Installation: *FilterSurf*

Hier werden nun die Schritte erläutert, die nacheinander zu durchlaufen sind, um den zentralen *FilterSurf*-Server verwenden zu können. Die Installationsschritte werden beispielhaft an einem Debian-Linux vorgeführt. Die Installation bei anderen Linux-Distributionen läuft völlig analog.

Es folgt nun eine Liste, die die notwendigen Voraussetzungen für die Verwendung von *FilterSurf*, näher beschreibt. Wenn bei Ihnen nicht alle Voraussetzungen vorliegen, wird in Abschnitt 1 beschrieben, wie diese Voraussetzungen geschaffen werden können.

1. Linux-Internet-Gateway
2. Squid-Proxy-Server (mit Redirect support)
3. Perl
4. *FilterSurf* Perl Redirector-Skript

1 Voraussetzungen schaffen

Die **Voraussetzung 1** ist wohl die entscheidenste. Sollten Sie ein anderes Betriebssystem oder eine Hardware-Routing-Lösung verwenden, so ist es nicht direkt möglich, *FilterSurf* zu verwenden. Sie können sich aber einen kleinen kostengünstigen Mini-PC besorgen, auf dem Linux installiert ist, und diesen als Network-Bridge und transparenten Proxy konfigurieren und zwischen ihrem Router und die Clients hängen. Informationen und Angebote für einen solchen Mini-PC sind bei uns erhältlich.

Wenn die Voraussetzung 1 erfüllt ist, dann wird mit an Sicherheit grenzender Wahrscheinlichkeit ein Squid-Proxy-Server (**Voraussetzung 2**) auf Ihrem Internet-Gateway installiert sein. Falls nicht, so können Sie mit

```
apt-get install squid
```

das entsprechende Package installieren. Wenn Sie nicht eine absolut uralte Version haben, dann unterstützt Ihr Squid auch ein sog. Redirect Program.

Die **Voraussetzung 3** ist in den aller meisten Fällen sowieso erfüllt.

Als **Voraussetzung 4** benötigen Sie das Redirector-Skript. Es kann unter

```
http://www.filtersurf.de/downloads/redirector.pl
```

heruntergeladen werden, z. B. wird durch

```
cd /usr/local/bin
wget http://www.filtersurf.de/downloads/redirector.pl
```

das Skript gleich ins `/usr/local/bin`-Verzeichnis geladen.

2 Perl Redirector-Skript vorbereiten

Es wird jetzt davon ausgegangen, dass sich das Perl Redirector-Skript im Verzeichnis `/usr/local/bin` befindet (vgl. Abschnitt 1). Setzen Sie die Rechte so, dass auch der User, auf dem der Squid läuft, das Recht zum Lesen und zum Ausführen hat:

```
cd /usr/local/bin
chmod 0755 redirector.pl
```

Jetzt können Sie gleich mal den Test machen, ob Sie alle nötigen Packages installiert haben. Starten Sie dieses Skript. Im Idealfall kommt keine Meldung und das Skript wartet auf Eingaben. Sie können es mit `CTRL-D` beenden.

3 Squid-Konfiguration anpassen

Als nächstes öffnen Sie die Datei `squid.conf` meist zu finden im Verzeichnis `/etc` oder `/etc/squid`. Hier müssen Sie zwei Parameter ändern. Suchen Sie den Abschnitt über `redirect_programm` und geben Sie das Perl Redirector-Skript an:

```
redirect_program /usr/local/bin/redirector.pl
```

Als nächstes suchen Sie den Abschnitt über `redirect_children`. Geben Sie hier einen hinreichend großen Wert an. Als Faustformel: Anzahl der Rechner die gleichzeitig auf den Squid zugreifen werden:

```
redirect_children 20
```

4 Squid neu starten

Starten Sie als letzten Schritt den Squid neu:

```
/etc/init.d/squid restart
```

5 (Optionaler) Test des Redirectors

Um manuell am Linux-Router zu überprüfen, ob das Perl Redirector-Skript einwandfrei arbeitet, geht man so vor: Man startet das Skript

```
/usr/local/bin/redirector.pl
```

Es darf keine einzige Meldung kommen und das Skript muss auf eine Eingabe warten. Geben Sie hier nun mal eine URL manuell ein, also z. B.

```
http://www.google.de
```

und drücken Sie return. Es sollte genau die eingegebene Adresse nochmals erscheinen. Geben Sie jetzt mal eine Adresse an, die gefiltert wird, z. B.

```
http://www.playboy.de
```

Hier sollte als Antwort dann eine andere Adresse erscheinen, nämlich etwas in der Art:

```
http://fsredir.dyndns.org/accessdenied.php?req=[...]
```

Daran können Sie erkennen, dass diese Seite nicht angzeigt, sondern umgeleitet wird. Sie können beliebige weitere Adressen testen. Beenden können Sie das ganze mit EOF, also mit `CTRL-D`.

6 Zwangsproxy und andere Optionen

Falls eine oder mehrere der Bedingungen

- Der Rechner, an dem der squid läuft, muss einen Proxy-Server verwenden
- Sie wollen zusätzliche Seiten sperren

erfüllt sind, so ist eine zusätzliche Konfiguration des Perl Redirector-Skripts notwendig.

Gehen Sie hierzu ins Verzeichnis des Skripts und laden sie die Datei `redirector.conf` herunter:

```
cd /usr/local/bin
wget http://www.filtersurf.de/downloads/redirector.conf
```

Editieren Sie nun diese Datei und tragen Sie an der entsprechenden Stelle den Proxy-Server ein, den das Redirector-Skript verwenden soll. In der gleichen Datei haben Sie auch die Möglichkeit zusätzlich Seiten zu sperren.

Eine `redirector.conf` Datei könnte also so aussehen:

```
#!/usr/bin/perl

# FilterSurf-Redirector Konfigurationsdatei
# Version V2.11 (05.03.2005)
# C. Ludwig & D. Herrmann
# www.filtersurf.de

package conf;

### 1. PROXY-SERVER ######################################################

# Wenn ein HTTP-Proxy-Server ("Zwangs-Proxy") für die Kommunikation
# mit dem Internet benötigt wird und dieser für die Verbindung zum
# FilterSurf-Server verwendet werden soll, dann kann der Proxy-Server
# hier eingetragen werden.
# Dies ist nur dann nötig, falls eine direkte Verbindung ohne
# Verwendung des Proxy-Servers unmöglich ist.
# Ist kein Proxy-Server vorhanden oder wird er nicht benötigt, dann
```

```
# müssen die Zeilen proxy_server_ip und proxy_server_port mit "#"
# auskommentiert werden.

# Hinweis:
# Hier KEINEN Hostnamen (z.B. proxy.local.net) eintragen, sondern
# die IP-Adresse (z.B. 10.0.0.1)!

$proxy_server_ip = "10.10.0.250";
$proxy_server_port = "8080";


### 2. KATEGORIEN ########################################################

# Das hier ist eine Bitmaske, die angibt, welche Kategorien gefiltert
# werden sollen:
#  Kategorie 1: porno/adult                (1)
#  Kategorie 2: aggressive/violence        (2)
#  Kategorie 3: search engines             (4)
#  Kategorie 4: hacking/warez              (8)
#  Kategorie 5: forums/chats              (16)
#  Kategorie 6: onlineauctions            (32)
#  Kategorie 7: (web)mail                 (64)
#  Kategorie 8: gambling/time waste      (128)
#  Beispiel: Um porno/adult & search engines & forums/chats zu sperren:
#   $categories = 1 + 4 + 16;

$categories = 3;


### 3. LOKALE BLACKLISTE #################################################

# Liste mit zusätzlich zu filternden Adressen (zusätzlich zu den oben
# ausgewählten Kategorien)
#
# Verwendungshinweise:
#   Ein Eintrag der Form a.b.c sperrt alle Adressen, die
#   so ENDEN, also auch x.y.a.b.c.
#
# Beispiel: Eintrag 'filtersurf.de' sperrt z.B.:
#   www.filtersurf.de, www.test.filtersruf.de
#
# Beispiel: Eintrag 'www.filtersurf.de' sperrt:
#   www.filtersurf.de, ABER NICHT: filtersurf.de
#
```

```
# Die Markierungen (begin/end locallist) werden bei der FilterSurf-
# Box für die korrekte Funktion benötigt. Unbedingt unangetastet lassen!
#
# Beispiel-Eintrag:
# @locallist = (
#  'url1.de', 'url2.de', 'url3.de'
# );

# begin locallist
@locallist = (

);
# end locallist


### 4. LOKALE WHITELISTE ################################################

# Liste mit Seiten, die auf keinen Fall gefiltert werden sollen:
# Konvention wie oben
# Hier sollten auch Webserver eingetragen werden, die im lokalen
# Netzwerk (LAN) vorhanden sind. Diese sollen ja i.d.R. ohne Filterung
# erreichbar sein.
# Die Markierungen (begin/end localwhitelist) werden bei der FilterSurf-
# Box für die korrekte Funktion benötigt. Unbedingt unangetastet lassen!
#
# www.filtersurf.de verwendet das BPjM-Modul. Dieses Modul bewirkt die
# Filterung der von der Bundesprüfstelle für jugendgefährdende Medien
# (BPjM) indizierten Telemedien (Online-Angebote). Die Einträge umfassen
# sowohl jugendgefährdende Angebote als auch solche, deren Verbreitung
# nach Jugendmedienschutz-Staatsvertrag der Länder (JMStV) unzulässig
# ist. Informationen zur BPjM und zum Indizierungsverfahren finden Sie
# unter www.bundespruefstelle.de.
#
# Wichtiger Benutzungshinweis:
# www.filtersurf.de erlaubt die Verwendung einer sog. "lokalen"
# Whiteliste, die von jedem FilterSurf-Benutzer (Systemadministrator)
# selbst gepflegt werden kann. In diese Whiteliste können Internet-
# Domains aufgenommen werden, die nicht gesperrt werden sollen. Mit
# solchen Einträgen ist es daher möglich, einzelne Einträge den
# Endbenutzern zugänglich zu machen, auch wenn Sie ohne Whitelist
# gesperrt worden wären. Wenn sich ein Systemadministrator Sicherheit
# verschaffen möchte, ob ein beabsichtigter Whitelist-Eintrag mit dem
# BPjM-Modul kollidiert, kann er die fragliche URL in einer
# Einzelabfrage an die BPjM schicken: liste@bundespruefstelle.de.
```

```
#
# Beispiel-Eintrag:
# @localwhitelist = (
#  'url1.de', 'url2.de', 'url3.de'
# );


# begin localwhitelist
@localwhitelist = (

);
# end localwhitelist

### 5. STATUS und ACCESS LOG ##############################################

# Wenn der Redirector alle Web-Zugriffe protokollieren soll, dann muss
# der Parameter $access_log auf '1' gesetzt werden.
# Der Dateiname des Logfiles sollte dann noch in /etc/syslog.conf
# gesetzt werden. Der Redirector verwendet standardmäßig die syslog
# Facility local0. Falls eine andere Facility gewünscht ist, kann dies
# mit dem Parameter $syslog_facility eingestellt werden.
# Ist keine Protokollierung erwünscht (minimal bessere Performance), so
# muss der folgende Parameter mit # auskommentiert werden
# In jedem Fall geloggt werden Status-Meldungen, Warnungen und Fehler,
# die der Redirector im Betrieb erkennt.
# Falls ein reines access_log gewünscht wird, kann man das generierte
# Logfile mit grep ganz leicht von den Status-Meldungen säubern.

$access_log = "1";

$syslog_facility = "local0";


### 6. SERVER-NAME UND USER-ID ############################################

# Welcher FilterSurf-Server soll verwendet werden? Derzeit existiert nur
# ein FilterSurf-Server mit dem Namen fsredir.dyndns.org. Dies ist auch
# der Default-Wert.

$filtersurfserver = "fsredir.dyndns.org";


# Wenn der verwendete FilterSurf-Server eine Authentifizierung verlangt,
# dann muss der Nachfolgende Parameter mit einer gültigen User-ID
```

```
# besetzt werden. Andernfalls sollte er mit # auskommentiert werden.
# Die User-ID wird bei jeder HTTP-Anfrage an den FilterSurf-Server
# geschickt

$userid = "100000000B";
```

In den nächsten Unterabschnitten wird die Einrichtung eines weiteren optionalen Features, der sog. shared Cache, erklärt.

6.1 Shared Cache; Zweck des Perl-Moduls `Cache::FastMmap`

Ab Version 2.11 unterstützt der FilterSurf Redirector einen sogenannten „shared“ (gemeinsam genutzten) Cache, um die Effizienz des lokalen URL-Zwischenspeichers zu steigern. Die zugrunde liegende Idee soll hier nur kurz erklärt werden.

Squid startet bekanntlich immer mehrere Redirector-Prozesse auf einmal (z. B. 15 Stück), um bei eingehenden Anfragen nicht unnötig auf einen Redirector-Prozess, der gerade noch mit einer vorherigen Anfrage beschäftigt ist, warten zu müssen. Jeder der FilterSurf-Redirector-Prozesse hatte bis Version 2.10 einen separaten URL-Cache, in dem die letzten paar URLs, die der FilterSurf-Server **erlaubt** hatte, zwischengespeichert wurden. Dummerweise wussten die einzelnen Caches nichts von einander (jeder Cache war einem Redirector zugeordnet) und so befanden sich viele URLs mehrfach in allen Caches. Dieses Verhalten führte dazu, dass eine URL von mehreren Redirector-Prozessen beim FilterSurf-Server nachgefragt werden musste, obwohl andere Redirector-Prozesse sie schon längst in ihren Cache aufgenommen hatten.

Ab Version 2.11 gibt es nun einen gemeinsamen Cache, auf den alle Redirector-Prozesse zugreifen können. Dies steigert die subjektive Geschwindigkeit beim Browsen deutlich. Hierfür kommt das `Cache::FastMmap` Perl-Modul von Rob Mueller zum Einsatz. Wenn beim Start von Version 2.11 festgestellt wird, dass das `Cache::FastMmap` Modul nicht vorhanden ist, verwenden die Redirector-Prozesse wie gewohnt den (langsameren) dedizierten Cache.

6.2 Installation von `Cache::FastMmap`

Herunterladen des Quellcodes (derzeit aktuell ist Version 1.09, falls Sie eine aktuellere Version testen möchten, können Sie zu

`http://search.cpan.org/~robm/Cache-FastMmap/`

gehen und dort „Download“ anwählen).

```
wget http://search.cpan.org/CPAN/authors/id/R/RO/ROBM/Cache-FastMmap-1.09.tar.gz
```

Jetzt packen Sie ddie `tar.gz`-Datei aus und wechseln in das Quellcode-Verzeichnis:

```
tar zxf Cache-FastMmap-1.09.tar.gz
cd Cache-FastMmap-1.09
```

Danach erstellen Sie das Makefile für Kompilierung und übersetzen den Quellcode

```
perl Makefile.PL
make
```

`Cache::FastMmap` wird nun mit

```
make install
```

installiert.

Nun den Redirector (ab V2.11) von der Console starten

```
cd <Verzeichnis des Redirectors>
./redirector.pl
```

und mit `CTRL-D` den Redirector gleich wieder zu beenden.

Mittels

```
tail /var/log/messages | grep "shared"
```

werfen Sie nun einen Blick in die Log-Datei. Wenn Sie nun ein paar Zeilen mit Statusmeldungen sehen, die darauf hindeuten, dass der Redirector mit dem `Cache::FastMmap` Modul gestartet wurde, dann war die Installation erfolgreich.

Ist Ihnen diese Installationsprozedur zu kompliziert? Vielleicht kommt für Sie dann die FilterSurf-Box in Frage. Diese enthält immer den aktuellsten Redirector inkl. des oben beschriebenen `Cache::FastMmap-Modul` fertig vorinstalliert.