Введение

Цель данной статьи рассказать и показать, как можно настроить систему фильтрации контента для пользователей, работающих в доменной структуре MS Windows. Возможности данной системы должны позволять делить пользователей на группы и, в зависимости от принадлежности, давать доступ только в рамках своей группы фильтрации. Так же должна происходить проверка на вирусы, а в случае скачивания файлов должно появиться информационное окошко, которое покажет сколько скачалось,сколько осталось и общий размер.

Задача

В качестве некоторой тестовой задачи хотелось получить функционирование системы фильтрации контента, в связке с антивирусом, с несколькими группами фильтрации на основе их аккаунтов. Для этой задачи было принято решение использовать следующий набор ПО:

1. Squid - как кеширующий прокси сервер
2. Dansguardian - как система фильтрации контента
3. Clamav - как антивирус
4. Samba - как прослойка между MS AD и Unix/Linux
5. Active Directory - как единое хранилище пользовательских данных

В качестве операционной системы был выбран RedHat Enterprise Linux 5. На вопрос почему я отвечу очень просто: все что было под рукой. Если раскрывать более детально, то RHEL 5 имеет более новое ядро, а так же важную функциональность **CPUSET**, благодаря которой у нас появилась возможность создавать защищённую среду для работы процессов. Защищённость в том смысле, что пользователь может задавать на каких процессорах исполнять тот или иной процесс и его потомков. После таких ограничений процесс не сможет перейти на свободный процессор, что дает нам возможность корректнее распределять нагрузку на процессоры, а так же даст возможность ограничить особо прожорливые процессы. Например в нашей конфигурации можно сделать так: 1 процессор для SQUID, 1 для DansGuardian, 1 для Clamav, 1 для самбы. В итоге у каждой службы будет своё процессор со своей нагрузкой (не считая системную). Описание процесса настройки CPUSET выходит за рамки данной статьи и будет описано позже (если найдется время ).

Задача формулировалась так:

- Система должна кешировать сайты
- Система должна фильтровать содержимое сайтов и запрещать доступ
- Система должна иметь несколько групп фильтрации и быть легкоуправляемой
- Система должна производить проверку на наличие вредоносного кода как при скачивании файлов, так и при просмотре странниц (вредные javascripts )
- Система должна интегрироваться в единый каталог пользователей
- Система должна визуально показывать пользователю, что идёт процесс скачивания и проверки и производить информирование по результату выполнения.

Для решения данной задачи оптимально - разбиваем её на несколько этапов.

Этап 1

На данном этапе будет произведена установка ОС и всех базовых служб, таких как:

- squid 2.6
- samba
- kerberos

Т.к. все эти службы входят в дистрибутив, то процесс установки будет банальным `rpm -ivh` <Нужные пакеты>

Этап 2

На этом этапе мы остановимся подробнее. Здесь будет описан процесс конфигурирования системных служб и проверка настроек.

SAMBA

Настройка самбы тривиальна. Базовый конфиг */etc/samba/smb.conf*, в котором исправляются нужные вам опции

```
  workgroup = RESEARCH
```

называем рабочую группу RESEARCH

```
  server string = server
```

имя сервера server

```
  security = ads
```

режим безопасности ads - т.е. Active Directory, т.к. у нас основная задача интеграция именно с ним.

```
  password server = x.x.x.x
```

указываем IP адресс контролера домена

```
  realm = RESEARCH.LAN
```

указываем полное наименование MS Windows домена. В нашем случае это RESEARCH.LAN. Важная особенность - полное наименование должно быть большими буквами.

```
  winbind cache time = 10
```

указываем период обновления кеш данных. Необходимо, если мы не хотим каждый раз при

изменении в АД перезапускать службу winbind

```
encrypt passwords = yes
winbind use default domain = no
winbind uid = 10000-20000
winbind gid = 10000-20000
winbind enum users = yes
winbind enum groups = yes
```

опции необходимые для маппинга доменных пользователей локально, установка локальных ID пользователей и групп и т.п. Теперь добавляем в автозапуск службу, чтобы при перезагрузки она стартовала автоматически:

```
# chkconfig winbind on
```

Убеждаемся в том, что служба занесена в автозагрузку

```
# chkconfig --list winbind
 winbind         0:off   1:off   2:on    3:on    4:on    5:on    6:off
```

Как мы видим на уровнях 2, 3, 4, 5 служба прописана. Производим запуск службы

```
# service smb start
# service winbind start
```

На этом вся настройка самбы может быть смело закончена. Теперь можно переходить к настройке kerberos и вводу машины в домен.

Kerberos

Надо отредактировать файл */etc/krb5.conf*. Выглядить он будет примерно так:

```
[logging]
 default = FILE:/var/log/krb5libs.log
 kdc = FILE:/var/log/krb5kdc.log
 admin_server = FILE:/var/log/kadmind.log
```

```
[libdefaults]
 default_realm =RESEARCH.LAN
 dns_lookup_realm = false
 dns_lookup_kdc = false
 ticket_lifetime = 24h
 forwardable = yes
```

```
[realms]
 RESEARCH.LAN = {
  kdc = x.x.x.x:88
  admin_server = x.x.x.x:749
  default_domain = research.lan
```

```
 }
```

```
[domain_realm]
 .research.lan = RESEARCH.LAN
 research.lan = RESEARCH.LAN
```

```
[kdc]
 profile = /var/kerberos/krb5kdc/kdc.conf
[appdefaults]
 pam = {
   debug = false
   ticket_lifetime = 36000
   renew_lifetime = 36000
   forwardable = true
   krb4_convert = false
 }
```

где адрес x.x.x.x совпадает с адресом контроллера домена, который указывали в конфигурационном файле самбы.

Важное замечание:
Необходимо соблюдать регистр, т.е. там где прописаны большими буквами домены они должны быть большими. Это требование `kerberos.`

Синхронизация времени

Чтобы избежать проблем с kerberos необходимо, чтобы время на контроллере домена и на машине было одинаковым. Допускается разница в 5 минут, но даже это крайне не желательно. Для начала мы просто синхронизируем время:

```
# ntpdate -u x.x.x.x
```

где x.x.x.x адрес контроллера домена. После чего у вас должен появиться такой ответ:

```
22 Aug 19:48:50 ntpdate[30264]: step time server x.x.x.x offset 4.082098
sec
```

Чтобы время синхронизировалось вне зависимости от вас, надо в файле */etc/ntp.conf* прописать строку

```
server x.x.x.x
```

Теперь добавляем в автозапуск службу, чтобы при перезагрузки она стартовала автоматически:

```
# chkconfig ntpd on
```

Убеждаемся в том, что служба занесена в автозагрузку

```
# chkconfig --list ntpd
 ntpd        0:off   1:off   2:on    3:on    4:on    5:on    6:off
```

Как мы видим на уровнях 2, 3, 4, 5 служба прописана. Производим запуск службы:

```
# service ntpd start
```

Вводим в домен

Для правильной работы нам потребуется ввести машину в домен. Это делается тривиальным способом:

```
# net ads join -U admin%admin123
```

где запись «admin%admin123» обозначает ни что иное как логин%пароль. Проверяем в домене ли мы:

```
# wbinfo -u
```

Данная команда выведет список всех пользователей домена следующего вида:

```
RESEARCH\administrator
RESEARCH\guest
```
и т.п.

Теперь необходимо, чтобы система видела пользователей и группы доменной системы. Для этого редактируем файл */etc/nsswitch.conf* следующим образом:

```
passwd:     files winbind
shadow:     files winbind
group:      files winbind
```

И проверяем правда ли мы видим, или есть ошибки.

```
# getent group
# getent passwd
```

Если все сделано правильно и без ошибок, то мы получим такие ответы

```
# getent group
```

```
root:x:0:root
bin:x:1:root,bin,daemon
daemon:x:2:root,bin,daemon
  ......
RESEARCH\domain computers:*:10010:
RESEARCH\domain controllers:*:10011:
RESEARCH\schema admins:*:10012:
```

```
и т.п.
```

```
  # getent passwd
```

```
  root:x:0:0:root:/root:/bin/bash
  bin:x:1:1:bin:/bin:/sbin/nologin
  daemon:x:2:2:daemon:/sbin:/sbin/nologin
   ......
  RESEARCH\backup:*:10028:10000:backup:/home/RESEARCH/backup:/bin/false
  RESEARCH\dfs:*:10029:10000:dfs:/home/RESEARCH/dfs:/bin/false
и т.п.
```

Мы убедились что у нас в системе отражаются пользователи. Теперь надо задать пользователя от которого будет происходить аутентификация:

```
  # wbinfo --set-auth-user=RESEARCH\\admin%admin123
  # wbinfo --get-auth-user
```

```
  RESEARCH\admin%admin123
```

Теперь переходим к настройке кеширующего прокси сервера.

Squid

В дистрибутиве уже есть squid версии 2.6 который нам подойдет. Все что нам надо - это произвести его настройку, для этого в файле */etc/squid/squid.conf* редактируем следующие опции:

```
  http_port 127.0.0.1:3128
```

опция задает сетевой адрес и порт, на котором сквид будет принимать соединения. На не нужно, чтобы к сквиду обращались напрямую, поэтому вешаем его на петлевой интерфейс

```
  auth_param ntlm  program /usr/bin/ntlm_auth --helper-protocol=squid-2.5-
ntlmssp
  auth_param ntlm children 5
  auth_param ntlm keep_alive on
  auth_param basic  program /usr/bin/ntlm_auth --helper-protocol=squid-2.5-
basic
  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
```

Заставляем сквид авторизовывать по NTLM, если браузер не поддерживает этого, то используем basic авторизацию. Здесь важна последовательность, поэтому другого варианта быть не может, иначе будут проблемы в работе.

```
  acl RESEARCH proxy_auth REQUIRED
```

Создаем список доступа, согласно которому все, кто подпадает под него должны проходить авторизацию в обязательном порядке.

```
  http_access allow RESEARCH localhost
```

Говорим сквиду, что все, кто идёт с адреса 127.0.0.1 обязаны пройти авторизацию.

Остальные опции настраивайте по вашему усмотрению. Документации на просторах интернета много, поэтому не вижу смысла описывать в очередной раз. Теперь добавляем в автозапуск службу, чтобы при перезагрузки она стартовала автоматически:

```
  # chkconfig squid on
```

Убеждаемся в том, что служба занесена в автозагрузку

```
  # chkconfig --list squid
   squid    0:off   1:off   2:on    3:on    4:on    5:on    6:off
```

Как мы видим на уровнях 2, 3, 4, 5 служба прописана. Производим запуск службы:

```
  # service squid start
```

На этом этап настройки завершён. Переходим к этапу настройки антивируса и системы контроля за контентом.

Этап 3

На данном этапе на понадобиться антивирус clamav и система контроля за контентом dansguardian.

Скачиваем с сайта www.clamav.net последнюю версию антивируса (на момент публикации версия была 0.91.1)

Скачиваем с сайта www.dansguardian.org версию 2.9.9.0

Скачиваем патч для dansguardian (позже объясню для чего) отсюда: ntlm_group.patch.gz

Clamav

Распаковываем скачанный файл

```
  $ tar xzvf clamav-0.91.1.tar.gz
```

Собираем антивирус со следующими опциями

```
  $ ./configure --prefix=/opt --with-user=squid --with-group=squid --with-
tcpwrappers--enable-dns-fix
```

Описание этих опций можно получить по команде

```
$ ./configure --help
```

Далее производим компиляцию и установку

```
$ make
# make install
```

Теперь добавим в */etc/ld.so.conf.d/clamav.conf* следующую строчку

```
/opt/clamav/lib
```

Добавим в */etc/profile.d/clamav.sh* следующее

```
PATH=/opt/clamav/bin:/opt/clamav/sbin:$PATH
LD_LIBRARY_PATH=/opt/clamav/lib:$LD_LIBRARY_PATH
```

Обновим кеш библиотек

```
# ldconfig -n /opt/clamav/lib
```

Обновим переменные окружения, которые мы задали, чтобы увидеть исполняемые файлы clamav

```
# source /etc/profile
```

Есть второй и более правильный вариант - использовать rpm репозитарии. Чтобы задействовать репозитарий надо скачать пакет *rpmforge-release*, в котором уже есть необходимые конфигурационные файлы и GPG ключи. И так качаем его с сайта

```
http://dag.wieers.com/rpm/packages/rpmforge-release/
```

Берем самый последний, который соответствует нашей операционной системе. Для RHEL 5 был последним пакет

```
http://dag.wieers.com/rpm/packages/rpmforge-release/rpmforge-release-0.3.6-1
.el5.rf.i386.rpm
```

После того, как скачивание будет завершено произведём его установку

```
# rpm -ivh rpmforge-release-0.3.6-1.el5.rf.i386.rpm
```

Теперь у нас подключен репозитарий, с которого будет производиться так же и обновление. Чтобы установить clamav последней версии воспользуемся стандартными средствами RHEL для управления пакетами, а именно yum

```
# yum update
```

произведём обновление пакетов, если есть более новые версии.

```
# yum install clamav
```

после этого у вас в системе появится clamav последней версии, а так же будет обновляться по мере необходимости. Этот вариант более предпочтителен, т.к. вам не надо заботиться об обновлении антивируса, система сделает это за вас. Установка завершена. Теперь осталось сконфигурировать антивирус и обновить базы. Описывать эти процедуры не буду, т.к. есть документация.

DansGuardian

Что такое DansGuardian? DG - это система фильтрации контента, в отличии от стандартных редиректоров сквида она умеет фильтровать не только по URL сайтов, но и по их содержимому, проставляя баллы и на основе набранных баллов разрешать или запрещать тот или иной сайт. DG умеет работать с несколькими группами фильтрации, что позволяет администратору давать гибко настраивать систему и разрешать/запрещать те или иные сайты для групп. Что не мало важно - DG умеет работать с антивирусами, что позволяет не только фильтровать но и проверять содержимое на наличии вирусов. Поддерживаются следующие виды антивирусов: Kaspersky и Clamav, остальные антивирусы могут работать либо через ICAP протокол, либо через командную строку. Для clamav есть следующие возможности:

```
libclamav
```

Данная возможность позволяет DG сканировать данные, используюя библиотеку антивируса. Здесь есть свои плюсы и минусы. К плюсам можно отнести полную интеграцию антивируса, сканирование будет производиться самостоятельно, без запуска демона clamd. Я рекомендую данную возможность на системах с 1 процессором. Минусы такого подхода в том, что DG тратит своё время на проверку вирусов, вместо того, чтобы отдать демону clamd контент и заниматься своим делом, ожидая ответа вида Чист/Заражён.

```
clamd
```

Данная возможность позволяет DG отдавать данные демону clamd и не заниматься самовыявлением. Такой подход хорош, когда у нас многопроцессорная машина и, в купе с CPUSET, мы можем отдать тяжёлую работу отдельному демону, который будет замаплен на процессор и не будет грузить остальные.

```
commanline
```

Самый тяжёлый метод - метод когда DG передает утилите сканирования, как параметр, файл с контентом.

Зачем нужен патч? Дело в том, что при использовании NTLM протокола DG удостоверится, что пользователю разрешён доступ и далее будет искать его в своих группах фильтрации. Проще говоря, нам надо будет ручками прописывать каждого пользователя к группе фильтрации, что может быть не эффективно, а если пользователей более 10 ещё и обременительно. Патч исправляет этот недочет. Благодаря ему, мы имеем возможность создавать системные группы вида

```
dgfilterX
```

где X - номер группы от 1 до 99. Теперь нам надо всего лишь задействовать механизм nsswitch для подключения нужных нам каталогов и создать в каталогах группы. Члены группы автоматически будут подключены к той или иной группе фильтрации, нам останется лишь создать необходимое количество конфигурационных файлов вида

```
  danguardianfX.conf
```

где X - номер группы, там же поправить опции

```
   groupmode = X
```

где X - номер политики

```
  0 = banned
  1 = filtered
  2 = unfiltered (exception)
```

и назвать группу (что не обязательно)

```
  groupname = 'XXXX'
```

где XXX - запись любого вида. Цель её просто назвать группу, ну например Full Access, WWW pages only и т.п. После этого в том же Active Directory мы просто добавляем пользователя в группу.

Важно знать:
1. Если пользователь в нескольких группах, то система автоматически направит его в группу по умолчанию, т.к. пользователь не может быть в 2-х и более группах одновременно
2. При создании группы в `AD` надо выбирать параметры `Group scope: "Global"` и `Group Type: "Security"`

Переходим непосредственно к установке. Распаковываем скачанный файл

```
  $ tar xzvf dansguardian-2.9.9.0.tar.gz
```

Накладываем патч

```
  $ zcat ../ntlm_group.patch.gz | patch -Np1
```

Производим конфигрурирование

```
 $ ./configure --prefix=/opt/dg --with-proxyuser=squid --with-
proxygroup=squid --with-zlib --enable-ntlm --enable-trickledm --enable-
fancydm --enable-commandline --enable-icap --enable-kavd  --enable-clamd  --
enable-clamav --enable-pcre
```

Опции

```
  '--enable-ntlm'
```

включить поддержку NTLM

```
'--enable-trickled'
```

включить поддержку менеджера докачки. Данный менеджер позволяет постепенно отдавать контент, чтобы пользователь видел его статус, по окочанию скачивани и проверки будет принято решение отдавать до конца или заблокировать.

```
'--enable-fancydm'
```

включить поддерку менеджера докачки. Данный менеджер будет выводить странницу со статусом скачивания файла и результатом проверки. В случае, если будет найден вирус, то появится сообщение в браузере об этом. Важно знать, что на сайтах, где закачка подключается через javascript (например sourceforge.net) данная странница не появится, поэтому надо будет переходить по реальной ссылке. При этом закачка не будет останавливаться и при переходе вы получите текущий статус скачивания.

```
'--enable-commandline'
'--enable-icap'
'--enable-clamd'
'--enable-clamav'
'--enable-kavd'
```

режимы проверки на вредоносность данных

```
'--enable-pcre'
```

включить поддержку библиотеки регулярных выражений. Далее производим сборку и установку

```
$ make
# make install
```

После этого производится конфигурация самого DG в файле *<PATH_TO_INSTALL>/etc/dansguardina/dansguardian.conf* Файл хорошо документирован, поэтому вдаваться в детали каждой опции не буду, опишу только те, на которые стоит обратить внимание:

```
language = 'russian-1251'
```

странница с сообщение о блокировке будет выдаваться на русском языке. Шаблоны лежат в директории

```
languagedir = '/opt/dg/share/dansguardian/languages'
```

можете перенести куда вам будет удобнее и исправить тут путь.

```
filterip = 10.0.60.150
filterport = 8080
```

IP адресс и порт, по которым DG будет работать.

```
proxyip = 127.0.0.1
proxyport = 3128
```

адрес прокси сервера, в нашем случае сквида. DG не умеет работать отдельно, только в связке.

```
filtergroups = 2
```

здесь необходимо прописывать количество групп фильтрации. Т.е. у вас в системе все группы должны идти последовательно и количество этих групп должно быть тут указано. Например у вас есть группы dgfilter1, dgfilter2 и dgfilter3 в итоге количество групп у нас будет 3.

```
deletedownloadedtempfiles = on
```

удалять файлы, после того как будут закачены.

```
downloadmanager = '/opt/dg/etc/downloadmanagers/fancy.conf'
downloadmanager = '/opt/dg/etc/downloadmanagers/default.conf'
```

тут указывается необходимый вам менеджер докачки, которые я описал выше

```
contentscanner = '/opt/dg/etc/contentscanners/clamav.conf'
```

указываем, что хотим использовать libclamav

```
authplugin = '/opt/dg/etc/authplugins/proxy-basic.conf'
authplugin = '/opt/dg/etc/authplugins/proxy-ntlm.conf'
```

указываем что хотим использовать авторизацию по NTLM и BASIC. На самом деле авторизация будет проходить на сквиде, а не на DG. Поэтому если у вас сквид не настроен, то работать DG не будет.

```
forwardedfor = off
usexforwardedfor = off
```

опции необходимы, если вы хотите чтобы в логах сквида отображался IP адрес клиента, вместо 127.0.0.1

Вот собственно и все. Конфигурация закончена. Теперь приведу некоторое объяснение как создавать группы фильтрации. Если вы обратите внимание, то рядом с конфигурационным файлом лежит файл dansguardianf1.conf, который есть ни что иное, как файл конфигурации группы фильтрации по умолчанию. Чтобы создать файлы для других групп надо просто скопировать заменив суффикс на соответствующий, т.е.:

```
dgfilter1, default - dansguardianf1.conf
dgfilter2 - dansguardianf2.conf
dgfilter3 - dansguardianf3.conf
```

Далее копируем директорию с шаблонами проверки *lists* и создаем нужную вам структуру шаблонов (1 группа фильтрации использует директорию *lists* поэтому можно начать со второй), например так:

```
  groups
  groups/fgroup2
  groups/fgroup3
```

И соответсвенно заменяем в файлах конфигурации пути до списков согласно вашей структуре. Так же в этих файлах вы можете отключать для групп фильтрацию всего контента, делать группы, которым будет заблокирован доступ вообще и т.п.

Теперь добавляем в автозапуск службу, чтобы при перезагрузки она стартовала автоматически, для этого скопируем из директории *<PATH_TO_INSTALL>/share/dansguardian/scripts/systemv-init* в директорию */etc/init.d/*

```
  # cp <PATH_TO_INSTALL>/share/dansguardian/scripts/systemv-init
/etc/init.d/dansguardian
  # chkconfig --add dansguardian
  # chkconfig dansguardian on
```

Убеждаемся в том, что служба занесена в автозагрузку

```
  # chkconfig --list dansguardian
   ntpd        0:off   1:off   2:on    3:on    4:on    5:on    6:off
```

Как мы видим на уровнях 2, 3, 4, 5 служба прописана. Производим запуск службы:

```
  # service dansguardian start
```

Важно знать:
При первом запуске DG будет ругаться на отсутствие файлов, просто досоздайте их по тому же пути, на который он ругается при помощи команды ''touch''.

Вот и все, остальное дело ваше. Играйтесь, настраивайте, отсылайте ошибки разработчику. Эксперементируйте и/или наслаждайтесь.

Я постараюсь в ближайшее время отправить автору этот патч, может быть его включат в официальную ветку и тогда задача упростится.

Успехов в ваших начинаниях 😀

Ресурсы

Ресурсы, которые помогли мне в работе:

www.samba.org

www.squid-cache.org

www.sysadmins.ru

www.sys-adm.org.ua

www.google.com

документация в комплекте с исходными кодами.

С уважением NoFate.