TABLA DE CONTENIDOS

PAQUETE: UMA-HW-15-(In)seguridad_Clientes-Demos

SOBRE LA LICENCIA

SOBRE RESPONSABILIDADES

Please, read this carefully:

This package consists of:

* A vulnerable application written in PHP / HTML

* A series of proofs of concept that allow experimenting with the application vulnerabilities. Please note that some of these proofs of concept have their own vulnerabilities or generate malware.

* Notes and lists of URLs to use when running proofs of concept

Note that:

* Under no circumstances should you install an application that has or may have vulnerabilities on a computer that is used for production purpouses or on a system whose security must be protected.

* The use of hacking techniques may constitute an offense under the legislation applicable in each case. Under no circumstances should you use these techniques without the express consent of the legally valid holders of the hardware and software equipment that could be affected.

* Under no circumstances should you perform any activity that would constitute a breach of applicable law.

Anyone not complying with this text will do so at their own risk and must bear the consequences.

Por favor, lea detenidamente lo siguiente:

Este paquete consta de:

* Una aplicación vulnerable escrita en PHP / HTML

* Una serie de pruebas de concepto que permiten experimentar con las vulnerabilidades de dicha aplicación. Algunas de estas pruebas contienen sus propias vulnerabilidades o generan malware

* Notas y listas de URLs a utilizar en la ejecución de las pruebas de concepto

Ten en cuenta que:

* Bajo ninguna circunstancia se debe instalar una aplicación que tenga o pudiera tener vulnerabilidades en un equipo usado para explotación o en un sistema cuya seguridad se desee salvaguardar.

* El uso de técnicas de hacking puede constituir un delito según la normativa aplicable en cada caso. Bajo ninguna circunstancia se debe utilizar este tipo de técnicas sin el consentimiento expreso y legalmente válido de las personas titulares de los equipos físicos y lógicos que pudieran verse afectados.

* Bajo ninguna circunstancia se debe realizar ninguna actividad que pudiera suponer un incumplimiento de la normativa aplicable.

Cualquier persona que no se atuviera a lo expresado anteriormente lo hará bajo su propia responsabilidad y deberá atenerse a las consecuencias que pudieran tener sus actos.

INTRODUCCIÓN

Este paquete contiene el código fuente y los recursos utilizados para la charla "(In)seguridad en componentes clientes de aplicaciones web", impartida en el marco de la UMA Hackers Week el 24 de marzo de 2015.

A las demos presentadas durante la ponencia se añaden otras que, habiendo sido diseñadas para ésta, terminaron no siendo utilizadas por motivos de tiempo.

Este material se publica para fines académicos y de concienciación en materia de Seguridad de la Información. Aunque ya se haya mencionado anteriormente, debe señalarse aquí que la realización de cualquier tipo de ataques puede suponer una violación de la normativa vigente en el caso de que no se disponga del consentimiento expreso (y, a efectos prácticos, demostrable) de las personas u organizaciones titulares de los mismos y, cuado proceda, de aquellas que, de una u otra forma, pudieran verse afectadas.

CONTEXTO DE LAS PRUEBAS

Para el desarrollo de las demos se estableció el siguiente entorno:

- La organización víctima de los ataques tiene una aplicación propia de mensajería, escrita en PHP y que utiliza una base de datos MySQL.
- Esta aplicación está alojada en un servidor llamado webserver.example.com
- Esta organización tiene varias cuentas del sistema de mensajería: jefazo, usuario1, usuario2, operador,... Una de ellas, "malicioso" pertenece a una persona dispuesta a realizar ataques contra la organizción.
- Alguien quiere realizar ataques contra la organización víctima. Para ello controla un servidor web llamado malicioso.uhw
- Este servidor responde también al nombre webserver.example.com.malicioso.uhw

INSTALACIÓN DEL ENTORNO DE SERVIDOR DE PRUEBAS

Para realizar las demos objeto de este documento se utilizó un entorno de pruebas con un equipo servidor que tenía insalado es software XAMPP 5.63, que puede ser obtenido de:

https://www.apachefriends.org

Posiblemente funcione en cualquier otro servidor LAMP o WAMP suficientemente actualizado. Las componentes del paquete XAMP utilizadas fueron:

- Apache 2.4.10
- MySQL 5.6.21
- PHP 5.6.3,
- phpMyAdmin 4.2.11,

Para la edición de ficheros de código fuente se recomienda usar un programa con características avanzadas, como Notepad++

http://notepad-plus-plus.org/

Para simular la existencia de los distintos servidores webse puede modificar el fichero HOSTS de la máquina en que se vaya

a realizar las pruebas. La ubicación típica de este fichero en sistemas Windows es:

C:\Windows\System32\drivers\etc\hosts

... y, para sisetemas *NIX:

/etc/hosts

A este fichero se añadirán tres línas del tipo

<direccion_IP><nombre_de_equipo>

Donde

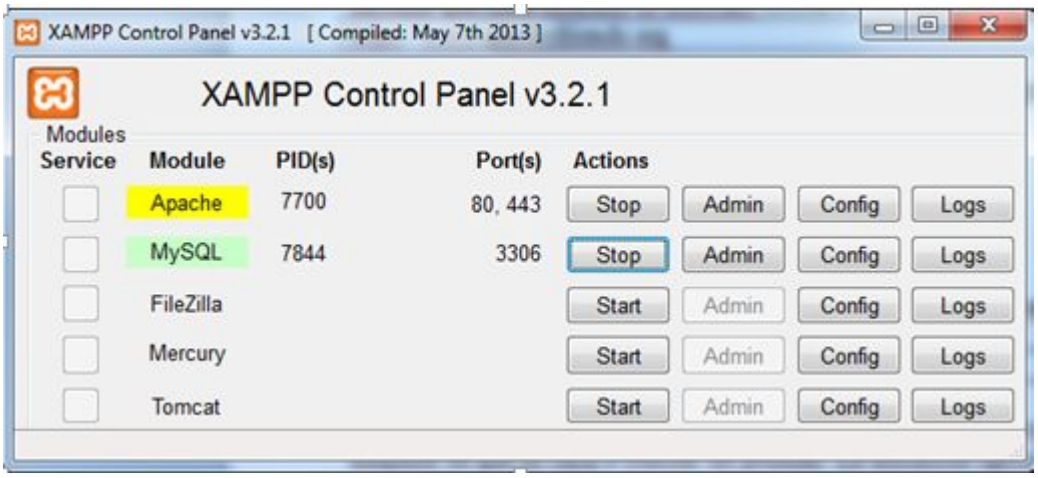
<direccion_IP>	Se corresponde con la dirección IP del equipo que tiene instalado el servidor WAMP o LAMP
<nombre_de_equipo>	Se corresponde con cada uno de los nombres de servidores mencionados en el apartado “CONTEXTO DE LAS PRUEBAS”

Por ejemplo, si el servidor LAMP o WAMP tuviera la IP 10.10.10.10, las líneas a añadir serían

10.10.10.10	webserver.example.com
10.10.10.10	malicioso.uhw
10.10.10.10	webserver.example.com.malicioso.uhw

Para editar el fichero HOSTS necesitarás permisos de administración del sistema.

Para el funcionamiento de las aplicaciones es necesario que los servicios “Apache” y “MySQL” estén puestos en marcha:

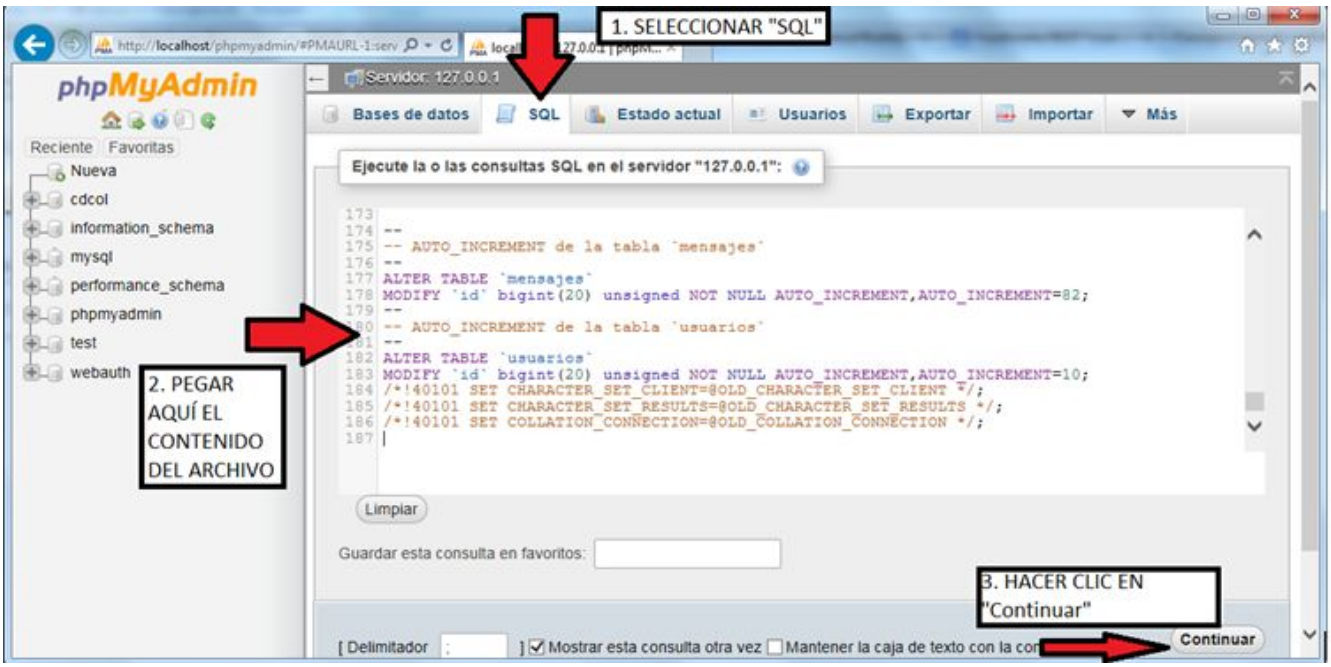


En la carpeta raíz de documentos del servidor web (“C:\xampp\htdocs” en XAMP para Windows) se debe copiar las carpetas “aplicacion” y “ataques” contenidas en el paquete “UMA-HW-15-(In)seguridad_Clientes-Demos”.

En ambas carpetas existe un fichero con extensión SQL:

Carpeta	Fichero
aplicacion	mensajeria.sql
ataques	ataques.sql

Utilizando PHPMyAdmin o cualquier otra interfaz que permita la ejecución de código SQL en el servidor LAMP o WAMP se debe ejecutar el contenido de ambos archivos para crear las respectivas bases de datos.



Se puede acceder a PHPMyAdmin mediante el botón “Admin” de la fila correspondiente a “MySQL” del panel de control de XAMPP.

CONFIGURACIÓN DEL ENTORNO CLIENTE DE PRUEBAS

Aunque la mayor parte de las pruebas son independientes del sistema operativo utilizado en el puesto cliente, aquellas que se refieren a Internet Explorer requerirán el uso de Windows.

Para las demos se utilizó un equipo con Windows 7, Google Chrome y Mozilla Firefox, todo ello actualizado a la fecha en que se impartió la charla.

Para evitar cualquier tipo de interferencia con configuraciones previas de los navegadores, se creó un perfil nuevo para la charla.

Para facilitar las pruebas es conveniente copiar el directorio “DEMOS” en el escritorio de la cuenta utilizada para realizar las demos.

IMPORTANTE:

La primera vez que se entre a la aplicación con el navegador Firefox, éste preguntará si se desea guardar la contraseña. Para poder realizar correctamente posteriores pruebas, responder en esta primera ocasión que “SÍ” se guarden las credenciales.

Si más adelante preguntara si se desea almacenar otras credenciales, se puede responder que “NO” las guarde.

GUIÓN DE LAS DEMOS

DEMO 1 - CSRF

Utilizando Internet Explorer, el Administrador inicia sesión en la aplicación. La URL de acceso es

http://webserver.example.com/aplicacion/login.php

Y las credenciales

Nombre	admin
Clave	passadmin

En el menú que le aparecerá, elige la opción “Administrar” y después selecciona “Cuentas de adminsitración”. Aparece sólo una cuenta de administración del sistema.

Hace clic en “Menú” (parte central superior de la ventana) y después en “Bandeja de Entrada”. Le aparecen dos mensajes-

Hace clic sobre el mensaje cuyo asunto es “Errores de acceso a Internet 01”. En la parte de la derecha se mostrará el mensaje. Entre otras cosas, tiene un enlace a

http://malicioso.uhw/ataques/demo%2001/

El administrador hace clic en el enlace y le aparece una página de error..

Para ver los efectos de este ataque, se puede volver a la página de gestión de “Cuentas de administración”. Se habrá creado un nuevo usuario llamado “adminmalicioso”.

DEMO 2 - ROBO DE CONTRASEÑAS

Se decidió no realizar esta demo durante la charla por condicionantes de tiempo.

Se cierra la sesión de “admin” utilizando el botón “Cerrar sesión” de la parte superior dercha de la ventana. Se sigue usando Internet Explorer.

El atacante visita, utilizando Google Chrome (no es necesaria ninguna autenticación para verla)

http://malicioso.uhw/ataques/demo%2002/lista_credenciales.php

Se abre el fichero “demo 02.htm” ubicado en la carpeta “DEMOS”. Éste simula un correo electrónico enviado al administrador en el que se le invita a hacer clic en un enlace que lleva a

http://malicioso.uhw/ataques/demo%2002/

Sale una página de error. Pero si el atacante vuelve a visitar usando Google Chrome

http://malicioso.uhw/ataques/demo%2002/lista_credenciales.php

... le aparecen dos credenciales

NOTA: Si se desea borrar la lista de credenciales robadas, se puede utilizar la URL: http://malicioso.uhw/ataques/demo%2002/lista_credenciales.php?borra=1

En esta demo se hace un ataque de diccionario sobre un conjunto de cuentas conocidas con una serie de posibles contraseñas.

Aprovechando la vulnerabilidad de CSRF se realizan los intentos de login. Además se aprovecha un fallo en el mecanismo de redirección que permite que, en caso de login exitoso, se redirija al usuario a una página controlada por el atacante.

El fallo radica en que “login.php” acepta un parámetro GET llamado “url” que usa para determinar la dirección a la que redirigir en caso de éxito. Este parámetro lleva la ruta del fichero a partir de la raíz de publicaciones del servidor y “login.php” le pone delante “http://webserver.example.com”.

De este modo, si se asigna a “url” un valor del tipo

url=.malicioso.uhw/ataques/demo%2002/roba_credenciales.php?nombre=usuario1&pwd=12345678

... la redirección se producirá hacia el servidor

http://webserver.example.com.malicioso.uhw

...que es un equipo del dominio controlado por el atacante

DEMO 3 - DETECCIÓN DE INICIO Y CIERRE DE SESIÓN

Se sigue usando Internet Explorer.

Incluso en ausencia de la vulnerabilidad de redirección a servidores ajenos, sigue siendo posible determinar si el usuario tiene no abierta la sesión.

Si se abre el fichero “demo 03 - Monitor.htm” de la carpeta “DEMOS” se encontrará un ejemplo del uso de anclas dentro de documentos HTML.

El enlace “Seguir con la demo” lleva a

```
http://malicioso.uhw/ataques/demo%2003/
```

Esta página monitoriza en tiempo real los logins y logouts del usuario. Para ello trata de detectar si al acceder a

```
http://webserver.example.com/aplicacion/principal.php
```

... se produce redirección a

```
http://webserver.example.com/aplicacion/login.php
```

Utiliza en esta tarea la detección del evento “load” de un IFRAME

DEMO 4 - CLICKJACKING 1

Se sigue usando Internet Explorer. Da igual si se ha iniciado o no sesión.

Se abre el fichero “demo 04 - Fondo.htm” de la carpeta “DEMOS”. Simula un mensaje en el que se avisa de que hay que cambiar el fondo de escritorio.

El usuario visita la página, hace clic con el botón secundario del ratón sobre la imagen y la establece como fondo.

Pero no es la imagen que esperaba.

DEMO 5 - CLICKJACKING 2

Se sigue usando Internet Explorer. Si es necesario, se inicia sesión como “admin”.

Se visita la Bandeja de Entrada y se hace clic en el mensaje “Gran Sorteo 05”. Se rellenan los datos del mensaje. El admin hace clic sobre el enlace del mensaje.

Eso le lleva a una página en la que, para participar en un sorteo, se le pide que resuelva un CAPTCHA e introduzca su dirección de correo.

Pero en realidad está rellenando el formulario que da de alta administradores de la aplicación de mensajeria.

DEMO 6 - XSS REFLEJADO

Se pasa a utilizar Mozilla Firefox.

Se inicia sesión en la aplicación como “usuario1”.

Nombre	usuario1
Clave	12345678

Si es la primera vez, preguntará si se desea almacenar las credenciales introducidas. Respondemos que SÍ.

Una vez dicho que SÍ, si en posteriores ocasiones se repite la pregunta, responderemos que NO.

En otra pestaña abrimos la URL

http://webserver.example.com/aplicacion/hola_mundo.php

La página implementa el típico “Hola mundo” que suele ser el primer programa que se ve cuando se aprende un nuevo lenguaje de programación.

Con

http://webserver.example.com/aplicacion/hola_mundo.php?nombre=Prueba

... la aplicación muestra el mensaje “Hola, Prueba”. Pero se puede insertar también código HTML. Como en

http://webserver.example.com/aplicacion/hola_mundo.php?nombre=<u>Prueba</u>

Esta vez el texto aparece subrayado.

Haciendo clic en los botones puede verse el código fuente PHP utilizado.

Para mayor comodidad, los botones “MOSTRAR CASO 1”, “MOSTRAR CASO 2” y “MOSTRAR CASO 3” permiten mostrar u ocultar unas URLs a probar.

MOSTRAR CASO 1	Mensaje fraudulento insertando un IFRAME
MOSTRAR CASO 2	En lugar de un mensaje, se introduce malware
MOSTRAR CASO 3	Se ejecuta código JavaScript que accede a la cookie de sesión de la aplicación

NOTA:

Cuidado con el antivirus que pueda tenerse instalado en el servidor web. En la carpeta “ataques/demo 06” hay un fichero v.pdf que tiene en su interior malware.

Para que no fuera detectado por Microsoft Security Essentials, el antivirus del equipo usado para la presentación, se modificó ligeramente el archivo, insertándole un texto al inicio y manteniendo el contenido tal cual.

El script virus.php se limita a eliminar la cadena insertada al inicio del fichero.

Se pasa a utilizar Internet Explorer.

Se visita la URL

http://webserver.example.com/aplicacion/hola_mundo.php

Se prueba la URL del “CASO 3” y falla. IE tiene un filtro anti-XSS.

Haciendo clic sobre el botón “Protección” aparece una URL del servidor webserver.example.com que utiliza scripts para protegerse de las técnicas de Clickjacking.

Haciendo clic en “Intento 1” se tiene una URL del servidor malicioso que trata de colocar la URL anterior en un IFRAME”. Pero no puede. El navegador es redirigido.

Utilizando Notepad++ se codifica como parámetro de URL el contenido del script que realiza la protección (incluyendo las etiquetas “<script>” y “</script>”). Si se añade eso en un parámetro GET a la URL que se protege con el script, el filtro anti-XSS creerá que está ante un caso de inyección de contenidos y bloqueará la ejecución del script

Haciendo clic en “Intento 2” se tiene una URL que implementa el ataque anterior.

Se pasa a utilizar Mozilla Firefox.

Si es necesario, se vuelve a visitar

http://webserver.example.com/aplicacion/hola_mundo.php

Se hace click en “Cookies / Firefox”. Aparece una nueva URL y, en otra línea, un código en JavaScript. La visitamos.

El codigo Javascript inyectado fuerza una cookie de sesión en la víctima. Si intenta acceder a cualquier componente de la aplicación, su sesión se habrá cerrado. Se vuelve a iniciar sesión.

El usuario malcioso usa Google Chrome para acceder a la aplicación con las credenciales

Nombre	malicioso
Clave	passmalicioso

En la parte superior izquierda de la pantalla puede verse que ha iniciado sesión como “El usuario peligroso”.

Ahora pulsa la tecla F12 para activar las herramientas de desarrollo y seleccion la opción “Console” o “Consola”. Le aparece un prompt pidiéndole instrucciones. Se introduce ahí el código JavaScript que apareció al hacer clic en el botón “Cookies / Firefox” (comienza por “document.cookie =”) y se pulsa Intro. Esto modifica la cookie de sesión.

Ahora, el usuario malicioso hace clic en el enlace “Menú” de la parte superior. Está trabajando con la sesión del “usuario1” y tiene acceso a todos sus datos y funcionalidades. Puede verse en la parte superior derecha de la página.

Otra opción para el atacante:

Con la cookie forzada, el atacante, usando Chrome, cierra sesión y la abre con una cuenta que no usa:

Nombre	temp
Clave	temp

Ahora la sesión corresponde a esta cuenta. El usuario1 hace clic en “Menú” y, como puede comprobarse mirando la parte superior derecha de la ventana, está trabajando con la cuenta proporcionada por el atacante. Si no se diera cuenta, estaría dejando mensajes y otros datos en una cuenta accesible para el atacante.

El usuario, utilizando Mozilla Firefox, cierra sesión y vuelve a abrirla como “usuario1”.

DEMO 7 - XSS Almacenado

Se sigue usando Firefox

Visitar

```
http://webserver.example.com/ataques/demo%2002/lista_credenciales.php?borra=1
```

... para borrar la lista de credenciales robadas.

El usuario1 va a su bandeja de entrada y visualiza el mensaje “Prueba esta pagina 07”.

El mensaje lleva código JavaScript en un enlace.

Y lleva un fichero adjunto. Si se abre, este fichero, que se aloja en “webserver.example.com” podrá acceder a los contenidos de otras páginas que tengan el mismo origen. En este caso, se lee el formulario de login en un IFRAME oculto y se extrae de él las credenciales.

Puede comprobarse si ha funcionado accediendo a

```
http://webserver.example.com/ataques/demo%2002/lista_credenciales.php
```

Esto no lo detecta el filtro anti-XSS.

DEMO 08 - XSS basado en DOM

Se sigue usando Firefox

El usuario1 va a su Bandeja de Entrada y visualiza el mensaje “Interesante 08”. En él hay un enlace a una URL que parece que realiza un ataque de XSS reflejado.

El ataque toma control de la página de envío de mensajes y realiza repetidos envíos al usuario “jefazo”.

Parar el envío pulsando la tecla “Esc” y ver el código fuente. No se trata de XSS Reflejado. Los filtros anti-XSS no lo suelen detectar.

Se aprovecha que la página modifica su contenido mediante código JavaScript, utilizando para ello parte de la URL.

DEMO 09 - ACCESO A DATOS

Esta prueba no se realizó en la charla por motivos de limitaciones de tiempo

Se sigue usando Firefox

El atacante visita

```
http://malicioso.uhw/ataques/demo%2009/lista.php?borra=1
```

... para borrar la lista de cosas robadas.

El usuario1 va a su Bandeja de Entrada y visualiza el mensaje “Demo 08”. Abre el adjunto que lleva el mensaje.

Este fichero lleva código HTML que realiza varias peticiones a la aplicación: primero, obtiene la lista de los mensajes; después, solicita cada mensaje al servidor. Toda la información es guardada y enviada al servidor utilizando un objeto XMLHttpRequest.

El atacante visita

```
http://malicioso.uhw/ataques/demo%2009/lista.php
```

... y ve los mensajes de la víctima.

DEMO 10 - CSRF PARA EXPLOTAR OTRAS VULNERABILIDADES

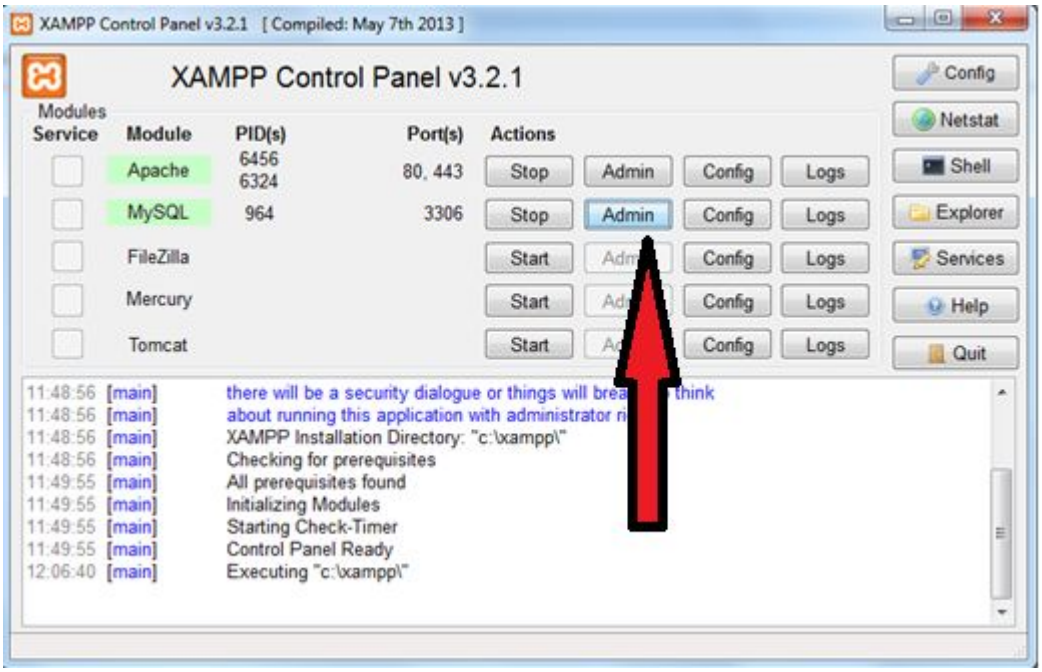
Se sigue usando Firefox

El usuario visita

http://malicioso.uhw/ataques/demo%2010/1.htm

Le sale la página de login. Pero cuando lo intenta le sale un error de MySQL.

La base de datos ha sido borrada, como se puede comprobar utilizando PHPMyAdmin. En XAMPP se puede acceder a PHPMyAdmin haciendo clic en el botón “Admin” de la línea “MySQL”



Si se desea seguir realizando pruebas con la aplicación es necesario volver a crear la base de datos de la aplicación. Las instrucciones para ello pueden encontrarse en el apartado “INSTALACIÓN DEL ENTORNO DE SERVIDOR DE PRUEBAS”.

SI QUIERES HACER TUS PROPIAS PRUEBAS...

Recuerda que la aplicación tiene tantos fallos de diseño e implementación como ha sido posible incluir en su código.

En particular, tiene problemas de SQL Injection. Por ejemplo, cuando vayas a poner una comilla simple (‘) en un mensaje, ten en cuenta que ese carácter es el que usa SQL para encerrar las cadenas de texto.

En definitiva, si quieres que tu mensaje lleve una comilla simple y que no cause un ataque de SQL Injection, ponle delante una barra invertida, como en:

```
\’
```