Rapid7 Blog

How To Protect SSH and Apache Using Fail2Ban on Ubuntu Linux

Rapid7 (/author/rapid7/)

Feb 13, 2017 | 5 min read

Synopsis

Fail2Ban is a free and open source intrusion prevention software tool written in the Python programming language that can be used to protects servers from different kinds of attacks. Fail2Ban works by continuosly monitoring various logs files (Apache, SSH) and running scripts based on them. Mostly it is used to block IP addresses that are trying to breach the system's security. It can be used to block any IP address that are trying to make many illegitimate login attempts. Fail2Ban is set up to block malicious IP address within a time limit defined by administrator. Fail2Ban can be configured to send email notifications when someone's attacking your server. Main purpose of Fail2ban is to scans log files for various services, such as SSH, FTP, SMTP, Apache and block the IP address that makes too many password failures.

Here, we will learn how to install and configure Fail2ban to protect SSH and Apache service from brute force login attacks.

System Requirements

- Newly deployed Ubuntu 16.04 server.
- Apache server with basic password authentication configured on your server.
- A static IP address 192.168.15.189 is configured on your server.

Prepare the System for Deployment

Before starting with the Fail2Ban, your system should be up to date and all installed software is running the latest version.

First, log in to root user and update your system by running the following command:

```
apt-get update -y
apt-get upgrade -y
```

Install Fail2Ban

By default Fail2ban is available in Ubuntu 16.04 default package repository. So you can easily install it by just running the following command:

```
sudo apt-get install fail2ban
```

Once installation is complete, you can proceed to configuring Fail2ban.

Configure Fail2Ban

By default Fail2ban keeps all the configuration files in `/etc/fail2ban/` directory. The main configuration file is `jail.conf` , it contains a set of pre-defined filters. It is recommended that you should not modify `jail.conf` itself, but override it by creating a new configuration file `jail.local` inside `/etc/fail2ban/` directory.

So let's create a new `jail.local` file for Apache and SSH:

```
sudo nano /etc/fail2ban/jail.local
```

Add the following lines:

```
##To block failed login attempts use the below jail.
[apache]
enabled = true
port = http,https
filter = apache-auth
logpath = /var/log/apache2/*error.log
maxretry = 3
bantime = 600
ignoreip = 192.168.15.189

##To block the remote host that is trying to request suspicious URLs, use th
[apache-overflows]
enabled = true
port = http,https
filter = apache-overflows
logpath = /var/log/apache2/*error.log
maxretry = 3
bantime = 600
ignoreip = 192.168.15.189

##To block the remote host that is trying to search for scripts on the websi
[apache-noscript]
enabled = true
port = http,https
filter = apache-noscript
logpath = /var/log/apache2/*error.log
maxretry = 3
bantime = 600
ignoreip = 192.168.15.189

##To block the remote host that is trying to request malicious bot, use belo
[apache-badbots]
enabled = true
```

```
port = http,https
filter = apache-badbots
logpath = /var/log/apache2/*error.log
maxretry = 3
bantime = 600
ignoreip = 192.168.15.189

##To stop DOS attack from remote host. [http-get-dos]
enabled = true
port = http,https
filter = http-get-dos
logpath = /var/log/apache*/access.log
maxretry = 400
findtime = 400
bantime = 200
ignoreip = 192.168.15.189
action = iptables[name=HTTP, port=http, protocol=tcp]

##To block the failed login attempts on the SSH server, use the below jail.
[ssh]
enabled = true
port = ssh
filter = sshd
logpath = /var/log/auth.log
maxretry = 3
bantime = 600
ignoreip = 192.168.15.189
```

Save the file when you are finished.

Next, you will also need to create the filter file `/etc/fail2ban/filters.d/http-get-dos.conf`.

```
sudo nano /etc/fail2ban/filters.d/http-get-dos.conf
```

Add the following contents:

```
# Fail2Ban configuration file
[Definition]

# Option: failregex
# Note: This regex will match any GET entry in your logs, so basically all v
# You should set up in the jail.conf file, the maxretry and findtime careful
failregex = ^<HOST> -.*"(GET|POST).*
# Option: ignoreregex
ignoreregex =
```

Save the file and restart the fail2ban service:

```
sudo systemctl restart fail2ban
```

Brief description of each configuration options are here:

logpath : Name of the logfile that fail2ban checks for failed login attempts.
maxretry : Maximum number of failed login attempts before a host is blocked by fail2ban.
bantime : Specifies the number of seconds that a remote host will be blocked by Fail2ban.
findtime : The time period in seconds in which we're counting "retries".
ignoreip : This is the list of IP addresses that can not be blocked by Fail2ban.

You can view the rules added by Fail2Ban using the following command:

```
sudo iptables -L
```

You should see all the rules as below:

```
Chain fail2ban-apache (1 references)
target prot opt source destination
RETURN all -- anywhere anywhere

Chain fail2ban-apache-badbots (1 references)
target prot opt source destination
RETURN all -- anywhere anywhere

Chain fail2ban-apache-noscript (1 references)
target prot opt source destination
RETURN all -- anywhere anywhere

Chain fail2ban-apache-overflows (1 references)
target prot opt source destination
RETURN all -- anywhere anywhere

Chain fail2ban-http-get-dos (1 references)
target     prot opt source               destination
RETURN     all  --  anywhere             anywhere
Chain fail2ban-ssh (1 references)
target prot opt source destination
RETURN all -- anywhere anywhere
```

You can also list out all the activated jails by running the following command:

```
sudo fail2ban-client status
```

You should see the following output:

```
Status
|- Number of jail: 5
`- Jail list: apache, apache-overflows, apache-badbots, apache-noscript, htt
```

Test Fail2Ban

Once everything is set up properly, it's time to test Fail2Ban.

Test Fail2Ban for Apache Failed Login Attempts

On the remote machine, open your web browser and type the URL `http://your-apacheserver-ip` , you should be asked for authentication. Enter the wrong username and password repeatedly. Once you have reached the failed login limit, you should be blocked and unable to access the Apache web server for `600` seconds.

On the Fail2Ban server machine, check the banning status of Fail2Ban with the following command:

```
sudo fail2ban-client status apache
```

You should see your remote host IP address being blocked by Fail2Ban:

```
Status for the jail: apache
|- filter
| |- File list: /var/log/apache2/*error.log
| |- Currently failed: 0
| `- Total failed: 3
`- action
|- Currently banned: 1
| `- IP list: 192.168.15.196
`- Total banned: 1
```

Test Fail2Ban for Apache DOS Attack

You can use ab (Apache Bench-mark tool) to test if it's really working.
On the remote machine, open your terminal and run the following command:

```
ab -n 1000 -c 20 http://192.168.15.189/
```

The above command will send 1000 page-loads in 20 concurrent connections against your web server. When you have reached the limit, you should be blocked for 200 seconds.

On the Fail2Ban server machine, take a look in your `/var/log/fail2ban.log` file you should see the following output:

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

Output:

```
2017-01-31 20:51:08,417 fail2ban.actions: WARNING [http-get-dos] Ban 192.168
```

You can also verify fail2ban banning status with the following command:

```
sudo fail2ban-client status http-get-dos
```

You should see that your remote host IP address being blocked by Fail2Ban:

```
Status for the jail: http-get-dos
|- filter
| |- File list: /var/log/apache2/access.log
| |- Currently failed: 0
| `- Total failed: 1000
`- action
    |- Currently banned: 1
    | `- IP list: 192.168.15.196
    `- Total banned: 1
```

Test Fail2Ban for SSH Failed Login Attempts

On the remote machine, open your command line interface and try to ssh to the server IP address:

```
ssh 192.168.15.189
```

You should be asked to enter password. Enter the wrong password repeatedly. Once you have reached the failed login limit, you should be blocked for `600` seconds.

```
root@192.168.15.189's password:
Permission denied, please try again.
root@192.168.15.189's password:
Permission denied, please try again.
root@192.168.15.189's password:
Permission denied, please try again.
root@192.168.15.189's password:
ssh: connect to host 192.168.15.189 port 22: Connection refused
```

On the Fail2Ban server machine, check the banning status of Fail2Ban with the following command:

```
sudo fail2ban-client status ssh
```

You should see that your remote host IP address being blocked by Fail2Ban:

```
Status for the jail: ssh |- filter
| |- File list: /var/log/auth.log
| |- Currently failed: 1
| `- Total failed: 4
`- action
    |- Currently banned: 1
    | `- IP list: 192.168.15.196
    `- Total banned: 1
```

If you want to unban the IP address of the remote host before the banning time limit expires, then run the following command on the server machine:

```
sudo fail2ban-client set ssh unbanip 192.168.15.196
sudo fail2ban-client set apache unbanip 192.168.15.196
```

Where, `192.168.15.196` is the IP address of the remote machine.

References

- Fail2Ban Manual (http://www.fail2ban.org/wiki/index.php/MANUAL_0_8)
- Fail2Ban Apache (https://github.com/miniwark/miniwark-howtos/wiki/Fail2Ban-setup-for-Apache)

POST STATS

9

POST TAGS

AUTOMATION AND ORCHESTRATION (/TAG/AUTOMATION-AND-ORCHESTRATION/) KOMAND (/TAG/KOMAND/)

SHARING IS CARING

(https://www.linkedin.com/shareArticle?mini=true&url=https://blog.rapid7.com/2017/02/13/how-to-protect-ssh-and-apache-using-fail2ban-on-ubuntu-linux/&title=How To Protect SSH and Apache Using Fail2Ban on Ubuntu Linux&summary=Synopsis Fail2Ban is a free and open source intrusion prevention software tool written in the Python programming language that can be used to protects)
(https://twitter.com/intent/tweet?text=How To Protect SSH and Apache Using Fail2Ban on Ubuntu Linux&url=https://blog.rapid7.com/2017/02/13/how-to-protect-ssh-and-apache-using-fail2ban-on-ubuntu-linux/)
(https://www.facebook.com/sharer/sharer.php?u=https://blog.rapid7.com/2017/02/13/how-to-protect-ssh-and-apache-using-fail2ban-on-ubuntu-linux/)

AUTHOR

Rapid7

(/author/rapid7/)

(/author/rapid7/) View Rapid7's Posts (/author/rapid7/)

Adding **extra** lines to the file sudo nano /etc/fail2ban/jail.local by copy-pasting the text that contains the new rules like

```
##To block failed login attempts use the below jail. [apache]
enabled = true
port = http,https
filter = apache-auth
logpath = /var/log/apache2/*error.log
maxretry = 3
bantime = 600
```

`ignoreip = 192.168.15.189` mentioned above, generates an startup error of fail2ban when you check with `sudo systemctl status fail2ban.service` if everything is running.

My server stated

```
Fail2ban.service - Fail2Ban Service
Loaded: loaded (/lib/systemd/system/fail2ban.service; enabled; vendor preset: enabled)
Active: failed (Result: exit-code) since Tue 2019-09-03 19:38:22 CEST; 5s ago
Docs: man:fail2ban(1)
Process: 9816 ExecStop=/usr/bin/fail2ban-client stop (code=exited, status=0/SUCCESS)
```

see more

Reply • Share ›

Nico Vink • 15 days ago

In my Ubuntu 18.04 the filters directory is named filter.d not filters.d So the text needs to be "Next, you will also need to create the filter file /etc/fail2ban/filter.d/http-get-dos.conf."

Reply • Share ›

mem • 7 months ago

@asif thank you for the correction

Reply • Share ›

Asif • a year ago

If you copied above code and restart didn't work for you, thats probably cause of following line. move [http-get-dos] to a new line.
##To stop DOS attack from remote host. [http-get-dos]

Reply • Share ›

mem ➚ Asif • 7 months ago

thanks for the correction

Reply • Share ›

M B • a year ago

sudo nano /etc/fail2ban/filters.d/http-get-dos.conf should be sudo nano /etc/fail2ban/filter.d/http-get-dos.conf. Atleast on my machine anyway. Either way thanks for the article, it was helpful

Reply • Share ›

Eddy Valencia • a year ago • edited

Thank you for this, but like the first comment, You have to do an enter for '[http-get-dos]', then restart

fail2ban, and it's it, fail2ban works fine!

Reply • Share ›

Vijay Kumar • a year ago

Hello,

Please check in screenshots

wrong command is mentioned

systemctl restart fal2ban

you have mentioned fal2ban in stead of fail2ban

Reply • Share ›

김택훈 • 2 years ago

Thanks for your article. It helped me a lot.

But please correct some configuration in your article about jail.local configuration file. The section for DDOS, which is '[http-get-dos]', is commented out mistakenly. That makes fail2ban goes wrong in my case.

Thanks!

Reply • Share ›