

13 Web-Seiten im Proxy-Cache zwischenspeichern und filtern

Das World Wide Web wird oft lästerhaft World Wide Wait genannt, weil immer mehr Anwender immer mehr Seiten anfordern, als Netz-Anbieter Bandbreite für nicht bevorzugte Anwender schaffen.

Anwender können Web-Seiten schneller abrufen, wenn sie

- Verträge für schnellere Zugänge, Zugänge mit garantierter Bandbreite oder für Zusatzbandbreite über Satellit abschließen oder
- Seiten, die sie selbst oder andere Anwender der gleichen Gruppe wiederholt anfordern, nicht jedes mal neu laden, sondern aus einem Zwischenspeicher abrufen.

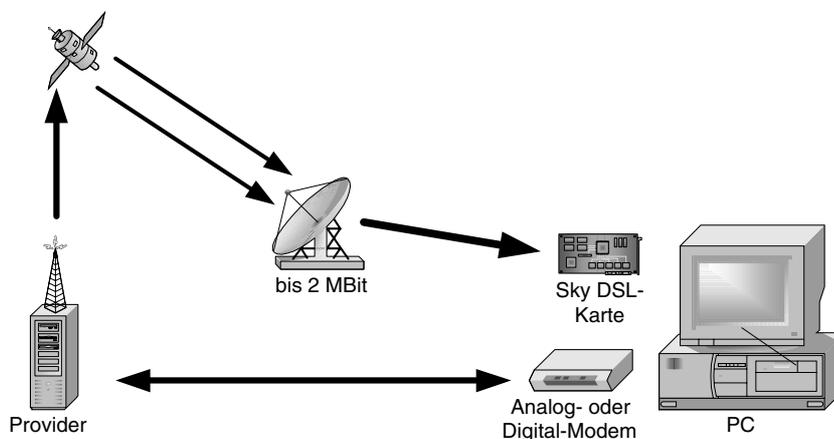


Abbildung 13.1: Mehr Bandbreite, z.B. durch satellitenbasiertes DSL

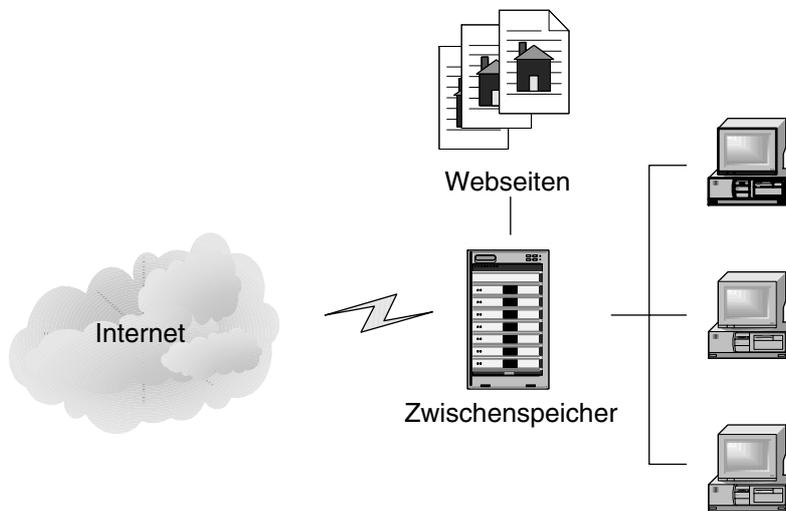


Abbildung 13.2: Web-Seiten im Proxy-Cache

Werden Internet-Seiten in geschützten Räumen wie Schulen, Betrieben, Internet-Cafes mit minderjährigen Besuchern oder Familien abgerufen, sind System-Verantwortliche gefordert, neben schnellem Seitenabruf auch Filter und Inhalts-Kontrolle einzurichten.

Web-Zugriffe lassen sich durch verschiedene Zwischenspeicher beschleunigen und filtern:

- Durch lokale Speicher und Filter beim Anwender oder
- zentrale Speicher und Filter zwischen Internet und Clients im Intranet.

Lokale Zwischenspeicher für Internetseiten, Cache genannt, benutzen fast alle Anwender, Anfänger sogar ohne es zu wissen, weil Web-Browser diese Funktion schon in der Grundausstattung bieten. Um im Interesse des Jugendschutzes Seiten zu filtern, benötigt man Zusatzprogramme, die Seiten mit unerwünschten Inhalten wie bestimmten Text- oder Grafikobjekten und von einschlägigen Web-Sites ignorieren. Löschen Anwender den Verlauf ihrer Web-Sitzungen nicht beim Verlassen des Surfplatzes, können Dritte ausspionieren, welche Web-Sites sie besucht haben.

Diese lokalen Zwischenspeicher legen bereits einmal geladene Internetseiten im Hauptspeicher oder auf der Festplatte ab, so dass bei einem erneuten Zugriff auf die Seite keine weiteres Laden aus dem Internet erforderlich ist, es sei denn, die Seite hätte sich geändert.

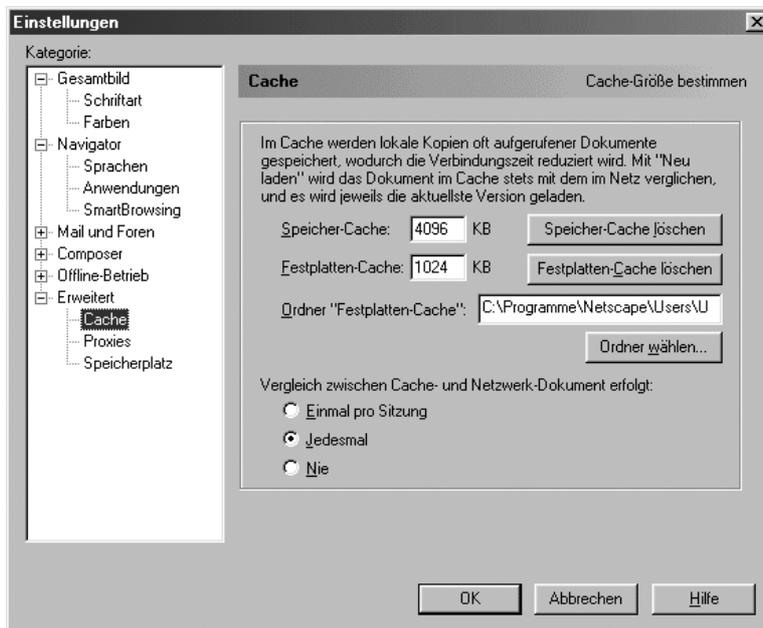


Abbildung 13.3: Cache-Einstellungen im Netscape Communicator



Abbildung 13.4: Cache-Einstellungen im Internet Explorer

Neben diesen lokalen Speichern kann man in lokalen Netzen die von den Anwendern besuchten Seiten zentral speichern. So kommt der Geschwindigkeitsvorteil für das erneute Laden allen zugute, da die Ladezeiten im lokalen Netz

vergleichsweise kurz sind und Systemverwalter können das Surfverhalten der Anwender auch überwachen, wenn diese die lokalen Speicher nach Arbeitsende löschen.

Für diesen Zweck setzt man auf Kommunikations-Servern einen Proxy-Server, bei Linux meist *Squid*, ein.

Zusätzlich zu der Cache-Funktion verfügt Squid über eine Stellvertreter- (Proxy) Funktion. Bei der Einwahl ins Internet stellt der Provider nur eine einzige offizielle IP-Adresse zur Verfügung, die der Linux-Server bekommt. Die anderen Rechner im Netz verfügen nur über lokale IP-Adressen, an die Web-Server keine Antworten schicken können. Diese lokalen Rechner fordern WWW-Seiten indirekt vom Squid an, welcher sie mit der IP-Adresse des Linux-Servers aus dem Internet abrufen, sofern er sie nicht schon lokal gespeichert hat.

13.1 Wann lohnt sich ein Proxy-Cache?

Ein Proxy-Cache hat mehrere Vorteile bzw. Aufgaben:

- Er beschleunigt den Internet-Zugriff,
- hat eine Stellvertreterfunktion für die Rechner im Netz;
- er kann kontrollieren, welche Inhalte Benutzer im lokalen Netz anfordern dürfen und
- er dokumentiert, wer welche Web-Seiten tatsächlich geladen hat.

Wie sehr der Proxy-Server das Laden von Web-Seiten beschleunigt, indem er mehrfach angeforderte Seiten aus dem lokalen Netz statt aus dem Internet bereitstellt, hängt in der Praxis davon ab, wie viele Nutzer die gleichen Seiten anfordern und wie viele Nutzer sich eine vielleicht nur schmalbandige Internet-Anbindung teilen müssen.

Die Proxy- (Stellvertreter-) Funktion ist die einfachste Möglichkeit, beliebig vielen Rechnern im Intranet den Zugriff auf WWW-Seiten zu ermöglichen. Da dabei nur der Proxy Anfragen ins Internet stellt, kommt man mit einer einzigen offiziellen IP-Adresse aus.

Will man den lokalen Rechnern erlauben, selbst direkt unter Umgehung des Proxy auf Web-Server zuzugreifen, muss der Server die lokalen IP-Adressen jeweils durch seine eigene ersetzen (IP-Masquerading). Um auch dann noch Sperr- und Kontrollmöglichkeiten zu garantieren, muss man jedoch einen Firewall einrichten und betreiben (s. Kapitel 14).

Proxies können gezielt einzelne Seiten oder ganze Internet-Domains sperren, damit kein darauf zugreifender Browser diese überhaupt sehen oder laden kann.

Da ein Proxy alle Zugriffe protokollieren kann, lässt sich überwachen, wer welche Seiten aufgerufen hat.

13.2 So funktioniert ein Proxy-Cache

Anfragen von Client-Browsern gehen nicht mehr direkt ins Internet, sondern zum Proxy-Server. Dieser prüft, ob er eine aktuelle Version der angeforderten Seite gespeichert hat. Wenn die Seite vorliegt und noch aktuell ist, liefert er sie direkt aus dem lokalen System heraus an den Browser.

Hat er die Seite nicht im Speicher oder ist sie nicht mehr aktuell, so lädt der Proxy sie aus dem Internet, speichert sie bei sich und stellt sie dann den Browsern der Clients zur Verfügung.

13.3 Squid installieren und konfigurieren

Da alle Linux-Distributionen Squid enthalten, lässt er sich einfach durch Auswahl des zugehörigen Pakets einrichten. Bei SuSE befindet sich die bewährte Version des Squid in der Serie *n* im Paket `squid2` bzw. in der Datei `squid2.rpm` im Verzeichnis *n1*.

Datei	Bedeutung
<code>/usr/sbin/squid</code>	Binärdatei des Squid-Servers
<code>/etc/init.d/squid</code>	Start/Stop-Script für Squid.
<code>/etc/squid.conf</code>	Squid-Konfigurationsdatei.

Tabelle 13.1: Die Dateien zu Squid

Nach der Installation muss man dafür sorgen, dass Squid automatisch startet. Dazu ruft man YaST auf und geht in *Administration des Systems* in das Menü *Konfigurationsdatei verändern*. Dort sucht man aus der Liste die Variable `START_SQUID` und setzt ihren Wert auf *yes*. Anschließend kann man YaST beenden.

Diese Änderung wird erst beim nächsten Neustart des Netzwerks wirksam. Von Hand starten Sie Squid mit

```
rcsquid start
```

Die für den laufenden Betrieb benötigten Ordner und Dateien legt Squid beim ersten Start selbstständig an.

Squid konfiguriert man über die 1.900 Zeilen große Datei `squid.conf`, deren größter Teil aus Kommentaren und Dokumentation besteht.

`/etc/squid.conf` (Auszug ab Zeile 513):

```
# TAG: emulate_httpd_log      on|off
# The Cache can emulate the log file format which many 'httpd'
# programs use. To disable/enable this emulation, set
# emulate_httpd_log to 'off' or 'on'. The default
# is to use the native log format since it includes useful
# information that Squid-specific log analysers use.
#
#emulate_httpd_log off
```

Die ersten sieben Zeilen sind Kommentartext, erkennbar an dem einleitenden `#` Zeichen. Der Kommentar erklärt die Schalter. Der Schalter selber ist hier durch ein `#` deaktiviert, wodurch die Vorgabe `emulate_httpd_log off` gilt. Will man die Vorgabe ändern, so muss man den Schalter durch Entfernen des Kommentarzeichens aktivieren und `off` durch `on` ersetzen.

Um die Vorgaben individuell einzustellen, sollte man die Konfigurationsdatei sorgfältig bearbeiten. Insbesondere sollte man

- die Größe des Cache im laufenden Betrieb beobachten (s. Logdateien des Squid) und
- den tatsächlichen Bedürfnissen anpassen (s.u.).

In der aktuellen Distribution hat SuSE den Squid so weit auf Sicherheit getrimmt, dass er auch Web-Zugriffe aus dem lokalen Netz ablehnt.

`/etc/squid.conf` (Auszug ab Zeile 1178):

```
#Defaults:
acl all src 0.0.0.0/0.0.0.0
acl manager proto cache_object
acl localhost src 127.0.0.1/255.255.255.255
acl SSL_ports port 443 563
acl Safe_ports port 80 21 443 563 70 210 1025-65535
acl Safe_ports port 280          # http-mgmt
acl Safe_ports port 488          # gss-http
acl Safe_ports port 591          # filemaker
acl Safe_ports port 777          # multiling http
acl CONNECT method CONNECT

# TAG: http_access
#   Allowing or Denying access based on defined access lists
#
```

```

#      Access to the HTTP port:
#      http_access allow|deny [!]aclname ...
#
#      Access to the ICP port:
#      icp_access  allow|deny [!]aclname ...
#
#      NOTE on default values:
#
#      If there are no "access" lines present, the default is
#      to allow the request.
#
#      If none of the "access" lines cause a match, the
#      default is the opposite of the last line in the list.
#      If the last line was deny, then the default is allow.
#      Conversely, if the last line is allow, the default
#      will be deny.  For these reasons, it is a good idea to
#      have an "deny all" or "allow all" entry at the end
#      of your access lists to avoid potential confusion.
#
#Default configuration:
http_access allow manager localhost
http_access deny manager
http_access deny !Safe_ports
http_access deny CONNECT !SSL_ports
#
# INSERT YOUR OWN RULE(S) HERE TO ALLOW ACCESS
# FROM YOUR CLIENTS
#
http_access allow localhost
http_access deny all

```

Die Regel in der letzten Zeile aus diesem Ausschnitt verbietet jeglichen Zugriff per HTTP, wenn ihn bis dahin nicht eine andere Regel erlaubt hat.

Ersetzen Sie diese Zeile durch

```
http_access allow all
```

und veranlassen Sie den Squid, seine Konfigurationsdatei neu einzulesen:

```
rcsquid reload
```

Nach dieser Änderung stellt der Squid seine Dienste im lokalen Netz zur Verfügung.

13.4 Zugriffskontrolle durch den Proxy-Cache

Squid kann jeglichen Zugriff auf Internetadressen ausschließen, die Systembetreiber als unerwünscht einstufen:

Um einzelne Server, hier die Server `www.mues.li` und `www.wapbu.ch` vollständig zu sperren, richtet man in `squid.conf` eine Zugriffsregel (Access List=`acl`) ein:

```
acl heutespernt dstdomain www.mues.li www.wapbu.ch
```

Hinter dem Schlüsselwort `acl` folgt erst ein frei definierbarer Name für diese Regel, dann deren Gültigkeitstyp und danach eine Aufzählung der zu sperrenden Adressen.

Den in `squid.conf` bereits voreingestellten `acl`-Zeilen, fügt man eigene einfach hinzu.

Die so definiert Regel muss man noch aktivieren:

```
http_access deny heuteverboten
```

Dadurch verweigert Squid Zugriff auf alle Seiten, auf die die Regel zutrifft. Diese Zeile muss vor der Zeile

```
http_access allow all
```

stehen.

Nach diesen Änderungen muss der Squid mit

```
rcsquid reload
```

seine Konfigurationsdatei neu einlesen.

Sobald die Sperren aktiv sind, zeigt der Browser des Clients beim Versuch, gesperrte Seiten aufzurufen, eine Fehlermeldung.

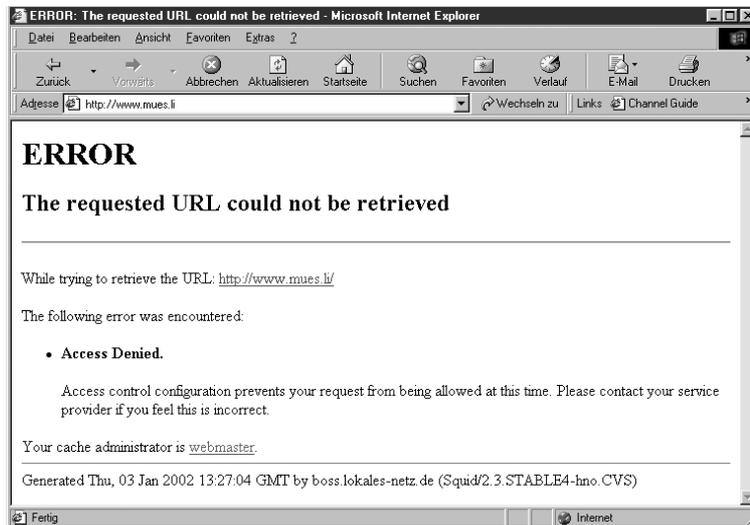


Abbildung 13.5: Zugriffsverweigerung bei gesperrten Seiten

Nach diesen Änderungen hat der besprochene Abschnitt der Konfigurationsdatei folgendes Aussehen:

/etc/squid.conf (Auszug ab Zeile 1178 nach Veränderungen):

```

acl all src 0.0.0.0/0.0.0.0
acl manager proto cache_object
acl localhost src 127.0.0.1/255.255.255.255
acl SSL_ports port 443 563
acl Safe_ports port 80 21 443 563 70 210 1025-65535
acl Safe_ports port 280          # http-mgmt
acl Safe_ports port 488          # gss-http
acl Safe_ports port 591          # filemaker
acl Safe_ports port 777          # multiling http
acl CONNECT method CONNECT

acl heuteverboten dstdomain www.mues.li www.wapbu.ch

# TAG: http_access
#   Allowing or Denying access based on defined access lists
#
#   Access to the HTTP port:
#   http_access allow|deny [!]aclname ...
#
#   Access to the ICP port:
#   icp_access  allow|deny [!]aclname ...

```

```

#
# NOTE on default values:
#
# If there are no "access" lines present, the default is
# to allow the request.
#
# If none of the "access" lines cause a match, the default
# is the opposite of the last line in the list. If the
# last line was deny, then the default is allow.
# Conversely, if the last line is allow, the default will
# be deny. For these reasons, it is a good idea to have
# an "deny all" or "allow all" entry at the end
# of your access lists to avoid potential confusion.
#
#Default configuration:
http_access allow manager localhost
http_access deny manager
http_access deny !Safe_ports
http_access deny CONNECT !SSL_ports
#
# INSERT YOUR OWN RULE(S) HERE TO ALLOW ACCESS FROM YOUR
# CLIENTS
#
http_access allow localhost
http_access deny heuteverboten
http_access allow all

```

Bei einer Festverbindung ins Internet sollten Sie aus Sicherheitsgründen die Default-Regel nicht wie hier beschrieben auf

```
http_access allow all
```

setzen, denn dann können alle Internetnutzer auf Ihren Squid zugreifen. In diesem Fall ist es sicherer, eine neue Zugriffsregel für Ihr lokales Netz zu erstellen und über diese Regel den Zugriff zu erlauben.

Wenn Ihr lokales Netz den Adressbereich 192.168.1.xx benutzt, dann brauchen Sie nur Rechnern, deren IP-Adresse mit 192.168.1 beginnt, den Zugriff zu erlauben. Die Netzwerkmaske ist also 255.255.255.0 bzw. 24.

```
acl lokal 192.168.1.0/24
```

Diese Zeile können Sie nach der `acl heuteverboten` einfügen.

Die letzten Zeilen des Ausschnittes aus der Konfigurationsdatei lauten dann:

```
http_access deny heuteverboten
http_access lokal allow
http_access deny all
```

Damit haben Sie Ihren Squid gegen Zugriffe von Rechnern aus fremden Netzen geschützt.

13.5 Browser der (Windows)-Clients einstellen

Clients müssen in ihren Browsern den Proxy-Cache aktivieren, damit sie ihn nutzen können. Dazu muss man im jeweiligen Browser die IP-Adresse des Proxy-Servers und seine Portnummer (voreingestellt 3128) eintragen.

Den Netscape Communicator konfiguriert man mit:

Bearbeiten • Einstellungen • Erweitert • Proxies • Manuelle Proxy-Konfiguration

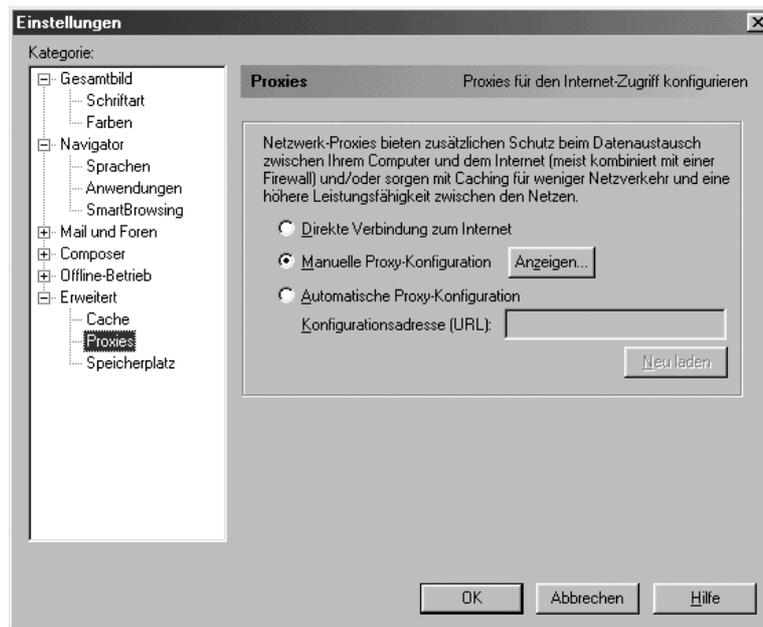


Abbildung 13.6: Einstellungen im Netscape Communicator

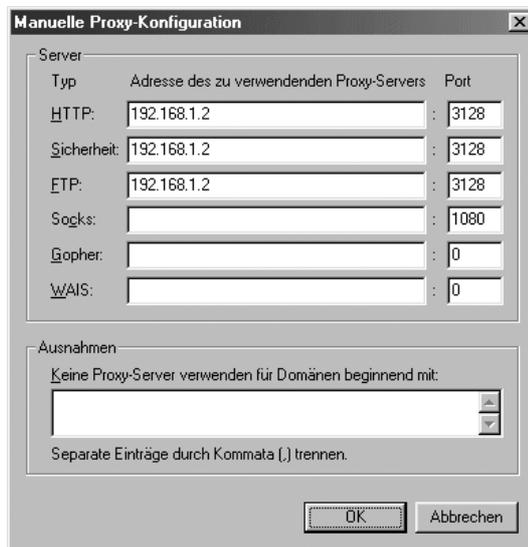


Abbildung 13.7: Manuelle Proxy-Konfiguration im Netscape Communicator

Für HTTP, HTTPS (Sicherheit) und FTP gibt man die IP-Nummer oder den Namen des Kommunikations-Servers und den Port 3128, die Voreinstellung von Squid an.

Die restlichen Zeilen bleiben wie voreingestellt. In dem großen Eingabefeld kann man Adressen (im lokalen Netz) angeben, für die der Browser den Proxy nicht benutzen soll.

Beim Microsoft Internet Explorer finden sich die gleichen Einstellmöglichkeiten unter

Extras • Internetoptionen • Verbindungen • LAN-Einstellungen

Geht man hier auf *Erweitert*, so öffnet der Explorer einen weiteren Dialog mit der praktischen Einstellmöglichkeit *Für alle Protokolle denselben Server verwenden*.

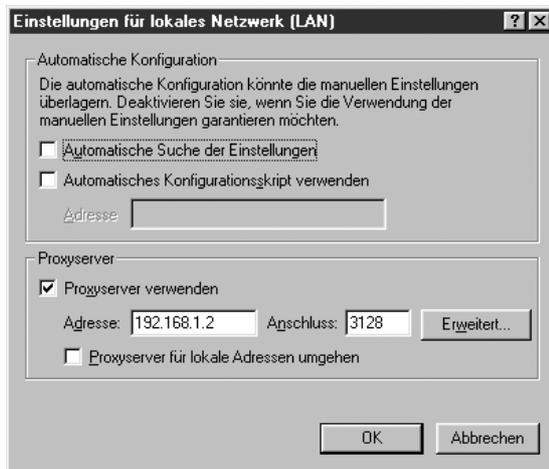


Abbildung 13.8: Menü Verbindung im Internet Explorer



Abbildung 13.9: Menü Proxy-Einstellungen im Internet Explorer

Auch hier kann man wieder die lokalen Adressen ausnehmen.

Achtung: Wenn auf dem Kommunikations-Server IP-Masquerading aktiviert ist, können Anwender den Proxy umgehen, indem sie im Browser die Proxy-Einstellungen deaktivieren.

13.6 Die Logdateien des Squid

Die folgenden Logdateien helfen Systembetreuern, Squid zu überwachen. Die angegebenen Pfade beziehen sich auf SuSE 7.3 und können bei anderen Distributionen abweichen.

<i>Datei</i>	<i>Bedeutung</i>
/var/squid/squid.out	Startmeldungen
/var/run/squid.pid	Prozess-ID
/var/squid/logs/cache.log	Sehr ausführliche Meldungen und Statistik-Informationen des Squid
/var/squid/logs/access.log	Hier wird jeder einzelne Zugriff auf den Proxy protokolliert. Das Format der Datei ähnelt dem der HTTP-Logdatei.
/var/squid/logs/store.log	Verzeichnis der gespeicherten Dateien mit Speicherort und Web-Quelle
/var/squid/cache/*	Vielzahl nummerierter Verzeichnisse, die den eigentlichen Cache bilden

Tabelle 13.2: Logdateien des Squid

Normalerweise interessiert es weniger, wo der Squid welche Datei abgelegt hat. Um bestimmten Zugriffen nachzugehen, will man feststellen können, wer welche Internetseiten aufgerufen hat. Dazu braucht man sich nur die Datei `access.log` anzuschauen. Eine typische Zeile sieht folgendermaßen aus:

```
065134.118    359 192.168.1.56 TCP_MISS/304 346 GET
➔ http://www.linuxbu.ch/ - DIRECT/www.linuxbu.ch -
```

In der ersten Spalte stecken Datum und Uhrzeit, leider nicht in einem menschenlesbaren Format, sondern als UNIX-Zeit, d.h. als Sekunden seit der Geburt der Programmiersprache C, (dem 1.1.1970). In der dritten Spalte steckt die Information, von welchem Rechner aus die Seite aufgerufen wurde und in der sechsten Spalte die URL. Will man diese Datei häufiger kontrollieren, sollte man das Logfile-Format für die Zeitangabe ändern. Aktiviert man in der `squid.conf` den Schalter

```
emulate_httpd_log on,
```

so legt er die Zeitangaben lesbar ab:

```
192.168.1.56 - - [03/Jan/2002:14:41:55 +0100] "GET
➔ http://www.linuxbu.ch/ HTTP/1.0" 304 346 TCP_MISS:DIRECT
```

Über die Datei `access.log` können Sie alle über den Proxy vermittelten Web-Zugriffe aus Ihrem Netz heraus nachvollziehen. In der ersten Spalte eines Eintrages steht immer die IP-Adresse des Rechners, der eine Seite aufgerufen hat. Danach folgen Datum und Uhrzeit, sowie die URL des angeforderten Dokumentes. Zuletzt kommen dann noch der Statuscode des Web-Servers, die Dateigröße und ob Squid das Dokument bereits im Cache vorgefunden hat oder nicht.

Bei den umfangreichen Möglichkeiten der Überwachung darf man die geltenden Gesetze und Vorschriften nicht aus dem Auge verlieren. Dazu gehören:

- Bundesdatenschutzgesetz,
- Landesdatenschutzgesetz des jeweiligen Bundeslandes und
- Telekommunikationsgesetz.

Daher ist es zwingend erforderlich, mit allen Benutzern genaue Regelungen für die Internet-Nutzung und die mögliche Überwachung dieser Regeln zu vereinbaren.

13.7 Cache-Dateien überwachen

In sehr aktiven Umgebungen kann es gelegentlich zu Problemen bei vielen gleichzeitig offenen Dateien kommen. Voreingestellt sind 8 MB Hauptspeicher und etwa 100 MB Festplattenspeicher für den Squid. Wird der Festplattenplatz wirklich ausgenutzt, dann kommt der Squid gelegentlich mit der maximalen Zahl gleichzeitig offener Dateien in Schwierigkeiten. Bei überlasteten Verbindungen ins Internet kann es auch dazu kommen, dass Squid unvollständig geladene Dateien im Cache speichert.

Sollte einer dieser Effekte auftreten, oder finden sich in der Datei `/var/log/warn` vermehrt Fehlermeldungen des Squid, so kann man einfach den kompletten Cache löschen. Dazu geht man folgendermaßen vor:

```
rscsquid stop
```

beendet den Squid. Man sollte ihm aber zum Beenden mindestens 30 Sekunden Zeit lassen, bevor man weitermacht. Die Zeile

```
rm -r /var/squid/cache/*
```

löscht einfach vollständig alle Cache-Ordner.

Vom root-Account aus richtet man die Cache-Ordner neu ein mit:

```
su squid -c "/usr/sbin/squid -z"
```

Danach kann man Squid wieder starten

```
rscsquid start
```

13.8 Auswertung mit Webalizer

Im Abschnitt 13.6 haben Sie gelesen, wie die Logdatei des Squid aufgebaut ist und wie Sie diese analysieren können. Manchmal ist man aber an statistischen Aussagen über die Squid-Nutzung interessiert. Es kann z.B. interessant sein festzustellen, welche Internet-Seiten die Nutzer am häufigsten aufrufen, eine Auswertung ähnlich wie die Auswertung der Zugriffe auf den Webserver Apache.

Die Datei `/var/squid/logs/access.log` ähnelt in ihrem Aufbau der Logdatei des Webservers Apache, vor allem wenn Sie wie beschrieben die http-Emulation aktivieren.

Daher können Sie auch diese Datei mit dem Programm Webalizer auswerten. Webalizer haben Sie bereits im Kapitel 6 kennen gelernt und vermutlich auch installiert.

Die folgende Beschreibung geht davon aus, dass Sie die Squid-Auswertung zusätzlich zu einer eventuell vorhandenen Webserver-Auswertung nutzen wollen.

Sie müssen ein Verzeichnis einrichten, in das Webalizer die FTP-Statistik ablegen kann. Eine Möglichkeit wäre `/usr/local/httpd/htdocs/squidalizer`:

```
mkdir /usr/local/httpd/htdocs/squidalizer
```

Die Lage der Logdateien unterscheidet sich zwischen Squid und Apache. Daher müssen Sie für die Squid-Auswertung auch eine spezielle Konfigurationsdatei erstellen.

Zum Erzeugen dieser zweiten Konfigurationsdatei sollten Sie einfach die vorhandene Datei kopieren, z.B. als `squidalizer.conf`:

```
cp /etc/webalizer.conf /etc/squidalizer.conf
```

Nun müssen Sie diese Datei für die Pfade des Squid anpassen, damit der Webalizer die richtige Logdatei bearbeitet.

`/etc/squidalizer.conf` (Auszug ab Zeile 23)

```
# LogFile defines the web server log file to use.  If not
# specified here or on the command line, input will default
# to STDIN.  If the log filename ends in '.gz' (ie: a gzip
# compressed file), it will be decompressed on the fly as it
# is being read.
```

```
LogFile          /var/squid/logs/access.log
```

```
# LogType defines the log type being processed.  Normally, the
```

```
# Webalizer expects a CLF or Combined web server log as input.
# Using this option, you can process ftp logs as well (xferlog
# as produced by wu-ftp and others), or Squid native logs.
# Values can be 'clf', 'ftp' or 'squid', with 'clf' the
# default. LogType clf

# OutputDir is where you want to put the output files. This
# should be a full path name, however relative ones might work
# as well.
# If no output directory is specified, the current directory
# will be used.

OutputDir      /usr/local/httpd/htdocs/squidalyzer
```

Damit ist die Konfiguration bereits funktionsfähig und Sie können den Webalizer mit dieser Konfigurationsdatei starten

```
webalizer -c /etc/squidalyzer.conf
```

Natürlich können Sie auch diese Squid-Daten automatisch auswerten, indem Sie den Programmaufruf in die Cron-Tab von root aufnehmen.

13.9 Benutzer authentifizieren

Im Abschnitt 13.8 konnten Sie lesen, wie Sie die Logdateien des Squid statistisch auswerten können. Manchmal kann es aber auch wichtig sein, zu überprüfen, welche Seiten einzelne Nutzer aufrufen.

Speziell bei der Nutzung in Schulen oder Jugendeinrichtungen lässt sich so feststellen, ob jemand und wer irgendwelche strafbaren oder jugendgefährdenden Seiten aufruft.

Achtung: Bevor Sie beginnen, Log-Dateien personenbezogen auszuwerten, sollten Sie diese Schritte rechtlich absichern. Dazu kann eine Vereinbarung mit den Nutzern bzw. dem Personalrat erforderlich sein.

Die Zuordnung zwischen einer aufgerufenen Seite und dem Nutzer, der sie aufgerufen hat, ist auch mit der bisherigen Konfiguration schon möglich, aber aufwändig. In der Datei `access.log` finden Sie für jede aufgerufene Seite die IP-Adresse des Rechners, von dem aus jemand die Seite aufgerufen hat. In den Samba-Logdateien können Sie dann feststellen, welcher Benutzer sich zu diesem Zeitpunkt an dem Rechner angemeldet hatte.

Das ist mühsam und funktioniert auch nur dann, wenn Sie eine Benutzeranmeldung am Netzwerk erzwingen (siehe Kapitel 9).

Squid verfügt über eine eigene Benutzer-Authentifizierung, über die Sie erreichen können, dass Benutzer nur nach Angabe ihres Benutzernamens und ihres Passworts auf Webseiten zugreifen können. Zusätzlich trägt Squid den Benutzernamen dann auch in die Datei `access.log` ein, was die Zuordnung der Aufrufe sehr erleichtert.

Im folgenden Auszug aus der Datei `access.log` ist die Benutzer-Authentifizierung aktiviert :

```
192.168.1.40 - - [28/Jan/2001:12:04:10 +0100] "GET
  ➤ http://www.linuxbu.ch/" 407 1
415 TCP_DENIED:NONE
192.168.1.40 debacher - [28/Jan/2001:12:04:19 +0100] "GET
  ➤ http://www.linuxbu.ch/
" 200 1627 TCP_MEM_HIT:NONE
192.168.1.40 debacher - [28/Jan/2001:12:04:19 +0100] "GET
  ➤ http://www.linuxbu.ch/
kopf.htm" 304 232 TCP_IMS_HIT:NONE
192.168.1.40 debacher - [28/Jan/2001:12:04:19 +0100] "GET
  ➤ http://www.linuxbu.ch/
links.htm" 304 233 TCP_IMS_HIT:NONE
```

Beim ersten Zugriff auf das Internet blockt Squid ab und erzwingt eine Benutzeranmeldung. Nach erfolgreicher Anmeldung liefert er die ursprünglich angeforderte Seite, im vorliegenden Fall sogar aus seinem Speicher. In jeder Zeile steht nun hinter der IP-Adresse des Rechners auch der Name des angemeldeten Benutzers.

Um die Authentifizierung zu aktivieren, benötigen Sie ein externes Modul, das für den Squid Benutzernamen und Passwort überprüft. Außerdem müssen Sie die Konfigurationsdatei `squid.conf` ein wenig bearbeiten.

Die externen Module funktionieren recht einfach. Nach dem Start erwarten sie ständig die Angabe von Benutzernamen und Passwort und liefern dann OK oder ERR zurück. Ein derartiges Modul können Sie daher auch leicht selber schreiben; in den folgenden Abschnitten lernen Sie die beiden Module `smb_auth`, `ncsa_auth` und `pam_auth` kennen.

Sie sollten sich je nach Ihrer Situation für eines der Module entscheiden. Wenn in Ihrem Netz Squid auf dem gleichen Rechner läuft, auf dem Sie Ihre Benutzer verwalten, sollten Sie `nasa_auth` oder `pam_auth` benutzen, da beide sehr schnell arbeiten. Läuft Squid nicht auf Ihrem Anmeldeserver, so müssen Sie `smb_auth` benutzen, da dieses auf einen beliebigen Samba-Server im Netz zugreifen kann. Sie benötigen dann keine Passwortdatei auf Ihrem Squid-Rechner und authentifizieren Benutzer über das Netz.

13.9.1 Das Modul `smb_auth`

Mit diesem Modul können Sie die Benutzeranmeldung von einem Samba-Server bestätigen lassen. Nutzen Sie dort ältere SuSE-Versionen als 6.4, so müssen Sie Samba auf dem Anmelde-Server gegebenenfalls aktualisieren. Sie finden das Paket `smb_auth` in der Serie `n` bzw. in der Datei `smb_auth.rpm` im Verzeichnis `n2`. Installieren Sie dieses Paket doch nach.

Damit steht Ihnen das Modul an der Adresse `/usr/bin/smb_auth` zur Verfügung.

Bevor Sie es testen können, müssen Sie auf Ihrem Anmelde-Server noch eine Datei `proxyauth` erstellen, die nur das Wort `allow` enthält. Diese Datei muss über die Freigabe `netlogon` erreichbar sein. Gemäß der Beschreibung aus Kapitel 9 wäre dies das Verzeichnis `/home/netlogon/`; schauen Sie in der Samba-Konfigurationsdatei Ihres Netzwerks nach, was Sie eingestellt haben.

Sie können das Modul ohne Squid testen. Sie müssen nur die Arbeitsgruppe wissen, für die die Anmeldung erfolgen soll. Gemäß der Beschreibung aus Kapitel 9 heißt diese einfach `ARBEITSGRUPPE`. Damit ergibt sich folgender Aufruf:

```
/usr/bin/smb_auth -W ARBEITSGRUPPE
```

Der Eingabeprompt erscheint nicht wieder, weil das Programm auf Eingabe wartet. Geben Sie nun Ihren Benutzernamen und nach einem Leerzeichen Ihr Passwort ein, folgt die Ausgabe `OK`. Geben Sie einen falschen Benutzernamen oder ein falsches Passwort an, gibt das Modul `ERR` aus.

Wenn das so klappt, ist Ihr Modul einsatzbereit. Falls es Probleme gibt, können Sie `smb_auth` mit dem zusätzlichen Parameter `-d` (debug) aufrufen.

```
root@boss:~ > /usr/bin/smb_auth -W ARBEITSGRUPPE -d
gast gast
Domain name: ARBEITSGRUPPE
Pass-through authentication: no
Query address options:
```

```
Domain controller IP address: 192.168.1.2
Domain controller NETBIOS name: SERVER
Contents of //SERVER/NETLOGON/proxyauth: allow
OK
```

Beenden können Sie den Dialog mit dem Modul über den Tastendruck auf `[Strg]+[D]`.

13.9.2 Das Modul *nlsa_auth*

Das Modul *nlsa_auth* ist deutlich schneller als *smb_auth*, kann aber nicht über das Netz arbeiten. Das Modul gehört zu Squid und wird mit ihm zusammen installiert

Auch bei diesem Modul können Sie die Funktionsfähigkeit ohne Squid testen. Das Modul erwartet als Aufruf-Parameter den Namen der Passwortdatei, gegen die es prüfen soll. Rufen Sie es mit dem Parameter `/etc/shadow` auf, wenn Sie die Passwort-Datei des Systems benutzen wollen:

```
/usr/sbin/nlsa_auth /etc/shadow
```

Sie können aber auch Passwort-Dateien angeben, die Sie mit dem *htpasswd*-Programm des Apache erzeugt haben (siehe Kapitel 6). Damit können Sie nur ausgewählten Nutzern den Internet-Zugriff erlauben. Wenn Sie allen Benutzern die Anmeldung erlauben wollen, dann sollten Sie besser das Modul *pam_auth* verwenden.

Auch dieses Modul erwartet wieder in einer Eingabezeile einen Benutzernamen und das dazugehörige Passwort, getrennt durch ein Leerzeichen, und liefert OK bzw. ERR zurück.

13.9.3 Das Modul *pam_auth*

Aus Sicherheitsgesichtspunkten am sinnvollsten ist das Modul *pam_auth*. Es braucht im Gegensatz zu *nlsa_auth* keinen direkten Zugriff auf die Passwort-Datei, sondern versucht mit den übergebenen Benutzerdaten ein Login. Wenn das Login klappt, dann sind die übergebenen Daten in Ordnung und das Modul liefert OK zurück. Wenn das Login nicht klappt, dann liefert das Modul ERR zurück.

Sie können auch dieses Modul von der Eingabezeile aus testen mit:

```
/usr/sbin/pam_auth
```

13.9.4 squid.conf anpassen

Nachdem Sie eines der Authentifizierungs-Module installiert haben, können Sie es in den Squid einbinden. Dazu müssen Sie ein paar Zeilen der `squid.conf` verändern. Zuerst geben Sie das Authentifizierungs-Programm an.

`/etc/squid.conf` (Auszug ab Zeile 690)

```
# TAG: authenticate_program
#   Specify the command for the external authenticator.
#   Such a program reads a line containing "username
#   password" and replies "OK" or "ERR" in an endless
#   loop.  If you use an authenticator,
#   make sure you have 1 acl of type proxy_auth.
#   By default, the authenticator_program is not used.
#
#   If you want to use the traditional proxy
#   authentication, jump over to the ../auth_modules/NCSA
#   directory and type:
#       % make
#       % make install
#
#   Then, set this line to something like
#
#   authenticate_program /usr/bin/ncsa_auth
#   /usr/etc/passwd
#
#authenticate_program none
authenticate_program /usr/sbin/pam_auth
#authenticate_program /usr/sbin/ncsa_auth /etc/shadow
#authenticate_program /usr/bin/smb_auth -W ARBEITSGRUPPE
```

In der hier gedruckten Erweiterung finden Sie die Einträge für alle drei Module, es darf aber immer nur eine der Zeilen aktiviert sein.

Nun müssen Sie noch Zugriffs-Regeln einfügen, die nur autorisierten Benutzern einen Zugriff erlauben. Dazu gehören je eine zusätzliche `acl`-Zeile (im Listing hervorgehoben) und je eine `http_access` Zeile:

`/etc/squid.conf` (Auszug ab Zeile 1178):

```
#Defaults:
acl all src 0.0.0.0/0.0.0.0
acl manager proto cache_object
acl localhost src 127.0.0.1/255.255.255.255
acl SSL_ports port 443 563
```

```
acl Safe_ports port 80 21 443 563 70 210 1025-65535
acl Safe_ports port 280          # http-mgmt
acl Safe_ports port 488          # gss-http
acl Safe_ports port 591          # filemaker
acl Safe_ports port 777          # multiling http
acl CONNECT method CONNECT
# proxy_auth
acl domainusers proxy_auth REQUIRED

# TAG: http_access
#     Allowing or Denying access based on defined
#     access lists
#
#     Access to the HTTP port:
#     http_access allow|deny [!]aclname ...
#
#     Access to the ICP port:
#     icp_access  allow|deny [!]aclname ...
#
#     NOTE on default values:
#
#     If there are no "access" lines present, the default is
#     to allow the request.
#
#     If none of the "access" lines cause a match, the
#     default is the opposite of the last line in the list.
#     If the last line was deny, then the default is allow.
#     Conversely, if the last line is allow, the default
#     will be deny.  For these reasons, it is a
#     good idea to have an "deny all" or "allow all" entry
#     at the end of your access lists to avoid potential
#     confusion.
#
#Default configuration:
http_access allow manager localhost
http_access deny manager
http_access deny !Safe_ports
http_access deny CONNECT !SSL_ports
#
# INSERT YOUR OWN RULE(S) HERE TO ALLOW ACCESS FROM YOUR
# CLIENTS
#
```

```
http_access allow localhost
#http_access allow all
http_access deny !domainusers
```

Die bisherige Zeile `http_access allow all` müssen Sie deaktivieren und die neue Zeile einfügen, die allen nicht bestätigten Benutzern den Zugriff verweigert.

Nach einem Neustart des Squid ist Ihre Benutzer-Authentifizierung aktiviert und Ihre Benutzer müssen sich beim ersten Zugriff auf eine Internet-Seite beim Squid anmelden.

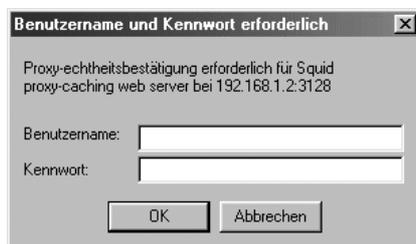


Abbildung 13.10: Authentifizierung für Squid

13.9.5 Feintuning

Das hier beschriebene Verfahren der Benutzer-Anmeldung am Squid hat einen Haken. Programme wie der Realplayer, die einen Proxy benutzen, aber keine Passwörter speichern können, bekommen keinen Internet-Zugriff mehr.

Hier können Sie über Firewall-Regeln (siehe Kapitel 14) den Zugriff auf den zugehörigen Port gezielt freigeben.

```
/usr/sbin/iptables -I FORWARD --dport rtsp -p tcp -j ACCEPT
/usr/sbin/iptables -I FORWARD --dport rtsp -p udp -j ACCEPT
```

Hiermit bekommt der Realplayer einen direkten Internet-Zugriff und muss nicht mehr über den Squid aufs Internet zugreifen.

Sollte Ihnen der Text `Squid proxy-caching web server` im Anmeldefenster nicht gefallen, so können Sie auch einen individuelleren Text vorgeben.

/etc/squid.conf (Auszug ab Zeile 1255)

```
# TAG: proxy_auth_realm
#     Specifies the realm name which is to be reported to
#     the client for proxy authentication (part of the text
#     the user will see when prompted their username and
#     password).
#
#proxy_auth_realm Squid proxy-caching web server
proxy_auth_realm Internet-Zugriff
```

Sie können den Text natürlich nach Ihren Vorstellungen gestalten.