



XSS / SQLi LABS

Ejercicio Práctico: XSS y SQL Injection



Técnicas de Hacking y Pentesting

Modulo 4: Hacking & Pentesting



CFP Centro de Formación
Permanente

Máster Seguridad TIC. (VIII Edición, 2021)



Objetivo:

Poner en practica los conocimiento adquiridos durante las clases teóricas, en particular los correspondientes a la detección de vulnerabilidades WEB del tipo XSS y SQL Injection. En esta practica el alumno aprenderá a detectar dichas vulnerabilidades de **forma manual** utilizando la herramienta proxy de OWASP (ZAP).

Descripción de la Practica:

En la practica de hoy, los alumnos trabajaran de **formar individual**. El ejercicio constará de dos partes diferenciadas:

- ▣ **Primera parte:** Detectar y practicar vulnerabilidades del tipo XSS en un entorno vulnerable.
- ▣ **Segunda parte:** Detectar y practicar vulnerabilidades del tipo SQL Injection.



Trabajo para casa, se deberá de entregar un documento con las respuestas. Se deberá describir los pasos y comandos utilizados, así como aportar capturas de pantalla del proceso.

Requisitos:

Para la realización de esta práctica será necesario que cada alumno disponga de un ordenador, y que cumpla con al menos los siguientes requisitos:

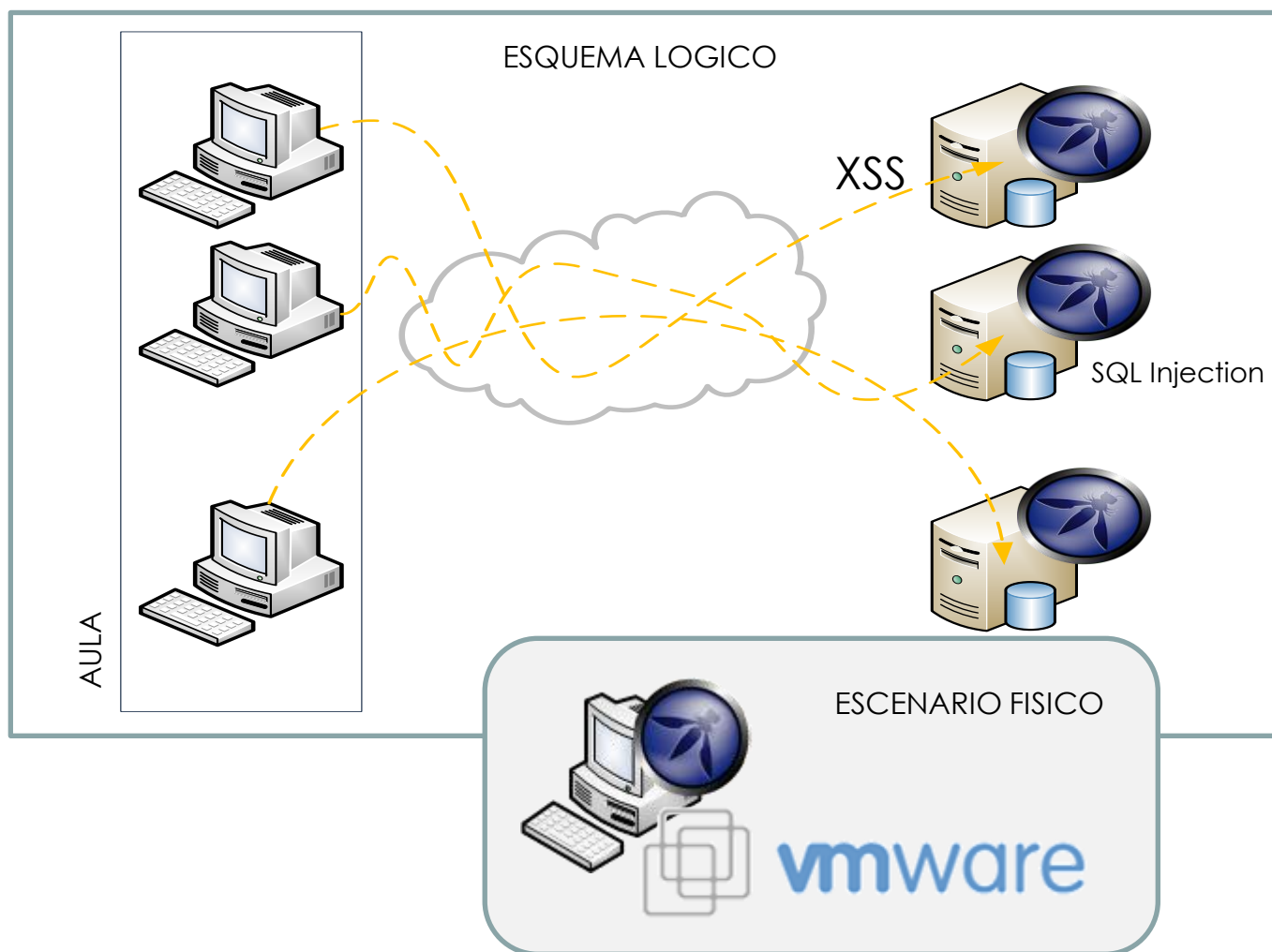
- Equipo con Windows instalado.
- Una máquina Virtual con el entorno WEB vulnerable (OWASP).
- Tener instalado en el equipo Windows la herramienta OWASP Zed Attack Proxy Project (ZAP Proxy).



OWASP ZAP: An easy to use integrated penetration testing tool for finding vulnerabilities in web applications.

<https://code.google.com/p/zaproxy/downloads/list>

Escenario



El esquema lógico permite **contextualizar** el desarrollo de la práctica, donde el alumno puede **comprender** las capacidades y formas de utilizar los conocimientos que se adquieran durante la misma.

Equipo del laboratorio:

Será un equipo del aula que tenga Windows instalado, para la práctica se deberán instalar la herramienta ZAP Proxy e iniciar la máquina virtual OWASP proporcionada por el Profesor.

Formará parte de la práctica, la instalación y configuración del entorno de pruebas.

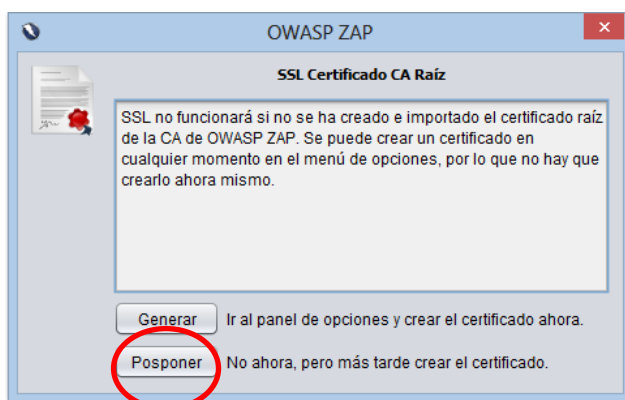
Preparación del escenario

Instalación de la herramienta OWASP

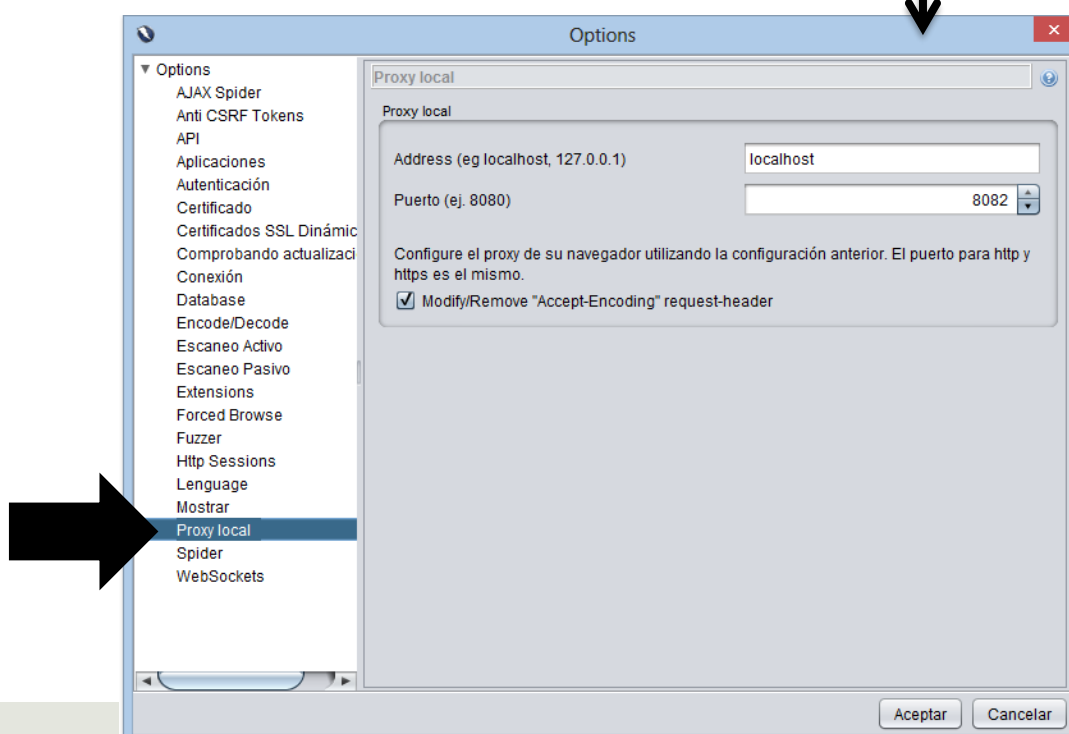
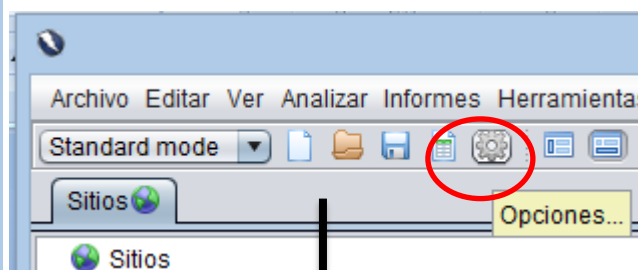
La instalación de la herramienta, sigue el procedimiento habitual de cualquier programa diseñado para el Sistema Operativo Windows. Siga las instrucciones de la pantalla, hasta que el proceso de instalación haya finalizado.

Configuración básica

En el arranque de la aplicación, se le solicitará la generación de unos certificados SSL para el posterior análisis, dado que no es relevante para el desarrollo de estas practicas, no se configurará y se hará "click" en "**Posponer**".



A continuación, se realizará la configuración del Proxy Local en el puerto 8082.



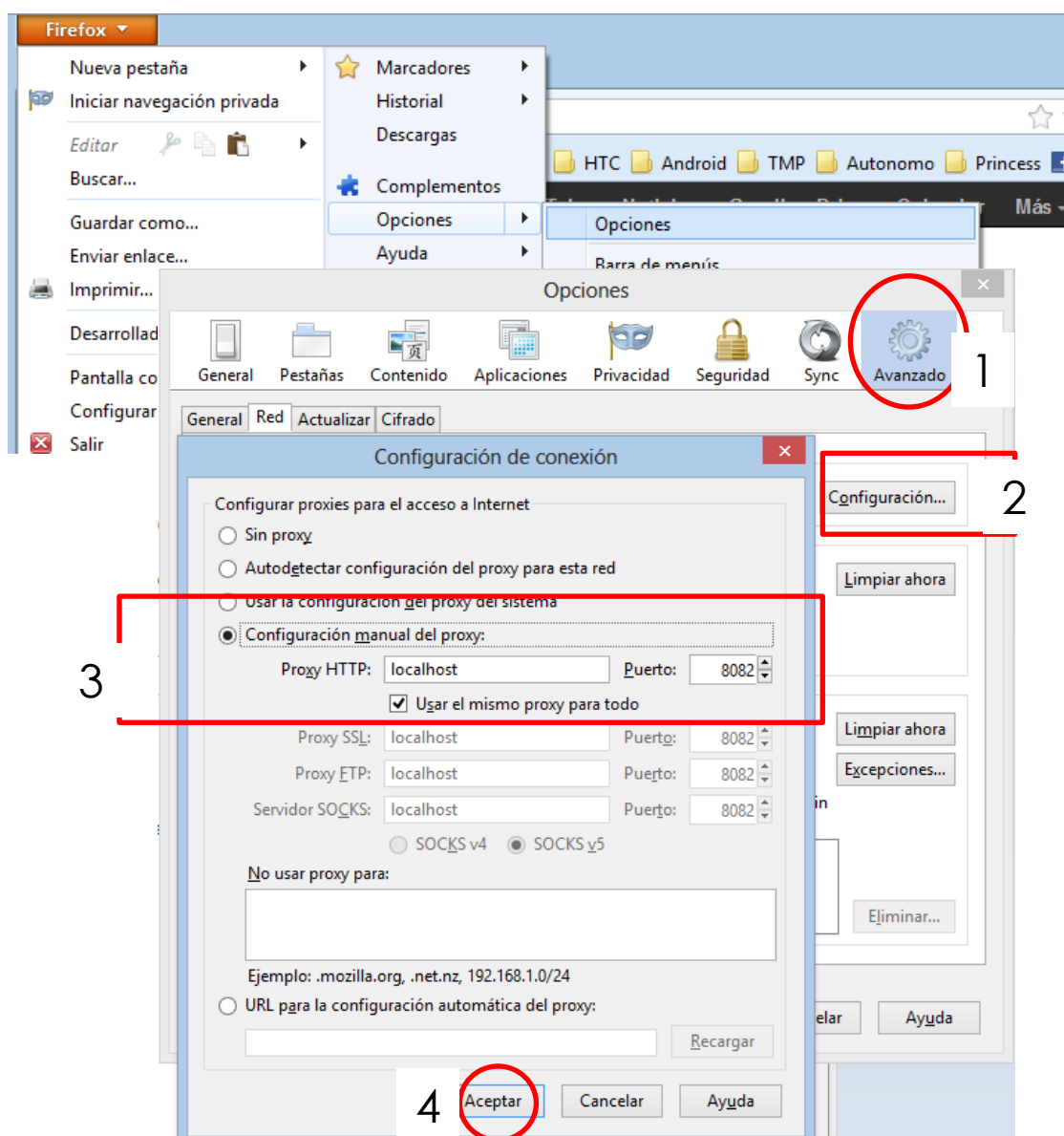
Preparación del escenario

Configuración del Navegador WEB

Para la practica se utilizara el **Mozilla Firefox**, en caso de no estar instalado en el equipo Windows del laboratorio, se procederá a realizar su instalación.

Una vez se encuentra instalado el navegador WEB Mozilla Firefox, se procederá a configurar éste para que se comunique con el nuestro proxy local (localhost) OWASP, en el puerto **8082**.

Acceder al Menú de **Opciones**, véase la imagen siguiente:

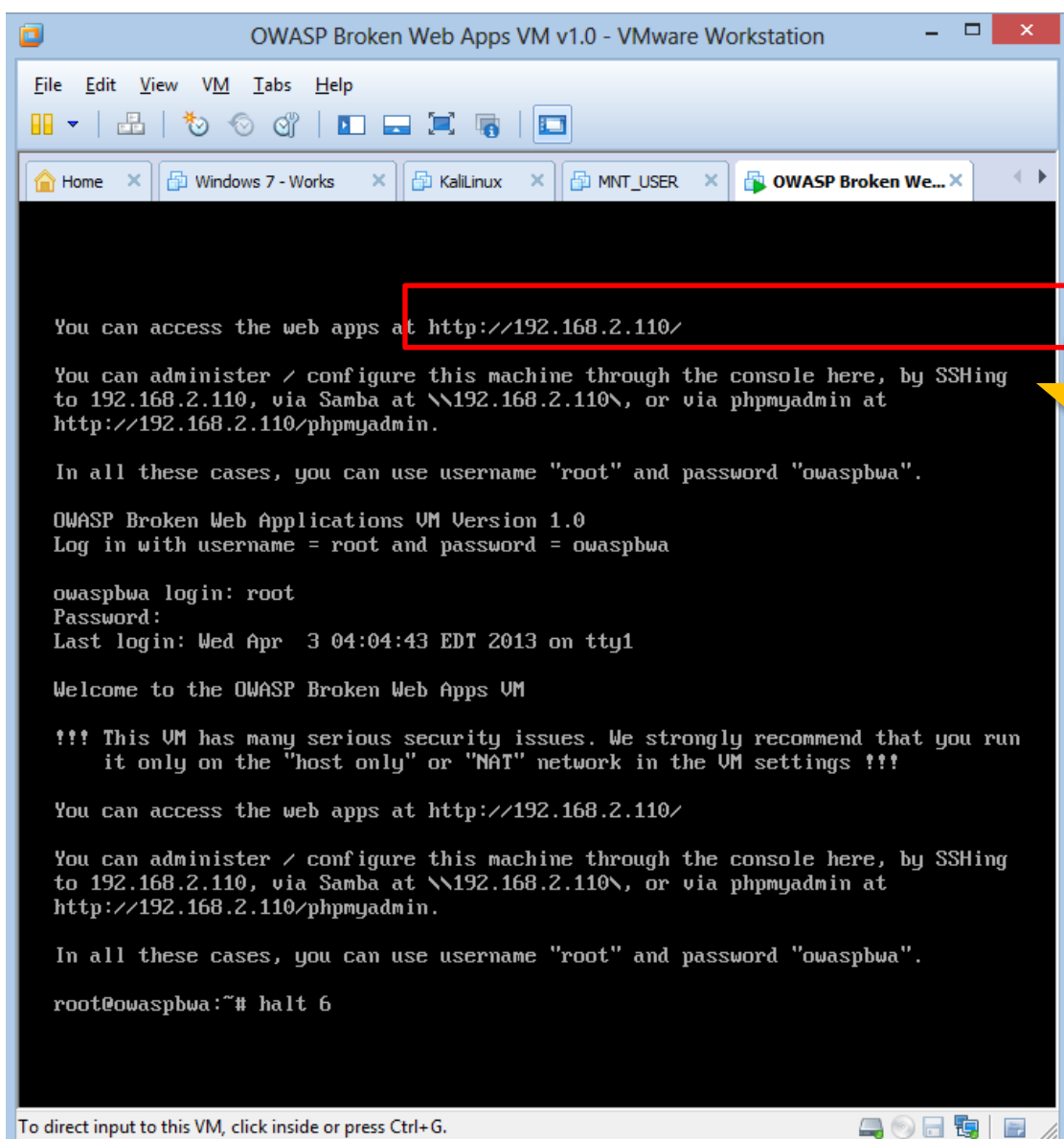


Preparación del escenario

Configuración del Entorno WEB Vulnerable

Abrir con el programa de VMware Player, o similar, el archivo **OWASP Broken Web Apps.vmx** proporcionado por el profesor.

Asegurarse que la **interfaz de red este configurada en modo "Bridged"** para que el sistema de DHCP del aula le proporcione una dirección IP válida de forma automática durante su inicio. Una vez confirmado este paso, inicie la maquina virtual correspondiente.



```

OWASP Broken Web Apps VM v1.0 - VMware Workstation
File Edit View VM Tabs Help
Home x Windows 7 - Works x KaliLinux x MNT_USER x OWASP Broken We... x
You can access the web apps at http://192.168.2.110/
You can administer / configure this machine through the console here, by SSHing
to 192.168.2.110, via Samba at \\192.168.2.110\, or via phpmyadmin at
http://192.168.2.110/phpmyadmin.
In all these cases, you can use username "root" and password "owaspbwa".
OWASP Broken Web Applications VM Version 1.0
Log in with username = root and password = owaspbwa
owaspbwa login: root
Password:
Last login: Wed Apr 3 04:04:43 EDT 2013 on tty1
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://192.168.2.110/
You can administer / configure this machine through the console here, by SSHing
to 192.168.2.110, via Samba at \\192.168.2.110\, or via phpmyadmin at
http://192.168.2.110/phpmyadmin.
In all these cases, you can use username "root" and password "owaspbwa".
root@owaspbwa:~# halt 6
To direct input to this VM, click inside or press Ctrl+G.
  
```

Anotar la dirección IP mostrada.

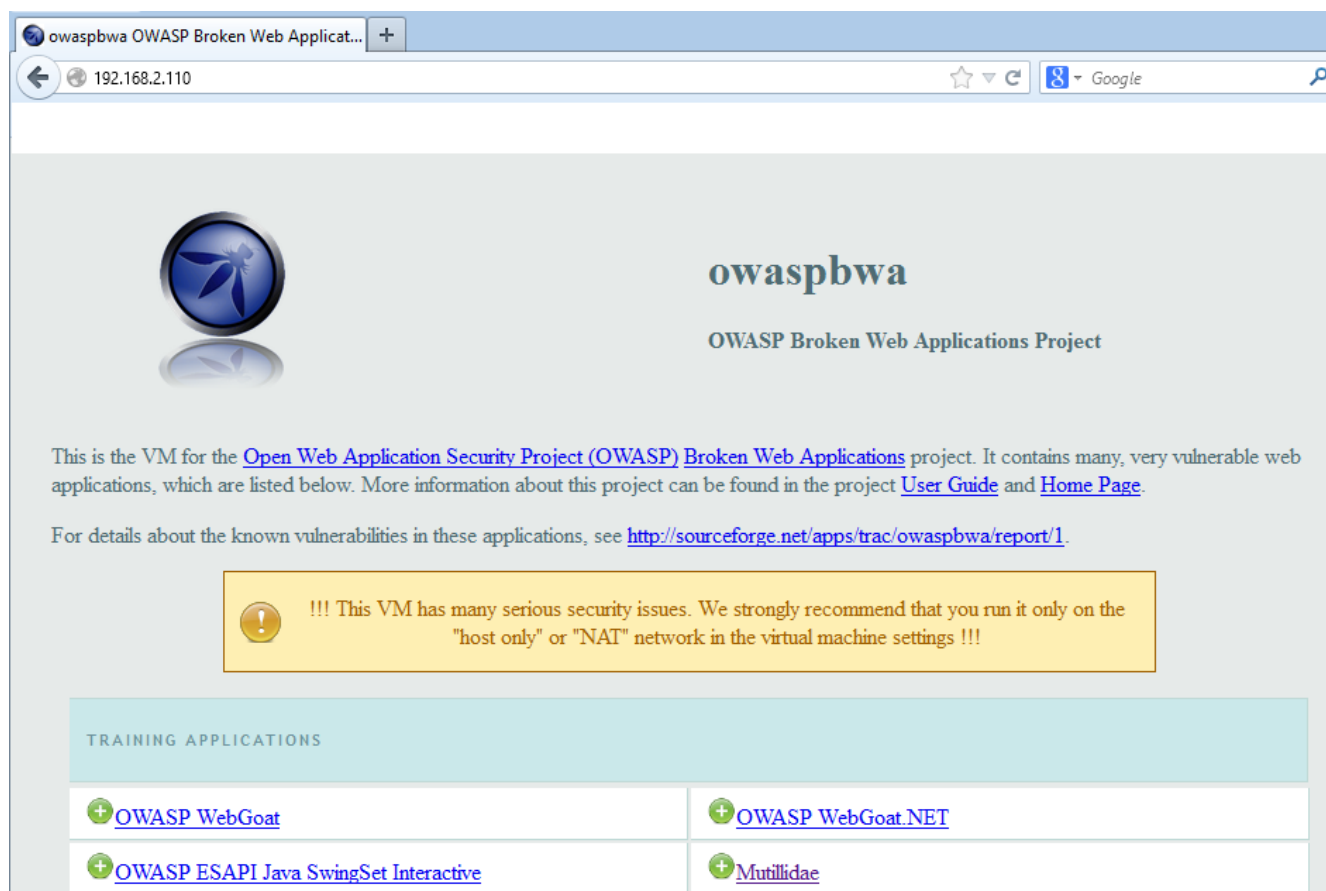
PRIMERA PARTE | Desarrollo de la práctica

En la **practica de hoy** se va a trabajar con el entorno WEB vulnerable proporcionado por OWASP, donde el alumno deberá poner en practica el conocimiento adquirido durante las clases teóricas, en particular el descubrimiento manual de vulnerabilidades del tipo XSS.

Ejercicio 1:

Abrir el navegador WEB (Mozilla Firefox) y teclear en la barra de navegación la dirección: <http://192.168.x.x> (donde la IP será la obtenida por la máquina virtual en ejecución).

Deberá mostrarse una página como la siguiente:

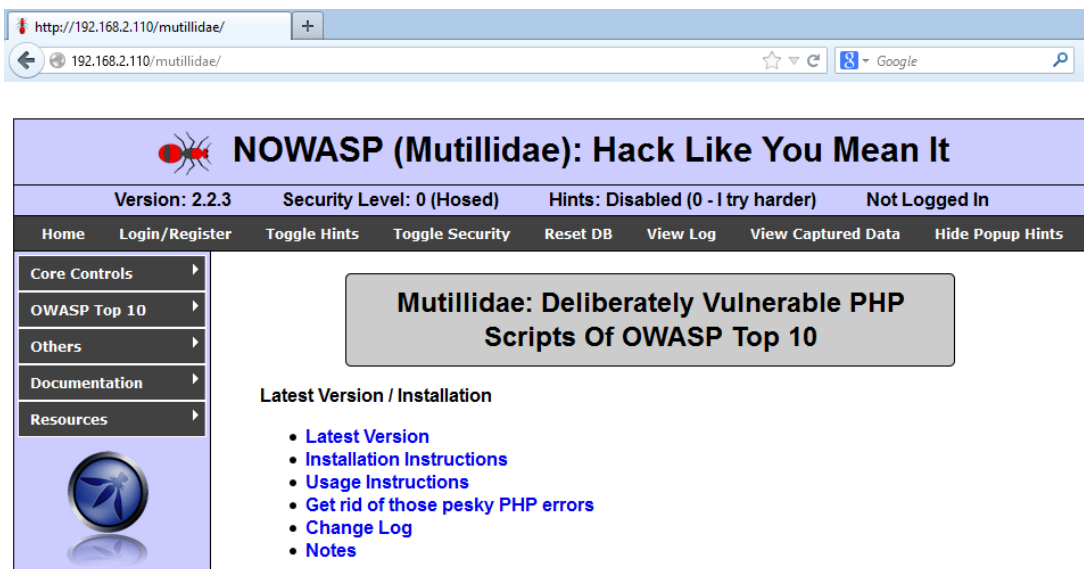



TRAINING APPLICATIONS	
+ OWASP WebGoat	+ OWASP WebGoat.NET
+ OWASP ESAPI Java SwingSet Interactive	+ Mutillidae

Acceder al entorno de entrenamiento "Multilliade".

Por favor - !! Queda totalmente prohibido atacar a otros grupos ii

PRIMERA PARTE | Tarea 1: Simple XSS



A continuación realiza las siguientes tareas:

Tarea 1: Detección de XSS (tipo Reflected) sobre un campo de texto

Ve al **Menú** OWASP Top 10 > A.2 Cross Site Scripting > Reflected First Order > DNS lookup.

Who would you like to do a DNS lookup on?

Enter IP or hostname

Hostname/IP

Tal y como se ha visto en la parte teórica ¿qué ocurre si se introduce?:



`<script>alert(1)</script>`

EJ1: Responde a las siguientes cuestiones:

1

1 . Muestra un mensaje por pantalla que ponga – **"MANDALORIAN 2021"**

2 . ¿Se pueden robar las cookies? Si la respuesta es SI, escribe el comando que muestre un mensaje en la pantalla con las cookies del navegador.

PRIMERA PARTE | Tarea 2: JSON Injection

EJ1: Responde a las siguientes cuestiones:

3 . Inyecta texto en la página WEB donde, con ayuda de código javascript, se muestre la plataforma del navegador, información del navegador, información de las cookies y datos de la página web (titulo,IP).

Todas las respuestas deben de estar correctamente razonadas, describiendo el comando utilizado, y explicando todo el proceso, incluyendo capturas de pantalla.



JAVASCRIPT: <http://norfipc.com/inf/javascript-lista-variables-funciones-usar-paginas-web.html>

Tarea 2: Inyección de XSS a través de la vulnerabilidad JSON Injection

La vulnerabilidad "JSON Injection" se produce cuando los desarrolladores de las páginas webs descuidan el control (filtrado) sobre los parámetros devueltos en una respuesta JSON desde el Servidor. Es en ese momento donde se produce la inyección de código, esto es debido a que "**piensan**" que el control de esos parámetros le corresponde al servidor.

Tal y como se ha visto en las clases teóricas, para detectar una vulnerabilidad del tipo "Code Injection", se deben de encontrar las vías de entrada (INPUT) de los parámetros, esto se puede hacer usando una simple navegador por ejemplo Firefox, y bastaría con examinar las líneas de código, donde en principio no se requieren herramientas especiales.

Sin embargo, existe una manera más eficiente de encontrar las vías de entrada (INPUT) de datos, esto es utilizando la interceptación del flujo WEB vía proxy (por ejemplo ZAP Proxy de OWASP).

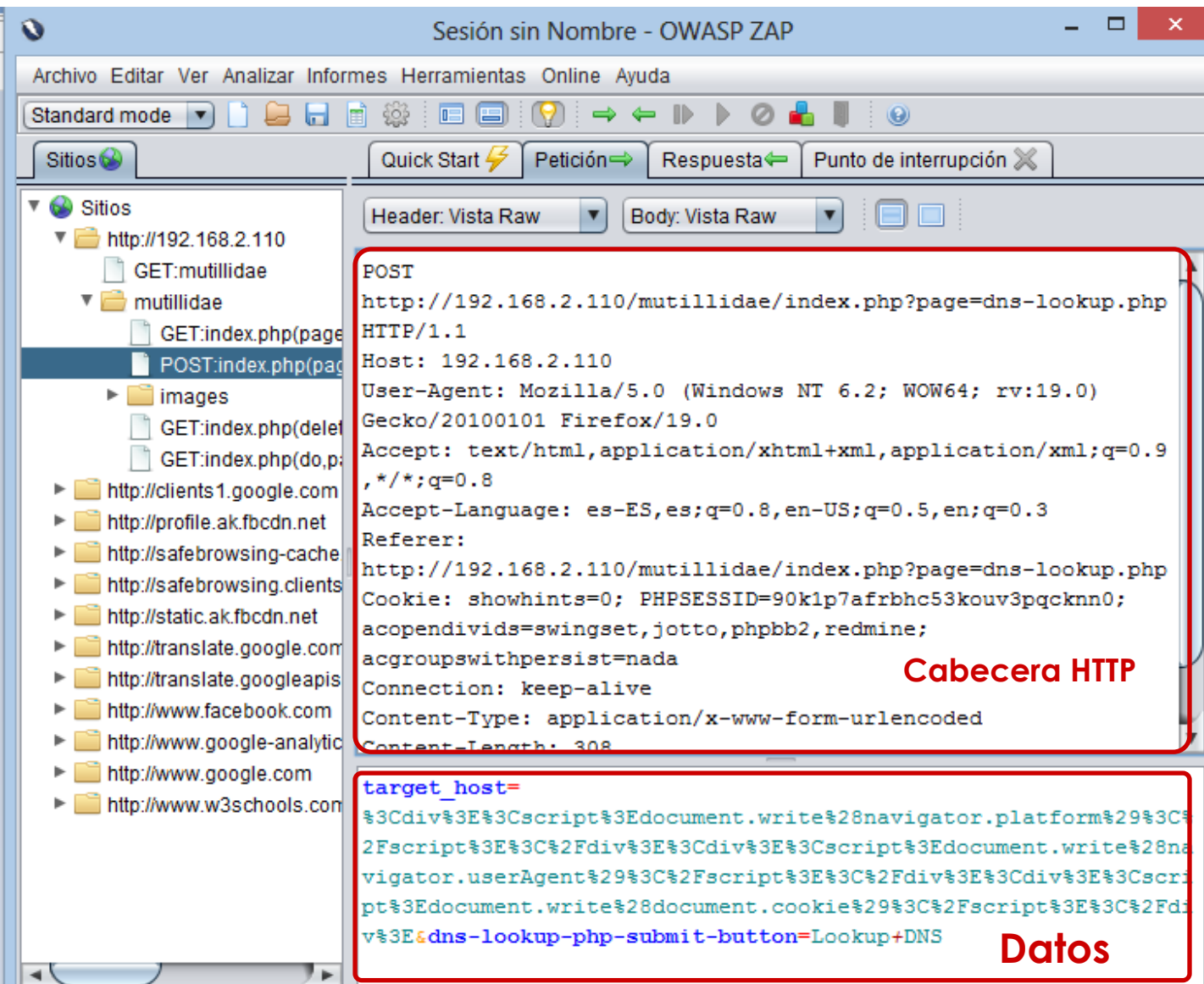
Un proxy de interceptación simple, fácil de usar es **Tamper Data**. Se trata de un plug-in para Firefox. Es rápido pero limitado en capacidad. Aún así, es un gran complemento para la realización de pruebas sencillas y rápidas.

PRIMERA PARTE | Tarea 2: JSON Injection

Para el ejercicio de hoy, se va a utilizar la herramienta gratuita Open-Source ZAP Proxy, que se encuentra instalada y lista para ser usada, tal y como se ha realizado durante la preparación del escenario al inicio de esta practica.

Ir a la herramienta ZAP Proxy, se comprobará que el proxy ha estado registrando todas las acciones realizadas en el entorno WEB vulnerable y/o cualquier otra página web que se haya estado visitando con el navegador Mozilla Firefox.

Si se observa la imagen inferior, se podrán identificar los parámetros de entrada de datos y el método utilizado para la comunicación con el servidor (POST).



The screenshot shows the OWASP ZAP Proxy interface. The left pane displays the site tree with the selected site being `http://192.168.2.110`. The right pane shows the details of a POST request to `http://192.168.2.110/mutillidae/index.php?page=dns-lookup.php`. The request is highlighted with a red box, and the body is also highlighted with a red box.

Cabecera HTTP

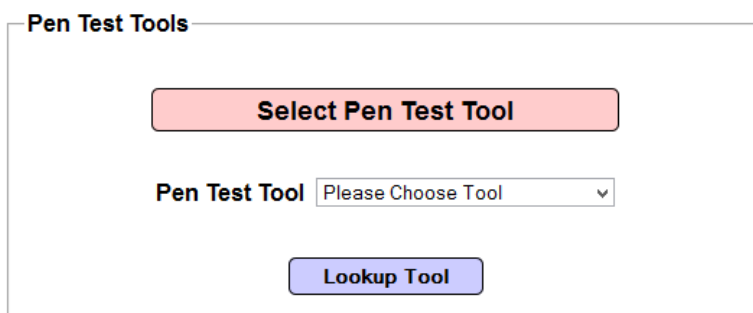
```
POST
http://192.168.2.110/mutillidae/index.php?page=dns-lookup.php
HTTP/1.1
Host: 192.168.2.110
User-Agent: Mozilla/5.0 (Windows NT 6.2; WOW64; rv:19.0)
Gecko/20100101 Firefox/19.0
Accept: text/html,application/xhtml+xml,application/xml;q=0.9
, */*;q=0.8
Accept-Language: es-ES, es;q=0.8, en-US;q=0.5, en;q=0.3
Referer:
http://192.168.2.110/mutillidae/index.php?page=dns-lookup.php
Cookie: showhints=0; PHPSESSID=90k1p7afrbhc53kouv3pqcknn0;
acopendivids=swingset, jotto, phpbb2, redmine;
acgroupswithpersist=nada
Connection: keep-alive
Content-Type: application/x-www-form-urlencoded
Content-Length: 308
```

Datos

```
target_host=
%3Cdiv%3E%3Cscript%3Edocument.write%28navigator.platform%29%3C%
2Fscript%3E%3C%2Fdiv%3E%3Cdiv%3E%3Cscript%3Edocument.write%28na
vigator.userAgent%29%3C%2Fscript%3E%3C%2Fdiv%3E%3Cdiv%3E%3Cscr
pt%3Edocument.write%28document.cookie%29%3C%2Fscript%3E%3C%2Fdi
v%3E%3Edns-lookup-php-submit-button=Lookup+DNS
```

PRIMERA PARTE | Tarea 2: JSON Injection

Para realizar esta tarea, Ve al **Menú OWASP Top 10** > A.2 Cross Site Scripting > Against JSON> Pen Test Tool lookup.



The screenshot shows a web interface titled "Pen Test Tools". It contains a red button labeled "Select Pen Test Tool". Below this is a label "Pen Test Tool" followed by a dropdown menu with the text "Please Choose Tool" and a downward arrow. At the bottom is a blue button labeled "Lookup Tool".

Esta página se encuentra preparada para soportar las respuestas JSON, el alumno deberá identificar los campos JSON e intentar "probar" que efectivamente la página es vulnerable a ese tipo de ataque.

Para encontrar los puntos JSON, se deberá añadir un "punto" de parada (intercepción) en la página WEB, esto se puede conseguir con la ayuda de la herramienta ZAP Proxy.

EJ 2: Responde a las siguientes cuestiones

1 . Examina detenidamente, con ayuda del Proxy, que ocurre cuando se seleccionan diferentes herramientas en la pestaña de elección de "Pen Test Tool" de la página WEB. A) Identifica las variables de entrada de datos. B) Encuentra y copia el código fuente responsable de su funcionamiento.

2 . Realiza una prueba sencilla de la vulnerabilidad inyectando `<script>alert(1)</script>` en una de sus variables. Describe ¿Qué ocurre?.

Para explotar con éxito la vulnerabilidad JSON se debe de encontrar el carácter que produce el salto (escape) de JSON a código Javascript. Para ello se debe de encontrar los caracteres de escape adecuados para introducir el código JavaScript sin romper la sintaxis de JSON original, y no provocar un fallo. **!! Es importante terminar la sentencia de la misma forma con la que empieza !!**

PRIMERA PARTE | Tarea 2: JSON Injection

En la sintaxis JSON cualquier punto y coma (;) puede provocar un error de sentencia, es por ello que se debe de utilizar la **técnica de encoding**, como por ejemplo: **; = %3b**. Se recomienda hacer la prueba sin encoding, y con encoding.

Observar cual es la forma de finalizar la sentencia correctamente y recordar que se puede utilizar el comando // (para comentar código, lo que obviará el resto)



Responda a la siguientes cuestiones:

3 . ¿Qué ocurre si se introduce el comando de mostrar un mensaje de alerta con con %3b en lugar de (;)?

4 . Con ayuda de la inyección XSS muestra un mensaje de dialogo con el texto: **M4_HACKING_MSTIC_2021.**

5 . Escribe y haz la prueba, donde el código javascript a inyectar tiene que tratar de robar la cookie utilizando la función PHP disponible en el servidor WEB denominada **capture-data.php** , cuya URL es: <http://localhost/multillidae/capture-data.php?cookie=>

Todas las respuestas se debe entregar debidamente razonadas, describiendo todos los pasos realizados y aportando capturas de pantalla del proceso.

SEGUNDA PARTE | Tarea 1: SQL Injection

En la segunda parte de la practica, el alumno intentará detectar la existencia de vulnerabilidades del tipo Inyección SQL en el entorno vulnerable OWASP.

Tarea 1: SQL Injection extracción de Información de una Base de Datos.

Con la información proporcionada en las clases teóricas, el alumno debe ser capaz de contestar adecuadamente a las cuestiones que se le plantean.

Acceda al **Menú OWASP Top 10** > A.1 Injection > SQLi Extract Data > User Info

Please enter username and password to view account details

Name

Password

[View Account Details](#)

Dont have an account? [Please register here](#)

Responde a las siguientes cuestiones:

Para cada una de las respuesta el alumno debe describir todo el proceso, incluyendo los comandos ejecutados y los resultados obtenidos. Se valorará la aportación de capturas de pantalla del proceso ejecutado.

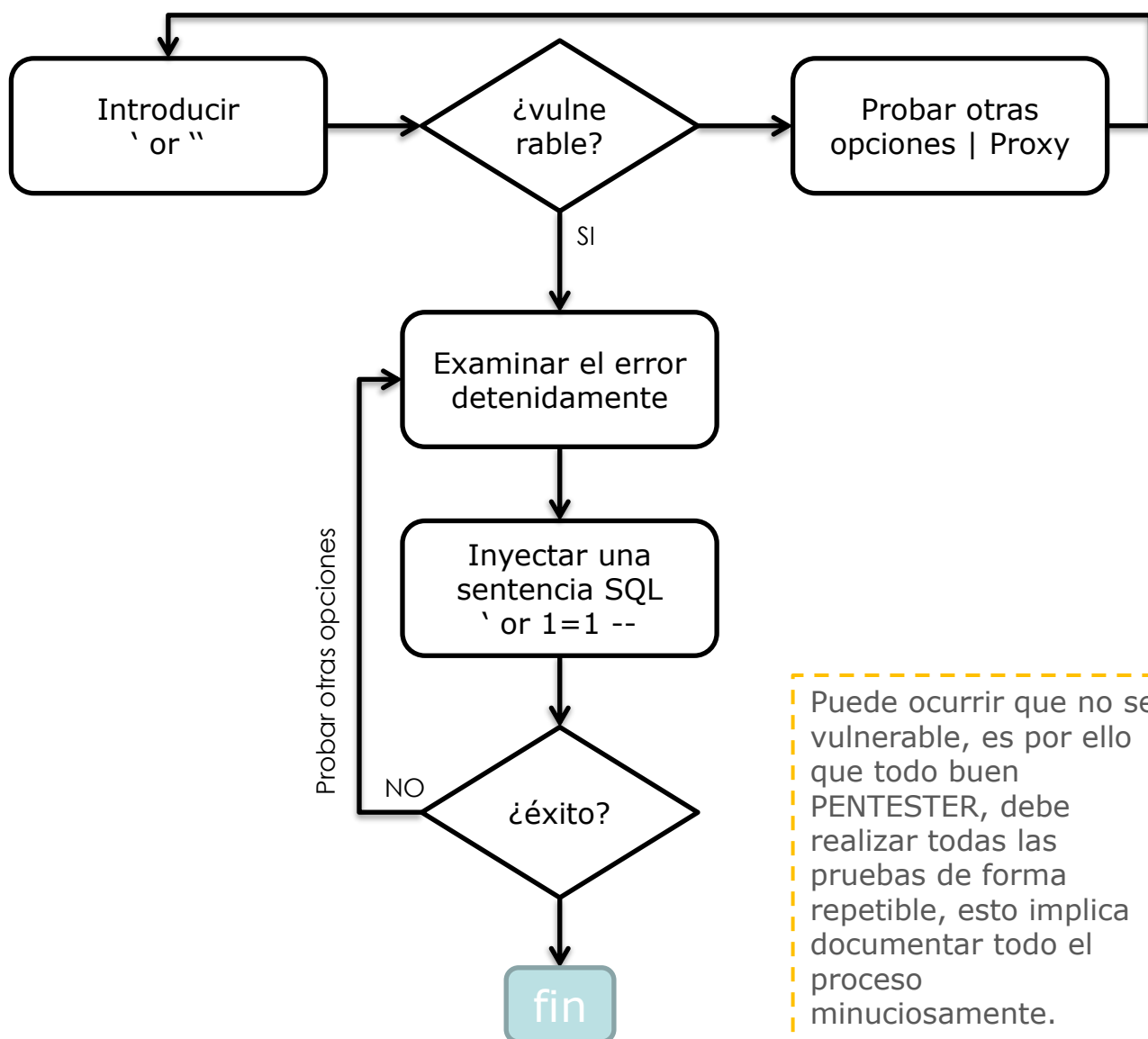
1 . ¿Es vulnerable a SQL Injection la siguiente URL? En caso afirmativo, Escribe la sentencia y el resultado de la Inyección utilizada.

2 . ¿Podrías lograr el mismo resultado que en la pregunta 1) con el comando UNION? En caso afirmativo. Describa la sentencia utilizada y el resultado obtenido.

SEGUNDA PARTE | Tarea 2: SQL Injection

Tarea 2: SQL Injection estructura de una Base de Datos.

Los ataques de SQL Injection necesitan un proceso de “reconocimiento” del Sistema de base de datos que se desea evaluar. Por lo tanto el Pentester deberá realizar un proceso de “prueba-error” que le ayude a ir desgranando la estructura de la Base de Datos, para posteriormente explotar con éxito la vulnerabilidad detectada.



SEGUNDA PARTE | Tarea 2: SQL Injection

Acceda al **Menú OWASP Top 10** > A.1 Injection > SQLi Extract Data > User Info.

Please enter username and password
to view account details

Name

Password

View Account Details

Dont have an account? [Please register here](#)

El proceso habitual para descubrir la estructura sería utilizando la sentencia UNION, y las variables del Sistema "null" y de relleno "1". Por ejemplo:



` union select null --



` union select null,null --

⋮

Se va introduciendo los comandos hasta obtener "éxito", es decir, no obtener un mensaje de error, sino resultados. Una vez obtenidos los resultado, se puede ir sustituyendo **null** por **1**, para identificar cada campo de la tabla en la sentencia SQL.



` union select null,1,null --

SEGUNDA PARTE | Tarea 2: SQL Injection

Responda a la siguientes cuestiones:

Para cada una de las respuesta el alumno debe describir todo el proceso, incluyendo los comandos ejecutados y los resultados obtenidos. Se valorará la aportación de capturas de pantalla del proceso ejecutado.

1 . Utilizando la técnica SQL Injection utilizando UNION, determine la estructura de la Base de Datos en la página de consulta de información de usuarios.

2 . ¿Puede añadir (insertar) nueva información en la tabla? En caso afirmativo, introduzca un usuario "**mrtest**" cuyo password sea "**2021**".

3 . Esta técnica le permite interactuar (inyectar comandos) en el Sistema de Base de Datos.

- a) Prueba a insertar version(), en uno de los campos, por ejemplo: username. Describa el proceso que ha seguido, incluyendo el resultado obtenido.
- b) Busca las funciones de MySQL, y obtén información del usuario de la base de datos, nombre de la base de datos, etc.



SEGUNDA PARTE | Tarea 3: SQL Injection

Tarea 3: SQL Injection Bypass

Acceda al **Menú OWASP Top 10** > A.1 Injection > SQLi Bypass Authentication > Login.

Please sign-in

Name


Password


Dont have an account? [Please register here](#)

Responde de forma razona al siguiente ejercicio:

- El alumno deberá saltarse el control de autenticación de la página WEB utilizando el conocimiento (SQL Injection) adquirido por el alumno tanto en las clases teóricas como en los ejercicios prácticos anteriores.

Objetivo es conseguir el acceso de usuario registrado realizando un "Bypass" al control de autenticación. El resultado deberá ser similar a:




NOWASP (Mutillidae): Hack Like You Mean It

Logged In Admin: admin

Version: 2.2.3
Security Level: 0 (Hosed)
Hints: Disabled (0 - I try harder) (Monkey!)

Home
Logout
Toggle Hints
Toggle Security
Reset DB
View Log
View Captured Data
Show Popup Hints

Core Controls ▶

OWASP Top 10 ▶

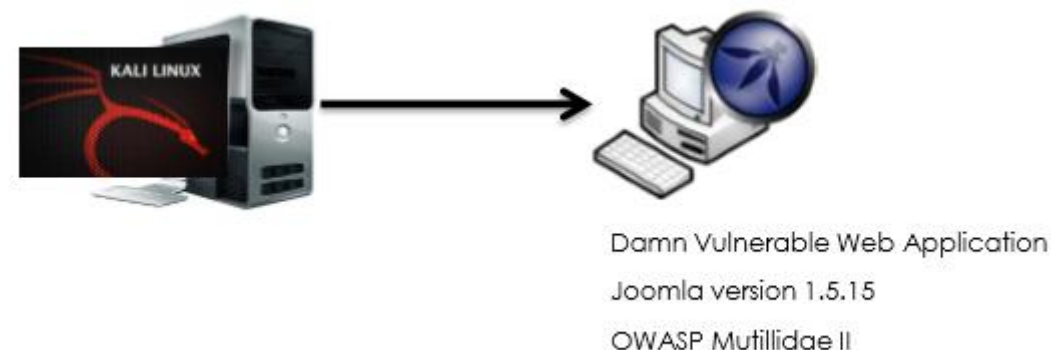
Others ▶

Mutillidae: Deliberately Vulnerable PHP Scripts Of OWASP Top 10

Tarea 4 – Análisis de Vulnerabilidades

Objetivo de esta tarea es utilizar algunas de las herramientas de análisis de vulnerabilidades WEB incluidas en la distribución KaliLinux.

Escenario



Tarea

Para la realización de este ejercicio se requiere trabajar en pareja, de modo que se requieren dos equipos (o maquinas virtuales) , el primero con la distribución KaliLinux y el segundo con la maquina OWASP BWA iniciada.

Ejercicio1: Realizar un análisis de la seguridad de la URL (Mutillidae II) incluido en OWASP BWA con la herramienta **nitko**. Anotar , estudiar y comentar los resultados obtenidos.

Ejercicio2: Realizar un análisis de la seguridad de la URL (Joomla 1.5.15) incluido en OWASP BWA con la herramienta **WhatWEB**. Anotar , estudiar y comentar los resultados obtenidos.

Ejercicio3: Realizar un análisis de la seguridad de la URL (DVWA) incluido en OWASP BWA con la herramienta **w3af**. Anotar , estudiar y comentar los resultados obtenidos.