Web Application Firewall Home Lab using SafeLine WAF

By Royden Rebello (The Social Dork)

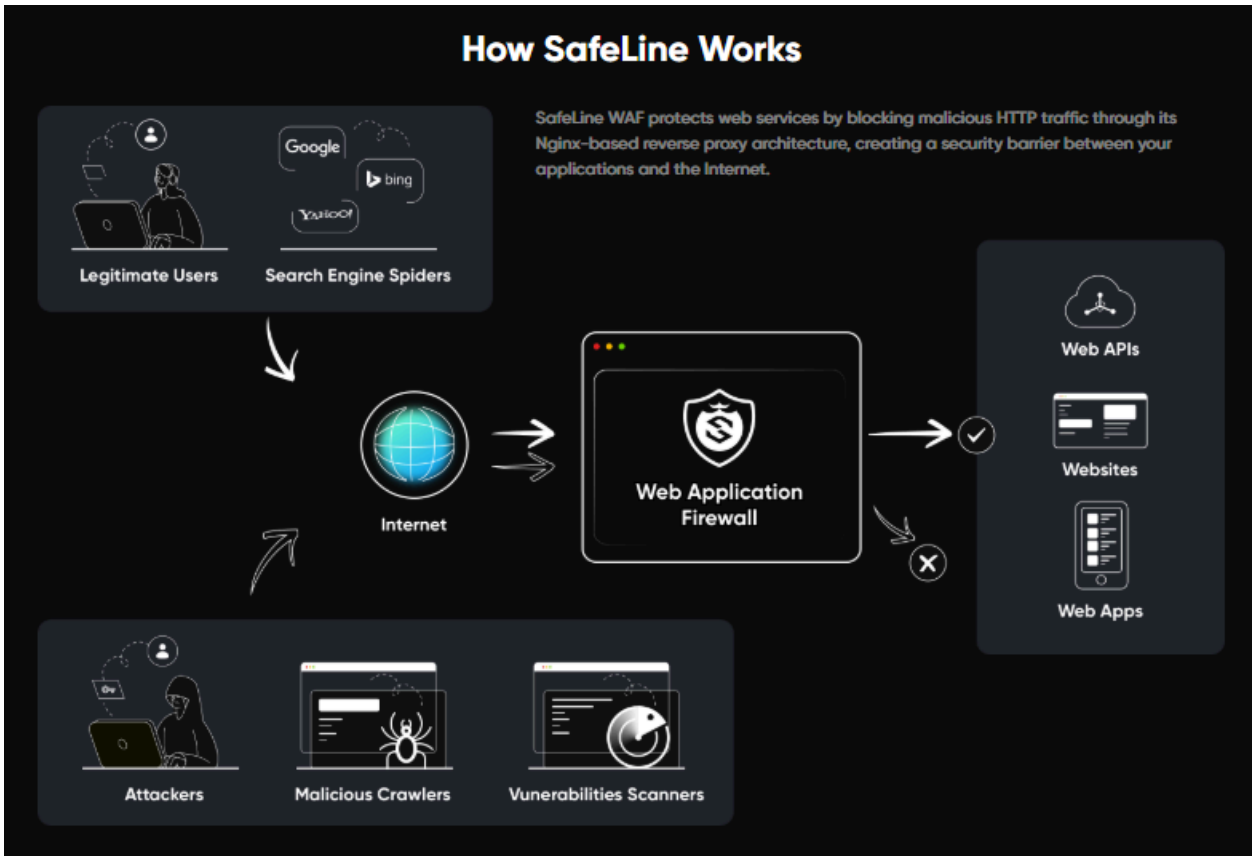

Table of Contents

Table of Contents **2**
1. Introduction **3**
2. Prerequisites **3**
3. Lab Environment Setup **3**
3.1. Download and Install VirtualBox 3
3.2. Create the Kali Linux Virtual Machine 4
3.3. Create the Ubuntu Server Virtual Machine 4
3.4. Enable Bridged Networking 5
3.5. Install Guest Additions (Optional but Recommended) 5
4. Ubuntu Server Configuration **6**
4.1. Initial System Updates and Installations 6
4.2. Installing and Configuring LAMP Stack 7
4.3. Installing and Configuring Damn Vulnerable Web App (DVWA) 7
4.4. Changing the DVWA Listening Port to 8080 8
4.5. Adding Custom Values to the DVWA Database 10
5. DNS Resolution Setup **11**
5.1. Using /etc/hosts for Local Resolution 11
5.2. (Optional) Installing and Configuring BIND DNS Server 11
6. Creating a Self-Signed SSL Certificate **13**
7. Installing and Configuring SafeLine WAF **14**
7.1. Automatic Deployment 14
7.2. Importing the Self-Signed Certificate 14
7.3. Onboarding the DVWA Application 15
8. Demonstrating SQL Injection from Kali Linux **15**
8.1. Launching the Attack 15
8.2. Observing SafeLine WAF Protection 16
9. SafeLine WAF Advanced Configurations **16**
9.1. HTTP Flood Defense 16
9.2. Authentication Sign-In 16
9.3. Custom Deny Rules (Blocking Kali IP) 17
10. Conclusion and Further Exploration **17**
Lab Handbook: Final Notes **18**

1. Introduction

In this lab, we will build a complete cybersecurity home lab using VirtualBox, Kali Linux, Ubuntu, and SafeLine WAF. The goal is to:

- Set up a vulnerable web application (DVWA) on Ubuntu.
- Demonstrate how to perform a basic SQL injection attack from Kali Linux.
- Show how SafeLine WAF protects against such attacks.
- Explore additional SafeLine WAF security measures (HTTP flood defense, custom deny rules, etc

If you get any errors during your home lab setup while following this guide, please refer to the internet to help get solutions for your specific issue.

Refer to the YouTube video for any guidance - https://youtu.be/N0dEC1nuWCQ

2. Prerequisites

- **Host Machine** with sufficient RAM (at least 8 GB recommended) and disk space (at least 50 GB free).
- **Internet Connection** for downloading software and updates.
  Basic Linux Command Line Knowledge (to install packages, edit configuration files, etc.).

3. Lab Environment Setup

3.1. Download and Install VirtualBox

1. **Download VirtualBox**
    - Go to the official VirtualBox download page: https://www.virtualbox.org/wiki/Downloads
    - Select the version for your host OS (Windows, macOS, Linux).
2. **Install VirtualBox**
    - Run the downloaded installer and follow the on-screen instructions.
    - Optionally, install the VirtualBox Extension Pack for additional features (USB 2.0/3.0, etc.).

3.2. Create the Kali Linux Virtual Machine

1. **Download Kali Linux**
    - Official Kali Linux download site: https://www.kali.org/get-kali
    - Choose the ISO file for your architecture (64-bit is most common).

2. **Create a New VM in VirtualBox**
    - **Name**: `KaliLinux`
    - **Type**: Linux
    - **Version**: Debian (64-bit)
    - **Memory**: At least 2 GB (2048 MB)
    - **Hard Disk**: Create a virtual hard disk (dynamically allocated), ~20 GB recommended.

3. **Attach the Kali ISO & Install**
    - Go to **Settings > Storage**, attach the Kali ISO under the “IDE” controller.
    - Start the VM, follow the Kali Linux installation steps (graphical install recommended).

- Set up a username and password you will remember (e.g., `kali` / `kali`).

3.3. Create the Ubuntu Server Virtual Machine

1. **Download Ubuntu Server**
   - Official Ubuntu download site: https://ubuntu.com/download/server
   - Choose the latest LTS release (e.g., 22.04 LTS).
2. **Create a New VM in VirtualBox**
   - **Name**: `UbuntuServer`
   - **Type**: Linux
   - **Version**: Ubuntu (64-bit)
   - **Memory**: At least 2 GB (2048 MB)
   - **Hard Disk**: ~20 GB (dynamically allocated).

3. **Attach the Ubuntu ISO & Install**
   - Under **Settings > Storage**, attach the Ubuntu ISO file.
   - Start the VM, follow the Ubuntu Server installation instructions.
   - Create a username/password (e.g., `ubuntu` / `ubuntu`).
   - **Note**: You may optionally install the OpenSSH server during installation if you want to access the VM remotely.

3.4. Enable Bridged Networking

To have the VMs appear on the same network as your host (and each other):

1. **Open VirtualBox** > select your VM (e.g., `UbuntuServer`) > **Settings** > **Network**.
2. **Adapter 1**: Choose **Bridged Adapter** from the “Attached to” drop-down.
3. Select your host’s network interface (Ethernet or Wi-Fi).
4. Click **OK**.
5. Grab the IP address of the machine after you install net-tools in section 4 by running the command “ifconfig” from the terminal.

Repeat for the **Kali Linux** VM as well.

3.5. Install Guest Additions (Optional but Recommended)

1. With the VM running, go to **Devices > Insert Guest Additions CD Image** in the VirtualBox window.

2. Follow on-screen instructions, or mount the CD and install manually using the commands below:

```
sudo apt-get update

sudo apt-get install build-essential dkms linux-headers-$(uname -r)

sudo mount /dev/cdrom /media/cdrom

sudo /media/cdrom/VBoxLinuxAdditions.run
```

3. Restart the VM.

   Note: Guest Additions can help with screen resizing, shared clipboard, and shared folders.

4. Ubuntu Server Configuration

4.1. Initial System Updates and Installations

Update the Package List and Upgrade:

```
sudo apt-get update
```

```
sudo apt-get upgrade -y
```

Install Net-Tools (for `ifconfig` and other network utilities):

```
sudo apt-get install -y net-tools
```

Install OpenSSL:

```
sudo apt-get install -y openssl
```

4.2. Installing and Configuring LAMP Stack

Install Apache2, PHP, and MySQL:

```
sudo apt-get install -y apache2 php php-mysql mysql-server
```

Secure the MySQL Installation (optional but recommended):

```
sudo mysql_secure_installation
```

Follow the prompts (set a root password, remove anonymous users, disable remote root login, etc.).

4.3. Installing and Configuring Damn Vulnerable Web App (DVWA)

Clone DVWA (or download):

```
cd /var/www/html

sudo git clone https://github.com/digininja/DVWA.git
```

If `git` is not installed, install it first:

```
sudo apt-get install -y git
```

Set File Permissions:

```
sudo chown -R www-data:www-data DVWA

sudo chmod -R 755 DVWA
```

Configure DVWA Database:

DVWA has a `config` file at `DVWA/config/config.inc.php`. Update it if necessary:

```
$DBMS = 'MySQL';

$db   = 'dvwa';

$user = 'dvwa_user';

$pass = 'p@ssw0rd';

$host = 'localhost';
```

Create a new database and user in MySQL:

```
sudo mysql -u root -p

CREATE DATABASE dvwa;

CREATE USER 'dvwa_user'@'localhost' IDENTIFIED BY 'p@ssw0rd';

GRANT ALL ON dvwa.* TO 'dvwa_user'@'localhost';

FLUSH PRIVILEGES;

exit;
```

Initialize DVWA:
Navigate to `http://<Ubuntu IP>/DVWA/setup.php` in your browser.
Click **Create/Reset Database**.

4.4. Changing the DVWA Listening Port to 8080

By default, Apache listens on port 80. To change it to 8080:

Edit the Apache Configuration:

```
sudo nano /etc/apache2/ports.conf
```

Change:

```
Listen 80
```

to:

```
Listen 8080
```

Update the default Virtual Host (optional if you want the default site on 8080):

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

1. Change `<VirtualHost *:80>` to `<VirtualHost *:8080>`.

Restart Apache:

```
sudo systemctl restart apache2
```

Now DVWA is accessible at `http://<Ubuntu IP>:8080/DVWA`.

4.5. Adding Custom Values to the DVWA Database

You can add custom data to the DVWA database to demonstrate SQL injection:

Log in to MySQL:

```
sudo mysql -u root -p
```

```
USE dvwa;
```

Add a sample table/entries:

```
CREATE TABLE test_users (
    id INT NOT NULL AUTO_INCREMENT,
    username VARCHAR(50) NOT NULL,
    password VARCHAR(50) NOT NULL,
    PRIMARY KEY (id)
```

```
);

INSERT INTO test_users (username, password) VALUES

('alice', 'alice123'),

('bob', 'bob123'),

('admin', 'admin123');
exit;
```

These entries give you something to target with SQL injection experiments.

5. DNS Resolution Setup

5.1. Using `/etc/hosts` for Local Resolution

Edit `/etc/hosts` on Both Ubuntu and Kali:

```
sudo nano /etc/hosts
```

Add a Line (replace `<Ubuntu IP>` with the actual IP address):

```
<Ubuntu IP>    dvwa.local
```

Now you can access the DVWA app at `http://dvwa.local:8080/DVWA` in your browser (on Kali).

5.2. (Optional) Installing and Configuring BIND DNS Server

If you prefer a dedicated DNS server:

Install BIND9:

```
sudo apt-get install -y bind9
```

Configure a Zone File:

Edit `/etc/bind/named.conf.local`:

```
zone "dvwa.local" {
    type master;
    file "/etc/bind/zones/db.dvwa.local";
};
```

Create the zone directory if not present:

```
sudo mkdir -p /etc/bind/zones
```

Create the zone file `/etc/bind/zones/db.dvwa.local`:

```
;
; BIND data file for dvwa.local
;
$TTL    604800
@       IN      SOA     ns.dvwa.local. admin.dvwa.local. (
                          1         ; Serial
                          604800    ; Refresh
```

```
                        86400      ; Retry
                        2419200    ; Expire
                        604800 )   ; Negative Cache TTL
;
@       IN      NS      ns.dvwa.local.
ns      IN      A       <Ubuntu IP>
www     IN      A       <Ubuntu IP>
```

-

Restart BIND:

```
sudo systemctl restart bind9
```

Configure Kali to Use the Ubuntu IP for DNS:

Edit `/etc/resolv.conf` (or `NetworkManager` settings):

```
nameserver <Ubuntu IP>
```

Test with:

```
dig www.dvwa.local
```

6. Creating a Self-Signed SSL Certificate

Refer to the YouTube video for any guidance - https://youtu.be/N0dEC1nuWCQ

On Ubuntu, create a self-signed certificate to use for HTTPS:

```
sudo mkdir /etc/ssl/dvwa

sudo openssl req -x509 -nodes -days 365 -newkey rsa:2048 \

 -keyout /etc/ssl/dvwa/dvwa.key \

 -out /etc/ssl/dvwa/dvwa.crt \
```

7. Installing and Configuring SafeLine WAF

Refer to the YouTube video for any guidance - https://youtu.be/N0dEC1nuWCQ

7.1. Automatic Deployment

Head to the following website https://ly.safepoint.cloud/zfYZD3l and select “Install”

SafeLine WAF typically provides an **automatic install script** or a manual install method, follow the steps below for the automatic install:

1. **Use the following command on your Ubuntu CLI -** `bash -c "$(curl -fsSLk https://waf.chaitin.com/release/latest/manager.sh)" -- --en`
2. **Follow On-Screen Prompts** to complete installation.
3. You will be provided with an admin password and username as well as the URL to access Safeline, typically on port 9443

4. Once you log in, it is highly recommended to upgrade to a PRO license, you can choose to do a 7 day trial from https://ly.safepoint.cloud/zfYZD3l which amounts to $1. **You can also choose for a longer term subscription, use the below code for 5% off - ZFGYUXVXABSUH7KTMQG4FG4B**

7.2. Importing the Self-Signed Certificate

In the SafeLine WAF interface (web UI or config files), import the `.crt` and `.key` files you created under the SSL certificate section.

- **Certificate File**: `/etc/ssl/dvwa/dvwa.crt`
- **Key File**: `/etc/ssl/dvwa/dvwa.key`

Follow the SafeLine documentation on how to manage certificates (it could be through a GUI or CLI).

7.3. Onboarding the DVWA Application

Use the SafeLine WAF management console or interface to **add a new application**:

1. **Add the DNS name -** www.dvwa.local or whatever Domain name you have selected
2. **Backend URL (Reverse Proxy)** : `http://<UbuntuIP>:8080`
3. Delete port 80 and only leave port 443
4. **Virtual Host** (for WAF): e.g., `dvwa-waf.local` or re-use the same domain.
5. **Attach the SSL Certificate** (if you want the WAF to serve HTTPS).

8. Demonstrating SQL Injection from Kali Linux

8.1. Launching the Attack

1. **Open Kali Linux** and browse to the DVWA site (This will now be redirected to https):
    - `http://dvwa.local/`
2. **Log In** to DVWA (default credentials in DVWA are `admin` / `password` unless changed).
3. **Set DVWA Security Level** to low (in the DVWA Security tab).
4. **Go to SQL Injection Section** and try typical SQL injection strings (e.g., `admin' OR '1'='1`).

8.2. Observing SafeLine WAF Protection

SafeLine WAF should detect and block malicious injection attempts. Look for **WAF logs** to see blocked SQL injection attempts. You can also see error or block pages if the WAF is in blocking mode.

9. SafeLine WAF Advanced Configurations

Refer to the YouTube video for any guidance - https://youtu.be/N0dEC1nuWCQ

9.1. HTTP Flood Defense

1. **Enable** HTTP Flood/DoS settings in SafeLine WAF.

2. Configure **Thresholds** (requests per second) and **Penalty** or **Ban** duration.

3. **Test** by sending multiple requests from Kali (e.g., using `ab`, `siege`, or other load tools).

9.2. Authentication Sign-In

SafeLine supports an **auth gateway**:

1. **Enable** Auth Sign-In in the WAF policy for DVWA.

2. **Configure** username/password or integrate with an external auth provider.

3. Attempt to access DVWA from Kali; the WAF should prompt for credentials before passing traffic to the server.

9.3. Custom Deny Rules (Blocking Kali IP)

1. **Identify** the IP of your Kali VM (e.g., `192.168.1.50`).
2. **Add a Deny Rule** in SafeLine WAF:
    - Match Source IP: `192.168.1.50`
    - Action: `Block` or `Deny`.
3. **Test** from Kali: You should receive a blocked response.

10. Conclusion and Further Exploration

- You have now set up a **complete homelab** environment with a vulnerable web application, a **web application firewall (WAF)**, and the tools to attack and defend.

- You've explored **SQL injection**, **WAF configuration**, **HTTP flood defenses**, **authentication controls**, and **custom blocking**.

- For further exploration:

    - Experiment with **different WAF options**
    - Set up additional **vulnerable applications** (e.g., **OWASP Juice Shop**) for broader testing.
    - Explore **other attacks** (XSS, file inclusion, command injection) and see how SafeLine WAF responds.
    - Integrate this lab with your home SIEM solution, NGFW, IPS/IDS etc for a more in-depth home lab set up to study cybersecurity.

Lab Handbook: Final Notes

- **Troubleshooting**: If something doesn't work, check logs on each layer (Kali, Apache, SafeLine WAF).
- **Security Best Practices**: Keep your homelab isolated from production networks.
- **Maintenance**: Regularly update Ubuntu, Kali, and SafeLine WAF to patch vulnerabilities.

Refer to the YouTube video for any guidance - https://youtu.be/N0dEC1nuWCQ

This guide should give you (and anyone following along) a robust roadmap to set up the entire homelab from scratch, test vulnerabilities, and see how SafeLine WAF protects web applications. Good luck, and enjoy exploring your new cybersecurity lab!