Security: Cross-site Scripting (XSS)

Cross-Site Scripting (XSS) is an attack to a website where data that contains scripts is executed and malicous code created by one user may be run by another user.

Servoy will sanitize all data that is shown in the WebClient and the NGClient to prevent this in Servoy solutions.

Example

A solution allows a user to register users and has a form for backoffice handling that lists all users.

When the user registers with a name that contains scripting '*John<script>doSomethingBad()</script>Doe*', Servoy will not execute the script but will sanitize the data and just show '*John Doe*'.

Trusting data as html

In some situations data used in elements contains html that has to be shown as-is.

Only in cases where the source of the html can be fully trusted, an element should be configured to disable sanitizing.

This is done via the *UI_PROPERTY.TRUST_DATA_AS_HTML* client property on an element:

```
elements.usernameLabel.putClientProperty(APP_UI_PROPERTY.TRUST_DATA_AS_HTML, true);
```

When this property is set on an element, data from its dataProvider will be trusted and not sanitized.

Alternatively, sanitizing of data can be turned off for the entire running client by applying the same property on the application node:

```
application.putClientProperty(APP_UI_PROPERTY.TRUST_DATA_AS_HTML, true);
```

Using this at application level is highly discouraged, your system may be vulnerable to XSS attacks.

For more information see UICONSTANTS**.**