Artifact Genome Project

This material is based upon work supported by the National Science Foundation under Grant No. 1900210. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the author(s) and do not necessarily reflect the views of the National Science Foundation.

1. Assignment Type

Learn By Doing

2. Assignment Name

Web Security: Installation and Configuration of ModSecurity Web Application Firewall

3. Assignment Overview

Attacks on the web are conducted at the application level in 70% of cases. Therefore, enterprises would benefit from deploying a Web Application Firewall(WAF) to ensure system security. It creates an additional layer of security that raises the level of defense for web servers and finds and stops assaults before they reach web applications. In order to add an additional layer of protection by guarding against numerous security risks like SQL injection, Cross-Site Scripting (XSS), Cross-Site Request Forgery (CSRF), and others, online applications must use Web Application Firewalls (WAFs). By analyzing and filtering incoming traffic in real-time to stop harmful requests before they can harm the application or steal critical data, WAFs serve as a barrier between the web application and the internet. WAFs can also assist with adhering to different security standards and regulations, including PCI-DSS. They work on the application layer (the 7th layer in the OSI model). This module discusses the steps to install an open-source web application firewall (WAF) called ModSecurity in an Ubuntu 20.04.03 operating system environment. This module also discusses the steps to add the Open Web Application Security Project (OWASP) ModSecurity Core Rule Set, which is updated regularly and can block a wide range of generic attacks, including OWASP's top-ten list of critical vulnerabilities. ModSecurity examines the incoming requests on the apache server, compares them to patterns described in the rules in the ruleset, and takes action on the requests based on the results of the tests. If the check succeeds, the HTTP request is passed to the website to retrieve the content. If not, pre-defined actions are performed. After completing the proper configuration, ModSecurity Firewall is tested by sending some malicious SQL Injection requests to the Apache server and seeing if the requests are being blocked or not.

4. Learning Objectives

- To install and configure ModSecurity Web Application Firewall
- To Add OWASP ModSecurity Core Rule Set to Apache configuration

- To understand the commands used in Linux filesystem

5. Tools Needed

- Oracle VirtualBox 6.1
- Ubuntu 20.04.3
- Kali Linux Debian (64-bit or 32 bit)

6. Artifact Tags

Demo Website running on Ubuntu Operating System - Open Virtual Application file (.ova)

7. Task 1: Artifact Investigation

Click on the Artifact Tag above or search the artifact names within AGP. Read through each artifact and gather an understanding of all the presented information. You may view the download files through the artifact view, or if a different view is desired, you may use the download files. Review the information. Once you have a basic understanding of the information these artifacts are presenting, continue with Task 2.

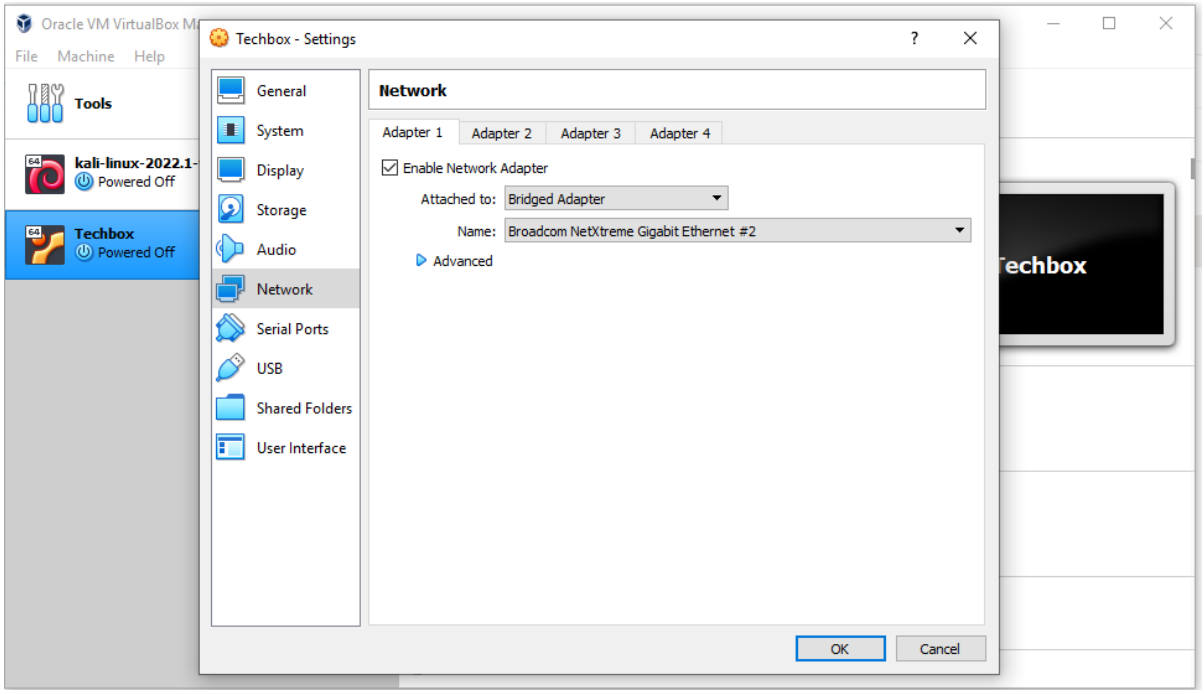
8. Task 2: Load the demo website in Oracle VirtualBox

- If you do not have VirtualBox already installed please use the link above to install it.
- Once VirtualBox is installed click on the artifact and download it.
- After installing VirtualBox, open it and import the artifact which contains the demo website.
- Afterwards import the downloaded Kali Linux file.

]

9. Task 3: Configure Internet Connection on the Ubuntu VM

- Click on the settings of the Ubuntu VM and go to the network settings, once there attach to the Bridged Adapter.
- Then choose any of the options available in the dropdown.

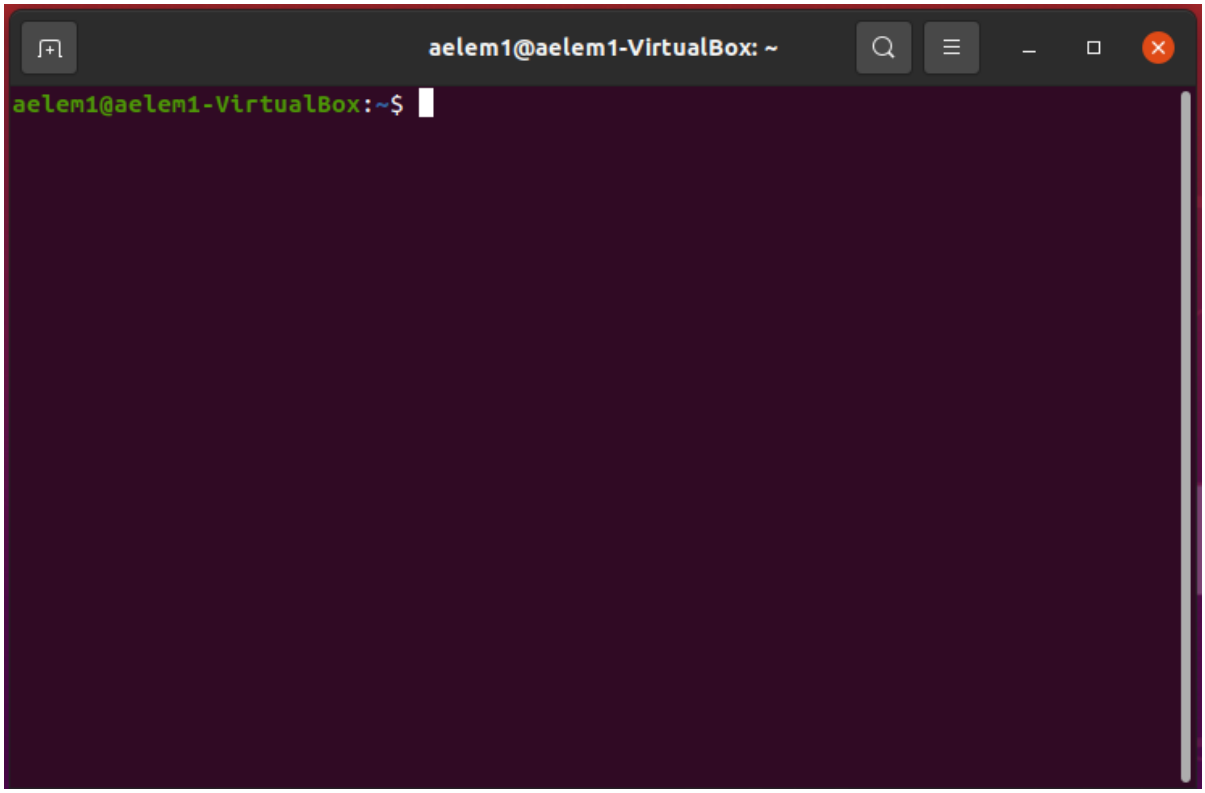


Ubuntu VM connected to bridged adapter

- Click OK and start the Ubuntu VM.

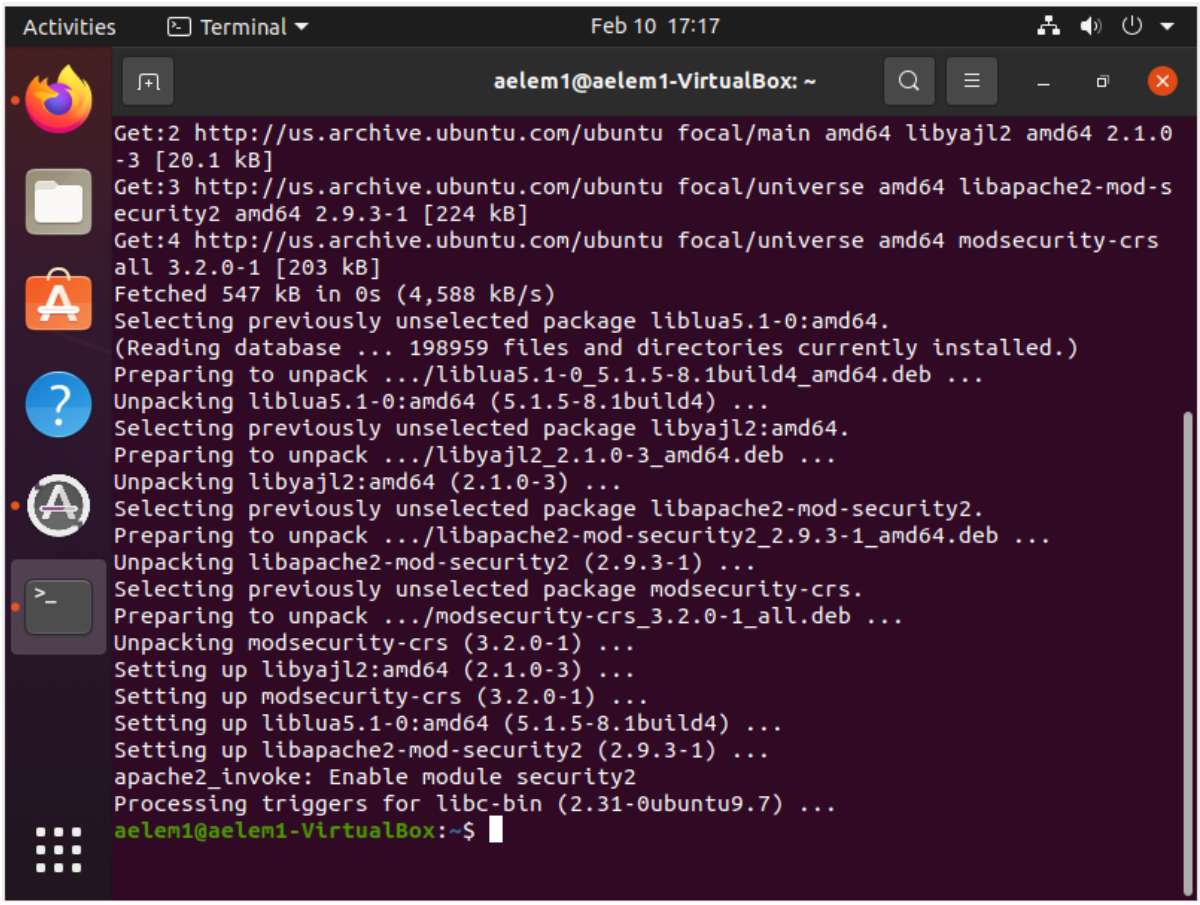
10. Task 4: Installation of ModSecurity with Apache on Ubuntu 20.04.3

- The Ubuntu operating system password is sqlinjection$12&
- Open the terminal on the Ubuntu VM as shown below.
- Here is what the output might look like:



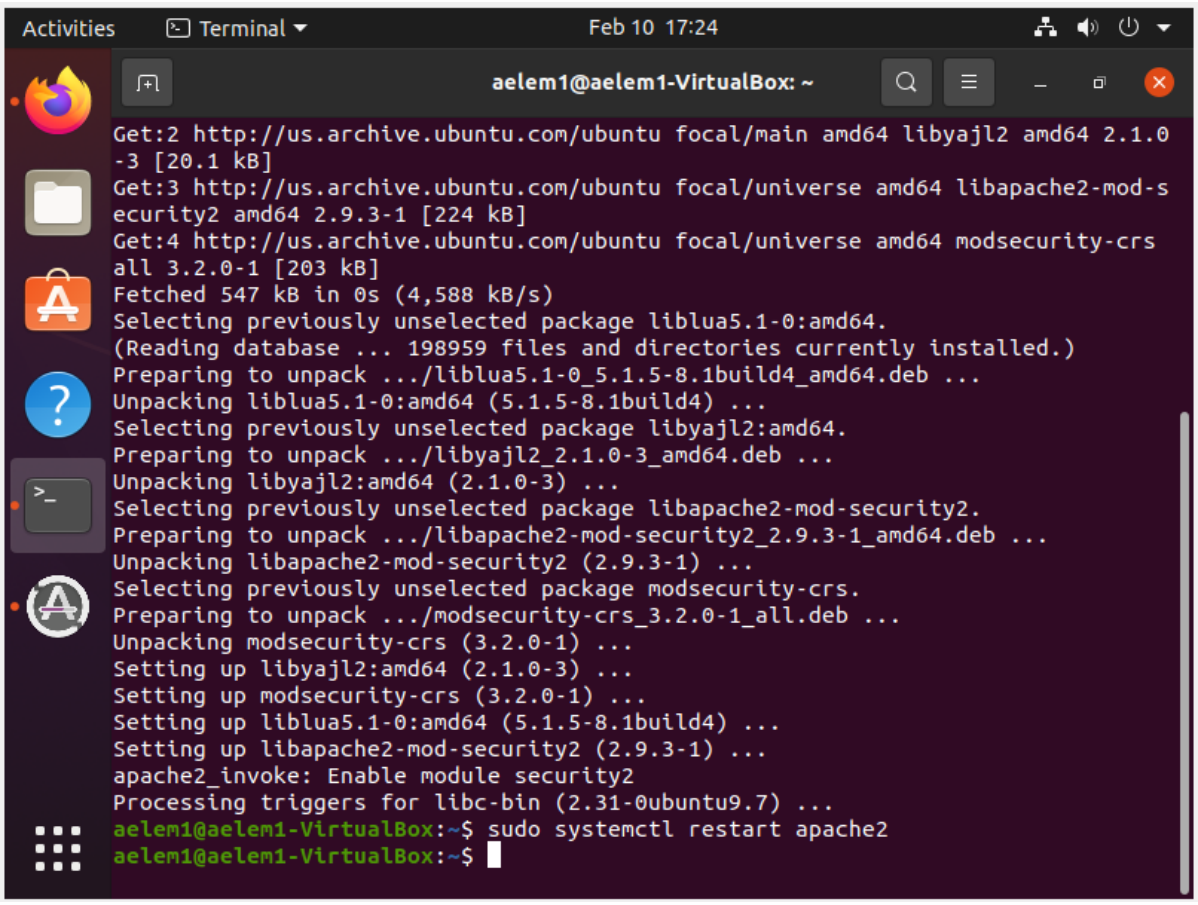
Ubuntu Terminal

- The ModSecurity module for Apache is available in the default Debian/Ubuntu repository.
- To install it, run the following commands in the terminal: sudo apt-get install libapache2-mod-security2
- You will be prompted for the password which is sqlinjection$12&
- Enter Y for all questions
- Here is what the output might look like:



Installing ModSecurity

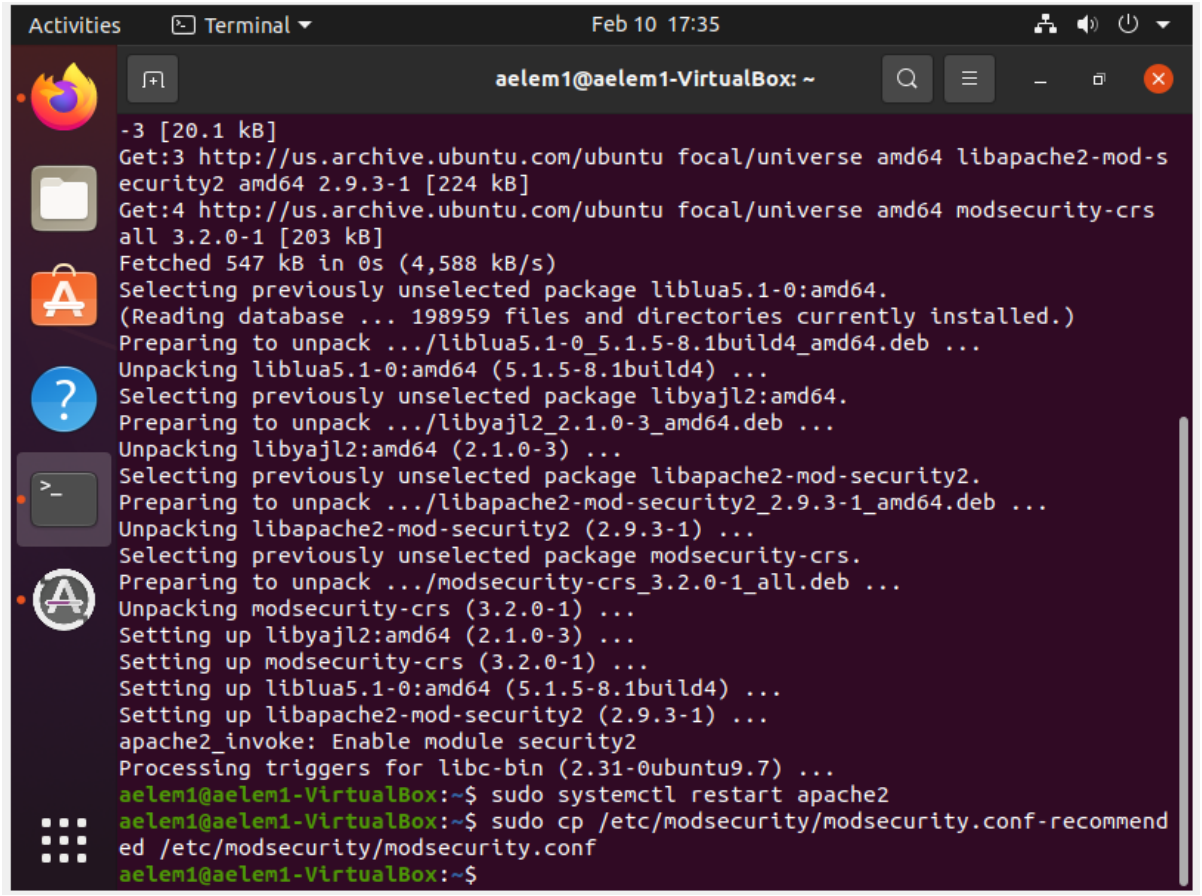
- Restart the Apache service by running the command: sudo systemctl restart apache2
- Here is what the output might look like:



Restarting Apache

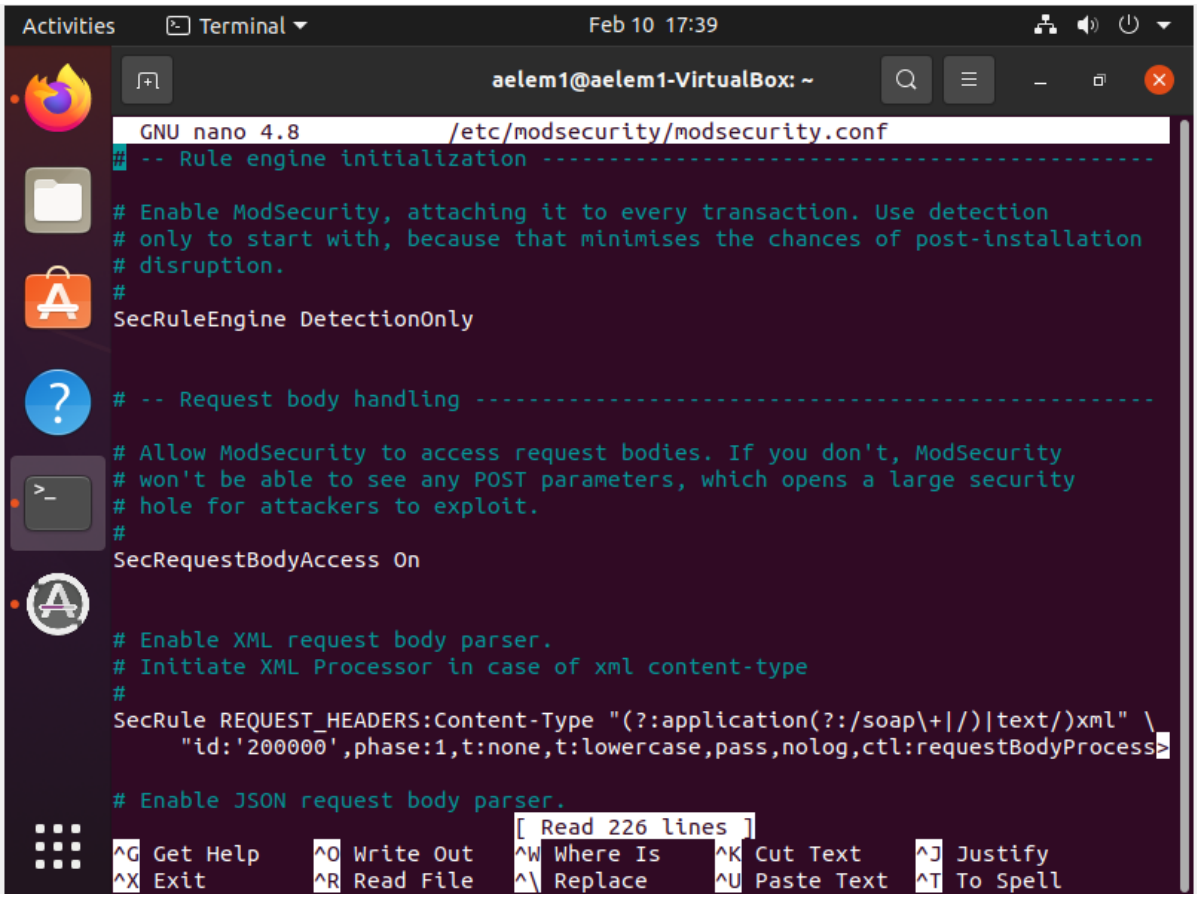
11. Task 5: Post-installation configuration of ModSecurity

- The default rule of ModSecurity is to log suspicious activity. And we need to change the configuration to detect and block traffic according to our requirements.
- By default, the configuration file is at /etc/modsecurity/modsecurity.conf-recommended and we need to copy this file and rename it as modsecurity.conf
- Run the command: sudo cp /etc/modsecurity/modsecurity.conf-recommended /etc/modsecurity/modsecurity.conf
- Here is what the output might look like:



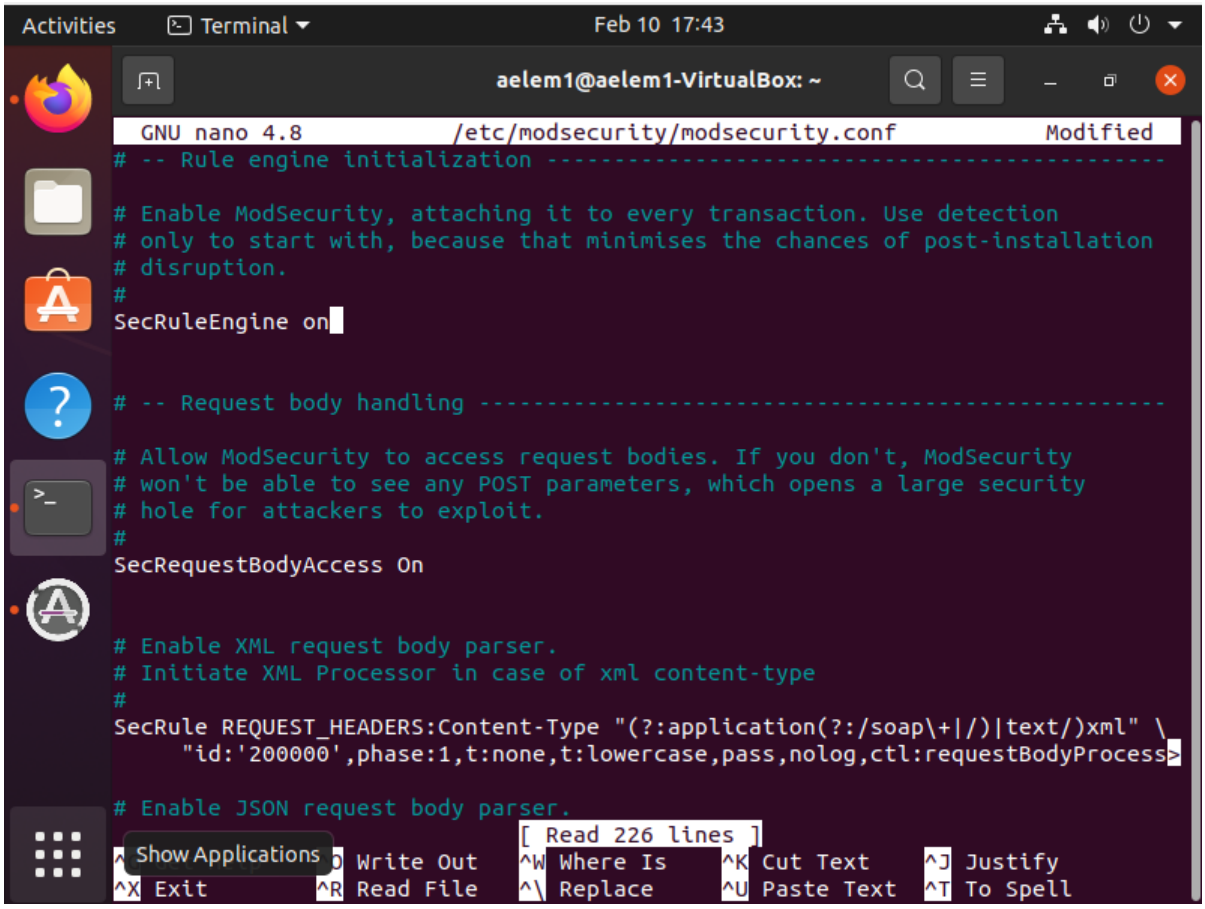
Renaming ModSecurity

- Next, we must change the ModSecurity detection mode to detect and block the requests.
- Run the command: sudo nano /etc/modsecurity/modsecurity.conf
- Here is what the output might look like:



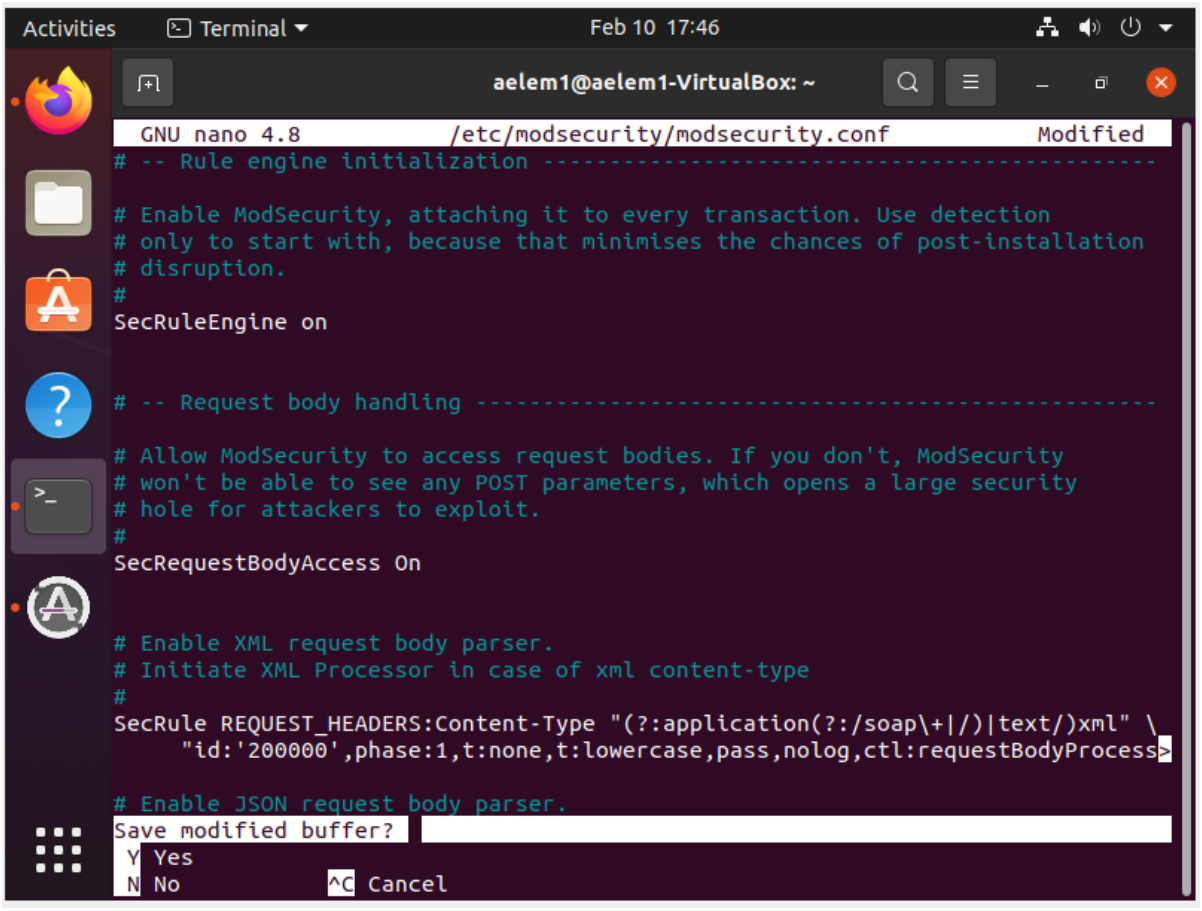
ModSecurity Settings

- Locate SecRuleEngine DetectionOnly and change it to SecRuleEngine on
- Here is what the output might look like:



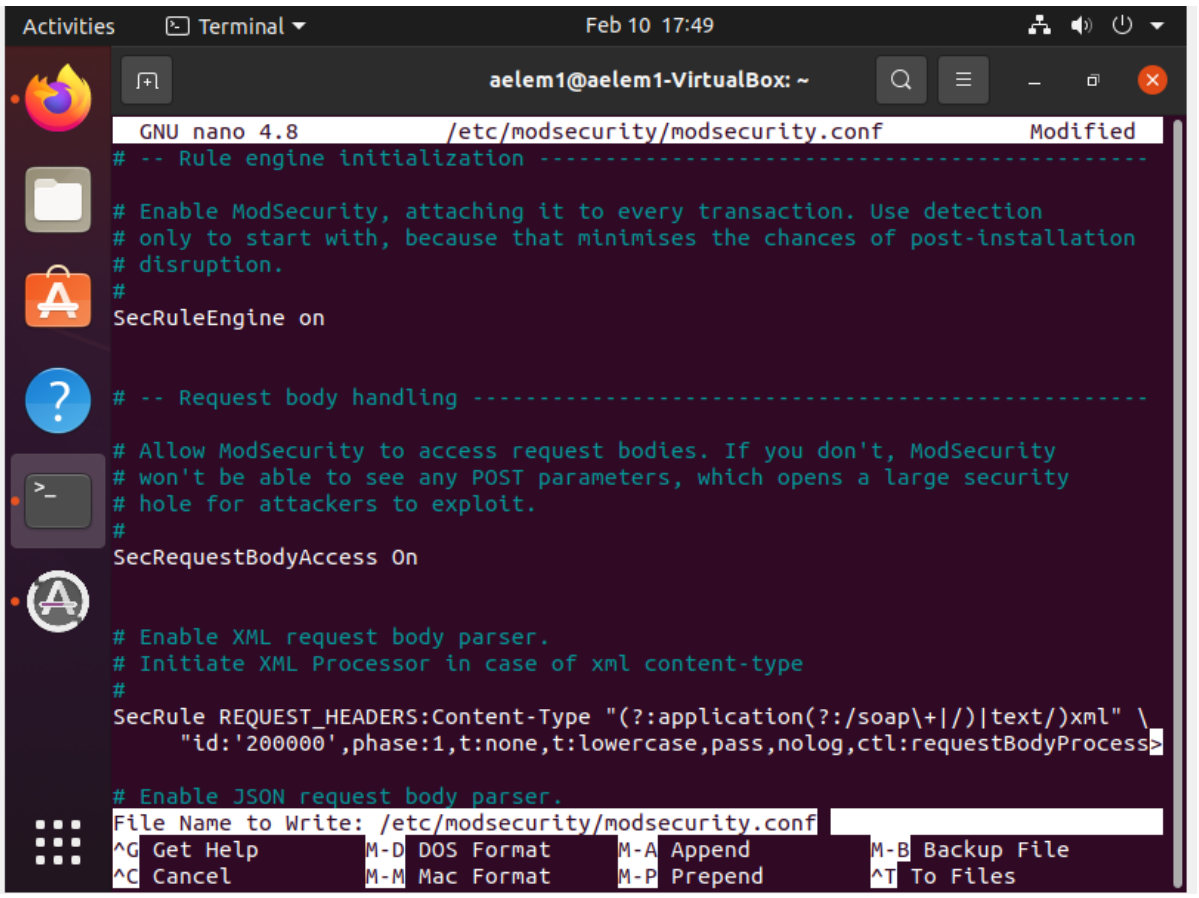
Turning on ModSecurity

- Press Ctrl+X to save the file
- Here is what the output might look like:



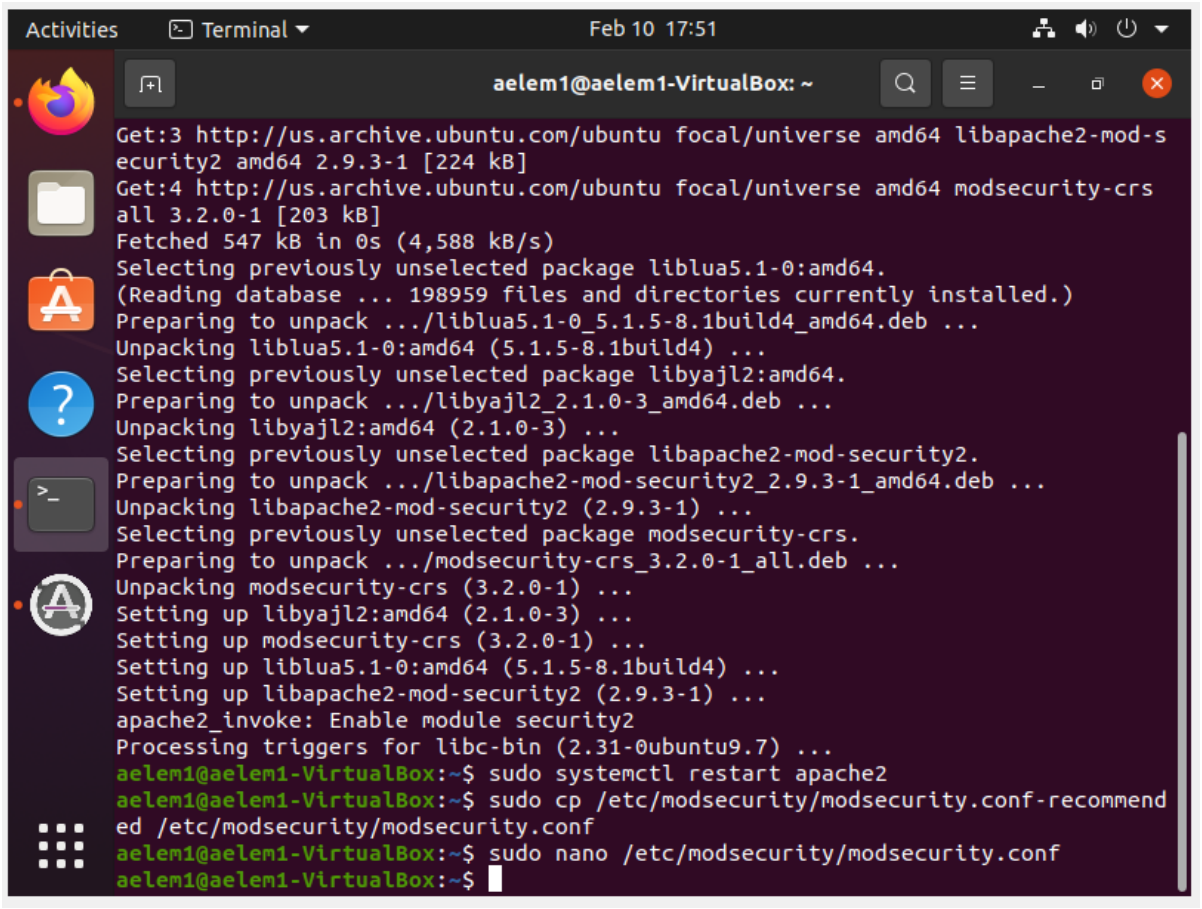
After pressing Ctrl+X

- Press Y
- Here is what the output might look like:



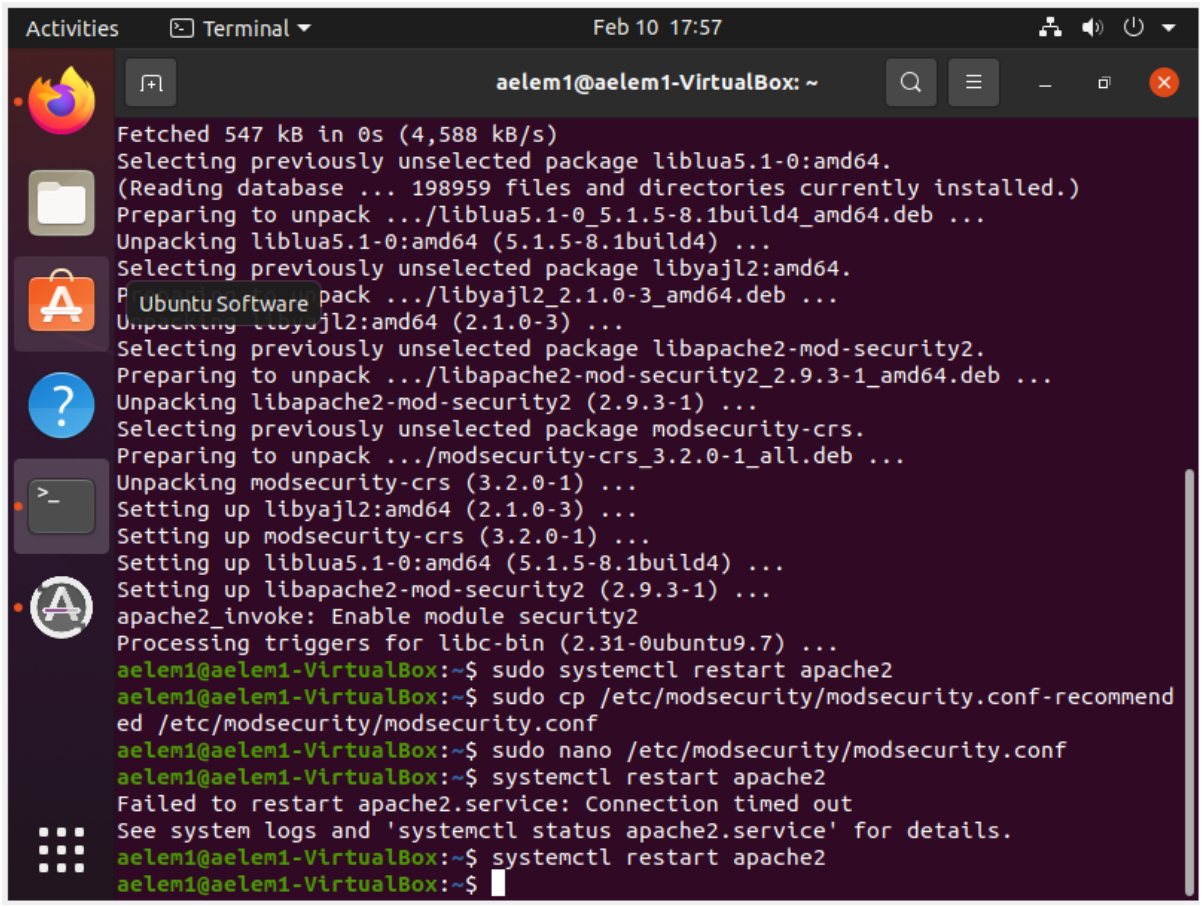
After pressing Y

- Press the Enter key
- Here is what the output might look like:



After pressing the enter key

- Next, Restart the Apache service for the changes to take effect.
- Run the command: systemctl restart apache2 and enter the ubuntu VM password - sqlinjection$12& when prompted
- Here is what the output might look like:

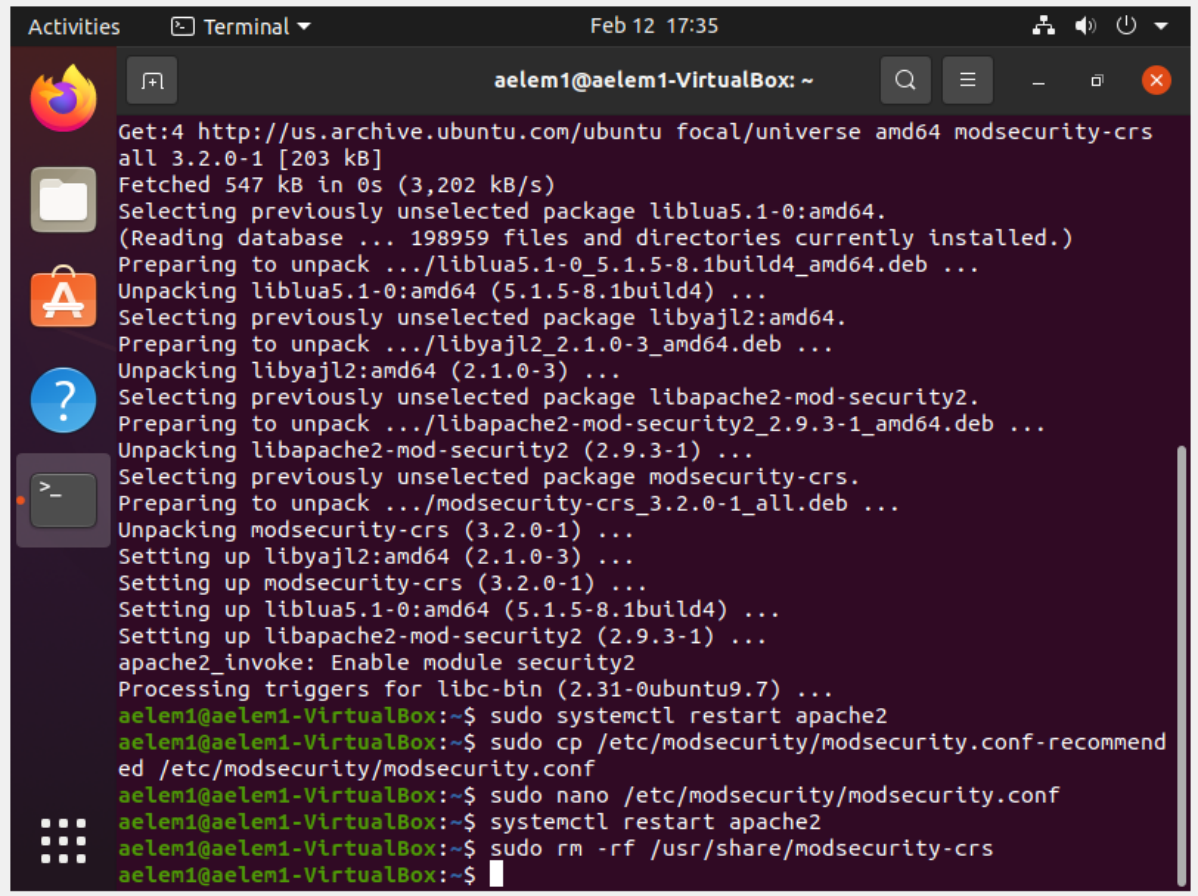


After restarting Apache

- This will turn to ModSecurity using the basic default rules.
- Versions of Linux come with OWASP Core Rule at user/share/modesecurity-crs directory.
- It is recommended to download the latest CRS from the GitHub repository since the developers frequently update it.

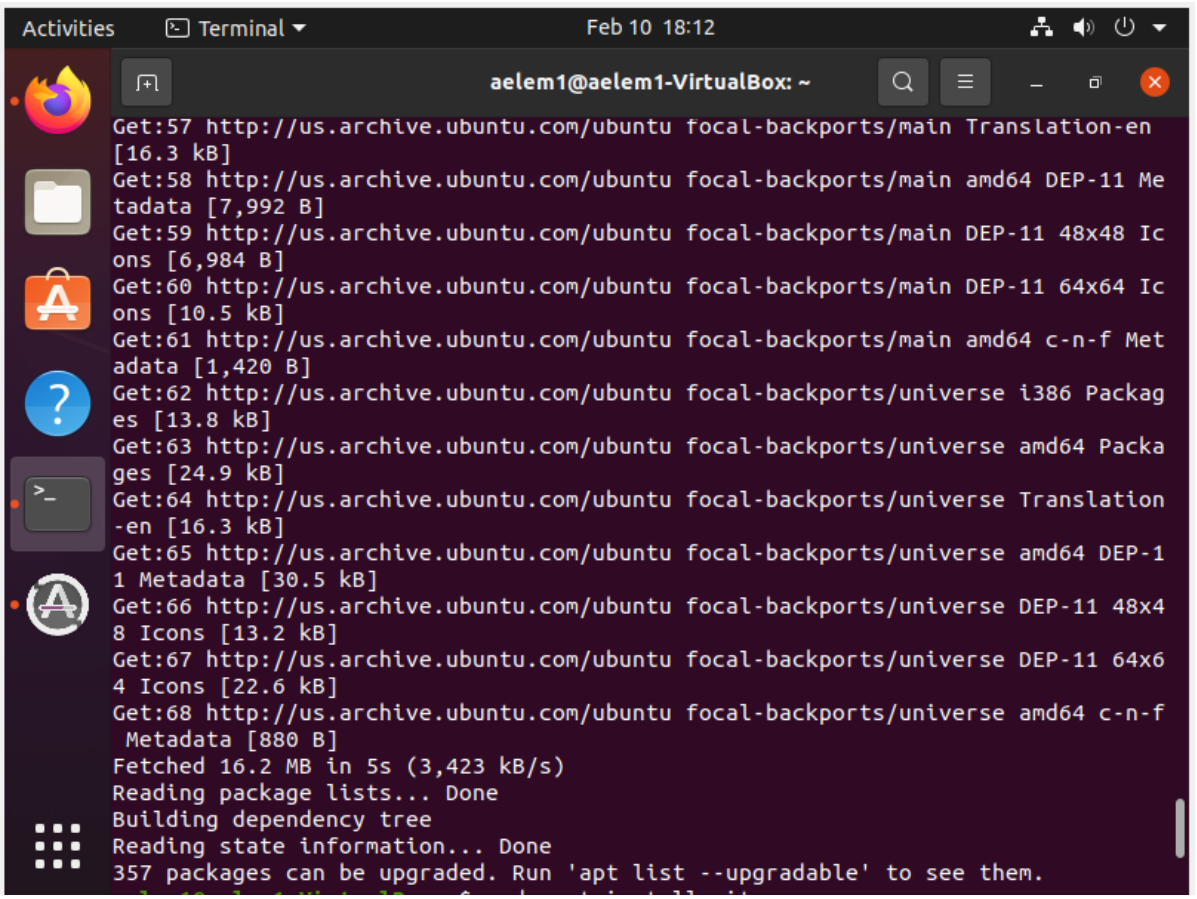
12. Task 6: Adding OWASP ModSecurity rules

- Remove the default Core Rule Set (CRS).
- Run the command: sudo rm -rf /usr/share/modsecurity-crs
- Enter the password - sqlinjection$12& when prompted
- Here is what the output might look like:



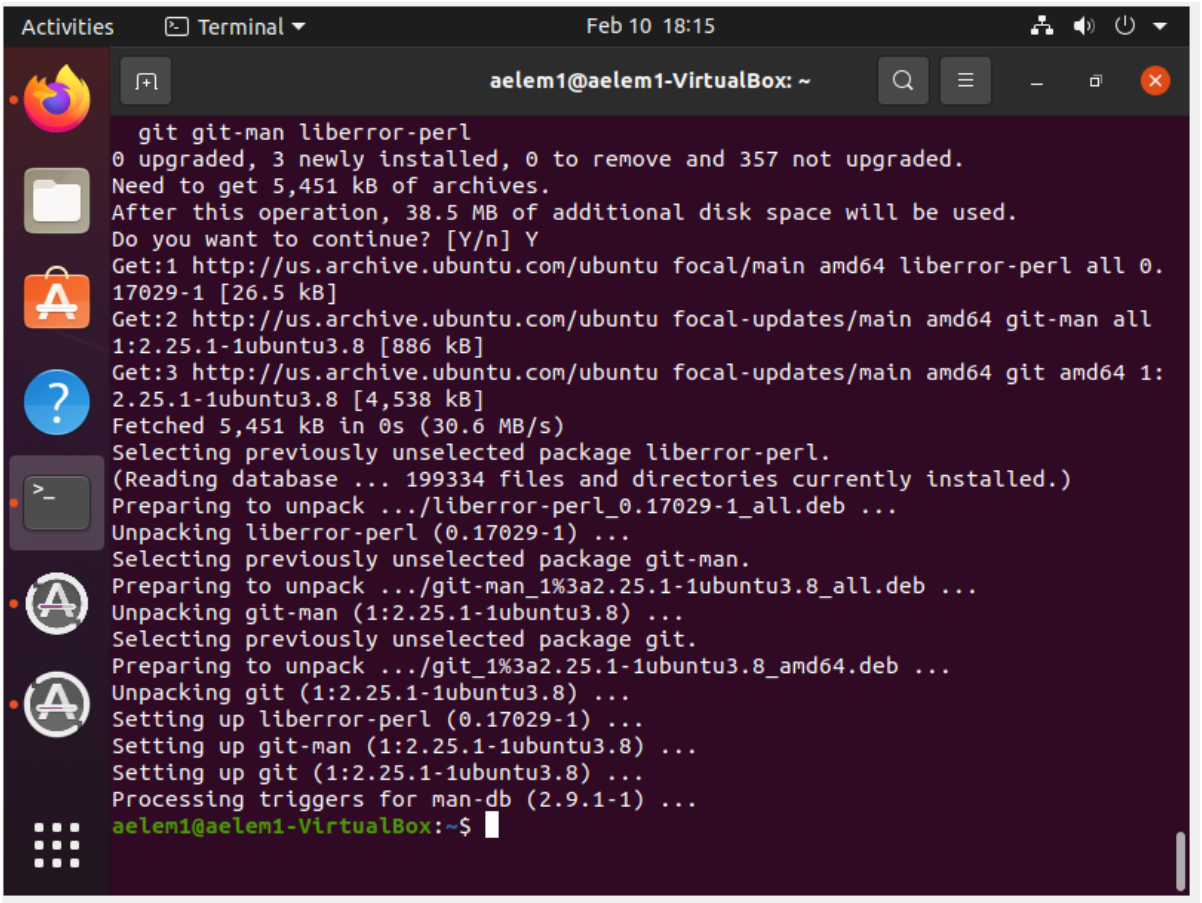
After removing default CRS

- Update your local package index
- Run the command: sudo apt update
- Here is what the output might look like:



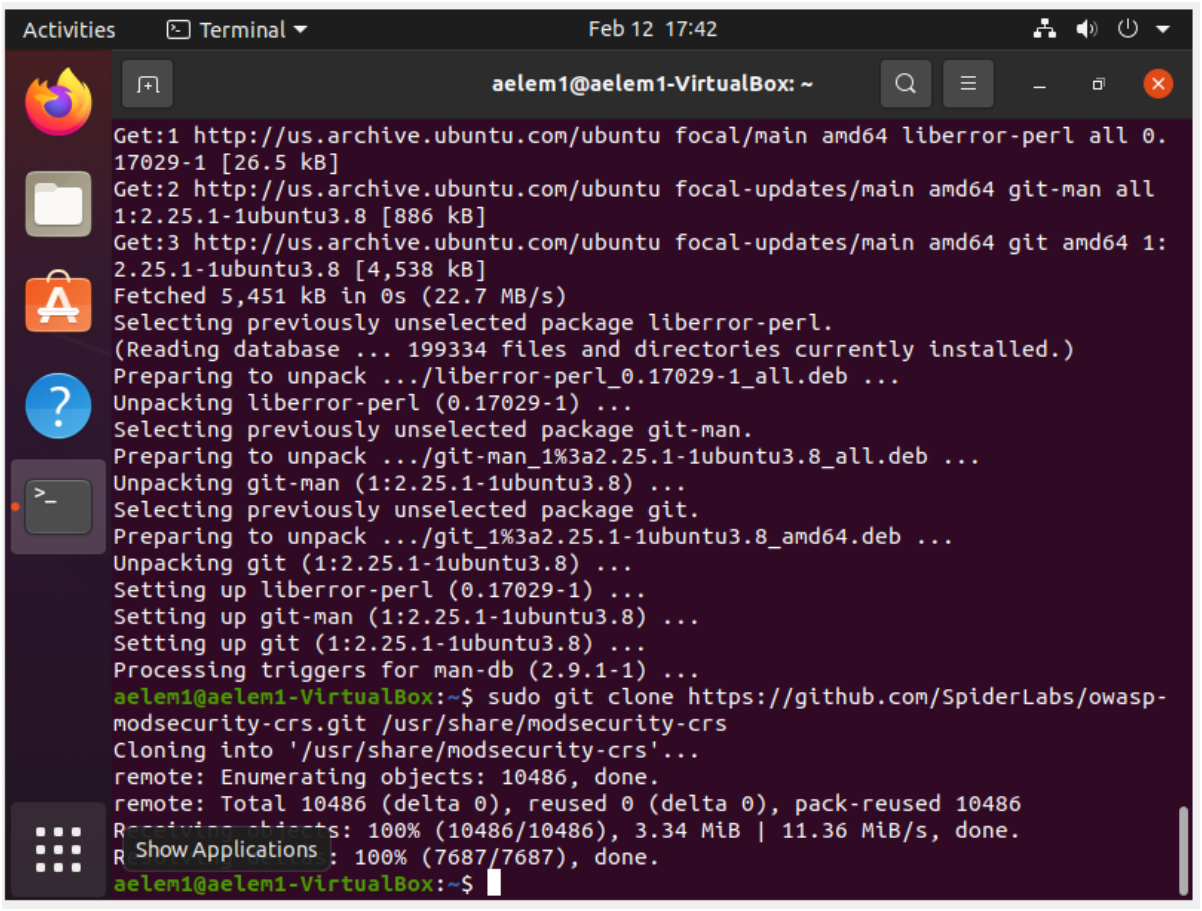
After Updating the local package

- Install Git
- Run the command: sudo apt install git and enter Y for any prompt question
- Here is what the output might look like:



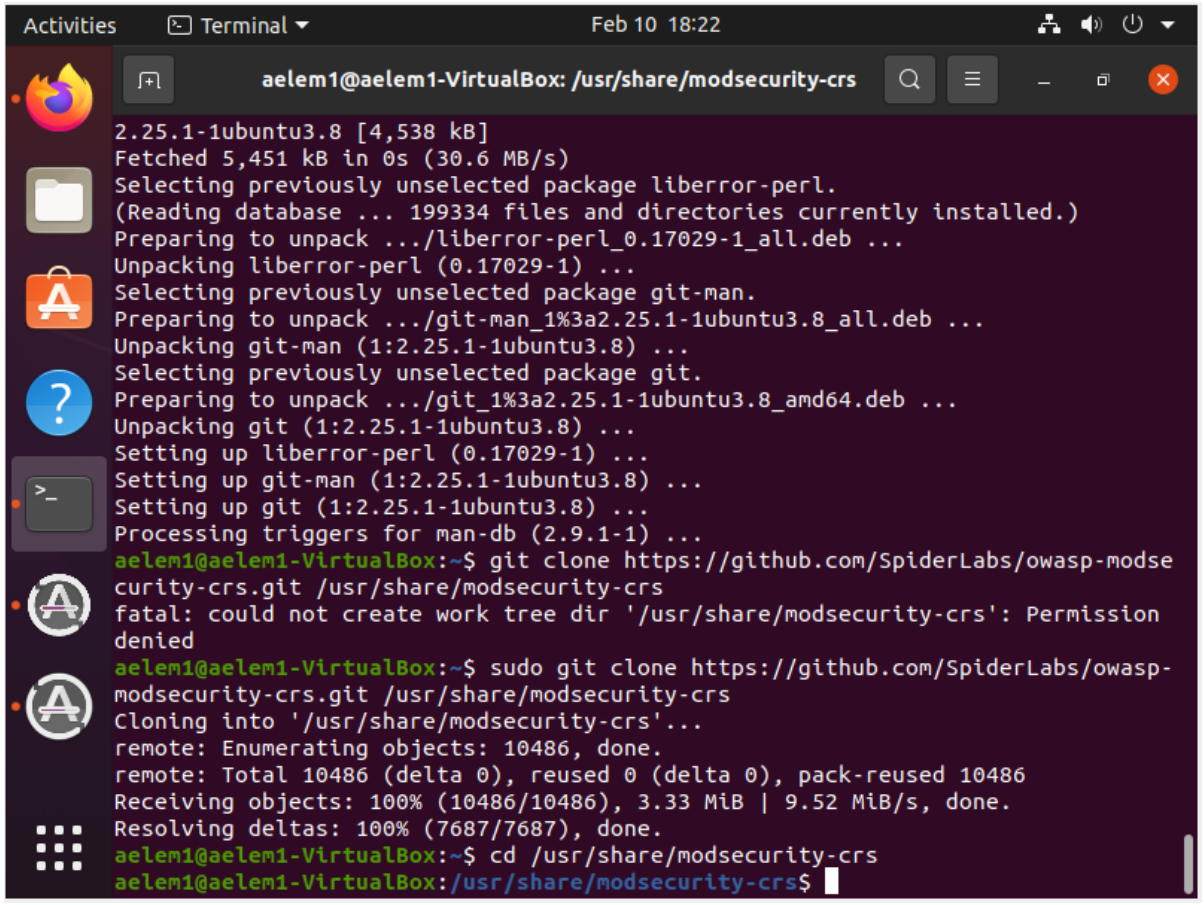
After installing Git

- Clone the latest OWASP CRS from GitHub to the /usr/share/ directory
- Run the command below:
- sudo git clone https://github.com/SpiderLabs/owasp-modsecurity-crs.git /usr/share/modsecurity-crs
- Here is what the output might look like:



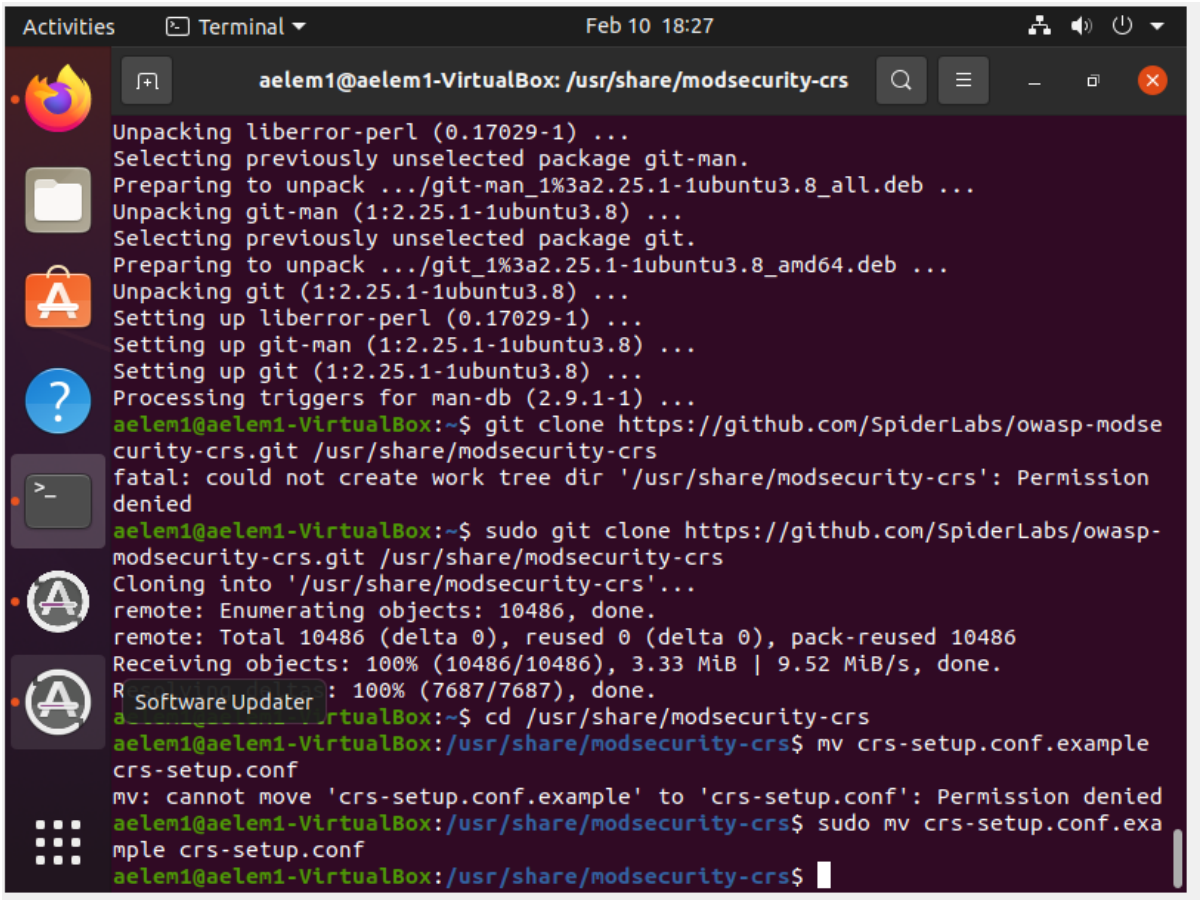
After cloning OWASP CRS from Github

- Switch to /usr/share/modesecurity-crs directory.
- Run the command: cd /usr/share/modsecurity-crs
- Here is what the output might look like:



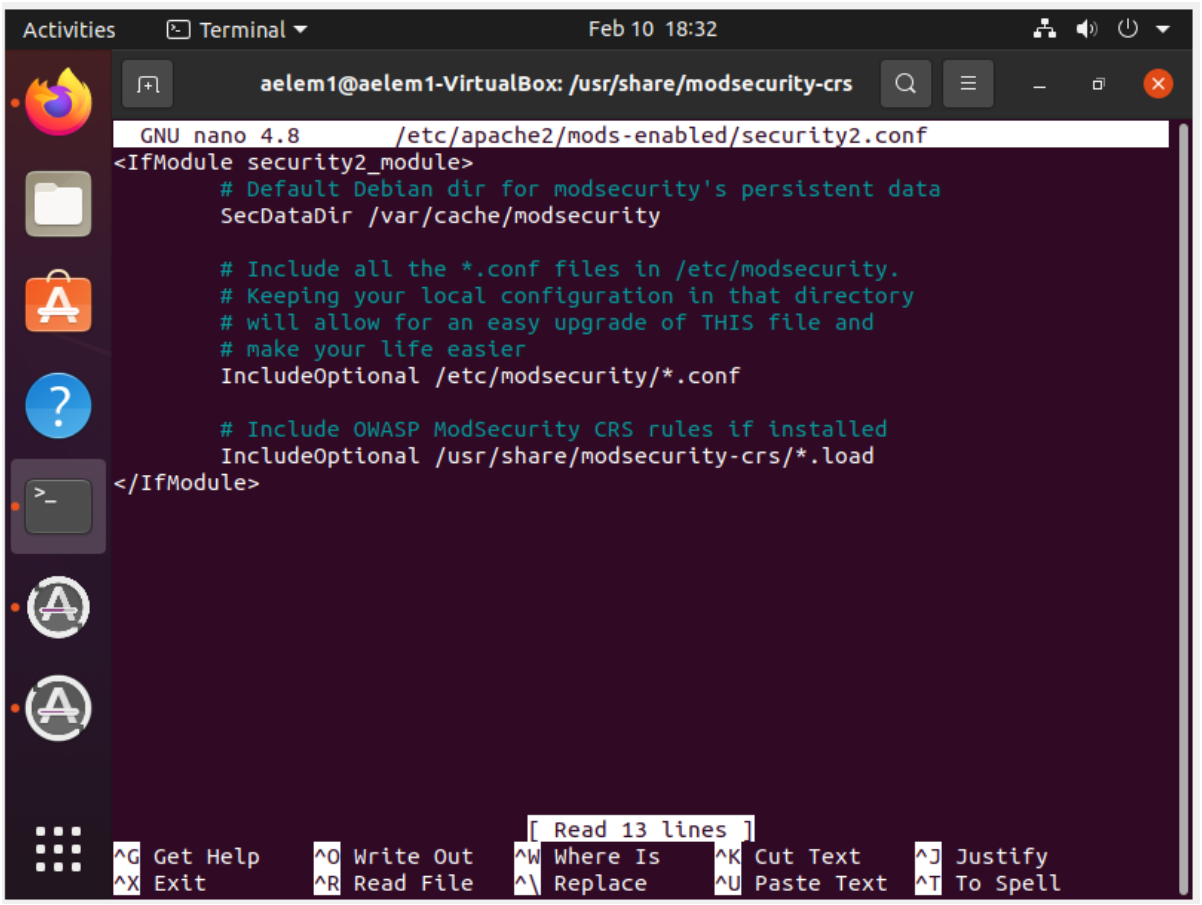
After changing to modsecurity directory

- Rename the example setup file
- Run the command: sudo mv crs-setup.conf.example crs-setup.conf
- Here is what the output might look like:



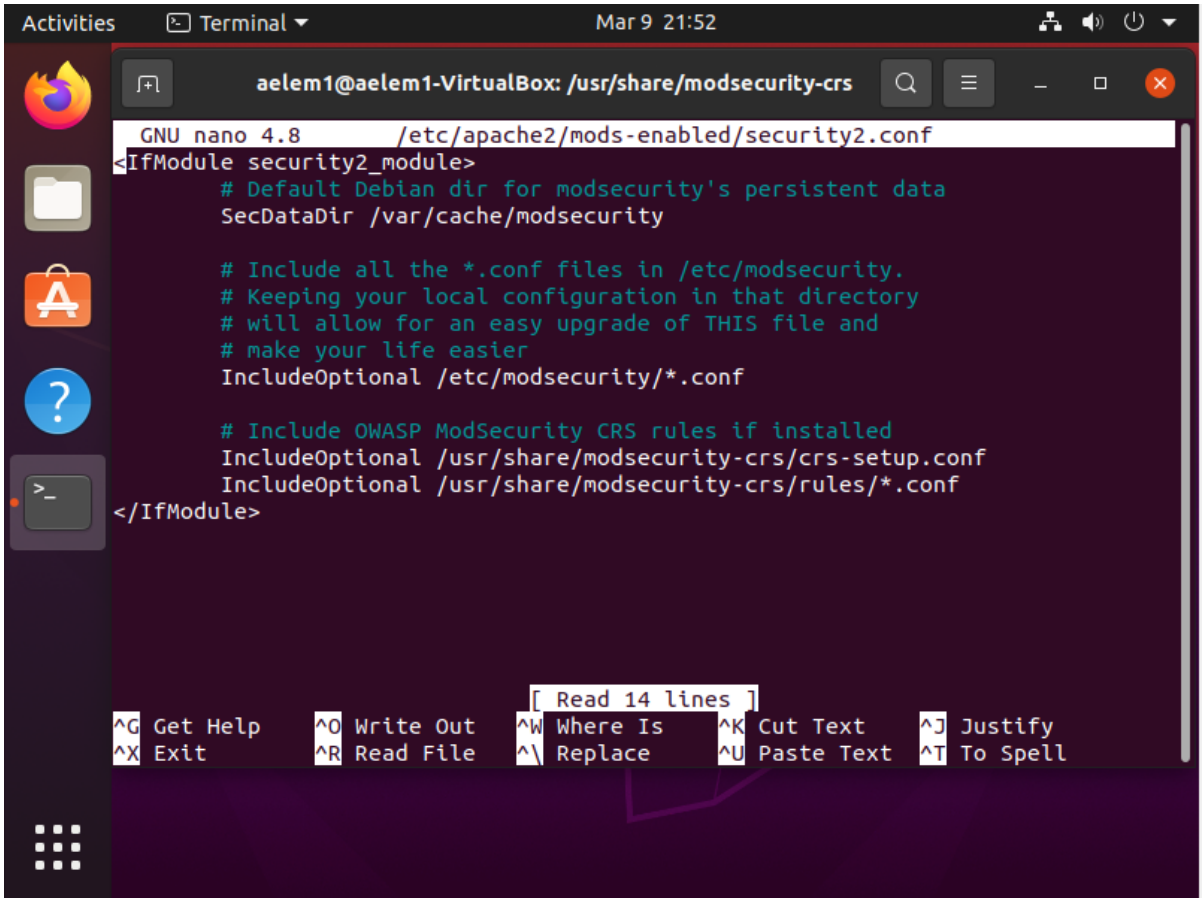
After renaming the example setup file

- Edit the rules in the security2 configuration file to get it working with Apache by making the required edits in the /etc/apache2/mods-enabled/security2.conf file
- Run the command: sudo nano /etc/apache2/mods-enabled/security2.conf
- Here is what the output might look like:



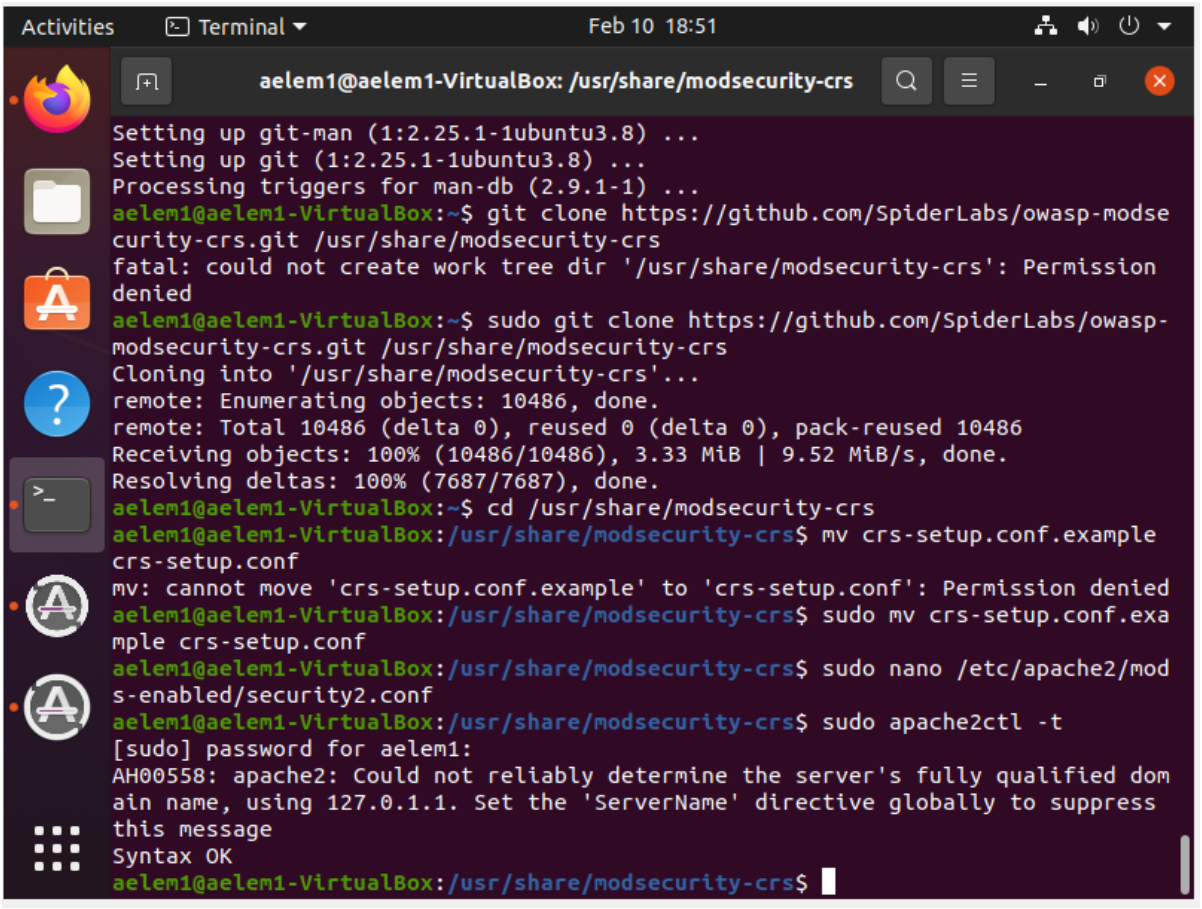
After opening security2.conf

- Locate IncludeOptional /usr/share/modsecurity-crs/*.load and change it as shown below:



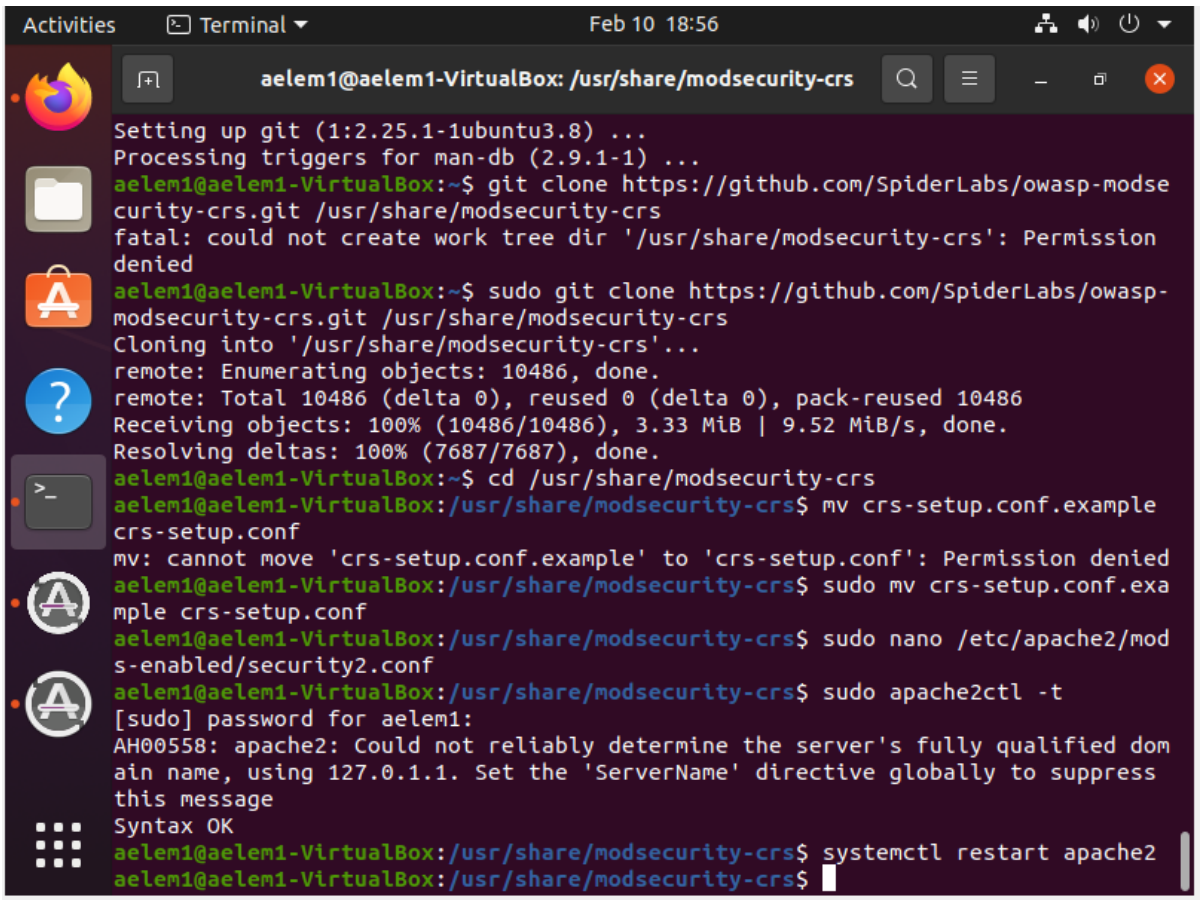
After editing security2.conf

- Save the file
- Test the Apache configuration
- Run the command: sudo apache2ctl -t and enter the password - sqlinjection$12& when prompted
- Here is what the output might look like:



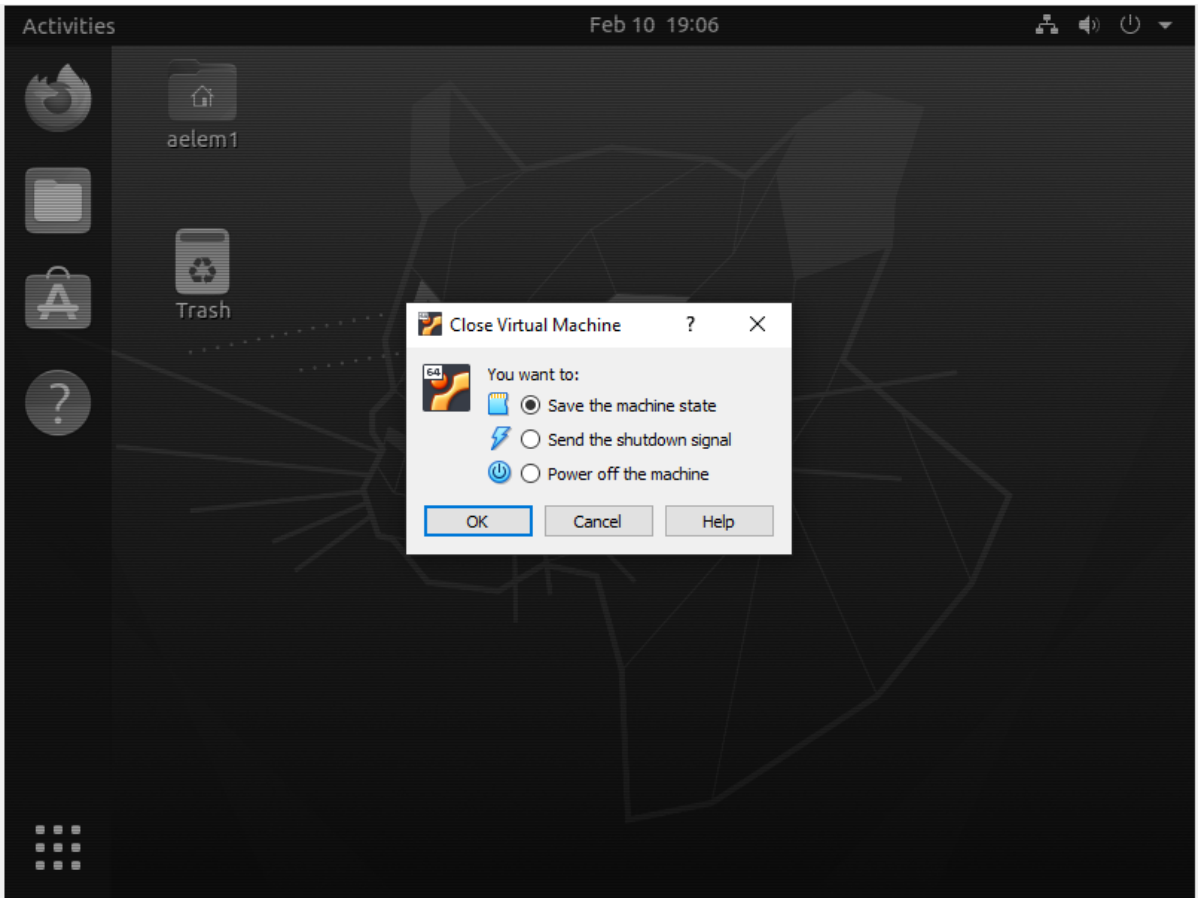
After testing the apache configuration

- Restart Apache
- Run the command systemctl restart apache2 and enter the password: sqlinjection$12& when prompted
- Here is what the output might look like:



After restarting Apache

- At this point, you can save the state of the Ubuntu VM and exit it as shown below:



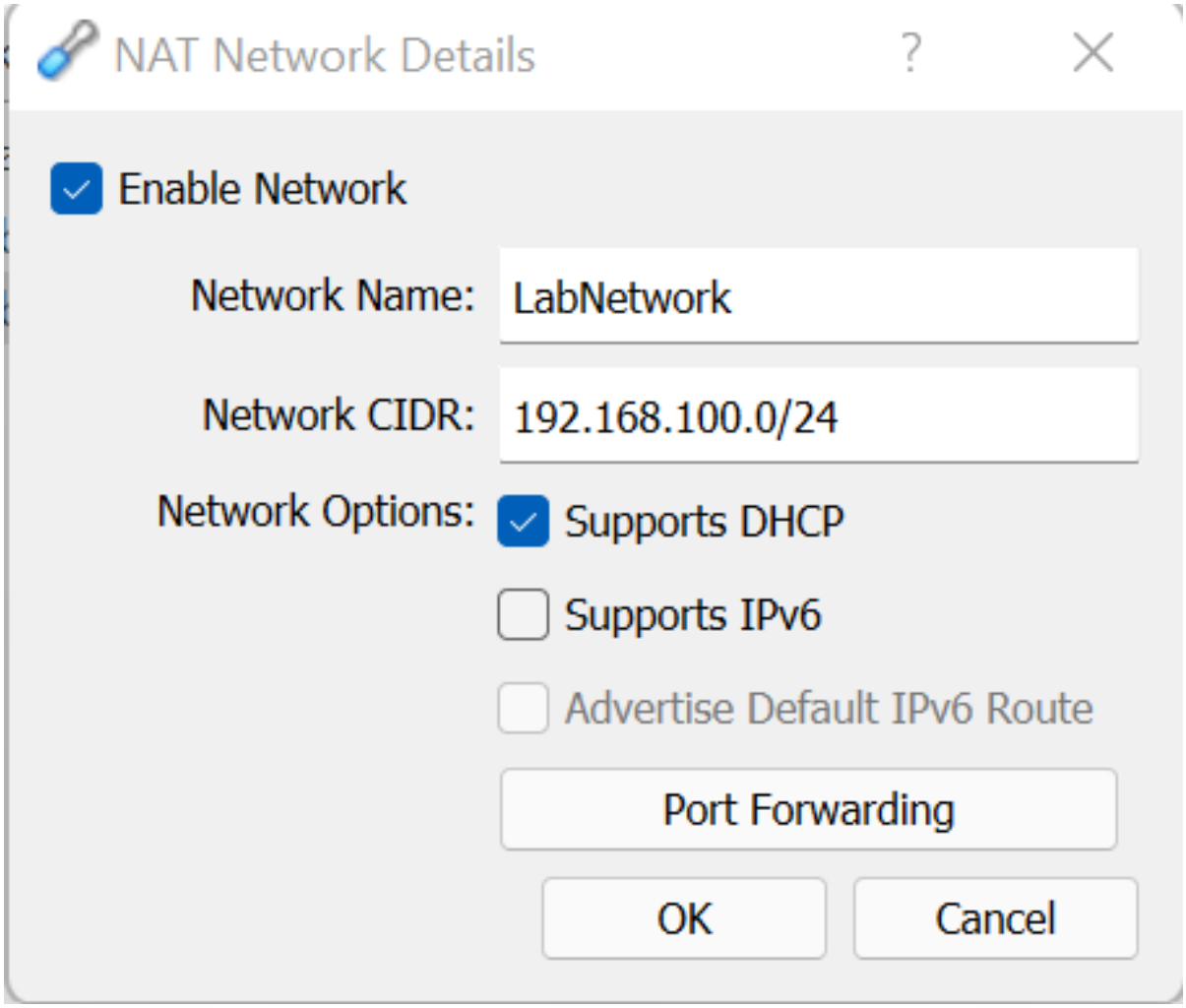
Saving the state of the Ubuntu VM

- Post the proper configuration, we can test ModSecurity by performing SQL Injection attacks from the Kali VM and see if the requests are being blocked.
- We need to first configure network connectivity between the two virtual machines.

13. Task 7: Configure Network Connectivity between the two Virtual Machines

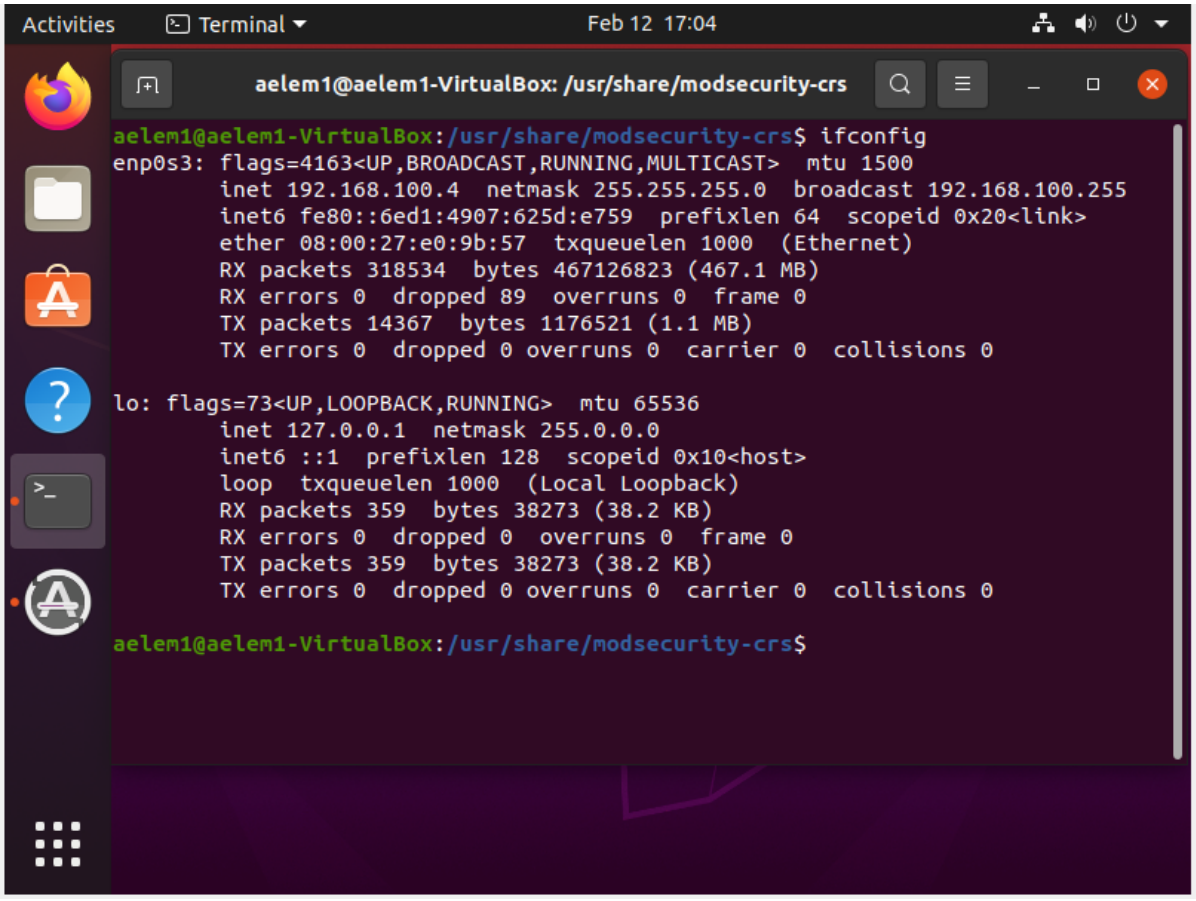
We will set up a separate NAT network for both virtual machines by doing the following:

- Open Oracle VirtualBox, then click on File, click on Preferences.
- Click on Network, click on the + sign to add a new network, and click on the gear icon to rename the Network to "LabNetwork".
- Change the Network CIDR to 192.168.100.0/24 and click OK. It should look like this:



NAT Network Settings

- Click on the settings of the Kali Linux machine and go to the network settings, once there attach the NAT Network.
- Then choose LabNetwork.
- Start the Kali VM. The username and password is 'kali'
- In the Network settings of the Ubuntu virtual machine, connect to NAT Network,
- Then select LabNetwork.
- Start the Ubuntu VM
- In the Ubuntu VM, open the terminal and enter ifconfig to check the IP address.
- Here is what the output might look like:

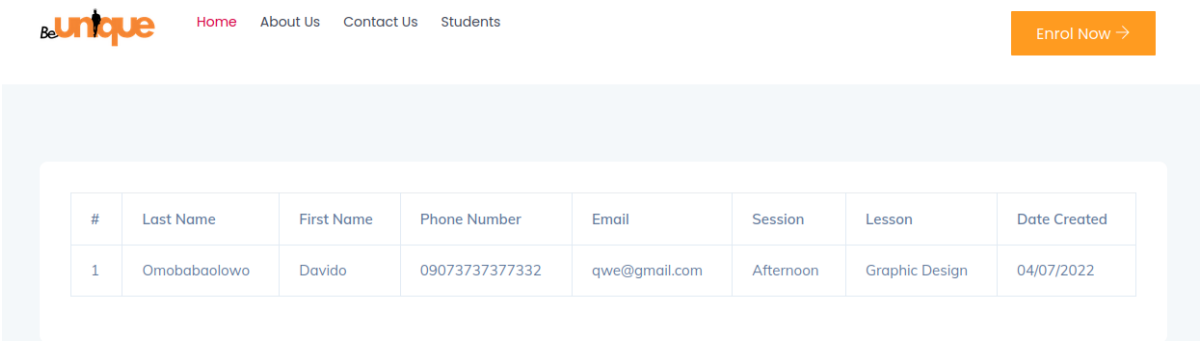


IP Address of Ubuntu VM

- To confirm that both VMs are connected, ping the Ubuntu IP address in the Kali terminal.
- Assuming the Ubuntu VM has the IP address - 192.168.100.5.
- Open the kali terminal and type "ping 192.168.100.5".
- Press Ctrl + Z to stop pinging.

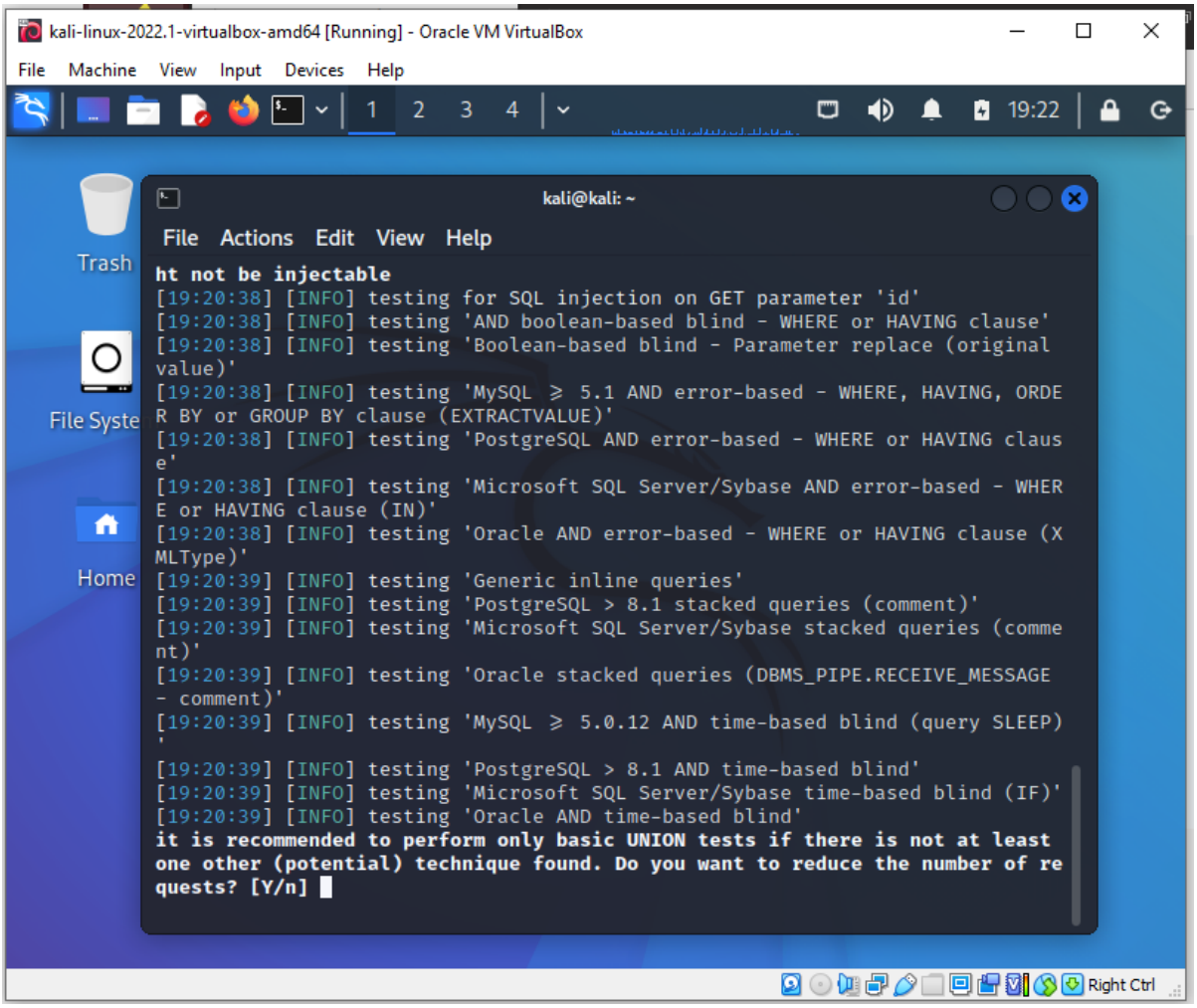
14. Task 8: Testing ModSecurity

- Note: In this section, we assume the IP address of the Ubuntu VM is 192.168.100.5. It could be different for yours.
- In the Ubuntu VM, open the browser and go to the website url: http://localhost/beunique/
- Once there click on the Students tab and select any entry from the table and click view details.
- Copy the current URL from the website.



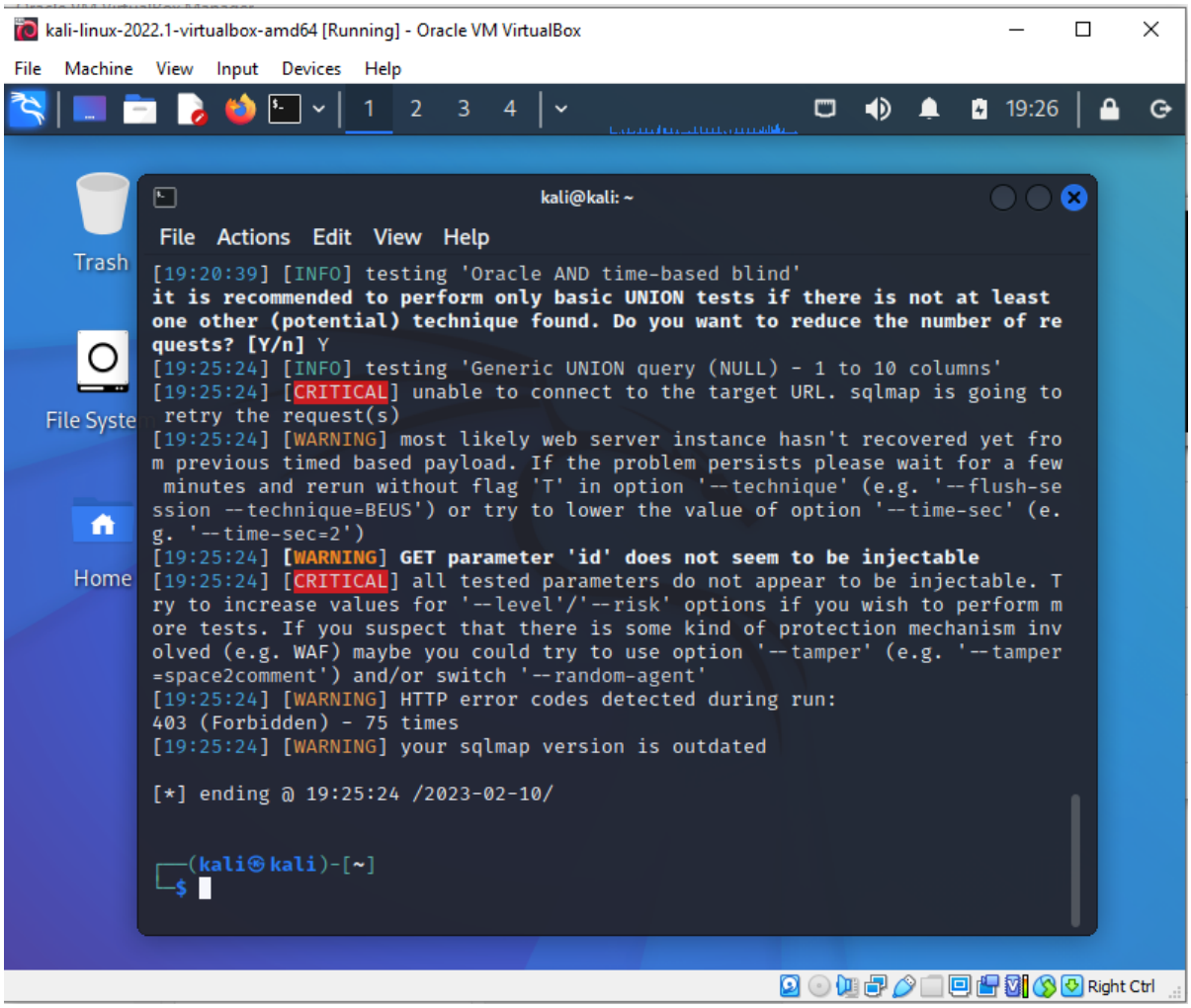
Vulnerable URL in demo website

- In the Kali VM, Open Terminal:
- The first step is to scan the website using `-u` command to see if it is vulnerable to SQL injections and then collect information about it.
- Assuming the url I copied is "http://localhost/beunique/view_student.php?id=001", yours could be different, change the localhost part of the URL to the Ubuntu VM IP address, which in this case is 192.168.100.5 so the url is now "http://192.168.100.5/beunique/view_student.php?id=001"
- Run this command in the Kali-Linux VM: `sqlmap -u http://192.168.100.5/beunique/view_student.php?id=001` Keep in mind your command may look a bit different depending on if your IP address is different and also depending on the entry you chose in bullet point 3.
- Here is what the output might look like:



Scanning the vulnerable URL

- Press 'Y' for all the prompt questions



This is the output of the scan

15. Analysis

- ModSecurity has been configured properly and actively blocks the SQL Injection attack. That is why SQLMAP returns a 403 HTTP-status-code response as shown below:

```
[19:25:24] [WARNING] HTTP error codes detected during run:
403 (Forbidden) - 75 times
```

SQL Injection blocked by ModSecurity

- ModSecurity is much beneficial in terms of protecting your websites from generic attacks such as XSS, SQL injection, etc. But being a technology, it has certain potential drawbacks, which the administrators should be aware of.
- In certain cases, ModSecurity might block legitimate web traffic (false positives), and it might fail to address Content Management System (CMS) specific vulnerabilities which should be mitigated by updating or patching the CMS.
- The authoritative administrators are expected to keep an eye on what is being blocked and add exceptions to the rules accordingly to prevent false positives.

16. Task 5: Test Your Understanding

Complete the questions listed below. At the end of this exercise, attempt to analyze the whole case and answer the *Thought Question(s)* on your own before displaying the answer(s). Note, the latter are not graded.

16.1 Flag Questions:

1. What is the Linux command to restart Apache service ?

   (a) apachectl -M | grep security
   (b) sudo a2enmod security2
   (c) sudo systemctl restart apache2
   (d) sudo apt restart apache2

   - **Correct Answer:** sudo systemctl restart apache2
   - **Answer Description:** To find this, check Task 4:Installation of ModSecurity with Apache on Ubuntu 20.04.3
   - **Difficulty:** easy
   - **Total Points:** 2
   - **Retries:** 2

2. By default, the configuration file of ModSecurity is located at ?

   (a) etc/modsecurity/modsecurity.conf-recommended
   (b) /usr/share/modsecurity-crs
   (c) /etc/modsecurity/*.conf
   (d) etc/modsecurity/modsecurity.conf

   - **Correct Answer:** etc/modsecurity/modsecurity.conf-recommended
   - **Answer Description:** To find this, check Task 5: Post-installation configuration of ModSecurity
   - **Difficulty:** easy
   - **Total Points:** 2
   - **Retries:** 2

3. What is the Linux command to remove the default Core Rule Set (CRS) ?

   (a) sudo apt update
   (b) sudo apt install git
   (c) cd /usr/share/modsecurity-crs
   (d) sudo rm -rf /usr/share/modsecurity-crs

   - **Correct Answer:** sudo rm -rf /usr/share/modsecurity-crs

- **Answer Description:** To find this, Go to Task 6: Adding OWASP ModSecurity rules
- **Difficulty:** easy
- **Total Points:** 2
- **Retries:** 2

4. What is the command to test the Apache configuration after configuring OWASP ModSecurity Core Rule Set?

   (a) sudo apache2ctl -t
   (b) apache2ctl -t
   (c) systemctl restart apache2
   (d) systemctl test apache2

   - **Correct Answer:** sudo apache2ctl -t
   - **Answer Description:** To find this, Go to Task 6: Adding OWASP ModSecurity rules
   - **Difficulty:** easy
   - **Total Points:** 2
   - **Retries:** 2

5. What is the HTTP status code response you got when the SQL Injection attack was blocked?

   (a) 503
   (b) 404
   (c) 403
   (d) 500

   - **Correct Answer:** 403
   - **Answer Description:** To find this, Go to Task 8:Testing ModSecurity
   - **Difficulty:** Easy
   - **Total Points:** 2
   - **Retries:** 2

16.2 Thought Questions:

1. why is a Web Application Firewall (WAF) important for the security of web applications?

   - **Correct Answer:** A web application firewall (WAF) is important for web applications because it provides an additional layer of security against a wide range of attacks. WAFs can protect web applications against both known and unknown vulnerabilities by detecting and blocking malicious traffic before it reaches the application. A WAF can provide protection against common web attacks such as cross-site scripting (XSS), SQL injection, and cross-site request forgery (CSRF). WAFs can help organizations comply with security standards such as OWASP, PCI DSS, and HIPAA, by providing protection

against known vulnerabilities and threats. WAFs can be customized to match the specific security needs of an organization, allowing for fine-tuned protection against specific threats. WAFs can provide real-time monitoring and reporting of suspicious activity, enabling organizations to quickly respond to potential threats and minimize damage. In conclusion, a WAF is an essential tool for protecting web applications against the ever-evolving threat landscape and ensuring that sensitive data remains secure.

17. References

https://beaglesecurity.com/blog/article/modsecurity-apache-installation-guide.html