Squid + ncsa аутентификация

Вступление

На днях настраивал в одном из офисов связку squid + squidGuard. Но в этот раз уже без поддержки АД, так как офис небольшой - 5 машин, то и необходимости в АД нет. Но как же быть в этом случае с аутентификацией пользователей? В этом случае на помощь приходит ncsa аутентификация и аутентификатор ncsa_auth.

При данной схеме имя пользователя и пароль хранятся в обычном текстовом файле, имеющем стиль файла паролей apache, который используется при базовой HTTP аутентификации.

Установка и настройка squid

Итак, как установить данные программы я уже рассказывал в предыдущих статьях - «SquidGuard - борьба с нарушителями», Squid и авторизация пользователей в AD«. Поэтому буду исходить из того, что squid и squidGuard уже установлены. Думаю, понятно что для работы данной схему устанавливать samba нет необходимости.

Производим минимальную настройку squid. Все изменения необходимо вводить после соответствующих тегов.

```
#
#   /etc/squid/squid.conf
#

#  TAG: http_port
# Указываем squid на каком порту и интерфейсе он будет работать. Именно эти
# параметры необходимо будет указывать в настройках интернет проводника.
http_port 192.168.0.1:3128

#  TAG: maximum_object_size_in_memory   (bytes)
# Объекты больше этого размера не будут сохраняться в памяти.
# По умолчанию 8КБ что очень мало на текущий момент, т.к. средний объем
# web странички ~75-100 КБ
maximum_object_size_in_memory 102400

#  TAG: cache_dir
# Объем кеша и его месторасположение. Объем задается в мегабайтах. (4096 ~ 4Гб)
cache_dir ufs /var/spool/squid/ 4096 16 256

#  TAG: dns_nameservers
# Здесь необходимо указать днс сервер(а). В принципе если ничего не указывать,
# то squid автоматически добавит сервер(а), которые указаны в /etc/resolv.conf
# Я указал адрес самого сервера, т.к. у меня на нем работает кеширующий днс.
# Если у вас нет своего днс сервера, то необходимо указывать днс провайдера.
```

```
dns_nameservers 192.168.0.1

#  TAG: redirect_program
# Указываем путь к squidGuard
redirect_program /usr/bin/squidGuard

#  TAG: redirect_children
# Количество одновременно запускаемых процессов squidGuard.
redirect_children 5

# TAG: auth_param
# Здесь мы указываем squid, как следует производить аутентификацию.
# Единственный параметр, который необходимо указать ncsa_auth -
# это имя файла, в котором хранятся имена пользователей и пароли.
# Данным файлом можно управлять (добавлять новых пользователей,
# удалять текущих, изменять пароли и т.д.) с помощью утилиты
# htpasswd, которая входит в состав пакета apache.
auth_param basic program /usr/lib/squid/ncsa_auth /etc/squid/internet_users
auth_param basic children 5
auth_param basic realm Squid proxy-caching web server
auth_param basic credentialsttl 2 hours
auth_param basic casesensitive off

# TAG: acl
# ACL - Access Control List. Списки доступа к нашему прокси серверу.
# Здесь мы указываем, кто имеет право, а кто нет, использовать наш прокси.
# Разрешаем использовать наш прокси только прошедшим авторизацию.
acl DARIM proxy_auth REQUIRED
http_access allow DARIM
http_access deny all

#  TAG: cache_effective_user
# Пользователь и группа, от которых работает squid
cache_effective_user squid

#  TAG: cache_effective_group
cache_effective_group squid

#  TAG: visible_hostname
# Данное имя будет указываться в различных сообщениях (об ошибках и т.п.)
# По умолчанию будет подставляться значение, возвращаемое функцией gethostname()
visible_hostname darim.proxy.server
```

В ветке 2.6.STABLE произошли изменения и теперь вместо параметров redirect_program и redirect_children необходимо использовать url_rewrite_program и url_rewrite_children соответственно.

На этом настойку squid можно считать завершенной.

Я думаю, вы заметили, какие пути я использовал в конфигурационных файлах, так сказать нехарактерные для FreeBSD. Это связанно с тем, что данную связку я настраивал на линуксе, а

именно CentOS-4.4. Так что если вы используете FreeBSD, то пути у вас будут другие.

Теперь нам осталось создать файл с паролями. Для этого необходимо выполнить следующую команду

```
# htpasswd -c /etc/squid/internet_users test
New password: *******
Re-type new password: *******
Adding password for user test
```

Ключ -c необходимо указывать только один раз, при создании файла с паролями. В последующем добавлять пользователей нужно без этого ключа. Итак, мы получили файл с примерно таким содержимым.

```
# cat /etc/squid/internet_users
test:89XzEEI/P0e56
```

Где test - имя пользователя, а 89XzEEI/P0e56 - хеш его пароля. Для добавления нового пользователя в существующий файл необходимо выполнить следующую команду

```
# htpasswd -b /etc/squid/internet_users admin 1234567
Adding password for user admin

# cat /etc/squid/internet_users
test:89XzEEI/P0e56
admin:EAJezc5eLXrV2
```

Думаю, что никаких проблем при использовании htpasswd возникнуть не должно. Более подробную информацию по использованию данной программы, а также по всем ключам можно получить как всегда прочитав мануал - man htpasswd.

После того как мы создали файл и пользователей не забываем выставить соответствующие права на файл /etc/squid/internet_users.

```
# chmod 440 /etc/squid/internet_users
# chown squid:squid /etc/squid/internet_users
```

Установка и настройка squidGuard

Настройка squidGuard практически аналогична, за исключением того, что вместо ip адресов мы будем использовать имена пользователей.

```
#
#   /etc/squid/squidGuard.conf
#

# Указываем где у нас располагаются наши базы
dbhome /var/lib/squidGuard
```

```
# Указываем где будет храниться log файл squidGuard
logdir /var/log/squid/

# Описываем классы доступа. Затем мы для каждого из классов
# (admins, clients) создадим свой набор правил доступа.
src admins {
user admin
}

src clients {
user test
}
# Думаю понятно, что test и admin имена пользователей,
# которых мы создавали с помощью htpasswd

# Далее описываем все категории запретов
dest ads {
domainlist       ads/domains
expressionlist   ads/expressions
urllist          ads/urls
redirect         http://192.168.0.1/block/ads.html
}

dest adult {
domainlist       adult/domains
urllist          adult/urls
redirect         http://192.168.0.1/block/adult.html
}

...
...
...

dest webmail {
domainlist       webmail/domains
urllist          webmail/urls
redirect         http://192.168.0.1/block/webmail.html
}

dest whitelist {
domainlist       whitelist/domains
urllist          whitelist/urls
redirect         http://192.168.0.1/block/whitelist.html
}

# Ну а теперь непосредственно делаем раздачу слонов.
acl {
clients  {
            pass !ads !adult ... !porn !warez all
         }
```

```
# Себе разрешаем все, кроме баннеров
admins   {
            pass !banner all
         }

# И наконец, задаем правило по умолчанию - всем все запрещаем.
default  {
            pass     none
            redirect http://192.168.0.1/block/default.html
         }
}
```

Тестирование

Теперь наша связка готова и можно преступить к ее тестированию. Для этого запускаем squid.

```
# service squid start
Starting squid: .                                          [  OK  ]

# cat /var/log/squid/cache.log | grep helper
2006/10/25 16:50:11| helperOpenServers: Starting 5 'squidGuard' processes
2006/10/25 16:50:11| helperOpenServers: Starting 5 'ncsa_auth' processes
```

В любимом интеренет проводнике открываем какую-нибудь страничку, при этом в логах должно быть следующее

```
1161784565.905     23 192.168.0.100 TCP_MISS/200 4765
GET http://www.sys-adm.org.ua/test.php admin DIRECT/www.sys-adm.org.ua
text/html
```

Ну вот собственно и вся настройка. Теперь осталось лишь настроить какой-нибудь анализатор логов squid (sarg, lightsquid) и радоваться жизни. Блокировка ip адресов

Допустим, вы закрыли пользователю test доступ на сайт www.sys-adm.org.ua. Т.е. добавили этот сайт в файл domains.diff и произвели обновление БД. Но этого недостаточно, если пользователь немного разбирается в основах интернет технологий, то он сможет узнать какой адрес соответствует имени www.sys-adm.org.ua и заходить на этот сайт уже по ip адресу.

Для исключения такой возможности в squidGuard существует встроенное регулярное выражение in-addr с таким содержимым:

```
^[^:/]+://[0-9]\{1,3\}\.[0-9]\{1,3\}\.[0-9]\{1,3\}\.[0-9]\{1,3\}($|[:/])
```

Т.е. если вы напишете такое правило

```
#
#   /etc/squid/squidGuard.conf
#
```

```
src clients {
user test
}

...
...
...

acl {
clients  {
           pass !in-addr !ads !adult ... !porn !warez all
         }
...
...
...

}
```

При таком правиле пользователь уже не сможет использовать ip адреса вообще. Поиск адреса в БД

Для того чтобы было удобно искать адрес в файлах domains.db и urls.db можно воспользоваться следующим скриптом.

```
#!/usr/bin/perl
use strict;
use DB_File;

foreach (@ARGV) {
  my (%db, $key, $val);
  die("$_: $!\n") unless(-f);
  tie(%db, "DB_File", $_, O_RDONLY, 0664, $DB_BTREE) || die("$_: $!\n");
  foreach $key (keys(%db)) {
    if($val = $db{$key}) {
      $val = "\"$val\"";
    } else {
      $val = "undef";
    }
    print "$key\n";
  }
  untie(%db);
}
```

Пользоваться им очень легко, необходимо лишь передать путь к файлу с данными

```
# ./search.pl /var/lib/squidGuard/good/domains.db
.mail.ru
.google.com
.private.com
```

Если вам надо найти какой то определенный сайт, то можно воспользоваться командой grep

для фильтрации результатов.

```
# ./search.pl /var/lib/squidGuard/good/domains.db | grep mail.ru
.mail.ru
```

Экономим трафик и ресурсы

Исследуя отчеты, созданные с помощью sarg или подобной программой, вы заметили, что очень много пользователей закачивают одни и те же файлы (причем если они большие по объему, то они не будут сохраняться в кеше), ну например - install_flash_player_8.msi

Для того чтобы сэкономить трафик и разгрузить канал можно отдавать такие файлы с локального сервера (http или ftp). Для этого необходимо использовать опцию rew | rewrite.

```
#
#   /etc/squid/squidGuard.conf
#

src clients {
user test
}

rew local-downloads {
  s@.*/flash_player_8.msi$@http://localhost/downloads/flash_player_8.msi@r
}

...
...
...

acl {
        clients {
                pass good !in-addr none
                rewrite local-downloads
        }

        default {
                pass     none
                redirect 302:http://localhost/default.html
        }
}
```

Теперь все пользователи из группы clients при попытке закачать файл install_flash_player_8.msi с любого сайта, будут получать копию с локального веб сервера. Естественно, чтобы работало это правило необходимо создать соответствующую папку на веб сервере (downloads) и положить в нее необходимый файл.

```
#
#   /etc/httpd/conf/httpd.conf
```

```
#

<Directory "/var/www/html/downloads">
    Options -Indexes
    AllowOverride None
    Order allow,deny
    Allow from localhost
</Directory>
```

```
# mkdir /var/www/html/downloads
# chmod 550 /var/www/html/downloads/
# chown apache:apache /var/www/html/downloads/
```

При такой конфигурации скачать этот файл можно будет только с localhost. При попытке скачать файл с любой другой машины в логах будут подобные записи.

```
[Wed Oct 25 20:57:18 2006] [error] [client 192.168.0.100] client denied by
server configuration: /var/www/html/downloads/flash_player_8.msi
```

Контроль доступа по времени

Также еще одной интересной возможностью squidGuard является ограничение доступа по времени. Например, вам необходимо сделать так, чтобы для группы clients интернет был доступен с 8:00 до 18:00 с понедельника по пятницу, т.е. в течение рабочего дня. Это легко реализуется средствами squidGuard.

```
#
#   /etc/squid/squidGuard.conf
#

time work-hours {
    weekly mtwhf 08:00-18:00
}

src clients {
user test
}

...
...
...

acl {
        clients within work-hours {
                pass good !in-addr none
        }

        default {
                pass     none
```

```
                redirect 302:http://localhost/default.html
        }
}
```

При таких правилах пользоваться интернетом можно будет с понедельника по пятницу с 8 утра и до 18 вечера. Учтите, что при блокировке по времени правило попадает под категорию default. Поэтому было бы логично указать на страничке default.html ограничения по времени, связанные с доступом в интернет.

Ну вот вроде и все, чем я хотел поделиться и о чем хотел рассказать. Настоятельно рекомендую прочитать документацию, которая идет вместе с squidGuard, в ней очень много интересного.

P.S. В следущей статье я расскажу как настроить анализатор логов lightsquid, для получения отчетов о использовании трафика.

From:
http://sys-adm.org.ua/ - **wiki.sys-adm.org.ua**

Permanent link:
http://sys-adm.org.ua/www/squid-ncsa

Last update: **2009/09/04 20:46**