

# 11 Webseiten im Proxy-Cache zwischenspeichern und filtern

Das World Wide Web wird oft lästerlich World Wide Wait genannt, weil immer mehr Anwender immer mehr Seiten anfordern, als Netzanbieter Bandbreite für Anwender liefern können.

Anwender können Webseiten schneller abrufen, wenn sie

- mit ihren Providern 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.

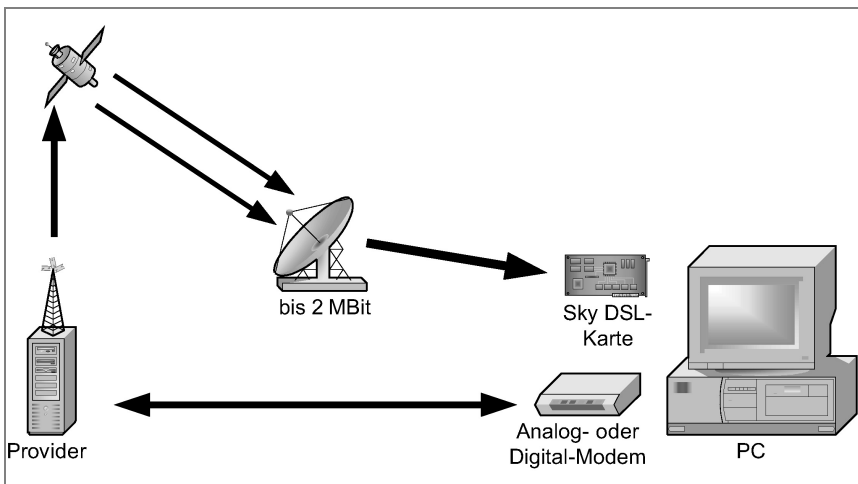
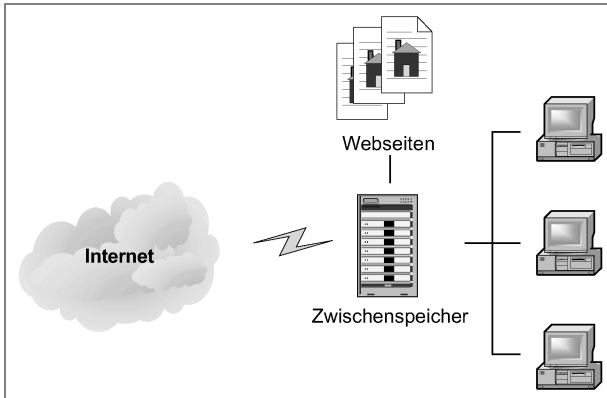


Abbildung 11.1: Hohe Bandbreite, z. B. durch satellitenbasiertes DSL



**Abbildung 11.2:** Webseiten im Proxy-Cache

Werden Internetseiten in geschützten Umgebungen wie Betrieben, Schulen, Internet-Cafes, zu denen auch minderjährige Besucher Zugang haben, oder Familien abgerufen, sind Systemverantwortliche gefordert, neben schnellem Seitenabruf auch Filter und Inhaltskontrollen einzurichten.

Webzugriffe lassen sich durch verschiedene Zwischenspeicher beschleunigen und filtern:

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

Fast alle Anwender nutzen lokale Zwischenspeicher für Internetseiten, Cache genannt. Anfänger tun dies sogar ohne es zu wissen, weil Webbrowser diese Funktion schon in ihrer Grundausstattung bieten. Löschen Anwender den Verlauf ihrer Websitzungen beim Verlassen des Surfplatzes nicht, können Dritte ausspionieren, welche Websites sie besucht haben.

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

Im Browser Mozilla Firefox finden Sie die Cache-Einstellungen unter *Bearbeiten • Einstellungen • Erweitert • Netzwerk*.

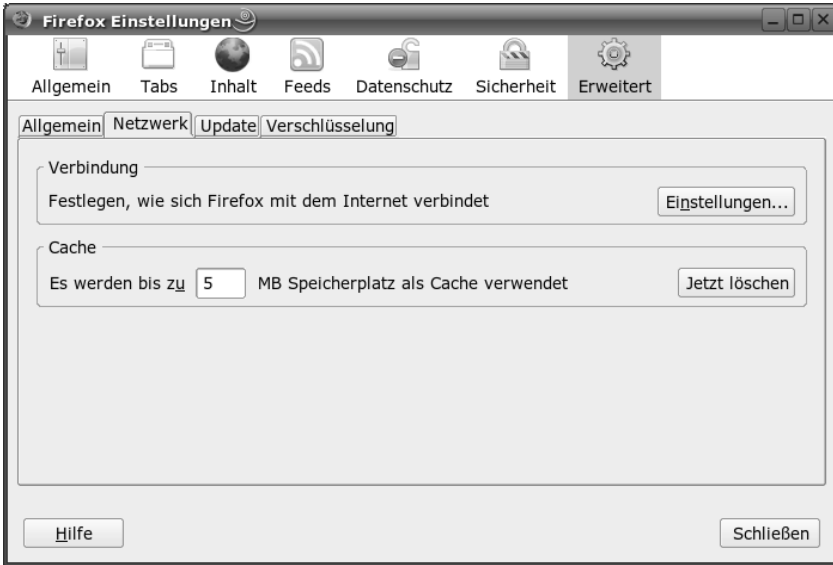


Abbildung 11.3: Cache-Einstellungen in Mozilla

Das entsprechende Menü finden Sie beim Internet Explorer unter *Extras • Internetoptionen • Temporäre Internetdateien • Einstellungen*.



Abbildung 11.4: Cache-Einstellungen im Internet Explorer

Bei Opera finden Sie die Cache-Einstellungen unter *Extras • Einstellungen • Erweitert • Verlauf*

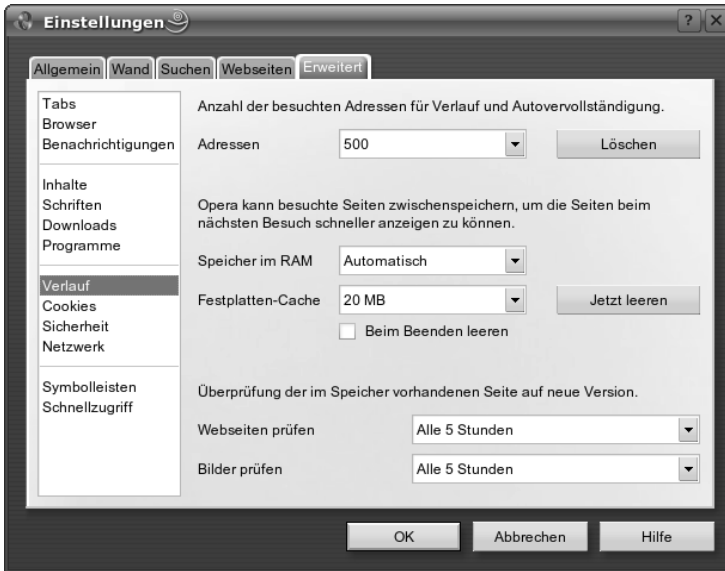


Abbildung 11.5: Cache-Einstellungen bei Opera

Um in Unternehmen, Bildungseinrichtungen und sonstigen Organisationen im Interesse des Jugendschutzes Seiten zu filtern, benötigt man Zusatzprogramme, die Seiten mit unerwünschten Inhalten wie bestimmten Text- oder Grafikobjekten und von einschlägigen Websites ignorieren.

Hierzu kann man in lokalen Netzen die von den Anwendern besuchten Seiten zentral speichern. Einerseits kommt der Geschwindigkeitsvorteil für das erneute Laden allen zugute, da die Ladezeiten im lokalen Netz vergleichsweise kurz sind. Andererseits können Systemverwalter damit das Surfverhalten der Anwender auch dann überwachen, wenn diese die lokalen Speicher nach ihrem jeweiligen Arbeitsende löschen.

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

**Hinweis:** Wenn Sie einen Proxy-Server wie Squid nutzen, können Sie den lokalen Cache im Browser entweder ganz deaktivieren oder auf einen möglichst kleinen Wert reduzieren.

Zusätzlich zu der Cache-Funktion verfügt Squid über eine Stellvertreterfunktion (Proxy). Bei der Einwahl ins Internet stellt der Provider 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 Webserver keine Antworten schicken können. Diese lokalen Rechner fordern WWW-Seiten indirekt von Squid an. Dieser ruft

sie mit der IP-Adresse des Linux-Routers aus dem Internet ab, sofern er sie nicht schon lokal gespeichert hat.

## 11.1 Wann lohnt sich ein Proxy-Cache?

Ein Proxy-Cache hat mehrere Aufgaben und Vorteile:

- er beschleunigt den Internetzugriff, da schon einmal geladene Seiten nicht erneut übertragen werden,
- er hat eine Stellvertreterfunktion für die Rechner im Netz;
- er kann kontrollieren, welche Inhalte die Benutzer im lokalen Netz anfordern dürfen und
- er dokumentiert, wer wann von welchem Endgerät welche Webseiten angefordert hat.

Wie sehr der Proxy-Server das Laden von Webseiten dadurch beschleunigt, dass 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 sich eine – vielleicht nur schmalbandige – Internetanbindung teilen müssen.

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

Will man den lokalen Rechnern erlauben, unter Umgehung des Proxy direkt auf Webserver 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 eine Firewall einrichten und betreiben (siehe dazu Kapitel 12).

Proxies können gezielt einzelne Seiten oder ganze Internet-Domains sperren, damit kein Browser, der sie anfordert, solche Seiten überhaupt sehen oder laden kann.

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

## 11.2 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 im lokalen Netz und stellt sie dann den Browsern der Clients zur Verfügung.

## 11.3 Squid installieren und konfigurieren

Da alle Linux-Distributionen Squid enthalten, lässt er sich einfach durch Auswahl des zugehörigen Pakets einrichten. Bei OpenSUSE befindet sich die aktuelle Version 3 unter dem Namen `squid3` in der Paketgruppe `Netzwerk`. Die bisherige Version des Squid steht in dieser Paketgruppe ebenfalls noch zur Auswahl. Installieren Sie das Paket `squid3` von Ihrer Installationsquelle.

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

**Tabelle 11.1:** Die Dateien zu Squid

Nach der Installation muss man dafür sorgen, dass Squid automatisch startet. Dazu rufen Sie im YaST-Kontrollzentrum unter *System* den *Runlevel-Editor* auf. Dort markieren Sie in der Liste die Zeile für den Squid und aktivieren die Runlevel 3 und 5. Anschließend können Sie YaST wieder beenden.

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

```
rscsquid start
```

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

Squid konfiguriert man über die 4.600 Zeilen große Datei `squid.conf`, die überwiegend aus Kommentaren und Dokumentation besteht. Jeder Schalter ist zuerst ausführlich dokumentiert, dabei ist auch immer die Standardeinstellung angegeben.

`/etc/squid/squid.conf` (Auszug ab Zeile 1912)

```
# 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 analyzers use.
#
#Default:
# emulate_httpd_log off
```

Die ersten acht Zeilen sind reiner 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 (siehe Log-Dateien des Squid) und
- den tatsächlichen Bedürfnissen anpassen (s. u.).

In der aktuellen OpenSUSE-Distribution ist Squid zum Glück nicht mehr so stark auf Sicherheit getrimmt, dass er auch Webzugriffe aus dem lokalen Netz ablehnt. Dies stellte in früheren Versionen zum Teil ein Problem dar.

/etc/squid/squid.conf (Auszug ab Zeile 566)

```
#Recommended minimum configuration:
acl manager proto cache_object
acl localhost src 127.0.0.1/32
acl to_localhost dst 127.0.0.0/8
#
# Example rule allowing access from your local networks.
# Adapt to list your (internal) IP networks from where browsing
# should be allowed
acl localnet src 10.0.0.0/8      # RFC1918 internal network
acl localnet src 172.16.0.0/12 # RFC1918 internal network
acl localnet src 192.168.0.0/16 # RFC1918 internal network
#
acl SSL_ports port 443
acl Safe_ports port 80          # http
acl Safe_ports port 21          # ftp
acl Safe_ports port 443        # https
acl Safe_ports port 70          # gopher
acl Safe_ports port 210         # wais
acl Safe_ports port 1025-65535  # unregistered ports
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 ...
#
# NOTE on default values:
#
# If there are no "access" lines present, the default is to
# deny 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, 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:
# http_access deny all
#
```

Was zu tun ist, wenn eine der Regeln zutrifft, legen Sie im folgenden Abschnitt fest:

/etc/squid/squid.conf (Auszug ab Zeile 612)

```
#Recommended minimum configuration:
#
# Only allow cachemgr access from localhost
http_access allow manager localhost
http_access deny manager
# Deny requests to unknown ports
http_access deny !Safe_ports
# Deny CONNECT to other than SSL ports
http_access deny CONNECT !SSL_ports
#
# We strongly recommend the following be uncommented to protect
innocent
# web applications running on the proxy server who think the only
# one who can access services on "localhost" is a local user
#http_access deny to_localhost
#
# INSERT YOUR OWN RULE(S) HERE TO ALLOW ACCESS FROM YOUR CLIENTS
#
# Example rule allowing access from your local networks.
# Adapt localnet in the ACL section to list your (internal) IP
networks
# from where browsing should be allowed
http_access allow localnet
# allow localhost always proxy functionality
```



```
http_access allow localhost
# And finally deny all other access to this proxy
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.

Sofern Sie in Ihrem Netz einen der freien IP-Bereiche wie 192.168.1.x einsetzen, sollten die Rechner in Ihrem Netz den Squid ohne weitere Änderungen der Konfiguration nutzen können. Sollten Sie einen anderen Netzbereich gebrauchen, so fügen Sie eine Zeile nach folgendem Muster an Nummer 577 an:

```
acl localnet src xxx.yyy.zzz.0/24
```

und veranlassen Sie den Squid, seine Konfigurationsdatei neu einzulesen:

```
rcsquid reload
```

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

## 11.4 Zugriffskontrolle durch den Proxy-Cache

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

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

```
acl heutesperret 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 definierte Regel muss man noch aktivieren:

```
http_access deny heutesperret
```

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

```
http_access allow localnet
```

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.

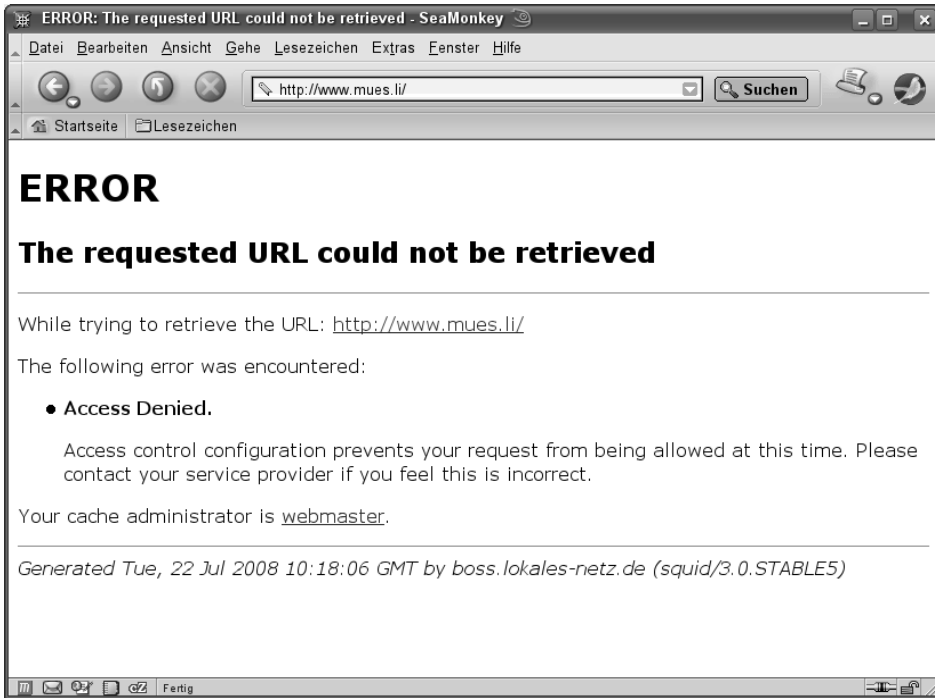


Abbildung 11.6: Zugriffsverweigerung bei gesperrten Seiten

Sie sollten auf alle Fälle in den Log-Dateien überprüfen, ob die Sperren auch wie geplant arbeiten. Hinweise zur Auswertung der Log-Dateien finden Sie im Abschnitt 11.6.

## 11.5 Browser der (Windows)-Clients einstellen

Damit Client-Browser den Proxy-Cache nutzen können, müssen Sie ihn aktivieren. Dazu muss man im jeweiligen Browser die IP-Adresse des Proxy-Servers und seine Portnummer (voreingestellt 3128) eintragen.

Den Browser Mozilla Firefox konfiguriert man unter *Extras • Einstellungen • Erweitert • Netzwerk • Verbindung • Einstellungen...*

Er öffnet ein Formular, in dem Sie Parameter für verschiedene Protokolle einstellen können.

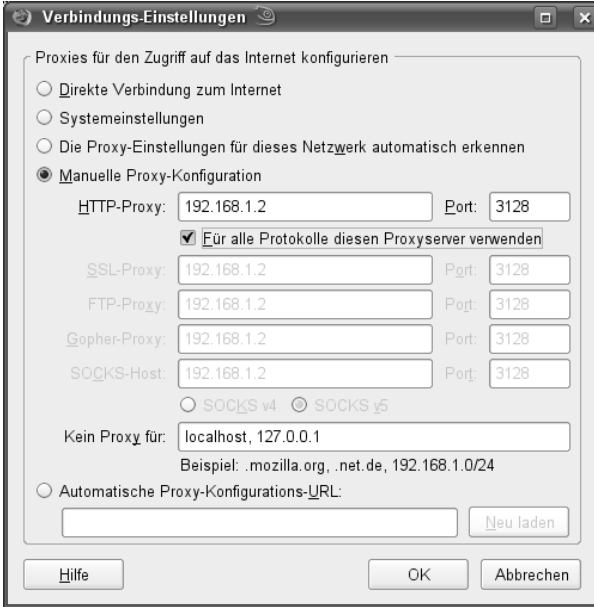


Abbildung 11.7: Proxy-Konfiguration in Mozilla

Für HTTP, HTTPS (Sicherheit) und FTP gibt man die IP-Nummer oder den Namen des Kommunikationsservers 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.

Bei Microsofts 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 11.8: Menü Verbindung im Internet Explorer

Ähnlich wie beim Mozilla Firefox gibt es auch hier ein weiteres Formular für die verschiedenen Protokolle.



Abbildung 11.9: Menü Proxy-Einstellungen im Internet Explorer

Auch hier kann man wieder die lokalen Adressen ausnehmen. Bei Opera finden Sie die entsprechenden Einstellungen in einem einzigen Formular unter *Extras • Einstellungen • Erweitert • Netzwerk • Proxyserver*.

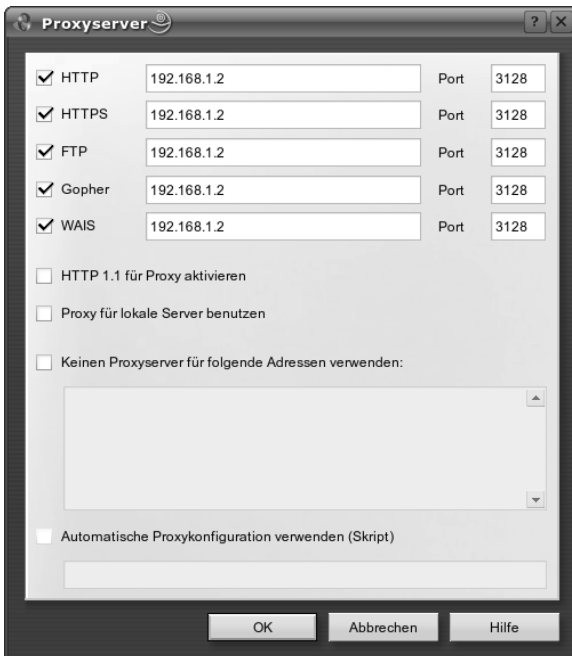


Abbildung 11.10: Menü Proxyserver bei Opera

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

## 11.6 Die Log-Dateien des Squid

Die folgenden Log-Dateien helfen Systembetreuern, Squid zu überwachen. Die angegebenen Pfade beziehen sich auf OpenSUSE 11 und können bei anderen Linux-Distributionen abweichen.

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

**Tabelle 11.2:** Log-Dateien des Squid

Normalerweise interessiert es nicht so sehr, wo der Squid welche Datei abgelegt hat. Um aber 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:

```
1167322665.280 301 192.168.1.56 TCP_MISS/200 1619 GET
http://www.linuxbu.ch/ - DIRECT/193.239.104.29 text/html
```

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. Hierzu aktiviert man in der `squid.conf` (Zeile 1920) den Schalter

```
emulate_httpd_log on
```

und in der Zeile 1834 entfernt man am Ende den Parameter `squid`, der das Logging-Format festlegt:

```
access_log /var/log/squid/access.log
```

Startet man dann den Squid neu, so legt er die Zeitangaben lesbar ab. Achten Sie darauf, dass Ihr Server seine eigene Zeit richtig synchronisiert (siehe Kapitel 10.9).

```
192.168.1.56 - - [28/Dec/2006:17:40:55 +0100] "GET
http://www.linuxbu.ch/ HTTP/1.1" 200 1618
TCP_CLIENT_REFRESH_MISS:DIRECT
```

Über die Datei `access.log` können Sie alle über den Proxy vermittelten Webzugriffe 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 Dokuments. Zuletzt kommen noch der Statuscode des Webservers, die Dateigröße und ob Squid das Dokument bereits im Cache vorgefunden hat oder nicht. Im Abschnitt 11.8 lesen Sie, wie Sie das Auswerten der Protokolle mit dem Programm `Webalizer` automatisieren.

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

- Bundesdatenschutzgesetz ([www.datenschutz.de/recht/gesetze/](http://www.datenschutz.de/recht/gesetze/)),
- Landesdatenschutzgesetz des jeweiligen Bundeslandes,
- Betriebsverfassungsgesetz und
- Telekommunikationsgesetz sowie
- eventuelle Betriebsvereinbarungen.

Daher ist es zwingend erforderlich, mit allen Benutzern genaue Regelungen für die Internetnutzung und die mögliche Überwachung der Einhaltung dieser Regeln zu vereinbaren. Unternehmen sollten in Arbeitsverträgen klar regeln, welche privaten Nutzungen des Internet sie ihren Mitarbeitern erlauben. Dies ist ein sehr problematischer Bereich, da Unternehmen damit in die Rolle von Internet-Providern schlüpfen. Wenn Betriebe hingegen ihren Mitarbeitern jegliche private Nutzung des Internet vom Firmen-Arbeitsplatz/Account aus untersagen, sparen sie sich viele rechtliche Risiken.

## 11.7 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 ganz ausgenutzt, kommt der Squid gelegentlich mit der maximalen Zahl gleichzeitig offener Dateien in Schwie-

rigkeiten. 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/cache/squid/*
```

löscht vollständig alle Cache-Ordner.

Sie müssen darauf achten, als Benutzer `squid` angemeldet zu sein, wenn Sie die Cache-Dateien neu einrichten. Vom `root`-Account aus richtet man die Cache-Ordner neu ein mit:

```
su squid
/usr/sbin/squid -z
exit
```

Danach kann man Squid wieder starten

```
rscsquid start
```

## 11.8 Auswertung mit Webalizer

Im Abschnitt 11.6 haben Sie gelesen, wie die Log-Datei 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 Internetseiten die Nutzer am häufigsten aufrufen, eine Auswertung ähnlich wie die Auswertung der Zugriffe auf den Webserver Apache.

Die Datei `/var/log/squid/access.log` ähnelt in ihrem Aufbau der Log-Datei 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 (»Informationen per Webserver verteilen«) kennengelernt 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 Squid-Statistik ablegen kann. Eine Möglichkeit wäre `/srv/www/htdocs/squidalizer`:

```
mkdir /srv/www/htdocs/squidalizer
```

Squid und Apache speichern ihre Log-Dateien an verschiedenen Orten. 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 `squidalyzer.conf`:

```
cp /etc/webalizer.conf /etc/squidalyzer.conf
```

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

`/etc/squidalyzer.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/log/squid/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       /srv/www/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 Crontab von `root` aufnehmen.

## 11.9 Benutzer authentifizieren

Im vorigen Abschnitt haben Sie erfahren, wie Sie die Log-Dateien des Squid statistisch auswerten können. Manchmal kann es aber auch wichtig sein, zu überprüfen, welche Seiten einzelne Nutzer aufrufen.



Speziell bei der Nutzung in Betrieben, Schulen oder Jugendeinrichtungen lässt sich so feststellen, ob jemand (und wer) 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 Betriebs- oder Personalrat über die Nutzung des Internet und über die Auswertung solcher Nutzungen erforderlich sein.

Die Zuordnung zwischen einer aufgerufenen Seite und dem Nutzer, der sie aufgerufen hat, ist auch mit der bisherigen Konfiguration möglich, aber aufwändig. Die Datei `access.log` enthält für jede aufgerufene Seite die IP-Adresse des Rechners, von dem aus jemand die Seite aufgerufen hat. In der Datei `lastlog` bzw. den Log-Dateien von Samba 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 Abschnitt 9).

Mit der eigenen Benutzerauthentifizierung von Squid kann man erreichen, dass Benutzer nur nach Angabe ihres Benutzernamens und ihres Passworts auf Webseiten zugreifen können. Squid trägt dann in die Datei `access.log` auch den Benutzernamen ein, was die Zuordnung und Auswertung der Aufrufe sehr erleichtert.

Im folgenden Auszug aus der Datei `access.log` ist die Benutzerauthentifizierung aktiviert:

```
192.168.1.56 - - [30/Jul/2006:19:14:15 +0200] "GET
http://www.linuxbu.ch/ HTTP/1.0" 407 1764 TCP_DENIED:NONE
192.168.1.56 - debacher [30/Jul/2006:19:14:21 +0200] "GET
http://www.linuxbu.ch/ HTTP/1.0" 200 1610 TCP_SWAPFAIL_MISS:DIRECT
192.168.1.56 - debacher [30/Jul/2006:19:14:21 +0200] "GET
http://www.linuxbu.ch/links.htm HTTP/1.0" 200 2313 TCP_HIT:NONE
```

Beim ersten Zugriff auf das Internet blockt Squid 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; die folgenden Abschnitte beschreiben die Module `smb_auth`, `ncsa_auth`, `pam_auth` und `squid_ldapauth`. Alle vier Module sind Bestandteil des Pakets `squid`.

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 `ncsa_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.

### 11.9.1 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 Anmeldeserver gegebenenfalls aktualisieren. Das Modul `smb_auth` haben Sie bereits zusammen mit dem Paket `squid` installiert. Es steht Ihnen unter `/usr/sbin/smb_auth` zur Verfügung.

Bevor Sie das Modul testen können, müssen Sie auf Ihrem Anmeldeserver 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 kennen, für die die Anmeldung erfolgen soll. Gemäß der Beschreibung aus Kapitel 9 heißt diese einfach `ARBEITSGRUPPE`. Damit ergibt sich folgender Aufruf:

```
/usr/sbin/smb_auth -W ARBEITSGRUPPE
```

Der Eingabeprompt erscheint nicht wieder, weil das Programm auf eine Eingabe wartet.

Wenn Sie nun Ihren Benutzernamen und nach einem Leerzeichen Ihr Passwort eingeben, antwortet das Modul mit `OK` beziehungsweise bei einem falschen Benutzernamen oder einem falschen Passwort mit `ERR`.

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

```
root@boss:~ > /usr/sbin/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: BOSS
Contents of //BOSS/NETLOGON/proxyauth: allow
OK
```

Beenden können Sie den Dialog mit dem Modul per Tastendruck auf `Strg-D`.

Falls Sie auf dem Rechner das Paket `samba-client` nicht installiert haben, so erscheint beim Test die Fehlermeldung

```
/usr/sbin/smb_auth.sh: line 51: /usr/bin/nmblookup: Datei oder
Verzeichnis nicht gefunden
```

Installieren Sie in diesem Fall das Paket einfach nach. Sie finden es in der Paketgruppe *Netzwerk*.

### 11.9.2 Das Modul `nca_auth`

Das Modul `nca_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 Aufrufparameter den Namen der Passwortdatei, gegen die es prüfen soll.

```
/usr/sbin/nca_auth /etc/httpd/passwort
```

Sie können hierbei Passwortdateien angeben, die Sie mit dem `htpasswd`-Programm des Apache erzeugt haben (siehe Kapitel 6, »Informationen per Webserver verteilen«). Damit können Sie ausgewählten Nutzern den Internetzugriff 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 sowie das dazugehörige Passwort, getrennt durch ein Leerzeichen, und liefert OK bzw. ERR zurück.

### 11.9.3 Das Modul `pam_auth`

Am sichersten ist das Modul `pam_auth`. Es braucht keinen direkten Zugriff auf die Passwortdatei, sondern versucht mit den übergebenen Benutzerdaten ein Login. Wenn das Login klappt, sind die übergebenen Daten in Ordnung und das Modul liefert OK zurück. Wenn das Login nicht klappt, dann liefert es ERR zurück.

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

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

### 11.9.4 Das Modul `squid_ldapauth`

Wenn Sie in Ihrem System LDAP zur Benutzerverwaltung einsetzen (siehe Kapitel 3.5, »Benutzerverwaltung in großen Netzen«), so finden Sie auch dafür ein Authentifizierungsmodul. Das Modul `squid_ldapauth` gehört ebenfalls zum Inhalt des Squid-Paketes.

Beim Aufruf dieses Moduls müssen Sie zumindest die Base-DN angeben, den allgemeinen Bezeichner für Ihre LDAP-Objekte.

```
/usr/sbin/squid_ldap_auth -b "ou=people,dc=your,dc=domain" boss
```

Die Angabe des LDAP-Servers kann entfallen, wenn Squid auf dem gleichen Rechner läuft.

### 11.9.5 squid.conf anpassen

Nachdem Sie eines der Authentifizierungsmodule installiert haben, können Sie es in den Squid einbinden. Dazu müssen Sie ein paar Zeilen der `squid.conf` verändern, teilweise einfach durch Auskommentieren. Zuerst geben Sie das Authentifizierungsprogramm an.

`/etc/squid/squid.conf` (Auszug ab Zeile 260)

```
#Recommended minimum configuration per scheme:
#auth_param negotiate program <uncomment and complete this line to
activate>
#auth_param negotiate children 5
#auth_param negotiate keep_alive on
#auth_param ntlm program <uncomment and complete this line to
activate>
#auth_param ntlm children 5
#auth_param ntlm keep_alive on
#auth_param digest program <uncomment and complete this line>
#auth_param digest children 5
#auth_param digest realm Squid proxy-caching web server
#auth_param digest nonce_garbage_interval 5 minutes
#auth_param digest nonce_max_duration 30 minutes
#auth_param digest nonce_max_count 50
#auth_param basic program <uncomment and complete this line>
#auth_param basic children 5
#auth_param basic realm Squid proxy-caching web server
#auth_param basic credentialsttl 2 hours

auth_param basic program /usr/sbin/pam_auth
#auth_param basic program /usr/sbin/nlsa_auth /etc/shadow
#auth_param basic program /usr/sbin/smb_auth -W ARBEITSGRUPPE
#auth_param basic program /usr/sbin/squid_ldap_auth -b
"ou=people,dc=your,dc=domain" boss
```

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

Nun müssen Sie noch Zugriffsregeln 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/squid.conf` (Auszug ab Zeile 625)

```
# INSERT YOUR OWN RULE(S) HERE TO ALLOW ACCESS FROM YOUR CLIENTS

# Example rule allowing access from your local networks. Adapt
# to list your (internal) IP networks from where browsing should
# be allowed
acl  heutigesperrt  dstdomain  www.mues.li  www.wapbu.ch

http_access  deny  heutigesperrt

acl  domainuser  proxy_auth  REQUIRED
http_access  deny  !domainuser

http_access  allow  localnet
http_access  allow  localhost

# And finally deny all other access to this proxy
http_access  deny  all
```

Nach einem Neustart des Squid ist Ihre Benutzerauthentifizierung aktiviert und Ihre Benutzer müssen sich beim ersten Zugriff auf eine Internetseite beim Squid anmelden.

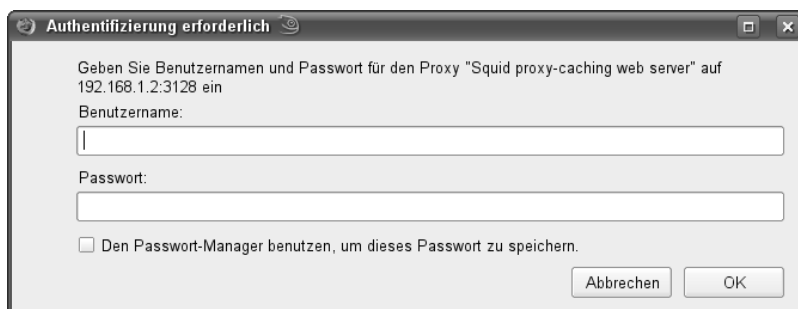


Abbildung 11.11: Authentifizierung für Squid

### 11.9.6 Feintuning

Das hier beschriebene Verfahren der Benutzeranmeldung am Squid hat einen Haken: Programme wie der Real Player, die einen Proxy benutzen, aber keine Passwörter speichern können, bekommen keinen Internetzugriff mehr.

Hier können Sie über Firewall-Regeln (siehe Kapitel 12) 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 Real Player einen direkten Internetzugriff und muss nicht mehr über Squid aufs Internet zugreifen.

Sollte Ihnen der Ausgabertext *Squid proxy-caching web server* im Anmeldefenster nicht gefallen, so können Sie auch einen selbstdefinierten Text vorgeben, z. B. *Internet-Zugriff*.

/etc/squid.conf (Auszug ab Zeile 280)

```
auth_param basic program /usr/sbin/pam_auth
#auth_param basic program /usr/sbin/ncsa_auth /etc/shadow
#auth_param basic program /usr/sbin/smb_auth -W ARBEITSGRUPPE
#auth_param basic program /usr/sbin/squid_ldap_auth -b
"ou=people,dc=your,dc=domain" boss

auth_param basic children 5
auth_param basic realm Internet Zugriff
auth_param basic credentialsttl 2 hours
```

Sie können diesen Text natürlich nach Ihren Vorstellungen gestalten.