Exploring the .hidden directoy

EXPLICATION

Comme relevé en énumération (voir ENUMERATION – partie 9), notre directory fuzzing nous a révélé le dossier suivant :

`http://<IP>/.hidden`

Celui-ci est accessible publiquement, et nous révèle une page de *directory listing* composée d'un fichier README, et de 26 autres dossiers, aux noms générés aléatoirement :

Index of /.hidden/

../		
amcbevgondgcrloowluziypjdh/	11-Sep-2001 21:21	-
bnqupesbgvhbcwqhcuynjolwkm/	11-Sep-2001 21:21	-
ceicqljdddshxvnvdqzzjgddht/	11-Sep-2001 21:21	-
doxelitrqvhegnhlhrkdgfizgj/	11-Sep-2001 21:21	-
eipmnwhetmpbhiuesykfhxmyhr/	11-Sep-2001 21:21	-
ffpbexkomzbigheuwhbhbfzzrg/	11-Sep-2001 21:21	-
ghouhyooppsmaizbmjhtncsvfz/	11-Sep-2001 21:21	-
hwlayeghtcotqdigxuigvjufqn/	11-Sep-2001 21:21	-
isufpcgmngmrotmrjfjonpmkxu/	11-Sep-2001 21:21	-
jfiombdhvlwxrkmawgoruhbarp/	11-Sep-2001 21:21	-
kpibbgxjqnvrrcpczovjbvijmz/	11-Sep-2001 21:21	-
ldtafmsxvvydthtgflzhadiozs/	11-Sep-2001 21:21	-
mrucagbgcenowkjrlmmugvztuh/	11-Sep-2001 21:21	-
ntyrhxjbtndcpjevzurlekwsxt/	11-Sep-2001 21:21	-
oasstobmotwnezhscjjopenjxy/	11-Sep-2001 21:21	-
ppjxigqiakcrmqfhotnncfqnqg/	11-Sep-2001 21:21	-
qcwtnvtdfslnkvqvzhjsmsghfw/	11-Sep-2001 21:21	-
rlnoyduccpqxkvcfiqpdikfpvx/	11-Sep-2001 21:21	-
sdnfntbyirzllbpctnnoruyjjc/	11-Sep-2001 21:21	-
trwjgrgmfnzarxiiwvwalyvanm/	11-Sep-2001 21:21	-
urhkbrmupxbgdnntopklxskvom/	11-Sep-2001 21:21	-
viphietzoechsxwqacvpsodhaq/	11-Sep-2001 21:21	-
whtccjokayshttvxycsvykxcfm/	11-Sep-2001 21:21	-
xuwrcwjjrmndczfcrmwmhvkjnh/	11-Sep-2001 21:21	-
yjxemfsgdlkbvvtjiylhdoaqkn/	11-Sep-2001 21:21	-
zzfzjvjsupgzinctxeqtzzdzll/	11-Sep-2001 21:21	-
README	11-Sep-2001 21:21	34

On se rend compte que chacun de ces dossiers aux noms aléatoires contiennent 26 autres dossiers aux noms aléatoires (ainsi qu'un README). On comprend que ce schéma est répété 3 fois, autrement dit il existe 26 dossiers aléatoires, qui ont chacun 26 autres dossiers aléatoires, qui ont chacun 26 autres dossiers aléatoires qui contiennent eux-même chacun un README.

Il existe bien trop de dossiers pour les examiner un par un manuellement. On attend probablement de nous qu'on trouve le bon README contenant le flag (assez orienté CTF comme défi).

On a créé un petit script qui va s'en occuper pour nous (attention à remplacer l'URL) :

```python
import requests

URL = "http://192.168.1.37/.hidden/"

r = requests.get(URL)
for path in r.text.splitlines()[4:-3] :
    lv1_path = path[9:26 + 9 + 1]
    r2 = requests.get(URL + lv1_path)
    readme = requests.get(URL + lv1_path + "README")
    print(readme.text)
    for path2 in r2.text.splitlines()[4:-3] :
        lv2_path = path2[9:26 + 9 + 1]
        r3 = requests.get(URL + lv1_path + lv2_path)
        readme2 = requests.get(URL + lv1_path + lv2_path + "README")
        print(readme2.text)
        for path3 in r3.text.splitlines()[4:-3] :
            lv3_path = path3[9:26 + 9 + 1]
            r4 = requests.get(URL + lv1_path + lv2_path + lv3_path + "README")
              print(r4.text)
```

Ce script ne fait qu'effectuer des requêtes en récupérant le nom des dossiers à parcourir à partir de la page de *directory listing*, sur 3 couches de profondeur. Il affiche, pour chaque couche, les fichiers README.

On exécute ce script, dont on redirige l'output vers un fichier afin de pouvoir travailler dessus plus facilement :

```
python3 fuzz.py > results
```

En observant le fichier **results**, on se rend compte que la plupart des fichiers README ont des messages standards ("Demande à ton voisin de gauche", "Toujours pas tu vas craquer non ?" ...).

On filtre tous ces messages standards avec une simple commande **grep** :

```
cat results      | grep -v "Non ce n'est toujours pas bon"
                 | grep -v "Tu veux de l'aide ? Moi aussi \!"
                 | grep -v "Demande Ã ton voisin du dessus"
                 | grep -v "Demande Ã ton voisin du dessous"
                 | grep -v "Demande Ã ton voisin de droite"
                 | grep -v "Demande Ã ton voisin de gauche"
                 | grep -v "Toujours pas tu vas craquer non ?"
```

Il ne reste qu'une seule ligne affichée, qui semble être le **flag** (le hash n'a pas la même longueur que les autres flags, cependant c'est peut-être pour faire en sortes que chaque fichier README ait la même taille afin de rendre plus dur à trouver celui qui contient le flag ?).

RESSOURCES

Le script permettant de bruteforce l'ensemble des dossiers, comme décrit ci-dessus.

MITIGATION

Pas vraiment de suggestion ici, puisque ce challenge était assez orienté CTF.