Cross Site Scripting Prevention on Forms

User Guide
About Cross Site Scripting

OWASP (Open Web Application Security Project) defines Cross-Site Scripting (XSS) attacks as a type of injection problem in which malicious scripts are injected into otherwise benign and trusted web sites.

XSS attacks occur when an attacker uses a web application to send malicious code, generally in the form of a browser side script, to an end user.

An attacker can use XSS to send a malicious script to an unsuspecting user. The user’s browser has no way of knowing that the script should not be trusted, and will execute the script. Because the browser treats the script as if it came from a trusted source, the malicious script can access any cookies, session tokens, or other sensitive information retained by the browser and used with that site.

As an example of a cross site scripting attack, the attacker sends a specially crafted e-mail message to a victim. The email contains a link to a Responsys form with malicious script such as example shown below:

<A HREF=http://website.com/registration.cgi?clientprofile=<SCRIPT>malicious code</SCRIPT>>Link</A>

When an unsuspecting user clicks on this link, the URL, including the malicious code, is sent to the Responsys server. If the Responsys server sends a page back to the user, the malicious code will be executed on the user’s Web browser.
About Preventing Cross Site Scripting in Forms

You can prevent XSS in forms in two ways:

**Output Encoding**

Output encoding means escaping characters. When the response is output encoded, the malicious script will be escaped and will not execute. Some examples of characters and their escaped equivalents are:

- `&` --> `&amp;`
- `<` --> `&lt;`
- `>` --> `&gt;`

**Input Validation**

Input validation ensures that input fields contain data only of the specified type, for example integer or string. When the form is submitted, the application checks the submitted data to ensure that the fields contain the specified type of data. If there is a malicious script in the field, the validation will fail and the form will not be submitted, thus preventing the script from executing.

The diagram below illustrates the steps you need to take to prevent XSS in forms.

Directions for each step are provided in the following sections.
Setting up Output Encoding

To set up a form for output encoding, you need to enable it for the form, then enable it for form fields and personalization variables.

Enabling output encoding for the form and form fields

By default, output encoding is enabled for the account, and will be enabled for newly created forms. Forms that you copy from an existing one retain the setting of the existing form.

After a form has been created in the Form Wizard, follow the steps below to enable output encoding for the form. When you enable output encoding for a form, all fields on that form are automatically protected.

To enable output encoding for the form and form fields

1. In the Form Wizard, click the arrow next to the form and select Output Encoding from the drop down menu.
   Note that if Output encoding is grayed out, it is not enabled for the account. In this case, please contact Responsys Support to enable it.
2. Select the **Enable Output Encoding for this form** checkbox. This protects the form and all its fields.

3. To disable encoding for a specific field, clear the field’s checkbox.

4. Click **Save**.

Note that these steps do not protect personalization variables on the form. To protect personalization variables, follow the steps in the next section.
To enable output encoding for personalization variables

To enable encoding for personalization variables, use the built-in functions $\text{outputencoding()}$ for HTML characters and $\text{outputjsencoding()}$ for JavaScript. These built-ins encode output by escaping characters for the personalization variable. For example, to encode a field called first_name, specify $\text{outputencoding(first_name)}$.

You can use these built-ins as part of Dynamic Content, as shown in the illustration below.
Setting up Input Validation

When you use input validation, Oracle Responsys checks the data submitted in the fields you select to ensure that the fields contain the specified types of data, for example email addresses or dates. If the data in the fields does not meet validation criteria, the form will not be submitted.

To set up input validation

1. In the Form Wizard, click the arrow next to the form and select Input Validation from the drop down menu.

2. Select the required fields for the form, and click Next.
3. For each required field, select the valid data type, and then click Finish.

Valid data types are:

- **String**
  Has no restrictions. This is default value with no restrictions.

- **Email**
  Allows input in the email address format.

- **Integer**
  Allows integers between -2,147,483,648 and 2,147,483,647.

- **Real**