ARPWatch - следим за новыми устройствами в сети

Опубликовано muff в Чт, 2011-10-20 18:11

Рано или поздно, любой сетевой администратор сталкивается с необходимостью контролировать смену/появление новых MAC-адресов в сети. Если сеть совсем маленькая - это не сложно, если же сеть на сотни устройств - контролировать подключение устройств к сети становится довольно проблематично. С помощью утилиты **ARPWatch** можно отслеживать изменения в сети. **ARPWatch** отслеживает соответствие Ethernet-адресов и IP-адресов. Активность регистрируется в **syslog** и с помощью почтовых оповещений. Для прослушивания ARP-трафика на локальном ethernet-интерфейсе используется библиотека pcap.

Назначение ARPWatch

- отслеживать появление в сети новых устройств
- отслеживать подмену IP-адресов
- обнаруживать атаки ARP-вирусов

Принцип работы

- **ARPWatch** запускается на Unix-сервере и работает в фоновом режиме как демон
- **ARPWatch** слушает на указанном сетевом интерфейсе все широковещательные ARP-уведомления вида "я, устройство с MAC-адресом 11-22-33-44-55-66, имею IP-адрес 77.88.99.111"
- Информация сохраняется во внутренней базе
- При появлении новых устройств или изменении существующих связок MAC-IP отправляется уведомление по электронной почте

Недостатки

- Каждое уведомление отправляется отдельным сообщением.
- Такая отчётность занимает много места, и самое главное - абсолютно лишена наглядности.
- В большой сети не всегда возможно подключить сервер с **ARPWatch** в каждый сегмент.

Режимы Веб-интерфейса

- Показ сообщений с группировкой по MAC или IP
- Фильтрация за последний день и час
- Показ всех сообщений для выбранного MAC или IP
- Статистика по количеству сообщений для MAC и IP

Более детально с информацией о **ARPWatch** можно ознакомиться на странице проекта [1].
После чтения документации можно приниматься за установку **ARPWatch** из системы портов:

```
# cd /usr/ports/net-mgmt/arpwatch/ && make install clean && rehash
```

По завершению установки выводится уведомление о возможности обновления файла **ethercodes.dat**:

```
You can update the ethercodes.dat file executing the following steps

cd /usr/local/arpwatchfetch http://standards.ieee.org/regauth/oui/oui.txt [2]
./massagevendor oui.txt > ethercodes.datrm oui.txt
```

Выполним предложенные шаги:

```
# cd /usr/local/arpwatch
 # fetch http://standards.ieee.org/regauth/oui/oui.txt [2]
 oui.txt                                 100% of 2437 kB  432 kBps
# ./massagevendor oui.txt > ethercodes.dat
 # rm oui.txt
```

Большинству администраторов известно, что в первых трех октетах MAC-адреса кодируется производитель оборудования. Вышеописанными действиями мы обновили локальную базу связки MAC/Производитель.

Для запуска утилиты необходимо добавить опции запуска в **rc.conf**:

```
# echo '# ARPWatch - IP-MAC monitoring' >> /etc/rc.conf
 # echo 'arpwatch_enable="YES"' >> /etc/rc.conf
 # echo 'arpwatch_flags="-m admin [at] domain [dot] com"' >> /etc/rc.conf
 # echo 'arpwatch_interfaces="vlan96 vlan97 vlan98"' >> /etc/rc.conf
```

В даном примере указан e-mail, на который отправлять уведомления и дополнительно перечислены сетевые интерфейсы, на которых нужно искать связки IP-MAC.

Запускаем утилиту:

```
# sh /usr/local/etc/rc.d/arpwatch start
 Starting arpwatch.
Starting arpwatch.
Starting arpwatch.
```

Проверяем, запустилась ли утилита:

```
# ps -ax | grep arpwatch
 56017  p0  S      0:00,05 /usr/local/sbin/arpwatch -m admin [at] domain [dot] com -i vlan96 -f
/usr/local/arpwatch//arp.vlan96.dat
56020  p0  S      0:00,05 /usr/local/sbin/arpwatch -m admin [at] domain [dot] com -i vlan97 -f
/usr/local/arpwatch//arp.vlan97.dat
56023  p0  S      0:00,04 /usr/local/sbin/arpwatch -m admin [at] domain [dot] com -i vlan98 -f
/usr/local/arpwatch//arp.vlan98.dat
```

Если на указанный e-mail начали приходить уведомления, значит **ARPWatch** корректно работает. Пример пришедшего уведомления с темой "**new station**":

```
hostname: <unknown>
ip address: 10.200.96.20
ethernet address: f4:ec:38:9a:e4:f
ethernet vendor: TP-LINK TECHNOLOGIES CO., LTD.
```

```
timestamp: Thursday, October 13, 2011 1:43:02 +0300
```

ARPWatch рассылает четыре вида сообщений.

- **new activity** - связка ethernet/ip-адресов снова проявила активность спустя шесть месяцев или больше
- **new station** - ethernet-адрес зафиксирован впервые
- **flip flop** - ethernet-адрес изменился с одного известного адреса на другой известный адрес
- **changed ethernet address** - хост перешёл на использование нового ethernet-адреса

Кроме отправки уведомлений на e-mail, **ARPWatch** также пишет события в **syslog**. Пример записи в syslog:

```
Oct 13 01:43:03 router0 arpwatch: new station 10.200.96.20 f4:ec:38:9a:e4:f
```

В **syslog** могут писаться следующие типы уведомлений:

- **ethernet broadcast** - MAC-адрес хоста является широковещательным.
- **ip broadcast** - IP-адрес хоста является широковещательным.
- **bogon** - адрес отправителя IP-пакета не входит в непосредственно подключённую сеть (directly connected network) для заданного интерфейса.
- **ethernet broadcast** - MAC-адрес отправителя состоит из одних нулей или одних единиц.
- **ethernet mismatch** - MAC-адрес отправителя пакета не соответствует MAC-адресу, указанному внутри ARP-запроса.
- **reused old ethernet address** - ethernet-адрес изменился с известного адреса на адрес, который был замечен ранее, но не только что. (Похоже на flip flop, но чуть-чуть другое.)
- **suppressed DECnet flip flop** - сообщение "flip flop" подавлено в связи с тем, что как минимум один из двух адресов является адресом DECnet.

Настроим ведение логов в **syslog**. Добавим необходимые опции в **syslog.conf**:

```
# echo '\!arpwatch' >> /etc/syslog.conf
 # echo '*.notice                                /var/log/arpwatch.log' >> /etc/syslog.conf
```

Не стоит забывать о самом файле логов. Необходимо создать его:

```
# touch /var/log/arpwatch.log
```

После всех этих манипуляций даем команду на перезапуск **syslogd**:

```
# killall -HUP syslogd
```

В **/var/log/arpwatch.log** теперь попадают следующие записи:

```
Oct 18 01:10:05 router0 arpwatch: new station 10.100.0.1 0:15:17:8f:bf:fcOct 18 01:1
0:05 router0 arpwatch: new station 10.100.0.116 0:12:cf:55:6e:40Oct 18 01:10:08 rout
er0 arpwatch: new station 10.100.0.2 0:1e:58:2d:46:11
```

Также не стоит забывать о регулярной ротации логов. Ротацию логов будем выполнять каждый шестой день недели:

```
# echo '/var/log/arpwatch.log                640  5     *    $W6D0 JN' >> /etc/newsyslog.conf
```

Ну и напоследок, попробуем "прикрутить" web-интерфейс к **ARPWatch**. Остановим **ARPWatch** и удалим dat-файлы, чтобы после запуска **ARPWatch** все вхождения считал по новому:

```
# sh /usr/local/etc/rc.d/arpwatch stop
 Stopping arpwatch.
# rm /usr/local/arpwatch/arp.*.dat
 # rm /usr/local/arpwatch/arp.*.dat-
```

Рассмотрено два различных Web-интерфейса для отображения результатов работы **ARPWatch**:

- Вариант 1
- Вариант 2

Лично я отдаю предпочтение варианту №2. Возможно я его и не тестировал бы, если бы первый вариант полностью устраивал.

Web-interface №1

Скачиваем себе в домашний каталог архив с дополнениями к **ARPWatch**. Распаковываем архив в созданный каталог **arpwatch**:

```
# mkdir /home/muff/arpwatch && cd /home/muff/arpwatch
 # fetch http://muff.kiev.ua/files/arpwatch.tar.gz [3]
 arpwatch.tar.gz                       100% of   21 kB 9548 kBps
# tar -xzf arpwatch.tar.gz
```

Рекомендую ознакомиться с файлом **README** - именно в нем и хранятся инструкции по дальнейшей настройке.

Шаг первый - создание базы данных:

```
mysql> create database arpwatch;
 Query OK, 1 row affected (0.02 sec)
mysql> use arpwatch;
 Reading table information for completion of table and column names
You can turn off this feature to get a quicker startup with -A

Database changed
```

Дальше необходимо создать структуру таблиц. Согласно **README**, необходимо выполнить импорт из файла **arpwatch.sql**, но его я никак не мог обнаружить, поэтому пошел альтернативным путем. В консоли MySQL необходимо выполнить следующий запрос для создания структуры таблиц (надеюсь консоль **MySQL** не закрывали и на данный момент задействована БД **arpwatch)**:

```
CREATE TABLE arpwatch(  id INTEGER UNSIGNED NOT NULL AUTO_INCREMENT PRIMARY KEY ,
subject_id ENUM( 'unknown', 'new_activity', 'new_station', 'flip_flop', 'changed_ethernet_address' ) ,
```

```
subject VARCHAR( 100 ) ,
ipaddr VARCHAR( 20 ) ,
macaddr VARCHAR( 20 ) ,
old_macaddr VARCHAR( 20 ) ,
tstamp INTEGER UNSIGNED,
previous_tstamp INTEGER UNSIGNED,
KEY ( ipaddr ) ,
KEY ( macaddr ) ,
KEY ( old_macaddr ) ,
KEY ( tstamp ) ,
KEY ( previous_tstamp ) ,
KEY ( subject_id )

);
```

Шаг второй - создание пользователей с необходимыми правами доступа:

```
mysql> grant insert on arpwatch.arpwatch to arpwatch2sql@localhost identified by
'VerySecretPassword1';
 Query OK, 0 rows affected (0.00 sec)
mysql> grant select on arpwatch.arpwatch to arpwatch2cgi@localhost identified by
'VerySecretPassword2';
 Query OK, 0 rows affected (0.00 sec)
mysql> flush privileges;
 Query OK, 0 rows affected (0.00 sec)
```

Шаг третий - запуск **arpwatch**. Стоит иметь ввиду, что в данном случае необходимо, чтобы почта доставлялась локальному пользователю **arpwatch**. Создадим пользователя **arpwatch** , которому и будем доставлять почту (я использовал **uid** 1005 - проверьте у себя какой **uid** можно использовать):

```
# pw useradd -n arpwatch -u 1005 -g mailnull -c ARPWatch -d /nonexistent -s
/usr/sbin/nologin
```

Отредактируем опции запуска **ARPWatch** в **/etc/rc.conf**, а именно - получателем уведомлений сделаем локального пользователя **arpwatch**:

```
# cat /etc/rc.conf | grep arpwatch_flags
 arpwatch_flags="-m arpwatch@localhost [4]"
```

Запустим **arpwatch**:

```
# sh /usr/local/etc/rc.d/arpwatch start
 Starting arpwatch.
Starting arpwatch.
Starting arpwatch.
```

В почтовом лог-файле можно обнаружить записи о доставке сообщений локальному пользователю **arpwatch**. Пример такой записи:

```
Oct 18 00:18:15 router0 sendmail[96661]: p9HLIAWV096661: to=arpwatch@localhost,
ctladdr=muff (1001/0), delay=00:00:05, xdelay=00:00:05, mailer=relay, pri=30269,
relay=[127.0.0.1] [127.0.0.1], dsn=2.0.0, stat=Sent (p9HLIFuN096663 Message accepted for
delivery)
```

Шаг четвертый - внесение запуска файла **arpwatch2sql** в **cron**. Для начала скопируем файл **arpwatch2sql** из домашнего каталога, куда был распакован архив, в рабочий каталог **ARPWatch**:

```
# cp arpwatch2sql /usr/local/arpwatch/
```

Необходимо "поправить" **arpwatch2sql** - переменную **$mbox** необходимо установить в **/var/mail/arpwatch**. Также я обратил внимание на то, что скрипт требует наличия **Perl**-ового расширения **Date::Parse** в системе (это будет оговорено в пятом шаге инструкции, но модуль требуется для шага 4, поэтому описываю его установку здесь). Выполним установку даного расширения из системы портов:

```
# cd /usr/ports/devel/p5-DateTime-Format-DateParse && make install clean && rehash
```

По ходу инсталяции **Date::Parse** установил еще три десятка модулей Perl, но это, как говорится, уже издержки производства.

Добавим в **cron** запуск скрипта **arpwatch2sql** с интервалом в 5 минут. Есть отличия от предложенного примера. Путь к исполняемым файлам необходимо указывать полностью. Строка запуска выглядит следующим образом:

```
*/5 * * * * root /usr/local/arpwatch/arpwatch2sql |/usr/local/bin/mysql -u arpwatch2sql -pVerySecretPassword1 arpwatch
```

База данных должна наполняться данными, а почтовый ящик пользователя **arpwatch** (**/var/mail/arpwatch**) - опустошаться.

Шаг пятый - настройка веб-интерфейса. Мануал советует поместить файлы в каталог **cgi-bin** веб-сервера. Однако, мы легких путей не ищем. Добавим такой блок в **httpd.conf** - конфигурационный файл веб-сервера **Apache**:

```
Alias /arp/ "/usr/local/www/arpwatch/"
<Directory "/usr/local/www/arpwatch/">
        Options ExecCGI
        DirectoryIndex arpwatch.cgi
        AllowOverride None
        Order Deny,Allow
        Allow from all
</Directory>
```

Создадим каталог **/usr/local/www/arpwatch** и скопируем в него файлы веб-интерфейса и установим владельцем файлов пользователя **www**:

```
# mkdir /usr/local/www/arpwatch
 # cp Webutils.pm arpwatch.cgi arpwatch-topstats.cgi  /usr/local/www/arpwatch/
 # chown -R www:www /usr/local/www/arpwatch
```

Также необходима поддержка таких модулей **Perl**, как **DBI** и **CGI**.

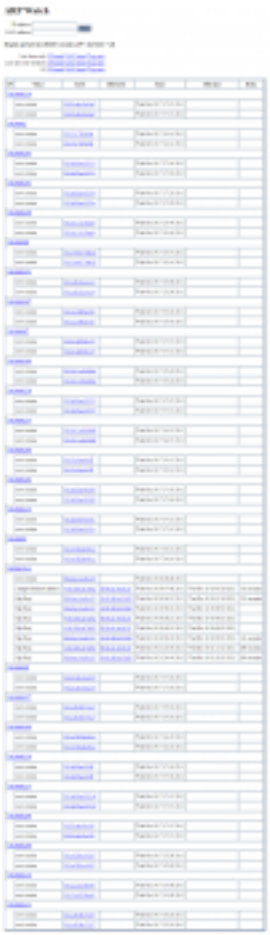
Шаг шестой - настройка параметров подключения к базе данных. Необходимо отредактировать файл **WebUtils.pm** и изменить параметры соединения с базой данных на корректные (секция **sub webutils_utminit**). После внесения изменений можно проверить скрипт на наличие ошибок синтаксиса:

```
# perl -c /usr/local/www/arpwatch/arpwatch.cgi
 /usr/local/www/arpwatch/arpwatch.cgi syntax OK
```

Отправим **Apache** команду на перечитывание конфигурации:

```
# apachectl graceful
```

В браузере вводим ссылку **http://ip_servera/arp/** и наблюдаем примерно такую картину:



[5]

Web-interface №2

Первым делом необходимо создать базу данных, куда будем записывать данные, полученные от **ARPWatch**. Создадим базу данных и пользователя с правами на эту базу данных:

```
mysql> create database arpwatch;
 Query OK, 1 row affected (0.00 sec)
mysql> grant all on arpwatch.* to arpwatch@localhost identified by
'VerySecretPassword';
Query OK, 0 rows affected (0.00 sec)
mysql> use arpwatch;
Database changed
```

SQL-запросы для создания структуры таблиц будут следующие:

```
CREATE TABLE flip_flop(  hostname VARCHAR( 255 ) ,
ip_address VARCHAR( 15 ) ,
ethernet_address VARCHAR( 17 ) ,
ethernet_vendor VARCHAR( 255 ) ,
old_ethernet_address VARCHAR( 17 ) ,
old_ethernet_vendor VARCHAR( 255 ) ,
TIMESTAMP VARCHAR( 19 ) ,
```

```
previous_timestamp VARCHAR( 19 ) ,
delta VARCHAR( 50 )

);

CREATE TABLE changed_ethernet_address(
 hostname VARCHAR( 255 ) ,
ip_address VARCHAR( 15 ) ,
ethernet_address VARCHAR( 17 ) ,
ethernet_vendor VARCHAR( 255 ) ,
old_ethernet_address VARCHAR( 17 ) ,
old_ethernet_vendor VARCHAR( 255 ) ,
TIMESTAMP VARCHAR( 19 ) ,
previous_timestamp VARCHAR( 19 ) ,
delta VARCHAR( 50 )

);

CREATE TABLE new_station(
 hostname VARCHAR( 255 ) ,
ip_address VARCHAR( 15 ) ,
ethernet_address VARCHAR( 17 ) ,
ethernet_vendor VARCHAR( 255 ) ,
TIMESTAMP VARCHAR( 19 )

);

CREATE TABLE new_activity(
 hostname VARCHAR( 255 ) ,
ip_address VARCHAR( 15 ) ,
ethernet_address VARCHAR( 17 ) ,
ethernet_vendor VARCHAR( 255 ) ,
TIMESTAMP VARCHAR( 19 )


);
```

Скачиваем скрипт **arpwatch.pl** в каталог **/usr/local/arpwatch** и распакуем его из архива:

```
# cd /usr/local/arpwatch
 # fetch http://muff.kiev.ua/files/arpwatch.pl.tar.gz [6]
 arpwatch.pl.tar.gz                    100% of 1210  B    9 MBps
# tar -xzf arpwatch.pl.tar.gz
```

Этот скрипт будет парсить информацию, получаемую от **ARPWatch** и раскладывать ее по таблицам базы данных. Для того, чтобы скрипт мог "достучаться" до базы данных, необходимо изменить в нем параметры коннекта к базе данных. Редактируем файл и выставляем переменные в необходимые значения:

```
$db_user    = "arpwatch";
$db_passwd  = "VerySecretPassword";
$db_name = "arpwatch";
$db_host = "localhost";
$db_port = "3306";
```

Для корректной работы скрипта необходима поддержка таких модулей **Perl**, как **DBI**, **DBD-mysql** и **Getopt-Long**. Если какой-то из модулей не установлен, его необходимо установить. Желательно из системы портов:

```
# cd /usr/ports/databases/p5-DBI && make install clean && rehash
```

```
# cd /usr/ports/databases/p5-DBD-mysql && make install clean && rehash
# cd /usr/ports/devel/p5-Getopt-Long && make install clean && rehash
```

Для того, чобы скрипт "скармливал" данные в БД, необходимо в него перенаправить уведомления электронной почты **ARPWatch**. В моем случае на роутере работает **Sendmail** в дефолтной конфигурации. Выполним его настройку так, чтобы письма отправленные пользователю **arpwatch** перенаправлялись в скрипт **arpwatch.pl**.

Создадим пользователя **arpwatch**, которому и будем доставлять почту (я использовал **uid** 1005 - проверьте у себя какой **uid** можно использовать):

```
# pw useradd -n arpwatch -u 1005 -g mailnull -c ARPWatch -d /nonexistent -s
/usr/sbin/nologin
```

Отредактируем опции запуска **ARPWatch** в **/etc/rc.conf**, а именно - получателем уведомлений сделаем локального пользователя **arpwatch**:

```
# cat /etc/rc.conf | grep arpwatch_flags
 arpwatch_flags="-m arpwatch@localhost [4]"
```

Создаем почтовый алиас для пользователя **arpwatch** с перенаправлением его почты в скрипт **arpwatch.pl**:

```
# echo 'arpwatch: "|/usr/bin/perl /usr/local/arpwatch/arpwatch.pl"' >> /etc/mail/aliases
```

Чтобы изменения, добавленные в **/etc/mail/aliases** вступили в силу, необходимо отправить **Sendmail**-у команду на перечитывание алиасов:

```
# sendmail -bi
 /etc/mail/aliases: 29 aliases, longest 40 bytes, 344 bytes total
```

Запускаем **ARPWatch** и проверяем, заполняются ли таблицы базы данных. Если заполняются - значит все в норме. Если же нет - смотрите **/var/log/maillog** и диагностируйте ошибку.

Приступим к настройке веб-интерфейса. Перейдем в каталог **/usr/local/www** и загрузим туда архив веб-интерфейса:

```
# cd /usr/local/www
 # fetch http://muff.kiev.ua/files/arpwatch-www.tar.gz [7]
 arpwatch-www.tar.gz                   100% of   51 kB   52 MBps
# tar -xzf arpwatch-www.tar.gz
 # chown -R www:www /usr/local/www/arpwatch
 # rm arpwatch-www.tar.gz
```

Необходимо указать параметры доступа к базе данных в файле **/usr/local/www/arpwatch/config.inc.php**. Редактируем следующие поля:

```
$dbhost = "localhost"; //Сервер базы данных
$dbuser = "arpwatch"; //Имя пользователя БД
$dbpassword = "VerySecretPassword"; //Пароль в БД
$dbname = "arpwatch"; //Имя БД
```

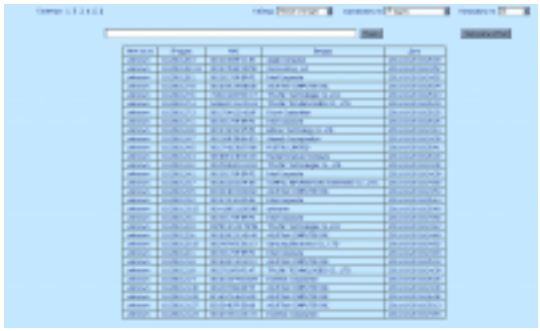
Добавим такой блок в **httpd.conf** - конфигурационный файл веб-сервера **Apache**:

```
Alias /arp/ "/usr/local/www/arpwatch/"
<Directory "/usr/local/www/arpwatch/">
        Options -Indexes
        DirectoryIndex index.php
        AllowOverride None
        Order Deny,Allow
        Allow from all
</Directory>
```

Отправим **Apache** команду на перечитывание конфигурации:

```
# apachectl graceful
```

В браузере вводим ссылку **http://ip_servera/arp/** и видим следующий интерфейс:



[8]

На этом настройку утилиты **ARPWatch** можно считать оконченной. Теперь сеть находится под более жестким контролем.

Источник (получено *2026-11-12 09:02*):
http://muff.kiev.ua/content/arpwatch-sledim-za-novymi-ustroistvami-v-seti

Ссылки:
[1] http://sources.homelink.ru/arpwatch/arpwatch-rus.html
[2] http://standards.ieee.org/regauth/oui/oui.txt
[3] http://muff.kiev.ua/files/arpwatch.tar.gz
[4] mailto:arpwatch@localhost
[5] http://muff.kiev.ua/files/imagepicker/1/arpwatch-00.png
[6] http://muff.kiev.ua/files/arpwatch.pl.tar.gz
[7] http://muff.kiev.ua/files/arpwatch-www.tar.gz
[8] http://muff.kiev.ua/files/imagepicker/1/arpwatch-01.png