

Install ModSecurity with Apache on Ubuntu 22.04

By **gen_too** - April 30, 2022

5081

Welcome to our guide on how to install ModSecurity with Apache on Ubuntu 22.04. [Modsecurity](#) is an open source, cross platform web application firewall (WAF) developed by Trustwave’s SpiderLabs. It provides a robust event-based programming language which protects web applications against a wide range of attacks such as SQL injection, Cross-site Scripting (XSS), Local File Include, Remote File Include e.tc. It also allows for HTTP traffic monitoring, logging and real-time analysis.

There exists two versions of ModSecurity. Modsecurity 2.x and libmodsecurity (Modsecurity 3.x) which is a complete rewrite of Modsecurity 2.

Install ModSecurity with Apache on Ubuntu 22.04

To begin with, re-synchronize your system packages to their latest versions.

```
apt update
```

Install Apache Web Server on Ubuntu 22.04

ModSecurity runs on a top of a web server. In this guide, we are using Apache. If not already installed, you can install Apache

[Install LAMP Stack on Ubuntu 22.04](#)

You can now choose to install Modsecurity 2 or install Modsecurity 3.

- [Install Modsecurity 2.x on Ubuntu 22.04](#)
- [Install Modsecurity 3.x on Ubuntu 22.04](#)

The two work fine. However, I would choose Modsecurity 2.x for due some issues whereby with Modsecurity 3.x, when you enable blocking of detected attacks, the logs are only written to modsec_audit.log file and not to Apache error.log. The choice is yours on which one to install, however.

Install Modsecurity 2.x on Ubuntu 22.04

If you want to install ModSecurity 2 (2.9.5 to be specific), you can run the command below;

```
apt install libapache2-mod-security2
```

Install Modsecurity 3.x on Ubuntu 22.04

- Install Required Build Tools and Dependencies

To install Libmodsecurity or Modsecurity 3 (3.0.6 to be specific) on Ubuntu 22.04, we are going to build it from source. Hence, you need to install some required build tools and dependencies for a successful build.

```
apt install g++ flex bison curl apache2-dev doxygen \
libyajl-dev ssdeep liblua5.2-dev libgeoip-dev libtool \
dh-autoreconf libcurl4-gnutls-dev libxml2 libpcre++-dev \
libxml2-dev git -y
```

- Compile and Install ModSecurity on Ubuntu 22.04

Once your system is setup, you can proceed to install Modsecurity v3 on Ubuntu 22.04.

- Download ModSecurity Source Code

To install the latest stable version of **ModSecurity**, you need to compile it from the source.

Therefore, navigate to [ModSecurity releases page](#) and download ModSecurity 3 source code. You can simply use wget to pull it.

```
wget https://github.com/SpiderLabs/ModSecurity/releases/download/v3.0.6/modsecurity-v3.0.6.tar.gz
```

Extract the ModSecurity source code.

```
tar xzf modsecurity-v3.0.6.tar.gz
```

- Compile and Install ModSecurity 3 on Ubuntu 22.04

Navigate to the ModSecurity source directory, configure, compile and install it

```
cd modsecurity-v3.0.6
```

Configure ModSecurity to adapt it to your system and check if any required dependency is missing.

```
./build.sh
```

You can safely ignore the **fatal: not a git repository (or any of the parent directories): .git** error and the obsolete warning messages.

```
./configure
```

Be sure to fix any dependency issue, if any, before you can proceed to compile and install ModSecurity with Apache on Ubuntu 22.04.

If the configure script above completes with no error, proceed to compile and install LibModSecurity on Ubuntu 22.04.

```
make
```

```
make install
```

LibModSecurity aka ModSecurity v3 has now been installed on Ubuntu 22.04

- Install ModSecurity-Apache Connector

The ModSecurity-apache connector provides a communication channel between Apache and libModsecurity. Now that libmodsecurity is installed, follow through the following steps to install Modsecurity Apache connector.

Clone the git repository for the ModSecurity Apache connector.

```
cd ~
git clone https://github.com/SpiderLabs/ModSecurity-apache
```

Navigate to ModSecurity-apache directory and run the following commands to compile and install it.

```
cd ModSecurity-apache
```

```
./autogen.sh
```

```
./configure --with-libmodsecurity=/usr/local/modsecurity/
```

```
make
make install
```

Configure Apache with ModSecurity

- [Configure Modsecurity 2.x](#)
- [Configure Modsecurity 3.x](#)

Configure Modsecurity 2.x

Rename sample configuration;

```
cp /etc/modsecurity/modsecurity.conf{-recommended,}
```

Download OWASP ModSecurity Core Rule Set (CRS) into /etc/modsecurity/crs/ directory.

```
git clone https://github.com/SpiderLabs/owasp-modsecurity-crs.git
```

```
cp -r owasp-modsecurity-crs/rules /etc/modsecurity/crs/
```

Configure Apache to load Modsecurity rules;

```
cp /etc/apache2/mods-available/security2.conf{,.old}
```

```
cat > /etc/apache2/mods-available/security2.conf << 'EOL'
<IfModule security2_module>
    SecDataDir /var/cache/modsecurity
    Include /etc/modsecurity/modsecurity.conf
    Include /etc/modsecurity/crs/crs-setup.conf
    Include /etc/modsecurity/crs/rules/*.conf
</IfModule>
EOL
```

Enable Modsecurity OWASP rules blocking;

```
vim /etc/modsecurity/crs/crs-setup.conf
```

Comment the lines below by adding # at the beginning of a line;

```
SecDefaultAction "phase:1,log,auditlog,pass"
SecDefaultAction "phase:2,log,auditlog,pass"
```

Such that they look like;

```
#SecDefaultAction "phase:1,log,auditlog,pass"
#SecDefaultAction "phase:2,log,auditlog,pass"
```

And then uncomment the lines below by removing the # at the beginning of the line;

```
# SecDefaultAction "phase:1,log,auditlog,deny,status:403"
# SecDefaultAction "phase:2,log,auditlog,deny,status:403"
```

Such that they look like;

```
SecDefaultAction "phase:1,log,auditlog,deny,status:403"
SecDefaultAction "phase:2,log,auditlog,deny,status:403"
```

Save the changes and exit the file.

Turn on Modsecurity Engine;

```
sed -i 's/SecRuleEngine DetectionOnly/SecRuleEngine On/' /etc/modsecurity/modsecurity.conf
```

Configure Modsecurity 3.x

Configure Apache to load Modsecurity Apache connector module by adding the line below to the main Apache configuration file.

```
echo "LoadModule security3_module /usr/lib/apache2/modules/mod_security3.so" | sudo tee -a /etc/apache2/apache2.conf
```

Create ModSecurity configuration directory under `/etc/apache2`

```
sudo mkdir /etc/apache2/modsecurity.d
```

Copy the sample ModSecurity configuration file from the source code directory to the ModSec configuration directory created above renaming it as follows.

```
sudo cp ~/modsecurity-v3.0.6/modsecurity.conf-recommended /etc/apache2/modsecurity.d/modsecurity.conf
```

Also copy the `unicode.mapping` file from ModSecurity source directory to Apache Modsecurity configuration directory.

```
sudo cp ~/modsecurity-v3.0.6/unicode.mapping /etc/apache2/modsecurity.d/
```

Turn on ModSecurity by changing the value of `SecRuleEngine` to `On` .

```
sed -i 's/SecRuleEngine DetectionOnly/SecRuleEngine On/' /etc/modsecurity/modsecurity.conf
```

Next, you need to configure ModSecurity rules.

Therefore, create a file where you can define the rules to include, say, `/etc/apache2/modsecurity.d/modsec_rules.conf` .

```
cat > /etc/apache2/modsecurity.d/modsec_rules.conf << 'EOL'
Include "/etc/apache2/modsecurity.d/modsecurity.conf"
Include "/etc/apache2/modsecurity.d/owasp-crs/crs-setup.conf"
Include "/etc/apache2/modsecurity.d/owasp-crs/rules/*.conf"
EOL
```

Since we have included the OWASP Rules as part of ModSecurity 3 rules, proceed to install them.

The **OWASP ModSecurity Core Rule Set (CRS)** is a set of generic attack detection rules for use with ModSecurity. It aims at protecting the web applications from a wide range of attacks, including the OWASP Top Ten, minimum of false alerts.

Clone the CRS from [GitHub repository](#) to `/etc/apache2/modsecurity.d/` as shown below;

```
git clone https://github.com/SpiderLabs/owasp-modsecurity-crs.git /etc/apache2/modsecurity.d/owasp-crs
```

Next, rename `crs-setup.conf.example` to `crs-setup.conf` .

```
sudo cp /etc/apache2/modsecurity.d/owasp-crs/crs-setup.conf{.example,}
```

Similarly, enable Modsecurity OWASP rules blocking by editing the `/etc/apache2/modsecurity.d/owasp-crs/crs-setup.conf` file above;

```
vim /etc/apache2/modsecurity.d/owasp-crs/crs-setup.conf
```

```
...
#SecDefaultAction "phase:1,log,auditlog,pass"
#SecDefaultAction "phase:2,log,auditlog,pass"
```

```
...
#
SecDefaultAction "phase:1,log,auditlog,deny,status:403"
SecDefaultAction "phase:2,log,auditlog,deny,status:403"
...
```

NOTE that with this change, it causes ModSecurity 3 to log to modsec_audit.log file ONLY and not to Apache error.log.

For ModSecurity 2, the attack logs are written to Apache error log. This makes it easy to process these logs with other external security monitoring tools.

Activate ModSecurity Apache Protection on Ubuntu

After all that, activate the Modsecurity on the default site configuration file or on any virtual host configuration file. In this guide, we are using Apache’s default site configuration file.

- [Activate Modsecurity 2.x on Ubuntu](#)
- [Activate Modsecurity 3.x on Ubuntu](#)

Activate Modsecurity 2.x on Ubuntu

If you are running Modsecurity 2.x, all you need to do at this point is to restart Apache web server.

```
apachectl -t
```

```
systemctl restart apache2
```

Activate Modsecurity 3.x on Ubuntu

If you are running Modsecurity 3.x, activate it as follows;

```
cp /etc/apache2/sites-available/000-default.conf{,.old}
```

See our sample default site virtual host configuration with no comments;

```
cat > /etc/apache2/sites-available/000-default.conf << 'EOL'
<VirtualHost *:80>
    modsecurity on
    modsecurity_rules_file /etc/apache2/modsecurity.d/modsec_rules.conf
    ServerAdmin webmaster@localhost
    DocumentRoot /var/www/html
    ErrorLog ${APACHE_LOG_DIR}/error.log
    CustomLog ${APACHE_LOG_DIR}/access.log combined
</VirtualHost>
EOL
```

The lines;

```
modsecurity on
modsecurity_rules_file /etc/apache2/modsecurity.d/modsec_rules.conf
```

Turns on Modsecurity and specifies the location of the Modsecurity rules.

Check Apache for configuration errors and restart it.

```
apachectl -t
```

If there is no error, you should get the output, **Syntax OK**.

```
systemctl restart apache2
```

Testing Modsecurity

Next, test the effectiveness of Modsecurity, for example, command injection. Run the command below;

```
curl localhost?doc=/bin/ls
```

```
<!DOCTYPE HTML PUBLIC "-//IETF//DTD HTML 2.0//EN">
<html><head>
<title>403 Forbidden</title>
</head><body>
<h1>Forbidden</h1>
<p>You don't have permission to access this resource.</p>
<hr>
<address>Apache/2.4.52 (Ubuntu) Server at localhost Port 80</address>
</body></html>
```

Check Modsecurity logs;

```
tail /var/log/modsec_audit.log
```

```
...

---GMiW89KJ---H--
ModSecurity: Access denied with code 403 (phase 2). Matched "Operator `PmFromFile' with parameter `unix-shell.data' against variable `ARGS:doc' (Value: `/bin/ls' ) [file "/etc/apache2/modsecurity.d/owasp-crs/rules/REQUEST-932-APPLICATION-ATTACK-RCE.conf"] [line "496"] [id "932160"] [rev ""] [msg "Remote Command Execution: Unix Shell Code Found"] [data "Matched Data: bin/ls found within ARGS:doc: /bin/ls"] [severity "2"] [ver "OWASP_CRS/3.2.0"] [maturity "0"] [accuracy "0"] [tag "application-multi"] [tag "language-shell"] [tag "platform-unix"] [tag "attack-rce"] [tag "paranoia-level/1"] [tag "OWASP_CRS"] [tag "OWASP_CRS/WEB_ATTACK/COMMAND_INJECTION"] [tag "WASCTC/WASC-31"] [tag "OWASP_TOP_10/A1"] [tag "PCI/6.5.2"] [hostname "127.0.1.1"] [uri "/"] [unique_id "1651339080"] [ref "o1,6v10,7t:urlDecodeUni,t:cmdLine,t:normalizePath,t:lowercase"]

---GMiW89KJ---I--

---GMiW89KJ---J--

---GMiW89KJ---Z--
...
```


Similarly, install Nikto on Ubuntu and use it to scan the server to test the modsecurity rules;

Install and Use Nikto Web Scanner on Ubuntu

Access the site from browser and test command injection attack, `http://domain.name/?exec=/bin/bash` .

Forbidden

You don't have permission to access this resource.



Kifarunix
*NIX TIPS & TUTORIALS

Apache/2.4.41 (Ubuntu) Server at kifarunix-demo.com Port 80

Tailing the Apache error logs;

```
tail -f /var/log/apache2/error.log
```

```
[Sat Apr 30 23:43:10.888508 2022] [:error] [pid 4504:tid 140113703069248] [client 127.0.0.1:57030] [client 127.0.0.1] ModSecurity: Access denied with code 403 (phase 2). Matched phrase "bin/ls" at ARGS:doc. [file "/etc/modsecurity/crs/rules/REQUEST-932-APPLICATION-ATTACK-RCE.conf"] [line "518"] [id "932160"] [msg "Remote Command Execution: Unix Shell Code Found"] [data "Matched Data: bin/ls found within ARGS:doc: /bin/ls"] [severity "CRITICAL"] [ver "OWASP_CRS/3.2.0"] [tag "application-multi"] [tag "language-shell"] [tag "platform-unix"] [tag "attack-rce"] [tag "paranoia-level/1"] [tag "OWASP_CRS"] [tag "OWASP_CRS/WEB_ATTACK/COMMAND_INJECTION"] [tag "WASCTC/WASC-31"] [tag "OWASP_TOP_10/A1"] [tag "PCI/6.5.2"] [hostname "localhost"] [uri "/"] [unique_id "Ym2fXltd6Nody7X50nHgVQAAAc"]
```

As a result, this confirms that the Modsecurity is functioning as expected.

That is just about it on our guide on how to install ModSecurity with Apache on Ubuntu 22.04.

Reference:

[ModSecurity-apache](#)

[ModSecurity](#)

Other Tutorials

[Install ModSecurity 3 with Apache in a Docker Container](#)

[Intercept Malicious File Upload with ModSecurity and ClamAV](#)

TAGS

Apache

install modsecurity on ubuntu 22.04

modsecurity

Modsecurity 3 and modsecurity 2

opensource WAF

Ubuntu 22.04

ubuntu 22.04 modsecurity

waf

Previous article

[Install LAMP Stack on Ubuntu 22.04](#)

Next article

[Process ModSecurity Logs using Wazuh](#)