

# **Squid HTTP cache proxy**

**Kris Lowet**  
**30 maart 2009**

[www.krislowet.be](http://www.krislowet.be)  
[www.linuxontdekt.be](http://www.linuxontdekt.be)

## Inleiding

Snelheid. Daar draait het allemaal om wanneer er gebruikt gemaakt wordt van een cache proxy. Alle pakketjes moeten vlug bij de gebruiker zijn. Ook komen we uit een tijd waarin we te maken hadden met harde datalimieten die de ISP's hun klanten oplegde – en die tijd is trouwens nog niet volledig voorbij!

Laat ons eerlijk zijn, het is ook onnodig om x aantal keer diezelfde foto van het internet te gaan downloaden. Stel: 20 collega's krijgen op kantoor dezelfde mail aan met daarin een link naar een online fotoalbum met 50 foto's van de barbecue van vorige maand. Als iedere foto 1MB bedraagt en alle collega's alle 50 foto's bekijken, dan zou er op het einde van de rit maar liefst 1000MB aan data gedownload zijn. Dit is eenvoudigweg absurd.

Wanneer het kantoor daarentegen gebruik maakt van een cache proxy, dan worden die 50 foto's slechts 1 keer gedownload. Dit wilt zeggen: in plaats van 1000MB wordt er dan 50MB gedownload. De collega's die diezelfde foto's willen bekijken krijgen deze geserveerd van de cache proxy die lokaal staat opgesteld, zonder dat zij hier iets van merken, dit proces gebeurt transparant.

Een groot verschil voor wie de factuur van de ISP moet betalen, de foto's worden vlugger getoond aan de gebruikers, de internetverbinding geraakt niet overbelast met enkel en alleen foto's en er treedt geen netwerkvervuiling op.

Het nut van een cache proxy lijkt me nu wel duidelijk. Op volgende pagina's ga ik bespreken hoe je zo een cache proxy kan opzetten.

Het voorbeeld wat ik zal gebruiken, wordt ook ingezet op Ubuntu release party's.

## **Aanbevolen hardware**

Zeker niet onbelangrijk is om even stil te staan bij welke hardware je gaat gebruiken voor een cache proxy.

### **Harde schijf**

Tegenwoordig is de opslagruimte van een harde schijf al niet meer zo een probleem. Harde schijven van 500, 700 tot 1000GB worden ons naar het hoofd gegooid. Een schijf van 160GB moet echter zeker voldoen. Wie redundatie wilt, gaat uiteraard voor twee schijven in RAID 1.

Een tweede en zeer belangrijk aandachtspunt bij de keuze van harde schijf is de snelheid. Een SATA 7200RPM schijf voldoet. Als je de kans hebt, kies dan voor een 10K of 15K RPM model, zoveel te vlugger de schijf, zoveel te vlugger de data bij de gebruiker is. Beschik je over voldoende financiële middelen, kies dan voor de nieuwste Solid State Disk's. Voel je vrij om te kiezen voor een performante RAID opstelling!

### **Processor**

Squid heeft geen al te sterke CPU nodig. Een degelijke dual core volstaat.

### **RAM geheugen**

RAM geheugen is iets wat dezer dagen zeer goedkoop is. Om de hoeveelheid RAM voor een cache proxy te berekenen, telt men 10MB per voorziene GB cache opslag ruimte + de cache\_mem optie in Squid (standaard 8MB) + 20MB. Doe deze som x 2.

### **Netwerkaart**

Om je cache server met het interne netwerk te verbinden kies je best voor een 1Gb netwerkaart. Ten opzichte van een 100Mbit kaart is de 1Gb veel sneller, maar toch niet extreem veel duurder. Let uiteraard op welke switch je vervolgens gebruikt! De snelheid zal alleen maar geapprecieerd worden door je gebruikers!

## **Gebruikte hardware**

Ikzelf heb deze hardware gebruikt:

Dell R300

2 x 160 GB in RAID 0

2 x 1 GB DDRII RAM

Intel Xeon X3363 @ 2.83GHz

# Installatie

## Operating system

Ik heb gekozen voor het OS Debian (5.0) en heb het volgend partitieschema gebruikt:

- 200 MB /boot
- 50 GB /cache – noatime, notail, noexec
- 4 GB SWAP
- 1 GB /var/log
- 1 GB /tmp – noatime, nosuid, noexec
- rest GB /

/boot en / beschikken over een EXT3 bestandssysteem. /cache, de partitie die gebruikt zal worden voor de opslag van de cache bestanden, beschikt over een ReiserFS bestandssysteem zodat deze kleine bestandjes vlugger kunnen aangesproken worden. Ook heb ik voor de /cache partitie gekozen voor 2 opties: noatime (toegangstijd wordt niet opgeslagen), notail en noexec (het uitvoeren van bestanden tegengaan).

## Squid

Squid laat zich op Debian eenvoudig installeren door volgend commando:

```
apt-get install squid
```

# Configuratie Squid

Het configuratie bestand van Squid is te vinden op: /etc/squid3/squid.conf

```
acl manager proto cache_object
acl localhost src 127.0.0.1/32
acl to_localhost dst 127.0.0.0/8
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
http_access allow manager localhost
http_access deny manager
http_access deny !Safe_ports
http_access deny CONNECT !SSL_ports
acl partyOnder src 192.168.2.0/24
acl partyBoven src 192.168.3.0/24
acl partyOnderWlan src 192.168.4.0/24
acl partyBovenWlan src 192.168.5.0/24
http_access allow partyOnder
http_access allow partyBoven
http_access allow partyOnderWlan
http_access allow partyBovenWlan
http_access allow localhost
http_access deny all
icp_access deny all
htcp_access deny all
http_port 3128 transparent
hierarchy_stoplist cgi-bin ?
cache_mem 256 MB
maximum_object_size_in_memory 1 MB
cache_dir ufs /cache/squid 50000 16 256
maximum_object_size 50 MB
access_log /var/log/squid3/access.log squid
acl QUERY urlpath_regex cgi-bin \?
cache deny QUERY
refresh_pattern ^ftp:          1440 20% 10080
refresh_pattern ^gopher:      1440 0% 1440
refresh_pattern .              0 20% 4320
icp_port 3130
coredump_dir /var/spool/squid3
```

Ik heb de standaard Squid instellingen in deze configuratie wat aangepast voor een optimale performance. Let vooral op de 4 netwerken die hier toegelaten worden en op de instellingen voor het geheugen.

Vervolgens gaan we in de /cache partitie de map “squid” aanmaken.

```
mkdir /cache/squid
```

Chmod deze naar de user en group “proxy”

```
chown proxy:proxy /cache/squid
```

Vervolgens gaan we Squid herstarten. Let op: de eerste keer je Squid herstart, zal hij in /cache/squid een aantal mappen en submappen aanmaken om de gecachte bestanden in op te slaan.

```
/etc/init.d/squid3 restart
```

In de volgende stap gaan we het HTTP verkeer doorverwijzen naar poort 3128 die door Squid gebruikt wordt. We doen dit met behulp van de volgende regel in IPtables (eth0 = externe interface):

```
iptables -A INPUT -m state --state ESTABLISHED,RELATED -j ACCEPT
iptables -t nat -A PREROUTING -s 192.168.0.0/24 -m state --state NEW -p tcp \
--dport 80 -j REDIRECT --to-port 3128
iptables -t nat -A POSTROUTING -o eth0 -j MASQUERADE
```

**Let op:** dit is slechts een klein gedeelte van de IPtables!

Wanneer er op de Squid server ook nog een webserver draait om de gebruikers van een lokale website te voorzien dient volgende regel voor de PREROUTING van bovenstaande IPtables geplaatst te worden (192.168.2.1 = interne interface):

```
iptables -t nat -A PREROUTING -s 192.168.0.0/24 -d 192.168.2.1 -p tcp --dport 80
-j REDIRECT --to-port 81
```

Op de volgende pagina's volgt een voorbeeld van een werkend IPtables script dat gebruikt maakt van een Squid cache proxy.

```
#!/bin/bash

#-----#
#
#           IPTables server.ubuntu.lan v2           #
#           maart 2009                               #
#
#-----#

# De opdracht van dit script is om de IPTables te genereren wanneer de server gestart wordt.
# Plaats daarom een vermelding in /etc/rc.local (Ubuntu).
#
# De IPTables zorgen ervoor dat de server als router kan optreden.
# De zelfgemaakte chain "INFWD" vangt alle pakketjes op die richting de INPUT en FORWARD chain gezonden worden.
# Op deze pakketjes worden vervolgens de regels in dit script toegepast.

#####
# Basisch configuratie

echo "#####"
echo "Firewall regels laden ..... START"

# IPTables binary
IPT="/sbin/iptables"

# Externe interface (WAN kant)
EXTIF="eth0"

# Interne interface (LAN kant)
INTIF="eth1"

# Iedereen in het WAN, de wereld
UNIVERSE="0/0"

# Eigen LAN netwerk
LAN="192.168.0.0/0"

# IP adres van localhost
LOOPBACK="127.0.0.1"

# Klasse A prive netwerk
CLASS_A="10.0.0.0/8"
```

```
# Klasse B prive netwerk
CLASS_B="172.16.0.0/12"

# Klasse C prive netwerk
CLASS_C="192.168.0.0/24"

#####
# Alle tabellen ledigen

$IPT -F
$IPT -X
$IPT -Z
$IPT -t nat -F
$IPT -t nat -X
$IPT -t nat -Z

# Ook ons eigen gemaakt chain "INFWD" (zie verder) ledigen
if [ "`$IPT -L | grep INFWD`" ]; then
    $IPT -F INFWD
fi

# Modules laden
modprobe ip_conntrack
modprobe ip_conntrack_ftp ports=21,2121
modprobe ip_nat_ftp ports=21,2121

# IP doorverwijzing toestaan
echo 1 > /proc/sys/net/ipv4/ip_forward

echo "Alle tabellen ledigen ..... OK"

#####
# Enkele veiligheden

# IP-spoofing beveiliging inschakelen
for f in /proc/sys/net/ipv4/conf/*/rp_filter ; do
echo 1 > $f
done

# ICMP redirect acceptatie uitschakelen
```

```
for f in /proc/sys/net/ipv4/conf/*/accept_redirects ; do
echo 0 > $f
done

# ICMP send_redirects uitschakelen
for f in /proc/sys/net/ipv4/conf/*/send_redirects ; do
echo 0 > $f
done

# Source routed pakketten niet accepteren
for f in /proc/sys/net/ipv4/conf/*/accept_source_route ; do
echo 0 > $f
done

# Log spoofed pakketten, source routed pakketten en redirected pakketten#
for f in /proc/sys/net/ipv4/conf/*/log_martians ; do
echo 1 > $f
done

# TCP SYN cookie beveiliging inschakelen
echo 1 > /proc/sys/net/ipv4/tcp_syncookies

# ICMP broadcasting protection inschakelen
echo 1 > /proc/sys/net/ipv4/icmp_echo_ignore_broadcasts

# ICMP dead error message protection inschakelen
echo 1 > /proc/sys/net/ipv4/icmp_ignore_bogus_error_responses

# Dynamische TCP/IP address hacking inschakelen
echo 1 > /proc/sys/net/ipv4/ip_dynaddr

echo "Veiligheden activeren ..... OK"

#####
# Sommige pakketten resoluut weigeren

## Weiger (en log) alle gefragmenteerde pakketten
$IPT -A INPUT -i $EXTIF -f -j LOG --log-prefix "FRAGMENT! "
$IPT -A INPUT -i $EXTIF -f -j DROP

## Weiger (en log) alles van privatenetwerken op externe iface
$IPT -A INPUT -i $EXTIF -s $LOOPBACK -j LOG --log-prefix "SPOOFING! "
```

```
$IPT -A INPUT -i $EXTIF -s $CLASS_A -j LOG --log-prefix "CLASS A ADDRESS! "  
$IPT -A INPUT -i $EXTIF -s $CLASS_B -j LOG --log-prefix "CLASS B ADDRESS! "  
$IPT -A INPUT -i $EXTIF -s $CLASS_C -j LOG --log-prefix "CLASS C ADDRESS! "  
$IPT -A INPUT -i $EXTIF -s $LOOPBACK -j DROP  
$IPT -A INPUT -i $EXTIF -s $CLASS_A -j DROP  
$IPT -A INPUT -i $EXTIF -s $CLASS_B -j DROP  
$IPT -A INPUT -i $EXTIF -s $CLASS_C -j DROP  
  
echo "Private netwerken op EXTIF weigeren ..... OK"  
  
#####  
# Uiteindelijke regels  
  
# Standaard gaan we alles DROPPEN  
$IPT -P INPUT DROP  
$IPT -P FORWARD DROP  
  
# Alles van de loopback interface toestaan.  
$IPT -A INPUT -i lo -j ACCEPT  
$IPT -A OUTPUT -o lo -j ACCEPT  
  
# Prioriteiten instellen  
$IPT -t mangle -A FORWARD -p tcp --dport 80 -j TOS --set-tos 16  
$IPT -t mangle -A FORWARD -p tcp --dport 22 -j TOS --set-tos 8  
$IPT -t mangle -A FORWARD -p tcp --dport 21 -j TOS --set-tos 2  
  
# Een nieuwe chain aanmaken. Deze chain zal (straks) de regels van zowel INPUT als FORWARD bevatten.  
# We zetten groeperen die regels in dit ene chain zodat we ze niet twee keer moeten aanmaken.  
$IPT -N INFWD  
  
# Paketjes van een bestaande verbinding worden geaccepteerd  
$IPT -A INFWD -m state --state ESTABLISHED,RELATED -j ACCEPT  
  
# Onderstaande regel uncommenten om ALLE (!) verbindingen van het LAN naar het WAN toe te staan  
#$IPT -A INFWD -m state --state NEW -i ! $EXTIF -j ACCEPT  
  
# Ping  
$IPT -A INFWD -m state --state NEW -p icmp -s $LAN -j ACCEPT  
  
# SSH en Telnet  
$IPT -A INFWD -m state --state NEW -p tcp --dport 22 -j ACCEPT  
$IPT -A INFWD -m state --state NEW -p tcp --dport 23 -j ACCEPT
```

```
# DNS
$IPT -A INFWD -m state --state NEW -p tcp -s $LAN --dport 53 -j ACCEPT
$IPT -A INFWD -m state --state NEW -p udp -s $LAN --dport 53 -j ACCEPT

# HTTP + HTTPS + Squid
$IPT -A INFWD -m state --state NEW -p tcp -s $LAN --dport 80 -j ACCEPT
$IPT -A INFWD -m state --state NEW -p tcp -s $LAN --dport 443 -j ACCEPT
$IPT -A INFWD -m state --state NEW -p tcp -s $LAN --dport 81 -j ACCEPT # Eigen HTTP server
$IPT -A INFWD -m state --state NEW -p tcp -s $LAN --dport 3128 -j ACCEPT # Eigen cache proxy Squid

# FTP
$IPT -A INFWD -m state --state NEW -p tcp -s $LAN --dport 20 -j ACCEPT # Data
$IPT -A INFWD -m state --state NEW -p tcp -s $LAN --dport 21 -j ACCEPT # Control
$IPT -A INFWD -m state --state NEW -p tcp -s $LAN --dport 35000:60000 -j ACCEPT # Passieve poorten voor eigen FTP server
$IPT -A INFWD -m state --state NEW -p tcp -s $LAN --dport 989 -j ACCEPT # FTPS data
$IPT -A INFWD -m state --state NEW -p tcp -s $LAN --dport 990 -j ACCEPT # FTPS Control

# MAIL
$IPT -A INFWD -m state --state NEW -p tcp -s $LAN --dport 587 -j ACCEPT
$IPT -A INFWD -m state --state NEW -p tcp -s $LAN --dport 587 -j ACCEPT
$IPT -A INFWD -m state --state NEW -p tcp -s $LAN --dport 465 -j ACCEPT
$IPT -A INFWD -m state --state NEW -p tcp -s $LAN --dport 110 -j ACCEPT
$IPT -A INFWD -m state --state NEW -p tcp -s $LAN --dport 995 -j ACCEPT
$IPT -A INFWD -m state --state NEW -p tcp -s $LAN --dport 143 -j ACCEPT
$IPT -A INFWD -m state --state NEW -p tcp -s $LAN --dport 993 -j ACCEPT

# Chat
$IPT -A INFWD -m state --state NEW -p tcp -s $LAN --dport 1863 -j ACCEPT # MSN
$IPT -A INFWD -m state --state NEW -p tcp -s $LAN --dport 1503 -j ACCEPT # MSN Whiteboard and Application Sharing
$IPT -A INFWD -m state --state NEW -p tcp -s $LAN --dport 6891:6900 -j ACCEPT # MSN File transfer
$IPT -A INFWD -m state --state NEW -p tcp -s $LAN --dport 531 -j ACCEPT # AOL Instant Messenger, IRC
$IPT -A INFWD -m state --state NEW -p tcp -s $LAN --dport 5050 -j ACCEPT # Yahoo! Messenger
$IPT -A INFWD -m state --state NEW -p tcp -s $LAN --dport 5190 -j ACCEPT # ICQ en AOL Instant Messenger
$IPT -A INFWD -m state --state NEW -p tcp -s $LAN --dport 6660:6664 -j ACCEPT # IRC
$IPT -A INFWD -m state --state NEW -p tcp -s $LAN --dport 6665:6669 -j ACCEPT # IRC

# NTP
$IPT -A INFWD -m state --state NEW -p tcp -s $LAN --dport 123 -j ACCEPT

# SMB
$IPT -A INFWD -m state --state NEW -p tcp -s $LAN --dport 445 -j ACCEPT
$IPT -A INFWD -m state --state NEW -p tcp -s $LAN --dport 901 -j ACCEPT # SWAT
```

```

# Controle panelen
$IPT -A INFWD -m state --state NEW -p tcp -s $LAN --dport 2222 -j ACCEPT # DirectAdmin
$IPT -A INFWD -m state --state NEW -p tcp -s $LAN --dport 2082 -j ACCEPT # Cpanel
$IPT -A INFWD -m state --state NEW -p tcp -s $LAN --dport 2083 -j ACCEPT # Cpanel SSL

# Enkel intern
$IPT -A INFWD -m state --state NEW -p tcp -s $LAN -d $LAN --dport 5500 -j ACCEPT # VNC
$IPT -A INFWD -m state --state NEW -p tcp -s $LAN -d $LAN --dport 5800 -j ACCEPT # VNC over HTTP
$IPT -A INFWD -m state --state NEW -p tcp -s $LAN -d $LAN --dport 5900 -j ACCEPT # VNC
$IPT -A INFWD -m state --state NEW -p udp -s $LAN -d $LAN --dport 5900 -j ACCEPT # VNC

# -----

# Alle andere poorten van ons zelfgemaakte chain DROPPEN
$IPT -A INFWD -j DROP

# Alle pakketjes voor INPUT of FORWARD gaan we doorsturen naar ons eigen chain
$IPT -A INPUT -j INFWD
$IPT -A FORWARD -j INFWD

echo "Regels laden ..... OK"

#####
# PREROUTING

# Squid HTTP cache-proxy
# Trafiek voor de lokale website toestaan richting poort 81 (zie /etc/apache2/ports.conf)
$IPT -t nat -A PREROUTING -s $LAN -d 192.168.2.1 -p tcp --dport 80 -j REDIRECT --to-port 81

# Alle ander HTTP verkeer doorsturen naar Squid, om vervolgens naar buiten te gaan
$IPT -t nat -A PREROUTING -s $LAN -m state --state NEW -p tcp --dport 80 -j REDIRECT --to-port 3128

echo "Prerouting laden ..... OK"

#####
# POSTROUTING

# Geef systemen binnen het LAN het IP van de externe interface om naar het WAN te gaan, principe van NAT
$IPT -t nat -A POSTROUTING -o $EXTIF -j MASQUERADE

```

```
echo "Postrouting laden ..... OK"
```

```
echo "Firewall regels laden ..... EINDE"
```

```
echo "#####"
```