UNIVERSIDAD
COMPLUTENSE
MADRID

Validación de entradas

Gestión de la Información en la Web
Enrique Martín - emartinm@ucm.es
Grados de la Fac. Informática

Entradas

- Una aplicación web recibe diferentes **entradas desde los clientes**:
  - Datos para dar de alta/modificar un cliente
  - Identificador de producto a mostrar
  - Texto y opciones para la búsqueda
  - Número de página a mostrar
- Todos estas entradas se transmiten como campos en una petición GET o POST.

Entradas

- Estas entradas sirven para realizan **consultas o modificaciones** en la base de datos, o incluso en ficheros del servidor.
- Por ello es muy importante que **TODAS** sean **validadas** para asegurar que son entradas legítimas y no realizan acciones peligrosas.
- Si las entradas no se comprueban pueden producir **inyección**: modificar de manera indeseada estructuras de datos que se van a ejecutar.

Aspectos a validar

- El dato está **bien formado**:
  - Es una ruta de directorio, una fecha, un fichero PNG correctamente construido, etc.
- El **tamaño**:
  - La ruta de directorio no sobrepasa la longitud máxima, el nombre de usuario no desborda el campo de la BD, los elementos no desbordarán el buffer, etc.
- Vigilar los ***hidden inputs*** de un formulario: un atacante podría introducir valores inesperados.

Aspectos a validar

- Asegurar la **ausencia de comandos** dentro de las entradas de los usuarios:
  - Inyección SQL para visualizar o modificar la BD
  - Ejecución de comandos en el servidor
  - Forzar errores:
    - Obtener información del sistema (BD, existencia de ficheros, etc.)
    - Denegación de servicio

Estrategias para validar las entradas del usuario

Declaración de variables

- Python es flexible y te permite utilizar y definir variables en cualquier lugar.
- Para evitar problemas de seguridad, se recomienda **declarar todas las variables** utilizadas y asignarlas a valores por defecto **antes de usarlas**.
- Así evitamos que su valor “por defecto” pueda depender inadvertidamente de la entrada del usuario.

Solo tratar entradas esperadas

- Filtrar las entradas recibidas y anular aquellas no esperadas:

```
params = {}
expected = ['carModel', 'year', 'bodyStyle']

if admin:
  expected.append( 'profit' )

for key in expected:
  if key in request.query:
    params[key] = request.query[key]

# Utilizar únicamente params
```

Tipo, longitud, formato

- Comprobar que las entradas están **bien formadas**:
  - Cadenas:
    len(cad), cad != “”
  - Números:
    isinstance(year, int), isintance(value, float)
  - Booleanos:
    isinstance(flag, bool)

Desinfectar entradas

- **Eliminar partes peligrosas** (*sanitize*) de las entradas de usuario cuando se las vas a pasar a otros sistemas (como MySQL, MongoDB o un terminal).
- **Metacaracteres** como **“**, ****, **'**, etc. Existen funciones como **MySQLdb.escape_string()**.
  - Puede no manejar todos los caracteres peligrosos, así que hay que estudiar su comportamiento o incluso crear una función personalizada.

Desinfectar entradas

- **Rutas de ficheros**:
  - Deben ser **cadenas de texto** → nada de datos binarios.
  - Aunque algunos sistemas permiten caracteres extendidos, mejor ceñirse a **ASCII**.
  - Evitar que contengan **signos de puntuación**, como mucho '**-**' y '_'. Si puedes evitar el punto de la extensión de fichero, mejor.
  - Comprobar la **longitud de la ruta**. Los sistemas de ficheros tienen un límite.

Desinfectar entradas

- **E-mail**:
  - Existen **expresiones regulares** para comprobar que una cadena representa una **dirección de e-mail**. En el caso más simple:
    - Debe ser una cadena de texto
    - Debe tener una sola @
    - Debe tener como mínimo un punto

Desinfectar entradas

- Si se genera salida **HTML** a partir de entradas, se puede utilizar **cgi.escape()** para evitar caracteres problemáticos:
  - '&'
  - '<'
  - '>'
- Ejemplo:

```
>>> cgi.escape('<img src="logo.png" />')
'&lt;img src="logo.png" /&gt;'
```

Mensajes de error

- Si al usar entradas del usuario se producen errores, es importante **no mostrar información valiosa** a un posible atacante.
  - "El usuario no existe" → prueba otro
  - "La contraseña es incorrecta" → ¡el usuario existe!
  - "Usuario o contraseña incorrectas" → ¿qué es lo que está mal?
- En producción **no mostrar mensajes internos del servidor web** (parámetro **debug=False**).

Mensajes de error

- **Capturar** los errores (valor de retorno o excepciones) y:
  1. Mostrar un **mensaje inocuo**, bien generando una página o redirigiendo a una página de error genérico.
  2. Almacenar todos los datos necesarios en un **log de errores y notificar** al responsable.