CSRF Protection

General information

CSRF protection in TeamCity has been implemented since TeamCity 2017.1 (issue). This protection implies a number of requirements on HTTP requests.

Implications for reverse proxy configuration

When a TeamCity server has a reverse proxy in front of it (Nginx, IIS, Apache), this proxy should be configured to pass headers from the original request to the TeamCity server.

Origin and Referer headers must be passed unmodified, when present.

The Host header should be passed as well, and if it is not possible due to some reason, X-Forwarded-Host must be set to the value of the original Host header.

Here are our recommendations for reverse proxy configuration for TeamCity.

Implications for non-browser HTTP clients

Non-browser HTTP clients which reuse authentication for REST scripting by supplying the `TCSESSIONID` cookie with the request need to be updated to supply the Origin HTTP header with the same value as the host the request is being sent to.

CSRF checks for HTTP request

When considering HTTP request safety from the TeamCity perspective, the following checks are sequentially made:

1. If an HTTP request is a non-modifying one (such as GET), it is considered safe
2. If an HTTP request has a secure CSRF token either in the parameter or in the HTTP header and this token matches the one stored in user session, it is considered safe.
3. If an HTTP request has the Origin header, it must match a host from a trusted domain (see below), otherwise, the request is rejected without further processing
4. If an HTTP request has the Referer header which matches the Host header or X-Forwarded-Host header, it is considered safe
5. If an HTTP request has the X-Requested-With=XMLHttpRequest header, it is considered safe
6. If an HTTP request uses basic authentication and there is no user in TeamCity session/cookies, the request is considered safe

Trusted domain

TeamCity considers the following domains/hosts as trusted:

- Host header value
- X-Forwarded-Host header value (can be set separately in case of proxy configuration)
- One of the CORS origins, configured for REST access.

Troubleshooting

When you face problems regarding CSRF protection in TeamCity (e.g. you get "Responding with 403 status code due to failed

CSRF check" response from the server), you can follow these steps:

- If you use a reverse proxy, make sure you correctly configure Host/Origin headers, as described above. In the meantime, you may want to add the public URL of your server to CORS-enabled origins.
- You can temporary disable CSRF protection at all by setting the `teamcity.csrf.origin.check.enabled=logOnly internal property`.
- Information about failed CSRF attempts are logged into TeamCity/logs/teamcity-auth.log files. For more detailed diagnostics of the requests, enable debug-auth logging preset.