Расширенное администрирование FreeBSD.
Блок 5.

v 1.06

Оглавление

Proxy-сервер Squid

Кеширование запросов по протоколам
- HTTP и FTP.
- Работа с протоколом HTTPS.
- Организация иерархии proxy-серверов.
- Прозрачное proxy (transparent proxy).
- Контроль доступа к ресурсам (ACL).
- Перенаправление запросов (redirect).
- Ограничение ширины канала (delay pools).
- Поддержка SNMP.

Proxy-сервер Squid поддерживает следующие возможности:
- Кеширование запросов пользователя по протоколам HTTP, FTP.
- Работу с протоколом HTTPS.
- Организацию иерархии proxy-серверов.
- Аутентификация.
- Прозрачное proxy (transparent proxy).
- Контроль доступа к ресурсам (ACL).
- Перенаправление запросов (redirect).
- Ограничение ширины канала (delay pools).
- Поддержка SNMP.
- И некоторые другие возможности.

Конфигурационный файл squid.conf

После установки конфигурационный файл прокси-сервера будет находиться в директории /usr/local/etc/squid. Файл называется squid.conf.
В файле squid.conf описываются все конфигурационные параметры, которые администратор может изменять по своему усмотрению.
Символ комментария — #. Все, что будет написано после этого символа до конца строки, считается комментарием. Комментарии в squid.conf являются достаточно полной документацией по опциям squid. Но все же, самую полную документацию можно найти на сайте проекта http://www.squid-cache.org.
Поскольку количество конфигурационных параметров достаточно велико, в следующих разделах будут рассмотрены только наиболее интересные. В файле будут встречаться строки, подобные приведенным ниже:

```
#Default
#http_port 3128
```

Таким образом, показывается значение по умолчанию соответствующего параметра. Если Вы хотите изменить параметр, необходимо убрать символ комментария перед ним и присвоить ему новое значение.

Параметры, влияющие на сетевые подключения

```
http_port
https_port
icp_port
```

Параметр http_port

В первую очередь необходимо определить номер порта, к которому будут подключаться клиенты, желающие воспользоваться услугами proxy-сервера. За определение номера порта отвечает параметр http_port.

```
http_port 3128
```

В качестве параметра можно использовать пару: IP:порт. В этом случае будет определен и интерфейс и порт, на котором squid будет слушать запросы от клиентов.

Значение по умолчанию: 3128.

Параметр https_port

```
[ip:]port cert=certificate.pem [key=key.pem] [options...]
```

Этот параметр необходимо применять только в режиме кеширования Вашего WEB-сервера. Он определяет порт, на котором proxy-сервер будет слушать SSL запросы.

Необходимо указывать параметр cert, определяющий файл с сертификатом.

Значение по умолчанию: не определено.

Параметр icp_port.

Параметр определяет порт, на которому squid принимает ICP запросы.

Если Вы будете объединять прокси-сервера в иерархическую структуру, этот порт будет использоваться для обмена командами.

Значение по умолчанию: 3130

Параметры, организации иерархии прокси-серверов

cache_peer cache_peer_domain hierarchy_stoplist

Прокси-сервер squid позволяет объединять сервера в иерархическую структуру. Вы будете сами подключать свой сервер к уже существующей иерархии. Для того, чтобы вы могли подключиться к какому либо серверу, на нем вам должно быть разрешено подключаться как клиенту прокси-сервера, и открыт доступ к ICP порту.

Параметр cache_peer

При помощи этого параметра можно указать прокси-сервер, к которому Вы будете подключаться, а также некоторые параметры подключения.

Например:

```
cache_peer  parent.foo.net  parent  3128  3130
cache_peer  sib1.foo.net  sibling 3128  3130
cache_peer  sib2.foo.net  sibling 3128  3130
```

Тут определены три сервера, к которым вы подключаетесь. Тип взаимоотношения с ними: parent, siblin или multicast. Порт, по которому будут передаваться данные: 3128. Порт, по которому будут передаваться команды: 3130.

Значение по умолчанию: не определено.

Типы взаимоотношений между серверами

При взаимоотношениях parent, если в кеш родительского сервера нет необходимого нам ресурса, он сам осуществляет его поиск в Интернет. И после этого передает ресурс нашему серверу.

При взаимоотношениях типа sibling, если в кеш родительского сервера нет необходимого ресурса, нам самим придется осуществлять его поиск в Интернет.

Параметр cache_peer_domain

При помощи этого параметра можно более точно распределить запросы между родительскими серверами.

Например, существует сервер, который наиболее лучше обслуживает запросы к WEB-серверам, находящимся в доменах ru и ua. И нам необходимо сделать так, чтобы запросы к ресурсам, находящимся в этих доменах, направлялись на этот прокси-сервер. Тогда параметр cache_peer_domain необходимо определить следующим образом.

```
cache_peer_domain proxy.net.ru .ru .ua
```

Обратите внимание на точки, с которых начинаются имена доменов. Если не указать точку, то на сервер proxy.net.ru будут пересылаться только запросы http://ru и http://ua. Все остальные домены в зонах ru и ua не будут обрабатываться этим параметром.

Значение по умолчанию: не определено.

Параметр hierarchy_stoplist

Параметр определяет последовательность символов, которые могут встречаться в запросе. Если эти символы присутствуют, тогда наш сервер не будет обращаться к серверу в иерархии.

Обычно при помощи параметра определяются символы, которые встречаются при обращении к динамически генерируемым ресурсам.

Параметр определен явно:

```
hierarchy_stoplist cgi-bin ?
```

Параметры, определяющие размер кеш прокси-сервера

cache_mem
memory_pools
cache_swap_low
cache_swap_high
maximum_object_size
maximum_object_size_in_memory
cache_replacement_policy

Параметр cache_mem

Параметр определяет размер оперативной памяти, отводимой под кеш proxy-сервера. Этот параметр не влияет на то, сколько памяти будут занимать процессы squid.

Учтите, что в оперативной памяти хранятся не только WEB-ресурсы, но и индекс поиска в кеш сервера.

В среднем прокси сервер использует около 10 Мбайт оперативной памяти на каждый Гбайт дискового кеш, плюс около 10-20 Мбайт на объекты хранящиеся в оперативной памяти, плюс память, определяемая параметром cache_mem.

Значение параметры выбирается следующим образом — если вы планируете выделить для работы прокси сервера X Мбайт оперативной памяти, то значение cache_mem должно быть равно X/3.

Значение по умолчанию: 8 MB.

Параметр memory_pools

Если значение параметра memory_pools равно off, тогда прокси-сервер освобождает не используемую оперативную память. Иначе оперативная память все равно занимается сервером, для того, что бы при необходимости быстро ее распределить.
На серверах с небольшим количеством оперативной памяти рекомендуется устанавливать значение параметра в off.

Параметры cache_swap_low и cache_swap_high

Параметры определяют, когда будет включаться механизм очистки кеш прокси-сервера.
Заполнение кеша прокси-сервера продолжается до тех пор, пока объем кеш не превысит значение параметра cache_swap_high. После достижения этого значения, операция помещения информации в кеш прекращается и включается механизм очистки кеш. Как только объем информации, хранящейся в кеш, достигнет значения параметра cache_swap_low, информация снова будет попадать в кеш.

Значения параметров определяются в процентах. Поэтому, если вы измените размер кеш, обязательно измените значения этих параметров для того, чтобы рационально использовать пространство кеш.

Значения по умолчанию:

```
cache_swap_low 90
cache_swap_high 95
```

Параметр maximum_object_size

Параметр определяет максимальный размер объекта, который можно поместить в кеш на диске.
Следует очень внимательно относиться к определению этого параметра. Размер объекта должен быть меньше свободного пространства, определяемого параметром cache_swap_high.

Значение по умолчанию: 4096 KB

Параметр maximum_object_size_in_memory.

Параметр определяет максимальны размер объекта, который можно поместить в кеш в оперативной памяти.

Значение по умолчанию: 8 KB

Параметр cache_replacement_policy

Параметр определяет политику удаления информации из кеш. Возможные значения: lru, heap GDSF, heap LFUDA и heap LRU.

Значение по умолчанию: lru

Параметры, определяющие месторасположения кеш и журнальных файлов

cache_dir cache_access_log cache_log cache_store_log debug_options

Параметр cache_dir

Параметр определяет положение на диске, размер и некоторые другие параметры кеш.

- Первый параметр — ufs, менять не надо.
- Второй параметр — определяет, в какой директории будет располагаться кеш proxy-сервера.
- Третий параметр — это размер, отводимый под кеш на диске в MB.
- Четвертый параметр — это количество директорий первого уровня.
- Пятый параметр — это количество директорий второго уровня.

Дальше могут быть определены необязательные параметры.

Для чего необходимо создавать такое количество директорий в кеш на диске? Для ускорения доступа к файлам. Доступ осуществляется гораздо быстрее, если объекты распределены по директориям, чем если бы они все были помещены в одной директории. Поэтому, чем больше размер кеш, тем большее количество директорий необходимо создавать.

Значение по умолчанию зависит от дистрибутива.

Параметр cache_access_log

Параметр определяет имя файла, в который squid будет помещать информацию о том, кто и когда обращался к прокси-серверу, какие объекты были запрошены, а так же получил ли он требуемый объект или нет.

Этот файл растет очень быстро. На каждый запрос в файл помещается одна строка размером около 65 байт.

Значение по умолчанию зависит от дистрибутива.

Параметр cache_log

Параметр определяет файл, в который будет помещаться отладочная информация прокси-сервера squid, а подробная информация будет помещена в файл, определяет параметр debug_options.

Значение по умолчанию зависит от дистрибутива.

Параметр cache_store_log

Параметр определяет файл, в который будет помещаться информация о том, какие объекты и куда попадали в кеш. Какие объекты были из него удалены и как долго они хранились в кеш.

Файл растет очень быстро, но его использование можно отключить, используя вместо пути к файлу ключевое слово none.

Значение по умолчанию зависит от дистрибутива.

Параметр debug_options

Параметр определяет, насколько подробная информация будет помещаться в файл cache.log.

Значение по умолчанию:

```
debug_options ALL,1
```

Параметры, влияющие на работу дополнительных программ

ftp_user
ftp_passive

Параметр ftp_user

Параметр определяет e-mail, который будет подставляться в качестве пароля при анонимном доступе к ftp серверам.

Желательно указать реальный e-mail, так как некоторые ftp сервера пытаются определить существование e-mail. И в случае неу-дачи запрещают доступ к серверу.

Значение по умолчанию:

```
ftp_user Squid@
```

Параметр ftp_passive

Параметр определяет режим передачи данных при работе с ftp сервером: пассивный или активный.

Значение по умолчанию:

```
ftp_passive on
```

Также существуют параметры, определяющие программы для перенаправления запросов и программы, используемые для аутентификации пользователей. Эти параметры мы рассмотрим в отдельном разделе.

Списки контроля доступа (ACL)

Типы acl:
- src
- dst
- dstdomain
- dstdom_regex
- urlpath_regex time
- maxconn
- и другие.

В squid ограничение доступа к ресурсам проходит в два этапа:
- Определение условий запроса.
- Разрешение или запрещение определенного на первом шаге условия.

Для определения условий запроса используется параметр acl

```
acl aclname acltype string1 ...
acl aclname acltype "file" …
```

Aclname — имя acl, которое вы задаете сами. Имя должно состоять из английских букв и не содержать пробелы.

Acltype — тип acl, зарезервированное слово.

Дальше необходимо указать дополнительные параметры, которые зависят от типа acl. Если параметров очень много, их можно поместить во внешний файл, по одному на строку. Путь к файлу с параметрами указывают внутри двойных кавычек.

Acl типа src

Этот acl определяет ситуацию, когда запрос к прокси-серверу пришел с IP адреса. В качестве параметров можно указывать:
- IP адрес. Например: 10.10.100.1/32.
- IP адрес сети. Например 10.10.100.0/24
- Диапазон IP адресов. Например: 10.10.100.1-10.10.100.12/32

При написании любых IP адресов обязательно требуется указывать маску подсети. В случае IP адреса машины используется маска 255.255.255.255 или в CIDR формате (битовая маска).

При определении acl можно указывать несколько параметров, разделенных пробелами. Они будут объединяться логическим ИЛИ.

Примеры acl.

```
acl myhost src 10.10.108.20/32
acl mynet src 10.10.108.0/24
acl fr  src 10.10.108.1-10.10.108.12/32
acl any src 10.10.108.21/32 10.10.108.54/32
```

Рассмотрим каждый из примеров:
- В первом примере определяется acl myhost, описывающий запрос, приходящий с машины с IP адресом 10.10.108.20.
- Во втором примере определяется acl mynet, определяющий любой запрос,

пришедший из сети 10.10.108.0/24.

- Третий пример показывает, как определяется диапазон IP адресов.
- Четвертый пример определяет ситуацию, когда запрос приходит с машины с IP адресом 10.10.108.21 или с машины с IP адресом 10.10.108.54.

Acl типа dst

Данный acl определяет ситуацию, когда пользователь посылает запрос к ресурсу, находящемуся на машине с указанным IP адресом.

Acl используется очень редко, поскольку возможна ситуация, когда интересующий WEB сайт меняет свой IP адрес и ограничения, вводимые Вами, перестают работать.

Acl типа dstdomain

Acl служит для описания ситуации, когда пользователь посылает запрос на ресурс с указанным именем.

В качестве параметра можно указывать полное или краткое имя ресурса.

Например:

www.FreeBSD.org — определяет запросы на сайт www.FreeBSD.org
.microsoft.com — определяет запросы на любые машины, имена которых оканчиваются на .microsoft.com.

Примеры acl:

```
acl example dstdomain .microsoft.com
acl rambler dstdomain www.rambler.ru
acl ya  dstdomain .ya.ru .yandex.ru
```

В первом acl показано определение ситуации, когда клиент посылает запрос на любую машину, имя которой заканчивается на .microsoft.com

Во втором примере определяется acl, описывающий ситуацию, когда клиент посылает запрос на www.rambler.ru. Если от пользователя прийдет запрос к другой машине в домене rambler.ru, например mail.rambler.ru, этот запрос не будет совпадать с этим acl.

В последнем примере показано определение acl, отслеживающего запросы от клиентов на все машины, имена которых заканчиваются на .ya.ru или на .yandex.ru.

Acl типа dstdom_regex

В качестве параметра этого acl можно использовать регулярное выражение, определяющее имя машины, к которой обращается клиент.

Для того, чтобы игнорировать регистр букв, перед регулярным выражением необходимо поставить опцию -i.

Пример acl:

```
acl anymail dstdom_regex -i mail
```

Этот acl определяет ситуация, когда пользователь обращается к машине, в FQDN имени которой присутствует слово mail.

Acl типа urlpath_regex

Этот acl можно использовать для определения слова или части слова, встречающегося в запросе клиента. Рассматривается только путь к запрашиваемому ресурсу и имя ресурса.

Пример acl:

```
acl media urlpath_regex -i \.mpg$ \.avi$ \.mp3$
```

Этот acl описывает ситуацию, когда пользователь пытается получить файлы, оканчивающиеся на .mpg, .avi, .mp3.

Acl типа time

При помощи этого acl можно определить в какое время пришел запрос от клиента. В качестве параметра acl следует указывать диапазон времени.

Пример acl:

```
acl wt time 10:00-17:00
```

Acl типа maxconn

Acl maxconn позволяет проверить, не превышает ли количество соединений от одного клиента указанное в acl число.

Пример acl:

```
acl connections maxconn 4
```

Кроме перечисленных выше, существует еще достаточно большое количество типов acl. С некоторыми из них мы познакомимся позже.

Ограничение доступа, параметр http_access

http_access allow|deny [!]acl ...

После описания различных условий запросов при помощи параметра acl, необходимо либо разрешить, либо запретить эти условия. Это можно сделать при помощи параметра http_access.

Предположим, что были описаны следующие acl:

```
acl myhost src 10.10.108.20/32
acl mynet src 10.10.108.0/24
acl wt time 10:00-17:00
acl rambler dstdomain .rambler.ru
```

Необходимо разрешить доступ к ресурсам со своей машины. Разрешить доступ к ресурсам в интернет с машин, находящихся в сети 10.10.108.0/24, но только в рабочее время. А так же запретить им доступ ко всем машинам, имя которых заканчивается на .rambler.ru.

Для этого следует добавить перечисленные ниже строки:

```
http_access allow myhost
http_access deny rambler
http_access allow mynet wt
http_access deny all
```

Особое внимание следует обратить на то, что порядок описания параметров http_access учитывается при рассмотрении правил. Если правило срабатывает, следующие в списке правила не рассматриваются. Порядок описания acl может быть любой, главное, чтобы они были определены до момента их использования в конфигурационном файле.
В приведенном примере сначала разрешается доступ к любым ресурсам с машины, определенной при помощи acl myhost.

Затем запрещается доступ к ресурсам, определенным при помощи acl rambler.

В третьей строке разрешается доступ к ресурсам, но при условии срабатывания обеих указанных acl: mynet и wt. В качестве опций http_access можно указывать несколько acl, перечисленных через пробел. В этом случае они объединяются при помощи логического И.

И, в завершении, запрещается доступ всем остальным. Acl all описан по умолчанию и означает любой источник запроса.

Административные параметры

cache_mgr cache_effective_user cache_effective_group

Параметр cache_mgr

Параметр определяет e-mail администратора proxy-сервера. Этот e-mail будет выводиться на странице ошибки.

Параметры cache_effective_user и cache_effective_group

Параметры определяют пользователя и группу, с правами которых будет работать proxy-сервер.

Директории, в которых находится кеш и журнальные файлы proxy-сервера, должны быть доступны на запись для этих пользователей. Значения по умолчанию зависит от дистрибутива.

Запуск, останов и работа с proxy сервером

squid squid -z

Управлять запуском и остановом proxy-сервера Squid можно как при помощи стартовых скриптов, так и путем явного вызова программы.

Инициализация кеш proxy сервера

После установки и первого изменения конфигурационного файла squid.conf, необходимо инициализировать кеш прокси-сервера на диске. Это можно сделать, выполнив следующую командную строку:

```
squid -z -D -F
```

Опция -z — заставляет инициализировать кеш, то есть. создать все необходимые для работы кеш директории, количество которых определяется параметром cache_dir.

Опция -D — при старте squid отключает проверку работоспособности DNS. При проверке прокси-сервер пытается получить IP адреса известных сайтов: www.microsoft.cm, www.netscape.com и т.д. Опцию -D желательно использовать когда машина, на которой установлен прокси-сервер, не имеет доступа к Internet или DNS серверу.

Опция -F — говорит прокси-серверу, чтобы он не обрабатывал запросов, во время создания кеш.

Запуск и останов proxy-сервера squid

Наиболее корректным способом запуска и останова демона является использование управляющего скрипта:

```
/usr/local/etc/rc.d/squid
```

Для того, чтобы разрешить запуск squid из управляющего скрипта необходимо разрешить его запуск в /etc/rc.conf:

```
squid_enable=yes
```

Запускать и останавливать прокси-сервер из командной строки можно следующим образом:

```
squid
```

Если необходимо, при запуске можно добавить опцию -D.

Останов прокси-сервера осуществляется,скриптом:

```
/usr/local/etc/rc.d/squid stop
```

или путем посылки сигнала 15:

```
killall squid
```

или следующим образом:

```
squid -k shutdown
```

В файле /usr/local/squid/logs/squid.pid хранится PID программы squid. Если программа была выключена не корректно, то она не удаляет этот файл. При старте прокси-сервер смотрит наличие этого файла и если он существует — прокси-сервер не запускается.
Как будет называться и где будет находится этот файл, зависит от параметра pid_filename в squid.conf.

Управление proxy-сервером

Если при работающем прокси-сервере в его конфигурационный файл были внесены изменения, сервер необходимо заставить перечитать свой конфигурационный файл:

```
/etc/init.d/squid reload
```

или

squid -k reconfigure

Ротация журнальных файлов proxy-сервера squid

Squid не пользуется услугами системы Syslog для ведения журнальных файлов. Это объясняется большим количеством информации, помещаемой в журнальные файлы. Поэтому ротацию журнальных файлов можно делать при помощи самого прокси-сервера:

```
squid -k rotate
```

Лабораторная работа А.

Цель работы.

Научиться производить первоначальную настройку-прокси сервера squid.

Задача лабораторной работы.

Произвести первоначальную настройку прокси-сервера. Разрешить клиентам обращаться ко всем ресурсам в интернет кроме *.rambler.ru.

Задачи	Описание
1. Внесение изменений в конфигурационный файл squid.conf.	1. Откройте на редактирование файл squid.conf. 2. Перейдите в раздел, где описываются ACL. 3. После списка стандартных ACL добавьте следующие строки: **acl myhost src ВАШ_IP_АДРЕС** **acl mynet src 172.16.1.0/24** **acl rambler dstdomain .rambler.ru** **acl wt time 10:00-22:00** Время уточните у преподавателя. 4. В конце секции, где описаны параметры http_acces сразу перед последним http_access deny all, добавьте следующие строки: **http_access allow myhost** **http_access deny rambler** **http_access allow mynet wt** **http_access deny all** 5. Сохраните файл. 6. Заставьте прокси-сервер перечитать squid.conf **squid -k reconfigure**
3. Подключение клиента.	1. Откройте браузер и настройте его на работу с proxy сервером соседа. 2. Попытайтесь зайти на страницу www.rambler.ru. 3. Попытайтесь зайти на любой другой ресурс в интернет. 4. Посмотрите содержимое директории /usr/local/squid/logs/

Настройка аутентификации

auth_param схема параметр [опции]

Прокси-сервер squid позволяет осуществлять аутентификацию пользователей, причем сам squid не проверяет правильность аутентификации, он только запрашивает логин и пароль у пользователя. Для проверки полученной от пользователя информации он использует сторонние программы, которые будут ее осуществлять.

Для настройки аутентификации используется параметр auth_param.

```
auth_param схема параметр [опции]
```

Параметр «схема» определяет схему аутентификации: ntlm, digest или basic.

Следует учитывать, что Internet Explorer, если первой в списке определена схема basic, будет использовать только ее, независимо от того, поддерживаются ли proxy сервером другие схемы или нет. Поэтому порядок описания схем имеет значение.

После включения или выключения поддержки новой схемы proxy сервер рекомендуется остановить и запустить.

Мы рассмотрим создание самого простого варианта basic аутентификации с использованием программы ncsa_auth.

Для работы программы ncsa_auth необходимо создать файл с перечислением пользователей и их паролей. Пользователи, описываемые в этом файле, могут не совпадать с пользователями FreeBSD.

Файл /usr/local/etc/squid/passwd следует создавать и редактировать при помощи программы htpasswd. Эта программа поставляется с WEB сервером Apache.

```
htpasswd -c /usr/local/etc/squid/passwd user
```

Опцию -c следует указывать, только если файл /usr/local/etc/squid/passwd еще не существует.

User — это имя пользователя, которого Вы будете добавлять в этот файл или у которого Вы будете изменять пароль.


Доступ к файлу //usr/local/etc/squid/passwd должен быть ограничен.

После создания файла passwd, необходимо внести изменения в конфигурационный файл proxy сервера.

```
auth_param basic program /usr/local/libexec/squid/ncsa_auth /usr/local/etc/squid/passwd
```

Запись, показанная выше, должна быть создана в виде одной строки.

```
auth_param basic children 5
auth_param basic realm Squid proxy-caching web server
auth_param basic credentialsttl 2 hours
```

В первой строке определяется программа, которая будет использоваться в случае basic схемы. Указывается полный путь к программе, а также необходимые для ее запуска

параметры.

Во второй строке определяется количество одновременно запущенных программ аутентификации. Наличие нескольких запущенных программ необходимо для ускорения процесса аутентификации.

В третьей строке определяется имя, которое будет передаваться клиенту.

В четвертой строке определяется, как долго squid будет хранить информацию о логине и пароле для того, чтобы не обращаться за подтверждением к программе аутентификации.

Включение аутентификации

После настройки основных параметров аутентификации необходимо ее включить. Делается это при помощи параметров acl и http_access.

Сначала необходимо определить acl типа proxy_auth. Например:

```
acl passwd proxy_auth REQUIRED
acl any_auth proxy_auth anna bell thom
```

В первой строке определяется запрос аутентификации для всех пользователей. Во второй строке пользователи указаны явно.

Ограничить доступ по паролю к ресурсу можно при помощи параметра http_access. Например, необходимо ограничить доступ к медиафайлам в интернет, тогда следует определить два acl и один http_access.

```
acl auth proxy_auth REQUIRED
acl media urlpath_regex -i \.mpg$ \.avi$ \.mp3$
http_access allow media auth
http_access deny media
```

Обратите внимание на порядок acl в параметре http_access. Acl, описывающий необходимость аутентификации в списке, стоит последним. Это сделано для того, чтобы при разборе инструкций http_access не вызывался механизм аутентификации, если пользователь запрашивает не media-ресурсы. То есть. сначала происходит проверка первого acl и, если условие совпадает, только тогда будет происходить проверка второго acl.

Лабораторная работа Б.

Цель работы.
Научиться настраивать аутентификацию пользователя.

Задача лабораторной работы.
Заставить proxy сервер аутентифицировать пользователя при попытке зайти на ресурсы *.ya.ru и *.yandex.ru.

Задачи	***Описание***
1. Внесение изменений в конфигурационный файл squid.conf.	1. Откройте на редактирование файл squid.conf. 2. Найдите строки, где определяются параметры basic аутентификации и раскомментируйте их: **`auth_param basic children 5`** **`auth_param basic realm Squid proxy-caching web server`** **`auth_param basic credentialsttl 2 hours`** 3. Перед этими строками добавьте в виде одной строки следующее: **`auth_param basic program /usr/local/libexec/squid/ncsa_auth /usr/local/etc/squid/passwd`** 4. Перейдите к описанным Вами в предыдущей лабораторной работе acl. 5. В списке acl, но перед определением http_access, добавьте следующие acl: **`acl passwd proxy_auth REQUIRED`** **`acl ya dstdomain .ya.ru .yandex.ru`** 6. Между строками: **`http_access deny rambler`** **`http_access allow mynet wt`** вставьте следующие строки: **`http_access allow ya passwd`** **`http_access deny ya`** 7. Сохраните файл.
2. Создание файла со списком пользователей.	1. Выполните следующую команду: **`htpasswd -c /usr/local/etc/squid/passwd authuser`** 2. Введите пароль пользователя authuser. 3. Посмотрите содержимое файла **`/usr/local/etc/squid/passwd`**
3. Перезапуск proxy-сервера.	1. После добавления аутентификации, прокси-сервер необходимо заставить перечитать свои конфигурационные файлы: **`squid -k reconfigure`**
4. Проверка работоспособности.	1. Откройте WEB браузер и попытайтесь зайти на сайт www.ya.ru. Браузер должен попросить ввести логин и пароль. 2. Введите authuser и пароль, который был ему назначен. 3. В этом же браузере откройте страницу www.FreeBSD.org 4. Теперь откройте страницу www.yandex.ru. Запрос на аутентификацию не должен появиться. 5. Выключите браузер и запустите его снова.
5. Откройте страницу www.ya.ru.	Должно появиться приглашение авторизации.

Ограничение ширины канала

delay_pools delay_class delay_parameters delay_access

В прокси-сервер squid встроена возможность ограничения ширины канала. Причем ограничение накладывается на acl, то есть на ситуацию. Например, можно наложить ограничение на весь трафик с определенной машины или весь трафик определенного пользователя. А можно определить и более сложные случаи, например, ограничивать пользователя, только если он попытается скачать файлы определенного типа или пользоваться протоколом ftp.

Для включения ограничений используется группа параметров, организующая так называемые емкости задержки (delay pools).

В первую очередь при помощи параметра delay_pools определяется количество емкостей. Например, чтобы определить две емкости, необходимо написать так:

```
delay_pools 2
```

емкости бывают трех классов:

- Первого класса, предназначены для ограничения трафика всем acl, подключенным к емкости.
- Второго класса, определяют ограничения для сети класса C и отдельно для каждого acl.
- Третьего класса, определяют ограничения для сети класса B, затем отдельные ограничения для подсетей класса C и еще одно ограничение для каждого пользователя.

Для определения, к какому классу принадлежит емкость, используют параметр delay_class:

```
delay_class 1 1
delay_class 2 1
```

Первое число определяет номер емкости, второе — ее класс. После определения количества и класса емкостей, необходимо задать параметры ограничения. Это делается при помощи параметра delay_parameters.

Количество опций параметра delay_parameters зависит от класса емкости:

- емкость первого класса — один параметр.
- Второго класса — два параметра.
- Третьего класса — три параметра.

При определении ограничения необходимо указать два значения в одном параметре: скорость и объем емкости. Например, необходимо ограничить скорость скачивания для объектов размером от 64 Килобайт до 800 бит в секунду. Следует использовать следующую пару значений: 800/64000.

В случае определения ограничения для емкости первого класса, параметр delay_parameters будет записан следующим образом:

```
delay_parameters 1 800/64000
```

Где 1 — это номер емкости.

Если вместо 800/64000 написать 800/800 — общая скорость скачивания будет 800 бит в секунду.

Число -1 означает, что нет никаких ограничений.

Если определяются параметры для емкости 2-го класса, указываются две пары значений: для сети и для каждой машины. Например:

```
delay_parameters 1 64000/64000 4000/4000
```

Общее ограничение трафика для всей сети 64 Килобит в секунду, при этом ограничение для каждой машины 4 Килобита в секунду.

Когда определены все параметры емкостей, необходимо описать, какие acl будут к ним подключены. Это делается при помощи параметра delay_access. Например, нам необходимо ограничить скорость скачивания мультимедийных файлов до 400 Килобит в секунду.

Сначала определяем acl, описывающий все типы мультимедийных файлов:

```
acl media urlpath_regex -i \.mpg$ \.avi$ \.mp3$
```

Затем определяем емкость задержки.

```
delay_pools 1
delay_class 1 1
delay_parameters 1 400/400
delay_access 1 allow media
delay_access 1 deny all
```

При определении параметра delay_access обязательно указывается номер емкости, к которой подключаются acl.

Лабораторная работа В.

Цель работы.
Научиться настраивать ограничение ширины канала.

Задача лабораторной работы.
Ограничить скорость скачивания материалов с сайта www.odnoklassniki.ru до 40 бит в секунду.

Задачи	***Описание***
1. Внесение изменений в конфигурационный файл squid.conf.	1. Откройте на редактирование файл squid.conf. 2. Найдите строки, где определяются acl и добавьте следующую строку: **acl odnoklassniki dstdomain .odnoklassniki.ru** 3. В конце файла добавьте описание емкостей задержки: **delay_pools 1** **delay_class 1 1** **delay_parameters 1 40/40** **delay_access 1 allow odnoklassniki** **delay_access 1 deny all** 4. Сохраните файл. 5. Заставьте прокси-сервер перечитать свой конфигурационный файл: **`/usr/local/etc/rc.d/squid reload`**
2. Проверка работоспособности.	1. В браузере откройте сайт www.odnoklassniki.ru. Обратите внимание на скорость скачивания информации.

Вопросы

1. Перечислите основные возможности proxy сервера squid.

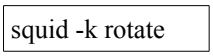
2. На что влияют параметры cache_swap_low и cache_swap_high?

3. В каком порядке необходимо определять параметры acl?

4. Влияет ли порядок определения параметров http_access на работу proxy сервера?

5. Как можно заставить proxy сервер squid перечитать свой конфигурационный файл?

Работа с журнальными файлами proxy-сервера Squid



При работе прокси сервер Squid ведет различные журнальные файлы. Среди них следует выделить файл /usr/local/squid/logs/access.log, в который попадает вся информация о том кто и какой ресурс пытается получить, был ли получен доступ к ресурсу или нет, и другая информация по использованию прокси сервера.

```
1081501287.175 64 127.0.0.1 TCP_MISS/200 1216 GET http://top.list.ru/counter? -
DIRECT/194.67.45.100 image/gif
```

На каждый запрошенный пользователем ресурс в этот файл помещается одна строка длинной около 60-70 байт (зависит от запроса). Из этого можно сделать вывод, что этот файл растет с очень большой скоростью.

Для ограничения роста файла (ротации) можно воспользоваться встроенной в squid возможностью.

```
squid -k rotate
```

Но сначала в конфигурационном файле squid.conf необходимо определить параметр logfile_rotate и указать количество хранимых файлов. Если этого не сделать, значение по умолчанию — 0, запрещает ротацию файлов.

После ротации появится файл, например, access.log.0.

Как часто производить процесс ротации, зависит от скорость роста файла access.log.

Для ротации файлов прокси-сервера не рекомендуется использовать программу logrotate. Лучшим вариантом считается применение CRON.

В директории /etc/periodic/daily необходимо создать скрипт следующего содержания:

```
#! /bin/sh
squid -k rotate
```

И не забыть сделать его исполняемым. Теперь раз в день будет происходить ротация журнальных файлов сервера.

Вопросы.

1. При помощи какого параметра определяется, где находится и как называется файл, в который будет помещена информация о клиентах и их запросах к proxy-серверу?

2. Какой параметр включает ротацию журнальных файлов?

3. Как часто необходимо производить ротацию журнальных файлов proxy-сервера?

Лабораторная работа Г.

Цель работы.
Научиться настраивать squid в режиме transparent proxy.

Задача лабораторной работы.
Включить прозрачное проксирование и перенаправление портов.

Задачи	*Описание*
1. Внесение изменений в конфигурационный файл squid.conf.	1. Откройте на редактирование файл squid.conf. 2. Найдите строку: **http_port 3128** **и скорректируйте ее** **http_port 3128 transparent** 3. В конце файла /etc/pf.conf добавьте это правило одной строкой: **rdr proto tcp from any to any port 80 -> 127.0.0.1 port 3128** 4. Разрешите запуск pf в /etc/rc.conf **pf_enable=yes** **pf_rules="/etc/pf.conf"** **pf_program="/sbin/pfctl"** **pf_flags=""** **pflog_enable="yes"** **pflog_logfile="/var/log/pflog"** **pflog_program="/sbin/pflogd"** **pflog_flags=""** 5. Запустите pf **/etc/rc.d/pf start** 6. Заставьте прокси-сервер перечитать свой конфигурационный файл: **/usr/local/etc/rc.d/squid reload**
2. Проверка работоспособности.	На клиенте WinXP: 1. Смените маршрут по умолчанию на адрес шлюза, а DNS на DNS шлюза 2.В браузере укажите «direct access» 3.В браузере откройте сайт www.rambler.ru.