Однажды мне понадобилось ограничить доступ к серверу по ip на основе стран. Я решил это сделать после того, как на этот сайт началось непонятное мне нашествие ботов. Шли они со всего мира и создавали хоть и не существенную, но нагрузку на сервер. Из спортивного интереса хотел их отсечь и искал средства, чтобы сделать это максимально эффективно. Ниже расскажу, как я в итоге настроил ограничение доступа к своему веб серверу на основе списка ip по странам.

Содержание:

Введение

Я заметил подозрительную активность на сайте. Заходили какие-то боты, немного скролили и тыкали по ссылкам. Кто и зачем направил их на мой сайт мне было не понятно. Ничего плохого, по сути, они не делали, просто нарушали подсчет статистики, искажая реальную картину. Я быстро прикинул по статистике сайта, с каких стран в основном ко мне заходят люди. В моем случае это страны СНГ и немного заграницы. Боты же лезли со всего мира. Было принято решение временно закрыть доступ к сайту для всех стран, кроме самых популярных, с которых приходит около 95-98% пользователей. Забегаю вперед скажу, что это дало очень хороший результат, отсеяв почти всех нежелательных посетителей.

Есть разные способы решить данную задачу. Наиболее простой и популярный - воспользоваться модулем для nginx - http_geoip_module. Не стал его

реализовывать по 2-м причинам:

1. По-умолчанию, этот модуль не включен в сборку, необходимо собирать веб сервер заново.
2. Nginx быстрый сервер, успешно справляется с высокими нагрузками, но все же мне казалось более правильно отсечь все лишнее еще на подходах к серверу.

В итоге решил блокировать ботов с помощью iptables. Если у вас он не настроен, можете воспользоваться моей статьей - настройка iptables. Первое, что пришло в голову, это вручную добавить полный список ip адресов конкретной страны. Эта информация без проблем ищется в интернете, например вот тут - http://ipdeny.com/ipblocks/. Я взял список адресов одной страны и загнал его в свой скрипт для iptables, о котором рассказано в моей статье.

Запуск скрипта с добавленным списком огромного количество ip адресов и диапазонов длился секунд 10. Сейчас уже точно не помню, какую страну взял, если не ошибаюсь, то список был на несколько тысяч строк. Я понял, что так дело не пойдет, но все же решил попробовать добавить еще одну страну. После запуска скрипта сервер просто завис. Перестал отвечать на запросы, по ssh меня отключило. Пошел в консоль и перезагрузил сервер.

Я понял, что просто в лоб нельзя в iptables загнать большие списки. Стал искать другой способ, как реализовать ограничение доступа к серверу на основе большого списка правил. Решение было найдено - **ipset**. Официальная страница проекта - http://ipset.netfilter.org/. Ipset представляет из себя модуль ядра и утилиту для настройки, поэтому работает все быстро и переваривает огромные списки правил. В iptables при этом добавляется только одно правило.

Дальше я расскажу, как установить и настроить ipset. Работать будем на сервере CentOS 7. В других дистрибутивах никаких отличий не будет, за исключением установки и запуска модуля ядра.

Установка ipset

Для установки ничего особенного не требуется. Устанавливаем ipset стандартным образом через yum.

```
# yum install ipset
```

Этого достаточно, можно приступать к настройке списка блокировки по странам.

Настройка блокировки доступа к сайту по странам

Дальше можно пойти двумя путями:

1. Блокировать доступ определенным странам.
2. Разрешить доступ списку стран, а от остальных закрыться.

Сначала я пошел по первому пути и стал вычислять страны, с которых приходит больше всего ботов и блокировать их. Потом я понял, что это не очень эффективно с точки зрения затраты времени. Список подходящих мне стран гораздо меньше, чем список нежелательных. Так что я пошел по другому пути и запретил доступ всем, кроме некоторых стран. Я расскажу дальше для примера оба способа. Технически они ничем не отличаются, просто разный подход.

Итак, у нас есть список ip адресов и подсетей, разделенных по странам. Списки очень большие, руками их обрабатывать не получится. Мы это дело автоматизируем, а пока пройдемся по теории. Разберемся с управлением списками в ipset. Для начала создадим список и назовем его - *blacklist*.

```
# ipset -N blacklist nethash
```

Этой командой я создал список *blacklist* типа *nethash*. Строго говоря, я создаю не список, а набор данных типа *nethash*. В данном случае это набор данных адресов подсетей. Именно в виде набора подсетей выглядят списки доступа стран по ip. Если у вас есть список ip адресов, то тип данных будет *iphash*. Более подробно о типах данных можно посмотреть в документации.

Добавим в наш список несколько подсетей.

```
# ipset -A blacklist 5.43.240.0/21
# ipset -A blacklist 5.100.64.0/18
# ipset -A blacklist 5.159.112.0/21
```

Посмотрим содержимое списка *blacklist*.

```
# ipset -L blacklist

Name: blacklist
Type: hash:net
Revision: 3
Header: family inet hashsize 1024 maxelem 65536
Size in memory: 16880
References: 0
Members:
5.100.64.0/18
5.43.240.0/21
5.159.112.0/21
```

Обращаю внимание на выделенные строки. Когда я создавал список стран для блокировки, он получился очень большой и не помещался весь в объект с дефолтными настройками. Их можно изменить таким образом:

```
# ipset create blacklist nethash hashsize 16348 maxelem 131072
```

Проверяем:

```
# ipset -L blacklist

Header: family inet hashsize 16384 maxelem 131072
Size in memory: 262544
```

Обратите внимание на занимаемую память. На маленьком VPS это может стать проблемой. В том числе из этих соображений я решил перейти к белому списку. Он существенно меньше, соответственно, меньше ресурсов тратится на его обработку.

Список мы создали, теперь его надо подключить в iptables. Для этого добавляем правило блокировки.

```
# iptables -A INPUT -m set --match-set blacklist src -j DROP
```

Смотрим, что получилось в правилах:

```
# iptables -L INPUT -v -n
```

```
[root@web etc]# iptables -L INPUT -v -n
Chain INPUT (policy DROP 0 packets, 0 bytes)
 pkts bytes target     prot opt in     out     source               destination
    0     0 ACCEPT     all  --  lo     *       0.0.0.0/0            0.0.0.0/0
    0     0 DROP       all  --  *      *       0.0.0.0/0            0.0.0.0/0           match-set blacklist src
   31  2160 ACCEPT     all  --  *      *       0.0.0.0/0            0.0.0.0/0           state RELATED,ESTABLISHED
    0     0 DROP       all  --  *      *       0.0.0.0/0            0.0.0.0/0           state INVALID
    0     0 DROP       tcp  --  *      *       0.0.0.0/0            0.0.0.0/0           tcp flags:!0x17/0x02 state NEW
    0     0 ACCEPT     tcp  --  eth0   *       0.0.0.0/0            0.0.0.0/0           tcp dpt:22
    0     0 ACCEPT     udp  --  eth0   *       0.0.0.0/0            0.0.0.0/0           udp dpt:53
    0     0 ACCEPT     udp  --  eth0   *       0.0.0.0/0            0.0.0.0/0           udp dpt:123
    0     0 ACCEPT     tcp  --  eth0   *       0.0.0.0/0            0.0.0.0/0           tcp dpt:80
[root@web etc]#
```


Все в порядке, правило добавили. Следите, чтобы запрещающее правило было первым и лишние подключения отваливались сразу. Давайте теперь на примере посмотрим, как организовать белый список. Создаем сам список.

```
# ipset -N whitelist nethash
```

Добавляем в него подсети.

```
# ipset -A whitelist 6.43.240.0/21
# ipset -A whitelist 7.100.64.0/18
# ipset -A whitelist 8.159.112.0/21
```

Создаем правило в iptables, по которому к нашему серверу будут иметь доступ только адреса из списка. Сначала у меня стояли вот эти два правила, которые разрешают доступ к веб серверу всем.

```
iptables -A INPUT -i $WAN -p tcp --dport 80 -j ACCEPT
iptables -A INPUT -i $WAN -p tcp --dport 443 -j ACCEPT
```

Я их изменил на следующие.

```
iptables -A INPUT -i $WAN -m set --match-set whitenet src -p tcp --dport 80 -j ACCEPT
iptables -A INPUT -i $WAN -m set --match-set whitenet src -p tcp --dport 443 -j ACCEPT
```

> Не забывайте важный момент - чтобы у вас белый список работал, у вас должен весь трафик, который не разрешен явно, блокироваться. Подробнее о настройке iptables читайте мою статью, о которой я говорил ранее.

Принцип работы ipset рассмотрели. Теперь автоматизируем все и создадим полный список ip адресов по странам, которым будет разрешен доступ к сайту. Я для этого написал простенький скрипт. Привожу его с комментариями.

```
#!/bin/bash
```

```
# Удаляем список, если он уже есть
ipset -X whitelist
# Создаем новый список
ipset -N whitelist nethash

# Скачиваем файлы тех стран, что нас интересуют и сразу объединяем в единый список
wget -O netwhite http://www.ipdeny.com/ipblocks/data/countries/{ru,ua,kz,by,uz,md,kg,de,am,az,ge,ee,tj,lv}.zone

echo -n "Загружаем белый список в IPSET..."
# Читаем список сетей и построчно добавляем в ipset
list=$(cat netwhite)
for ipnet in $list
 do
 ipset -A whitelist $ipnet
 done
echo "Завершено"
# Выгружаем созданный список в файл для проверки состава
ipset -L whitelist > w-export
```

Запускаете скрипт, он скачивает и объединяет списки нужных вам стран, затем добавляет их в ipnet и в конце делает экспорт в файл. Посмотрите на этот файл. Список в нем должен совпадать с исходным списком, плюс несколько информационных строк в самом начале. На этом все. Если вы добавили правила в iptables, ограничение доступа по ip уже начало работать.

Заключение

Я проверил на своем сайте. Способ простой и очень эффективный. Все лишние боты отваливаются на моменте подключения к серверу. Не нагружают сервер и не портят статистику. Если кто-то знает еще эффективные способы борьбы с нежелательными посетителями, прошу поделиться в комментариях. Когда нашествие ботов прекратилось, запреты убрал.

Этот способ хорош только как временная мера. Постоянно его нельзя использовать, так как часть пользователей все же не сможет попасть к вам на сайт. Но если у вас какой-то специфичный сервис, к которому точно не нужен доступ, к примеру, из Китая, можете воспользоваться описанным мной методом.

Полезную информацию по защите сайта от ботов можно посмотреть в моих статьях на тему настройки fail2ban:

- Защита админки wordpress с помощью fail2ban
- Спам от спама однотипными запросами на веб сервере