

Управляем неуправляемым

Автор: Сергей Клязник, системный администратор [ООО "КСИТ"](#)

Статья публикуется в рамках конкурса "Формула связи"

Большинство маленьких провайдеров начинают с неуправляемого оборудования, как правило разношерстных "мыльниц". И каждый админ такой сети мечтает навести в ней порядок: запретить паразитный трафик, несанкционированные подключения, сделать привязки IP адресов пользователей к MAC адресам, иметь возможность отключать абонентов от услуги не бегая по чердакам в поисках нужного провода. Об одном из способов выжать все возможное из неуправляемой сети пойдет речь в статье. Сразу оговорюсь, что 100% управляемости достигнуть не получится, ибо толковые люди есть везде, однако мне известно о двух достаточно крупных сетях (более 5000 абонентов), успешно использующих данную технологию.

Основная идея основана на технологии ARP spoofing (<http://ru.wikipedia.org/wiki/ARP-spoofing>). Необходим сервер (далее guard) с установленным Linux, на который нужно "загнать" все подконтрольные сегменты (например, с помощью vlan). На сервер устанавливается ip-sentinel (<http://www.nongnu.org/ip-sentinel/>), который с помощью ARP spoofing-а блокирует нежелательные компьютеры.

Теперь более подробно о реализации. В каждом VLAN-е запускается свой экземпляр ip-sentinel-а, который управляется отдельным конфигурационным файлом, например таким:

```
=====
# разрешаем использование IP адреса 192.168.0.123 только компьютеру с MAC-адресом
00:12:79:93:19:58

192.168.0.123@!00:12:79:93:19:58

# блокируем полностью компьютер с MAC адресом 00:02:44:11:a4:6f

*@00:02:44:11:a4:6f

# разрешаем использовать IP адрес 192.168.0.20 любым компьютерам

!192.168.0.20

# полностью блокируем работу IP адресов из сети 10.0.0.0/8

10.0.0.0/8

# полностью блокируем работу всех IP адресов, кроме прописанных выше!

0.0.0.0/0
=====
```

Особенно интересна последняя строка, она позволяет блокировать любую нежелательную активность в IP сети, оставляя возможность работать только разрешенным выше адресам.

Но это еще не все. При блокировании ip-sentinel позволяет запускать внешнюю программу. Эту функцию можно использовать для автоматического создания привязок IP адресов к MAC адресам новых абонентов. Остается только автоматизировать создание конфигурационных файлов и привязать ip-sentinel к биллингу. Для этого в БД биллинга для каждого пользователя обязательно должны быть определены id (уникальный идентификатор), IP, MAC, login, segment (номер сегмента). Также нужно создать таблицу mac_log, куда будет помещаться информация о нарушителях:

```
CREATE TABLE `mac_log` (
  `user_id` int(11) NOT NULL default '0',
  `tstamp` int(11) NOT NULL default '0',
  `mac` char(17) default NULL
) ENGINE=MyISAM;
```

На локальной машине: редактируем скрипт ip_sentinel_maker.pl: прописываем реквизиты для доступа к биллинговой БД, описываем серверы с запущенными ip-sentinel. Создаем каталог, указанный в cfg_local_path, генерируем ssh ключи для доступа на guard-ы ("ssh-keygen -t dsa"). Создаем задание cron для запуска

ip_sentine_maker.pl каждую минуту:

```
1 * * * * /ваш_путь_до/ip_sentinel_maker.pl
```

На guard-ах создаем каталог `cfg_remote_path/intruders`, копируем в `/root/.ssh/authorized_keys` содержимое файла `/root/.ssh/id_dsa.pub` с локальной машины. Устанавливаем ip-sentinel (из исходных текстов либо с помощью пакетного менеджера). Компилируем `logger.c` (`gcc -o cfg_remote_path/logger logger.c`). Создаем скрипт запуска `/etc/init.d/ip-sentinel`, который будет запускать свою копию программы на каждом интерфейсе (во вложении примеры для gentoo и redhat). Не забываем добавить `/etc/init.d/ip-sentinel` в автозагрузку.

Иногда, по неизвестным мне причинам, ip-sentinel аварийно завершается, поэтому на guard-ах делаем задание cron для рестарта процессов каждый час:

```
1 * * * * /etc/init.d/ip-sentinel restart >/dev/null
```

В `/var/log` будут создаваться логи работы каждой копии ip-sentinel, желательно настроить logrotate для ротации этих логов. Пример `/etc/logrotate.d/ip-sentinel`:

```
/var/log/ip-sentinel.* {
    compress
    rotate 4
    weekly
    sharedscripts
    postrotate
        /etc/init.d/ip-sentinel restart >/dev/null
    endscript
}
```

В логах дата хранится в машинном виде, можно использовать `tai64nlocal` из пакета `daemontools` для преобразования ("`cat log | tai64nlocal`").

Теперь рассмотрим ситуацию, когда нужно узнать какие MAC адреса использовал известный IP адрес, либо наоборот какие IP адреса занимал определенный MAC.

Для этого можно запустить на guard-ах замечательную утилиту `arpwatch` (<http://en.wikipedia.org/wiki/Arpwatch>). Она ведет базу данных, в которой хранится информация о том, какие MAC и IP адреса были видны в определенный момент времени. В приложенных файлах примеры `/etc/init.d/arpwatch` для запуска `arpwatch` на каждом интерфейсе, а также скрипт `arpsearch.pl`, который позволяет вытаскивать из базы информацию в удобочитаемом виде (не забываем подредактировать в нем `path` в зависимости от дистрибутива). Примеры запуска скрипта:

1. Узнать какие MAC адреса и когда использовали адрес 192.168.0.14:

```
guard ~ # arpsearch.pl 192.168.0.14
00:1E:58:4C:36:3F    192.168.0.14  Thu Oct 29 19:42:55 2009
00:11:95:86:74:CC    192.168.0.14  Thu Jan  1 20:24:06 2009
```

2. Узнать какие IP адреса использовал MAC 00:11:95:86:74:CC:

```
guard ~ # arpsearch.pl 00:11:95:86:74:CC
00:11:95:86:74:CC    192.168.0.71  Thu Oct 15 12:56:40 2009
00:11:95:86:74:CC    192.168.0.14  Thu Jan  1 20:24:06 2009
```

Подводя итог, упомяну, что ip-sentinel и arpwatch можно откомпилировать и установить на роутере с прошивкой `openwrt`, что делает данное решение относительно дешевым и легко масштабируемым. А вообще, используйте управляемое оборудование, не создавайте себе лишних проблем :)

[arpwatch.init.gentoo](#)

[arpwatch.init.redhat](#)

[arpsearch.pl](#)

[ip-sentinel.init.redhat](#)

[ip-sentinel_maker.pl](#)

[ip-sentinel.init.gentoo](#)