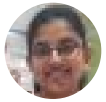
How To Protect Against DoS and DDoS with mod_evasive for Apache on CentOS 7

Posted July 30, 2015 175.8k SECURITY APACHE CENTOS



By Veena K John
Become an author

Introduction

The mod_evasive Apache module, formerly known as mod_dosevasive, helps protect against DoS, DDoS (Distributed Denial of Service), and brute force attacks on the Apache web server. It can provide evasive action during attacks and report abuses via email and syslog facilities. The module works by creating an internal dynamic table of IP addresses and URIs as well as denying any single IP address from any of the following:

- 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

If any of the above conditions are met, a 403 response is sent and the IP address is logged. Optionally, an email notification can be sent to the server owner or a system command can be run to block the IP address.

In this tutorial, we will discuss how to install, configure, and use mod_evasive on your server.

Prerequisites

Before you get started with this tutorial, you should have the following:

- CentOS 7 64-bit Droplet (works with CentOS 6 as well)
- Non-root user with sudo privileges. To setup a user of this type, follow the Initial Server Setup with CentOS 7 tutorial. All commands will be run as this user.
- Apache web server running on the Droplet. To install Apache please follow Step #1 of the How To Install Linux, Apache, MySQL, PHP (LAMP) stack on CentOS article.

Step 1 — Installing mod_evasive

In this section, we will be installing the packages required for mod_evasive to function and finally install mod_evasive.

First, we need to install the EPEL (Extra Packages for Enterprise Linux) yum repository on the server. EPEL is a Fedora Special Interest Group that creates, maintains, and manages a high quality set of open source add-on software packages for Enterprise Linux. Run the following command to install and enable the EPEL repository on your server:

On CentOS 7:

```
$ sudo rpm -ivh http://dl.fedoraproject.org/pub/epel/7/x86_64/e/epel-release-7-5.noarch.rpm
```

On CentOS 6:

```
$ sudo rpm -ivh http://download.fedoraproject.org/pub/epel/6/x86_64/epel-release-6-8.noarch.rpm
```

Let us verify that the EPEL repo is enabled using:

```
$ sudo yum repolist
```

If enabled, you will see the following repo listed in the output:

```
epel/x86_64                                                            Extra Packages for Enterprise Linux 7 - x86_64
```

Now, let us protect the base packages from EPEL using the yum plugin **protectbase**.

```
$ sudo yum install yum-plugin-protectbase.noarch -y
```

The purpose of the **protectbase** plugin is to protect certain yum repositories from updates from other repositories. Packages in the protected repositories will not be updated or overridden by packages in non-protected repositories even if the non-protected repo has a later version.

Now we are ready to install mod_evasive module. Run the following command to install it:

```
$ sudo yum install mod_evasive -y
```

Step 2 — Verifying the Installation

Now that mod_evasive is installed, let's verify that configuration file has been installed and that the module is being loaded.

During installation, the mod_evasive configuration file `/etc/httpd/conf.d/mod_evasive.conf` was added. To verify this run:

```
$ sudo ls -al /etc/httpd/conf.d/mod_evasive.conf
```

Output should look similar to:

```
-rw-r--r-- 1 root root 3473 Jul 21 01:41 /etc/httpd/conf.d/mod_evasive.conf
```

By default, the following `LoadModule` line will be added to the top of configuration file `mod_evasive.conf`. Open the file and add the line if it is not already present. This line tells the Apache web server to load and use the mod_evasive module.

On CentOS 7, the line should read as follows:

/etc/httpd/conf.d/mod_evasive.conf

```
LoadModule evasive20_module modules/mod_evasive24.so
```

On CentOS 6, the line should be as follows:

/etc/httpd/conf.d/mod_evasive.conf

```
LoadModule evasive20_module modules/mod_evasive20.so
```

Let us list the modules loaded for the Apache web server and look for mod_evasive:

```
$ sudo  httpd -M | grep evasive
```

The output should show:

```
 evasive20_module (shared)
```

Step 3 — Configuring mod_evasive

Now that the installation is complete and verified, let us look into the configuration of the module. mod_evasive can be easily customized through the `mod_evasive.conf` configuration file. We will discuss some of the configuration parameters in this tutorial. Please refer to the configuration file for information on all the parameters — it contains a description of each parameter.

One of the configuration options you need to change is `DOSEmailNotify`. This is a very useful directive. If this value is set, an email will be sent to the email address specified whenever an IP address is blacklisted. The email body will show `mod_evasive HTTP Blacklisted 111.111.111.111`

For example, if you want to send mod_evasive alerts to say, sammy@example.com, edit the file:

```
$ sudo nano /etc/httpd/conf.d/mod_evasive.conf
```

Uncomment the `DOSEmailNotify` line by removing the `#` in front of the line, and change the email address to yours:

/etc/httpd/conf.d/mod_evasive.conf

```
DOSEmailNotify    sammy@example.com
```

> **Note:** mod_evasive uses `/bin/mail` for sending email alerts. You need to have a mail server installed and working, please refer to this tutorial for information on how to set up a simple mail server so that email notifications work.

Another parameter you might want to set is `DOSWhitelist`. Using this option, IP addresses of trusted clients can be added to the whitelist to ensure they are never denied. The purpose of whitelisting is to protect software, scripts, local search bots, or other automated tools from being denied for requesting large amounts of data from the server.

To whitelist an IP address, for example 111.111.111.111, add an entry to the configuration file like this:

/etc/httpd/conf.d/mod_evasive.conf

```
DOSWhitelist    111.111.111.111
```

Wildcards can be used on up to the last 3 octets of the IP address if necessary.

To whitelist multiple IP addresses from different IP ranges, you can add separate DOSWhitelist lines in the configuration file like this:

/etc/httpd/conf.d/mod_evasive.conf

```
DOSWhitelist    111.111.111.111
DOSWhitelist    222.222.222.222
```

`DOSPageCount` and `DOSSiteCount` are two other parameters recommended to be changed to less aggressive values to avoid clients getting blocked unnecessarily.

`DOSPageCount` is the limit for the number of requests for the same page per page interval (usually set to one second) by an IP address. Once the threshold for that interval has been exceeded, the IP address of the client will be added to the blocked list. The default value is set quite low at 2. You can change it to a higher value, say 20, by editing the following in `/etc/httpd/conf.d/mod_evasive.conf`:

/etc/httpd/conf.d/mod_evasive.conf

```
DOSPageCount 20
```

`DOSSiteCount` is the limit for the total number of requests for the same website by an IP address per site interval (defaults to 1 second). To change it to a larger value such as 100 seconds:

/etc/httpd/conf.d/mod_evasive.conf

```
DOSSiteCount 100
```

There are a few other parameters you can change to achieve better performance.

One is `DOSBlockingPeriod`, which is the amount of time (in seconds) that a client (IP address) will be blocked for if they are added to the blocked list. During this time, all subsequent requests from the client will result in a 403 (Forbidden) error and the timer being reset (defaults to 10 seconds).

For example, if you want to increase the blocking period to 300 seconds:

/etc/httpd/conf.d/mod_evasive.conf

```
DOSBlockingPeriod    300
```

Another is `DOSLogDir` which refers to the temporary directory used by mod_evasive. By default `/tmp` will be used for a locking mechanism, which opens some security issues if your system is open to shell users. In the event you have non-privileged shell users, you will want to create a directory writeable only to the user Apache is running as (usually **apache**) then set this parameter in your mod_evasive.conf file.

For example, to set the directory used by mod_evasive to `/var/log/mod_evasive`, create the directory using:

```
$ sudo mkdir /var/log/mod_evasive
```

Then set the ownership to `apache` user:

```
$ sudo chown -R apache:apache /var/log/mod_evasive
```

Now edit the mod_evasive configuration and change the directory as follows:

/etc/httpd/conf.d/mod_evasive.conf

```
DOSLogDir           "/var/log/mod_evasive"
```

Another parameter is `DOSSystemCommand`. If a value is set, the command specified will be executed whenever an IP address is blacklisted. Using this parameter, you can integrate mod_evasive with the firewall installed on your server or a shell script and block the IP addresses blacklisted by mod_evasive in the firewall.

Step 4 — Loading the mod_evasive Module

Once we have made the changes in the configuration file, we need to restart the Apache web server for them to take effect. Run the following command to restart Apache.

On CentOS 7:

```
$ sudo systemctl restart httpd.service
```

On CentOS6:

```
$ sudo service httpd restart
```

Note: Please note that mod_evasive appears to conflict with the FrontPage Server Extensions. You might also want to check your Apache web server settings to make sure mod_evasive is able to function well. Suggested Apache tweaks are to have a very high value for MaxRequestsPerChild but not unlimited (A value of zero implies unlimited) and to have KeepAlive enabled with KeepAliveTimeout set reasonably long.

Step 5 — Testing mod_evasive

Let us do a short test to see if the module is working correctly. We will be using a perl script **test.pl** written by mod_evasive developers. To execute the script, we need to first install perl package on the server using:

```
$ sudo yum install -y perl
```

The test script is installed with mod_evasive at the following location:

```
/usr/share/doc/mod_evasive-1.10.1/test.pl
```

By default, the test script requests the same page from your Apache web server 100 times in a row to trigger mod_evasive. In the last section, we modified mod_evasive to be more tolerant of requests per second to the same page. We need to change the script to 200 requests in a row instead of 100 to make sure we trigger all of mod_evasive's notification methods.

Edit /usr/share/doc/mod_evasive-1.10.1/test.pl:

```
$ sudo nano /usr/share/doc/mod_evasive-1.10.1/test.pl
```

Find the following line:

/usr/share/doc/mod_evasive-1.10.1/test.pl

```
for(0..100) {
```

Replace 100 with 200:

/usr/share/doc/mod_evasive-1.10.1/test.pl

```
for(0..200) {
```

Save and exit.

To execute the script, run:

```
$ sudo perl /usr/share/doc/mod_evasive-1.10.1/test.pl
```

You should see output similar to:

```
HTTP/1.1 403 Forbidden
HTTP/1.1 403 Forbidden
HTTP/1.1 403 Forbidden
HTTP/1.1 403 Forbidden
HTTP/1.1 403 Forbidden
...
```

The script makes 100 requests to your web server. the 403 response code indicates access is denied by the web server. mod_evasive also logs to syslog when the IP address is blocked. Check the log file using:

```
$ sudo tailf /var/log/messages
```

It should show a line similar to:

```
Jul 29 00:11:18 servername mod_evasive[18290]: Blacklisting address 127.0.0.1: possible DoS attack.
```

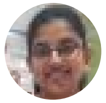
indicating the IP address is blocked by mod_evasive.

If you have configured mod_evasive to send email alerts when an IP is blocked, you will have an email in your inbox with the following content:

```
mod_evasive HTTP Blacklisted 127.0.0.1
```

Conclusion

mod_evasive is great at fending off single server, scripted attacks as well as distributed attacks. However, it is only useful to the point of your server's total bandwidth and processor capacity for processing and responding to invalid requests. For this reason, it is a good idea to integrate this module with your server firewall for maximum protection. Without a really good infrastructure and a firewall in place, a heavy DDoS might still take you offline. If an attack is very heavy and persistent, you might need to move to a hardware-based DDoS mitigation solution.



By Veena K John

Upvote (9) Subscribe Share

Editor: Tammy Fox

Related Tutorials

How To Set Up Apache with a Free Signed SSL Certificate on a VPS

How To Install phpMyAdmin From Source on Debian 10

How To Create a Self-Signed SSL Certificate for Apache in Debian 10

How To Secure Apache with Let's Encrypt on Debian 10

How To Use Certbot Standalone Mode to Retrieve Let's Encrypt SSL Certificates on Debian 10

15 Comments

B *I*

Leave a comment...

Log In to Comment

ruwille *August 11, 2015*

0 How many times i tried to use mod_evasive to protect my server against DDOS, it didn't work. I will try this now as my last hope. Good work btw.

garciam44 *March 13, 2016*

1 Any idea how I can test to make sure mod_evasive is functioning properly?

When I execute the script (sudo perl /usr/share/doc/mod_evasive-1.10.1/test.pl) I get the following response:

HTTP/1.1 301 Moved Permanently

and it doesn't trigger a response.

I have virtual hosts setup (Apache 2.4.6), but otherwise nothing fancy.

Log:

```
DHCPDISCOVER on eth1 to 255.255.255.255 port interval
<warn> (eth1): DHCPv4 request timed out.
<info> (eth1): DHCPv4 state changed unknown -> timeout
<info> (eth1): canceled DHCP transaction, DHCP client pid
<info> (eth1): DHCPv4 state changed timeout -> done
<info> (eth1): device state change: ip-config -> failed (reason 'ip-config-unavailable')
<info> Disabling autoconnect for connection 'Wired connection 1'.
<warn> (eth1): Activation: failed for connection 'Wired connection 1'
<info> (eth1): device state change: failed -> disconnected (reason 'none')
avahi-daemon[469]: Withdrawing address record for on eth1.
```

> KeitelDOG *August 27, 2018*
>
> I get the same HTTP/1.1 301 Moved Permanently in my staging server.
> I'm using 2 VirtualHost, one for port 80 and one for port 443 for SSL. Have you resolved this?

HostLittle *June 16, 2016*

Very good tutorial, I went ahead and implemented some of the techniques on my own website. https://hostlittle.com

techspecx *September 27, 2016*

I put in

sudo rpm -ivh http://dl.fedoraproject.org/pub/epel/7/x86_64/e/epel-release-7-5.noarch.rpm
and I get the following:

Retrieving http://dl.fedoraproject.org/pub/epel/7/x86_64/e/epel-release-7-5.noarch.rpm
curl: (22) The requested URL returned error: 404 Not Found
error: skipping http://dl.fedoraproject.org/pub/epel/7/x86_64/e/epel-release-7-5.noarch.rpm - transfer failed

Could anyone update this article with the correct links so we could have a install of mod_security? Thanks.

campos6 *January 10, 2017*

0

Just look int main site for the version: http://dl.fedoraproject.org/pub/epel/7/x86_64/e/
At this reply the epel its in 7-8. So try:

```
rpm -ivh http://dl.fedoraproject.org/pub/epel/7/x86_64/e/epel-release-7-8.noarch.rpm
```

salasiapro *March 1, 2017*

0

Thanks. It works.
I must agree with you the default values are too low for an average busy server.

giuseppe889156420286953667 *March 7, 2017*

0

Hello, thank you for this article very comprensible.
Is there a way to receive the mail alert with a subject filled in?

Thx in advance
Giuseppe

giuseppe889156420286953667 *March 8, 2017*

0

solved using DOSSystemCommand

yvj9391 *April 5, 2017*

0

Hi,
In your evasive conf file you mentioned “DOSSiteCount 100”, as per my understanding, it will serve first 100 requests and will gives response as 403 for the requests beyond that for page interval 1 sec. But, here, you are getting HTTP 403 response from starting. I didn’t understand it ?

yvj9391 *April 5, 2017*

0

Is there any relationship between "DOSSiteCount" and "DOSPageCount" ?

filippo.lazzarini *June 7, 2017*

0

At step 3:
"DOSSiteCount is the limit for the total number of requests for the same website by an IP address per site interval (defaults to 1 second). To change it to a larger value such as 100 seconds"

I think that the words "seconds" should be removed, because it's not the correct unit of measure

MadhurAsati *May 4, 2018*

0 Thanks for the great article.
I have successfully installed Mod*Evasive on my sever,However i am not getting mails for the blocked email*ids.
Can any one help me on this.!

Thanks in advance.

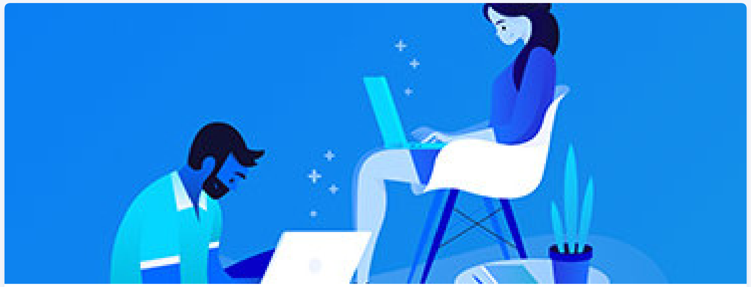
adminea625fd195c0072cd5de7 *November 15, 2018*

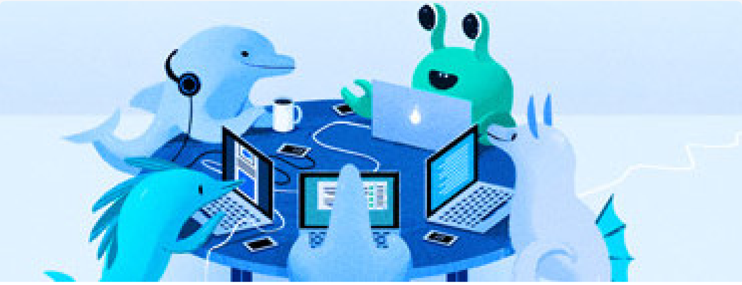
0 Does anybody know how to do this:

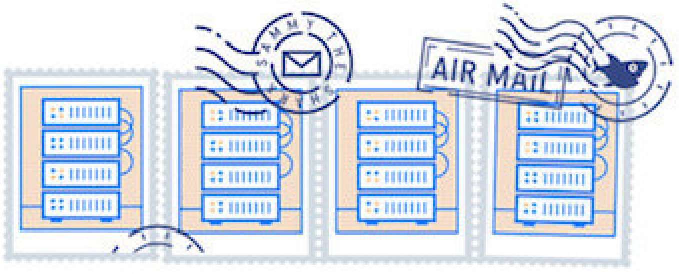
- %s in subject of mail -- I put the following system-command: `"echo 'mod-evasive HTTP Blacklisted %s more info here: www.projecthoneypot.org/ip_%s' | mail -s 'Blocked IP %s by mod-evasive' root@localhost"`
  --> it's always shown `Blocked IP (null) by mod-evasive` in the subject. Is there a way to display IP in subject!? What am I doing wrong?
- Is there a way to get a more verbose log? In the entries in the log-folder there is always only a number displayed.
  -> What does it mean?
  -> Is there a way to log which sites were used?
  -> In general more info?

pseudokool *February 6, 2019*

0 Appears that mod_evasive doesn't work with APache in prefork mode, but does work reasonably with mpm turned on. Is this true?







GET OUR BIWEEKLY NEWSLETTER

Sign up for Infrastructure as a Newsletter.

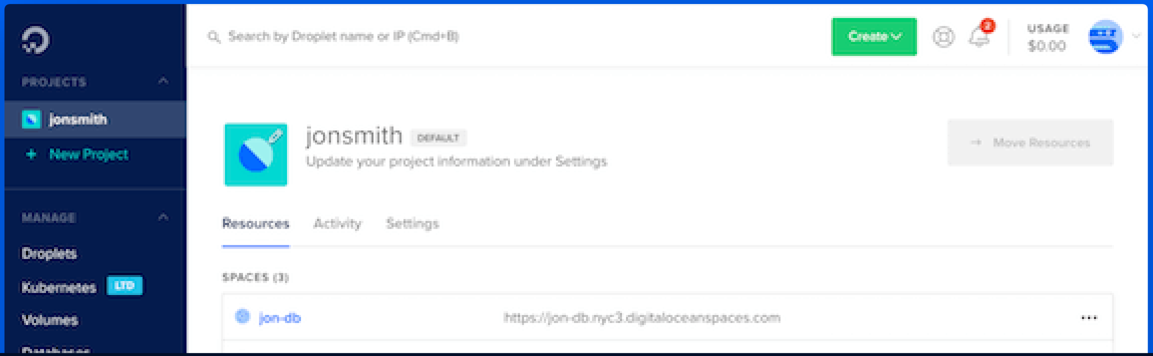
Featured on Community Intro to Kubernetes Learn Python 3 Machine Learning in Python Getting started with Go Migrate Node.js to Kubernetes

DigitalOcean Products Droplets Managed Databases Managed Kubernetes Spaces Object Storage Marketplace

Welcome to the developer cloud

DigitalOcean makes it simple to launch in the cloud and scale up as you grow – whether you’re running one virtual machine or ten thousand.

Learn More



Company

About
Leadership
Blog
Careers
Partners
Referral Program
Press
Legal & Security

Products

Products Overview
Pricing
Droplets
Kubernetes
Managed Databases
Spaces
Marketplace
Load Balancers
Block Storage
Tools & Integrations
API
Documentation

Community

Tutorials
Q&A
Projects and Integrations
Tags
Product Ideas
Meetups
Write for DOnations
Droplets for Demos
Hatch Startup Program
Shop Swag
Research Program
Currents Research