**Aim:** Configuration of ModSecurity, Core Rule Set (CRS) on Apache server

**Theory:**

**ModSecurity and CRS**

ModSecurity is an open-source web-based firewall application (or WAF) supported by different web servers: Apache, Nginx, and IIS. With over 70% of all attacks now carried out over the web application level, organizations need every help they can get in making their systems secure.

ModSecurity is a web application firewall that can work either embedded or as a reverse proxy. Web application firewalls are deployed to establish an external security layer that increases the protection level, detects, and prevents attacks before they reach web-based software programs.

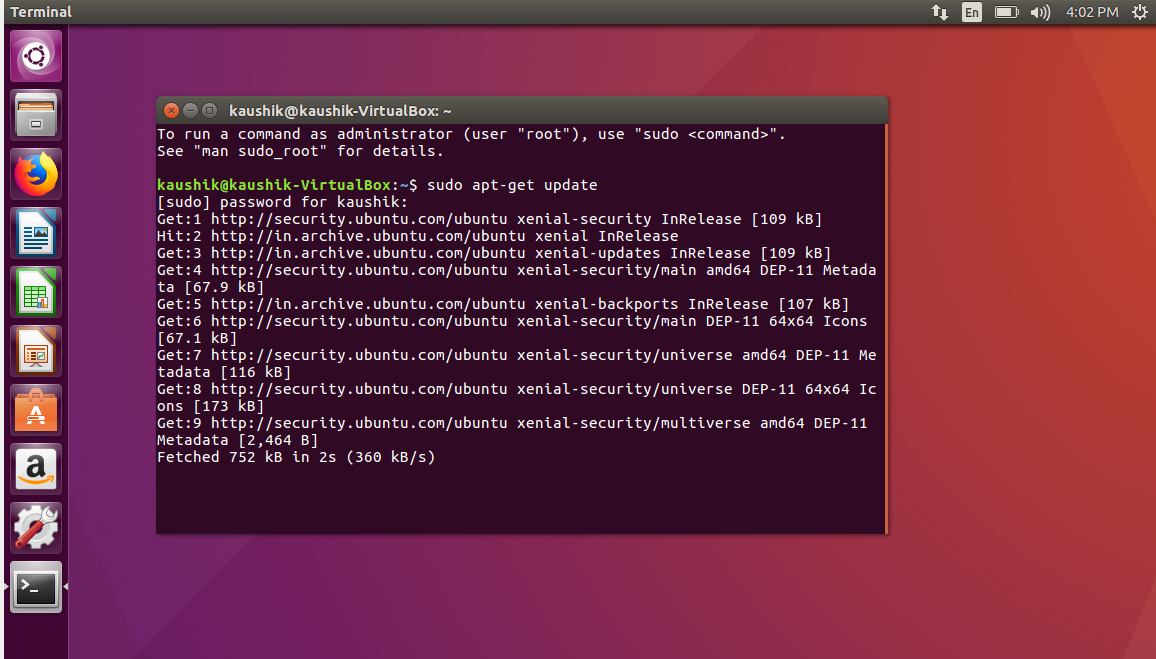
It provides protection from a range of attacks against web applications and allows for HTTP traffic monitoring, logging, and real-time analysis. ModSecurity commonly installed in conjunction with Apache, an open source web server. The benefits of using mod\_security are numerous and encompass defense from many kinds of web-based attack including code injection and brute force attacks.

The module is configured to protect web applications from various attacks. ModSecurity supports flexible rule engine to perform both simple and complex operations. It can potentially block common code injection attacks which strengthens the security of the server. It comes with a Core Rule Set (CRS) which has various rules for cross website scripting, bad user agents, SQL injection, trojans, session hijacking, and other exploits.

**Steps of Performance:**

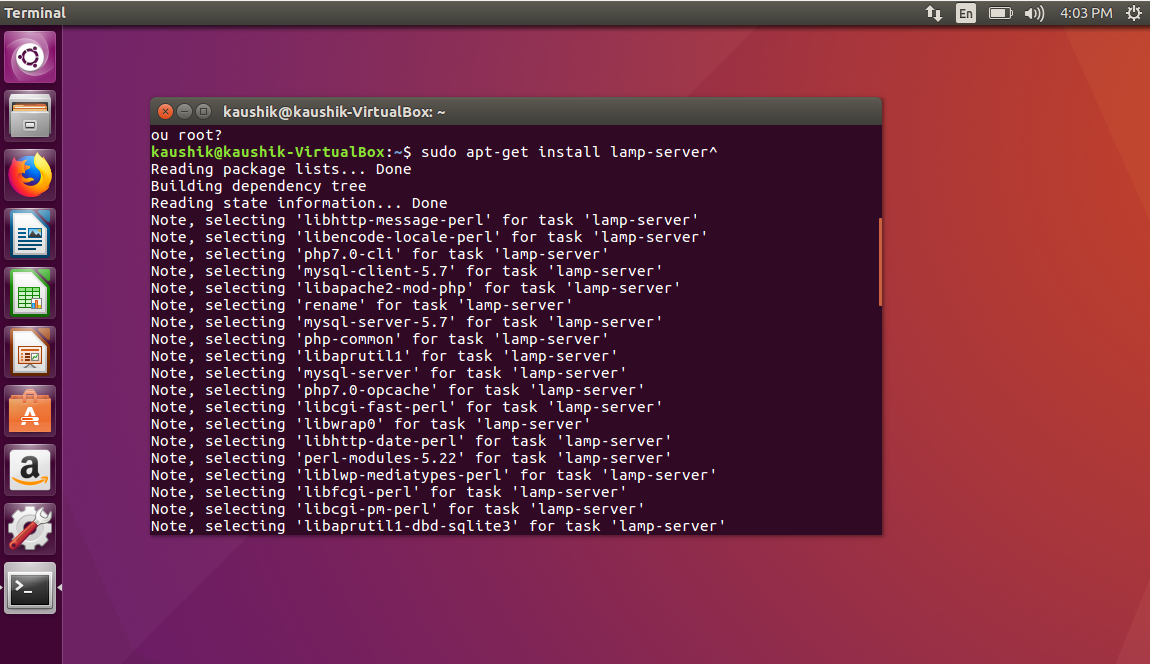
**Step 1:** Right Click and open Terminal. Before we begin, we need to update all the packages, libraries, and repositories installed on our machine by running the following command:

sudo apt-get update



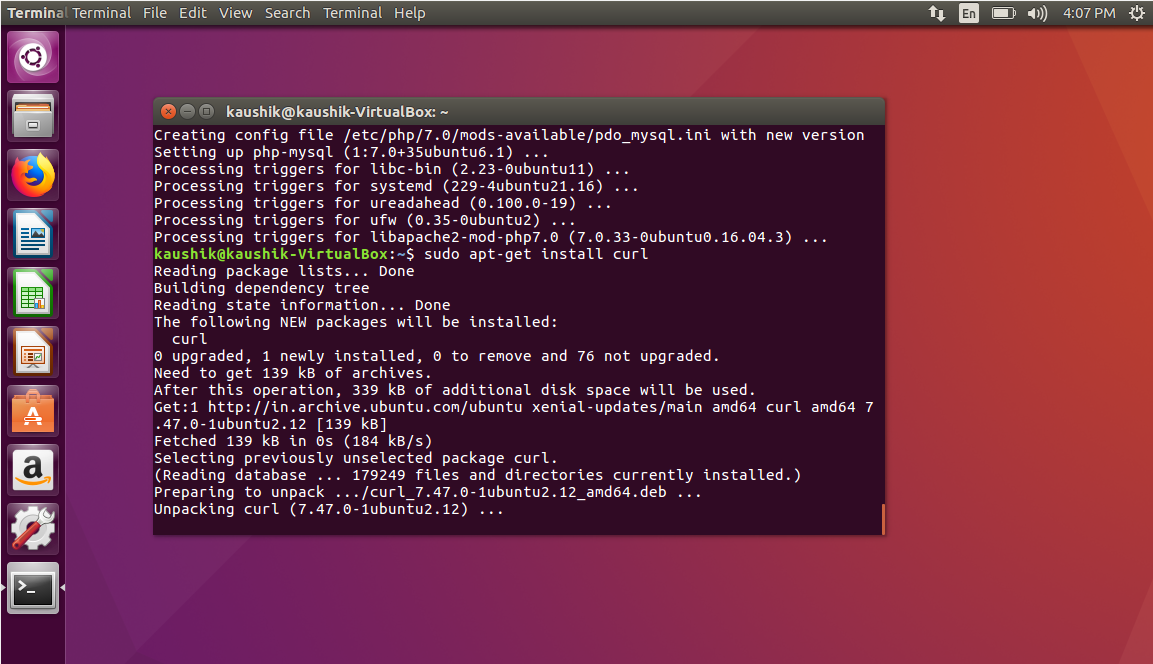
**Step 2:** In order to serve the client requests, you may require LAMP Server. LAMP Server is a Web Development Platform for Linux and stands for **L**inux **A**pache **M**ySQL **P**HP. Install LAMP Server using the following command:

sudo apt-get install lamp-server^



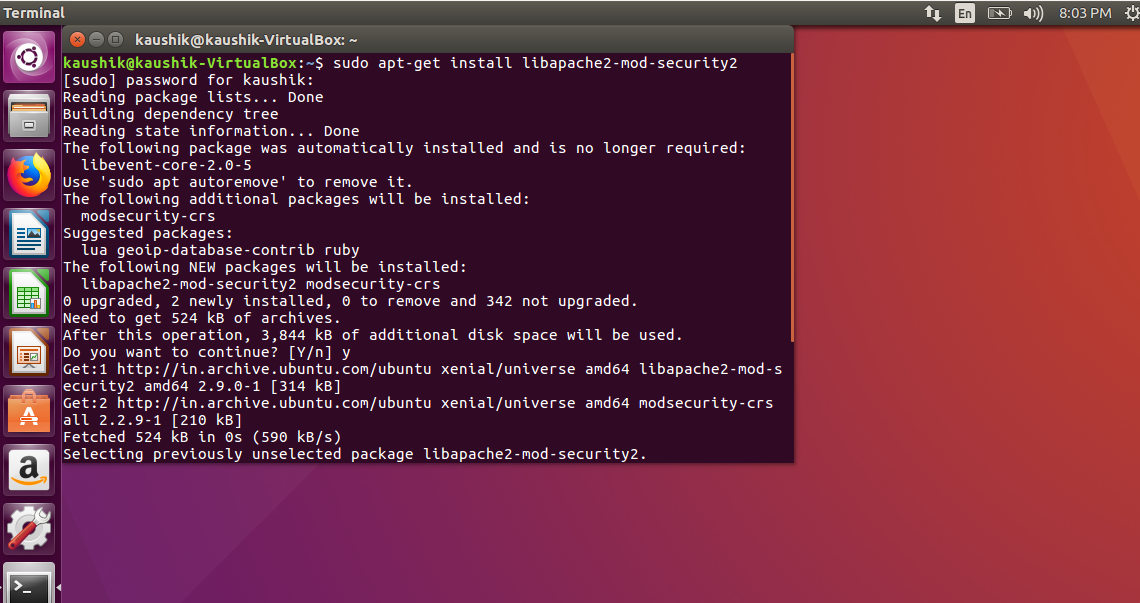
**Step 3:** The Release Key for ownCloud need to be downloaded which are stored in the cURL library. So, first install cURL in your system by running the following command:

sudo apt-get install curl



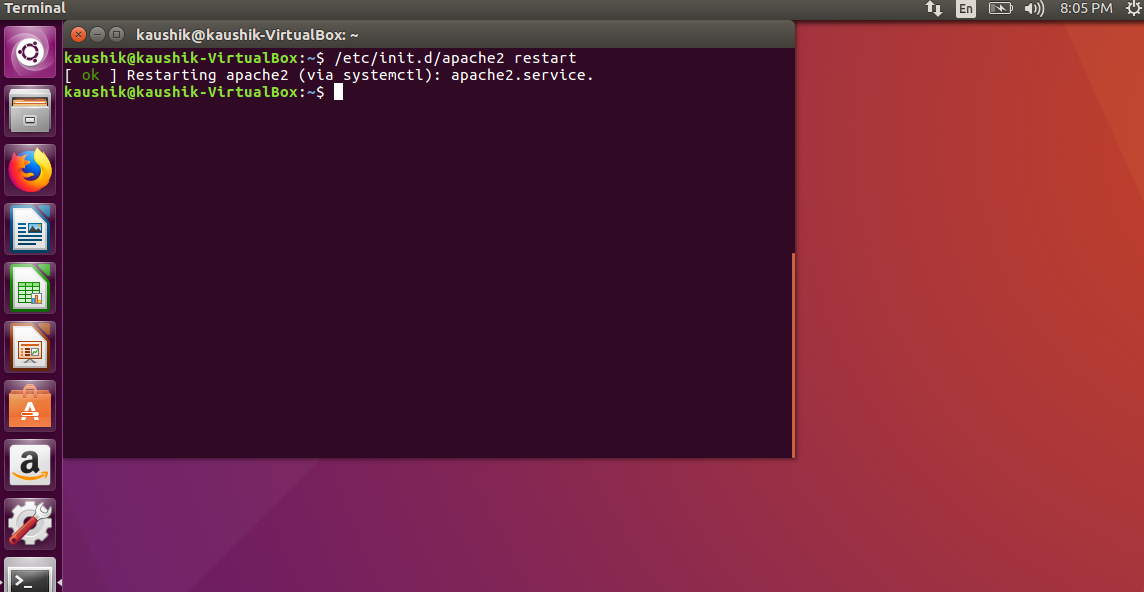
**Step 4:** Now, to install ModSecurity on Ubuntu, use the following command:

sudo apt-get install libapache2-mod-security2



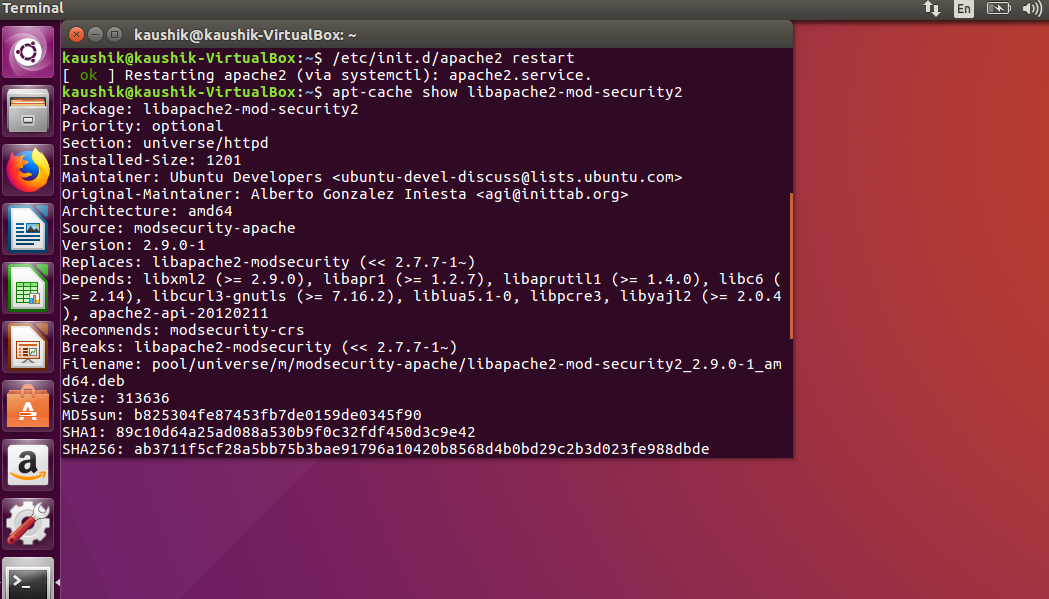
**Step 5:** Restart the Apache service:

/etc/init.d/apache2 restart



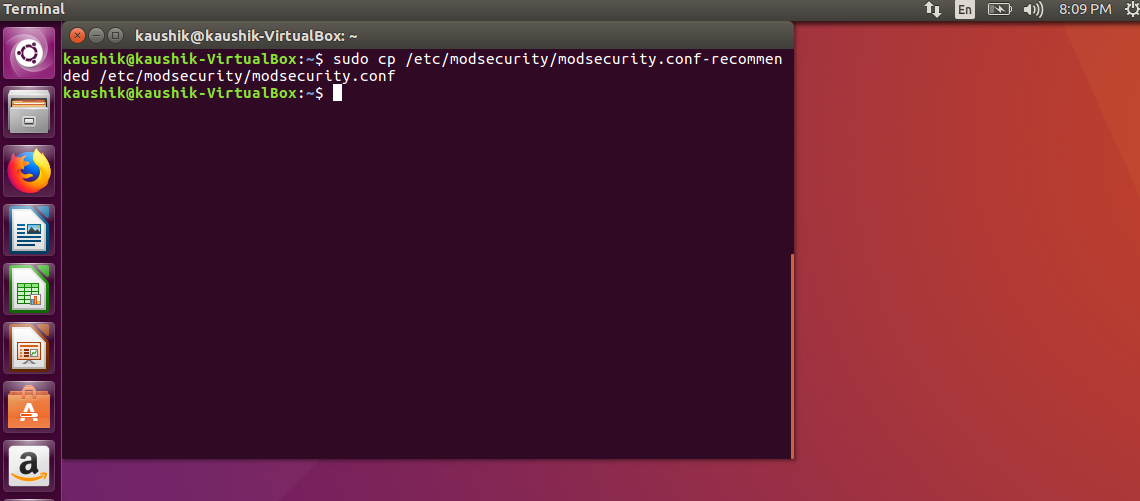
**Step 6:** Check the software version (should be 2.8.0 or later):

apt-cache show libapache2-mod-security2



**Step 7:** Copy and rename the default configuration file:

sudo cp /etc/modsecurity/modsecurity.conf-recommended /etc/modsecurity/modsecurity.conf



**Step 8:** Next, change the ModSecurity detection mode.

Open the configuration file in a text editor:

sudo nano /etc/modsecurity/modsecurity.conf

Near the top, you should see an entry labeled:

SecRuleEngine DetectionOnly

Change this to read as follows:

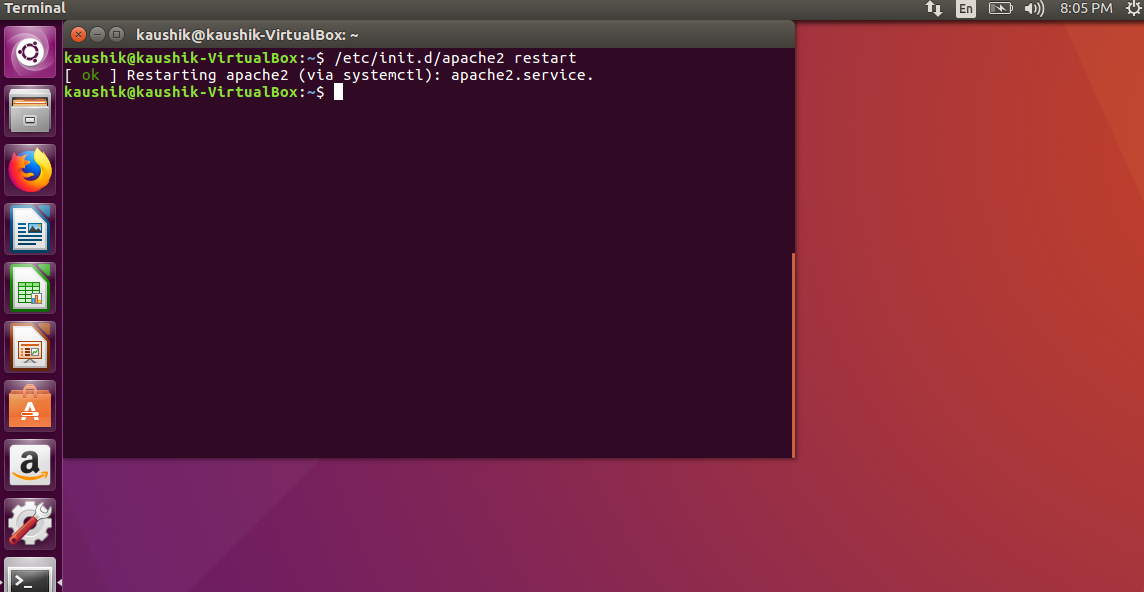
SecRuleEngine On

Use CTRL+X to exit, then press y then Enter to save the changes.



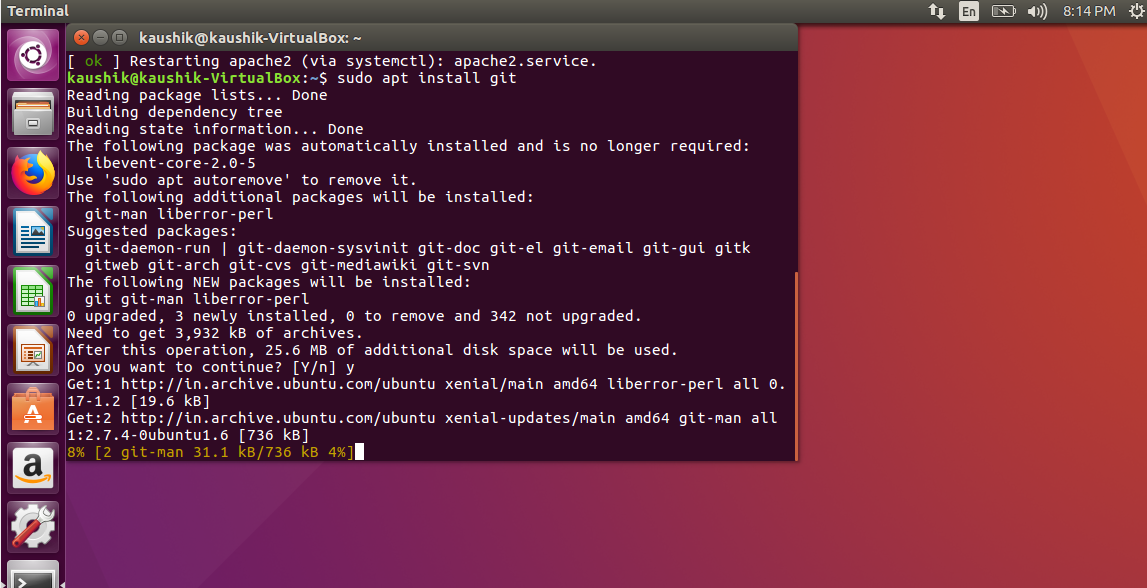
**Step 9:** Restart the Apache service:

/etc/init.d/apache2 restart



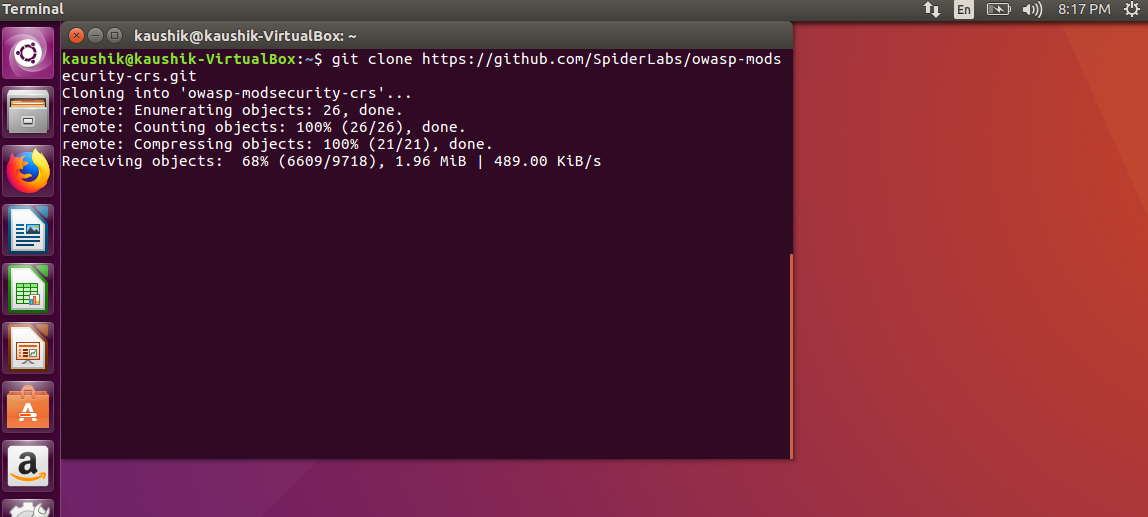
**Step 10:** Install Git if it’s not already included on your system:

sudo apt install git



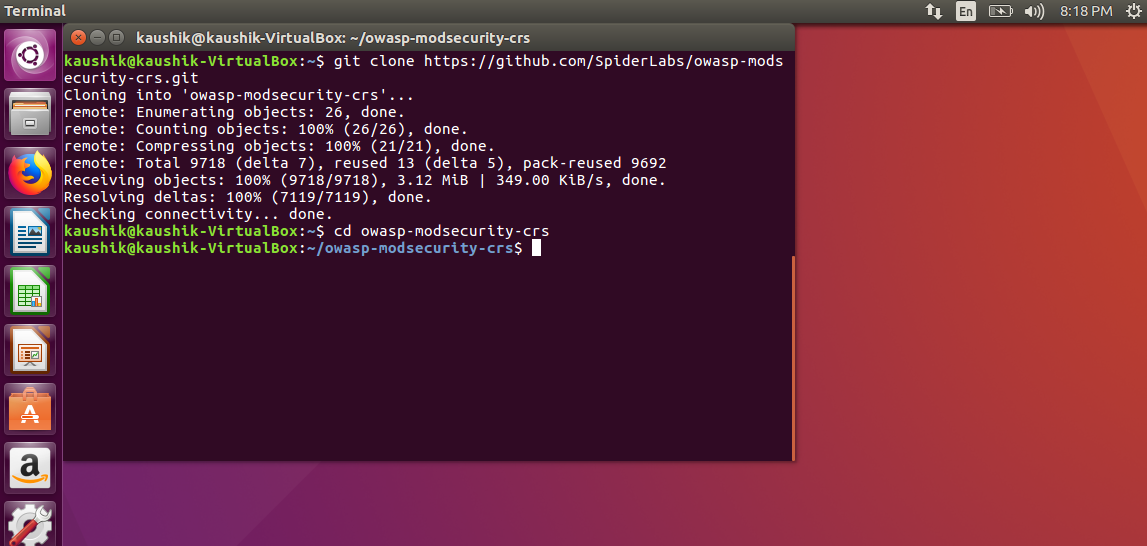
**Step 11:** Download a copy of the CRS:

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



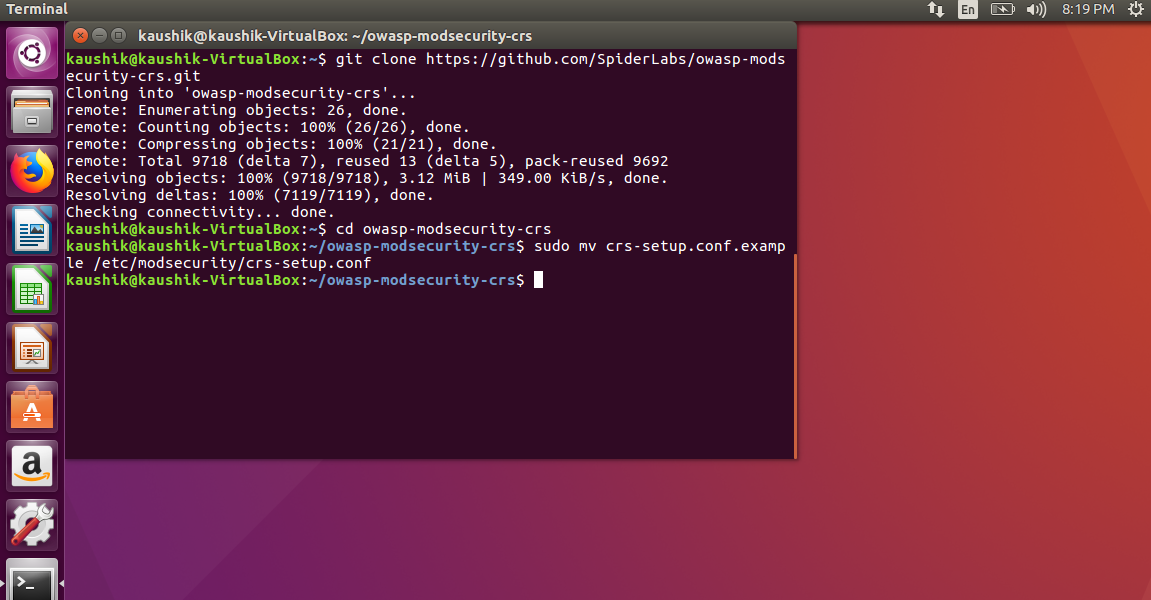
**Step 12:** Open a new directory:

cd owasp-modsecurity-crs



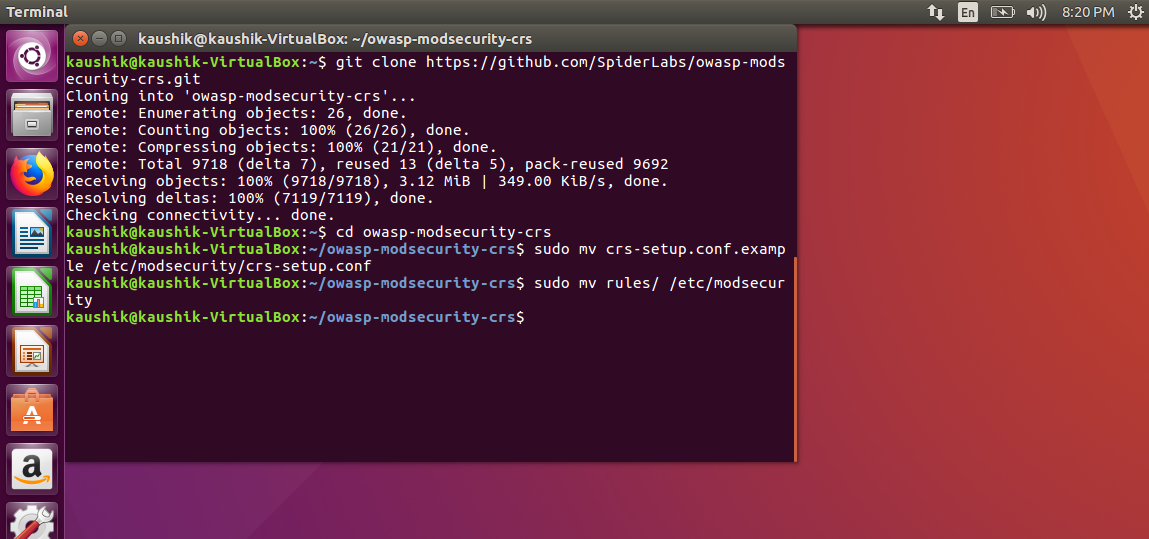
**Step 13:** Move the crs-setup file:

sudo mv crs-setup.conf.example /etc/modsecurity/crs-setup.conf



**Step 14:** Then move the rules/ directory:

sudo mv rules/ /etc/modsecurity



**Step 15:** Next, check your security2.conf file to verify it’s set to load the ModSecurity rules:

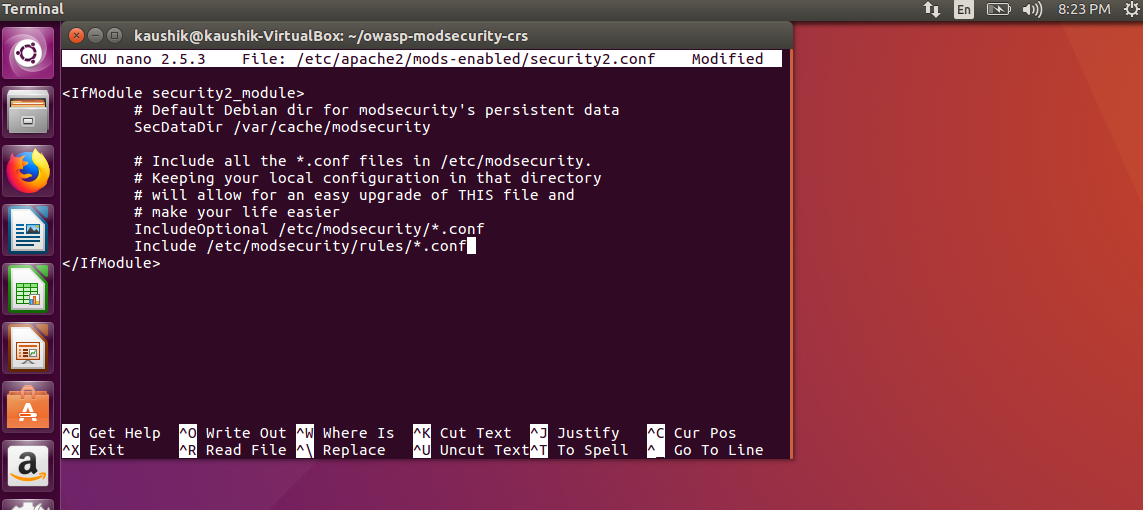
sudo nano /etc/apache2/mods-enabled/security2.conf

Verify you have the following lines included and uncommented:

IncludeOptional /etc/modsecurity/\*.conf

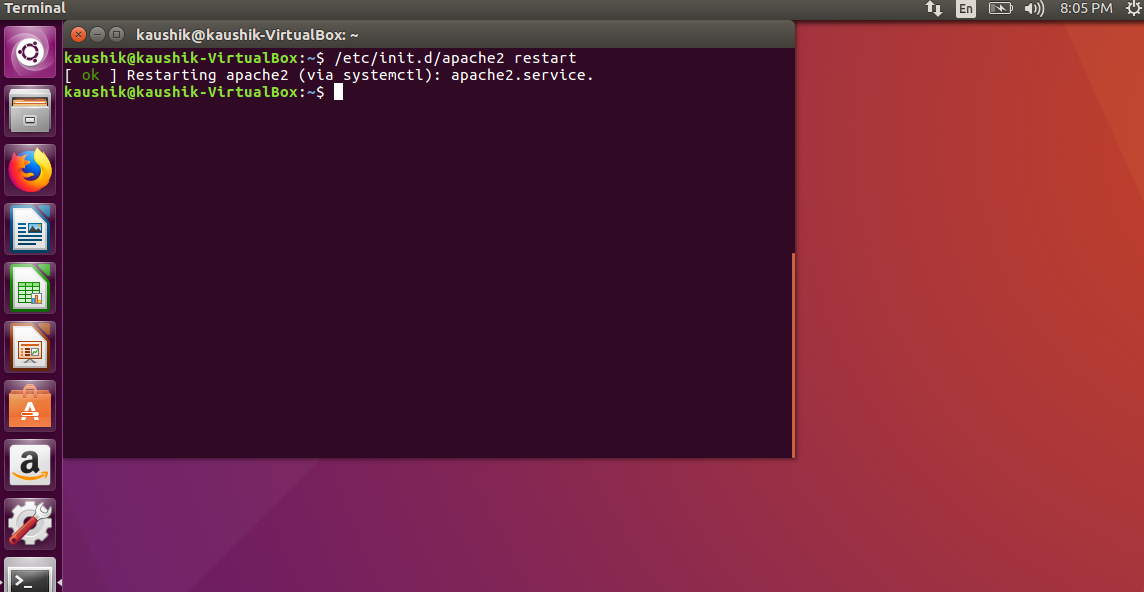
Include /etc/modsecurity/rules/\*.conf

If they are not there, add them. Do not duplicate them, or you risk disabling your Apache service.



**Step 16:** Again, Restart the Apache service:

/etc/init.d/apache2 restart



**Step 17:** Open the default Apache configuration:

sudo nano /etc/apache2/sites-available/000-default.conf

At the bottom of the file, just above the </virtualhost> tag, add the following lines:

SecRuleEngine On

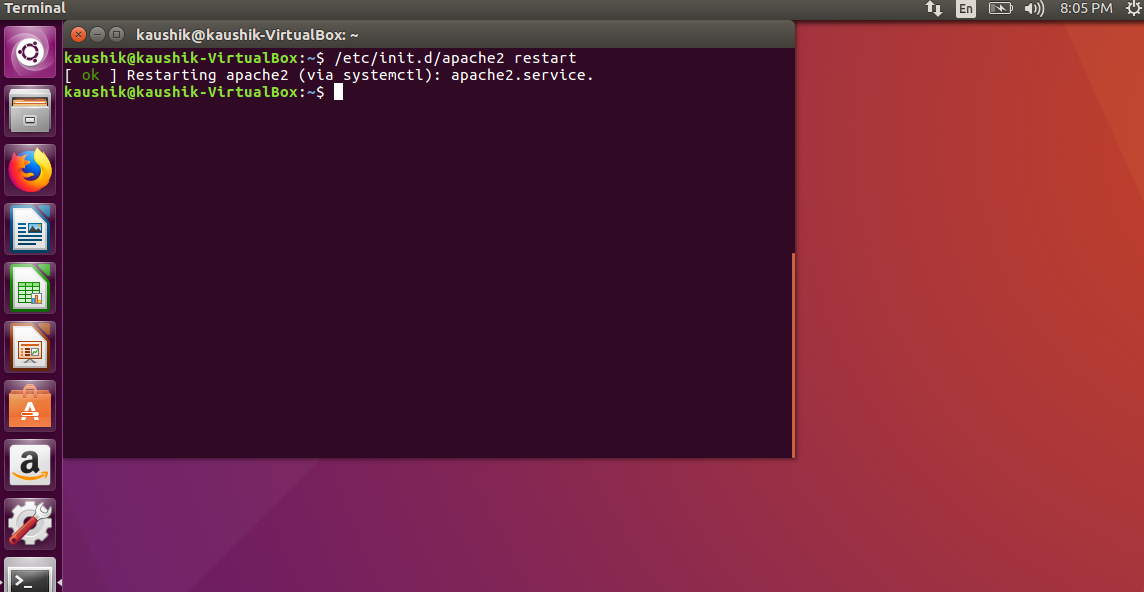
SecRule ARGS:testparam “@contains test” “id:1234,deny,status:403,msg:’The test rule was triggered’”

Save and exit the file (CTRL+X > y > Enter)



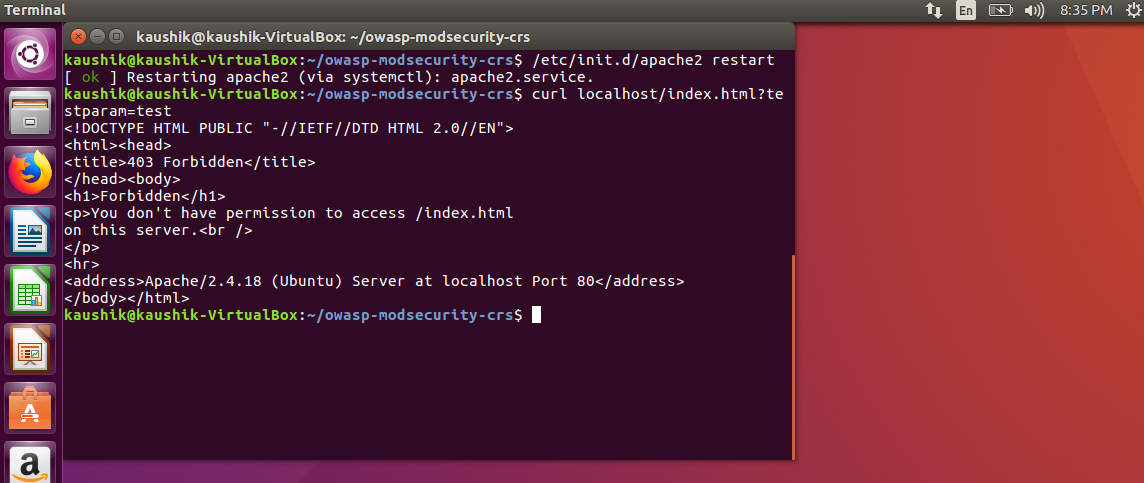
**Step 18:** Again, Restart the Apache service:

/etc/init.d/apache2 restart



**Step 19:** Then enter the following command:

curl localhost/index.html?testparam=test

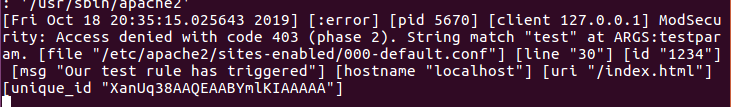


**Step 20:** The system should respond by attempting to display the default webpage. Instead, it will generate 403 forbidden error codes. You can confirm this by displaying the Apache error logs with the command:

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



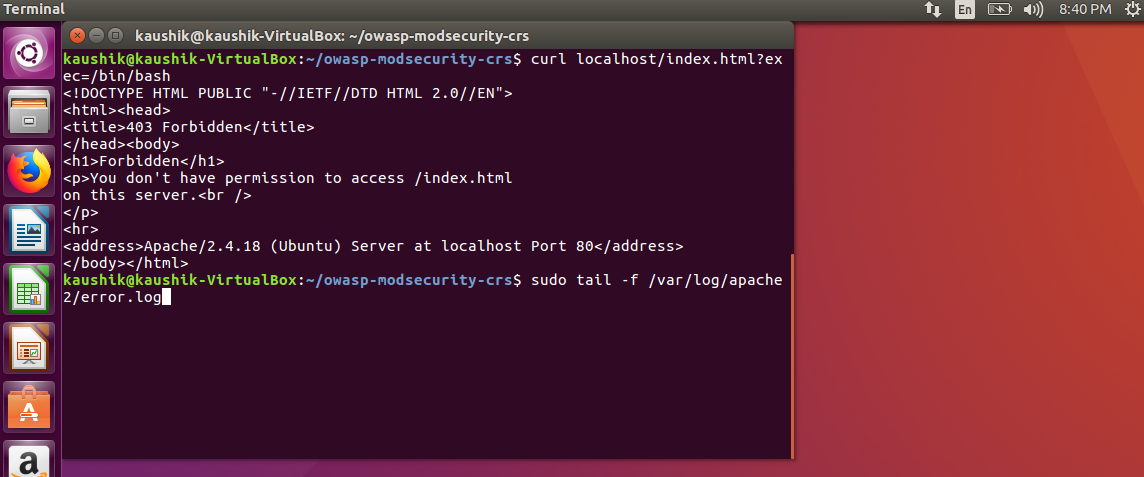
The last entry on the list should have a log of the error test.



**Step 21:** Enter the following:

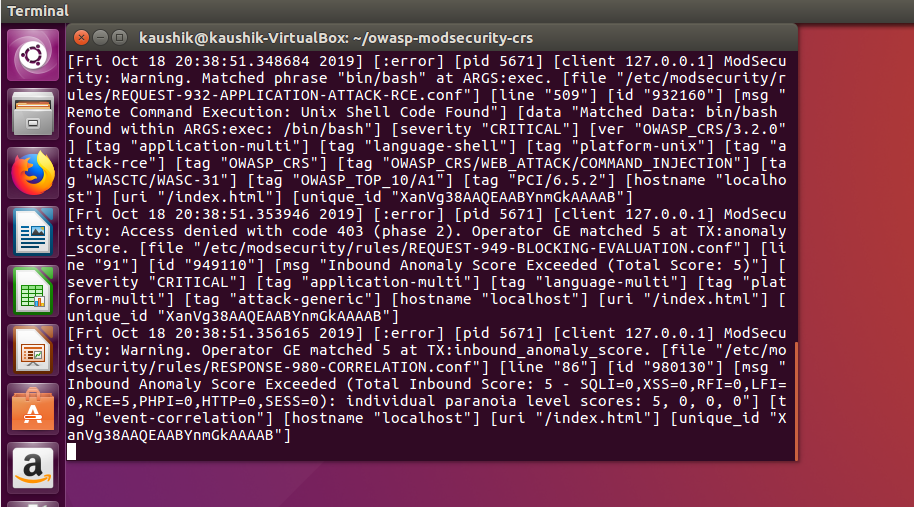
curl localhost/index.html?exec=/bin/bash

Again, you should receive a 403 forbidden error.



Display the Apache error logs with the command:

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



**Conclusion:** Thus, we have studied, installed and configured ModSecurity, Core Rule Set (CRS) on Apache server successfully!!!