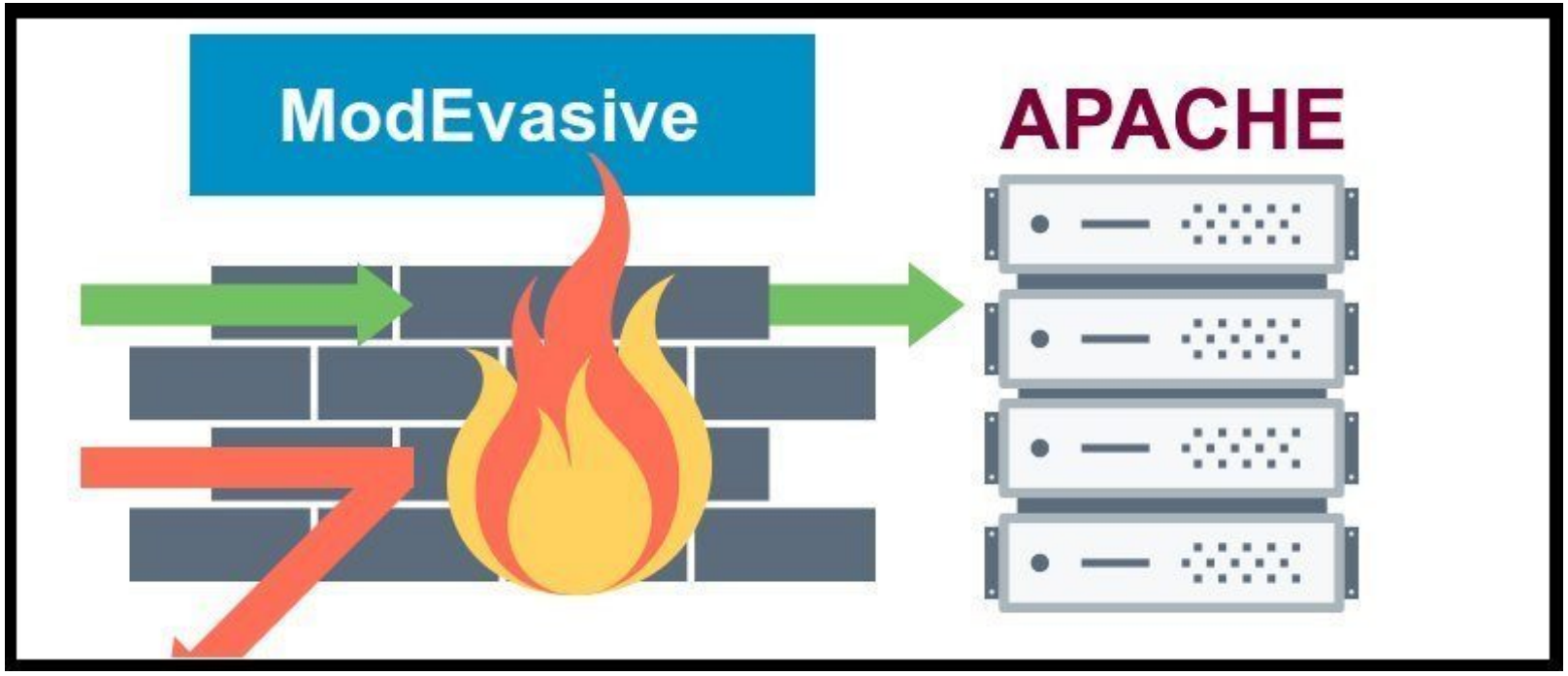
How to Protect Against DDoS with Mod_evasive on Apache Server

By Hitesh Jethva – Posted on Nov 26, 2015 in Linux



This article is part of the Apache Server Guide series:

- Securing Apache on Ubuntu – Part 1
- Securing Apache on Ubuntu – Part 2
- Optimizing Apache Performance – Part 1
- Optimizing Apache Performance – Part 2
- Setting Up Name-Based Virtualhost Apache
- Setting Up IP and Port-Based Virtualhost in Apache
- How to Set Up the Password Protect Web Directory in Apache
- Setting up Apache Server with SSL Support on Ubuntu
- Setting Up Fail2ban to Protect Apache from a DDOS Attack
- How to Set Up Webdav with Apache on Ubuntu
- Monitor Apache Web Server Using Mod_status
- How to Protect Against DDoS with Mod_evasive on Apache Server

Mod_evasive is an Apache module that provides evasive action in the event of an HTTP DoS or DDoS attack or brute force attack. mod_evasive presently reports malicious activity via email and syslog. The mod_evasive module works by creating an internal dynamic hash table of IP addresses and URIs and denying any single IP address from any of the following conditions:

- Requesting the same page more than a few times per second

- Making more than 50 concurrent requests on the same child per second
- Making any requests while temporarily blacklisted (on a blocking list)

In this tutorial I will discuss how to install, configure and use mod_evasive on your Apache server. This tutorial uses a Ubuntu 14.04 server.

Installing mod_evasive

First, make sure Apache server is installed and running.

Next, you can install mod_evasive module by running:

```
sudo apt-get install libapache2-mod-evasive
```

After installing mod_evasive, you can verify this module by running the following commands:

```
sudo apachectl -M | grep evasive
```

If mod_evasive is enabled, you will see the following output:

```
evasive20_module (shared)
```

Configure Mod_evasive

The mod_evasive module reads its configuration from “/etc/apache2/mods-enabled/evasive.conf.” You can easily customize the mod_evasive module through the “evasive.conf” configuration file. By default, mod_evasive configuration options are disabled, so you will need to enable them first. To do this, edit the “evasive.conf” file:

```
sudo nano /etc/apache2/mods-enabled/evasive.conf
```

Remove `#` from the following lines:

```
    DOSHashTableSize    3097
    DOSPageCount        2
    DOSSiteCount        50
    DOSPageInterval     1
    DOSSiteInterval     1
    DOSBlockingPeriod   10
```

```
    DOSEmailNotify      mail@yourdomain.com
    DOSLogDir           "/var/log/apache2/"
```

Save the file and restart Apache for your changes to take effect:

```
sudo /etc/init.d/apache2 restart
```

You can change the above values according to the amount and type of traffic that your web server needs to handle.

`DOSHashTableSize :` This directive specifies how mod_evasive keeps track of who's accessing what. Increasing this number will provide a faster lookup of the sites that the client has visited in the past.

`DOSPageCount :` This directive specifies how many identical requests to a specific URI a visitor can make over the DOSPageInterval interval.

`DOSSiteCount :` This is similar to DOSPageCount but corresponds to how many requests overall a visitor can make to your site over the DOSSiteInterval interval.

`DOSBlockingPeriod :` If a visitor exceeds the limits set by DOSSPageCount or DOSSiteCount, his IP will be blocked during the DOSBlockingPeriod amount of time. During this interval, he will receive a

403 (Forbidden) error.

`DOSEmailNotify :` An email will be sent to the email address specified whenever an IP address is blacklisted.

`DOSLogDir :` This directive specifies the location of the log directory.

Testing Mod_evasive

Now it's time to test whether the mod_evasive module is working or not. You can do this by using a perl script "test.pl" located in the "/usr/share/doc/libapache2-mod-evasive/examples/" directory.

You can execute the script by running the following command:

```
sudo perl /usr/share/doc/libapache2-mod-evasive/examples/test.pl
```

You should see the following output:

The script makes 100 requests to your web server. The 403 response code indicates access is denied by the web server.

Conclusion

mod_evasive is a very important tool to secure an Apache web server against several threats. You can experiment with mod_evasive ano different options in a testing environment. If you have any questions, you can write them in the comment box below.

Is this article useful? Yes No

Ebooks

The Complete Beginner's Guide to Linux Mint

The Complete Beginner's Guide to Ubuntu 18.04

Linux for Beginners

More ebooks »»

Comments (1)

Previous story
‹ Ask the Experts: What We Used to Stay Productive

Next story
Best Captcha Plugins for WordPress ›

Related Posts

How to Speed Up Your Linux PC

How to Upgrade a Raspberry Pi to Raspbian Buster

8 of The Best Linux Distros in 2019

What Is /dev/null in Linux?

How to Use Topgrade to Easily Upgrade Your Linux System

How to Build a DIY Wireless Printer with a Raspberry Pi

Th Most Handy du (Disk Usage) Commands in Linux

How to Install Snap Applications in Arch Linux