**НАСТРОЙКА FAIL2BAN**

Как уже понятно из названия - утилита следит за попытками логина на определенный сервис и банит за попытку ввести неверные учетные данные. Определяет она этот факт через чтение логов конкретного приложения. Утилита довольно гибко настраивается. В комплекте уже есть много подходящих модулей. Здесь рассмотрим только модуль **ssh**.

Первым шагом как всегда будет установка:

sudo apt install fail2ban

sudo systemctl enable fail2ban

sudo service fail2ban start

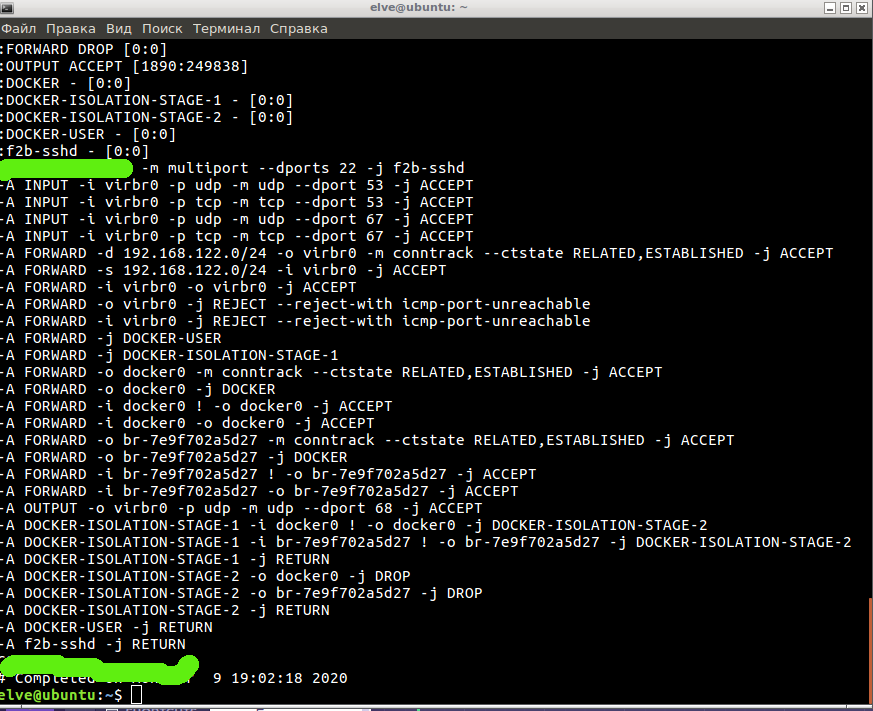
Сервис уже имеет некоторые настройки по умолчанию. И включенный модуль **ssh**.

Ты можешь это легко проверить.

Вводи команду

sudo iptables-save

В выводе должны присутствовать подчеркнутые на скриншоте далее строки. Если они есть, значит все хорошо.



Это значит что в файрволе создан список, в который будут добавляться забаненные.

Рассмотрим основные настройки. Они находятся в файле **/etc/fail2ban/jail.conf**

Выглядит этот файл как-то так:

**[INCLUDES]** Тут по умолчанию указывается файл со специфическими для дистрибутива путями к файлам (в основном к логам).

**[DEFAULT]** Тут определяются настройки по умолчанию.

И далее секции конкретных приложений. Тут задаются персональные параметры для срабатывания по определенному приложению. Если не указывать, то будут использоваться значения по умолчанию.

**[app 1]**

**....**

**[app N]**

Рассмотрим подробнее секцию [**DEFAULT**]

Первый параметр, который будет интересен **ignoreip**. Тут можно перечислить доверенные подсети и адреса, с которых можно сколько угодно раз ошибаться с паролем, но не попасть в бан.

Очень удобно добавлять сюда **ip**-адрес машины системного администратора, чтобы случайно не потерять доступ до сервера =).

Адреса перечисляются через пробел. Можно указывать не только адреса, но и подсети и даже доменные имена. Тут надо знать, что доменное имя можно указывать лишь тогда, когда есть соответствующая ему **PTR**-запись. Иначе скрипту не удастся сопоставить **ip**, с которого подключаются и доменное имя.

**bantime** - время на которое нарушитель будет добавлен в бан. Указывается в секундах

**findtime** - отрезок времени, в течение которого нужно несколько раз ошибиться с паролем, чтобы попасть в бан. Тоже указывается в секундах.

**maxretry** - количество попыток неправильно введенного пароля, после которого хост будет добавлен в бан

По такому словесному описанию не каждый сразу сможет понять как этими опциями пользоваться. Так что рассмотрим на примере:

**bantime = 18000**

**findtime = 3600**

**maxretry = 5**

Эта запись означает, что если кто-то в течение одного часа 5 раз неправильно введет пароль, то будет отправлен в бан на 5 часов.

**backend = auto**

Оставляем как есть. У скрипта есть несколько рабочих "движков", написанных под разные условия. Если вы не понимаете что вы делаете, то лучше оставить "авто". В нашем уроке мы эту опцию рассматривать не будем.

Также поступим и с опцией **usedns.** По умолчанию там указано оптимальное значение.

**logencoding** - можно вручную задать кодировку, в которой пишется информация в логи, которые **fail2ban** читает. Менять ее стоит тогда, когда автоматически кодировка определяется некорректно. Иначе пусть будет **auto**.

Помимо **auto**, можно указать конкретную кодировку. К примеру acsii, utf-8, windows-1251 и т.д.

**enabled** - задает будут ли включены и выключены по-умолчанию секции отдельных приложений.

Если указать значение **false**, то по умолчанию контроль по всем приложениям в конфиге выключен и мы включаем каждое по отдельности

Если указать **true** - то все секции приложений будут включены по умолчанию и если что-то надо отключить точечно, то нужно это делать в секции конкретного приложения.

Рекомендую оставить значение по умолчанию - **false**

**destemail** - адрес, на который присылать отчет о работе **fail2ban**

**sender** - почтовый ящик, от имени которого **fail2ban** будет высылать отчеты

**mta** - как отправлять электронную почту.

Может принимать значения:

**sendmail** - отправлять с помощью программы sendmail

**mail** - отправлять напрямую через smtp.

Пока оставляем без изменений.

**protocol** - указываем протокол по умолчанию для блокировки. **tcp** используется чаще, так что оставляем его. К тому же в настройках для каждого приложения указываются свои сетевые настройки и этот параметр не особо важен.

**port** - порты по умолчанию, доступ к которым мы закрываем для забаненного. Тоже можно оставить, т.к. мало на что влияет.

**chain** - указываем цепочку, в которую мы добавляем наш список забаненных. Файрвол будет рассмотрен чуть позже. Пока разумнее оставить значение по умолчанию - **INPUT**. Так меньше шансов, что блокировка не достигнет адресата.

Настраивать блокировки по приложениям можно прямо в **jail.conf**, а можно создавать отдельные файлы в папке **/etc/fail2ban/jail.d**

Создадим файл **/etc/fail2ban/jail.d/sshd.conf**

Содержимое файла будет следующим:

**[sshd]**

**filter = sshd**

**enabled = true**

**port = ssh**

**logpath = /var/log/auth.log**

**findtime = 3600**

**bantime = 86400**

Т.к. тут появились новые параметры, то надо разобрать их подробнее.

**logpath** - полный путь к лог-файлу, который будем анализировать.

**filter** - критерий, по которому выбирать уведомления конкретного приложения.

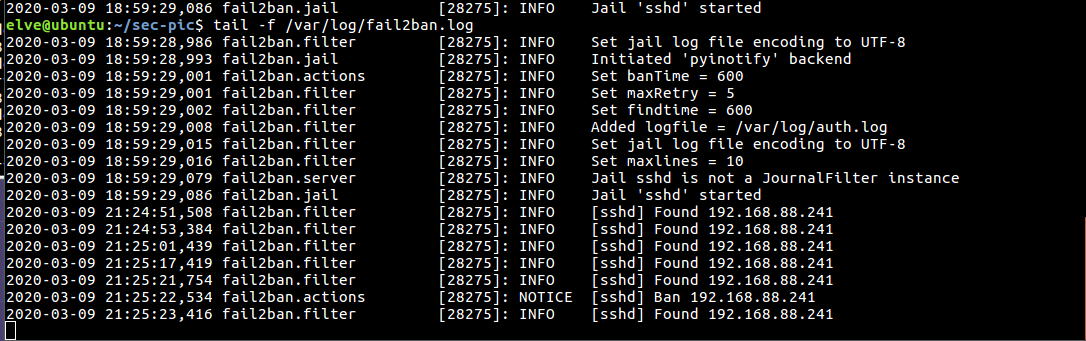
Итак. Файл создан. Теперь перезапускаем сервис и пробуем попасть в бан с соседнего компьютера (он не должен быть добавлен в **ignoreip**, чтобы эксперимент удался ;) ).

Нужно 5 раз неправильно ввести пароль.

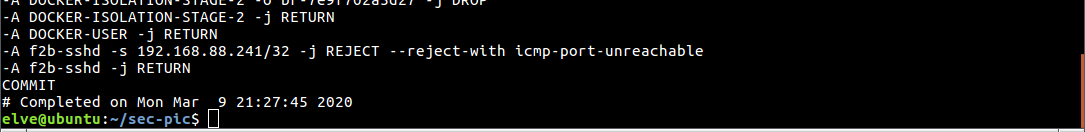
Наблюдать за работой fail2ban можно следующей командой:

**tail -f /var/log/fail2ban.log**

Я пытался подключаться с ip **192.168.88.241** с неправильным паролем и попал в бан.



А вот как это выглядит в файрволе

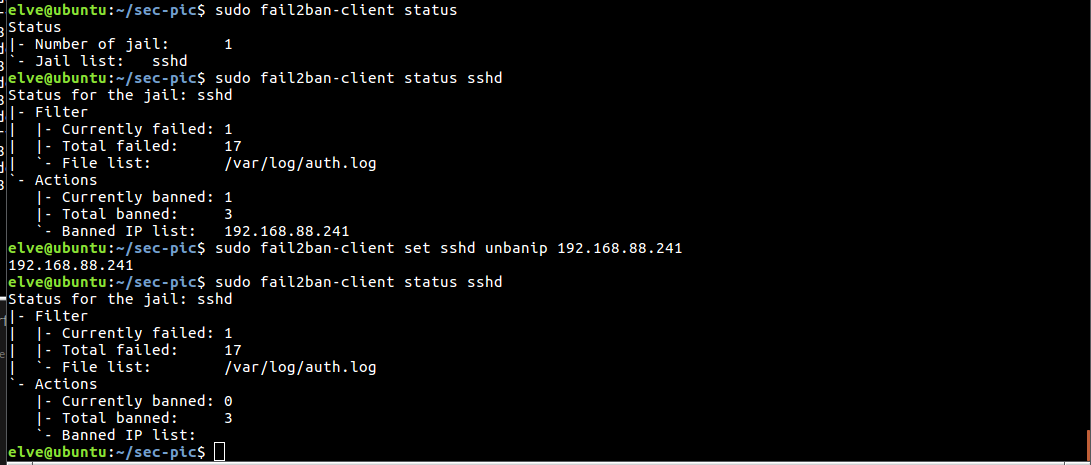


Но и это еще не все. Представь ситуацию, что какой-то адрес попал в бан по ошибке и его нужно оттуда удалить. Есть два варианта:

1. Удалить напрямую из файрвола. Это не совсем корректный путь, т.к. в памяти **fail2ban** адрес остается в списке рецидивистов и в зависимости от настроек может быть забанен снова.

2. Удалить из списка рецидивистов средствами **fail2ban-client**. Это корректный способ.

Рассмотрим работу с **fail2ban-client** на примере скриншота:



Сначала мы смотрим контроль каких приложений у нас активирован:

**sudo fail2ban-client status**

Список в строке **Jail list**

Теперь смотрим список забаненных по конкретному приложению:

**sudo fail2ban-client status sshd**

Ну и удаляем командой:

**sudo fail2ban-client set sshd unbanip наш\_IP**

Все. **Fail2Ban** настроен.