

Práctica: Seguridad web**Grupo:** 04

Autores: PETAR KONSTANTINOV IVANOV, JORGE SAN FRUTOS IGLESIAS, IGNACIO VILLEGAS DE MIQUEL y YUEJIE XU

PETAR KONSTANTINOV IVANOV, JORGE SAN FRUTOS IGLESIAS, IGNACIO VILLEGAS DE MIQUEL y YUEJIE XU declaramos que esta solución es fruto exclusivamente de nuestro trabajo personal. No hemos sido ayudados por ninguna otra persona ni hemos 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.

INFORME DE VULNERABILIDAD	
Ruta(s) de la aplicación involucrada(s)	<p>http://localhost:5000/show_all_questions</p> <p>http://localhost:5000/insert_question</p>
Tipo de vulnerabilidad	SQL Injection.
Causante de la vulnerabilidad	<pre>qbody = ""INSERT INTO Questions(author, title, tags, body, time) VALUES ('{0}','{1}','{2}','{3}',CURRENT_TIMESTAMP)""</pre> <pre>query = qbody.format(author, title, tags, body)</pre> <pre>cur.executescript(query)</pre> <p>La función executescript() de la insert_question permite ejecutar varias sentencias de sql.</p> <p>La sentencia de qbody no escapa la entrada.</p>
Situaciones peligrosas o no deseadas que puede provocar	Por ello, además del Insert, el usuario puede ejecutar cualesquiera sentencias tras finalizar Insert, es decir, realizar cualquier acción con la BD.
Ejemplo paso a paso de cómo explotar la vulnerabilidad (con capturas de pantalla)	<p>Paso 1: Introducir datos sin escapar.</p> <ul style="list-style-type: none"> • Autor: 1 • Título: 1 • Etiqueta: 1 • Pregunta: 1', 0); delete from questions;--

← → ↻ ⓘ http://localhost:5000/show_all_questions

El Coladero

Foro de preguntas y respuestas

Búsqueda por etiqueta:

Título: **Mejor editor para programar**
Autor: pepe
Fecha: 2015-12-27 16:40:43
Etiquetas: Editor, programar
[Ver](#)

Título: **Listas en Python**
Autor: pepe
Fecha: 2013-06-14 12:00:42
Etiquetas: listas, Python
[Ver](#)

Título: **Diccionarios**
Autor: ana
Fecha: 2012-03-19 11:54:23
Etiquetas: diccionarios, Python, programar
[Ver](#)

Autor:
Título:
Etiquetas:

```
1', 0); delete from questions;--
```


Pregunta:

Paso 2: Insertar la pregunta en la BD.

← → ↻ ⓘ http://localhost:5000/insert_question

Mensaje insertado con éxito

[Volver](#)

Paso 3: Pulsar volver para ver todas las preguntas.

← → ↻ ⓘ http://localhost:5000/show_all_questions

El Coladero

Foro de preguntas y respuestas

Búsqueda por etiqueta:

Autor:
Título:
Etiquetas:

Pregunta:

Una vez llegado a este paso, podemos visualizar que no tiene ninguna pregunta en la página de show_all_questions. Ya que fueron borrados tras delete from questions.

Medidas para mitigar la vulnerabilidad


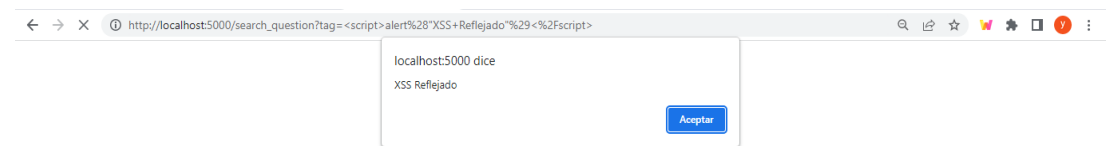
- Escapar todas las entradas.

```
qbody = """INSERT INTO Replies(author,body,time,question_id)
VALUES (:author, :body, CURRENT_TIMESTAMP, :question_id)"""
params = {'author': author, 'body': body, 'question_id': question_id}
```

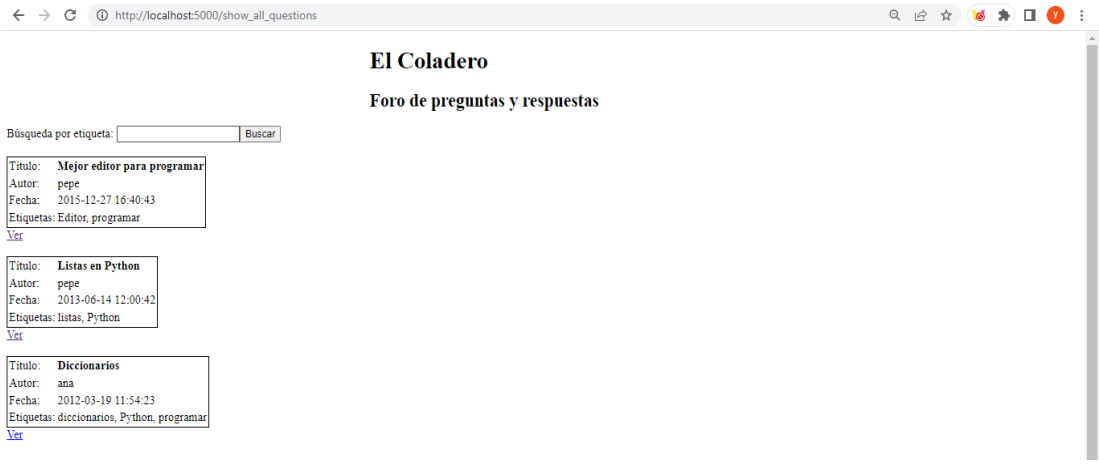
- Cambiar el método `executescript` por `execute`.

```
cur.execute(qbody, params)
```

- Aplicar principios de mínimo privilegio y segregar usuarios, es decir, los usuarios deben tener únicamente los privilegios necesarios para su tarea y ninguno más y utilizar diferentes usuarios para los distintos accesos a la base de datos, no un solo usuario omnipotente.

INFORME DE VULNERABILIDAD	
Ruta(s) de la aplicación involucrada(s)	<p>http://localhost:5000/show_all_questions</p> <p>http://localhost:5000/search_question</p>
Tipo de vulnerabilidad	XSS Reflejado.
Causante de la vulnerabilidad	Por la falta de desinfectar los datos introducidos por el usuario (tag = request.args['tag']) y por la incorporación de ello en la página devuelta(return render_template('messages_search.html', questions=res, tag=tag)).
Situaciones peligrosas o no deseadas que puede provocar	Robo de cookies, redireccionamiento a sitios maliciosos, cambio de apariencia de la pagina web y robo de credenciales
Ejemplo paso a paso de cómo explotar la vulnerabilidad (con capturas de pantalla)	<p>Paso 1: Introducir <script>alert("XSS Reflejado")</script> en la barra de búsqueda.</p>  <p>Paso 2: Pulsar el botón de buscar y aparece “XSS Reflejado”.</p> 
Medidas para mitigar la vulnerabilidad	<ul style="list-style-type: none"> Escapar todas las entradas. tag = html.escape(request.args['tag']) Desinfectar todo el texto que va a aparecer en la página HTML generada, tanto el que provenga de la petición como aquel obtenido de la BD.

```
En search_question()
    res = list(list())
    for i in cur.fetchall():
        aux = list()
        for j in i:
            aux.append(html.escape(j))
        res.append(aux)
    conn.close()
    return render_template('messages_search.html', questions=res, tag=tag)
```

INFORME DE VULNERABILIDAD
Ruta(s) de la aplicación involucrada(s) http://localhost:5000/show_all_questions http://localhost:5000/show_question?id=3 http://localhost:5000/insert_reply
Tipo de vulnerabilidad XSS Persistente.
Causante de la vulnerabilidad Por la falta de desinfectar los datos introducidos por el usuario (body = request.form['body'] y author = request.form['author']) y la incorporación de ello en la BD. <pre> qbody = """INSERT INTO Replies(author,body,time,question_id) VALUES (:author, :body, CURRENT_TIMESTAMP, :question_id)""" params = {'author': author, 'body': body, 'question_id': question_id} cur.execute(qbody, params) conn.commit() </pre> Provoca que el servidor utilice dicho dato con código malicioso en la construcción de la propia página web (return render_template("message_detail.html", q=question, replies=replies, ident=ident)).
Situaciones peligrosas o no deseadas que puede provocar Robo de cookies, redireccionamiento a sitios maliciosos, cambio de apariencia de la página web y robo de credenciales.
Ejemplo paso a paso de cómo explotar la vulnerabilidad (con capturas de pantalla) Paso 1: Pulsar Ver de una pregunta.
 <p>The screenshot shows a web browser at the URL http://localhost:5000/show_all_questions. The page title is 'El Coladero' and the subtitle is 'Foro de preguntas y respuestas'. There is a search bar with the text 'Búsqueda por etiqueta:'. Below the search bar, there are three question cards. The first card has the title 'Mejor editor para programar', author 'pepe', date '2015-12-27 16:40:43', and tags 'Editor, programar'. The second card has the title 'Listas en Python', author 'pepe', date '2013-06-14 12:00:42', and tags 'listas, Python'. The third card has the title 'Diccionarios', author 'ana', date '2012-03-19 11:54:23', and tags 'diccionarios, Python, programar'. Each card has a 'Ver' link below it.</p>
Paso 2: Introducir <code><script>alert("XSS Persistente")</script></code> en el campo de Autor y pulsar el botón de Contestar.

← → ↻ ⓘ http://localhost:5000/show_question?id=3

El Coladero

Foro de preguntas y respuestas

Búsqueda por etiqueta:

Título: **Mejor editor para programar**
Autor: pepe
Fecha: 2015-12-27 16:40:43
Etiquetas: Editor, programar
Mensaje: Vim o Emacs?

Autor:

Respuesta:

Paso 3: Pulsar Volver y acudir a la pregunta respondida por el usuario. A partir de ello, cada acceso a dicha pregunta mostrará “XSS Persistente”.

← → ↻ ⓘ http://localhost:5000/insert_reply

Mensaje insertado con éxito

[Volver](#)

← → × ⓘ http://localhost:5000/show_question?id=3

localhost:5000 dice
XSS Persistente

Medidas para mitigar la vulnerabilidad

- Escapar todas las entradas.
author = html.escape(request.form['author'])
body = html.escape(request.form['body'])
question_id = html.escape(request.form['question_id'])
- Desinfectar todo el texto que va a aparecer en la página HTML generada, tanto el que provenga de la petición como aquel obtenido de la BD.
En **show_question()**
question = list()
for i in cur.fetchone():
question.append(i)
cur.execute(qbody2, params)
replies = list(list())
for i in cur.fetchall():

7

```
        aux = list()
        for j in i:
            aux.append(html.escape(j))
        replies.append(aux)
    conn.close()
    return render_template("message_detail.html", q=question, replies=replies, ident=ident)
```