

Gestión de la Información en la Web

Curso 2016-17

Práctica Auditoría de seguridad

Fecha de entrega: viernes 27 de enero de 2017, 13:55h

Entrega de la práctica

La entrega de la práctica se realizará a través del Campus Virtual de la asignatura mediante un fichero **auditoriaXX.pdf** donde **XX** es el número de grupo.

Calificación

Se tendrá en cuenta la cantidad de vulnerabilidades detectadas, la valoración de las situaciones problemáticas que pueden generar, la explicación paso a paso de cómo explotar cada vulnerabilidad y la precisión y calidad de las medidas de mitigación propuestas. Para obtener la máxima calificación es imprescindible detectar al menos una vulnerabilidad de cada tipo.

Declaración de autoría e integridad

Todos los ficheros entregados contendrán una cabecera en la que se indique la asignatura, la práctica, el grupo y los autores. Esta cabecera también contendrá la siguiente declaración de integridad:

(Nombres completos de los autores) declaramos que esta solución es fruto exclusivamente de nuestro trabajo personal. No hemos sido ayudados por ninguna otra persona ni hemos obtenido la solución de fuentes externas, y tampoco hemos compartido nuestra solución con nadie. Declaramos además que no hemos realizado de manera deshonesto ninguna otra actividad que pueda mejorar nuestros resultados ni perjudicar los resultados de los demás.

No se corregirá ningún fichero que no venga acompañado de dicha cabecera.

Auditoría de seguridad web

En este apartado vamos a auditar la seguridad de una aplicación web muy sencilla desarrollada con el *framework* `bottle`. Esta aplicación permite a los usuarios añadir preguntas y contestarlas. Internamente utiliza *SQLite* para almacenar los datos. Por simplicidad esta aplicación no tiene ningún tipo de gestión de usuarios ni sesiones: cualquiera puede publicar y contestar usando cualquier nombre.

Lo primero que hay que hacer es descargar la aplicación llamada *El Coladero* desde el Campus Virtual y descomprimirla. Antes de ejecutar la aplicación es necesario crear una base de datos con formato *SQLite* llamada `database.db` a partir del fichero `database.sql` y colocarla en el misma carpeta que el fichero `coladero.py`. (En el Campus Virtual podéis encontrar un fichero `database.db` generado en Linux por si os sirve). Una vez creada la base de datos podéis arrancar el servidor y acceder a su página principal `http://localhost:8080/show_all_questions` para ver el funcionamiento de la aplicación, que únicamente produce contestación en 5 rutas:

1. `@get('/show_all_questions')`
2. `@post('/insert_question')`
3. `@get('/show_question')`
4. `@post('/insert_reply')`
5. `@get('/search_question')`

Además del funcionamiento también debéis revisar con detalle el código Python de la aplicación web para tratar de detectar posibles vulnerabilidades. El resultado de la auditoría será un fichero llamado `auditoria.pdf` donde se explicará con detalle las vulnerabilidades detectadas en *El Coladero*. En esta práctica nos centraremos únicamente en 3 tipos de vulnerabilidades: **inyección SQL**, **XSS persistente** y **XSS reflejado**. Por cada vulnerabilidad detectada se debe rellenar una **tabla** como la de la Figura 1, que contiene los siguientes apartados:

- **Ruta de la aplicación** que exhibe la vulnerabilidad. Si consideráis que la responsabilidad de la vulnerabilidad es compartida, incluid todas las rutas involucradas.
- **Tipo de vulnerabilidad**: inyección SQL, XSS persistente o XSS reflejado.
- Explicación de qué **situaciones peligrosas o no deseadas** puede provocar en esta aplicación web concreta.
- **Ejemplo paso a paso** de cómo explotar esta vulnerabilidad. Se deben incluir capturas de pantalla en la que se vea con detalle cada uno de los pasos.
- Explicación detallada de qué **medidas** hay que incluir en la aplicación web para **mitigar esta vulnerabilidad**. Tened en cuenta el principio de *defensa en profundidad* por el cual es preferible implantar varias medidas

INFORME DE VULNERABILIDAD
Ruta(s) de la aplicación involucrada(s)
Tipo de vulnerabilidad
Situaciones peligrosas o no deseadas que puede provocar
Ejemplo paso a paso de cómo explotar la vulnerabilidad (con capturas de pantalla)
Medidas para mitigar la vulnerabilidad

Figura 1: Tabla de información de una vulnerabilidad.

de seguridad si es posible. En el caso de recomendar la validación de las entradas del usuario, debéis incluir el formato esperado de cada parámetro con toda la precisión posible (por ejemplo “*cadena de letras minúsculas de longitud entre 5 y 10 caracteres*”).

Nota: Probad varios navegadores ya que cada uno puede tener distintas protecciones contra XSS. Si únicamente usáis un navegador podéis llegar a la conclusión de que una vulnerabilidad no está presente cuando realmente sí que lo está pero el navegador os está protegiendo de manera silenciosa.