WEB APPLICATION FIREWALL

YASSINE LARHLID – ADAM HACHAM

28/02/2021

WEB APPLICATION FIREWALL

1. Introduction

Dans ce projet de sécurité réseau nous allons être amenés à découvrir ensemble la technologie WAF (Web Application Firewall) cette dernière protège les applications web, il peut être considéré comme un pare-feu au niveau de la couche 7 du modèle osi.
Après avoir exposé les outils qu'on va utiliser ainsi que leur installation, nous découvrirons cette technologie et le fonctionnement de sa synthèse.

2. Qu'est-ce que un WAF ?

Un pare-feu applicatif web (WAF) protège les applications contre les attaques sophistiquées de la couche 7 qui pourraient autrement entraîner la perte de données sensibles, le détournement du système par des attaquants et des temps d'arrêt.

Un WAF protège les applications en surveillant et en filtrant activement le trafic. Il recherche les types d'attaques les plus courants tels que l'injection SQL, les scripts intersites (XSS), les inclusions de fichiers et d'autres types d'intrusion active. C'est comme un bouclier placé juste devant votre serveur web pour empêcher les attaques potentiellement dangereuses.

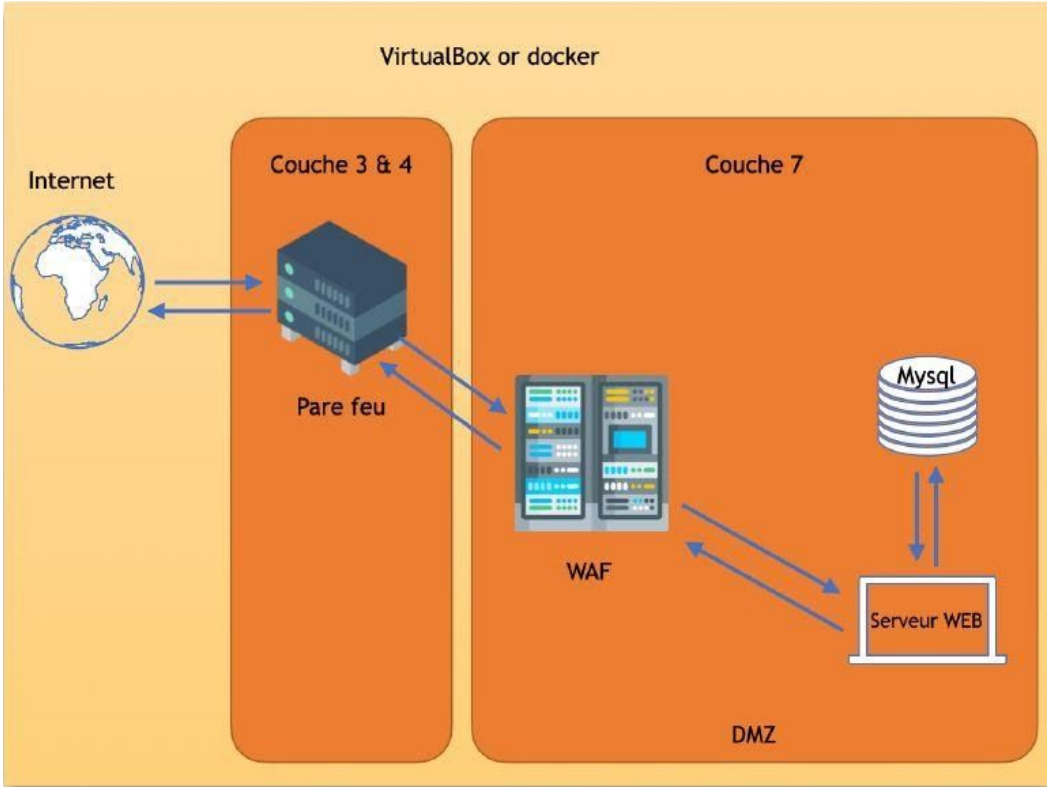
Même lorsque l'on comprend la sécurité, il est difficile de créer des applications sécurisées, surtout lorsqu'on travaille sous la pression si courante dans les entreprises d'aujourd'hui. Un WAF protège les applications contre les attaques sophistiquées de la couche 7 qui pourraient autrement entraîner la perte de données sensibles, le détournement de systèmes par des attaquants et des temps d'arrêt.

a) Les Différentes type de WAF

- Hardware : Un produit physique installé par votre service informatique qui a une faible latence mais des coûts accrus.
- Cloud : Généralement fournie par un tiers et donc non préférée pour la protection des applications en raison du manque de contrôles et d'accès.

3. Architecture du projet

L'architecture du projet consiste en une réflexion approfondie pour la mise en place d'une politique de sécurité, dans le but de rendre notre SI (système d'information) le moins vulnérable possible aux attaques réseaux. Comme notre projet est accessible sur l'adresse IP 176.187.98.30 , il est nécessaire de mener à bien cette politique de sécurité dans le but d'éviter toute intrusion au sein de notre réseau. Pour cela , avant de mettre en place le WAF , il est primordial de réduire un maximum notre surface d'attaque et ne laisser passer que les flux nécessaires pour mener à bien notre projet WAF. On est aussi amené à toucher à la couche 3 et 4 du modèle OSI.



Le choix de cette architecture pour notre projet nous permet de bien segmenter et sécuriser nos flux de données. Ainsi par exemple seul le serveur web à accès à la base de donnée ce qui fait qu'on ne peut pas se connecter directement à la base de données depuis l'extérieur .

4. Politique de sécurité

Notre politique de sécurité consiste dans un premier temps en la segmentation de notre réseau.

Qu'est-ce que la segmentation réseau ?

La segmentation réseau est une technique ayant pour objectif de diviser un réseau informatique en plusieurs sous-réseaux. La segmentation est principalement utilisée afin d'augmenter les performances globales du réseau et améliorer sa sécurité.

Dans un deuxième lieu, la réduction de notre surface d'attaque.

Qu'est-ce qu' une surface d'attaque et comment la réduire pour minimiser les attaques visant notre SI ?

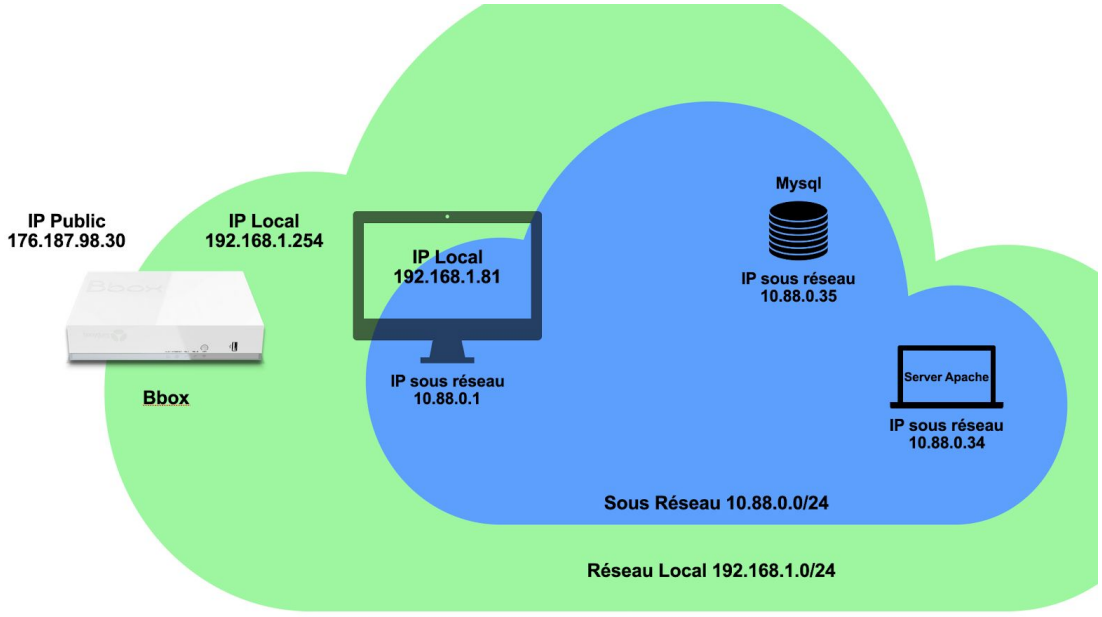
La surface d'attaque ou surface d'exposition est la somme des différents points faibles (les « vecteurs d'attaque ») par lesquels un utilisateur non autorisé (un « pirate ») pourrait potentiellement s'introduire dans un environnement logiciel et en soutirer des données.

Minimiser le plus possible la surface d'attaque fait partie des mesures de sécurité de base.

La réduction de la surface d'attaque consiste en la limitation du nombre de points d'entrée disponibles pour les utilisateurs non approuvés et élimination des services relativement peu demandés.

Dans le cas de notre projet on a besoin que du port 80 (protocole HTTP) le reste des protocoles non utilisés tel que le protocole FTP ou d'autre service tel que MYSQL qui en général utilise le port 3306 doivent être impérativement fermé pour mener à bien cette politique de sécurité que nous avons mis en place.

Ainsi une utilisation de nmap (scanner de port) détectera un seul port ouvert qui est le port 80 nécessaire au fonctionnement du protocole HTTP.



Cette image montre bien le cloisonnement qu'on a réalisé.

Bbox deux interfaces une interface 176.187.98.30 gérer par le fournisseur d'accés et une seconde interface 192.168.1.254 qui nous donne accés à la box pour pouvoir la configurer

une machine avec deux interfaces, la première en 192.168.1.81 une IP local qui permet à la box de rediriger le trafic vers la machine et une seconde 10.88.0.1 qui joue le rôle de routeur au niveau du sous-réseau 10.88.0.0

5. Les outils

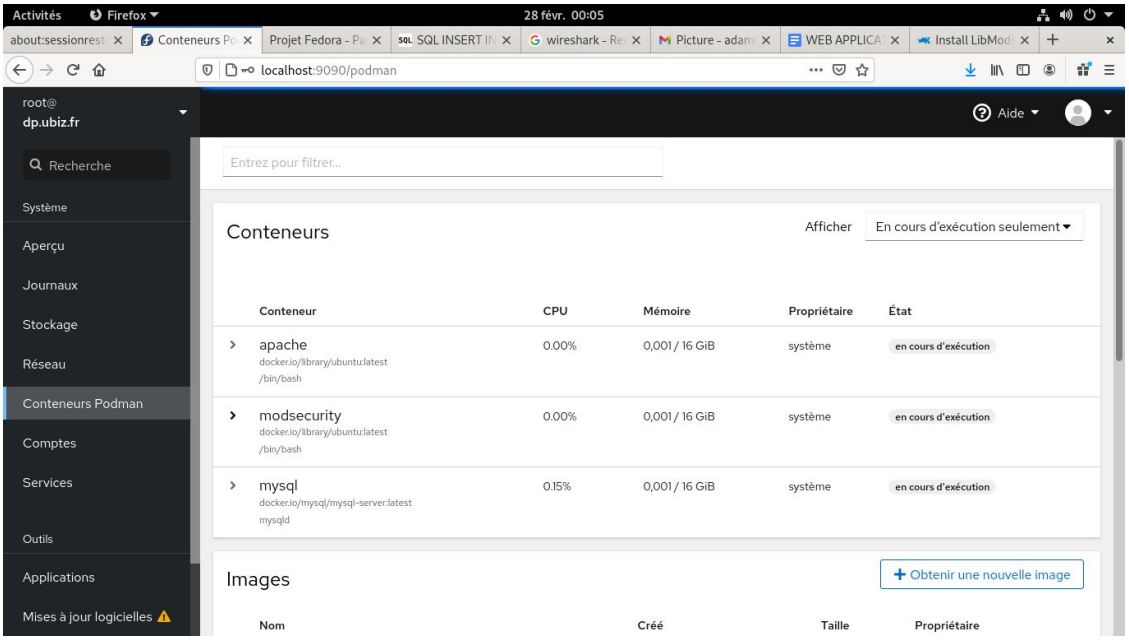
Premièrement on a choisi une distribution Open Source basé sur linux orientée serveur nommée Fedora Server.

Un environnement de virtualisation plus précisément un gestionnaire de conteneur utilisant des images docker ici nommée Podman c'est l'équivalent de docker sur fedora.

Une Box Bouygue Telecom configuré pour autoriser les flux entrants depuis l'extérieur où le trafic est redirigé vers un poste fix qui fait office de serveur ou SI (système d'information) .

Trois conteneur (équivalent d'une machine virtuel) :

- Un serveur Web qui tourne sur Ubuntu pour l'hébergement du site web codée en PHP, on a choisi le logiciel libre APACHE situé dans le sous-réseau 10.88.0.0
- Une machine Linux qui héberge une base de données MYSQL pour la mise en place d'une authentification nécessaire pour la mise en place d'une attaque par injection SQL.
- Une machine Linux qui fait office de WAF avec le logiciel libre ModSecurity.



Outils réseau:

- firewall-cmd commande fedora basée sur iptables utilisée pour la redirection de port (port forwarding) , ou encore pour fermer des ports ou des services tel que MYSQL et qui joue le role d'un pare-feu au niveau de la couche réseau (couche 4 du modèle OSI)
- route commande linux pour définir le routage nécessaire pour joindre le sous réseau 10.88.0.0
- tcpdump et wiresharke pour analyser les paquets
- ifconfig

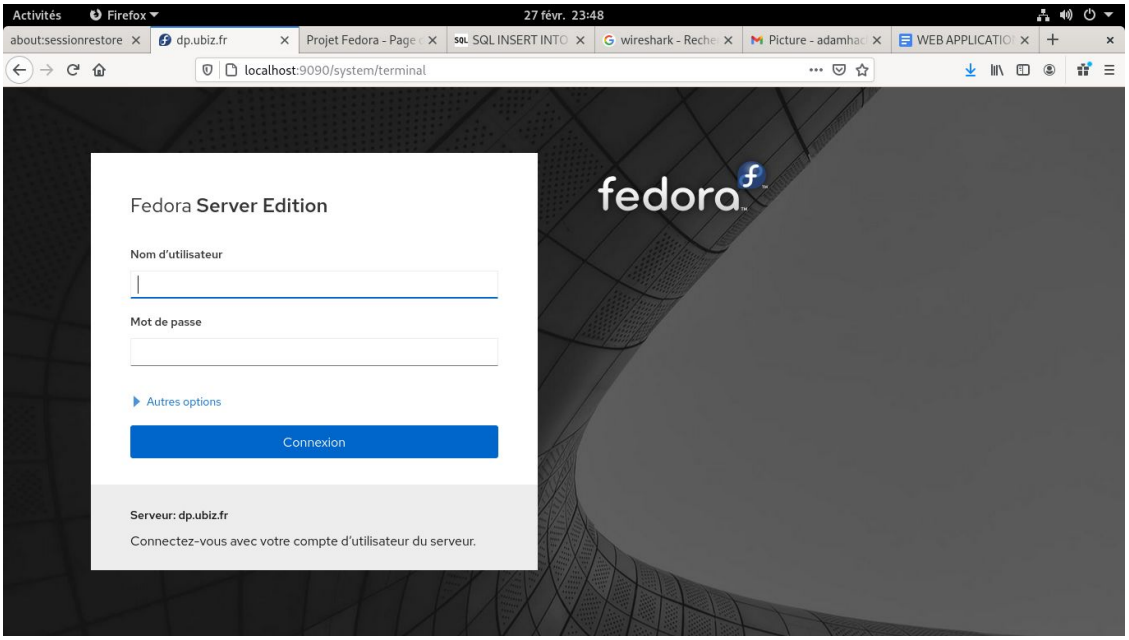
- ping commande utilisé lors de la configuration du SI utile pour vérifier l'acheminement des paquets

Outils Système :

- Service (commande Fedora équivalent à systemctl ou apachectl dans d'autre distribution Linux) utilisé pour le démarrage d'un service tel que apache (Serveur Web)
- podman (commande Fedora) utiliser pour démarrer les conteneurs et aussi pour accéder au machine virtuel.
- mysql pour la configuration de la base de donnée

6. La réalisation du projet

Pour pouvoir travailler à deux et à distance on a utilisé le service Cockpit qui est une interface graphique de Fedora Serveur et qui met à notre disposition un terminal accessible sur le port 9090 une configuration de la box est nécessaire pour autoriser le trafic vers ce port.



Interface Cockpit accessible via l'adresse 176.187.98.30:9090

Une fois authentifié on peut utiliser la commande suivant pour afficher la liste des conteneurs (machine virtuel) qui sont démarrer, on peut alors voir le nom, l'image docker et aussi l'ID nécessaire pour l'accés à la machine depuis le terminal

```
[root@dp ~]# podman ps -a --pod
CONTAINER ID  IMAGE                                 COMMAND    CREATED        STATUS            PORTS   NAMES        POD ID
  PODNAME
f243502518e8  docker.io/library/ubuntu:latest       /bin/bash  4 minutes ago  Up 4 minutes ago          modsecurity

e03537e81e89  docker.io/mysql/mysql-server:latest   mysqld     2 weeks ago    Up 2 days ago             mysql

dc2d7f054afb  docker.io/library/ubuntu:latest       /bin/bash  2 weeks ago    Up 2 days ago             apache
```

listez les images qui tourne avec la commande ***podman ps -a --pod***

Pour avoir un terminal sur une machine il suffit de taper la commande podman exec -it dc2d7f054afb bash et ainsi on obtient un shell de la machine demandé. On a juste à renseigner l'ID et la commande à exécuter

```
[root@dp ~]# podman exec -it dc2d7f054afb bash
root@dc2d7f054afb:/#
```

L'image ci-dessus présente un changement de machine et accès à la machine Ubuntu où on peut configurer notre Serveur Web Apache

Le fichier index.php est le fichier root qui contient la page principale de notre Web Interface et qui contient le code HTML ainsi que la balise <form> qui présente un formulaire d'authentification où l'on exécutera l'attaque par injection SQL. Cette balise form renvoi au fichier action page.php une fois les champs username et password rempli. c'est dans ce fichier où l'on voit si les données insérer dans les champs dans le index.php pour l'authentification son tester pour voir s'ils correspondent bien à un utilisateur qui figure dans la base de données .

```
root@dc2d7f054afb: /var/www/html
```

```
root@dc2d7f054afb:/var/www/html# ls
action_page.php  config.php  index.html  index.php
root@dc2d7f054afb:/var/www/html#
```

La configuration du serveur web Apache reste simple quelque ligne à décommenté dans un des fichiers de configuration pour pouvoir utiliser le Langage PHP et une création d'un virtualHost sur le port 80 liée au Dossier html situé dans le /var/www/html comme on peut le voir ci-dessous.

```
root@dc2d7f054afb:/etc/apache2# cd sites-enabled/
root@dc2d7f054afb:/etc/apache2/sites-enabled# cat 000-default.conf
<VirtualHost *:80>
        # The ServerName directive sets the request scheme, hostname and port that
        # the server uses to identify itself. This is used when creating
        # redirection URLs. In the context of virtual hosts, the ServerName
        # specifies what hostname must appear in the request's Host: header to
        # match this virtual host. For the default virtual host (this file) this
        # value is not decisive as it is used as a last resort host regardless.
        # However, you must set it for any further virtual host explicitly.
        #ServerName 10.88.0.27

        ServerAdmin webmaster@localhost
        DocumentRoot /var/www/html

        # Available loglevels: trace8, ..., trace1, debug, info, notice, warn,
        # error, crit, alert, emerg.
        # It is also possible to configure the loglevel for particular
        # modules, e.g.
        #LogLevel info ssl:warn

        ErrorLog ${APACHE_LOG_DIR}/error.log
        CustomLog ${APACHE_LOG_DIR}/access.log combined

        # For most configuration files from conf-available/, which are
        # enabled or disabled at a global level, it is possible to
        # include a line for only one particular virtual host. For example the
        # following line enables the CGI configuration for this host only
        # after it has been globally disabled with "a2disconf".
        #Include conf-available/serve-cgi-bin.conf
</VirtualHost>
```

Comme on peut le voir sur cette image le DocumentRoot /var/www/html c'est ce qui permet au Serveur Web de connaître le chemin des fichier qu'il vas présenter au utilisateur.

Le fichier config.php est le fichier qui fait la connexion avec la base de données Mysql

```
root@dc2d7f054afb: /var/www/html
```

```
 GNU nano 4.8                                    config.php
<?php
$servername = "10.88.0.35";
$username = "root";
$password = "adamadam";
$db = "utilisateur";
// Create connection
$conn = mysqli_connect($servername, $username, $password, $db);

// Check connection
if (!$conn) {
  die("Connection failed: " . mysqli_connect_error());
}
return $conn;
?>
```

Fichier Config.php

7. MYSQL

Notre base de données consiste d'un tableau qui contient 3 colonnes, un ID (le primary key), une colonne "Username" et une autre "password".

Afin de maximiser la sécurité, seul le serveur web à accès à la base de données, et il faut lui autoriser l'accès depuis MySQL.

```
mysql> create user 'root'@'10.88.0.34' identified with mysql_native_password by 'adamadam';
Query OK, 0 rows affected (0.49 sec)

mysql> GRANT ALL PRIVILEGES ON *.* TO 'root'@'10.88.0.34';
Query OK, 0 rows affected (0.38 sec)

mysql> flush privileges;
Query OK, 0 rows affected (0.07 sec)
```

L'autorisation au serveur web l'accès à MySQL

```
mysql> select host from user;
+------------+
| host       |
+------------+
| 10.88.0.34 |
| localhost  |
| localhost  |
| localhost  |
| localhost  |
| localhost  |
+------------+
6 rows in set (0.00 sec)
```

Liste des autoriser

```
mysql> select * from accounts;
+----+----------+------------+-------------------+
| id | username | password   | email             |
+----+----------+------------+-------------------+
|  1 | ahacham  | univparis8 | adam@univ.com     |
|  2 | yassine  | yassine212 | yassine@univ.com  |
|  3 | salah    | salah001   | salah@univ.com    |
+----+----------+------------+-------------------+
3 rows in set (0.00 sec)

mysql>
```

Le Tableau accounts

8. Filtrage d'une attaque par injection SQL

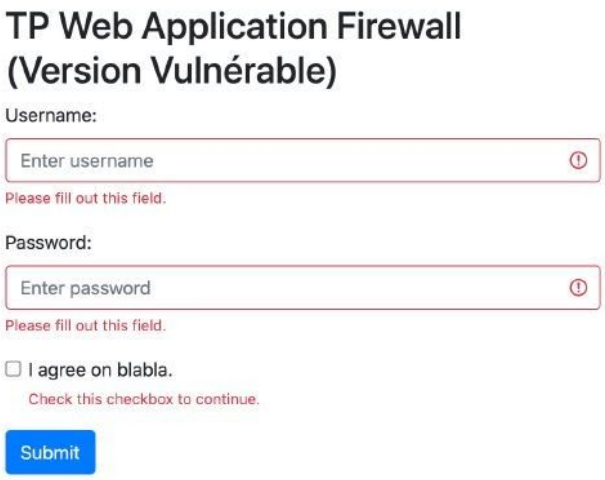
Contexte

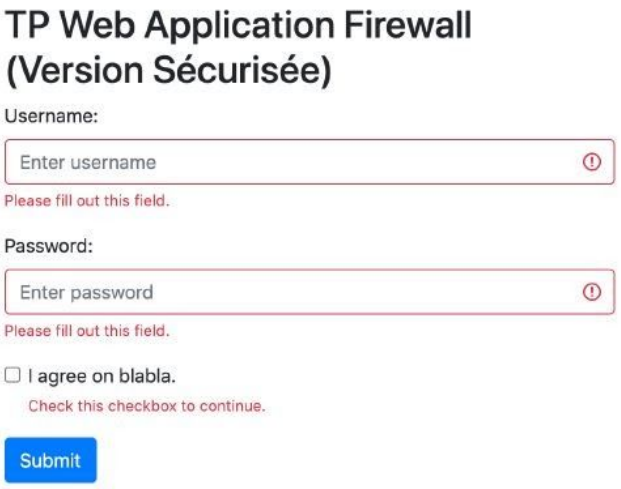
On se place dans le cas d'une entreprise qui désire mettre à disposition d'entreprises externes une application web permettant de gérer un répertoire des employés.

Constatation du problème

Il faut se connecter sur http://176.187.98.30

Cette application va nous permettre de tester nos attack





Lorsque le bouton « Submit » est cliqué, le script "action page.php" est appelé sur le serveur et deux paramètres (Username et Password) lui sont passés.

Le code de action page.php qui nous intéresse le plus est le suivant :

```
  $mysqli = $conn;
if (isset($_POST['usname']) && isset($_POST['pswd'])){
    $user = $_POST['usname'];
    $pass = $_POST['pswd'];
    $query = "SELECT * FROM accounts WHERE username= '".$user."' AND password = '".$pass."';";
    $result = $mysqli->query($query);
    if ($result -> num_rows > 0) {
        while ($row = $result->fetch_assoc()) {
            $field1name = $row["username"];
            $field2name = $row["email"];
            echo '<b> Bienvenue '.$field1name.' email = '.$field2name.'</b><br />';
        }
```

Les variables Username et Password sont récupérées de la requête, ces derniers sont utilisés pour récupérer dans une base de donnée présente sur le serveur la liste ($result) des utilisateurs dont l'identifiant est égal à username et le mot de passe à une valeur dérivée de Password.

Si cette liste est vide, l'utilisateur est considéré comme non-authentifié et reçoit un message d'erreur. Si cette liste n'est pas vide et que l'utilisateur n'est pas l'administrateur, alors les informations propres à cet utilisateur sont affichées. Enfin, si l'utilisateur est l'administrateur, une nouvelle requête est réalisée pour obtenir les informations de tous les utilisateurs existants et cet ensemble est affiché.

Afin de faire en sorte que la liste ne soit pas vide sans connaître de mot de passe, l'attaquant peut jouer sur la valeur de la variable Username dans la clause WHERE username= '$user' and Password='$pass';. S'il modifie sa valeur en «1' or '1» qui est une tautologie .

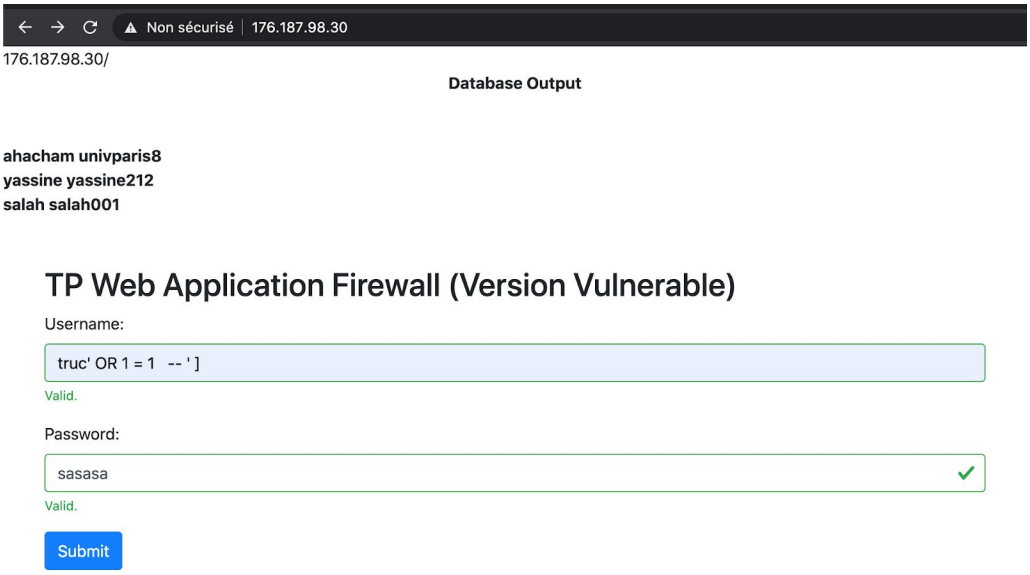
Et la clause prend la valeur (une fois les variables remplacées par leurs valeurs):

Dans notre cas on a effectué deux attaques pour bien expliquer comment un Hacker pense pour attaquer un site par injection SQL.

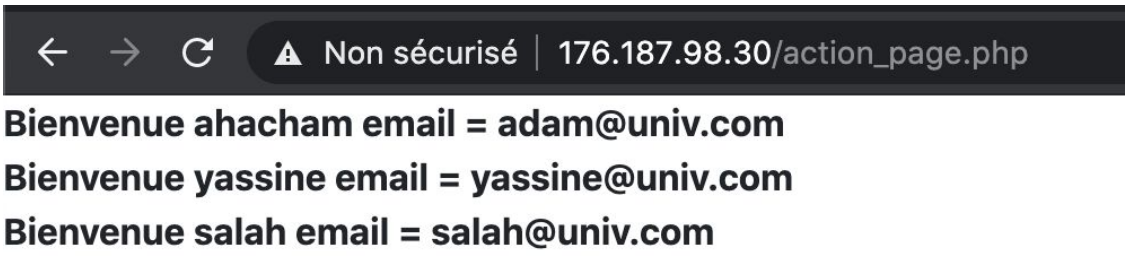
La première Attaque sert à afficher toute la base de données sans avoir d'identifiant mais en renseignant par exemple dans le champ username

xxx' OR 1 = 1 --'] ce qui revient a trafiqué la requête SQL le caractère --'] nous sert à commenter le reste de la requête SQL (AND password = ' ') le faite de commenter le reste de la requête revient à faire un SELECT * FROM TABLE et nous donne donc accès à l'ensemble des infos de la table du site et dans le champ password on met n'importe quoi.

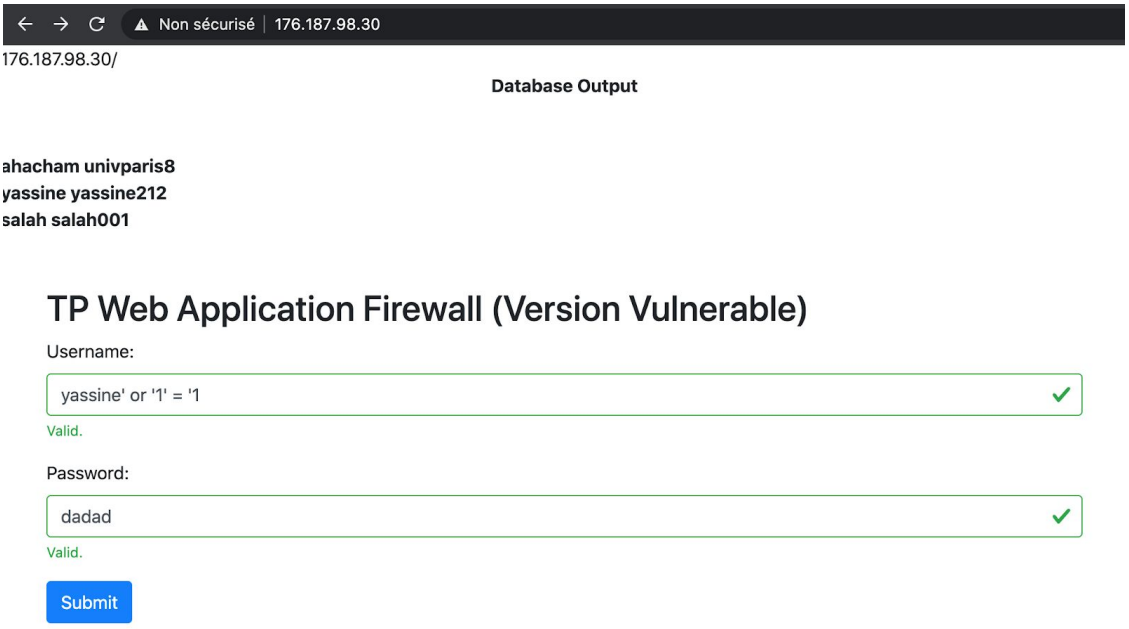
Après cela une seconde Attaque peut être mise en place c'est une attaque qui nous permet de nous authentifier avec un mot de passe faux il suffit de mettre dans le champs username par exemple (yassine' OR '1' = '1)



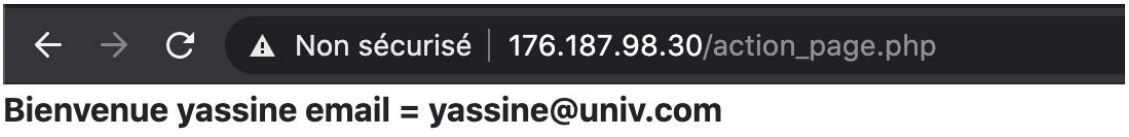
La première Attaque pour récupérer toutes données de la Base MYSQL



Le Résultat de la première attaque



Attaque pour s'authentifier en tant que user il suffit de mettre un nom d'utilisateur présent dans la base de donnée



Résultat de l'attaque

9.Sécurisation

Pour Sécuriser les attaques par injections SQL deux façons de faire la première étant de sécuriser le code en lui même par exemple en utilisant des fonction comme htmlspecialchar() ou encore mysql magic_quote() ces fonctions sont nécessaire pour éviter les injections SQL en évitant l'utilisation des symboles pouvant aider à l'exécution de tels attaques.

Une seconde manière est l'utilisation d'un WAF qui fait office de reverse proxy et qui filtre et analyse les requêtes demander par le client avant de les envoyer au serveur Web ou MYSQL

9. Configuration Initiale de Mod_Security et de la machine qui fait office de reverse proxy

Pour démarrer le proxy il suffit d'utiliser la commande a2enmod proxy proxy_http

on a crée ensuite un fichier de configuration spécifique au proxy proxy.conf dans le répertoire /etc/apache2/sites-available/ il contient

```
ServerName 10.88.0.34
ProxyPass "/" "http://10.88.0.34/"
ProxyPassReverse "/" "http://10.88.0.34/"
ProxyRequests Off
ErrorLog ${APACHE_LOG_DIR}/error.log
CustomLog ${APACHE_LOG_DIR}/access.log combined
```

Fichier de configuration proxy.conf

ServerName indique le nom du serveur.

ProxyPass fait le lien entre les URIs du serveur distant (le serveur web) et les URIs servies par le proxy. Ici on indique que chaque requête à un objet sur le proxy se traduira par une requête au même objet vers le serveur 10.88.0.34.

ProxyPassReverse traduit les réponses du serveur afin de remplacer son nom par le nom du proxy.

ProxyRequests désactive la fonction de proxy traditionnel.

ErrorLog et CustomLog définissent l'emplacement des fichiers de logs.

On a ensuite désactivé le site par défaut et activé la nouvelle configuration :

sudo a2dissite 000-default

sudo a2ensite proxy

Puis redémarré le serveur :

sudo service apache2 restart

Configuration initiale de modsecurity

Modsecurity[6] est un module de filtrage intégrable dans un certain nombre de serveurs web permettant de construire un WAF. Il permet au travers de règles de filtrer le trafic entre un client web et une application web.
Initialement, modsecurity n'est pas configuré. Un fichier de configuration minimal modsecurity.conf-recommended est présent dans le répertoire /etc/modsecurity/. Celui-ci active certaines options de modsecurity (analyse du contenu des requêtes et des réponses, activation de l'analyse de certains types de contenus, définition des tailles maximales des contenus traités, type de logs conservés, emplacement des logs, ...). Modsecurity utilise tous les fichiers ayant l'extension .conf dans ce répertoire . Afin de l'utiliser, il suffit donc de modifier l'extension du fichier .

/etc/modsecurity$ sudo cp modsecurity.conf-recommended 00- modsecurity.conf

Modifiez 00-modsecurity.conf afin qu'il bloque les communications jugée dangereuses au lieu de générer des alertes. Ceci est fait en changeant la directive suivante :

SecRuleEngine DetectionOnly

On peut ensuite tester la configuration pour vérifier qu'elle ne comprend pas d'erreur

sudo service apache2 configtest

Puis relancer apache

sudo service apache2 reload

Politique pour modsecurity

Afin de définir une politique permettant de bloquer l'attaque, nous allons placer nos règles dans un nouveau fichier :/etc/modsecurity/03-sqli.conf.

Une règle dans modsecurity prend le format général suivant :

SecRule VARIABLES OPERATEUR ID,PHASE,TRANSFORMATIONS,ACTIONS
Dans celles-ci :

SecRule est le mot clef définissant une règle.
VARIABLES défini quel(s) champ(s) du protocole utiliser pour savoir si la règle s'applique.
OPERATEUR défini comment les variables sont comparées et avec quoi.
ID défini un identifiant pour la règle.
PHASE défini dans quelle phase la règle va s'appliquer. Modsecurity défini 5 phases. Les phases de 1 et 2 correspondent au traitement de la requête (en-tête puis corps), les phases 3 et 4 correspondent au traitement de la réponse (en-tête puis corps).
SecRuleEngine On

10. Sources

https://fr.wikipedia.org/wiki/Wikip%C3%A9dia:Accueil_principal

https://cloud.ibm.com/docs/cis?topic=cis-waf-q-and-a&locale=fr

https://www.offensive-security.com/offsec/attacking-the-web-offsec-way/