How to Prevent SSH Brute Force Attacks with Fail2Ban on Debian

**[Fail2ban](https://www.fail2ban.org/wiki/index.php/Main_Page" \t "_blank)** is an open-source intrusion prevention system that can be used to prevent brute force attacks and other suspicious malicious attacks. It scans log files (e.g. **/var/log/apache/error\_log**) and bans IP’s that show the malicious signs such as too many password failures, seeking for exploits etc.

Generally Fail2Ban then used to update firewall rules to reject the IP addresses for a specified amount of time, although any arbitrary other **action** (e.g. sending an email, or ejecting CD-ROM tray) could also be configured. Out of the box Fail2Ban comes with pre-configured **filters** for various services (Apache, curier, SSH etc.).

**Install Fail2Ban on Debian**

Login as root user and enter the following command to install Fail2Ban:

**root@server:~# apt-get install fail2ban**

**Backup Fail2Ban Main Configuration File**

All configuration files are found under **/etc/fail2ban** directory. The main configuration file is **/etc/fail2ban/jail.conf**. Its a good idea to take backup of main config file to avoid merges during upgrades. Take local copy of **/etc/fail2ban/jail.conf** file as shown below:

**root@server:~# cp /etc/fail2ban/jail.conf /etc/fail2ban/jail.local**

**Configure Fail2Ban**

Open up **/etc/fasil2ban/jail.local** file in any editor:

**root@server:~# nano /etc/fail2ban/jail.local**

You will find a section called **[Default]**. This section contains the basic set of rules that Fail2Ban will follow. Set the values as per your requirement. Here is my settings:

**[DEFAULT]**

**# "ignoreip" can be an IP address, a CIDR mask or a DNS host**

**ignoreip = 127.0.0.1/8 192.168.1.100/24**

**bantime  = 600**

**maxretry = 3**

**# "backend" specifies the backend used to get files modification. Available**

**# options are "gamin", "polling" and "auto".**

**# yoh: For some reason Debian shipped python-gamin didn't work as expected**

**#      This issue left ToDo, so polling is default backend for now**

**backend = auto**

**#**

**# Destination email address used solely for the interpolations in**

**# jail.{conf,local} configuration files.**

**destemail = root@localhost**

**#**

**ignoreip** – White list your IP address that you trust to prevent blocking from Fail2Ban. You can add multiple addresses separate by a space character.

**bantime** – Number of seconds that a host would be banned if it is caught by Fail2Ban. The default time is **600** seconds (10 minutes). You can increase the time if you like.

**maxretry** – Number of incorrect login attempts before a host is blocked by Fail2Ban.

**Service Configuration**

By default, Fail2Ban contains set of pre-defined filters for various services. So you don’t need to enter any manual entries in the configuration files. All you need to do is just change the values of **enabled** to **true** or **false**, the respective services are automatically watched by Fail2Ban.

Here is sample output of SSH section in **jail.local** file. By default, it is enabled and turned on, so you don’t need to change anything:

**[ssh]**

**enabled  = true**

**port     = ssh**

**filter   = sshd**

**logpath  = /var/log/auth.log**

**maxretry = 6**

**enabled** – This means that the ssh service protection is on. If you want to turn it off, just set to **false**.

**port**– SSH service port

**filter** – It refers to the config file containing the rules that Fail2Ban uses to find matches. By default it is set to sshd that refers to **/etc/fail2ban/filter.d/sshd.conf** file.

**logpath** – The log file for failed login attempts.

**maxretry** – Number of incorrect login attempts before a host is blocked by Fail2Ban.

Once you have changed the configuration, restart Fail2Ban service to save the changes:

**root@server:~# /etc/init.d/fail2ban restart**

You can verify the rules that added by Fail2Ban in iptables using the following command:

**root@server:~# iptables -L**

**Chain INPUT (policy ACCEPT)**

**target     prot opt source               destination**

**fail2ban-ssh  tcp  --  anywhere             anywhere             multiport dports ssh**

**Chain FORWARD (policy ACCEPT)**

**target     prot opt source               destination**

**Chain OUTPUT (policy ACCEPT)**

**target     prot opt source               destination**

**Chain fail2ban-ssh (1 references)**

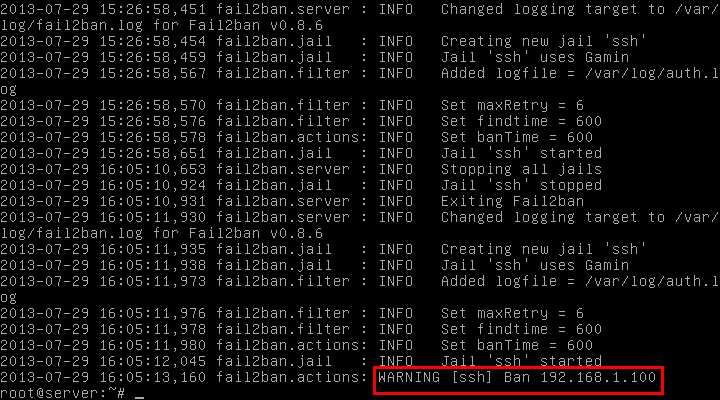
**target     prot opt source               destination**

**RETURN     all  --  anywhere             anywhere**

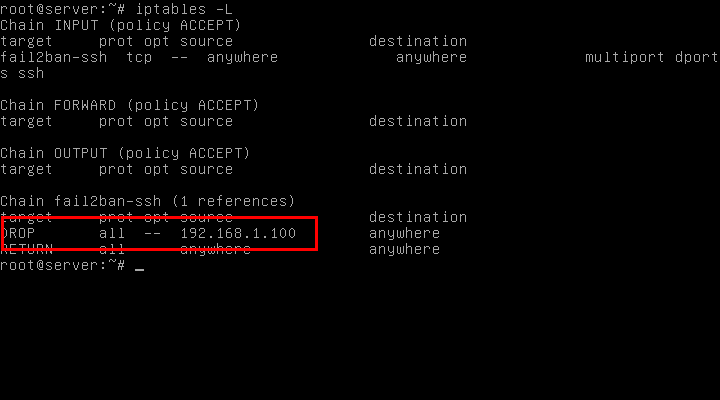
**Testing Fail2Ban**

I have done some failed attempts from my local client to my Debian server to test Fail2Ban. Then I verified the failed login attempts in the **/var/log/fail2ban.log** file:

**root@server:~# cat /var/log/fail2ban.log**

[](https://www.unixmen.com/wp-content/uploads/2013/07/Debian-7-1-nic-internet-bridge-local-repo-Running-Oracle-VM-VirtualBox_003.png)or

**root@server:~# iptables -L**

[](https://www.unixmen.com/wp-content/uploads/2013/07/Debian-7-1-nic-internet-bridge-local-repo-Running-Oracle-VM-VirtualBox_004.png)

As you seen in the above two outputs, my local IP **192.168.1.100** is banned by Fail2Ban.

**Remove Blocked IP Address from Fail2Ban**

If you found a blocked IP and want to unblock it, just enter the following command:

**root@server:~# iptables -D fail2ban-ssh 1**

And restart Fail2Ban service too:

**root@server:~# /etc/init.d/fail2ban restart**

Now you will be able to SSH login from the blocked host.