Secure Web Infrastructure Lab with SafeLine WAF & DVWA

Réalisation d'un laboratoire de cybersécurité complet :

Apache, PHP, MariaDB, DVWA, DNS, SSL, Reverse Proxy &

SafeLine WAF

Mohammed Ryad Dermouche

November 21, 2025

Contents

Abstract

This report presents the full deployment and configuration of a secure web laboratory designed to study web attacks and defensive mechanisms. The environment includes a vulnerable web application (DVWA), a complete LAMP stack (Apache, PHP, MariaDB), an internal DNS override, SSL/TLS certificate generation, and the deployment of the SafeLine Web Application Firewall acting as a reverse proxy.

The goal of this lab is to emulate real-world attack scenarios using a Kali Linux attacker machine and an Ubuntu web server, both hosted on VirtualBox. SafeLine WAF is configured to enforce HTTPS, filter malicious traffic, inspect SQL injection attempts, rate-limit HTTP floods, block brute-force attacks, and provide custom rules.

This document details the setup, configurations, troubleshooting steps, and the behaviour of the WAF during real attacks.

Chapitre 1

Introduction

Dans ce rapport, nous présentons la mise en place complète d'un laboratoire professionnel de cybersécurité permettant de comprendre les attaques web courantes et les mécanismes de défense avancés.

Ce projet repose sur :

- Kali Linux en tant que machine attaquante ;
- Ubuntu Server en tant que serveur web vulnérable ;
- VirtualBox comme hyperviseur ;
- Apache2, PHP et MariaDB formant une pile LAMP ;
- DVWA (Damn Vulnerable Web Application) comme application vulnérable ;
- un serveur DNS local simulé via `/etc/hosts` ;
- des certificats SSL/TLS générés manuellement ;
- SafeLine WAF jouant le rôle de reverse proxy sécurisé.

Ce laboratoire reproduit une architecture réaliste, comparable à celle utilisée dans un environnement de production, afin d'étudier de manière pratique :

- les attaques SQLi, XSS, brute force ;
- les attaques par flood HTTP ;
- la sécurisation SSL/TLS ;
- l'efficacité d'un WAF moderne.

Chapitre 2

Architecture du laboratoire

Le laboratoire repose sur deux machines virtuelles connectées via un réseau VirtualBox en mode pont (bridged) :

- **Kali Linux** — simulateur d'attaques ;
- **Ubuntu Server** — hébergement de DVWA et du WAF SafeLine.

Le mode pont permet aux VM de recevoir une adresse IP du routeur local, comme des machines physiques, facilitant ainsi la communication directe.

Le serveur Ubuntu héberge :

- Apache2 sur le port interne 8080 ;
- DVWA configuré pour pointer vers MariaDB ;
- SafeLine WAF accessible via son interface d'administration sur 9443 ;
- le reverse proxy de SafeLine exposé en HTTPS (port 443).

Chapitre 3

Configuration du serveur Ubuntu

3.1 Installation Apache

```
sudo apt update
sudo apt install -y apache2
```

Le port d'écoute interne est modifié pour être 8080 :

```
/etc/apache2/ports.conf
Listen 8080
```

3.2 Installation de PHP

```
sudo apt install -y php php-mysqli php-gd
```

3.3 MariaDB

```
sudo apt install -y mariadb-server
sudo mysql_secure_installation
```

Création de la base DVWA :

```
CREATE DATABASE dvwa;
CREATE USER 'dvwa'@'localhost' IDENTIFIED BY 'p@ssw0rd';
```

```
GRANT ALL PRIVILEGES ON dvwa.* TO 'dvwa'@'localhost';
FLUSH PRIVILEGES;
```

Chapitre 4

Installation de DVWA

Les fichiers DVWA sont placés dans :

`/var/www/html/DVWA`

Configuration dans `config.inc.php` :

```
$_DVWA['db_server'] = '127.0.0.1';
$_DVWA['db_user'] = 'dvwa';
$_DVWA['db_password'] = 'p@ssw0rd';
$_DVWA['db_database'] = 'dvwa';
```

Une fois DVWA accessible via :

`http://IP_UBUNTU:8080/DVWA`

La base est initialisée via “Create / Reset Database”.

Chapitre 5

DNS local (override)

Pour simuler un domaine sans déployer Bind9, nous ajoutons :

`/etc/hosts`

```
192.168.X.X   webserver.lab
```

Même entrée sur Kali.

Cela permet d'accéder à DVWA par :

```
http://webserver.lab:8080
```

Chapitre 6

Génération SSL / TLS

6.1 Clé privée

```
openssl genrsa -out priv.key 4096
```

6.2 CSR

```
openssl req -new -key priv.key -out priv.csr
```

6.3 Certificat auto-signé

```
openssl x509 -req -in priv.csr -signkey priv.key \
-out priv.crt -days 365
```

Les fichiers seront importés dans SafeLine WAF.

Chapitre 7

Installation du WAF SafeLine

Installation automatique :

```
sudo bash -c "$(curl -fsSL https://cdn.jsdelivr.net/gh/safeline-waf/safeline@mai
```

L'interface est accessible sur :

```
https://IP_UBUNTU:9443
```

Après activation de la licence, configuration du WAF via :

- Domaine : `webserver.lab`
- Port exposé : 443 (HTTPS)
- Mode : Reverse Proxy
- Backend : `http://IP_UBUNTU:8080`
- Certificat SSL : priv.crt + priv.key

Chapitre 8

Tests d'attaques et défenses du WAF

8.1 SQL Injection

Exemple depuis Kali :

```
' OR '1'='1
```

Résultat :

- WAF actif : rejet immédiat (HTTP 403)
- WAF désactivé : fuite de la base DVWA

8.2 Brute Force

Le module "Brute Force" de DVWA a permis de valider :

- blocage via détection d'anomalies ;
- journalisation détaillée côté WAF.

8.3 HTTP Flood (DDoS simulé)

Configuration SafeLine :

- 3 requêtes / 10 secondes ;
- blocage 5 minutes.

Kali :

```
while true; do curl https://webserver.lab; done
```

Résultat : bannissement automatique.

8.4 Règles personnalisées

Exemple :

```
If source_ip == 192.168.X.X → block
```

Testé et validé depuis Kali.

Chapitre 9

Analyse des résultats

Après l'ensemble des tests :

- SafeLine bloque efficacement les injections SQL ;
- Les attaques par brute force sont détectées immédiatement ;
- Le reverse proxy force le passage en HTTPS ;
- L'interface fournit des métriques professionnelles (requêtes, visiteurs, IP bloquées) ;
- Le système est stable et reproduit fidèlement une architecture de production.

Chapitre 10

Conclusion

Ce laboratoire constitue une démonstration complète et professionnelle d'une infrastructure web sécurisée. Il combine le déploiement d'un environnement vulnérable, la configuration SSL, le routage via reverse proxy, et la protection avancée via un WAF moderne.

SafeLine WAF a démontré sa capacité à bloquer les attaques applicatives courantes, à fournir un contrôle granulaire, et à offrir une visibilité complète sur le trafic.

Ce projet est directement exploitable pour un portfolio professionnel, un entretien technique, ou une démonstration pratique en cybersécurité.

Références

- DVWA : `https://github.com/digininja/DVWA`
- SafeLine WAF : `https://safeline.cloud`
- OpenSSL documentation
- Apache2 documentation