE XMKD RNRF RNTD RMD XRMD>
  AllowAll
  </Limit>
</Directory>
```

Wichtig ist, dass der von FTP freigegebene Ordner, in diesem Fall `/var/www` mit vollen Rechten ausgestattet wird. Die geschieht mit dem Befehl `chmod 777 /var/www`.
Alle Einstellungen werden übernommen mit: `/etc/init.d/proftpd restart`

2.2.7 - Munin - Systemüberwachung

Munin³⁸ ist ein Programm zur Status-Überwachung von z.B. Traffic (Datendurchsatz bei verschiedenen Netzwerkgeräten), Festplattenkapazität oder Prozessoraktivität beliebig vieler Rechner. Das Programm fasst dann alle Informationen auf einer Webseite zusammen.
Auf den Server installieren wir mit diesem Befehl den Client- und Serverteil von Munin mit diesem Befehl:



```
$ apt-get install munin-node munin liblwp-protocol-http-socketunix-perl
liblwp2 lsof
```

Folgende Konfigurationsdateien müssen editiert werden, damit der Server seine eigenen Informationen abrufen und sammelt – als erstes die Datei `/etc/munin/munin.conf`:



```
dbdir /var/lib/munin
htmldir /var/www/munin
logdir /var/log/munin
rundir /var/run/munin

tmpldir /etc/munin/templates

[server.fwsv]
address 127.0.0.1
use_node_name yes
```

Die zweite Konfigurationsdatei `/etc/munin/munin-node` muss wie folgt angepasst werden:



```
log_level 4
log_file /var/log/munin/munin-node.log
```

38 <http://munin.projects.linpro.no/>

```

port 4949
pid_file /var/run/munin/munin-node.pid
background 1
setseid 1
host *
user munin #root
group munin #root
setsid yes

ignore_file ~$
ignore_file \bak$
ignore_file %$
ignore_file \dpkg-(tmp|new|old|dist)$
ignore_file \rpm(save|new)$
host_name localhost
allow ^127\.\.\.1$
allow ^127\.\.\.1$

```

Achtung: Munin funktionierte bei unserem Setup erst dann, nachdem die Dateibefugnisse folgender Ordner geändert wurden:



```
$ chown munin:munin -R /var/lib/munin /var/www/munin /var/log/munin
/var/run/munin
```

Bevor Munin gestartet wird, werden folgende Plugins für das Überwachen der OpenVPN-Netzwerke geladen:



```
$ wget http://muninexchange.projects.linpro.no/download.php?phid=62
-O /usr/share/munin/plugins/ovpn1
$ wget http://muninexchange.projects.linpro.no/download.php?phid=62
-O /usr/share/munin/plugins/ovpn2
```

In beiden Dateien muss für das jeweilige OpenVPN-Netzwerk folgende Parameter geändert werden:

```
my $statuslogfile = "/etc/openvpn/ovpn*-status.log";
```

Bei dem Plugin *ovpn1* heißt die Logfile *ovpn1-status.log*, bei *ovpn2* *ovpn2-status.log*. Sobald der OpenVPN-Daemon läuft und somit die Dateien erstellt sind, muss die Logfile für Munin zugänglich machen:



```
$ chmod o+r /etc/openvpn/ovpn1-status.log
$ a2enmod status
```

Die einzelnen Plugins zur Überwachung verschiedener Dienste können mit diesen Befehlen aktiviert werden:

```
$ munin-node-configure --suggest --shell | sh -
$ ln -s /usr/share/munin/plugins/ping_ /etc/munin/plugins/ping_8.8.8.8
```



```
$ ln -s /usr/share/munin/plugins/if_/etc/munin/plugins/if_tap0
$ ln -s /usr/share/munin/plugins/if_/etc/munin/plugins/if_tap1
$ ln -s /usr/share/munin/plugins/apache_accesses/etc/munin/plugins/
$ ln -s /usr/share/munin/plugins/apache_processes/etc/munin/plugins/
$ ln -s /usr/share/munin/plugins/apache_volume/etc/munin/plugins/
```

In der Datei `/etc/apache2/apache2.conf` muss sichergestellt werden, dass folgende Passagen auskommentiert und wirksam werden:



```
SetHandler server-status
Order deny,allow
Deny from all
Allow from 127.0.0.1
```

```
ExtendedStatus On
```

Das provisorisch selbsterstellte PPTPD-Munin-Plugin lässt in folgenden Schritten einrichten: Als erstes wird ein systemweiter Cronjob angelegt, der in regelmäßigen Abständen eine Logfile anlegt und darin die Anzahl der gerade eingelogten PPTP-Benutzern hinterlegt. Die Datei `/etc/crontab` wird dazu wie folgt angepasst (die letzte Zeile ist dabei unser hinzugefügter Befehl):



```
SHELL=/bin/sh
PATH=/root:/usr/local/sbin:/usr/local/bin:/sbin:/bin:/usr/sbin:/usr/bin

# m h dom mon dow user  command
17 * * * * root    cd / && run-parts --report /etc/cron.hourly
25 6 * * * root    test -x /usr/sbin/anacron || ( cd / && run-parts --report /etc/cron.daily )
47 6 * * 7 root    test -x /usr/sbin/anacron || ( cd / && run-parts --report /etc/cron.weekly )
52 6 1 * * root    test -x /usr/sbin/anacron || ( cd / && run-parts --report /etc/cron.monthly )
39 19 * * * root    bash /root/backup.sh
*/5 * * * * root    echo "PPTPD.value ""Isot -i:1723 | grep pptpctrl | wc -l"".00" > /var/log/pptpusercount
#
```

Eine neue Datei, das Munin-Plugin, wird erstellt mit z.B. `vim /etc/munin/plugins/pptpd` und diesem Inhalt:



```
#!/bin/sh

if [ "$1" = "autoconf" ]; then
    echo yes
    exit 0
fi

if [ "$1" = "config" ]; then
    echo 'graph_title PPTPD'
    echo 'graph_args --base 1000 -l 0 '
    echo 'graph_vlabel clients'
    echo 'graph_category network'
    echo 'PPTPD.label pptpd'
    exit 0
fi

env LC_ALL=C cat /var/log/pptpusercount
```

Munin muss neugestartet werden, damit alle Änderungen aktiv werden:



```
$ /etc/init.d/munin-node restart
```

HTTP-Basic-Auth Setup

Um die Munin-Seite mit einem Passwort zu sichern, legt man zuerst eine Passwortdatei an.

```
$htpasswd -c /usr/local/apache/passwd/passwords <username>
```

Anschließend wird das Passwort für den angegebenen Benutzer eingegeben, dies wird in der angegebenen Datei verschlüsselt gespeichert.

Die Rechte für die Passwortdatei sollte stark beschränkt werden, um unerlaubten Zugriff zu vermeiden.

Achtung! Bei Fehlfunktion ist die Lesebeschränkung eine häufige Fehlerquelle.

Zum Schluss wird in die Apache-Config-Datei (/etc/apache2/sites-enabled/000-default) ein Eintrag für jeden Ordner gemacht, der gesichert werden soll. Ein Beispiel hierfür sieht so aus.



```
<Directory /usr/local/apache/htdocs/sekrit>
AuthType Basic
AuthName "By Invitation Only"
AuthUserFile /usr/local/apache/passwd/passwords
Require user <username>
</Directory>
```

2.2.8 - Samba-Freigaben

Unter Samba-Freigaben versteht man die normalen Windowsfreigaben.

Mit `apt-get install samba` wird der Dienst installiert und die Konfigurationsdatei liegt unter `/etc/samba/smb.conf`³⁹.

Da wir mehrere Bereiche haben wollen, von denen einige nur aus dem VPN zu erreichen sein sollen, benutzen wir sogenannte „Virtuelle Hosts.“ Das heißt es laufen mehrere virtuelle Samba-Server auf einem physikalischen Server.

Hierzu wird die `smb.conf` komplett gelöscht (vorher Backup erstellen!!) und anschließend durch folgenden Text ersetzt.



```
[global]
workgroup = FWS-VAIHINGEN
netbios aliases = datenserver, schueler, admin
server string = SMB Server for %L
security = SHARE
log file = /var/log/samba/%m.log
max log size = 50
smb ports = 139
socket options = TCP_NODELAY SO_RCVBUF=8192 SO_SNDBUF=8192
disable spoolss = Yes
include = /etc/samba/smb.conf.%L
```

Dies ist die allgemein gültige Config-Datei. Für die Namen, die unter „*netbios aliases*“ angegeben werden, werden nun neue Config-Dateien angelegt! Diese werden `smb.conf.name`

39 <http://www.samba.org/samba/docs/man/manpages-3/smb.conf.5.html>

genannt. Der Inhalt für VPN-Geschützte Bereiche sollte so aussehen:



```
[global]
wide links = no
workgroup = fws-vaihingen
hosts allow = 10.8.0.0/24
hosts deny = 0.0.0.0/0

[lehrer]
path = /mnt/ehdd/lehrer/
printable = no
browsable = yes
writable = yes
valid users = lehrer
read only = no
guest ok = no
```

Der Inhalt für öffentliche Bereiche so:



```
[global]
wide links = no
workgroup = fws-vaihingen

[Schueler Bereich]
path = /mnt/ehdd/schueler
printable = no
browsable = yes
writable = yes
guest ok = yes
```

Der Bereich unter „[Schueler Bereich]“ kann beliebig oft kopiert und angepasst werden.

2.2.9 - SQUID-Proxy

Squid ist der Proxy-Server. Er soll alle Webanfragen aus dem öffentlichen Netz abfangen und filtern, während er die Anfragen aus dem VPN-Netz nicht beachten soll.

Squid hat eine mächtige Config-Datei die unter `/etc/squid/squid.conf`⁴⁰ zu finden ist. Da die Datei sehr viel Text beinhaltet und der meiste davon für uns bedeutungslos ist, hier nur die wichtigen Auszüge.


Als erstes müssen IPs und Ports richtig gesetzt werden: In der Datei findet man diese Einstellung unter dem TAG: `http_port` (gleich am Anfang).



```
http_port 0.0.0.0:3128 transparent
tcp_outgoing_address 0.0.0.0
udp_outgoing_address 0.0.0.0
```

Die Einstellung `transparent` bewirkt, dass der Proxy an den Clients nicht per Hand eingetragen werden muss, sondern der Proxy hört alles ab, was über seinen Port 3128 läuft. Der `http` Port 80 des Servers wird später in der `IPTables` auf Port 3128 umgeleitet.

Ein weiterer wichtiger Teil ist unter TAG: `acl` zu finden. `acl` steht für Access List. Hier kann man IP-Ranges den Zugriff auf den Proxy erlauben oder verbieten. Außerdem kann man hier eine Whitelist definieren.



```
acl all src 192.168.179.0/255.255.255.0
acl errorpage dst 192.168.179.1/255.255.255.255
acl whitelist dstdomain "/etc/squid/whitelist"

deny_info http://192.168.179.1/error.php all

http_access allow all whitelist
http_access allow errorpage
http_access deny all
```

In der Datei „*/etc/squid/whitelist*“ werden nun alle Domainnamen eingetragen die im Internet zugänglich sein sollen. Z. B.: **.wikipedia.de** wobei der Punkt am Anfang der Adresse zu beachten ist. Unter *deny_info*, wird die URL zu der Errorpage angegeben für die Regel „*all*“.

Unsere Whitelist (ein Ausschnitt davon) sieht so aus:



```
.dhm.de
.fwsv
.periodictable.com
.spiegel-online.de
.bahn.de
.encarta.com
.encarta-de
.wikipedia.org
.wikipedia.fr
.wikipedia.de
.leo.org
.wikimedia.org
.zeit.de
.stern.de
.focus.de
.heise.de
.golem.de
.wikileaks.org
.ubuntuusers.de
.dastelefonbuch.de
.ccc.de
.bbc.co.uk
.cnn.com
.nytimes.com
.guardian.co.uk
.telegraph.co.uk
.nature.com
.technologyreview.com
.sciencemag.org
.waldorfschule-vaihingen.de
.kernel.org
.ubuntu.com
.tagesschau.de
.debian.org
.entropia.de
.dict.cc
.openstreetmap.org
```

```
.wiktioary.org
.wikibooks.org
.wikiiversity.org
.wikinews.org
.wikiquote.org
.wikisource.org
.sueddeutsche.de
.spiegel.de
.zivildienst-stellen.net
.mozilla.org
.hausaufgaben.de
.spektrumdirekt.de
.weltderphysik.de
.pro-physik.de
.nasa.gov
.hdg.de
.brockhaus.de
.wga.hu
.pons.de
.pons.eu
```

Die ganze Whitelist ist im Dokumentationsordner zu finden! Unter `/usr/share/squid/errors/` sind die verschieden Error-Seiten zu finden. Einstellen welcher error Ordner benutzt wird kann man in der *squid.conf* unter dem „TAG: error_directory“.

2.2.10 - OpenVPN-Server

OpenVPN sorgt dafür, dass alle Rechner die sich am VPN anmelden komplett verschlüsselt ihre Daten übertragen. Sie sind somit von Rechner außerhalb des Vpn nicht zu sehen! In unserem Fall sollen 2 VPNs erstellt werden für 2 völlig voneinander getrennte Netzwerkbereiche.

Man loggt sich in ein VPN mit einem speziell für jeden Rechner erstellten Zertifikat⁴¹ ein. Es ist **sehr wichtig**, dass die Zertifikate geschützt werden und nicht in **falsche Hände** geraten, da **JEDER** sich mit einem Zertifikat ins VPN einloggen kann.

Nun zum Erstellen der VPN-Zertifikate:

Der Einfachheit halber erstellen wir die Zertifikate unter Windows.

Nachdem OpenVPN (bei <http://www.openvpn.net>) heruntergeladen und installiert wurde, öffnet man eine Windows-Konsole⁴² (Start → Ausführen → „cmd“).

Man wechselt im OpenVPN Installationsordner in den Unterordner „easy-rsa“ und führt als erstes die Bat-Datei „init-config.bat“ aus. Die bereitet alles für die Zertifikatbildung vor.

Anschließend wird „vars.bat“ und „clean-all.bat“ ausgeführt. Diese löscht alle temporären Dateien die eventuell bei der Installation stören könnten.

Nun beginnt das Erstellen der Zertifikate. Dazu wird die Datei „build-ca“ gestartet, die das allgemein gültige Zertifikat erstellt. Diese wird von allen Clients sowie dem Server benötigt. Man gibt die Angeforderten Informationen ein. Wichtig ist die Eingabe „Common Name“ es darf in einem physikalischen Netzwerk keine zwei VPNs mit dem selben Common Name existieren.

⁴¹ <http://openvpn.net/index.php/open-source/documentation/howto.html#pki>

⁴² <http://ss64.com/nt/>

Jedes weitere Zertifikat bekommt nun seinen eigenen „Common Name.“ Es darf nicht zweimal der gleiche „Common Name“ in einem VPN benutzt werden.

Als nächstes wird das spezielle Zertifikat für den Server erstellt:

„build-key-server.bat <Name_des_Zertifikats(z. B. server01)>“

Auch hier werden die entsprechenden Infos eingetragen.

Nun folgen die, Zertifikate für die Clients. Dieser Vorgang kann beliebig oft wiederholt werden, je nach dem wie viele benötigt werden. Wichtig ist natürlich jedes mal einen anderen Namen zu benutzen. Hierzu wird die Datei „build-key <Name_des_Zertifikats(z. B. client01)>“ aufgerufen.

Die nun folgende Abfrage ist schon von den anderen Zertifikaten bekannt.

Als letztes wird der Diffi Hellman⁴³ Parameter gebildet. Dieser ist nur für den Server und wird nicht auf den Clients benötigt: „build-dh“.

Die Erstellten Dateien befinden sich nun im unterordner Keys. Sie werden entsprechend auf den Clients und dem Server Verteilt.

Beispiel:

	server.crt		
	server.key		Diese Dateien benötigt
	dh1024.pem		der Server
	ca.crt		

Zum Abschluss müssen noch auf Server und Client jeweilige Konfigurationsdateien angelegt werden. Die Konfigurationsdateien haben immer die Endung .conf auf dem Server und .ovpn im Windows.


SERVER-Config-File:

	port 1194	#Port muss für jedes VPN-Netz geändert werden
	proto udp	
	dev tap0	
	mode server	
	tls-server	
	server 10.8.0.0 255.255.255.0	#IP Ranges des VPN
	client-to-client	
	ca /etc/openvpn/ca.crt	#Die Zertifikatdateien
	cert /etc/openvpn/server-01.crt	
	key /etc/openvpn/server-01.key	
	dh /etc/openvpn/dh1024.pem	
	auth SHA1	
	cipher AES-256-CBC	
	push "redirect-gateway def1"	
	ping 10	
	push "ping 10"	

43 <http://de.wikipedia.org/wiki/Diffie-Hellman-Schl%C3%BCsselaustausch>

```
ping-restart 60
push "ping-restart 60"
status /etc/openvpn/ovpn1-status.log
```

CLIENT-Config-File:



```
# Grundsätzliches (Was soll der CLIENT nutzen)
port 1194
proto udp
dev tap
# Client-Einstellungen
tls-client
ns-cert-type server
remote 192.168.179.1 1194           #Die Adresse des Server
(NICHT                             die VPN-Adresse)
ca ca.crt                         # Authentifizierung und
                                   Verschlüsselung
cert client01.crt
key client01.key
auth SHA1
cipher AES-256-CBC
# Sonstiges
pull
```

Damit der OpenVPN-Server automatisch bei jedem Start geladen wird, und dabei nicht den Terminal blockiert, verschiebt man die Debug-Ausgabe des VPN-Programms in sogenannte Screens⁴⁴ (virtuelle Terminals) und legt dafür den richtig Starteintrag in z.B. /etc/init.d/rc.local an:

```
$ screen -A -m -d -S ovpn1 openvpn --config /etc/openvpn/vpnserver1.conf --log
/var/log/openvpn/ovpn1.log
$ screen -A -m -d -S ovpn2 openvpn --config /etc/openvpn/vpnserver2.conf --log
/var/log/openvpn/ovpn2.log
```

Die Parameter starten 2 virtuelle Consolen auf denen im Hintergrund die OVPN Daemons laufen (im non-daemon mode->live-log). Laufende Screens lassen sich mit `screen -ls` auflisten und mit `screen -r` wieder vorholen. Einen `screen` bringt man mit CTRL+A und dann d wieder in den Background.

2.2.11 - IPTables2-Firewall

Die IPTables⁴⁵ sind sozusagen die Firewall⁴⁶ des Servers. Außerdem sind sie verantwortlich den Traffic zwischen den beiden Netzwerkkarten hin und her zu routen.

Ziel ist es, dass alle Ports des Servers gesperrt werden, bis auf diejenigen, die man benötigt für

⁴⁴ <http://www.rackaid.com/resources/linux-screen-tutorial-and-how-to/>

⁴⁵ <http://linux.die.net/man/8/iptables>

⁴⁶ <http://de.wikipedia.org/wiki/Firewall>

den transparenten Proxy.

Folgender Text wird unter `/etc/init.d/iptables` gespeichert.

```
#!/bin/sh

IPT=/sbin/iptables
echo "IP-Tables Script"
case "$1" in
start)
sysctl -w net/ipv4/ip_forward=1

#New Custom Chain
$IPT -N genericchain
$IPT -N eth1
$IPT -N tap0
$IPT -N tap1
$IPT -N reject

#eth1 prerouting
$IPT -A PREROUTING -t nat -i eth1 -p tcp --dport 80 -j REDIRECT --to-port 3128
$IPT -A PREROUTING -t nat -i eth1 -p tcp --dport 443 -j REDIRECT --to-port 3128

#tap0/tap1 nat
$IPT -A POSTROUTING -t nat -o eth0 -j MASQUERADE

#reject eth1 forwarding
$IPT -A FORWARD -i eth1 -j REJECT

#genericchain, all interfaces
$IPT -A genericchain -m state --state ESTABLISHED,RELATED -j ACCEPT
$IPT -A genericchain -p tcp --dport 3128 -j ACCEPT #SQUID
$IPT -A genericchain -p tcp --dport 1723 -j ACCEPT #PPTP
$IPT -A genericchain -p tcp --dport 47 -j ACCEPT #PPTP
$IPT -A genericchain -p tcp --dport 80 -j ACCEPT #HTTP
$IPT -A genericchain -p tcp --dport 443 -j ACCEPT #SSL
$IPT -A genericchain -p udp --dport 53 -j ACCEPT #DNS-UDP
$IPT -A genericchain -p tcp --dport 53 -j ACCEPT #DNS-TCP
$IPT -A genericchain -p tcp --dport 22 -j ACCEPT #SSH
$IPT -A genericchain -p tcp --dport 953 -j ACCEPT #bindrndc

#tap0
$IPT -A tap0 -j ACCEPT

#tap1
$IPT -A tap1 -j ACCEPT

#eth1
$IPT -A eth1 -p udp --dport 1194:1195 -j ACCEPT
$IPT -A eth1 -p icmp -s 0/0 -m state --state NEW,ESTABLISHED,RELATED -j ACCEPT
SERVER_IP="192.168.179.1"
iptables -A INPUT -p icmp --icmp-type 8 -s 0/0 -d $SERVER_IP -m state --state NEW,ESTABLISHED,RELATED -j ACCEPT
iptables -A OUTPUT -p icmp --icmp-type 0 -s $SERVER_IP -d 0/0 -m state --state ESTABLISHED,RELATED -j ACCEPT
iptables -A OUTPUT -p icmp --icmp-type 8 -s $SERVER_IP -d 0/0 -m state --state NEW,ESTABLISHED,RELATED -j ACCEPT
iptables -A INPUT -p icmp --icmp-type 0 -s 0/0 -d $SERVER_IP -m state --state ESTABLISHED,RELATED -j ACCEPT
iptables -A INPUT -p tcp --dport 4949 -j ACCEPT

#reject all non-matched requests
$IPT -A reject -j REJECT

#add chains to input
$IPT -A INPUT -j genericchain
$IPT -A INPUT -i eth1 -j eth1
$IPT -A INPUT -i tap0 -j tap0
$IPT -A INPUT -i tap1 -j tap1
$IPT -A INPUT -j reject
```



```

exit 0
;;

stop)
sysctl -w net/ipv4/ip_forward=0
$IPT -t nat -F
$IPT -F INPUT
$IPT -F genericchain
$IPT -F tap0
$IPT -F tap1
$IPT -F eth1
$IPT -F reject

$IPT -t nat -X
$IPT -X genericchain
$IPT -X tap0
$IPT -X tap1
$IPT -X eth1
$IPT -X reject
$IPT -F OUTPUT
$IPT -F FORWARD
$IPT -F INPUT
exit 0
;;
*)
echo "Usage: /etc/init.d/iptables {start|stop}"
exit 1
;;
esac

```

Alle Anfragen im Netzwerk bezüglich HTTP-Traffics werden jetzt auf den Standard-SQUID-Port 3128 weitergeleitet. Der Port 53 (UDP) wird für den DNS-Server benötigt und Port 1194 sowie 1195 (beide UDP) werden von den VPN-Verbindungen genutzt. Nur ein Nutzer, der sich in das VPN-Netzwerk einklingt, hat fortan kompletten Zugriff auf alle Ports und Services.

Mit `chmod47 a+x /etc/init.d/iptables`, wird das Script ausführbar gemacht. Somit kann man die Firewall-Einstellungen mit `/etc/init.d/iptables start` setzen und mit `/etc/init.d/iptables stop` deaktivieren. Um die Einstellungen bei Start zu laden, sollten folgende Zeilen in die Datei `/etc/rc.local` hinzugefügt werden:



```

$ /etc/init.d/iptables stop
$ /etc/init.d/iptables start

```

2.2.12 - Rsync⁴⁸ Backup

Als Backup Speicher wird ein NAS benutzt welches sich hinter dem Server befindet. Es ist also an die „Internet Netzwerkkarte“ des Servers angeschlossen.

Unter `/root/backup.sh` wird ein Script mit folgendem Inhalt angelegt:



```

#!/bin/sh
mount -t smbfs -o password=empty //192.168.10.114/Sicherung
/mnt/smbbackup
screen -A -m -d -S rsync rsync -r --delete --progress -a /mnt/ehdd

```

⁴⁷ <http://linux.die.net/man/1/chmod>

⁴⁸ <http://linux.die.net/man/1/chmod>

```
/mnt/smbbackup  
umount /mnt/smbbackup
```

Und mit folgendem Befehl ausführbar gemacht:



```
$ chmod a+x /root/backup.sh
```

Anschließend wird ein Cronjob⁴⁹ für das Backup angelegt. Dieser sorgt dafür, dass das Backup jeden Tag oder jede Woche zu einer bestimmten Uhrzeit ausgeführt wird.

Hierzu wird der Befehl `crontab -e` ausgeführt und folgende Zeile in den nun erscheinenden Editor eingefügt:

```
0 23 * * * root    bash /root/backup.sh
```

So wird jeden Tag um 23:00 Uhr ein Backup erstellt, welches nur neue Dateien kopiert und gelöschte auch im Backup löscht.


2.2.13 - pptpd

Der pptpd-Server soll „Gastzugänge“ ermöglichen, vorzugsweise für Windows-Systeme. D.h., dass Besucher der Schule sehr schnell, einfach und ohne zusätzliche Software, eine VPN-Verbindung mit Windows zum Server aufbauen können. Gäste sind weiterhin vom gesicherten OpenVPN-Netzwerk getrennt, haben aber vollen Internetzugang. Der Server kann unkompliziert, temporär eingesetzt werden – je nach bedarf. Installiert wird er mit folgendem Befehl:



```
$ aptitude install pptpd
```

Folgende Zeilen müssen geändert bzw. auskommentiert werden in der pptpd-Konfigurationsdatei, mit: `vim /etc/pptpd.conf`:



```
localip 192.168.179.1  
remoteip 192.168.179.234-238,192.168.179.245
```

Die „localip“ wird auf die des Servers angepasst, wohingegen „remoteip“ die IP-Range der PPTP-Client definiert. Die Einstellungen werden übernommen mit:



```
$ /etc/init.d/pptpd restart
```

PPTP-Benutzerkonten werden wie folgt hinzugefügt:



```
$ echo "username pptpd password *" >> /etc/ppp/chap-secrets
```

49 <http://www.manpagez.com/man/5/crontab/>

Username könnte hierbei z.B. „gast“ sein und password „gastpassword“. Die Änderungen sind sofort aktiv!

3. Client

Allgemeines

Der Client dient als Internetzugang für die Schüler. Im Interesse des Opensource Gedanken kommt als Betriebssystem das kostenlose, nicht-kommerzielle und freie *Ubuntu*⁵⁰ 9.04 *Jaunty Jackalope* mit dem *Edubuntu-Addon* zum Einsatz. Das Surfen und Arbeiten mit Schulsoftware und Office-Programmen soll hier in einem eingeschränkten Konto ohne Zugriff auf das System stattfinden. „XX“ steht hier für die jeweilige Clientnummer.

Verwendete Abkürzungen:

IFC = Interface, steht für den Name eines Netzwekadaptors
 XX = z.B. 01,02,05... Nummer eines Clients, Servers, Kontos oder anderem.

• 3.1 Installation der Software

Ubuntu 9.04 Jaunty kann unter folgender Internetadresse heruntergeladen werden:



<http://releases.ubuntu.com/9.04/ubuntu-9.04-desktop-i386.iso>

Die Datei kann wahlweise auf eine CD gebrannt ohne entsprechend auf einen USB-Stick kopiert werden⁵¹ und wird nun auf dem Client-Rechner installiert.

Als Computernamen wird in der Installation bereits *clientXX* angegeben und ein gleichnamiges Konto mit '*clientXX*' als Passwort wird angelegt. Administrative Aufgaben werden später direkt von '*root*' Konto aus erledigt. Der Rest der Installation ist selbsterklärend.

Nach der Installation, muss im Menü unter „System > Systemverwaltung > Software-Quellen“⁵² noch die Quellen „multiverse“ und „universe“ hinzugefügt werden. Im Terminal („Anwendungen → Zubehör“) müssen die Paketquellen aktualisiert und ein Update ausgeführt werden mit:



```
$ sudo apt-get update
$ sudo apt-get upgrade
```

Die Programme, unter anderem *pessulus*⁵³, *epiphany*⁵⁴, *Codecs* und der *Edubuntu-Desktop*, werden benötigt und installiert per Konsole:

⁵⁰ <http://www.ubuntu.com/>

⁵¹ <http://unetbootin.sourceforge.net/>

⁵² <http://wiki.ubuntuusers.de/Paketquellen>

⁵³ <http://live.gnome.org/Pessulus>

⁵⁴ <http://projects.gnome.org/epiphany/>



```
$ sudo apt-get install pessulus epiphany-browser openvpn flashplugin-
nonfree edubuntu-desktop libxvidcore4 gstreamer0.10-plugins-base
gstreamer0.10-plugins-good gstreamer0.10-plugins-ugly gstreamer0.10-
plugins-ugly-multiverse gstreamer0.10-plugins-bad gstreamer0.10-plugins-
bad-multiverse gstreamer0.10-ffmpeg gstreamer0.10-pitfdll libdvdrad4
libquicktime1 openjdk-6-jre icedtea6-plugin qt octave
$ sudo aptitude purge firefox-3.0 pidgin evolution
```

◦ 3.2 Einrichten der Software

3.2.1 - Konten einrichten

Als erstes ist es wichtig, dass das 'root' Konto sich auch über die Normale Benutzeranmeldung anmelden kann. Unter *System->Systemverwaltung->Anmeldefenster* im Reiter Sicherheit die automatische Anmeldung des *clientXX* Kontos aktivieren sowie die automatische Wiederanmeldung nach 10 Sekunden. Zur weiteren Einschränkung kann noch die Option *'Benutzern erlauben Schriften und Farben zu ändern'* im Reiter Barrierefreiheit deaktiviert werden.

Wichtig: ist nun, dass die Option *'Lokalen Systemadministratoren erlauben sich einzuloggen'* aktiviert wird und dass unter *'System->Systemverwaltung->Benutzer und Gruppen'* ⁵⁵ das Konto 'root' entsperrt und ein kompliziertes Passwort gesetzt wird.

In der *Gruppenverwaltung* wird bei der Gruppe 'root' und 'admin' der Benutzer 'root' aktiviert und der Benutzer 'clientXX' deaktiviert.

3.2.1 - Konten anpassen

Jetzt den Rechner neustarten und überprüfen ob alles funktioniert hat (automatische Anmeldung, automatische Wiederanmeldung, manuelle Anmeldung als 'root').

Im 'clientXX' Konto kann nun alles entsprechend konfiguriert werden:

- *Panel anpassen, z.B. nur ein transparenter Panel unten*
- *Zeitanzeige anpassen (nur Uhrzeit anzeigen) – Rechtsklick auf Uhr → Einstellungen*
- *Rechtsklick auf leere Fläche im Pannel → „Zum Panel hinzufügen ...“ auswählen und z.B. Shutdown-Button hinzufügen (einfach „Eintrag“ in den Pannel ziehen und verschieben) oder bei Anwendungsstarter Menüs zu „Bildung“, „Spiele“, „Büro“, „Grafik“ etc ...*
- *Desktop-Verknüpfungen erstellen zu Epiphany und OpenOffice*
- *Mülleimer-Shurtcut und altes Gnome-Menü aus Panel entfernen*

Zusätzlich sollte man folgende Starteinträge (*System → Einstellungen → Startprogramme*) deaktivieren:

Aktualisierungsbenachrichtigung, Auf neue Hardware-Treiber überprüfen,
Bluetooth-Verwaltung, Energieverwaltung, Evolution Alarm Notifer,
GNOME-Begrüßungsbildschirm, GNOME Login Sound,
Netzwerkmanager, Tracker, Tracker Applet, Visuelle Rückmeldung

⁵⁵ http://wiki.ubuntuusers.de/Benutzer_und_Groupen

3.2.2 - Zeitplaner Cron konfigurieren

Die Client-PCs sollen nur an bestimmten Tagen/Wochen zu bestimmten Zeiten verfügbar sein – d.h. man richtet mit sogenannten „Cron-Jobs“ die Zeiten ein, in denen der Rechner hoch- bzw. herunterfährt.

Mit dem Batchdaemon *cron* lassen sich Befehle zu bestimmten Zeiten ausführen. Um z.B. die Client-PCs um 18:10 Uhr automatisch zu beenden, wird folgende Zeile in die Datei */etc/crontab* hinzugefügt:

```
#m h dom mon dow user command
10 18 * * * root shutdown -h now
```

Einstellungen werden übernommen mit */etc/init.d/cron restart*

3.2.3 – Iptables-Firewall

Client-Side-Protection :) Temporärer Full-Axx für Admins

◦ **3.3 System absichern**

Die Client-Rechner sollten auch ohne Aufsicht in ihrer Konfiguration unangetastet bleiben, deshalb dürfen Benutzer nur im sogenannten „Kiosk-Modus“ arbeiten. Dies bedeutet, dass Änderungen am System nicht möglich sind und wenn doch, nur temporär, bis zum Neustart erhalten bleiben.

3.3.1 - Bootloader Grub

Sehr Wichtig ist es den **Grub-Bootloader**⁵⁶ mit einem Passwort zu sichern, da ansonsten ein Benutzer beim Bootvorgang die Möglichkeit hat direkt in eine *root*-Konsole zu wechseln und somit Vollzugriff auf das komplette System hätte.

Hierzu wird am besten als erstes ein Backup der Datei */boot/grub/menu.lst* erstellt und anschließend im Terminal der Befehl *grub-md5-crypt* ausgeführt. Der Einfachheit wegen, # kann hier das schon gesetzte *root*-Passwort erneut benutzt werden. Man erhält darauf einen MD5-hashes der in die Datei */boot/grub/menu.lst* mit folgender Zeile eingetragen wird: *password -md5 <MD5-Hash>* ohne *<>* eingetragen.

3.3.2 - Epiphany Webbrowser

Im Client-Konto bei **Epiphany** unter „Bearbeiten → Einstellungen“ die Startseite auf die IP des Servers stellen, z.B. 192.168.179.1 und den Dateidownload-Ordner auf Dokumente setzen (Achtung! Menüleiste hierfür in Pessulus wieder temporär entsperren!)

3.3.3 - Gnome-Desktop

In der Datei */usr/share/nautilus/ui/nautilus-desktop-icon-view-ui.xml* werden nun die beiden Einträge:

```
<menuitem name="New Launcher" action="New Launcher Desktop"/> und
<menuitem name="Change Background" action="Change Background"/>
```

entfernt, damit der Benutzer den Desktophintergrund nicht mehr ändert und keine

⁵⁶ <http://wiki.ubuntuusers.de/GRUB>

Verknüpfungen mehr erstellen kann.


Um bei laufendem Betrieb ins „*root-Konto*“ zu wechseln, kann man sich jederzeit mit ALT + DRUCK + K „abmelden“.

Anschließend kann im *root* Konto mit dem Terminal-Befehl *pessulus* das Panel gesperrt werden und viele andere Einstellungen vorgenommen werden:

- Unter Allgemein „Kommandozeile, Drucken und Druckereinrichtung“ deaktivieren.
- Unter Panel „Panel, Abmelden und Bildschirmsperren“ deaktivieren.
- Unter Epiphany Web-Browser „Bearbeiten der Lesezeichen/Menüleiste, Chronik und unsichere Protokolle deaktivieren“. Die Protokolle sind: about:config, about:settings, [file://](#) und ftp://.

Wichtig ist, dass die Option '*Panel sperren*' als obligatorisch markiert wird (das kleine orangene Symbol). Außerdem kann unter *System->Systemverwaltung->Zugriffsberechtigungen* für verschiedenste Aktionen eine Admin-Authentifizierung als erforderlich gesetzt werden.

Um Änderungen an Menüeinträgen zu unterbinden, müssen mit folgenden Befehlen einige Konfigurationsdateien und Programme gesperrt werden:



```
$ sudo chmod o-x-w /usr/bin/alacarte /usr/bin/gmenu-simple-editor
/usr/bin/pessulus /usr/bin/gconf-editor /usr/bin/gconftool
/usr/bin/gconftool-2sudo chmod o-x-w /usr/bin/alacarte /usr/bin/gmenu-
simple-editor /usr/bin/pessulus /usr/bin/gconf-editor /usr/bin/gconftool
/usr/bin/gconftool-2
$ sudo chown root:root
/home/clientXX/.config/menus/applications.menu
/home/clientXX/.config/menus/settings.menu
```

3.3.4 - Zugriffsrechte

Zum Schluss werden dem Clientbenutzerkonto noch die Schreibrechte in seinem eigenen /home Verzeichnis entzogen, somit können keine zusätzlichen Dateien auf dem Desktop angelegt werden. Zusätzlich wird der Zugriff auf fast alle Programme verboten, nur eine ausgewählte Liste wird zugelassen:



```
$ sudo chmod o-w /home/clientXX/Desktop
$ sudo chmod -R o-x-w /usr/bin
$ sudo chmod o+x /usr/bin/{javaws,gobby,vinagre,mono,dia,f-
spot,ghostscript,scribus,tomboy,dbus-
launch,basename,dirname,env,ooffice,soffice,oowriter,xsane,step,parley,x-
session-manager,x-window-manager,epiphany*,expr,gk*,gnome-settings-
daemon,gnome-panel,nautilus,X11,kturtle,kwortquiz,gnome-wm,gnome-
window-properties,ktouch,knplot,kig,kbruch,kalzium,kalgebra,gnome-
dictionary,xaos,inkscape,guchamap,gedit,gcalctool,gnome-
screenshot,liferea} /usr/bin/{sudo,xterm}
```

3.3.5 - X-Server absichern

Der Wechsel in den Terminal mit Strg+Alt+F(1-10) kann verhindert werden durch einen zusätzlichen Eintrag in `/etc/X11/xorg.conf`⁵⁷

```
Section "ServerFlags"
    Option "DontVTSwitch" "yes"
    #Option "DontZap" "false"
Endsection
```

3.3 Grafisches

3.3.1 - Wallpaper (1. Konzept)



Um beim Client das Wallpaper zu ersetzen, wird einfach das `DEFAULT-WALLPAPER` (png!) überschrieben -> `usr/share/backgrounds/edubuntu_default.png` mit einem von unseren Wallpapers: `Wallpaper1.png`, `Wallpaper2.png` und `Wallpaper3.png` (zu finden im Dokumentationsordner). Alle anderen Wallpaper in `usr/share/backgrounds` können gelöscht werden!

3.3.2 - Splash-Screen (Bootscreen)

Um den Bootscreen zu ändern wird einfachheitshalber **GSPLASHY**⁵⁸ benutzt, eine graphische Oberfläche von **SLPASHY**⁵⁹. Zum Installieren benutzen wir dieses Installationscript (befindet sich auch im Dokumentationsordner):

```
#!/bin/bash
sudo apt-get autoremove usplash
sudo apt-get install libglade2-dev libsplashy1-dev build-essential libsplashy1
libdirectfb-dev libdirectfb-extra libsysfs-dev
wget
http://ppa.launchpad.net/tohms/bugfixes/ubuntu/pool/main/s/splashy/splashy_0.3.
13-3ubuntu2~ppa4_i386.deb
sudo dpkg -i splashy_0.3.13-3ubuntu2~ppa4_i386.deb
rm splashy_0.3.13-3ubuntu2~ppa4_i386.deb
echo "splashy wurde erfolgreich installiert"
mkdir ~/bejonet
cd ~/bejonet
wget -c http://alioth.debian.org/frs/download.php/2243/gsplashy-0.1.tar.gz
tar -xzf gsplashy-0.1.tar.gz
rm gsplashy-0.1.tar.gz
cd ~/bejonet/gsplashy-0.1 ##kompilieren
./configure
make
sudo make install
echo "gsplashy wurde erfolgreich installiert"
```

⁵⁷ <http://wiki.ubuntuusers.de/XServer>

⁵⁸ <http://splashy.alioth.debian.org/wiki/gsplashy>

⁵⁹ <http://splashy.alioth.debian.org/wiki/>

Oder **sudo bash installer.sh** ausführen in dem Downloadordner (installer.sh)!

Danach **sudo gsplashy** ausführen und bei "Installieren" das Theme fwsv_splashy.tar.gz auswählen. Danach mit einem Klick auf den neuen Eintrag in der Übersicht das Theme bestätigen. Damit der Bootscreen in der richtigen Auflösung angezeigt wird, muss folgender Eintrag in **/boot/grub/menu.lst** angepasst werden – also **sudo gedit /boot/grub/menu.lst** ausführen.

Der Standard-Booteintrag für Ubuntu 9.04 wird von

```
title          Ubuntu 9.04, kernel 2.6.28-11-generic
uuid           d8ea6f1d-ba70-[...]
kernel         /boot/vmlinuz-[...]-generic root=UUID=[...] ro quiet
splash
initrd         /boot/initrd.img-2.6.28-11-generic
quiet
```

nach

```
title          Ubuntu 9.04, kernel 2.6.28-11-generic
uuid           d8ea6f1d-ba70-44f7-aa1b-b0fc56604d34
kernel         /boot/vmlinuz-2.6.28-11-generic
root=UUID=d8ea6f1d-ba70-44f7-aa1b-b0fc56604d34    ro    quiet
vga=791 splash
initrd         /boot/initrd.img-2.6.28-11-generic
quiet
```

geändert. Je nach System ist die Kernel-Version und UUID unterschiedlich, deswegen werden nur die Parameter von "Kernel" angepasst. Zum Schluß muss noch dieser Befehl ausgeführt werden:



```
$ sudo update-initramfs -u
```

Achtung: /etc/LSB-BASE-LOGGING.SH nur mit USPLASH nutzen! Beim Einsatz von SPLASHY sollte diese Datei nicht existieren!

3.3.3 - GDM-Theme (Login)

Das Login-Theme, welches dem Bootscreen ähnelt und auch sehr schick aussieht, ist sehr schnell über Ubuntu Administration-Tools eingestellt. System -> Systemverwaltung -> Anmeldefenster. Im Tab "Lokal" dann auf "Hinzufügen" klicken, die fws-edubuntu-black.tar.gz auswählen und Theme als einzigstes auswählen. Alle anderen Themes können dann gelöscht und die Hintergrundfarbe auf „Schwarz“ gesetzt werden. Die Konfigurationen werden spätestens nach Logout/Neustart übernommen! Theme-Einstellungen sind: Sprache- und Sessionmenüs deaktiviert - Restart und Shutdown aktiviert!

3.3.4 - Bildschirmschoner

Um den speziellen Bildschirmschoner zu installieren, extrahiert man das Paket screensaver.tar.gz im Dokumentationsordner. Danach kann man über die Bildschirmschoner-

Einstellungen (System / Einstellungen / Bildschirmschoner) den Screensaver auf den Vorher extrahierten *Fws-Screensaver* ändern.

◦ **3.4 Priviligierter Client (ITG-Raum)**

Den Clients im ITG-Raum soll freier Internetzugriff gewährt werden. Hier werden alle Clients in ein VPN-Netz eingebunden, welche dann auf dem Server nicht gefiltert wird. Wichtig ist hierbei zum einen, dass die Clients sich in dem anderen VPN anmelden, sofern sich der Administrator einloggt. In diesem VPN ist dann auch SMB-Zugriff auf die Netzwerkplatten möglich (evtl. für Lehrer).

3.4.1 - Konfiguration für Windows-Systeme

Nach dem auch hier OpenVPN (**Achtung: Bei Versionen unter 2.1 rc33 könnte es zu Verbindungsproblemen kommen unter Windows XP SP3!**) heruntergeladen⁶⁰ und installiert wurde, werden die für den Client vorgesehenen Zertifikate und eine entsprechende Konfigurationsdatei im Ordner `C:\Programme\OpnevVPN\config\` abgelegt. In der `services.msc`⁶¹ (Eingegeben in Start->Ausführen) wird nun der OpenVPN Service als Automatisch ausgewählt. Somit werden jede `.ovpn` im config Ordner beim Start ausgeführt. In einem Unterordner „vpn1“ werden die VPN-Zertifikate für den Client im anderen VPN-Netz abgelegt.

Außerdem wird die Datei „vpn1.bat“ mit folgendem Inhalt angelegt.

```
@echo off
net stop „OpenVPN Service“
start ..\..\bin\openvpn-gui.exe --connect clientXX.ovpn
```

Von dieser Datei kann eine Verknüpfung auf dem Desktop des Admins angelegt werden um das Standard VPN zu stoppen und das privilegiere VPN zu starten. Die OpenVPN Gui kann unter heruntergeladen werden.

<http://openvpn.se/files/binary/openvpn-gui-1.0.3.exe>

Die Datei wird einfach nach `C:\Programme\OpenVPN\bin\` kopiert.

Als letztes wird in dem Eigenschaftsfenster des Config Ordners unter Sicherheitsrichtlinie dem Administrator Vollzugriff gewährt und allen anderen Benutzern (JEDER) der Zugriff komplett verweigert.

⁶⁰ <http://openvpn.net/index.php/open-source/downloads.html>

⁶¹ http://www.theeldergeek.com/services_guide.htm