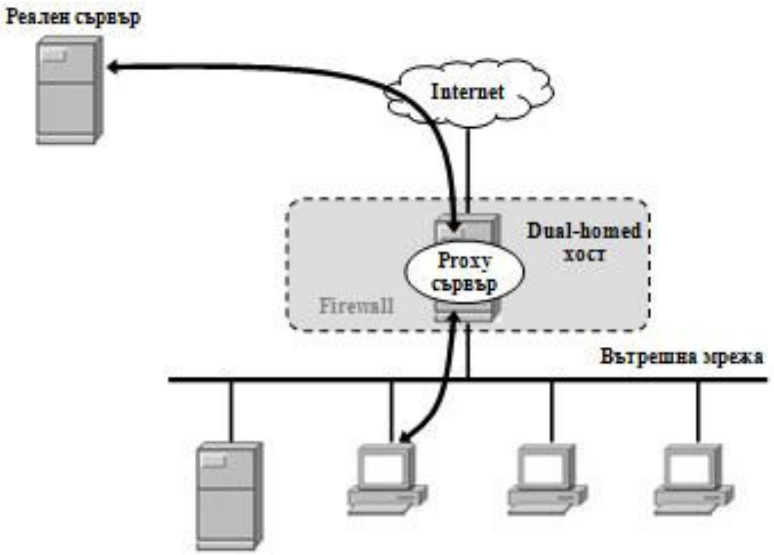
Прокси сървъри

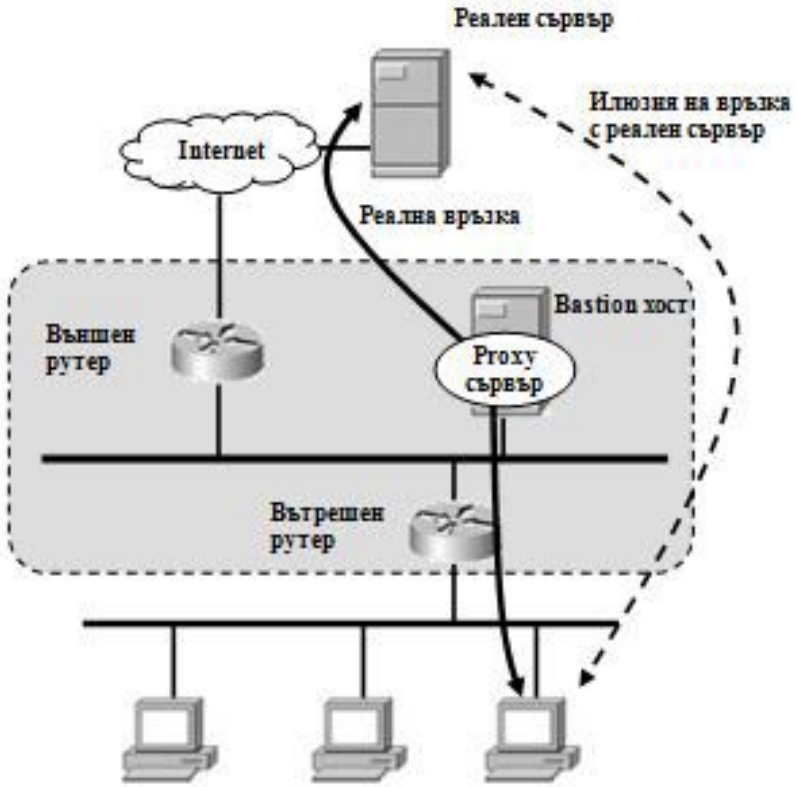
Прокси сървър

- Това е сървър-посредник -компютърна система или приложни програми.
- Той действа като посредник за исканите от клиентите ресурси/услуги (файл, връзка, webстраница) от други сървъри.
- Прокси сървърът създава илюзия у клиента, че той директно комуникира с реалния сървър

Разполагане

Проксисървър може да бъде поставен на различни места между локалния компютър на потребителя и сървъра в Интернет, който генерира резултат.





Функциониране на прокси сървър

- Проверява заявката в съответствие със зададени правила за филтриране:

– по IP адрес,

– по протокол.

- Ако искането бъде утвърдено, той:

– свързва се към реалния сървър, който предоставя услугата

– заявява услугата от името на клиента

– предоставя на клиента заявения ресурс

- Прокси сървърът може да променя заявките на клиента или отговора на сървъра.
- В определени случаи той може да обслужи заявката, без да се свърже с конкретен сървър, като връща кеширанрезултат от същата предишна заявка.

Използване

По два начина:

- Прозрачно (transparent) - и двете страни не знаят, че конекцията минава през прокси сървър;
- Пакетирано (opaque) - клиентският софтуер трябва да се конфигурира, за да използва прокси сървър.

Приложение

- Сигурност в мрежата

– да се запазят анонимни машините, които ползват услугата;

• Да се ускори достъпът до ресурси

– чрез кеширане;

• За реализиране на политики за достъп до мрежи и съдържание

– блокиране на нежелани сайтове;

• За акаунтинг

– да се следи и анализира потребление,

– да се сканират предавани данни срещу зловреден софтуер преди тяхното доставяне до потребителя,

– да се сканират изходящи данни;

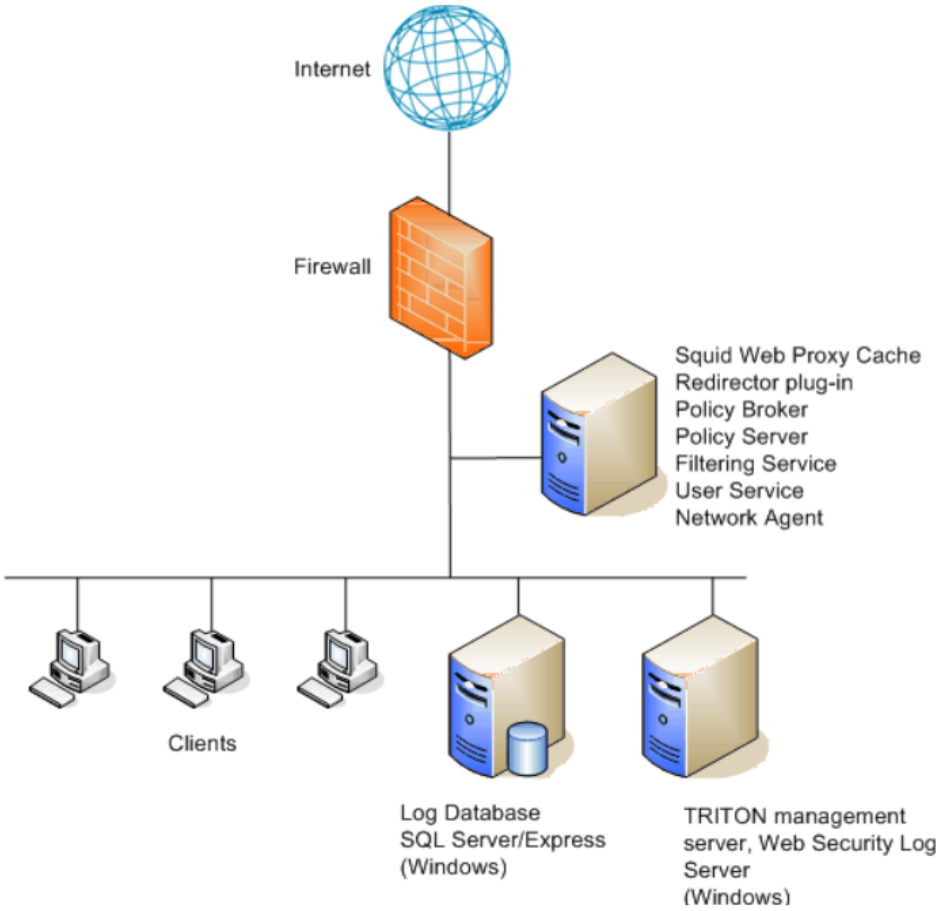
• Да се заобиколятрегионални ограничения;

• Повишаване на производителността

– чрез loadbalancing-в обратна посока, за да балансира натоварването на множество идентични сървъри, които са зад защитна стена, а потребителите са външни

Web прокси сървър

Това е специализиран HTTP сървър, който функционира и като защитна стена, тъй като защитава клиентските машини чрез ограничаване на външен достъп до тях.

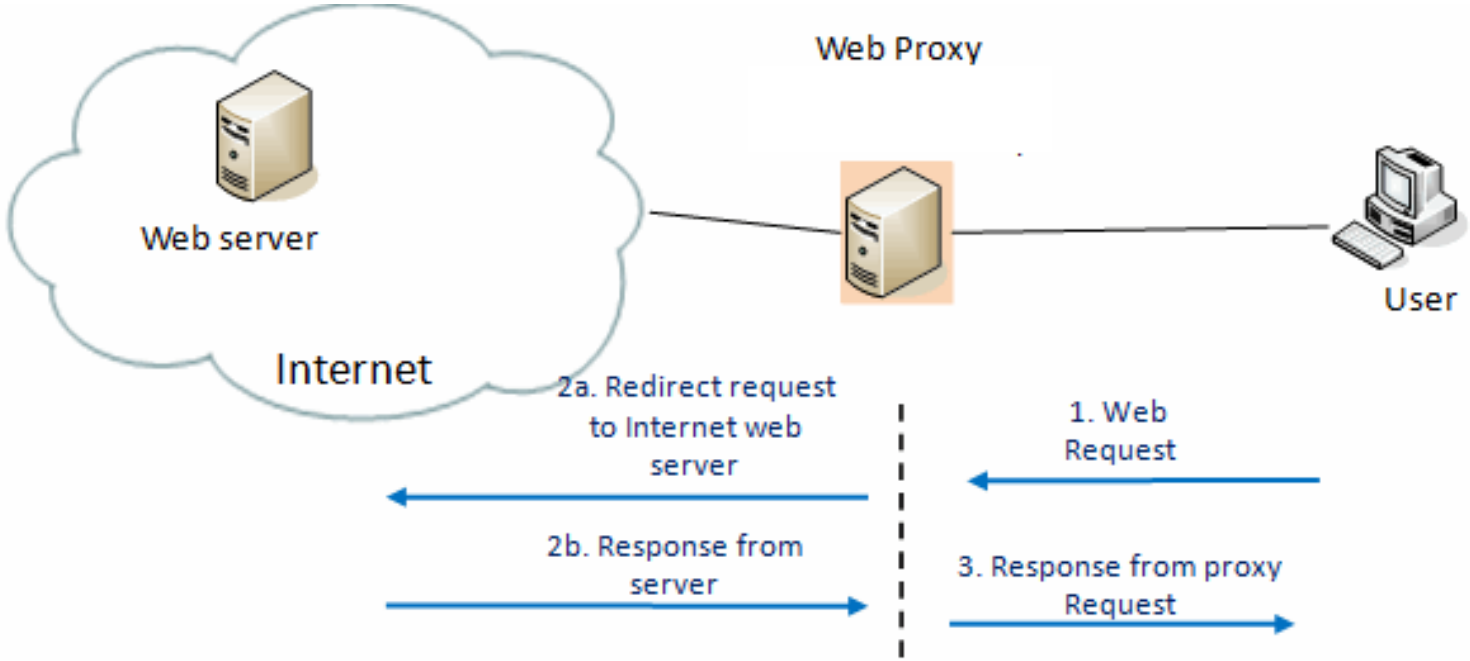


Функциониране на web прокси

- Обикновено всички клиентски машини, които са в една подмрежа, използват един прокси сървър, който кешира заявените документи от клиентските машини.
- Клиентските машини имат илюзията, че комуникират директно с реалния сървър. Така се дава достъп до WWW и на клиентски машини, които нямат конфигуриран DNS, защото те разполагат само с IP адреса на прокси сървъра.

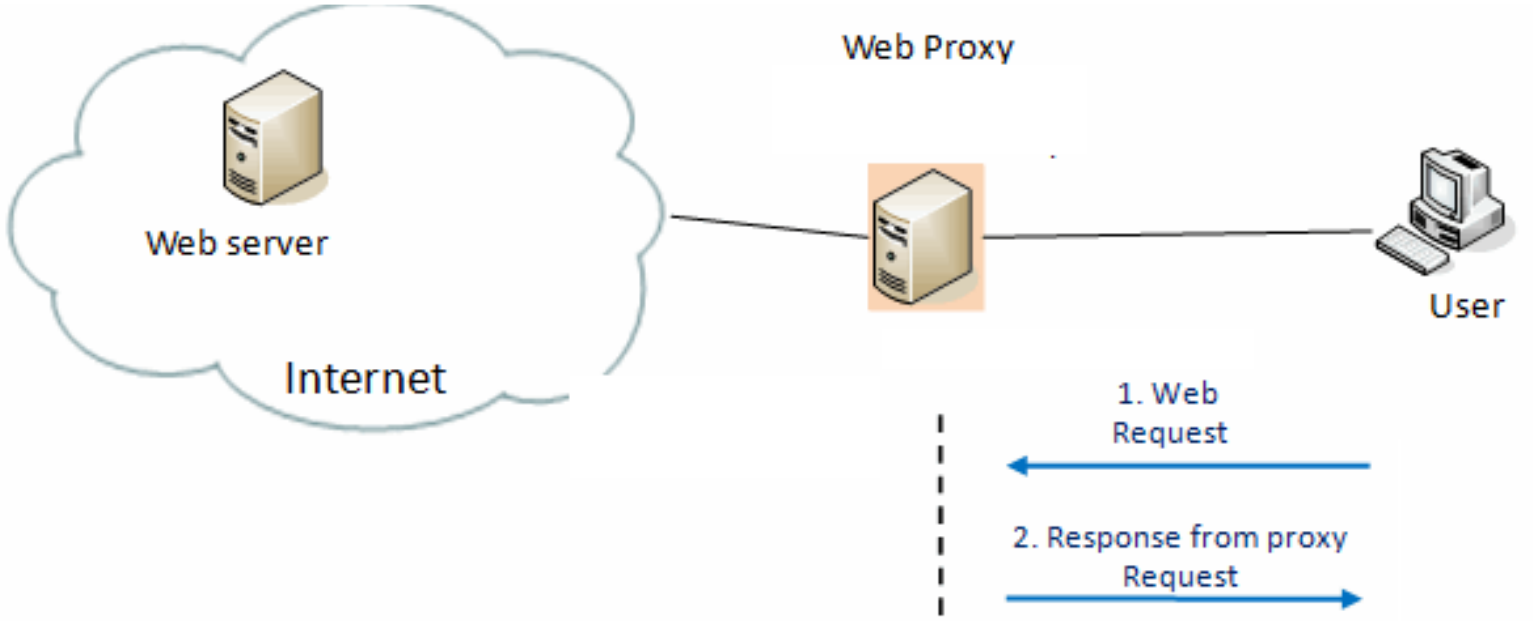
Принцип на работа на web прокси

- Когато заявената страница не е в кеша



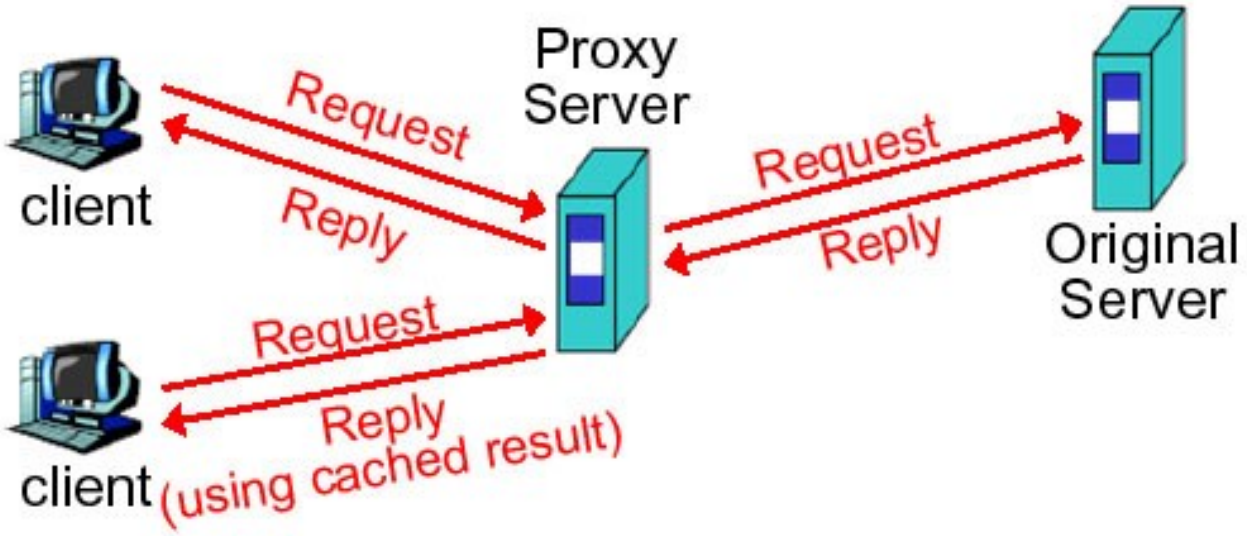
Принцип на работа на web прокси

- Когато заявената страница е в кеша



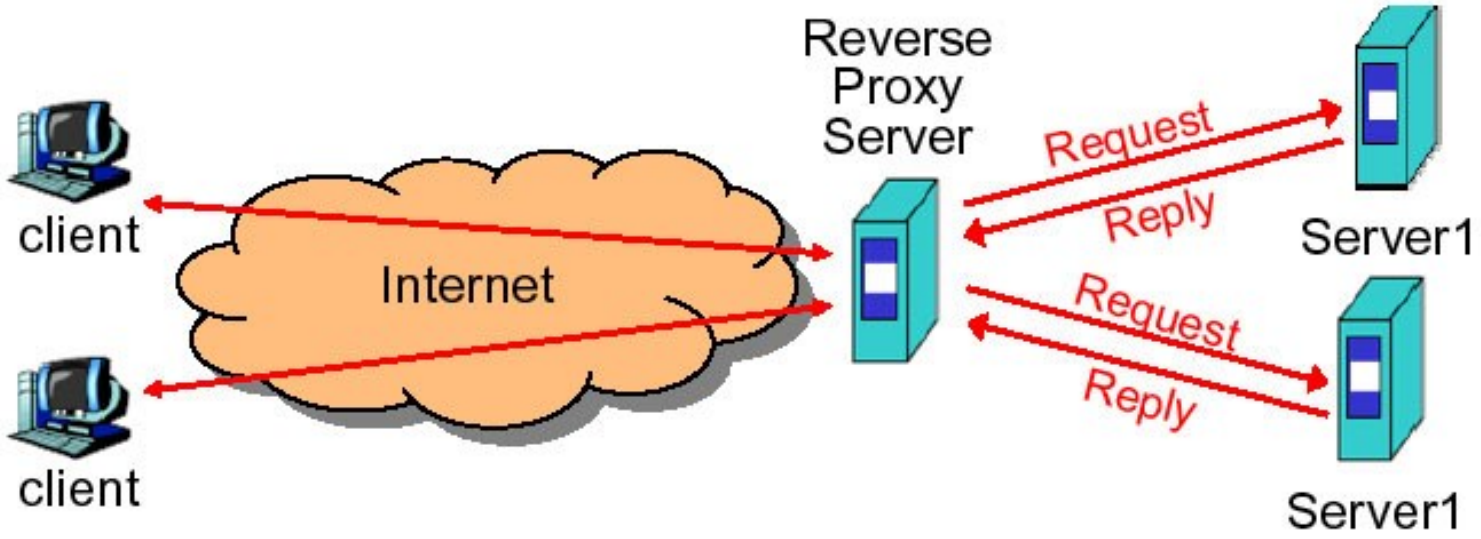
Обслужване на заявки

- **Forward прокси** – прокси за излизащи заявки. Осигурява запазване на bandwidth; подобряване на производителността; централен контрол на трафика. Ако изискваните обекти са в кеша- директно ги доставя. В противен случай изпраща заявката до заявения сървър, кешира отговора и после го връща на клиента



Обслужване на заявки

- **Reverse прокси** – прокси за идваши заявки. Осигурява редуциране на натоварването на сървърите (чрез кеширане); балансиране на натоварването; резервираност при отказ. Действа като оригиналния сървър, приема идващите заявки и връща съответния отговор.



Прокси сървър Squid

- Това е кеширащ прокси сървър под Linux.
- Поддържа протоколи като HTTP, HTTPS, FTP, Ghoper и др.
- Той редуцира мрежовия трафик и подобрява времето за отговор чрез кеширанеи повторно използване при често изискваните web страници.
- Той притежава множество функционалности, като прецизен контрол на достъпа, оторизиране, репликация на съдържанието, контрол на трафика и др.

Инсталиране

Създаване на главна директория

 # mkdir –p /usr/local/squid/src – създава директорията за сорса. Опцията **–p** създава всички родителски директории ако не съществуват (в случая */usr/local/squid*)

Инсталиране

Създаване на потребител и група от чието име ще се стартира squid

 # useradd squid -d /usr/local/squid – задава му се home директорията

 # groupadd squid - групата

Инсталиране

Задаване права на директориите

 # chmod g+s /usr/local/squid /usr/local/squid/src – сменя са правото на достъп до тези папки за групата (+s – указва потребителския или груповия ID да се установи по време на изпълнение)

Инсталиране

Разархивиране

 # cd /usr/local/squid/src

 # tar zxvf squid-3.5.3.tar.gz

Инсталиране

Компилиране

 # cd /usr/local/squid/src/squid-3.5.3

 # ./configure --prefix=/usr/local/squid – стартира се конфигурационен скрипт, които проверява системата за наличието на библиотеки, версия на компилатора и т.н. След завършването му, е генериран Makefile, с който ще се стартира процеса на компилацията. Указаната опция е къде е началото на дървото на директориите за компилиране.

Инсталиране

Компилиране

 # make all – стартира процесът на компилация. Тя чете този Makefile, в който е указано как да компилира всичките файлове според текущата система.

 # make install – инсталира библиотеките и създава необходимите директории

 # make clean – изчиства временните файлове

Инсталиране

Промяна на собственика на новосъздадените директории и файлове

 # chown -R squid.squid /usr/local/squid – задава се собственик.група (-R – рекурсивно и за поддиректориите)

Конфигуриране

В конфигурационния файл /usr/local/squid/etc/squid.conf се записва от името на кой потребител и група ще работи squid и по кой порт ще очаква заявки:

cache_effective_user squid

cache_effective_group squid

http_port 3128

Конфигуриране

Създаване на директорната структура на кеша:

 # /usr/local/squid/sbin/squid –z

Стартиране

 # /usr/local/squid/sbin/squid

Контрол на достъпа

- ACL- списъци за достъп в които се указват
  - хостове
  - мрежи,
  - времеви интервали,
  - сайтове

- Правила за третиране на ACL списъци (оператори) – разрешават или забраняват HTTP достъпа на участниците в списъка

ACL изрази

ACL изрази - осигуряват санкциониран достъп до кешираната информация на сървъра.

Форматът на ACL израз е следният:

acl <aclname> <acltype> <aclvalue> ...

- aclname - произволно, уникално име на списък
- acltype - src; dst; srcdomain; dstdomain; time; url_regex; urlpath_regex; path_auth; maxconn и др. – около 25 типа
- aclvalue - IPAddresses; NetAddresses;

Типове ACL

- src: Източник, т.е. IP-адрес на клиента
- dst: Приемник, т.е. IP-адрес на сървъра
- srcdomain: Източник, т.е. домейна на клиента
- dstdomain: Приемник, т.е. домейна на сървъра
- time: Време в денонощието и ден в седмицата
- url_regex: Регулярен израз за проверка на съответствието на URL
- urlpath_regex: Регулярен израз за проверка на съответствието на URL-пътя заедно с протокола и името на хоста
- proxy_auth: Проверка за username и парола с помощта на външен процес
- maxconn: Максимално количество конекции на един клиентски IP-адрес;

ACL списъци

Последователността от acl изрази с едно и също име формира списък за контрол на достъпа.

ACL оператори

Синтаксис на оператор:

http_access allow| deny [!] <aclname> [[!] <aclname>]

Операторът дефинира две възможни действия, които ще се приложат към конекции, съвпадащи с условията, описани в списъка *aclname*:

- разрешаване на достъпа (allow)
- забрана (deny).

ACL оператори

- Могат да се задават множество списъци към даден оператор, както да има и множество оператори в конфигурационния файл.
- Символът „!” има значение на отрицание на списъка.
- Вместо име на списък може да се зададе *all*, което има смисъл на “всички”.

Изпълнение на списъците

http_access Action statement1 AND statement2 AND statement3

OR

http_access Action statement4 AND ….

Ако заявката удовлетворява описание1 И описание2 И описание3

ИЛИ

Ако заявката удовлетворява описание4

….

ACL изрази

- **Адрес на източника/получателя**

– Указва се с опцията *src* или *dst*.

– Маската може да се зададе в десетична нотация или като мрежов префикс.

acl myNet src 10.0.0.0/255.255.255.0

acl otherNet dst 20.20.20.0/24

http_access allow myNet otherNet

http_access deny all

ACL изрази

- **Домейн на източник/получател**

- Проверката за име на домейн се изпълнява както за източник (*srcdomain*), така и за получател (*dstdomain*).

- Това е удобен начин да се ограничи достъпа до определени сайтове.

- Препоръчва се (от гледна точка на сигурност) задаване на acl и за IP адресите на тези домейни.

acl BadDomain dst domain www.bad.com

http_access deny BadDomain

ACL изрази

- **Регулярни изрази**

- Позволява постигане на по-голяма гъвкавост при списъците за достъп.

- Могат да се проверяват думи, част от думи или шаблони в URL или имената на домейните.

acl Music url_regex music

http_access deny Music

ACL изрази

- Ако думите, които трябва да се проверяват са много, препоръчва се те да се запишат в отделен файл, чието име се задава в израза

acl Allowed_clients src 192.168.0.0/24

acl Banned_sites url_regex "/usr/banned.list"

http_access deny Banned_sites

http_access allow Allowed_clients

http_access deny all

ACL изрази

- **Портове на получателя**

- Указват се кои портове на получателя да се проверяват за съвпадение.

- Могат да се задават множество портове или диапазони от тях.

acl Safe_ports port 443 # ssl

acl Safe_ports port 80 # http

acl Safe_ports port 21 # ftp

acl Safe_ports port 1024-65535 # unregistered ports

http_access deny !Safe_ports

ACL изрази

- **По време на достъп**

acl aclname time [day-list] [start_hour:minute-end_hour:minute]

- day-list указва дните, за които ще се прилага израза.

- Ако не се укаже, се подразбират всичките дни.

- Форматът на времето е 24-часов.

- Крайното време трябва винаги да е по-голямо от началното (ако се указва за другия ден, трябва да се дефинират два израза).

- Ако не е зададено, подразбира се 24 часа.

- Дните на седмицата се указват с първите букви: S -Sunday, M -Monday, T -Thuesday, W -Wednesday, H -Thursday, F -Friday, A -Saturday

ACL изрази

acl School_hours time MTWHF 08:00-14:00

acl Weekend_days time A S

http_access allow Weekend_days

http_access allow !School_hours

http_access deny all

ACL изрази

- **Тип на браузър**

– Контролира от кой вид браузър е заявката.

– Задава се името на браузъра, което е указано в хедъра User-agent на HTTP заявката.

acl Firefox browser firefox

http_access deny !Firefox

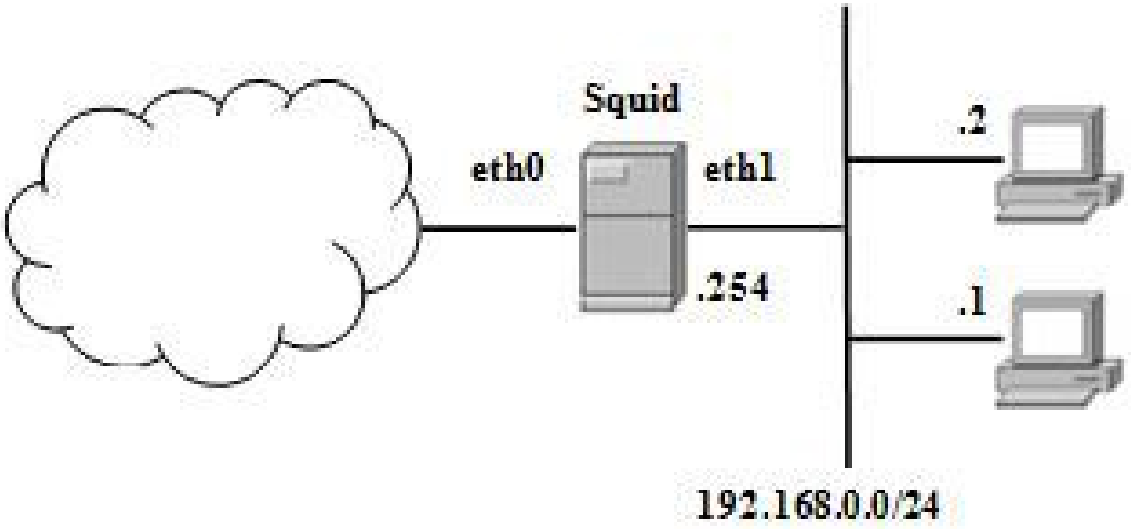
Конфигуриране на прозрачен прокси сървър

- Прозрачен прокси сървър се инсталира на машина, която е конфигурирана да очаква TCP конекциина порт 80.
- За функционирането на прокси сървъра е необходимо на машината, където е инсталиран, да е наличен софтуер за редиректване на трафика.
- Реализира се с пакета iptables

Конфигуриране на прозрачен прокси сървър

- В конфигурационния файл на Squid трябва да се укаже функциониране като прозрачен прокси сървър:

```
http_port 3128 intercept
```

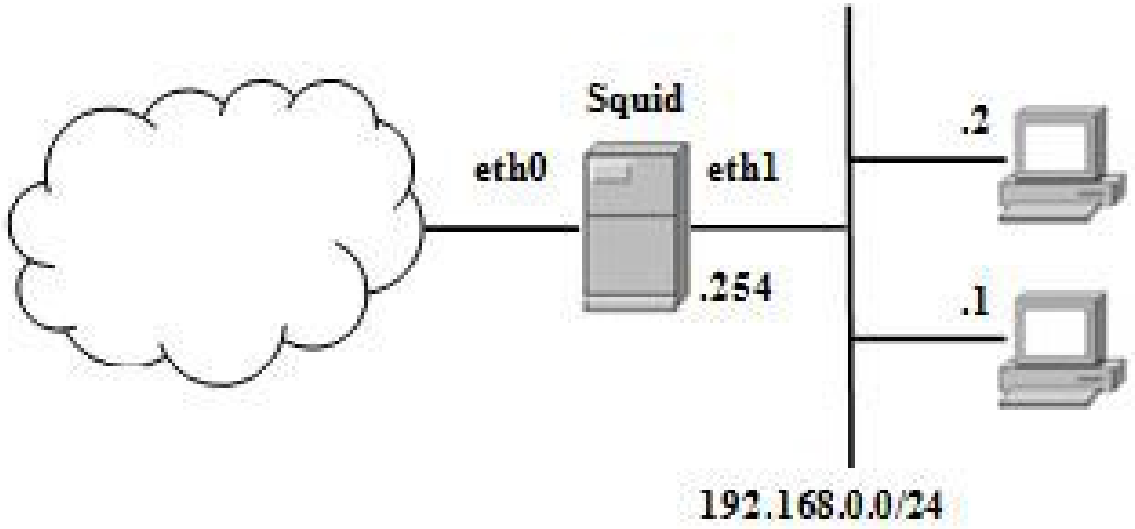


Действие на прозрачен прокси сървър

- Пакети за TCP конекции на порт 80 се прихващат от машината и се предават на прокси сървъра.
- Той проверява разрешенията за достъп и препраща заявката към реалния web сървър.
- Машините от вътрешната мрежа не разбират за преминаването на конекциите през прозрачния прокси.
- Те трябва единствено да бъдат конфигурирани да използват като default gateway машината с прокси сървъра.

Конфигуриране на прозрачен прокси сървър

```
iptables -t nat -A PREROUTING -s 192.168.0.254 \
      -p tcp --dport 80 -j ACCEPT
iptables -t nat -A PREROUTING -p tcp --dport 80 \
      -j REDIRECT --to-port 3128
iptables -t nat -A POSTROUTING -j MASQUERADE
iptables -t mangle -A PREROUTING -p tcp\
      --dport 3128 -j DROP
```



Въпроси ?