How To Protect SSH Access with Fail2Ban on RHEL 7

Table of Contents

How To Protect SSH Access with Fail2Ban on RHEL 7

Overview

In this guide we will install fail2ban on a Linux based operating system. Installing and configuring fail2ban is important when your system is accessible from public network.

One of the vulnerable ports or service is "**ssh**" which grants access to the system, even though if the system is configured to grant access only for authorized users with sshd enabled.

This issue can be mitigated with "**Fail2Ban**", wherein automatic rule will be created to block access of unsuccessful login attempts with a specific time frame.

Fail2ban is a **log-parsing** application that **monitors system logs for symptoms of an automated attack**. When an attempted compromise is located, using the defined parameters.

Fail2ban will add a new rule to **iptables**, thus blocking the IP address of the attacker, either for a set / configured amount of time or permanently.

Fail2ban can also alert you through email that an attack is occurring.

Fail2ban is primarily focused on SSH attacks, although it can be further configured to work for any service that uses log files and can be subject to a compromise.

Applies To

- RHEL 7

Pre-Requisites

- Python 2.6 or higher
- epel-release repository is installed on the server.
  - To install, run the command; **yum install epel-release -y**

Package Install – Fail2Ban

After installing "**epel-release**" repository package; install fail2ban-firewalld package, run the command;

yum install fail2ban-firewalld -y

```
[root@mail ~]#
[root@mail ~]# yum install fail2ban-firewalld -y -q
[root@mail ~]#
```

Verify Package Install – Fail2Ban

In order to verify if the package has been installed, you can run the below command; alternatively, you can verify using the command "**yum history**".

rpm -qai fail2ban* | grep -E "Name|\ Install Date"

```
[root@mail ~]#
[root@mail ~]# rpm -qai fail2ban* | grep -E "Name|\Install Date"
Name         : fail2ban-server
Install Date: Wed 17 May 2017 01:44:17 PM IST
Name         : fail2ban-sendmail
Install Date: Wed 17 May 2017 01:44:19 PM IST
Name         : fail2ban-firewalld
Install Date: Wed 17 May 2017 08:45:46 PM IST
[root@mail ~]#
```

Fail2Ban – Configuration

Before we start the service starting let us look into the configuration files and its purpose.

Jail – Configuration Files Types

Fail2ban has 4 configuration file types; that is responsible stored in "**/etc/fail2ban/**" folder. Listed below are the configuration files and its purpose.

Configuration Files	Purpose
fail2ban.conf	Fail2Ban global configuration (such as logging)
filter.d/*.conf	Filters specifying how to detect authentication failures
action.d/*.conf	Actions defining the commands for banning and unbanning of IP address
jail.conf	Jails defining combinations of Filters with Actions.

```
[root@mail ~]#
[root@mail ~]# ll /etc/fail2ban/
total 64
drwxr-xr-x. 2 root root  4096 May 17 13:44 action.d
-rw-r--r--. 1 root root  2328 Dec  9 20:06 fail2ban.conf
drwxr-xr-x. 2 root root     6 Feb 16 00:07 fail2ban.d
drwxr-xr-x. 3 root root  4096 May 17 13:44 filter.d
-rw-r--r--. 1 root root 21284 Feb 16 00:07 jail.conf
drwxr-xr-x. 2 root root    30 May 17 20:45 jail.d
-rw-r--r--. 1 root root  2375 Dec  9 20:06 paths-common.conf
-rw-r--r--. 1 root root   642 Dec  9 20:06 paths-debian.conf
-rw-r--r--. 1 root root  1070 Dec  9 20:06 paths-fedora.conf
-rw-r--r--. 1 root root  1174 Dec  9 20:06 paths-freebsd.conf
-rw-r--r--. 1 root root   975 Dec  9 20:06 paths-opensuse.conf
-rw-r--r--. 1 root root   290 Dec  9 20:06 paths-osx.conf
[root@mail ~]#
```

Jail Configuration Files – Parsing Order

Fail2ban will parse all the configuration files in the following order; first all *.conf and subsequently *.local will be parsed.

Configuration File / Directory	Parsing Order
jail.conf	First Parsing File
jail.d/*.conf	Files in directory; all files with extension **.conf** files are parsed in alphabetical order
jail.local	Next Parsing File
jail.d/*.local	Files in directory; all files with extension **.local** files are parsed in alphabetical order

jail.conf - Configuration Files

***.conf** files are distributed by Fail2Ban by default located under "**/etc/fail2ban/**" folder. It is recommended that ***.conf** files **should remain unchanged to ease upgrades**. If needed, customizations should be provided in ***.local** files.

jail.local – Configuration Files

In **.local** files **specify only the settings that you intend to change** and the rest of the configuration will then come from the corresponding **.conf** file which is parsed first.

Configure – Enable SSH Port Monitoring

In order to monitor **ssh port** and ban hosts that fail to authenticate on the server. The configuration below in the file "**/etc/fail2ban/jail.local**" will ban the IP address for one hour who try to connect on ssh port.

Add the below entry in the fail2ban local configuration into file; **vi /etc/fail2ban/jail.local**

[DEFAULT]

# Ban hosts for one day:
bantime = 86400

Monitoring SSH Service
[sshd]
enabled = true

# Ignore IP
ignoreip = 127.0.0.1/8 192.168.1.1/24 192.168.3.1/24 192.168.7.1/24

# Max Retry Times
maxretry = 3

# Find Number of attempts
findtime = 3600

```
[root@mail ~]#
[root@mail ~]# vi /etc/fail2ban/jail.local
[root@mail ~]#
[root@mail ~]# cat /etc/fail2ban/jail.local
[DEFAULT]

# Ban hosts for one hour:
bantime = 3600

# Monitoring SSH Service
[sshd]
enabled = true
[root@mail ~]#
```

Enable and Start Service – fail2ban

After adding customized configuration the "**jail.local**" file, enable and start "**fail2ban**" service, run the command;

systemctl enable fail2ban; systemctl status fail2ban -l

```
[root@mail ~]#
[root@mail ~]# systemctl enable fail2ban; systemctl status fail2ban -l
ln -s '/usr/lib/systemd/system/fail2ban.service' '/etc/systemd/system/multi-user
.target.wants/fail2ban.service'
fail2ban.service - Fail2Ban Service
   Loaded: loaded (/usr/lib/systemd/system/fail2ban.service; enabled)
   Active: inactive (dead)
     Docs: man:fail2ban(1)

[root@mail ~]#
```

Next step is to start the service, to start run the command;

systemctl start fail2ban; systemctl status fail2ban -l

```
[root@mail ~]#
[root@mail ~]# systemctl start fail2ban; systemctl status fail2ban -l
fail2ban.service - Fail2Ban Service
   Loaded: loaded (/usr/lib/systemd/system/fail2ban.service; enabled)
   Active: active (running) since Thu 2017-05-18 16:39:55 IST; 6min ago
     Docs: man:fail2ban(1)
  Process: 9407 ExecStart=/usr/bin/fail2ban-client -x start (code=exited, status=0/SUCCESS)
 Main PID: 9410 (fail2ban-server)
   CGroup: /system.slice/fail2ban.service
           └─9410 /usr/bin/python2 -s /usr/bin/fail2ban-server -s /var/run/fail2ban/fail2ban.sock -p /var
/run/fail2ban/fail2ban.pid -x -b

May 18 16:39:55 mail.etech.com fail2ban-client[9407]: 2017-05-18 16:39:55,763 fail2ban.server
        [9408]: INFO    Starting Fail2ban v0.9.6
May 18 16:39:55 mail.etech.com fail2ban-client[9407]: 2017-05-18 16:39:55,763 fail2ban.server
        [9408]: INFO    Starting in daemon mode
May 18 16:39:55 mail.etech.com systemd[1]: Started Fail2Ban Service.
[root@mail ~]#
```

View Firewall Rules

After starting fail2ban service, a firewall rule will be automatically be added into "**Direct interface**";

Direct Interface, which enables directly passing rules to iptables, ip6tables and ebtables.

It is primarily intended for use by applications

The direct interface is used by adding the **--direct** option to the **firewall-cmd** command.

firewall-cmd --direct --get-all-rules

```
[root@mail ~]#
[root@mail ~]# firewall-cmd --direct --get-all-rules
ipv4 filter INPUT 0 -p tcp -m multiport --dports ssh -m set --match-set fail2ban-sshd
src -j REJECT --reject-with icmp-port-unreachable
[root@mail ~]#
```

Next, to know the if jail has been configured successfully, run the command;

fail2ban-client status

```
[root@mail ~]#
[root@mail ~]# fail2ban-client status
Status
|- Number of jail:      1
`- Jail list:   sshd
[root@mail ~]#
```

You can also know the status of IP Addresses that have timed out, with "**ipset**" – IP sets administration tool, run the command;

ipset list fail2ban-sshd

```
[root@mail ~]#
[root@mail ~]# ipset list fail2ban-sshd
Name: fail2ban-sshd
Type: hash:ip
Revision: 1
Header: family inet hashsize 1024 maxelem 65536 timeout 3600
Size in memory: 17168
References: 1
Members:
113.122.49.108 timeout 1113
193.201.224.210 timeout 1568
116.31.116.53 timeout 849
58.218.198.159 timeout 743
114.231.15.63 timeout 2653
61.177.172.14 timeout 2992
108.161.134.5 timeout 1972
[root@mail ~]#
```

IP Address Whitelisting

In order to add a IP Address to white-listing, add entry to the "**ignoreip**" attribute, to add additional new IP Address or CIDR separated by space.

ignoreip = 127.0.0.1/8 192.168.1.1/24

Banning IP Address

In order to add a IP Address to ban list, following attributes can to be customized.

Attribute	Purpose
bantime	The length of time in seconds for which an IP is banned. If set to a negative number, the ban will be permanent. The default value of 600 is set to ban an IP for a 10-minute duration.
findtime	The length of time between login attempts before a ban is set. For example, if Fail2ban is set to ban an IP after five (5) failed log-in attempts, those 5 attempts must occur within the set 10-minute findtime limit. The findtime value should be a set number of seconds.
maxretry	How many attempts can be made to access the server from a single IP before a ban is imposed. The default is set to 3.

EMail Alerts

In order to configure email alerts these attributes have to be configured, email notification will sent to **destemail** (recipient).

Attribute	Purpose
destemail	The email address where you would like to receive the emails.
sendername	The name under which the email shows up.
sender	The email address from which Fail2ban will send emails.

fail2ban Client – Command

fail2ban can be managed with command line, "**fail2ban-client**" command; listed below are the various argument's and its purpose.

Command	Purpose
start	Starts the Fail2ban server and jails.
reload	Reloads Fail2ban's configuration files.
reload JAIL NAME	Replaces JAIL with the name of a Fail2ban jail; this will reload the jail.
stop	Terminates the server.
status	Will show the status of the server, and enable jails.
status JAIL NAME	Will show the status of the jail, including any currently-banned IPs.

fail2ban Client – Status

To know the current jail status, run the command;

fail2ban-client status

```
[root@mail ~]#
[root@mail ~]# fail2ban-client status
Status
|- Number of jail:      1
`- Jail list:   sshd
[root@mail ~]#
```

fail2ban Client – Status Jail Name

To know the current jail status of specific jail (name), run the command;

fail2ban-client status sshd

```
[root@mail ~]#
[root@mail ~]# fail2ban-client status sshd
Status for the jail: sshd
|- Filter
|  |- Currently failed: 1
|  |- Total failed:     161
|  `- Journal matches:  _SYSTEMD_UNIT=sshd.service + _COMM=sshd
`- Actions
   |- Currently banned: 8
   |- Total banned:     13
   `- Banned IP list:   58.218.198.159 116.31.116.53 113.122.49.108 193.201.224.210
108.161.134.5 114.231.15.63 61.177.172.14 110.78.137.37
[root@mail ~]#
```

Service Management – fail2ban

By default after installation, fail2ban is not enabled not started automatically, so in the next steps we will enable and start the service and subsequently configure the firewall rules.

Enable Service – fail2ban

To enable daemon and start at OS startup and check the status of the service, run the command;

systemctl enable fail2ban; systemctl status fail2ban -l

```
[root@mail ~]#
[root@mail ~]# systemctl enable fail2ban; systemctl status fail2ban -l
ln -s '/usr/lib/systemd/system/fail2ban.service' '/etc/systemd/system/multi-user
.target.wants/fail2ban.service'
fail2ban.service - Fail2Ban Service
   Loaded: loaded (/usr/lib/systemd/system/fail2ban.service; enabled)
   Active: inactive (dead)
     Docs: man:fail2ban(1)

[root@mail ~]#
```

Start Service – fail2ban

To start the daemon and check the status of the service, run the command;

systemctl start fail2ban; systemctl status fail2ban -l

```
[root@mail ~]#
[root@mail ~]# systemctl start fail2ban; systemctl status fail2ban -l
fail2ban.service - Fail2Ban Service
   Loaded: loaded (/usr/lib/systemd/system/fail2ban.service; enabled)
   Active: active (running) since Thu 2017-05-18 16:39:55 IST; 6min ago
     Docs: man:fail2ban(1)
  Process: 9407 ExecStart=/usr/bin/fail2ban-client -x start (code=exited, status=0/SUCCESS)
 Main PID: 9410 (fail2ban-server)
   CGroup: /system.slice/fail2ban.service
           └─9410 /usr/bin/python2 -s /usr/bin/fail2ban-server -s /var/run/fail2ban/fail2ban.sock -p /var
/run/fail2ban/fail2ban.pid -x -b

May 18 16:39:55 mail.etech.com fail2ban-client[9407]: 2017-05-18 16:39:55,763 fail2ban.server
        [9408]: INFO    Starting Fail2ban v0.9.6
May 18 16:39:55 mail.etech.com fail2ban-client[9407]: 2017-05-18 16:39:55,763 fail2ban.server
        [9408]: INFO    Starting in daemon mode
May 18 16:39:55 mail.etech.com systemd[1]: Started Fail2Ban Service.
[root@mail ~]#
```

Stop Service – fail2ban

To stop the daemon and check the status of the service, run the command;

systemctl stop fail2ban; systemctl status fail2ban -l

```
[root@mail ~]#
[root@mail ~]# systemctl stop fail2ban; systemctl status fail2ban -l
fail2ban.service - Fail2Ban Service
   Loaded: loaded (/usr/lib/systemd/system/fail2ban.service; enabled)
   Active: inactive (dead) since Mon 2017-05-22 11:22:39 IST; 11ms ago
     Docs: man:fail2ban(1)
  Process: 16268 ExecStop=/usr/bin/fail2ban-client stop (code=exited, status=0/SUCCESS)
  Process: 15539 ExecStart=/usr/bin/fail2ban-client -x start (code=exited, status=0/SUCCESS)
 Main PID: 15542 (code=exited, status=0/SUCCESS)

May 22 11:21:01 mail.etech.com fail2ban-client[15539]: 2017-05-22 11:21:01,413 fail2ban.server
 [15540]: INFO    Starting Fail2ban v0.9.6
May 22 11:21:01 mail.etech.com fail2ban-client[15539]: 2017-05-22 11:21:01,413 fail2ban.server
 [15540]: INFO    Starting in daemon mode
May 22 11:21:01 mail.etech.com systemd[1]: Started Fail2Ban Service.
May 22 11:22:37 mail.etech.com systemd[1]: Stopping Fail2Ban Service...
May 22 11:22:38 mail.etech.com fail2ban-client[16268]: Shutdown successful
May 22 11:22:39 mail.etech.com systemd[1]: Stopped Fail2Ban Service.
[root@mail ~]#
```

Restart Service – fail2ban

To restart the daemon and check the status of the service, run the command;

systemctl restart fail2ban; systemctl status fail2ban -l

```
[root@mail ~]#
[root@mail ~]# systemctl restart fail2ban; systemctl status fail2ban -l
fail2ban.service - Fail2Ban Service
   Loaded: loaded (/usr/lib/systemd/system/fail2ban.service; enabled)
   Active: active (running) since Mon 2017-05-22 11:22:52 IST; 6ms ago
     Docs: man:fail2ban(1)
  Process: 16268 ExecStop=/usr/bin/fail2ban-client stop (code=exited, status=0/SUCCESS)
  Process: 16311 ExecStart=/usr/bin/fail2ban-client -x start (code=exited, status=0/SUCCESS)
 Main PID: 16314 (fail2ban-server)
   CGroup: /system.slice/fail2ban.service
           ├─16314 /usr/bin/python2 -s /usr/bin/fail2ban-server -s /var/run/fail2ban/fail2ban.sock -p /var/run/fa
il2ban/fail2ban.pid -x -b
           ├─16321 /bin/sh -c ipset create fail2ban-sshd hash:ip timeout 3600 firewall-cmd --direct --add-rule ip
v4 filter INPUT 0 -p tcp -m multiport --dports ssh -m set --match-set fail2ban-sshd src -j REJECT --reject-with i
cmp-port-unreachable
           └─16323 /bin/sh -c ipset create fail2ban-sshd hash:ip timeout 3600 firewall-cmd --direct --add-rule ip
v4 filter INPUT 0 -p tcp -m multiport --dports ssh -m set --match-set fail2ban-sshd src -j REJECT --reject-with i
cmp-port-unreachable

May 22 11:22:52 mail.etech.com fail2ban-client[16311]: 2017-05-22 11:22:52,513 fail2ban.server
 [16312]: INFO    Starting Fail2ban v0.9.6
May 22 11:22:52 mail.etech.com fail2ban-client[16311]: 2017-05-22 11:22:52,514 fail2ban.server
 [16312]: INFO    Starting in daemon mode
May 22 11:22:52 mail.etech.com systemd[1]: Started Fail2Ban Service.
[root@mail ~]#
```