CSRF   
Cross Site Request Forgery- it is used to prevent users from unintentionally making changes because of a CSRF attack. We can set validation for activities and streams, add host names to allow list and specify host names that you want to check for CSRF token.

How can we check this in Pega

Go to **Configure> System> Settings > Cross Site Request Forgery**

Now let us understand deeply how a user is tricked to click a link and how the attacker tries to get into the site.

Let us assume a user who is trying to access the application via SSO kind of authentication.

So we all are aware of authentication so whenever user tries to log in to the application for the first time with the authentication which is in place assume that it is a single sign on which means user is logging in for the first time for the rest of the items user need not to login again and again that is the whole idea of single sign-on.

So as part of this user is logging into the application and establishing the session from then onwards whenever user is trying to hit the same URL or getting redirected to the same application user will not be challenged with any login screen which we are well aware of so that is why we are calling that user is logging into the system for the first time let’s say where SSO token is already established and its already active from then onwards all the hit HTTP requests that are going into the web browser obviously will have the valid cookie with the SSO token which will allow the user to get redirected to application and see the expected screen.

One thing we understood user is successfully getting redirected to the application because the valid SSO token which is there in the system that means in the user system any HTTP head is coming to this application that gets successfully passed through and user will be able to look at that particular information and also importantly that session allows this user’s access to access the protected resources of this particular application.

The attacker is not having access but through any other means he can make the user to click on the hyperlink or any typical web application. So since the end users will not be as mch as well trained or not too technically sound they will be tricked too click on that hyperlink with the help of which attacker has done his job. Now this is an example of CSRF attack.

So now we understand how a cross-site request forgery can happen.

Now as a Developer or Tester we might think how Pega as a Product how can I do something so that this kind of security vulnerability can be avoided good news is it’s way to simple as a developer you need not do anything literally I mean you need to design or code something to avoid this attack. Pega has already taken care of this CSRF vulnerability and how you need to enable is via configuration to validate HTTP request.

So this will check whether the URL is Forged or not.

A diagrammatical representation is shown below.

Hyperlink (DANGER)  
Received in Mailbox)

User machine where SSO token is already active(already logged in with SSO)

Now click on Configure > System > Settings > Cross Site Request Forgery

To prevent the browser from submitting the PegaRULES cookie in a request from a non-originating site, complete the following steps:

Select the Enable samesite cookie attribute checkbox.

In the Samesite Options list, choose one of the following options:

None – If you select this option, Pega Platform offers no protection. The browser attaches the cookies in all cross-site browsing contexts.

Lax – If you select this option, Pega Platform provides a reasonable balance between security and usability for websites that want to maintain logged-in sessions after users arrive from an external link. The browser does not send cookies in requests from non-originating sites.

Strict – If you select this option, Pega Platform prevents the browser from sending cookies to the target site in all cross-site browsing contexts, including when following a regular link.

To enable CSRF settings, select Enable CSRF token check.

Selecting this check box will enable the CSRF token validation.

If you have enabled CSRF token check, select one of the following Secure fields:

All activities & streams

CSRF validation checks all activities and streams for CSRF tokens in your system. If you select this option, you can specify certain streams and activities to be excluded from CSRF token validation by entering them in the Allowed Activities field and the Allowed Streams field. Separate multiple activities and streams with commas.

Specific activities & streams

CSRF validation checks the activities and streams that you specify in the Secured Activities and Secured Streams fields for CSRF tokens. Separate multiple activities and streams with commas.

Optional: To add names to a safe "allow list" of host names to ignore during CSRF token validation, perform the following actions.

In the Referrer Settings section, select Enable referrer check.

Select the Allow domains only if matches exactly with Referrer check box to allow requests which exactly match the URL given in the Allowed referrers field.

If Enable referrer check is checked and http://pega.com is provided in Allowed referrers, then http://pega.com.xyz is valid, but http://xyz.pega.com is not valid.

If Allow domains only if matches exactly with Referrer is checked in addition to the conditions above, only the exact match http://pega.com is valid, but not http://pega.com.xyz or http://xyz.pega.com.

In the Allowed referrers field, enter host names that you want to be checked for a CSRF token. Separate multiple host names with commas.

Click Submit.

If you changed the value of Enable CSRF token check, you must restart your system for the new value to take effect.

Let’s see in the configurations

A screenshot of a computer

Description automatically generated

**A screenshot of a computer

Description automatically generated**