

22 Proxy-Server

22 Proxy-Server.....	1
22.1 Auswahl der wichtigsten Konfigurationsparameter.....	2
22.2 Zugriffssteuerung mit Access-Listen.....	3
22.2.1 Definieren der Regeln.....	3
22.2.2 Anwenden der Regeln.....	4
22.3 Einrichten der Clients.....	8
22.4 Einrichten eines eigenen Proxyservers.....	9
22.5 SquidGuard.....	9
22.6 Inhaltsfilterung.....	10
22.7 Aufgaben.....	11

Das Programm Squid ist der am weitesten verbreitete Proxy-Cache für Linux/Unix-Plattformen. Ein Proxy-Cache ist ein Zwischenspeicher für Internetanfragen, das heißt die Kopie einer Seite, die heruntergeladen wurde,



wird auf der Festplatte abgelegt. Wenn dann ein anderer Nutzer die gleiche Seite aufruft, wird er direkt von der Festplatte bedient. Der Zugriff erfolgt damit schneller. Dynamisch generierte CGI-Seiten werden nicht im Cache gehalten.

Zusätzlich bietet Squid die Möglichkeit, Zugriffsregeln zu definieren. Dadurch können Zugriffe auf bestimmte Webseiten verweigert werden. Weiterhin kann mithilfe von Analyseprogrammen das Surfverhalten der Nutzer ausgewertet werden. Wenn man Squid mit einer Firewall kombiniert, wird das interne Netz geschützt und der Internetzugang ist vom internen Netz nur über Squid möglich.

Bei Verwendung einer SuSE-Distribution ist Squid bereits vorkonfiguriert, so dass nur Detail-Anpassungen erforderlich sind. Bevor Squid gestartet wird, sollte das Netzwerk fertig konfiguriert sein (Internetverbindung besteht, Squid startet nur, wenn mindestens ein Nameserver erreichbar ist). Der Start von Squid erfolgt über die Konsole mittels `rcsquid start (als root)`. Wenn Squid bei jedem Systemstart mitgestartet werden soll, muss mittels Runlevel-Editor ein entsprechender Eintrag vorgenommen werden.

Sofern in einem Netzwerk mehrere Proxy-Server installiert sind, können diese untereinander verknüpft werden.

Zum Absichern eines Netzwerkes wird Squid zusammen mit einem Firewall verwendet. Dabei wird der Internetzugang ausschließlich über den Proxyserver zugelassen.

Squid wird in der Datei `/etc/squid/squid.conf` konfiguriert. Genau wie der FTP-Server, vergleiche Abschnitt , wird auch Squid über den Xinetd gestartet. Die Zugriffssteuerung erfolgt allerdings in der Datei `/etc/squid/squid.conf`.

Bei der Installation von *Squid* über *YaST* ist der Proxyserver bereits vorkonfiguriert. `/etc/squid/squid.conf` ist sehr gut dokumentiert und weitgehend selbsterklärend. Der Zugriff von externen Clients ist der Vorkonfiguration allerdings gesperrt: Aus Sicherheitsgründen ist nur ein Zugriff von *localhost* zugelassen!

Der Zugriff auf das Internet wird protokolliert. Die Protokolldateien können zum Beispiel mit dem Programm *Calamaris* ausgewertet werden, <http://Calamaris.Cord.de>.

22.1 Auswahl der wichtigsten Konfigurationsparameter

<code>http_port 3128</code>	Port, auf dem Squid auf die Client-Anfragen lauscht
<code>acl <acl_name> <type> <data></code>	Einstellung zur Zugriffskontrolle. Die Regeln werden nacheinander abgearbeitet. Der Name <code><acl_name></code> ist frei wählbar. Für <code><type></code> kann eine Reihe verschiedener Parameter ausgewählt werden (siehe Abschnitt <i>access controls</i> in <code>/etc/squid/squid.conf</code>). <code><data></code> kann ein Rechnername, eine IP, eine URL oder auch eine Datei mit mehreren Rechnernamen sein. Darüber hinaus kann auch der Internetzugang zu bestimmten Zeiten gesperrt werden.
<code>http_access allow <acl_name></code>	Hier steht, wer auf das Internet zugreifen darf. Die <code>acl</code> muss zuvor definiert worden sein.
<code>redirect_programm /usr/bin/squidGuard</code>	Mit dieser Option wird ein „Redirector“ eingerichtet, zum Beispiel <i>SquidGuard</i> . Mit <i>SquidGuard</i> kann der Internetzugriff für verschiedene Gruppen differenziert gesteuert werden.
<code>ident_lookup_access allow <acl_name></code>	Eine Ident-Anfrage wird bei den durch die <code>acl</code> definierten Clients durchgeführt. Dadurch wird erreicht, dass die Identität der jeweiligen Benutzer ermittelt wird (und in den log-Dateien protokolliert wird). Auf den Clients muss ein Ident-Dämon laufen (<i>pidentd</i> für Linux, <i>ident.exe</i> für Windows, http://identd.sourceforge.net). Damit nur Clients mit erfolgreicher Identitätsanfrage zugelassen werden, muss eine weitere <code>acl</code> definiert werden: <pre>acl identhosts ident REQUIRED http_access allow identhosts</pre>

Mehr zu diesem Thema: <http://www.squid-handbuch.de>.
<http://www.squidguard.org>

22.2 Zugriffssteuerung mit Access-Listen

Die Zugriffssteuerung erfolgt immer in zwei Schritten: Zunächst müssen Regeln definiert werden. Im zweiten Schritte müssen die Regeln zu einer Erlaubnis oder einem Verbot führen.

22.2.1 Definieren der Regeln

Die Zugriffssteuerung durch den Proxyserver erfolgt mit so genannten Access-Listen (*acl*, *access control list*). Diese definieren jedoch nicht eine Erlaubnis oder ein Verbot, sondern einfach nur eine Regel.

Der Aufbau einer *acl* hat immer die gleiche Struktur:

```
acl aclname acltype string1 [string2] ...
```

oder

```
acl aclname acltype "filename"
```

Nach *acl* folgt der (frei wählbare) Name der ACL. *acltyp* gibt an, um welche Sorte Regel es sich handelt. Danach folgt eine Auflistung der Parameter, die jeweils durch Leerzeichen getrennt sind.

Für die jeweiligen Anwendungen steht eine Vielzahl verschiedener Regeln zur Verfügung, von denen hier nur eine kleine Auswahl beschrieben werden soll (vollständige Liste siehe: www.squid-handbuch.de):

src beschreibt die IP-Adresse/Netzmaske von Clients (Quelle der Anfrage).

Beispiele:

```
acl TEST1 src 192.168.10.25/255.255.255.255
acl TEST2 src 192.168.1.0/24
acl TEST3 src 192.168.5.10-192.168.5.20/255.255.255.0
```

dst beschreibt Server-IP-Adressen (Ziele der Anfrage). Syntax siehe *src*.

time beschreibt ein Zeitfenster mit Wochentag und Uhrzeit, wobei der englisch abgekürzte Wochentag zu verwenden ist (siehe

/etc/squid.conf). *h1:m1* und *h2:m2* sind Angaben in *Stunden:Minuten*, wobei *h1:m1* < *h2:m2* sein muss.

Beispiele:

```
acl TEST1 time 08:00-16:00 # jeden Tag 8-16 Uhr
acl TEST2 time S 00:00-23:59 # jeden Sonntag 0-24 Uhr
```

proto beschreibt Protokolltypen einer Verbindung. Hiermit kann nach den von Squid unterstützten Protokollen gefiltert werden.

Beispiel:

```
acl TEST proto FTP
```

url_regex [-i] beschreibt reguläre Ausdrücke in einer *url*. Die Option *-i* sorgt dabei für eine Ignorierung der Groß- und Kleinschreibung. Mehr zu regulären Ausdrücken siehe Abschnitt .

Beispiele:

```
acl sperre url_regex "/etc/njet" # alle urls, die Wörter aus
                               der Datei /etc/njet enthalten
acl killer url_regex -i Horst*  # alle Seiten, die Horst
                               enthalten, also www.Horst.de oder
                               www.Horstilein.com
```

22.2.2 Anwenden der Regeln

Im zweiten Schritt müssen die Regeln angewendet werden. Die Syntax der Anwendungen hat immer den gleichen Aufbau:

```
regeltyp allow|deny [!]acl1[![!]acl2] ...
```

regeltyp beschreibt die Art der Regel. Darauf folgt ein *allow* oder *deny*, je nachdem ob die Regel zu einer Erlaubnis oder einem Verbot führen soll. Abschließend steht noch der Name der Regel. Falls mehrere Regeln angegeben werden, wird der Regeltyp nur dann erlaubt/verboten, wenn **alle** Regeln zutreffen.

Im folgenden Beispiel wird eine Regel mit dem Namen TEST von Typ *src* (Quell-IP-Adressen) mit dem Inhalt 192.168.11.3 definiert.

```
acl TEST src 192.168.11.3/32
http_access allow TEST
```

In der zweiten Zeile bestimmt der Regeltyp *http_access*, dass eine http-Anfrage erlaubt ist (*allow*), wenn die *ACL TEST* zutrifft. Konkret: Die http-Anfrage ist erlaubt, wenn der Client die IP-Adresse 192.168.11.3 und die entsprechende Subnetzmaske hat.

Eine ACL kann mit einem vorangestellten "!" umgekehrt werden (! = nicht). Bezogen auf unser Beispiel bedeutet die Regel

```
http_access allow !TEST
```

dass die http-Anfrage erlaubt ist, wenn die ACL TEST nicht zutrifft, d.h. der Client nicht die IP-Adresse 192.168.11.3 hat. Daraus ergibt sich auch, dass

```
http_access deny !TEST
```

und

```
http_access allow TEST
```

grundsätzlich die gleiche Aussage haben, in der Zusammensetzung mit weiteren Regeln jedoch unterschiedliche Auswirkungen haben können.

Eine Auswahl wichtiger Regeltypen:

http_access erlaubt oder verbietet Clients den Zugriff auf den http Port von Squid. Der Internetzugang ist mit einem Browser über diesen Proxy erlaubt oder verboten. Wenn keine *http_access* Zeile existiert, ist der Zugriff grundsätzlich verboten.

Wenn keine der angegebenen Zeilen zutrifft, ist der Standardwert das Gegenteil der letzten *http_access=* Zeile. Regelt die letzte Zeile ein *deny*, ist der Zugriff für alle nicht zutreffenden Anfragen erlaubt. Regelt die letzte Zeile ein *allow*, ist der Zugriff für alle nicht zutreffenden Anfragen verboten.

http_reply_access erlaubt oder verbietet Squid eine Antwort (*reply*) an den Client zu senden. Diese Regel ergänzt die *http_access*-Regel. Wenn keine *http_reply_access*-Zeile existiert, ist das Versenden von

Antworten grundsätzlich erlaubt. Ansonsten gelten die gleichen Regeln wie bei *http_access*.

Access-Regeln können eine oder mehrere ACLs enthalten. Sind mehr als eine ACL enthalten, werden die ACLs UND-verknüpft. Eine Regel trifft also nur zu, wenn jede enthaltene ACL zutrifft.

Für jeden einzelnen Request werden die Regeln in der Reihenfolge, in der sie in der Konfigurationsdatei stehen, abgearbeitet. D.h. die Konfigurationsdatei wird zeilenweise von oben nach unten durchlaufen. Es wird dabei jede durchlaufene Regel geprüft, so lange bis die erste Regel erfüllt ist. Ist eine Regel erfüllt, wird an dieser Regel die weitere Regelbearbeitung abgebrochen. Also: alle folgenden Regeln - egal ob sie zutreffen oder nicht - werden nicht mehr berücksichtigt!

Um eine sinnvolle Reihenfolge in den Access-Regeln beizubehalten und zur besseren Übersichtlichkeit, ist in der Standardkonfiguration ein bestimmter Bereich vorgesehen, in dem eigene Access-Regeln definiert werden sollten. Dieser Bereich ist überschrieben mit der Zeile:

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

Nach dieser Zeile können Sie ihre eigenen Regeln einfügen. Damit ist sichergestellt, dass die Standardregeln des Systems unverändert bleiben.

Folgendes Beispiel zeigt, wie eine abweichende Regelschreibweise das Verhalten des Proxy-Servers beeinflusst. Dabei sind drei Regeln zu beachten:

- Innerhalb einer ACL gilt ODER-Verknüpfung. D.h. mindestens ein Element der ACL muss zutreffen.
- Innerhalb von Access-Regeln gilt UND-Verknüpfung. D.h. alle angegebenen ACLs müssen zutreffen.
- Access-Regeln werden von oben nach unten abgearbeitet, bis die erste Regel zutrifft. Alle auf einen Treffer folgenden Regeln gleichen Typs werden ignoriert.

Ein einfaches Beispiel mit drei Regelwerken, die auf den ersten Blick scheinbar das Gleiche tun:

Die ACLs:

```
acl ALL src 0.0.0.0/0.0.0.0
acl USER1 ident Max
```

```
acl USER2 ident Moritz
acl USER3 ident Max Moritz
```

Es wird eine ACL mit dem Namen *ALL* definiert, die alle Quell-IP-Adressen enthält (0.0.0.0/0.0.0.0 = alle IP-Adressen, aller Netze). Es gibt weiter eine ACL *USER1*, die den User *Max* enthält, eine weitere (*USER2*) mit dem User *Moritz* und eine letzte ACL (*USER3*), die die User *Max* und *Moritz* enthält.

Regelwerk 1:

```
http_access allow USER1
http_access allow USER2
http_access deny ALL
```

- 1) Wenn die ACL *USER1* erfüllt ist (User = *Max*), wird der Zugriff erlaubt. Wenn nicht, geht's weiter:
 - 2) Wenn die ACL *USER2* erfüllt ist (User = *Moritz*), wird der Zugriff erlaubt. Wenn nicht, geht's weiter:
 - 3) Wenn keine vorherige Regel erfüllt ist, wird allen Clients der Zugriff verweigert.
- Es wird also *Max* und *Moritz* der Zugriff erlaubt, allen anderen verboten.

Regelwerk 2:

```
http_access allow USER1 USER2
http_access deny ALL
```

- 1) Wenn die ACL *USER1* **und** *USER2* erfüllt sind, wird der Zugriff erlaubt. Wenn nicht, geht es weiter:
 - 2) Wenn keine vorherige Regel erfüllt ist, wird allen Clients der Zugriff verweigert.
- In diesem Beispiel müssen in Zeile 1 **beide** ACLs erfüllt sein (UND-Verknüpfung). Da beide vom gleichen Typ sind, jedoch mit unterschiedlichem User, kann diese Bedingung nie erfüllt werden. Der User für eine Anfrage kann nicht gleichzeitig *Max* und *Moritz* heißen. Es wird also **keinem** Client der Zugriff erlaubt.

Regelwerk 3:

```
http_access allow USER3
http_access deny ALL
```

1) Wenn die ACL *USER3* erfüllt ist (der User ist in der Liste "*Max Moritz*" enthalten), wird der Zugriff erlaubt. Wenn nicht, geht's weiter:

2) Wenn keine vorherige Regel erfüllt ist, wird allen Clients der Zugriff verweigert.

In der ACL *USER3* sind die User *Max* und *Moritz* definiert. Innerhalb einer ACL gilt eine ODER-Verknüpfung. Die ACL trifft demnach zu, wenn mindestens eines der darin enthaltenen Elemente zutrifft.

Der Zugriff wird also erlaubt wenn der User *Max* oder *Moritz* heißt, allen anderen wird der Zugriff verboten. Die Regel ist identisch mit Regelwerk 1.

Beispiel für eine sinnvolle Anwendung der UND-Regel:

```
acl ALL src 0.0.0.0/0.0.0.0
acl USER ident Max Moritz
acl SERVER dstdomain .squid-cache.org
http_access allow USER SERVER
http_access deny ALL
```

Das oben gezeigte Regelwerk definiert eine ACL *USER*, in der die Benutzer *Max* und *Moritz* enthalten sind und eine ACL *SERVER* für die Zieldomain **.squid-cache.org*.

In der nächsten Zeile wird der Zugriff erlaubt, wenn die ACLs *USER* und *SERVER* zutreffen, das bedeutet wenn der Benutzer in der ACL *USER* enthalten ist und das Ziel der Anfrage in der ACL *SERVER* enthalten ist. Sind nicht beide Bedingungen erfüllt, greift die letzte Regel, die allen Clients den Zugriff verweigert.

Es wird also der Zugriff erlaubt, wenn *Max* oder *Moritz* auf einen Server in der Domain *squid-cache.org* zugreifen. Anfragen anderer Benutzer oder zu anderen Ziel-Domains werden verboten.

Ausführlich dokumentierte Anwendungsbeispiele siehe www.squid-handbuch.de.

22.3 Einrichten der Clients

Die Einrichtung der Clients erfolgt über die Konfiguration der Browser. Im Konfigurationsmenü wird der Internetzugriff über einen Proxy aktiviert. Der Name beziehungsweise die Adresse des Servers muss in das entsprechende Feld eingetragen werden. Der Zugriff erfolgt standardmäßig über den Port 3128.

22.4 Einrichten eines eigenen Proxyservers

- 1) Auf dem eigenen Computer wird ein Webserver (Apache) eingerichtet und der Proxyserver Squid installiert.
- 2) Squid muss jetzt gestartet werden. Entweder über Konsole: `rcsquid start` oder automatisch bei jedem Neustart mit dem *Runlevel*-Editor in *YaST*.
- 3) Soll über den eigenen Browser auf den Webserver zugegriffen werden, so muss im Konqueror unter *Einstellungen/Proxyserver* „Proxyserver verwenden“ angeklickt und unter *Benutzerdefinierte Einstellungen* die eigene IP eingetragen werden: „`http://localhost`“.
- 4) Wenn über den Browser eines anderen Computers (im gleichen Subnetz oder über Router) auf den Webserver zugegriffen werden soll, so muss das in der Squid-Konfiguration erst einmal erlaubt werden (Squid verbietet aus Sicherheitsgründen in der Basiskonfiguration fast alles). Hierzu muss die Konfiguration `/etc/squid/squid.conf` im Abschnitt „*Access Control*“ der betreffende Computer beziehungsweise das Subnetz zugelassen werden.

22.5 SquidGuard

SquidGuard ist ein *redirector*, das heißt, Web-Anfragen werden vom Proxyserver an den *redirector* übergeben, dieser prüft die Anfrage und gibt sie dann entweder zurück oder leitet sie auf eine andere Adresse um, so dass zum Beispiel eine Fehlermeldung ausgegeben wird. Die Überprüfung der Webanfragen kann auch mit Squid durchgeführt werden. Ein *redirector* ist jedoch speziell für diese Aufgabe optimiert und bearbeitet diese dadurch schneller.



Um mittels Squidguard jugendgefährdende Inhalte zu sperren, stehen im Internet umfangreiche Sammlungen von Adressen und Schlüsselwörtern, die gesperrt werden sollten, zur Verfügung¹. Dabei werden die Rubriken „*Gewalt*“,

¹ Alternativ wird Filter-Software angeboten: zum Beispiel *Cobion*. Aus Newsgroups heißt es allerdings, dass *Cobion* (für immerhin 399 Euro/Jahr, aber kostenlose 30-Tage-Trial-Version)

„Internethandel“, „Pornographisch“, „Drogen“, „Glücksspiel“ und andere unterschieden.

Quellen für Blacklists:

- englisch: - DansGuardian: <http://dansguardian.org/>
 - ftp.teledanmark.no (?)
- deutsch: - Bürgernetz Pfaffenhofen: http://www.bn-paf.de/filter/index_de.html
 - Medienzentrum Oberberg: www.medienzentrum-oberberg.de

22.6 Inhaltsfilterung

Inhalts- oder auch Contentfilter sollen Internetseiten, insbesondere solche mit bedenklichem oder verbotenen Inhalt, für ein Subnetz unzugänglich machen. Die technischen Verfahren zur Lösung dieser Aufgabe reagieren bereits beim Abrufen dieser Seiten (Adressfilter) oder erst kurz vor dem Aufbau der Seiten. Wieder andere Verfahren sperren bestimmte Protokolle, um zum Beispiel illegalen Dateidownload zu verhindern. Neuere Verfahren filtern die Inhalte der Seiten und sind damit deutlich wirkungsvoller aber auch rechenintensiver [Ric03].

Adressfilter sind wartungsintensiv, da die Liste der zu sperrenden Seiten (Blacklists) ständig angepasst werden muss. Ergänzend müssen Ausnahmen (Whitelists) definiert werden, die dann in Kombination mit den Blacklists eine sinnvolle Filterung erreichen. Das Verfahren läuft der Erzeugung neuer Seiten also ständig hinterher. Der Vorteil dieses Verfahrens liegt in der geringen Anforderung an die Rechenleistung und auch in den geringen Kosten. Geeignete Blacklists können auch kostenlos bezogen werden, siehe Abschnitt 22.5.

Inhaltsfilter durchsuchen die Seiten vor dem Bildschirmaufbau nach verbotenen Worten. Einige Systeme untersuchen auch eingebundene Bilder nach bestimmten Mustern. Dabei ist die Auffassung, was denn eigentlich bedenklich ist, sehr stark vom Herkunftsland der Systeme abhängig. So wird zum Beispiel bei Systemen aus Amerika bereits ein wenig nackte Haut als pornographisch eingestuft, während rechtradikale und menschenverachtende Inhalte oftmals zugelassen werden [Ric03].

weniger effektiv filtert, als Squidguard.

Das Hauptproblem der Inhaltsfilterung liegt in den gesteigerten Anforderungen an die eingesetzte Hardware, um den Internetzugang nicht übermäßig zu behindern.

Wer den Internetzugang eines Netzwerkes kontrollieren soll, muss sich mit der Vielzahl der vorhandenen Soft- und Hardware-Lösungen auseinandersetzen. Einen ersten Überblick geben [Ric03] oder <http://www.lehrer-online.de>. Neben den technischen Systemen zur Filterung der aufgerufenen Seiten gibt es auch eine Initiative, die Webautoren die Kennzeichnung ihrer Seiten nachlegt (<http://www.icra.org/de>). Anhand dieses Labels werden dann jugendgefährdende Inhalte erkannt. Das System basiert allerdings auf der freiwilligen und auch korrekten Selbsteinschätzung der Webautoren.

22.7 Aufgaben

- 1) Verändere die Konfiguration des Proxyservers, so dass ein Zugriff auf den Webserver nur zu einer bestimmten Zeit möglich ist.
- 2) Ändere die Konfiguration des Proxyservers ab, damit ein Zugriff auf html-Dokumente, deren Name die Zeichenkette „*auto*“ enthält gesperrt wird. Wird damit auch der Zugriff auf Dokumente gesperrt, deren Name die Zeichenkette „*Auto*“ enthält, gesperrt?
- 3) Richte Squid so ein, dass nur dein Nachbar in einer bestimmten Zeitspanne auf deinen Webserver zugreifen darf und sonst niemand.
- 4) Erstelle ein shell-Skript, das die Datei `/var/log/squid/squid.log` auswertet und ermittelt, welche 10 Seiten in den letzten 7 Tagen am häufigsten aufgerufen wurden (*top ten*).