**📖 Introduction**

The SafeLine Web Application Firewall (WAF) Project was developed as part of my cybersecurity home lab to strengthen my practical skills in network security and application defense mechanisms. The primary objective of this project was to simulate a real-world environment where web applications are exposed to common threats such as SQL Injection, HTTP Flood attacks, and unauthorized access attempts.

By deploying SafeLine WAF on an Ubuntu server and integrating it with an Apache web application, I was able to configure security rules, generate SSL certificates, and test the system against simulated attacks launched from a Kali Linux machine. The project provided valuable hands-on experience in configuring WAF policies, managing web traffic, and validating security protections, which are essential skills for roles in SOC operations, network security, and cybersecurity engineering.

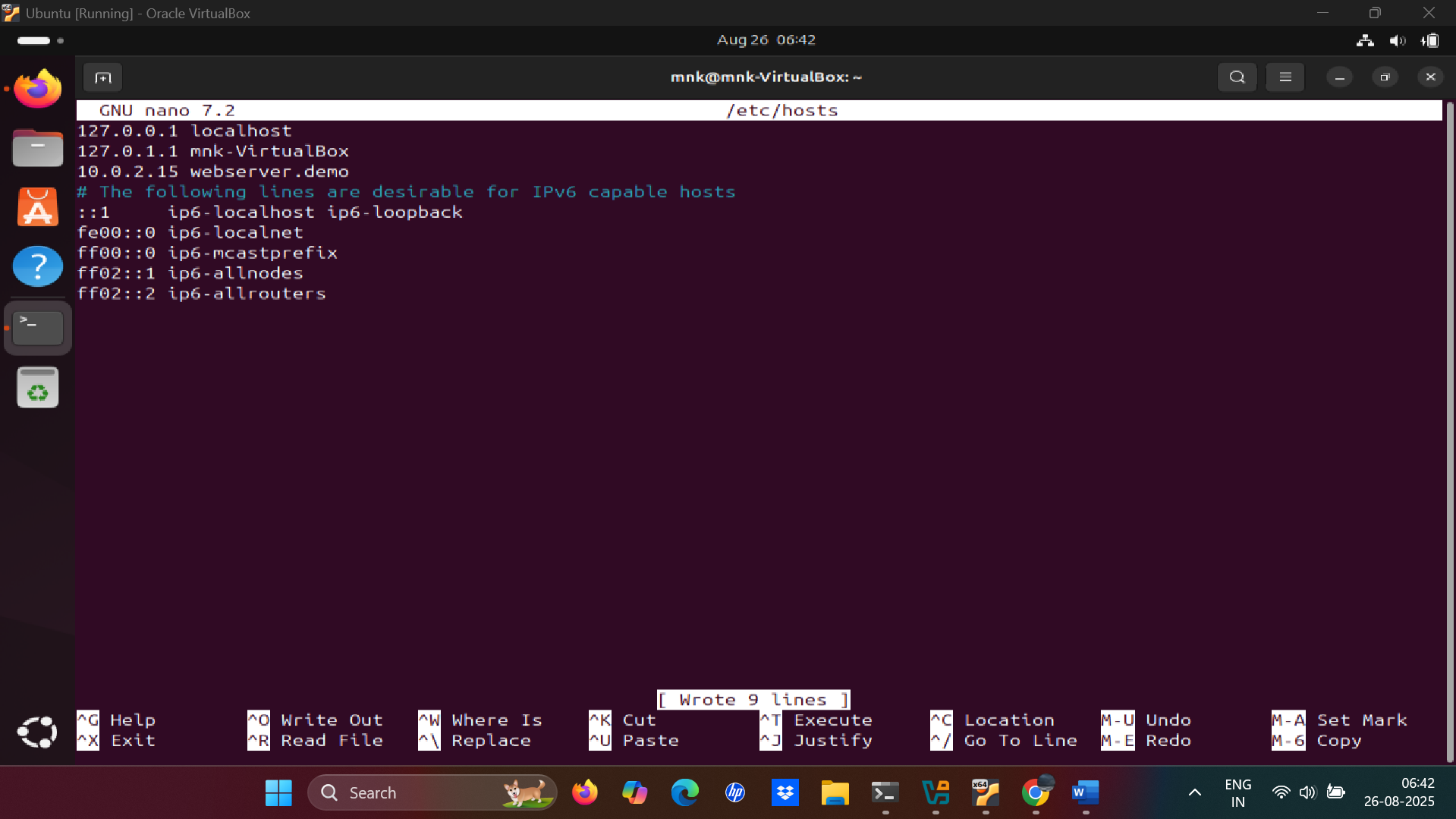
This project not only demonstrates my ability to deploy and secure applications but also reflects my interest in building a strong foundation in offensive and defensive security practices.

**🌐 DNS & Host Configuration**

To simulate a real-world environment with domain-based access, I configured the DNS resolution locally within the Ubuntu server using the /etc/hosts file. This allowed the web application and WAF to be accessed via custom domain names rather than IP addresses.

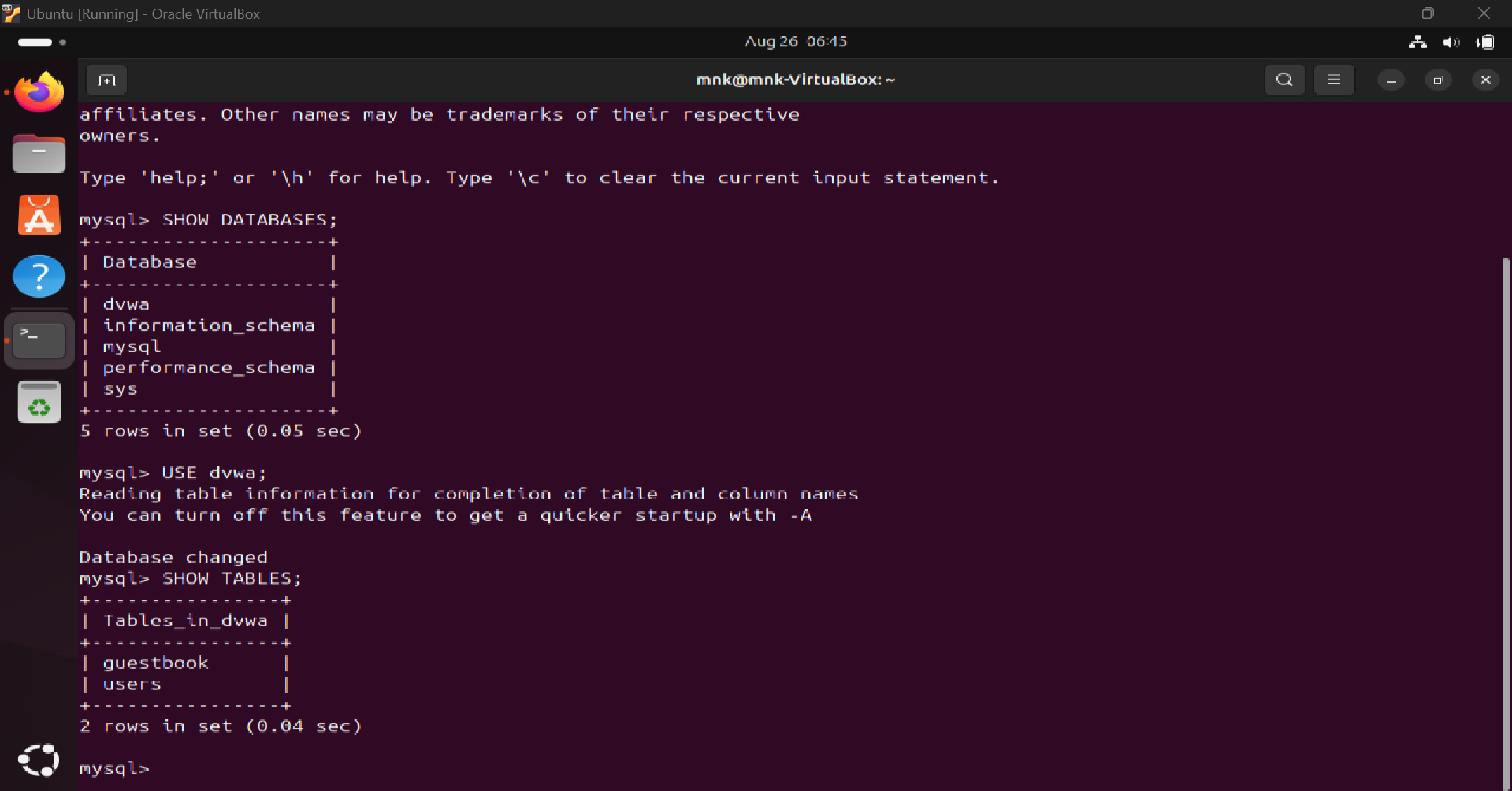
Steps:

1. Open the hosts file with root privileges:
2. sudo nano /etc/hosts
3. Added the Domain Name



**🗄️ Database Configuration (MySQL – DVWA Setup)**

To provide a realistic vulnerable web application for testing the WAF, I installed and configured DVWA (Damn Vulnerable Web Application) on the Ubuntu web server.

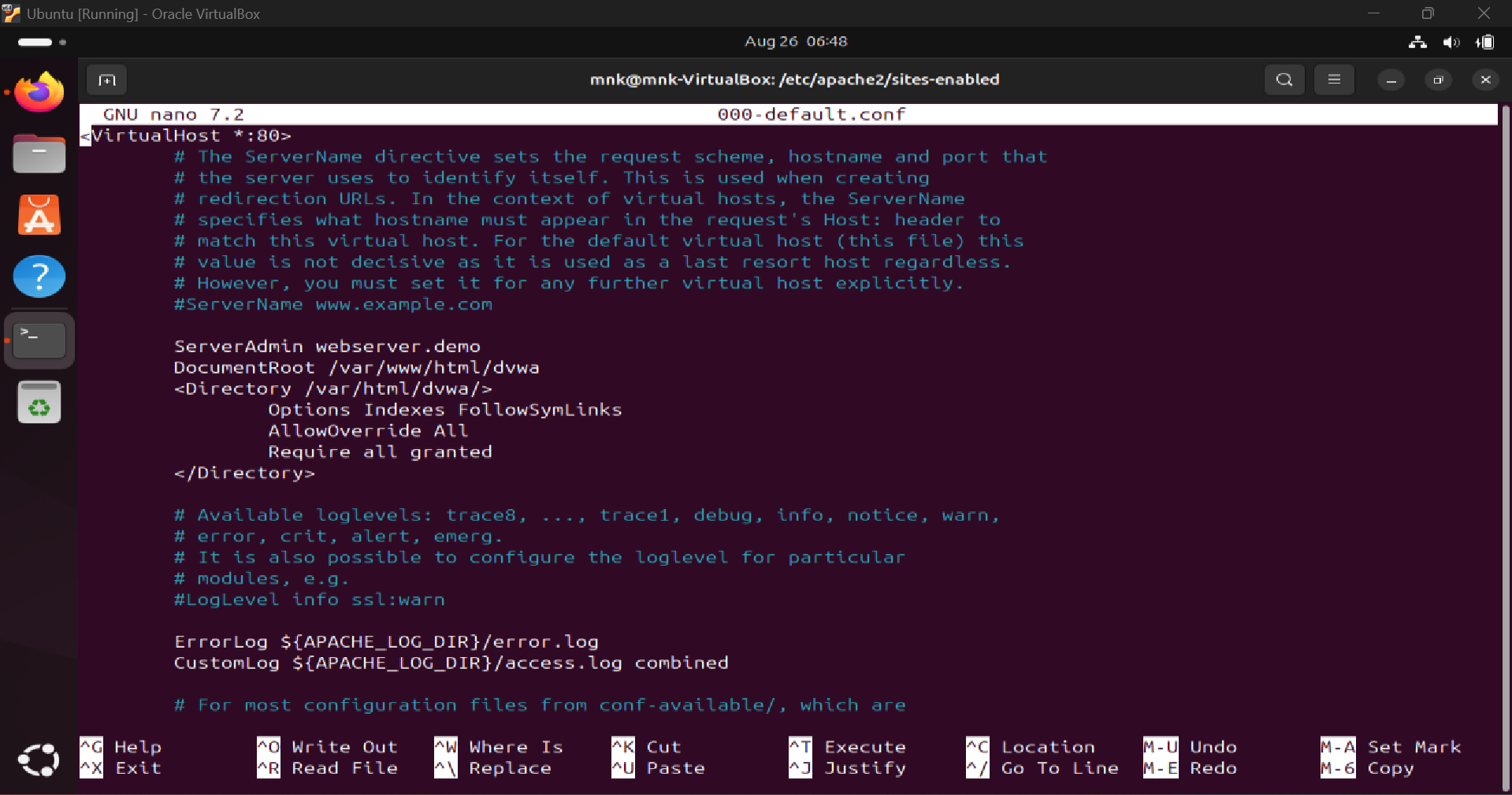


Steps:

1. Install MySQL Server (if not already installed):
2. sudo apt update
3. sudo apt install mysql-server -y
4. Access MySQL Shell:
5. sudo mysql -u root -p
6. Create DVWA Database:
7. CREATE DATABASE dvwa;
8. Create a DVWA User:
9. CREATE USER 'dvwauser'@'localhost' IDENTIFIED BY 'dvwapass';
10. Grant Privileges to DVWA User:
11. GRANT ALL PRIVILEGES ON dvwa.\* TO 'dvwauser'@'localhost';
12. FLUSH PRIVILEGES;
13. Exit MySQL:
14. EXIT;
15. Configure DVWA Application:
    * Open DVWA’s configuration file (usually config.inc.php).
    * Update the database settings with the credentials:
    * $\_DVWA[ 'db\_user' ] = 'dvwauser';
    * $\_DVWA[ 'db\_password' ] = 'dvwapass';
    * $\_DVWA[ 'db\_database' ] = 'dvwa';

**⚙️ Apache Web Server Configuration**

To serve the DVWA web application directly on the Ubuntu host and integrate it with the SafeLine WAF, I modified the default Apache2 configuration.



Steps:

1. Open the default Apache configuration file:
2. sudo nano /etc/apache2/sites-enabled/000-default.conf
3. Update the DocumentRoot to point to the DVWA web directory:

DocumentRoot /var/www/html/dvwa

<Directory /var/www/html/dvwa>

Options Indexes FollowSymLinks

AllowOverride All

Require all granted

</Directory>

1. Restart Apache to apply changes:
2. sudo systemctl restart apache2

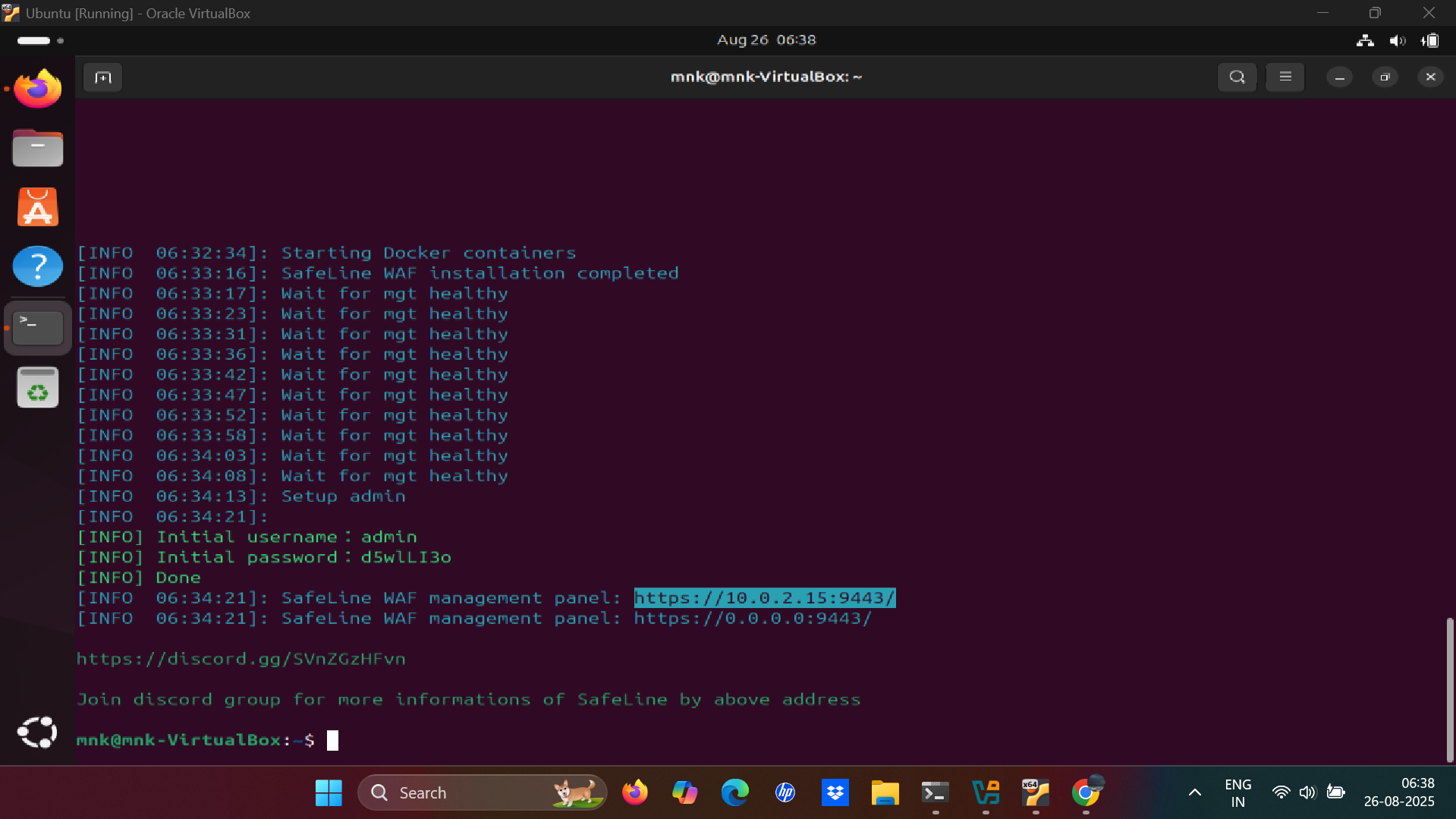
A screenshot of a computer

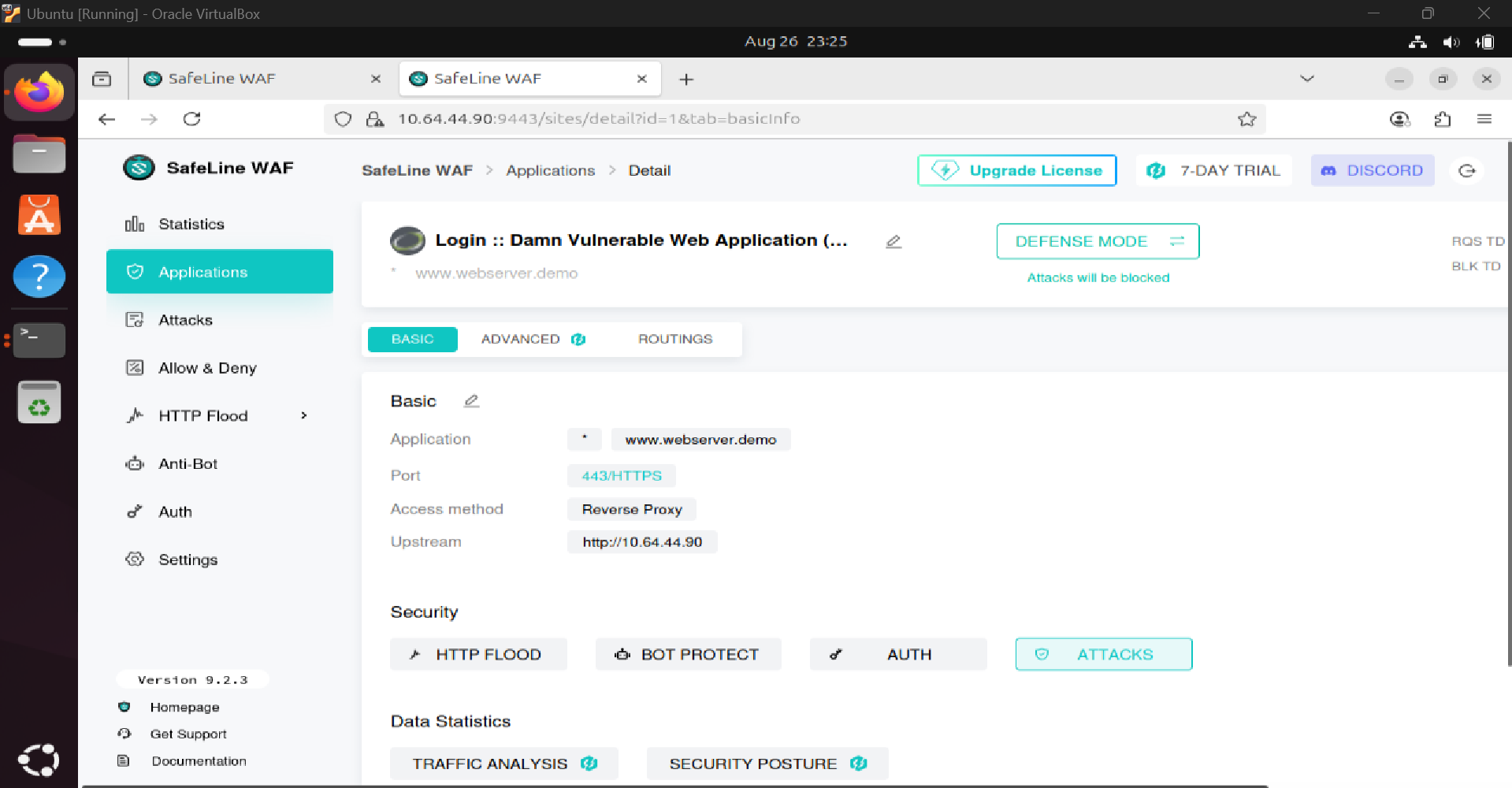
AI-generated content may be incorrect.

* DVWA is now accessible directly on the local host via:
* <http://webserver.demo>

**🛡️ SafeLine WAF Installation**

After setting up the web application and database, I installed SafeLine Web Application Firewall (WAF) on the Ubuntu server to protect DVWA from web-based attacks.





Steps:

1. **Download SafeLine WAF**
   * Download the WAF installation package or Docker image from the official source.
   * Ensure the server meets the minimum requirements for SafeLine.
2. **Install via Docker Compose (preferred for lab setup):**
3. **sudo docker compose up -d**
   * This command sets up the SafeLine WAF container in the background.
4. **Access the WAF Dashboard**
   * Open a browser and navigate to the WAF portal
   * Log in using the admin credentials created during installation.
5. **Configure Initial Settings**
   * Set default security policies.
   * Enable logging and alert notifications.
   * Verify that the WAF is active and monitoring incoming requests.
6. **Integrate with DVWA Application**
   * Add DVWA as a protected application in SafeLine WAF.
   * Assign rules for SQL Injection prevention, HTTP Flood protection, and authentication checks.

**🔒 SSL/TLS Configuration via SafeLine WAF**

To secure the DVWA application and ensure encrypted traffic passes through the WAF, I configured **SSL/TLS settings directly in SafeLine WAF**.

**Steps:**

1. **Access WAF Dashboard**
   * Open the SafeLine WAF admin panel in a browser:
2. **Navigate to SSL/TLS Settings**
   * Go to the **“Certificates” or “SSL Settings”** section in the WAF dashboard.
3. **Create/Upload Certificate**
   * Upload the existing certificate and private key created for DVWA.

A screenshot of a computer program

AI-generated content may be incorrect.

1. **Assign Certificate to DVWA Application**
   * Select the DVWA application under WAF-protected apps.
   * Apply the SSL certificate to ensure **HTTPS traffic is inspected and filtered**.
2. **Enable HTTPS Enforcement**
   * Force all traffic to use **HTTPS**, preventing unencrypted access.

**🏁 Project Conclusion**

The SafeLine Web Application Firewall (WAF) Home Lab Project successfully demonstrates the deployment, configuration, and testing of a web application firewall in a controlled lab environment. By integrating DVWA with Apache, MySQL, and SSL/TLS, and protecting it through SafeLine WAF, I created a realistic environment to simulate and mitigate common web-based attacks such as SQL Injection, HTTP Flood, and unauthorized access attempts.

**Key outcomes of the project include:**

* Hands-on experience in network and web application security.
* Practical knowledge of WAF deployment, rule configuration, and SSL integration.
* Ability to design, configure, and monitor a secure web application in a production-like environment.
* Strengthened understanding of attack vectors and defense strategies, preparing for roles in SOC operations, cybersecurity engineering, and penetration testing.

This project not only highlights my technical skills but also reflects my ability to implement end-to-end security solutions and simulate real-world cyber threats, making it a strong addition to my cybersecurity portfolio.