Proxy сървъри

проф. д-р инж. Венета Алексиева

Основни моменти

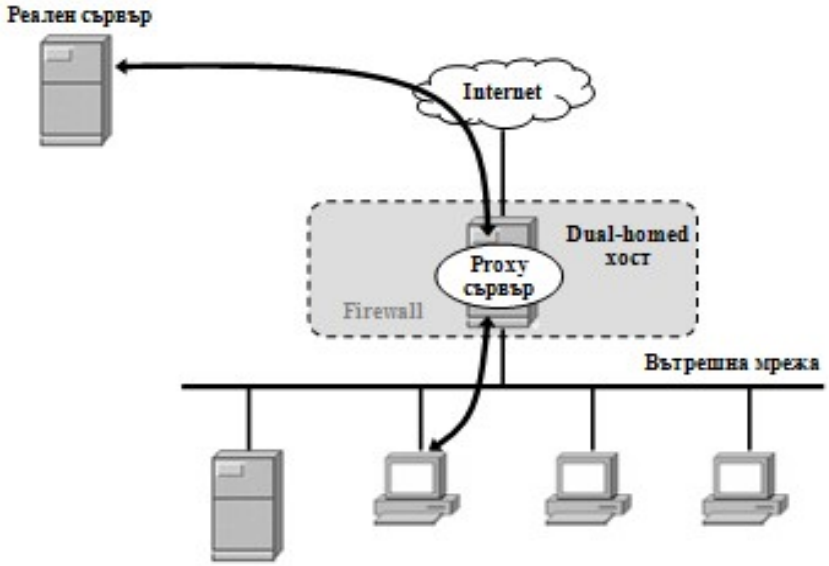
- Proxy сървър
- Топологии
- Конфигуриране
- Squid

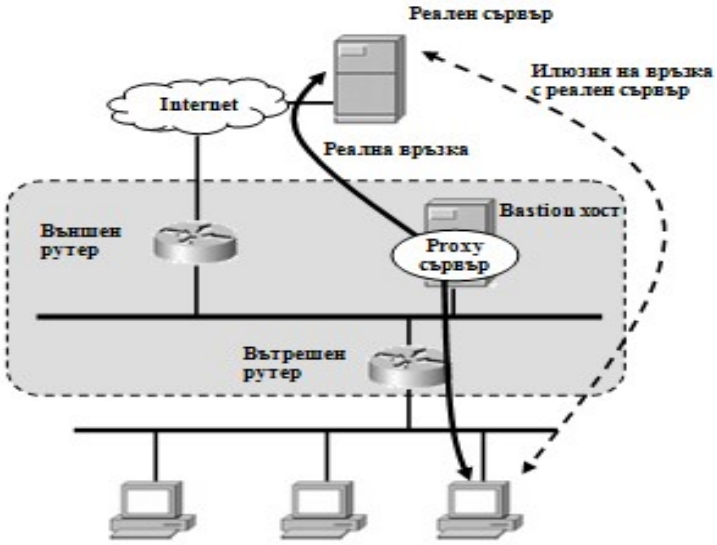
Прокси сървър

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

Място на proxy сървъра в топологията

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





Принцип на действие

- 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.
- Той 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 кеширан резултат от същата предишна заявка.

Употреба

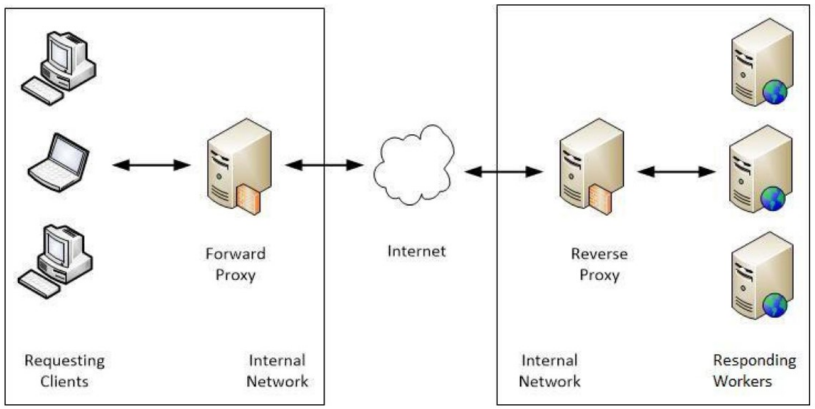
По два начина:

- 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 статистика
  - 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 заобиколят регионални ограничения;
- Повишаване на производителността
  - чрез *load balancing* - в обратна посока, за да балансира натоварването на множество идентични сървъри, които са зад защитна стена, а потребителите са външни

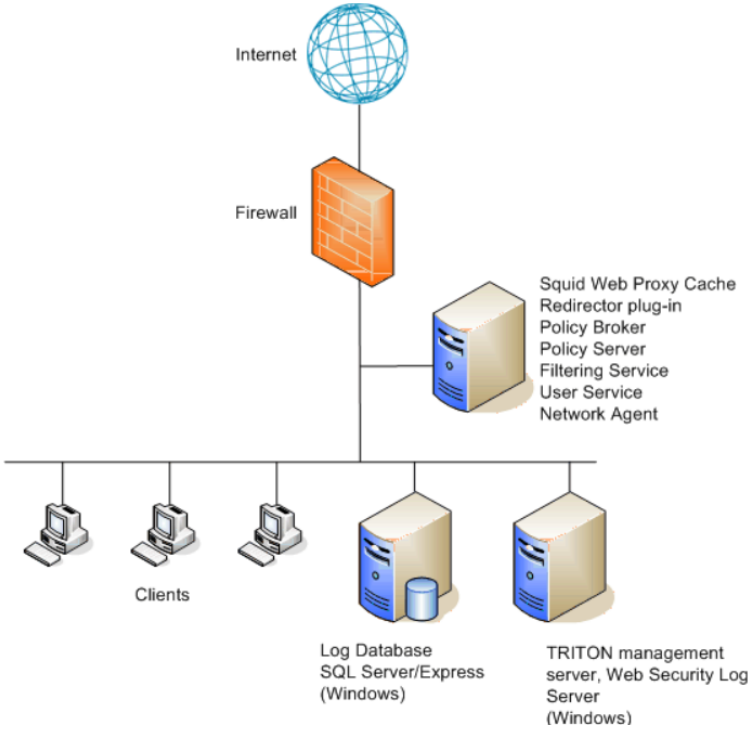
Reverse и forward proxy сървър

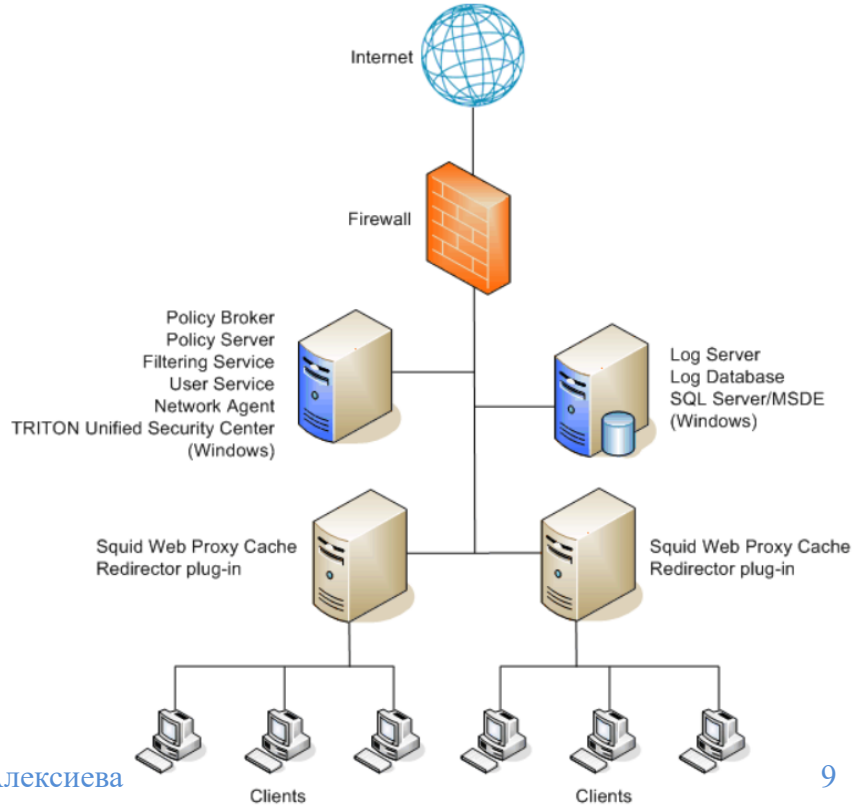


- Forward Proxy пази клиентите в безопасност. Когато клиент изпраща заявки до сървър в Интернет, той работи като посредник между клиента и Интернет. Осигурява:
  - Заобикаляне на блокирани уеб сайтове в мрежата
  - Осигурява анонимност на клиента в Интернет
  - Кешира заявените страници
- Reverse Proxy пази оригиналния сървър в безопасност. Той също се намира между клиента и оригиналния сървър, но тук целта е да се попречи на клиента да се свърже директно с оригиналния сървър. Осигурява:
  - Load balancing
  - Кеширано съдържание
  - Защитава оригиналния сървър от атаки
  - Маскира IP адреса на оригиналния сървър
  - Позволява SSL криптиране на връзката

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.



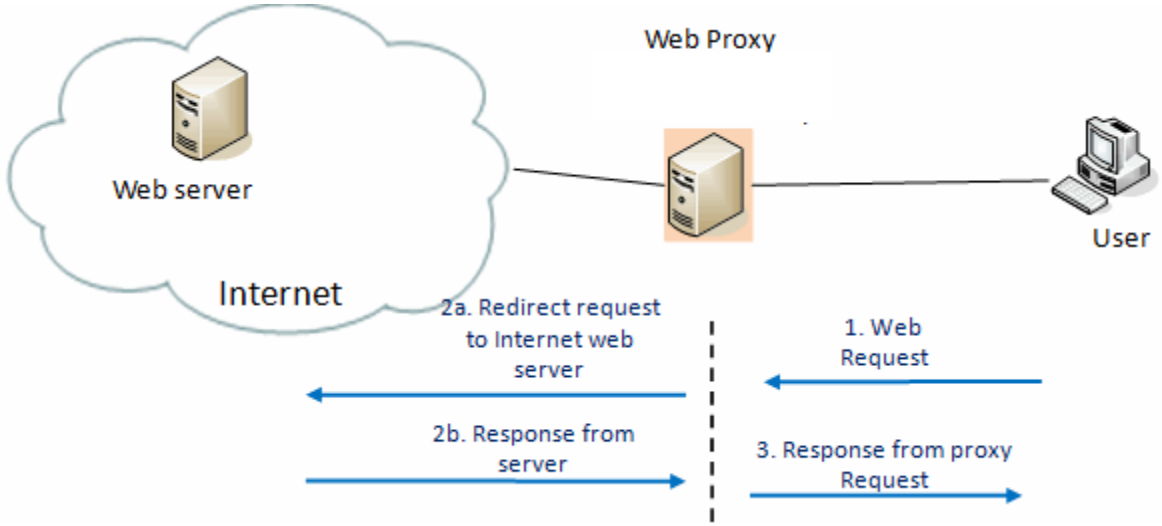


Принцип на работа на Web прокси

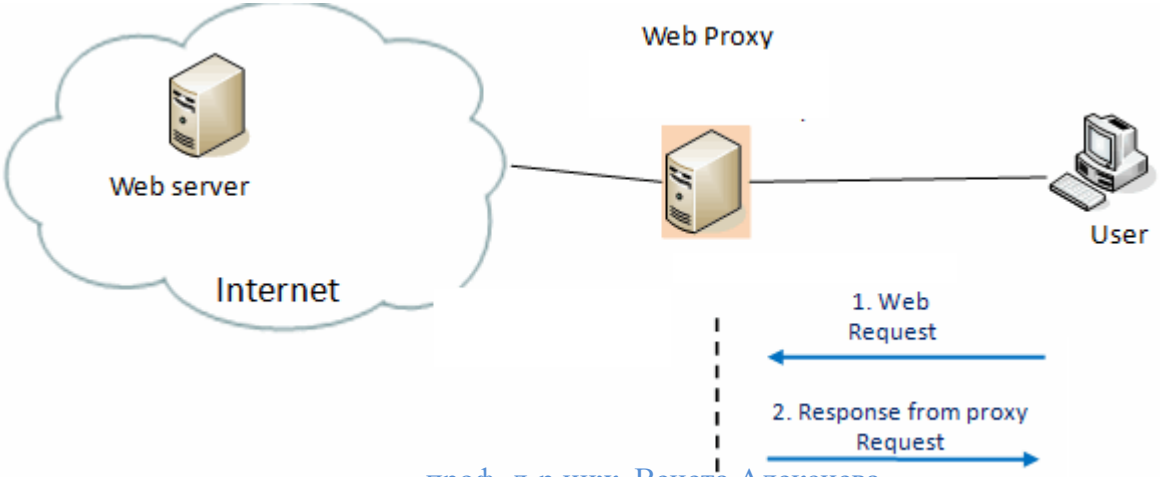
- Обикновено всички клиентски машини, които са в една подмрежа, използват един прокси сървър, който кешира заявените документи от клиентските машини.
- Клиентските машини имат илюзията, че комуникират директно с реалния сървър. Така се дава достъп до WWW и на клиентски машини, които нямат конфигуриран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

Принцип на работа на Web прокси

- Когато заявената страница не е в кеша



- И когато е в кеша

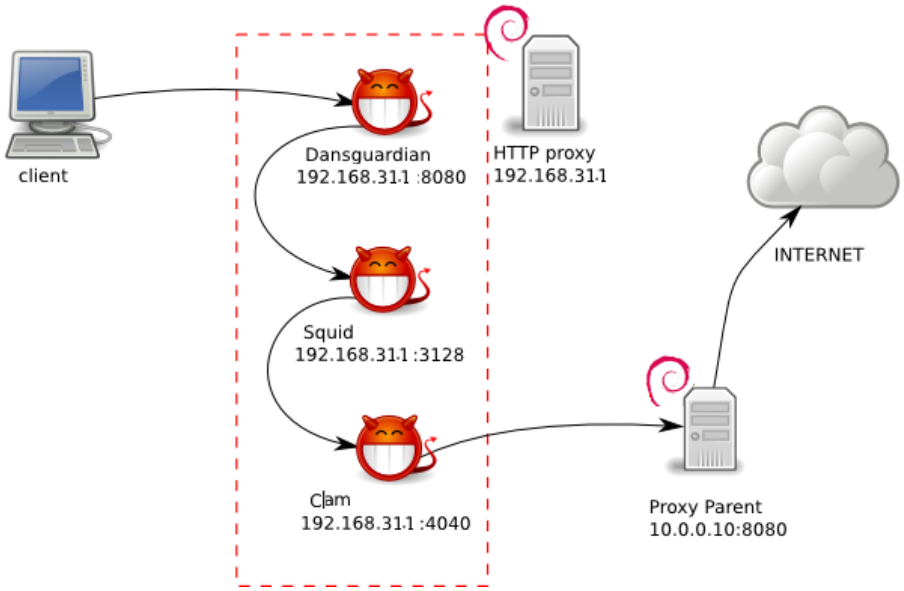


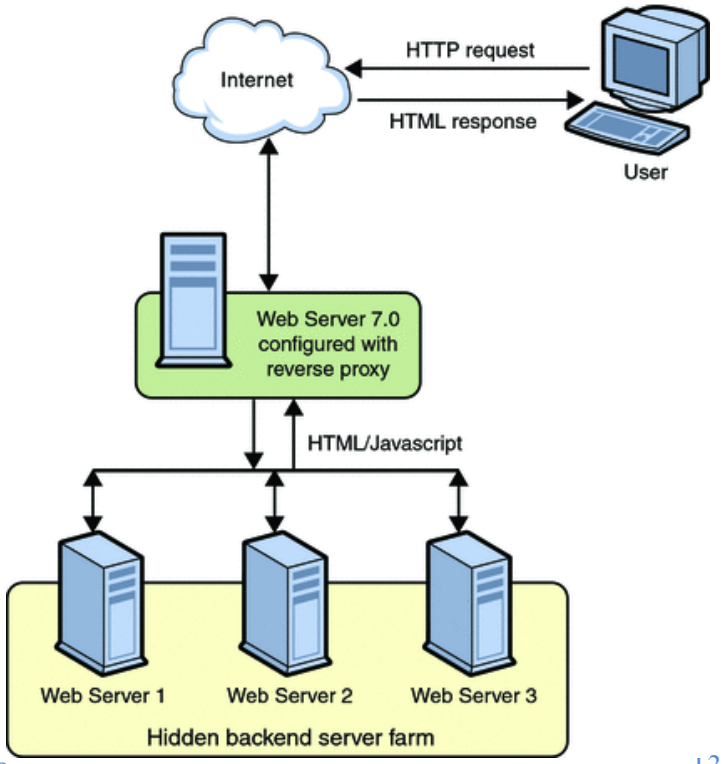
Принцип на работа на Web прокси

- Web прокси сървърът работи едновременно и като сървър, и като клиент.
- Клиентът изпраща заявка до прокси сървър по HTTP.
- Прокси сървърът се обръща към реалния сървър по протоколите HTTP или FTP, в зависимост от контекста на заявката.
- Ако клиентът въведе заявка за HTTP, прокси сървърът я конвертира в GET заявка към съответния реален сървър, получава от него отговор, след което връща на клиента реално изпратения документ.
- Ако клиентът въведе заявка за FTP, прокси сървърът я конвертира в FTP заявка към съответния реален сървър, получава FTP отговор и генерира HTTP документ, който връща на клиента.

Едновременни функции

- Squid сървърът функционира като нормален и reverse proxy едновременно.
- Клиентите се свързват отвън и Squid сървърът работи като reverse proxy между тях и web сървър.
- Едновременно с това Squid сървърът работи като класически кеш сървър, кеширайки HTTP заявките от вътрешните клиенти.





Филтриране

- Web прокси сървърът може да филтрира заявките по приложен протокол:
  - да се изпълняват заявки по FTP, но не и по HTTP;
  - заявки до специфични дестинации по URL.
- Тези рестрикции и разрешения могат да са различни за различните клиентски машини и се базират на техните IP адреси или други критерии.

Кеширане на страници

- Клиентът генерира заявка по HTTP (за web страница).
- Прокси сървърът проверява за наличието й в паметта си (кеша).
  - Ако не е налична, се обръща със заявка към реалния сървър, полученият отговор се съхранява в кеша и след това се изпраща и до клиента.
  - Ако страницата е налична в кеша, се изпраща веднага отговор на клиента без да се осъществява достъп до Интернет и реалния 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 кеширане и 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.

Конфигуриране

- Настройки в конфигурационния файл: */usr/local/squid/etc/squid.conf*
- Указва се по кой порт ще очаква заявки:

```
http_port 3128 8080
```

  - 3128 - по подразбиране
  - 80 - сървърът трябва да се стартира като *root*.
  - 8080 - псевдо стандартен порт.

Конфигуриране

- Трябва предварително да са създадени в системата потребител и група:

```
#groupadd squid
#useradd –M squid –s /bin/false –g
  squid
```

  - Потребителят се създава без home директория (-M) и без възможност за логване в системата (`/bin/false`)

- Задава се от името на кой потребител и група ще се изпълнява сървърът:

```
cache_effective_user squid
cache_effective_group squid
```

Конфигуриране

- Непосредствено след инсталирането на squid кеш директорната структура не съществува.
- Указва се по какъв начин да се създаде директорната структура на кеша.
  - Например: файлова система на кеша от тип *ufs*, 100Mb дисково пространство за кеша, 16 директории от 1-во ниво и 256 директории от второ ниво:

```
# cache_dir ufs /usr/local/squid/var/cache 100
  16 256
```

- Кешираните обекти се съхраняват в голяма йерархия, която трябва да бъде създадена преди първото стартиране на прокси сървъра:

```
# /usr/local/squid/sbin/squid -z
```

- Промяна на правата на кеш директориите на squid-а да не са вече за root, а за squid

```
#chown -R squid:squid /usr/local/squid/var
```

Стартиране

```
# /usr/local/squid/sbin/squid
```

- Лог файлът по подразбиране:

/usr/local/squid/var/logs/cache.log.

Контрол на достъпа в Squid

- с ACL (Access Control Lists)
  - да осигурят санкциониран достъп до кешираната информация на сървъра.
  - Контролът на достъпа включва два елемента - *класове* и *оператори*.
  - Класовете се дефинират с конфигурационния таг ***acl***,
  - Имената на операторите са различни, но най-често използваният е ***http_access***.
- Форматът на ACL израз е следният:

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

  - ***aclname*** – произволно, уникално име на списък.
  - ***acltype*** – определя типа на израза, от който зависи начина на прилагането му (>25 типа изрази)
  - ***aclvalue*** – определя условието за съвпадение - диапазон на IP адреси, регулярен израз, имена на домейни и др.
- Последователността от ***acl*** изрази с едно и също име формира **списък за контрол на достъпа.**

Контрол на достъпа в Squid

- Всички оператори имат един и същ синтаксис:

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***).
- 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*”
- Пример: Оператор ***http_access***.

Контрол на достъпа в Squid

- Правилата се изпълняват по реда, в който са въведени и в следната последователност:

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 ***И*** *….*

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.
  - Пример:

```
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
```

- **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 dstdomain www.sex.com
http_access deny BadDomain
```

Регулярни изрази (1)

- 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.
- Например:

```
acl Music url_regex music
http_access deny Music
```

- 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
```

Регулярни изрази (2)

- Портове на получателя
  - Указват се кои портове на получателя да се проверяват за съвпадение.
  - Могат да се задават множество портове или диапазони от тях.
  - Пример :

```
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
```

Регулярни изрази (3)

- По време на достъп
- Форматът на този израз е следният:

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**
  - Пример:

```
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
```

Регулярни изрази (4)

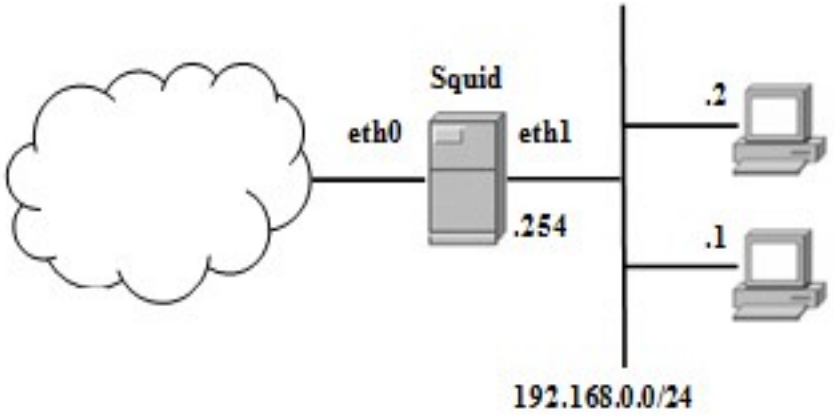
- Тип на браузър
  - Контролира от кой вид браузър е заявката.
  - Задава се името на браузъра, което е указано в хедъра *User-agent* на HTTP заявката.
  - Пример:

```
acl Firefox browser firefox
http_access deny !Firefox
```

Конфигуриране на прозрачен прокси сървър

- Transparent proxy сървър се инсталира на машина, която е конфигурирана да очаква TCP конекции на 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*.
- В конфигурационния файл на Squid трябва да се укаже функциониране като прозрачен прокси сървър:

```
http_port 3128 intercept
```



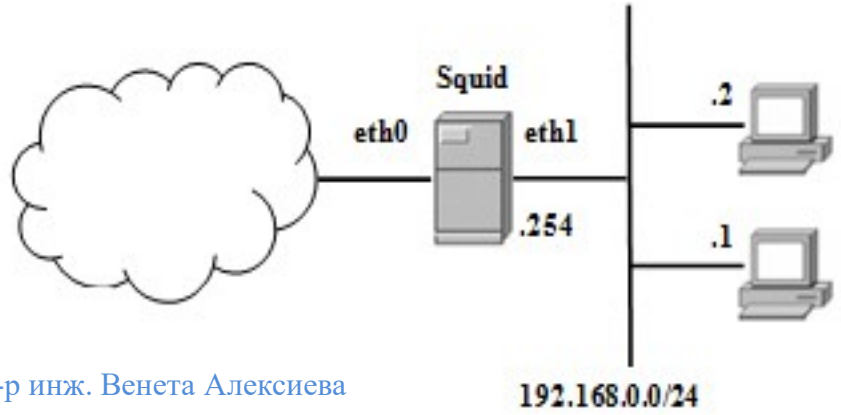
Действие на прозрачен прокси сървър

- 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 прозрачното прокси.
- Те трябва единствено да бъдат конфигурирани да използват като default gateway машината с прокси сървъра.

Препоръчителна техника

- Препоръчва се да се блокира външния достъп до вътрешния порт, по който очаква заявки прокси сървърът.
- Пример:

```
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 ?

Благодаря за вниманието !