**Hackaton Ynov / Floa : évaluation des capacités de protection d’un Web Application Firewall**

# Problématique entreprise

Les solutions de Web Application Firewall sont une pièce importante en termes de protection et de détection pour toutes les entreprises amenées à exposer des services à leurs clients ou partenaires par Internet.

Les WAF permettent notamment :

1. De limiter le trafic malveillant à la source via plusieurs méthodes
2. De protéger certaines fonctions vulnérables dans les applications
3. De disposer d’une traçabilité renforcée et unifiée indépendante du serveur web

Les capacités de protection d’un Web Application Firewall sont basées sur des règles de détection.

La tendance sur les acteurs clefs du marché WAF est de plus en plus :

* De proposer les solutions en mode Cloud
* De proposer des jeux de règles gérées

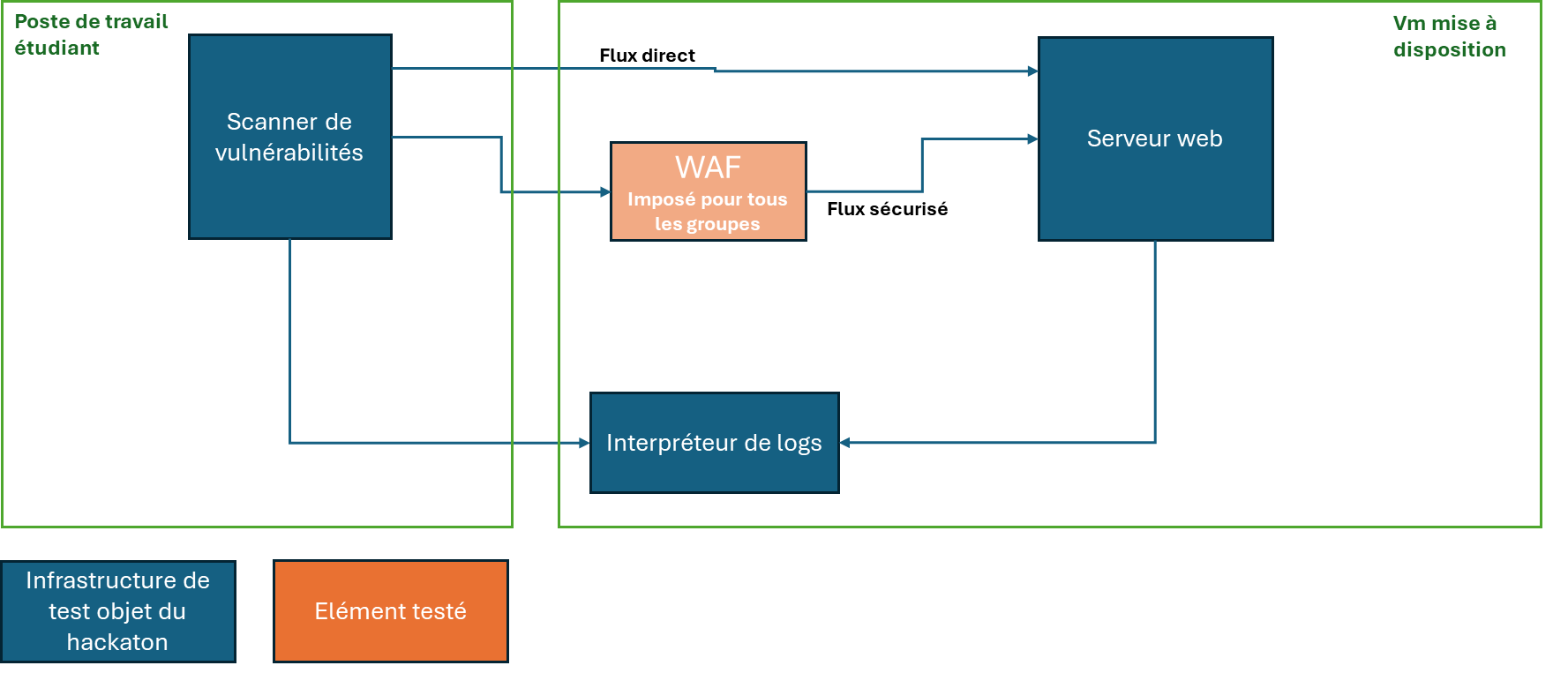
Si cela évite aux clients de gérer eux même un composant potentiellement complexe à administrer, la conséquence est que les règles sont de moins en moins accessibles pour les clients et deviennent des « boites noires »

Cela pose des questions en termes de maitrise par le client d’une part et de capacité également à travailler ses politiques de filtrage ou de protection pour couvrir ses risques réels.

# Objectifs du Hackaton

Proposer une méthode et des outils permettant d’évaluer la capacité d’un Web Application Firewall à bloquer les attaques.

# Architecture



# Hypothèses et indications

* **Le but est de proposer des méthodes d’évaluation des capacités de protection des WAF, pas de proposer une politique WAF performante. Pour comparer les méthodes des différents groupes, la configuration WAF sera imposée et identique pour tous les groupes**

Hypothèse posée de manière à simplifier la mise en œuvre

* Le serveur web fait partie de l’architecture de tests. Il doit être configuré de manière à permettre l’analyse, pas nécessairement à être réellement vulnérable, surtout que le WAF est théoriquement là pour bloquer les attaques avant qu’elle ne parvienne au serveur
* Le test ne vise pas à évaluer le taux de faux positif du WAF
* En termes de méthodes d’attaques, une liberté est laissée sur la constitution du « jeu de test », néanmoins :
  + La phase de reconnaissance dans les attaques semble particulièrement adaptée pour construire les premiers indicateurs car les attaques sont souvent déjà automatisées et pensées pour être jouée en série (logique de scan)
  + Les attaques « massives » (Déni de service ou bruteforce) ne sont pas l’objectif du test. Le WAF fourni n’est pas configuré pour les bloquer.
* Les statistiques peuvent être produites à partir :
  + Des résultats du test côté attaquant
  + Et/OU des requêtes parvenues ou non au serveur

*Il peut être intéressant de comparer les deux visions*

# Eléments à restituer

* Description de la méthode (Description de l’architecture, type de tests réalisés, configuration du serveur, construction des indicateurs)
* Résultat brut des indicateurs
* Interprétation du contexte des résultats (limites et angles morts de l’approche, problématiques potentielles de qualité des données, points à approfondir, …)
* Interprétations des résultats : nature de la protection apportée par le WAF,
* Propositions de pistes d’amélioration de la politique WAF

# Eléments fournis

* 1 machine virtuelle avec IP publique
* 1 nom dns
* 1 configuration Apache Mod Security

# Instructions techniques

## Infrastructure

* L’infrastructure est accessible via Internet, via son IP publique ou son nom DNS
* Elle consiste en une VM Linux en configuration de base par Groupe
* Le ssh est configuré sur le port 22, avec une clef ssh qui vous sera fournie. Néanmoins de la translation d’adresse est réalisée, il sera accessible depuis internet sur le port 220X, cf tableau ci-dessous
* Le serveur web sera à monter pas vos soins sur le port 80. Néanmoins de la translation d’adresse est réalisée, il sera accessible depuis internet sur le port 800X, cf tableau ci-dessous
* Le proxy apache mod security sera à monter pas vos soins sur le port 8080. Néanmoins de la translation d’adresse est réalisée, il sera accessible depuis internet sur le port 808X, cf tableau ci-dessous
* Pour simplifier le hackaton, l’utilisation de SSL n’est pas prévue
* Pour simplifier le hackaton, la dockerisation des différents composants n’est pas requise

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
|  | Groupe 2 | Groupe 3 | Groupe 4 | Groupe 5 | Groupe 6 | Groupe 1 |
| Entrée DNS publique | Hackaton.linkmyaccounts.com | | | | | |
| IP publique entrée | 20.199.27.146 | | | | | |
| IP publique sortie | 20.199.27.63 | | | | | |
| Port accès SSH | 2202 | 2203 | 2204 | 2205 | 2206 | 2207 |
| Port http (80 sur la machine virutelle) | 8002 | 8003 | 8004 | 8005 | 8006 | 8007 |
| Port http proxy (8080 sur la machine virtuelle) | 8082 | 8083 | 8084 | 8085 | 8086 | 8087 |
| Login | Hackaton02 | Hackaton03 | Hackaton04 | Hackaton05 | Hackaton06 | Hackaton07 |

## Politique WAF

Le WAF doit être monté en utilisant apache mod security et apache http proxy

Il écoute sur le port 8080 et redirige vers le port 80

La configuration du WAF comprends :

* La configuration modsecurity par défaut ( modsecurity.conf-recommended) avec le passage en mode bloquant
* Les jeux de règles du Core Rule Set de base (base\_rules, les optional\_rules et experimental\_rules) ne sont pas activées

*Ce point est important pour permettre de comparer les résultats des différents groupes.*

## Script de test

Le script de test peut rester sur votre machine ou être déplacé sur le serveur.

Vous avez totale liberté pour concevoir le script. Il peut utiliser des outils / libs disponibles mais vous devez savoir précisément expliquer ce qu’il fait. C’est à vous de concevoir votre politique de log pour pouvoir les interpréter par la suite.

## Serveur web

Le serveur web doit écouter sur le port 80.

Vous avez la liberté de choisir le type de serveur web et la manière de la configurer / programmer. Néanmoins il est important d’avoir une configuration flexible notamment pour prévoir des patterns de réponses génériques.

De la même manière c’est à vous de concevoir votre politique de log pour pouvoir les interpréter par la suite.

## Script(s) d’interprétations des logs

Le ou les scripts servant à la consolidation des résultats / l’interprétation des logs peut rester sur votre machine ou être déplacé sur le serveur.

Comme évoqué dans les critères de notation, l’explicabilité des résultats est un point important

## Dépôt du code

Un serveur git pour déposer votre code (scripts de test, d’interprétation des résultats + serveur web) vous sera fourni. L’ensemble des éléments devra être déposé avant la fin du hackaton.