

RAPPORT DE SCAN DE VULNERABILITE

PROJET TEST D'INTRUSION PYTHON

Membres du projets

Noms et prénoms des membres	Etablissement	Filières et Niveau
Kignon Gninnaha Abel	UPB	Master 1 / SAS
Obin Yapi Hermann	UPB	Master 1 / SAS
Bamba N'giantchan Allassane	UPB	Master 1 / SAS
Nassa Grace Marie Paule	UPB	Master 1 / SAS
Kouame Akissi Esther	UPB	Master 1 / SAS

Objectif

Analyse de vulnérabilités : Liste des vulnérabilités trouvées et détails sur chaque vulnérabilité.

Résolution

Cette partie s'appuie principalement sur la bibliothèque python-nmap avec l'utilisation de nmap dans notre script pour le scan de vulnérabilité des applications web testées.

CAS de l'URL : <http://192.168.110.131/wordpress/wp-login.php>

Nous utiliserons ainsi dans cette partie le script python **Analysefull.py** pour l'analyse des vulnérabilités de l'application web WordPress avec en même temps, des tests de fuzzing sur les différents formulaires et comme vous le savez déjà, dans les tests précédents, nous avons pu récupérer des informations sur le dit formulaire.

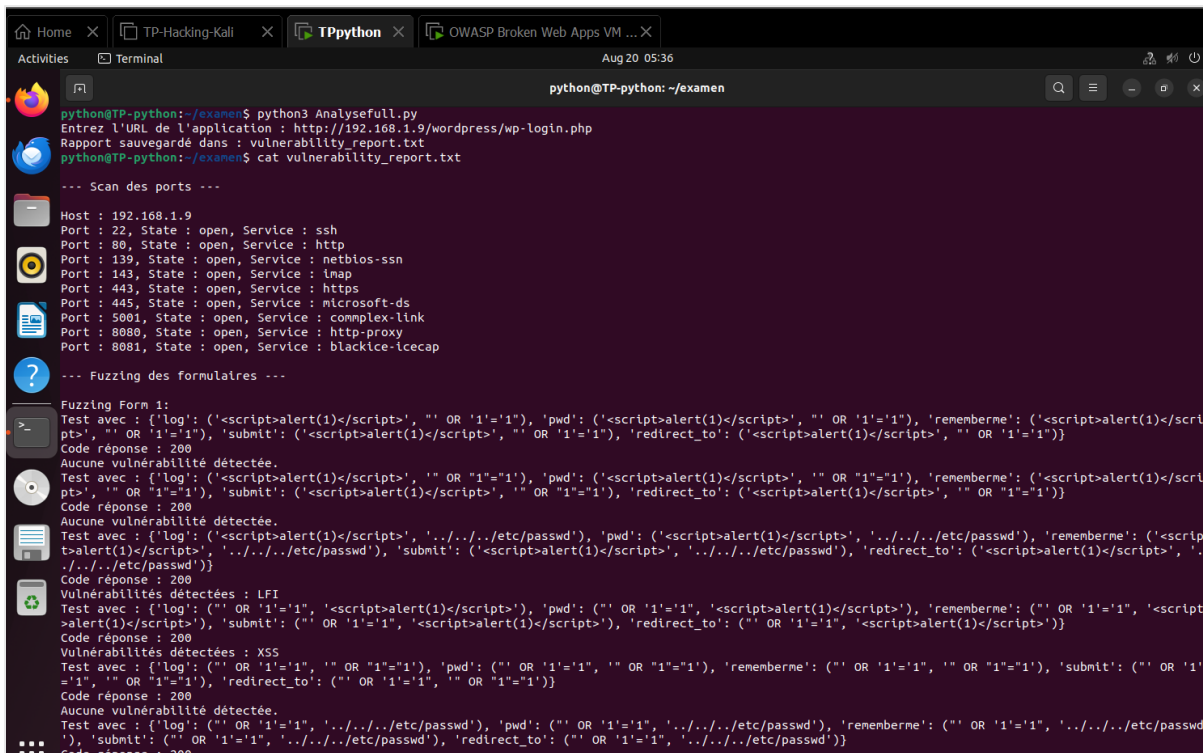
Script Python

```
python3 "/chemin/ Analysefull.py"
```

Résultats de l'exécution du script

Vous verrez ci-dessous, la sortie de l'exécution du script. Deux (02) analyses peuvent être faite en fonction du résultat obtenu. Nous nous concentrerons dans

un premier temps sur celle concernant la sortie nmap et terminerons par celle concernant les tests de fuzzing du formulaire.



```

python@TP-python: ~/examen$ python3 Analysefull.py
Entrez l'URL de l'application : http://192.168.1.9/wordpress/wp-login.php
Rapport sauvegardé dans : vulnerability_report.txt
python@TP-python: ~/examen$ cat vulnerability_report.txt

--- Scan des ports ---

Host : 192.168.1.9
Port : 22, State : open, Service : ssh
Port : 80, State : open, Service : http
Port : 139, State : open, Service : netbios-ssn
Port : 143, State : open, Service : imap
Port : 443, State : open, Service : https
Port : 445, State : open, Service : microsoft-ds
Port : 5001, State : open, Service : complex-link
Port : 8080, State : open, Service : http-proxy
Port : 8081, State : open, Service : blackice-icecap

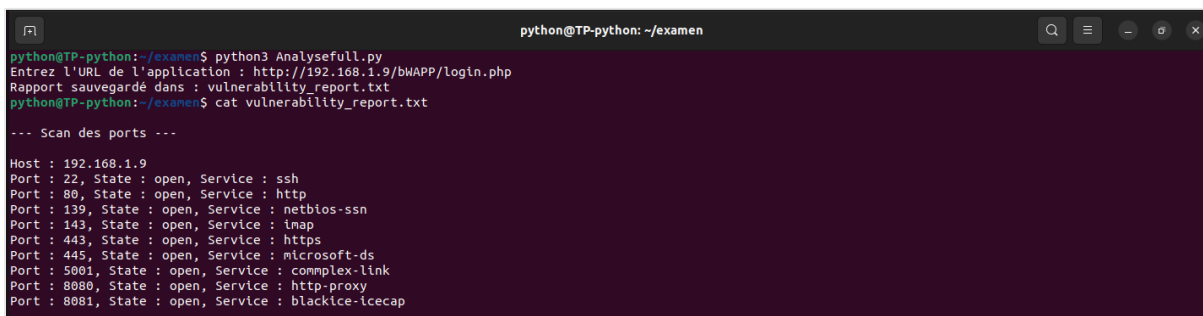
--- Fuzzing des formulaires ---

Fuzzing Form 1:
Test avec : {'log': ('<script>alert(1)</script>', '" OR '1'='1"', 'pwd': ('<script>alert(1)</script>', '" OR '1'='1"', 'rememberme': ('<script>alert(1)</script>', '" OR '1'='1"', 'submit': ('<script>alert(1)</script>', '" OR '1'='1"', 'redirect_to': ('<script>alert(1)</script>', '" OR '1'='1'))}
Code réponse : 200
Aucune vulnérabilité détectée.
Test avec : {'log': ('<script>alert(1)</script>', '" OR "1"="1"', 'pwd': ('<script>alert(1)</script>', '" OR "1"="1"', 'rememberme': ('<script>alert(1)</script>', '" OR "1"="1"', 'submit': ('<script>alert(1)</script>', '" OR "1"="1"', 'redirect_to': ('<script>alert(1)</script>', '" OR "1"="1'))}
Code réponse : 200
Aucune vulnérabilité détectée.
Test avec : {'log': ('<script>alert(1)</script>', '../..../etc/passwd'), 'pwd': ('<script>alert(1)</script>', '../..../etc/passwd'), 'rememberme': ('<script>alert(1)</script>', '../..../etc/passwd'), 'submit': ('<script>alert(1)</script>', '../..../etc/passwd'), 'redirect_to': ('<script>alert(1)</script>', '../..../etc/passwd')}
Code réponse : 200
Vulnérabilités détectées : LFI
Test avec : {'log': ('" OR '1'='1', '<script>alert(1)</script>', 'pwd': ('" OR '1'='1', '<script>alert(1)</script>', 'rememberme': ('" OR '1'='1', '<script>alert(1)</script>', 'submit': ('" OR '1'='1', '<script>alert(1)</script>', 'redirect_to': ('" OR '1'='1', '<script>alert(1)</script>')}
Code réponse : 200
Vulnérabilités détectées : XSS
Test avec : {'log': ('" OR '1'='1', '" OR "1"="1"', 'pwd': ('" OR '1'='1', '" OR "1"="1"', 'rememberme': ('" OR '1'='1', '" OR "1"="1"', 'submit': ('" OR '1'='1', '" OR "1"="1"', 'redirect_to': ('" OR '1'='1', '" OR "1"="1'))}
Code réponse : 200
Aucune vulnérabilité détectée.
Test avec : {'log': ('" OR '1'='1', '../..../etc/passwd'), 'pwd': ('" OR '1'='1', '../..../etc/passwd'), 'rememberme': ('" OR '1'='1', '../..../etc/passwd'), 'submit': ('" OR '1'='1', '../..../etc/passwd'), 'redirect_to': ('" OR '1'='1', '../..../etc/passwd')}
Code réponse : 200

```

Analyse NMAP

Lorsque nous avons exécuté le script, nous avons obtenu dans un premier temps, le résultat nmap suivant :



```

python@TP-python: ~/examen$ python3 Analysefull.py
Entrez l'URL de l'application : http://192.168.1.9/bWAPP/login.php
Rapport sauvegardé dans : vulnerability_report.txt
python@TP-python: ~/examen$ cat vulnerability_report.txt

--- Scan des ports ---

Host : 192.168.1.9
Port : 22, State : open, Service : ssh
Port : 80, State : open, Service : http
Port : 139, State : open, Service : netbios-ssn
Port : 143, State : open, Service : imap
Port : 443, State : open, Service : https
Port : 445, State : open, Service : microsoft-ds
Port : 5001, State : open, Service : complex-link
Port : 8080, State : open, Service : http-proxy
Port : 8081, State : open, Service : blackice-icecap

```

Vous trouverez ci-dessous les informations de l'analyse.

Tableau récapitulatif

Port	État	Service	Description	Analyse et Vulnérabilité Potentielle
22	Ouvert	SSH	Service de connexion à distance sécurisée.	Risque de brute force. Nécessite une sécurisation via des politiques de mot de passe et 2FA.

80	Ouvert	HTTP	Serveur web non sécurisé.	Cible pour les attaques web (XSS, injection SQL). Vérification des configurations web nécessaire.
139	Ouvert	NetBIOS-SSN	Service de partage de fichiers sous Windows.	Expose des ressources partagées, sujet à des attaques d'énumération et d'accès non autorisé.
143	Ouvert	IMAP	Protocole de récupération des emails.	Peut exposer des identifiants de messagerie si mal sécurisé.
443	Ouvert	HTTPS	Serveur web sécurisé via SSL/TLS.	Vérification des certificats et de la configuration SSL pour éviter les failles SSL/TLS.
445	Ouvert	Microsoft-DS (SMB)	Protocole de partage de fichiers et d'imprimantes.	Risque élevé d'exploitation (WannaCry). Vérifier les correctifs et la configuration SMB.
5001	Ouvert	Commplex-Link	Service de communication peu commun.	Nécessite une analyse plus approfondie pour identifier le logiciel et les vulnérabilités associées.
8080	Ouvert	HTTP Proxy	Proxy HTTP ou serveur web alternatif.	Potentiel d'utilisation abusive pour contourner des restrictions réseau ou des attaques relayées.
8081	Ouvert	BlackICE-ICEcap	Associé à des systèmes de surveillance réseau (IDS).	Possible point d'entrée si le service de surveillance est mal configuré.
Port	État	Service	Description	Analyse et Vulnérabilité Potentielle

Analyse du retour fuzzing

Pour le fuzzing, technique qui consiste à injecter des données aléatoires, incorrectes ou inattendues dans une application ou un système afin de détecter des failles de sécurité, des vulnérabilités, ou des erreurs de fonctionnement (Pousser le programme à ses limites pour voir comment il réagit à des entrées non valides).

On constate que pour deux de nos tests, il y'a eu une réponse (voir ci-dessous).

```

--- Fuzzing des formulaires ---

Fuzzing Form 1:
Test avec : {'log': ('<script>alert(1)</script>', '' OR '1'='1'), 'pwd': ('<script>alert(1)</script>', '' OR '1'='1'), 'remember': ('<script>alert(1)</script>', '' OR '1'='1'), 'submit': ('<script>alert(1)</script>', '' OR '1'='1'), 'redirect_to': ('<script>alert(1)</script>', '' OR '1'='1')}
Code réponse : 200
Aucune vulnérabilité détectée.

Test avec : {'log': ('<script>alert(1)</script>', '' OR '1'='1'), 'pwd': ('<script>alert(1)</script>', '' OR '1'='1'), 'remember': ('<script>alert(1)</script>', '' OR '1'='1'), 'submit': ('<script>alert(1)</script>', '' OR '1'='1'), 'redirect_to': ('<script>alert(1)</script>', '' OR '1'='1')}
Code réponse : 200
Aucune vulnérabilité détectée.

Test avec : {'log': ('<script>alert(1)</script>', '../..../etc/passwd'), 'pwd': ('<script>alert(1)</script>', '../..../etc/passwd'), 'remember': ('<script>alert(1)</script>', '../..../etc/passwd'), 'submit': ('<script>alert(1)</script>', '../..../etc/passwd'), 'redirect_to': ('<script>alert(1)</script>', '../..../etc/passwd')}
Code réponse : 200
Vulnérabilités détectées : LFI

Test avec : {'log': ('' OR '1'='1', '<script>alert(1)</script>'), 'pwd': ('' OR '1'='1', '<script>alert(1)</script>'), 'remember': ('' OR '1'='1', '<script>alert(1)</script>'), 'submit': ('' OR '1'='1', '<script>alert(1)</script>'), 'redirect_to': ('' OR '1'='1', '<script>alert(1)</script>')}
Code réponse : 200
Vulnérabilités détectées : XSS

Test avec : {'log': ('' OR '1'='1', '' OR '1'='1'), 'pwd': ('' OR '1'='1', '' OR '1'='1'), 'remember': ('' OR '1'='1', '' OR '1'='1'), 'submit': ('' OR '1'='1', '' OR '1'='1'), 'redirect_to': ('' OR '1'='1', '' OR '1'='1')}
Code réponse : 200
Aucune vulnérabilité détectée.

Test avec : {'log': ('' OR '1'='1', '../..../etc/passwd'), 'pwd': ('' OR '1'='1', '../..../etc/passwd'), 'remember': ('' OR '1'='1', '../..../etc/passwd'), 'submit': ('' OR '1'='1', '../..../etc/passwd'), 'redirect_to': ('' OR '1'='1', '../..../etc/passwd')}
Code réponse : 200

```

Les réponses concernent entre autres :

- Vulnérabilité LFI (Local File Inclusion)

L'inclusion de fichiers locaux (LFI) se produit lorsque l'application web permet à un attaquant d'inclure des fichiers situés sur le serveur via des paramètres non sécurisés. Cela peut permettre à un attaquant de lire des fichiers sensibles sur le serveur (comme `/etc/passwd` sous Linux, qui contient des informations sur les utilisateurs) et potentiellement exécuter des scripts malveillants et permettre ainsi des accès non autorisés à des fichiers critiques du système, vol de données sensibles, et, dans certains cas, possibilité d'exécution de code malveillant (si l'attaquant parvient à inclure des fichiers exécutables).

- Vulnérabilité XSS (Cross-Site Scripting)

Le script injecté dans notre cas ici est une simple alerte JavaScript :

`<script>alert(1)</script>`. Cela montre que l'application est potentiellement vulnérable à des attaques de type Cross-Site Scripting.

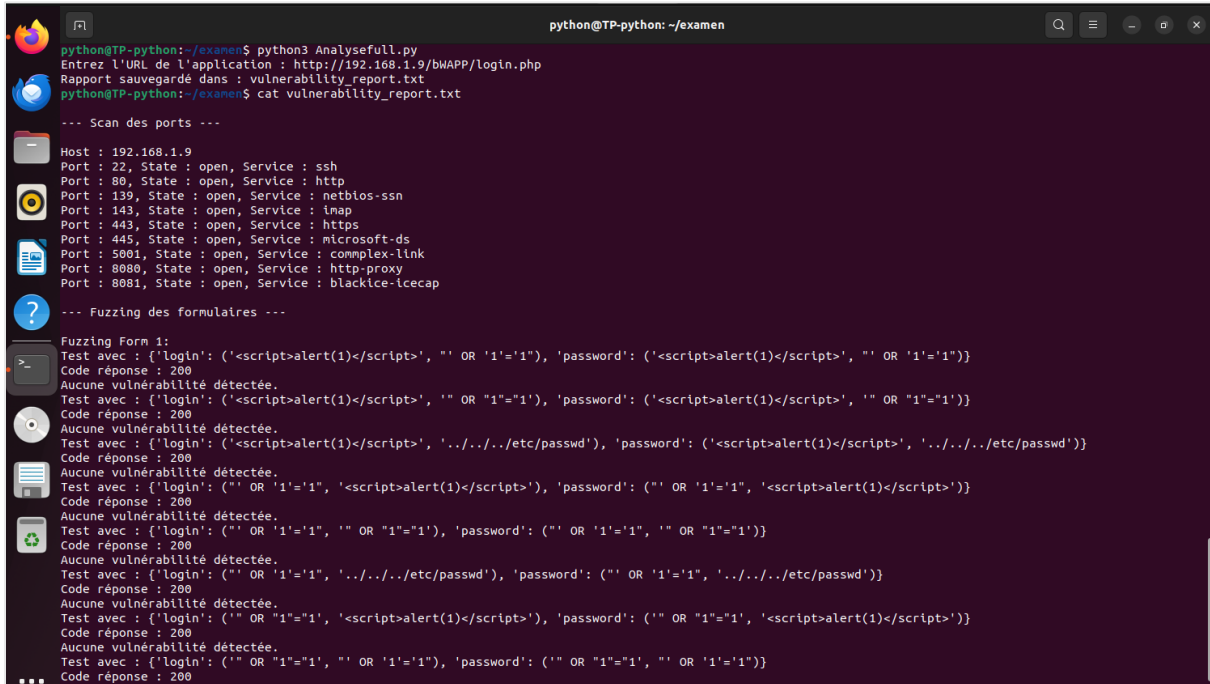
Si l'attaque XSS est réussie, un attaquant peut injecter du contenu malveillant dans les pages web, ce qui peut conduire au vol de cookies, à la redirection vers des sites malveillants, ou même à l'exécution de commandes dans le contexte d'un utilisateur légitime. Et comme risque, les utilisateurs de l'application peuvent être victimes de phishing, de détournement de session, ou d'autres attaques ciblant leurs informations personnelles.

CAS de l'URL : http://192.168.110.131/bWAPP/login.php

Cette partie sera abordée de la même manière que la partie précédente.

Résultats de l'exécution du script

On a constaté qu'avec le même script : **Analysefull.py**, on n'a pas détecté de vulnérabilité concernant cette application contrairement à celle traitée dans la première partie.



```

python@TP-python: ~/examen$ python3 Analysefull.py
Entrez l'URL de l'application : http://192.168.1.9/bWAPP/login.php
Rapport sauvegardé dans : vulnerability_report.txt
python@TP-python: ~/examen$ cat vulnerability_report.txt

--- Scan des ports ---
Host : 192.168.1.9
Port : 22, State : open, Service : ssh
Port : 80, State : open, Service : http
Port : 139, State : open, Service : netbios-ssn
Port : 143, State : open, Service : imap
Port : 443, State : open, Service : https
Port : 445, State : open, Service : microsoft-ds
Port : 5001, State : open, Service : complex-link
Port : 8080, State : open, Service : http-proxy
Port : 8081, State : open, Service : blackice-icecap

--- Fuzzing des formulaires ---
Fuzzing Form 1:
Test avec : {'login': ('<script>alert(1)</script>', '" OR '1'=1'), 'password': ('<script>alert(1)</script>', '" OR '1'=1')}
Code réponse : 200
Aucune vulnérabilité détectée.
Test avec : {'login': ('<script>alert(1)</script>', '" OR "1"=1'), 'password': ('<script>alert(1)</script>', '" OR "1"=1')}
Code réponse : 200
Aucune vulnérabilité détectée.
Test avec : {'login': ('<script>alert(1)</script>', '../..../etc/passwd'), 'password': ('<script>alert(1)</script>', '../..../etc/passwd')}
Code réponse : 200
Aucune vulnérabilité détectée.
Test avec : {'login': ('" OR '1'=1', '<script>alert(1)</script>'), 'password': ('" OR '1'=1', '<script>alert(1)</script>')}
Code réponse : 200
Aucune vulnérabilité détectée.
Test avec : {'login': ('" OR "1"=1', '" OR "1"=1'), 'password': ('" OR "1"=1', '" OR "1"=1')}
Code réponse : 200
Aucune vulnérabilité détectée.
Test avec : {'login': ('" OR '1'=1', '../..../etc/passwd'), 'password': ('" OR '1'=1', '../..../etc/passwd')}
Code réponse : 200
Aucune vulnérabilité détectée.
Test avec : {'login': ('" OR "1"=1', '<script>alert(1)</script>'), 'password': ('" OR "1"=1', '<script>alert(1)</script>')}
Code réponse : 200
Aucune vulnérabilité détectée.
Test avec : {'login': ('" OR "1"=1', '" OR '1'=1'), 'password': ('" OR "1"=1', '" OR '1'=1')}
Code réponse : 200

```

On verra dans la partie d'exploitation qu'il y avait des indices concernant une potentielle exploitation.