В связи с развитием доступа в интернет сейчас стали не так популярны прокси сервера, как раньше, тем не менее они до сих пор остаются востребованными. На мой взгляд, самой удобной связкой для прокси является squid + sams2, прежде всего за свое удобство администрирования. Остальные продукты со схожим функционалом выглядят гораздо хуже и менее удобно, а зачастую они старее и тоже не поддерживаются.

Содержание:

Введение

Я успешно разобрался с настройкой sams2 под CentOS 7 и решил проделать то же самое на Freebsd 10.2. Мне нравится эта система. Не могу аргументировать чем именно, просто нравится и все. С нее началось мое первое знакомство с unix системами и с тех пор я стараюсь не забывать о ней, хотя работаю в ней все меньше и меньше.

Sams популярная и в своем роде уникальная система управления конфигурацией squid, которая работает через свой демон и php панель управления. Аналогов по удобству и функционалу я не встречал. Есть что-то похожее, но еще более старое, например STC. Но мне она внешне вообще не нравится, тоже очень старая и давно не развивается.

Можно схожий функционал реализовать отдельными средствами. Например, настроить ту или иную авторизацию в squid, создать списки доступа, списки ресурсов, включить все это. Потом анализировать логи, например, с помощью SARG. По идее, получится то же самое, что и самс, но управлять неудобно, отчеты в SARG не такие красивые и наглядные. В общем, не то.

Так что будем разбираться и настраивать самую последнюю из выпущенных версий самс — sams 2.0 от 1 апреля 2014 года.

Подготовка системы и установка зависимостей

Если вы только знакомитесь с материалом и еще не подготовили свою систему, то рекомендую воспользоваться моей подробной инструкцией с видео по установке freebsd 10.2. Именно на ней я буду производить установку и настройку sams2.

После установки необходимо на всякий случай обновить freebsd до последней версии. Если вы это уже сделали, то рекомендую выполнить предварительную настройку системы, ну или хотя бы установите MC, с ним удобнее.

Теперь установим необходимые пакеты. Можно все собрать из портов, но я буду ставить готовые пакеты, просто потому что так быстрее. Это не принципиально, если вы привыкли собирать из портов, делайте это.

Нам понадобится web сервер для работы панели администрирования. Подробно этот вопрос я уже рассматривал отдельно в статье посвященной настройке web сервера. Можно подсмотреть там, как быстро установить apache + php + mysql + phpmyadmin. Здесь я просто приведу список команд, которые я использовал при установке. За всеми подробностями и комментариями прошу обращаться по приведенной ссылке.

Сервер, с которым будем работать:

```
# uname -v
FreeBSD 10.2-RELEASE-p8
```

Выполним установку **mysql**, ее использует самс для хранения данных:

```
# pkg install -y mysql56-server
```

Добавляем mysql_enable=»YES» в /etc/rc.conf и запускаем mysql:

```
# /usr/local/etc/rc.d/mysql-server start
```

Выполним начальную настройку mysql с помощью скрипта:

```
# /usr/local/bin/mysql_secure_installation
```

Устанавливаем web сервер **apache**, он нам нужен для работы панели администрирования:

```
# pkg install -y apache24
```

Добавляем apache24_enable=»YES» в /etc/rc.conf и запускаем:

```
# /usr/local/etc/rc.d/apache24 start
```

Устанавливаем все необходимое, связанное с **php**:

```
# pkg install -y php56 php56-extensions mod_php56 php56-mysql
```

Этот шаг не обязательный, можно пропустить. Я просто привык использовать phpmyadmin в работе. С ним удобно. Устанавливаем **phpmyadmin**:

```
# pkg install -y phpmyadmin
```

Редактируем конфиг apache, добавляем в самый конец для работы php скриптов и доступа к phpmyadmin:

```
<IfModule mod_php5.c>
 AddType application/x-httpd-php .php
 AddType application/x-httpd-php-source .phps
 DirectoryIndex index.php index.html
</IfModule>
```

```
Alias /phpmyadmin/ "/usr/local/www/phpMyAdmin/"

<Directory "/usr/local/www/phpMyAdmin/">
 AllowOverride All
 Require all granted
 DirectoryIndex index.php index.html index.htm
 Order allow,deny
 Allow from All
</Directory>
```

Создаем файл настроек php.ini, скопировав дефолтный:

```
# cp /usr/local/etc/php.ini-production /usr/local/etc/php.ini
```

И устанавливаем там временную зону:

```
date.timezone = Europe/Moscow
```

Если не установить, то в веб интерфейсе самса будут постоянно вылезать ошибки, связанные с временной зоной. После всех изменений перезапускаем apache:

```
# apachectl restart
```

Все, web сервер готов. Для теста можете зайти на *http:///ip-сервера/phpmyadmin/* и проверить, все ли в порядке. Если что-то не так, то разберитесь сначала с работой веб сервера, и только потом двигайтесь дальше.

> Если веб сервер не работает, то обратитесь либо к моему руководству на тему настройки веб сервера, либо к какому-нибудь другому. Подобного материала в интернете много. Его можно было не включать в эту статью, но я для целостности картины привел краткую настройку. Банального копи паста достаточно, чтобы все заработало.

Установка прокси сервера squid

Теперь установим сам прокси сервер **squid**:

```
# pkg install -y squid
```

Добавляем squid_enable=»YES» в /etc/rc.conf и запускаем его:

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

Сейчас можно проверить работу прокси сервера squid. В дефолтной конфигурации он должен выпускать в интернет по ip, список которых есть в конфигурационном файле — /usr/local/etc/squid/squid.conf. Рекомендую на всякий случай это сделать, чтобы убедиться в корректности работы сквида.

Если вы будете использовать кэш, то перед запуском сквида раскомментируйте в конфигурационном файле соответствующую настройку и выполните команду в консоли:

```
# squid -z
```

Чтобы самс корректно добавлял свои изменения в файл конфигурации, в нем должны быть метки, начинающиеся с **# TAG:**. Раньше дефолтный конфиг сквида уже был с ними, но затем его существенно уменьшили в размере и метки из него исчезли. Нам нужно добавить их самим. Вот необходимый набор этих меток. Между ними обязательно должна быть пустая строка:

```
# TAG: acl

# TAG: url_rewrite_access

# TAG: url_rewrite_program

```

```
# TAG: url_rewrite_children

# TAG: delay_pools

# TAG: delay_class

# TAG: delay_access

# TAG: delay_parameters

# TAG: http_access

# TAG: http_access2

# TAG: icp_access
```

> Предупреждаю, если вы вдруг забудете. Проверьте ротацию логов squid. Если она не будет работать, то есть большая вероятность, что логи займут весь раздел диска. При интенсивном использовании прокси сервера, логи разрастаются очень быстро. Обратите на это внимание.

Установка и настройка sams2 в Freebsd

Переходим к самому главному — настройке **sams2**. Перед этим установим wget и unzip. Они нам нужны будут для скачивания и распаковки исходных текстов.

```
# pkg install -y wget unzip
```

Скачиваем исходники с github:

```
# cd /root
# wget https://github.com/PavelVinogradov/sams2/archive/master.zip --no-check-certificate
```

Извлекаем содержимое архива:

```
# unzip master.zip
```

После этого устанавливаем необходимые для сборки пакеты autoconf, automake и libtool:

```
# pkg install -y autoconf automake libtool
```

Переходим в папку *sams2-master* и начинаем сборку программы:

```
# cd /root/sams2-master
# make -f Makefile.cvs
```

Запускаем конфигурирование:

```
# sh ./configure
```

Скрипт должен отработать без ошибок и вывести информацию:

```
Use MySQL API: yes
Use PostgreSQL API: no
Use unixODBC API: no
Use LDAP API: no
Using pcre: pcrecpp
Use dynamic plugin: yes

Locations:
```

```
config file: /usr/local/etc/sams2.conf
daemons: /usr/local/bin
web interface: /usr/local/share/sams2
documentation: /usr/local/share/doc/sams2-2.0.0

Note: If later on, you will use
make install exec_prefix=/foo
or make install DESTDIR=/tmp/package
the locations above would be incorrect

Configure completed. Run make (or gmake) to build the programs.
```

Если у вас так же, то все в порядке, продолжаем. Если какие-то ошибки, то надо разбираться. Обязательно должны быть указаны значения **yes** в пунктах Use MySQL API и Using pcre. Без первого ничего не заработает, так как хранить конфигурацию будет негде, без второго не будет работать редиректор.

Устанавливаем sams2 на freebsd сервер:

```
# make install
```

Установка не проходит, выскакивает ошибка:

```
chmod 0777 //usr/local/share/sams2/data
chmod: //usr/local/share/sams2/data: No such file or directory
*** Error code 1

Stop.
make[2]: stopped in /root/sams2-master/php
*** Error code 1
```

```
Stop.
make[1]: stopped in /root/sams2-master/php
*** Error code 1

Stop.
make: stopped in /root/sams2-master
```

Где-то в исходниках ошибка — добавлен лишний слеш в пути. Вместо того, чтобы разбираться, где ошибка, проще создать директорию data самим:

```
# mkdir /usr/local/share/sams2/data
```

Запускаем установку снова. Теперь все должно пройти без ошибок.

Дальше нам нужно добавить web интерфейс sams2 в конфигурацию apache. Для этого в */usr/local/etc/apache24/httpd.conf* добавляем в самый конец:

```
Alias /sams2 /usr/local/share/sams2

<Location "/sams2">
 AddDefaultCharset UTF-8
 Options Indexes
 Require all granted
 Order allow,deny
 Allow from All
</Location>
```

Перезапускаем apache для применения изменений:

```
# apachectl restart
```

Отредактируем файл настроек sams2. Он располагается по адресу */usr/local/etc/sams2.conf*. Меняем только 2 параметра:

```
DB_USER=root
DB_PASSWORD=passroot
```

Указываем в них пользователя mysql root и его пароль. Теперь идем в браузер по адресу *http://ip-сервера/sams2/setup.php* Скорее всего вы увидите чистую белую страницу, либо вообще ошибку при загрузке. Если посмотреть при этом в лог веб сервера, то там будет следующее:

```
PHP Parse error:  syntax error, unexpected '$configbuttom_7_log_2' (T_VARIABLE) in /usr/local/share/sams2/lang/lang.EN on line 1100
```

Это баг самса, его нужно исправить. Для этого открываем в редакторе файл */usr/local/share/sams2/lang/lang.EN* и убираем пробелы в начале строк с 1100 по 1105, там есть лишние символы. Должно получиться вот так:

После редактирования обновляем страничку и начинаем настройку самса через web интерфейс. Выбираем кодировку **Russian UTF-8**. Нужна именно эта кодировка, так как мы ее же указали в свойствах алиаса в апаче. Если там этого не сделать, а здесь выбрать привычную кодировку 1251, то в интерфейсе будут проблемы с отображением текста. Я опытным путем выяснил, что именно в такой конфигурации все работает нормально.

Проверяем, чтобы все важные параметры были зеленые, на safe_mode не обращаем внимание.

Указываем root учетку от mysql и данные для вновь создаваемой базы данных и учетной записи sams:

Если все нормально, то должны увидеть следующую картинку:

После нажатия на **Далее >>** откроется окно для входа в панель администрирования самс.

Чтобы авторизоваться администратором, нужно снизу нажать на ссылку «*Авторизация администратора SAMS*» и ввести учетные данные по-умолчанию: пользователь **admin**, пароль **qwerty**. Открывается главная страница с системной информацией.

Теперь еще раз отредактируем конфигурационный файл sams.conf и введем туда данные от учетки sams, которую мы создали. Заодно поменяем еще несколько параметров, так как изначально там стоят не верные данные. Должно быть вот так:

```
SQUIDROOTDIR=/usr/local/etc/squid
SQUIDCACHEDIR=/var/squid/cache
SQUIDPATH=/usr/local/sbin
```

Чтобы запускать sams2 нужно создать для него скрипт запуска и поместить в /usr/local/etc/rc.d/. Делаем это:

```
# mcedit /usr/local/etc/rc.d/sams
```

```
#!/bin/sh

. /etc/rc.subr

name="samsd"
rcvar=`set_rcvar samsd`
command="/usr/local/bin/sams2daemon"

load_rc_config $name
command_args="${samsd_server_flags}"
sig_stop="USR1"
```

```
run_rc_command "$1"
```

Сохраняем файл и делаем его исполняемым:

```
# chmod 0555 /usr/local/etc/rc.d/sams
```

Добавляем samsd_enable=»YES» в rc.conf и запускаем **sams2daemon**:

```
# /usr/local/etc/rc.d/sams start
```

Логинимся в веб интерфейс и настраиваем sams.

Сразу обращаю внимание на один нюанс. По-умолчанию в самс уже указан один временной диапазон **Full day**. Он включает в себя весь день, но при применении настроек squid, в его конфиге этот интервал почему-то обозначается как

```
acl Sams2Time1 time MTWHFAS 23:00-23:59
```

Соответственно, все, что не попадает в этот интервал, в интернет не пускает. Я лично этими интервалами не пользуюсь, поэтому просто удалил его и все доступы нормально заработали. Вы либо заново создайте свой интервал, либо тоже просто удалите, если вам не нужно.

Добавим теперь тестового пользователя с доступом по ip. Для этого идем в раздел **Пользователи** и жмем на кнопочку *Добавить пользователя*.

Заполняем необходимые поля и добавляем юзера.

Не забываем перезапустить демон squid через веб интерфейс самса. Для этого идите в раздел **SQUID -> Proxy Server** и снизу нажимайте на вторую иконку слева.

Синяя строка с текстом Команда на реконфигурирование Squid получена демоном означает, что команда на перезапуск нормально отработала.

В конфигурационном файле */usr/local/etc/squid/squid.conf* должны добавиться следующие строки:

```
# TAG: acl
acl Sams2Template1 src 192.168.1.10

# TAG: url_rewrite_access
acl Sams2Proxy dst 192.168.1.21
url_rewrite_access deny Sams2Proxy

# TAG: http_access
# Setup Sams2 HTTP Access here
http_access allow Sams2Template1
```

Теперь можно включать этому пользователю в настройках браузера proxy сервер и выходить в интернет. Через несколько минут его статистика посещений появится в панели управления.

Дальше каждый продолжает настройку по своим потребностям. Там все достаточно понятно и практически не отличается от настроек 1-й версии, описание которой достаточно много в интернете.

Заключение

Поле завершения установки и настройки sams2, а затем и реальной эксплуатации у меня ложилось двоякое впечатление. С одной стороны вроде работает, но как-то криво. В программе баги, в логи она во время работы постоянно сыпет сообщениями:

```
Dec 27 01:20:08 freebsd samsdaemon[15997]: ***ERROR: squidlogparser.cpp:445 Unknown cache result
Dec 27 01:20:08 freebsd samsdaemon[15997]: +++WARNING: Unknown cache result TCP_TUNNEL
```

Хотя на работу это вроде как не влияет. В инет пускает, статистику считает. Я еще не настраивал редиректор и списки запрета доступа, но по отзывам, там тоже есть проблемы и работает все нестабильно. Начиная с версии 3.4 поменялся формат ответа редиректора и он стал несовместим с самсовским. Нужно править исходники и собирать с исправлениями. На CentOS я обошелся просто более старой версией squid, чтобы не разбираться с этой проблемой. Думаю, можно вообще без запретов обойтись, это сейчас не актуально из-за большого развития мобильных гаджетов и хорошего доступа в интернет с них. Но если вам это важно, имейте в виду.

Несмотря на все проблемы, я не нашел замены, которая была бы такой же удобной и наглядной, как sams. Придется пока с ним дружить и находить общий язык. Надеюсь, что кто-нибудь возьмется за его развитие и поддержку.

Дополнительные материалы по Freebsd

- Умение подбирать конфигурацию под необходимые задачи, управлять процессами и обеспечивать безопасность системы.
- Владение основными рабочими инструментами системного администратора.
- Понимание особенностей развертывания, настройки и обслуживания сетей, построенных на базе Linux.
- Способность быстро решать возникающие проблемы и обеспечивать стабильную и бесперебойную работу системы.

Проверьте себя на вступительном тесте и смотрите подробнее программу по .

Рекомендую полезные материалы по Freebsd:

• Установка
• Настройка
• Обновление
• Шлюз
• Прокси сервер
• Веб сервер NGINX
• Веб сервер Apache

Описание установки Freebsd 11 на одиночный диск, либо на софтовый raid1, сделанный средствами zfs, которые поддерживает стандартный установщик.
Базовая настройка Freebsd, которую можно выполнить после установки сервера общего назначения. Представлены некоторые рекомендации по повышению удобства пользования и безопасности.
Описание и нюансы обновления системы Freebsd с помощью утилиты freebsd-update. Показано пошагово на конкретном примере обновления.
Настройка Freebsd шлюза для обеспечения выхода в интернет. Используется ipfw и ядерный нат, dnsmasq в качестве dhcp и dns сервера. Мониторинг сетевой активности с помощью iftop.
Подробная настройка на Freebsd прокси сервера squid + sams2 - панели управления для удобного администрирования.
Настройка максимально быстрого web сервера на базе Freebsd и nginx + php-fpm. Существенный прирост производительности по сравнению с классическим apache.
Настройка web сервера на Freebsd в связке с apache, nginx, php и mysql. Пошаговая установка и настройка каждого компонента.