ModSecurity2 Installation, and Configuration

Hi,

I actually searched a lot of times through Mr. Google looking for a ModSecurity2 HOWTO, but unfortunately I didn't find any. So I decided to write this simple HOWTO to help Sys Admin's to install and implement the ModSecurity2 Apache Module.

For those whom are now wondering by now !!!, what is this ModSecurity???, Well let's use the official definition:
ModSecurity™ is an open source, free web application firewall (WAF) Apache module. With over 70% of all attacks now carried out over the web application level, organizations need all the help they can get in making their systems secure. WAFs are deployed to establish an external security layer that increases security, detects and prevents attacks before they reach web applications. It provides protection from a range of attacks against web applications and allows for HTTP traffic monitoring and real-time analysis with little or no changes to existing infrastructure.

Great now that you know what it is, for sure you know if you really need it or not. Before we start, please note that this HOWTO is for ModSecurity2 with Apache2. I shall not be discussing ModSecurity Version1 and Apache Version1. And by the way they are completely different in configuration and other issues.

Okay before we start the installation, first of all we need to install the Apache2 and all the other software needed to install ModSecurity2. In case you don't have already an Apache2 server up and running, please do the following:

On an RPM Based Distro:

```
# yum install httpd system-config-httpd httpd-devel mod_ssl
```

On a DEB Based Distro:

```
# apt-get install apache2 apache2-utils
```

Okay, now Apache is ready, lets move on to install other ModSecurity2 prerequisites. We need mod_unique_id which comes by default with the Apache2 main package, and we need to other libraries:
libxml2 ---> http://xmlsoft.org/downloads.html
liblua5 ---> http://www.lua.org/download.html

Make sure they don't currently exists by:

```
$ ls -l /usr/lib/
```

If you find them there? well great let's move on. If not? then please download them from the sites supplied above.

Before we start the installation process, let's just stop Apache from running.
On an RPM Based Distro:

```
/etc/init.d/httpd stop
```

On a DEB Based Distro:

```
/etc/init.d/apache2 stop
```

Now the great part, Installing the ModSecurity2. I shall be using the packages provided from the main developers (BreachSecurity), which you can find here --> DOWNLOAD

I used *modsecurity-apache_2.5.6* the time I was writing this HOWTO, so please make sure to choose the latest, because there always is updates and other cool stuff added to the new releases. Okay let's start downloading:

```
$ wget -c http://www.modsecurity.org/download/modsecurity-apache_2.5.6.tar.gz
```

Now let's extract the files:

```
$ tar xvfz modsecurity-apache_2.5.6.tar.gz
```

Change into the directory:

```
$ cd modsecurity-apache_2.5.6/apache2/
```

Now we shall start with the configuration part, I always like to use the *--with-apxs* option (it is optional), now do:

```
$ ./configure --with-apxs=/usr/sbin/apxs
```

After that lets compile the package:

```
$ make
```

A test is always good to go for:

```
$ make test
```

And if everything went well, a clean install is done (as root user):

```
# make install
```

Okay, now we have ModSecurity2 installed this doesn't mean that it is activated. To make it active we need to add some lines into the Apache main configuration files which tell Apache to load the ModSecurity Modules. So let's edit the main Apache configuration file (please use any editor that you prefer, for me I stick with vim):

```
# vim /etc/httpd/conf/httpd.conf
```

Now go to the end of the LoadModule lines, and add the following:

```
## Load Mod Security Modules and Required Libraries
LoadFile /usr/lib/libxml2.so
LoadFile /usr/lib/liblua5.1.so
LoadModule security2_module modules/mod_security2.so
Include conf.d/modsecurity2/*.conf
Include conf.d/modsecurity2/optional_rules/*.conf
LoadModule unique_id_module modules/mod_unique_id.so
```

Save the file and exit.

Now I assume that you are still in the directory *modsecurity-apache_2.5.6/apache2/* ? If you are then let's copy the rules to the *modsecurity2/* directory in the Apache configuration directory. This is done like this:

```
# mv ../modsecurity-apache_2.5.6/rules/* /etc/httpd/conf.d/modsecurity2/
```

If you are not in the modsecurity-apache_2.5.6/apache2/ directory then please switch to the following:

```
# cp /path2/modsecurity-apache_2.5.6/rules/* /etc/httpd/conf.d/modsecurity2/
```

I always like to make checks as long as there is a syntax checker available, so lets check our Apache syntax:

```
apachectl configtest
```

If an error is shown please check the line number where the error exists, and solve it, if not? Then lets continue.

Now let's start the Apache server:

On an RPM Based Distro:

```
/etc/init.d/httpd start
```

On a DEB Based Distro:

```
/etc/init.d/apache2 start
```

Okay, now how can we know if ModSecurity2 is running or not? Well let's make a test and see. First of all create a PHP file name it anything (for me I chose *checkmd.php*) and add the following code to it:

```
<?
   file $text=$_GET['file'];
   echo "Content of File $text";
   echo `cat $text`;
?>
```

Then goto your favorite web browser and try to browse the to the following link:

http://localhost/checkmd.php?file=/etc/passwd

Aha, you shall now have got a White Empty page saying in its title "501 Method Not Implemented", and its context is:

```
Method Not Implemented

GET to /checkmd.php not supported.
```

Well done, this insures that ModSecurity2 is working and it blocked the try we are making to access the */etc/passwd* file.

--== CONGRATULATIONS INSTALLATION COMPLETE ==--

--== Some questions that you might be woundering of ==--

Question1: If for a reason or another I want to stop the ModSecurity2 from working, how is that Done?
Answer1: Simply edit the following file:

```
# vim /etc/httpd/conf.d/modsecurity2/modsecurity_crs_10_config.conf
```

And switch the *SecRuleEngine* variable from *On* to *Off*:

```
SecRuleEngine Off
```

Save the file and a quick reload is needed to tell Apache to reread its configuration files:
On an RPM Based Distro:

```
# /etc/init.d/httpd reload
```

On a DEB Based Distro:

```
# /etc/init.d/apache2 force-reload
```

You can also switch the *SecRuleEngine* to *DetectionOnly* in case you need to monitor you web application.

Question2: I am hosting my own Blog, and every time I add a post I keep getting a not implemented error. Why is this happening? And how can I disable it for my posts? (I.e: disabling ModSecurity2 for a specific file).
Answer2: This is happening because ModSecurity thinks that you are trying to inject some code to the PHP files of your Blog based on the pretested rules BreachSecurity provides with the ModSecurity2 package. It is called "*False Positives*", because it is a false alert (you are only posting a new post) and it's positive because it matched one of the rules that ModSecurity2 depends on.

Disabling it shall need some work from your side, because as I told in the beginning, ModSecurity2 does not work like ModSecurity Version1, and disabling it in version 1 was much easier, but here in ModSecurity2 it is really much more powerful. First you need to monitor your Apache error log files. So let us start the following:

```
# tail -f /var/log/httpd/error_log
```

I shall assume that we have the Blog parked on the domain *www.example.com*. Open your favorite browser and goto:
http://www.example.com/wp-admin/post-new.php

And write the same post you got an error from. Now you shall get the same error, right? Great, now lets go and check what our error log has reported us. We shall see something like this:

```
[Mon Sep 22 11:01:12 2008] [error] [client 211.158.21.152] ModSecurity: Access denied with code 501 (phase 2). Pattern match "^(?:ht|f)tp:/" at ARGS:referredby. [file "/etc/httpd/conf.d/modsecurity2/optional_rules/modsecurity_crs_42_tight_security.conf"] [line "32"] [id "950117"] [msg "Remote File Inclusion Attack"] [severity "CRITICAL"] [hostname "www.example.com"] [uri "/wp-admin/post-new.php"] [unique_id "dGP7GXAAA8AAAPQC4AAEAAFZ"]
```

Hey, wait a minute, What are these????
This is actually a single line of error. Let me explain the important parts of it, and what we shall need to disable this alert from happening in the future. I am interested now in:

1st: 211.158.21.152 this is the IP of the host who because of him the alert was raised.
2nd: modsecurity_crs_42_tight_security.conf is the file that includes the rule.
3rd: ModSecurity classified this as a Remote File Inclusion Attack.
4th: www.example.com is the domain where the problem came from (in case you are hosting more domains on the same server).
5th: The file that made the alert is /wp-admin/post-new.php.
Finally: id, This is the important thing for us here, because we shall be using this number which is actually a rule number from the ModSecurity rule set.

Great we have the id let us disable this error from happening again. Goto Apache's main config file and edit it:

```
# vim /etc/httpd/conf/httpd.conf
```

Add the following lines to disable the error above:

```
<LocationMatch "/wp-admin/post-new.php">
SecRuleRemoveById 950117
</LocationMatch>
```

Now save, close the file and reload Apache.

Try making the same post again. If everything went well then great, if you get another error? then go back to the log files and get the id and add it to the *SecRuleRemoveById* we wrote above.

Notes:
1- If you are hosting more than one domain on the server (*VirtualHosting*), then it is better to add the lines above to the configuration file of that domain.

2- You can also solve this error by witting your own rules, but it is not quite easy and you need some knowledge in writing Regex codes, and to be more precise you need to know how to write Perl-Compatible Regular Expression (PCRE), which ModSecurity rules are written with.

If you are looking for further documentation, then the main site is always a good place to look in:
http://www.modsecurity.org/documentation/modsecurity-apache/2.5.6/html-multipage/installation.html

So that's it :)

Written by:
Ali Al-Shemery (B!n@ry)
Linux Arab Community