# Securing an Apache Web Server

## Project Overview

This project focuses on securing an Apache web server on an Ubuntu system. The key security measures implemented include updating the server, configuring firewalls, enabling secure communication, and restricting access to critical resources.

## System Information

- Operating System: Ubuntu (running in VirtualBox)

- Web Server: Apache

- Objective: Strengthen the security of the Apache web server against common vulnerabilities.

## Steps Performed

### 1. Updating System Packages

Before making any changes, the system was updated to ensure the latest security patches were applied.

Command Used:

sudo apt update && sudo apt upgrade -y

### 2. Installing Apache Web Server

Apache was installed and started to serve web content.

Command Used:

sudo apt install apache2 -y  
sudo systemctl enable apache2  
sudo systemctl start apache2

### 3. Verifying Apache Installation

After installation, the Apache server was verified by accessing `http://127.0.0.1` in a web browser.

Screenshot: 127.0.0.1 works.png

### 4. Configuring the Firewall (UFW)

To allow traffic only on required ports, the firewall was configured to permit HTTP (port 80) and HTTPS (port 443) traffic.

Commands Used:

sudo ufw allow 'Apache Full'  
sudo ufw enable  
sudo ufw status

### 5. Enabling Secure Communication with SSL/TLS

To enable HTTPS, an SSL certificate was configured using OpenSSL or Let's Encrypt.

Commands Used:

sudo apt install certbot python3-certbot-apache -y  
sudo certbot --apache

### 6. Restricting Directory Access

Apache configuration files were modified to enhance security by disabling directory listing and restricting access to sensitive files.

Commands Used:

sudo nano /etc/apache2/apache2.conf

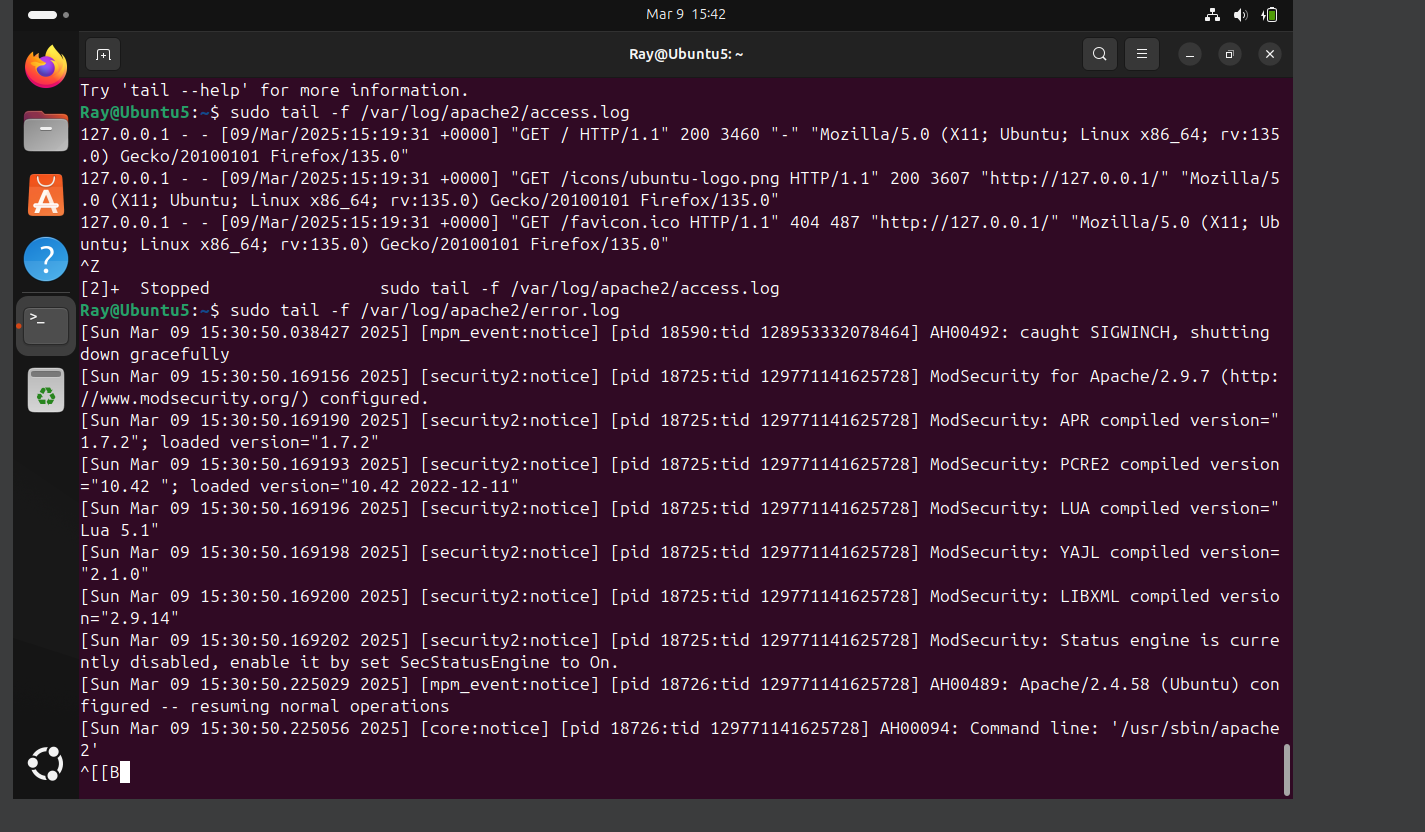
<Directory /var/www/>  
 Options -Indexes  
 </Directory>

### 7. Disabling Unnecessary Modules

To reduce the attack surface, unused Apache modules were disabled.

Command Used:

sudo a2dismod status  
sudo systemctl restart apache2

1. **Monitr apache log files: **

## Conclusion

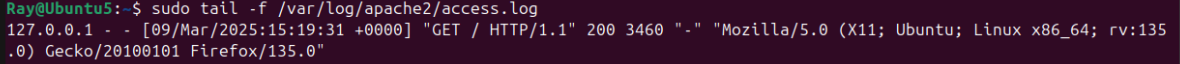
By implementing these security measures, the Apache web server was hardened against common vulnerabilities. The firewall was configured, SSL was enabled, directory access was restricted, unnecessary modules were disabled, and authentication was enforced to secure sensitive resources.

You are viewing Apache log files (access.log and error.log) using the tail -f command, which allows you to monitor logs in real time.

### How to Understand These Logs:

**Access Log (**/var/log/apache2/access.log)

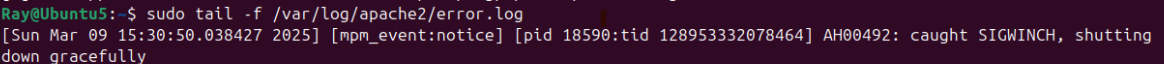
* 1. Records requests to your server.



* + 1. 127.0.0.1: Client IP (localhost in this case)
    2. 09/Mar/2025:15:19:31 +0000: Timestamp
    3. "GET / HTTP/1.1": Request type and URL
    4. 200: HTTP response code (200 = OK, 404 = Not Found, etc.)
    5. 3460: Response size
    6. "Mozilla/5.0 ...": User agent (browser info)

**Error Log (**/var/log/apache2/error.log**)**

* 1. Logs server errors and security notices.



* + 1. [Sun Mar 09 15:30:50.038427 2025]: Timestamp
    2. [mpm\_event:notice]: Module reporting the message
    3. AH00492: caught SIGWINCH, shutting down gracefully: Message (here, the server is shutting down)

### How to Investigate Issues:

* **Check for HTTP 4xx or 5xx errors** in access.log (e.g., 404 Not Found, 500 Internal Server Error).
* **Look for "error", "critical", or "failed" messages** in error.log.

**Monitor live logs** using: sudo tail -f /var/log/apache2/access.log /var/log/apache2/error.log

**Search for specific errors**:

sudo grep "error" /var/log/apache2/error.log