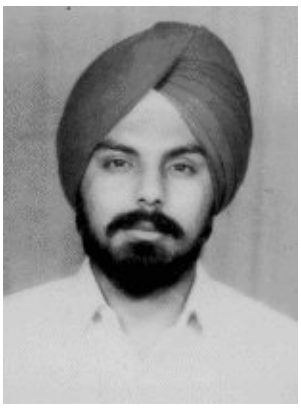
Einen Squid–Proxy Server einrichten



by D.S. Oberoi
<ds_oberoi/at/yahoo.com>

About the author:
D.S. Oberoi lebt in Jammu, Indien und hat z. Zt. wegen der anhaltenden politischen Spannungen Probleme, Verbindung zum Internet zu bekommen.

Abstract:

Linux ist ein Synonym für Vernetzung geworden. Es wird gleichermassen in Büro– wie in privaten Umgebungen als Datei–, Druck–, Email– und Anwendungsserver eingesetzt und auch in steigendem Maße als Proxy–Server benutzt.

Ein Proxy–Server bietet mehreren Benutzern Internet–Zugang zur gleichen Zeit, indem z. B. eine einzelne Internet–Verbindung geteilt wird. Ein guter Proxy–Server bietet auch die Möglichkeit, die angeforderten Daten lokal zwischenzuspeichern, um so die Daten aus lokalen Ressourcen anstelle aus dem Web zu holen und dadurch die Zugriffszeit zu reduzieren und Bandbreite zu sparen. Squid ist eine solche Software, die die Proxyfunktion sowie das lokale Speichern von HTTP–, ftp–, Gopher– und anderen Anfragen unterstützt. Es unterstützt ausserdem SSL, Zugriffskontrollen, DNS–Zwischenspeicherung und führt ein vollständiges Log über alle Anforderungen. Squid ist auch für Windows NT erhältlich von Logi Sense.

Der Schwerpunkt dieses Artikels liegt auf grundlegenden Hinweisen zum Einrichten eines Proxy–Servers und den Möglichkeiten, den Benutzerinnen kontrollierten Zugriff zu bieten.

Ist Squid Installiert ?

Die Squid–rpm–Dateien kommen zusammen mit der RedHat 7.1–Version und werden automatisch installiert, wenn die Option Netzwerk–Installation gewählt wird. Ob Squid installiert ist oder nicht, kann mit dem folgenden rpm–Befehl überprüft werden:

```
rpm -q squid
```

Die aktuellste Squid–Version kann immer von der Squid–Webseite und anderen Spiegelrechnern abgerufen werden. Squid kann auf dem gewünschten System mit dem folgenden rpm–Befehl installiert werden:

```
rpm -ivh squid-2.3.STABLE4-10.i386.rpm
```

Squid−Konfiguration

Die Arbeitsweise und das Verhalten von Squid wird durch die Konfigurationseinzelheiten gesteuert, die in der Konfigurationsdatei festgelegt werden (squid.conf); diese Datei findet sich normalerweise im /etc/squid−Verzeichnis. Die Konfigurationsdatei squid.conf ist sehr umfangreich, das gute daran ist aber, das alle Optionen klar mit Erläuterungen aufgeführt sind.

Das erste, was editiert werden muss, ist http_port, welches die Socket− Adresse spezifiziert, auf der Squid auf Client−Anforderungen lauscht. Standardmäßig ist als Port 3128 gesetzt, aber dies kann auf einen benutzerdefinierten Wert gesetzt werden. Zusammen mit dem Port−Wert kann auch die IP−Adresse der Maschine angegeben werden, auf der Squid läuft. Dies kann z.B. geändert werden zu:

http_port 192.168.0.1:8080

Mit der obigen Deklaration wird Squid auf die IP−Adresse 192.168.0.1 und die Port−Adresse 8080 gebunden. Jede Port−Adresse kann angegeben werden, aber stellen Sie sicher, das keine andere Anwendung auf dem gesetzten Port läuft. Mit ähnlichen Konfigurationszeilen können auch die Ports anderer Service−Anforderungen gesetzt werden.

Zugangskontrollen

Durch die Zugangskontroll−Möglichkeiten kann der Internet−Zugang kontrolliert werden hinsichtlich bestimmter Zeiträume, der lokalen Zwischenspeicherung, zu bestimmten Rechnern oder Rechnergruppen usw. Die Squid−Zugangskontrollen haben zwei verschiedene Komponenten, nämlich ACL−Elemente und Zugangslisten. Eine Zugangsliste erlaubt oder versagt den Zugriff auf den Dienst.

Einige wichtige ACL−Elemente sind nachstehend aufgeführt

- src: Quelle d. h. Client−IP−Adressen
- dst: Ziel d. h. Server−IP−Adressen
- srcdomain: Quelle d. h. Client−Domain−Namen
- dstdomain: Ziel d. h. Server−Domain−Namen
- time: Zeit/Wochentag
- url_regex: URL−Überprüfung auf Übereinstimmung mit regulären Ausdrücken
- urlpath_regex: URL−Pfad−Überprüfung auf Übereinstimmung mit regulären Ausdrücken ohne Berücksichtigung von Protokoll und Hostname
- proxy_auth: Benutzer−Authentifizierung durch externe Prozesse
- maxconn: maximal erlaubte Anzahl von gleichzeitigen Verbindungen von einer einzelnen Client−IP−Adresse

Zur Anwendung der Kontrollen muss man zuerst eine Reihe von ACLs definieren und dann Regeln auf sie anwenden. Das Format einer ACL−Anweisung ist:

acl acl_element_name type_of_acl_element values_to_acl

Hinweise:

1. acl_element_name kann ein benutzerdefinierter Name für ein ACL− Element sein
2. Es dürfen keine ACL−Elemente den gleichen Namen haben.
3. Jedes ACL besteht aus einer Liste von Werten. Wenn eine Überprüfung stattfindet, werden mehrere Werte mit oder verknüpft. In anderen Worten, ein ACL−Element trifft zu, wenn irgendeins seiner

Werte übereinstimmt.
4. Nicht jedes ACL−Element kann mit jedem Typ von Zugangsliste benutzt werden.
5. Verschiedene ACL−Elemente werden in getrennten Zeilen angegeben und Squid fasst sie in einer Liste zusammen.

Es ist eine Anzahl unterschiedlicher Zugangslisten verfügbar. Diejenigen, die wir hier benutzen werden, sind nachstehend aufgeführt

- **http_access:** Erlaubt es HTTP−Clients, auf den HTTP−Port zuzugreifen. Dies ist die vorrangige Zugangsliste.
- **no_cache:** Definiert das Zwischenspeichern der Antworten auf Anforderungen

Eine Zugangslisten−Regel besteht aus Schlüsselwörtern wie allow oder deny, welche den Zugang zu einem speziellen ACL−Element oder einer Gruppe erlauben oder verweigern.
Hinweis:

1. Die Regeln werden in der Reihenfolge überprüft, in der sie in der Konfigurationsdatei erscheinen; sobald eine Übereinstimmung vorliegt, wird der Vergleich abgebrochen.
2. Eine Zugangsliste kann aus mehreren Regeln bestehen.
3. Wenn keine Regel zutrifft, ist die Fehlwert−Aktion das Gegenteil zur letzten Regel in der Liste, daher ist es ratsam, die Fehlwert−Aktion ausdrücklich aufzuführen.
4. Alle Elemente eines Zugangseintrages werden mit UND verbunden und in der folgenden Weise ausgeführt:
   http_access Aktion Anweisung1 UND Anweisung2 UND Anweisung ODER
   http_access Aktion Anweisung3
   Mehrere http_access−Anweisungen werden durch ODER verknüpft während die Elemente eines Zugangseintrages mit UND verbunden werden.
5. Erinnern Sie sich daran, das die Regeln immer von oben nach unten gelesen werden.

Zurück zur Konfiguration

Standardmäßig gewährt Squid keinem Client Zugriff und die Zugangslisten müssen für diesen Zweck angepasst werden. Man muss seine eigenen Regeln aufführen, um den Zugang zu erlauben. Blättern Sie in der squid.conf−Datei abwärts und fügen Sie die folgenden Zeilen direkt vor der "http_access deny all"−Zeile ein:

acl mynetwork 192.168.0.1/255.255.255.0
http_access allow mynetwork

Mynetwork ist der ACL−Name und die nächste Zeile die Regel, die auf einen bestimmten ACL−Eintrag anwendbar ist, z. B. mynetwork. 192.168.0.1 bezieht sich auf die Adresse des Netzwerks, dessen Netzmaske 255.255.255.0 ist. mynetwork gibt also einer Gruppe von Maschinen in dem Netzwerk einen Namen und die folgende Regel erlaubt den Clients den Zugang. Die vorstehenden Änderungen zusammen mit der http_port−Anweisung sind ausreichend, um Squid zu aktivieren. Nach den Änderungen kann Squid mit dem folgenden Befehl gestartet werden:

service squid start

Hinweis:
Squid kann auch automatisch zur Bootzeit gestartet werden, indem es in ntsysv oder setup (System Service Menu) aktiviert wird. Nach jeder Änderung in der Konfigurationsdatei muss der aktuelle Squid−Prozess

gestoppt und Squid erneut gestartet werden, damit die neuen Konfigurationsänderungen wirksam werden. Diese zwei Schritte können durch folgende Befehle erreicht werden:

1. service squid restart or
2. /etc/rc.d/init.d/squid restart

Konfiguration der Client−Maschinen

Da die Clientanforderungen auf einem besonderen Port des Proxy−Servers erwartet werden, müssen die Client−Maschinen für den gleichen Zweck konfiguriert werden. Es wird hier vorausgesetzt, dass diese Maschinen bereits mit dem LAN (mit gültigen IP−Adressen) verbunden sind und in der Lage sind, den Linux−Server mittels ping zu erreichen.
Internet Explorer

1. Tools −> Internet−Optionen
2. Wählen Sie den Verbindungs−Tab und LAN−Einstellungen
3. Markieren Sie Proxy−Server und geben Sie die IP−Adresse des Proxy−Servers und die Port−Adresse ein, unter der die Anforderungen angenommen werden (http_port−Adresse).

Netscape Navigator

1. Edit −> Einstellungen −> Advanced −> Proxies.
2. Wählen Sie Manual Proxy Configuration.
3. Wählen Sie den View Button &
4. Geben Sie die IP−Adresse des Proxy−Servers und die Port−Adresse ein, unter der die Anforderungen angenommen werden (http_port−Adresse).

Benutzung von Zugangslisten

Mehrere Zugangslisten und −Regeln bieten eine sehr gute und flexible Art, den Zugang von Clients zum Internet zu kontrollieren. Beispiele für die am häufigsten vorkommenden Steuerungen werden unten angegeben; diese sollten auf keinen Fall als die einzig vorhandenen Steuerungen angesehen werden.

1. Ausgewählten Maschinen den Zugang zum Internet erlauben

   ```
   acl allowed_clients src 192.168.0.10 192.168.0.20 192.168.0.30
   http_access allow allowed_clients
   http_access deny !allowed_clients
   ```

   Dies erlaubt nur den Rechnern mit den IP−Adressen 192.168.0.10, 192.168.0.20 und 192.168.0.30 Zugang zum Internet und dem Rest der IP−Adressen (nicht aufgeführt) wird der Service verweigert.
2. Beschränkung des Zugangs nur zu bestimmten Zeiten

   ```
   acl allowed_clients src 192.168.0.1/255.255.255.0
   acl regular_days time MTWHF 10:00−16:00
   http_access allow allowed_clients regular_days
   ```

http_access deny allowed_clients

Dies erlaubt allen Clients im Netzwerk 192.168.0.1 den Netzzugang von Montag bis Freitag zwischen 10.00 und 16.00 Uhr.

3. Unterschiedliche Zeitzugänge für verschiedene Clients

```
acl hosts1 src192.168.0.10
acl hosts2 src 192.168.0.20
acl hosts3 src 192.168.0.30
acl morning time 10:00−13:00
acl lunch time 13:30−14:30
acl evening time 15:00−18:00
http_access allow host1 morning
http_access allow host1 evening
http_access allow host2 lunch
http_access allow host3 evening
http_access deny all
```

Die obige Regel erlaubt Host1 Zugang sowohl während der morning−Zeit als auch in der evening−Zeit; während host2 und host3 entsprechend jeweils nur während der lunch−Zeit bzw. der evening−Zeit Zugriff haben.

Hinweis:
Alle Elemente eines Zugangseintrages werden mittels UND verbunden und in der folgenden Art ausgeführt

http_access Aktion Anweisung1 UND Anweisungt2 UND Anweisung ODER.

Mehrere http_access−Anweisungen werden mit ODER verknüpft, während Elemente eines Zugangseintrages mit UND verbunden werden; aus diesem Grunde wird

http_access allow host1 morning evening

niemals zutreffen, weil die Zeitdefinitionen morning und evening (morning UND evening) nie übereinstimmen und daher keine Aktion stattfindet.

4. Sperrung von Webadressen
Squid kann den Zugang zu einem bestimmten Rechner oder zu Seiten, die besondere Worte enthalten, sperren. Dies kann in der folgenden Weise implementiert werden:

```
acl allowed_clients src 192.168.0.1/255.255.255.0
acl banned_sites url_regex abc.com *()(*.com
http_access deny banned_sites
http_access allow allowed_machines
```

Die gleichen Definitionen können genutzt werden, um den Zugang zu Seiten zu sperren, die bestimmte Worte enthalten wie z.B. dummy , fake

```
acl allowed_clients src 192.168.0.1/255.255.255.0
acl banned_sites url_regex dummy fake
http_access deny banned_sites
http_access allow allowed_clients
```

Es ist nicht besonders praktikabel, alle Wortlisten oder Seiten aufzuführen, zu denen der Zugang verhindert werden soll, diese können in eine Datei ausgelagert werden (z. B. Sperr.Liste im /etc−Verzeichnis) und die ACL−Definition kann diese Information aus dieser Datei einlesen und dann den Zugang zu den gesperrten Seiten verhindern.

```
acl allowed_clients src 192.168.0.1/255.255.255.0
acl banned_sites url_regex "/etc/Sperr.Liste"
http_access deny banned_sites
http_access allow allowed_clients
```

5. Optimieren der Benutzung
   Squid kann die Anzahl gleichzeitiger Verbindungen von den Client−Maschinen durch das maxconn−Element kontrollieren. Zur Benutzung dieser Option sollte die client_db−Eigenschaft vorher aktiviert werden.

   ```
   acl mynetwork 192.168.0.1/255.255.255.0
   acl numconn maxconn 5
   http_access deny mynetwork numconn
   ```

   Hinweis:
   Das maxconn−ACL benutzt einen kleiner−als−Vergleich. Dieser Zugangseintrag trifft zu, wenn die Anzahl gleichzeitiger Verbindungen größer als der angegebene Wert ist. Dies ist der Hauptgrund dafür, weshalb dieser Zugangseintrag nicht zusammen mit der http_access allow Regel benutzt wird.
6. Zwischenspeichern der Daten
   Antworten auf die Anforderungen werden direkt gespeichert, dies ist sinnvoll für statische Seiten. Es macht aber keinen Sinn, cgi−bin− oder Servlet−Dateien zwischenzuspeichern. Dies kann über das no_cache ACL−Element verhindert werden.

   ```
   acl cache_prevent1 url_regex cgi−bin /?
   acl cache_prevent2 url_regex Servlet
   no_cache deny cache_prevent1
   no_cache deny cache_prevent2
   ```

7. Erstellen Ihrer eigenen Fehlermeldungen
   Es ist möglich, mit einer deny−Regel eigene Fehlermeldungen zu verwenden; dies geschieht mittels der deny_info−Option. Alle Squid−Fehlermeldungen finden sich standardmäßig im Verzeichnis /etc/squid/errors. Das Fehler−Verzeichnis kann über die error_directory−Option gesetzt werden. Sie können sogar die vorhandenen Fehlermeldungen anpassen.

   ```
   acl allowed_clients src 192.168.0.1/255.255.255.0
   acl banned_sites url_regex abc.com *()(*.com
   http_access deny banned_sites
   deny_info ERR_BANNED_SITE banned_sites
   http_access allow allowed_clients
   ```

   Im obigen Beispiel wird eine besondere Nachricht angezeigt, wann immer Benutzer/innen versuchen, auf eine Seite mit den aufgeführten gesperrten Worten zuzugreifen. Der Dateiname in der Option, z. B. .ERR_BANNED_SITE muss in den angegebenen Fehlerverzeichnis existieren. Diese Fehlerdatei sollte im HTML−Format erstellt sein. Die oben angeführten Beispiele sind nur einige wenige der Optionen, Möglichkeiten und Fähigkeiten der ACL−Elemente. Sie sollten sich die FAQ section auf der Squid−Webseite für ausführlichere Benutzung und Erklärungen zu anderen ACL−Elementen und Zugriffseinträgen durchlesen.

Log−Dateien

Alle Squid−Logdateien sind im Verzeichnis /var/log/squid enthalten; dies sind cache.log, access.log und store.log. Die Datei access.log enthält die Information über die Client−Anforderungen, Aktivität und enthält Einträge für jede HTTP− & ICP−Anfrage, die an den Proxy−Server gestellt wurde, die Client−IP−Adresse, die Anforderungsart, angeforderte URL, usw. Der Inhalt dieser Datei kann zur Analyse der Zugriffsinformationen genutzt werden. Viele Programme wie sarg, calamaris, Squid−Log−Analyzer sind verfügbar, um diese Daten zu analysieren und Berichte (im HTML−Format) zu generieren. Diese Berichte können hinsichtlich Benutzern, IP−Adressen, besuchten Seiten, usw. erstellt werden.

Die Speicherung der Log−Dateien kann über die folgenden Optionen geändert werden:

```
cache_access_log     für access.log
cache_log            für cache.log
cache_store_log      für store.log (Store manager)
pid_filename         Squid Prozess-ID-Dateiname
```

Authentifizierungs−Methoden

Squid erlaubt in der Standard−Konfiguration jedermann und −frau den Zugang ohne einen Authentifizierungsprozess. Um die Benutzer zu überprüfen (d. h. nur zulässige Benutzerinnen von jeder Maschine im Netzwerk), bietet Squid die Möglichkeit zur Authentifizierung, jedoch über ein externes Programm, für das dann ein gültiger Benutzername und Passwort erforderlich sind. Dies wird ermöglicht durch die Benutzung des proxy_auth−ACL und der authenticate_program−Anweisung, welches die Benutzer zwingt, Benutzername und Passwort zu verifizieren, bevor Zugang gewährt wird. Es sind mehrere Authentifzierungsprogramme vorhanden, die Squid benutzen kann; diese sind

1. LDAP : benutzt das Linux Lightweight Directory Access Protocol
2. NCSA : benutzt NCSA−ähnliche Benutzer− und Passwort−Dateien
3. SMB : benutzt SMB−Server wie SAMBA oder Windows NT
4. MSNT : benutzt Windows NT Domain−Authentifizierung
5. PAM : benutzt Linux Pluggable Authentication Modules
6. getpwam : benutzt Linux Passwort−Datei.

Man muss das benutzte Authentifizierungsprogramm angeben und dies geschieht durch die authenticate_program−Option. Stellen Sie sicher, dass das benutzte Authentifizierungsprogramm installiert ist und funktioniert.

Die Änderungen in der squid.conf−Datei sollten nun auch auf das gleiche authenticate_program /usr/local/bin/pam_auth verweisen:

acl pass proxy_auth REQUIRED
acl mynetwork src 192.168.0.1/255.255.255.0
http_access deny !mynetwork
http_access allow pass
http_access deny all

Hiermit wird das PAM−Authentifizierungsprogramm benutzt und alle Benutzer müssen sich authentifizieren, bevor sie auf das Internet zugreifen.

Optionen wie authenticate_ttl und authenticate_ip_ttl können benutzt werden, um das Verhalten des Authentifizierungsprozesses zu ändern, z.B. um eine Revalidierung von Benutzername und Passwort zu erzwingen.

Referenzen

Dieser Artikel berührt gerade nur die Spitze des Squid−Eisberges, für weitere Informationen sollten Sie die folgenden Web−Seiten aufsuchen:

- Squid−Webseite www.squid−cache.org
- Squid Dokumentations−Projekt, squid−docs.sourceforge.net
- visolve.com
- Zur Proxy Authentifizierung, home.iae.nl/users/devet/squid/proxy_auth

---

2005−01−11, generated by lfparser_pdf version 2.51