# Security FAQ

This page provides the answers to common security questions for Totara Learn.

## Input/Output Validation

### Where are security relevant validations done? (client- or server-side)

Validation is consistently and reliably performed on the server to ensure a secure environment.

Client side validation is performed where applicable and possible, it provides the user a quicker response in many cases, but is never relied upon as an authoritative validation.

### When calling sites with visible parameters (e.g. https://...?file=xyz.php), how does the application ensure that there are no invalid inputs sent e.g. os commands?

All inputs are cleaned using type specific filtering functions, to ensure that it contains only the expected content. See required_param(), optional_param() and clean_param() functions.

### For file uploads: how does the application ensure that no executable code is uploaded?

Files are uploaded to a specific upload location without execute permission and no user submitted files are ever executed. Uploaded file names are replaced with a hash. Totara Learn supports anti-virus scanning of uploaded files using ClamAV.

What can be uploaded is in many cases not restricted. Any files that are found by ClamAV to contain viruses or malicious code are immediately quarantined or destroyed (depending on configuration).

Totara is then very careful in how it provides those files. If the file is of a recognised and expected format and is being used within an area that supports inline display (such as when you upload an image for use in the editor) then it is mediated with headers allowing its use within the browser. Any file that is not of a recognised nor expected format is served explicitly with headers that prevent the file from executing in the users browsers. This forces the user to download the file. Once downloaded, the user should observe general internet security and only open files that they trust.

### Which security measures are taken to prevent cross site scripting (XSS)?

User data is escaped on output in a range of ways depending on content type and output context. This includes stripping or escaping HTML tags for simple text or using HTML purifier for cleaning user-supplied HTML. See s(), p(), format_string() and format_text() functions.

User data is also filtered on input using required_param(), optional_param() and clean_param() functions.

User data that is used within JavaScript code is passed into the code via a dedicated method which properly escapes the data. See $PAGE->requires->data_for_js() method.

For XSS via user-supplied files, we prevent it by forcing download of files that are uploaded to untrusted areas or uploaded by untrusted users.

### Which security measures are taken to prevent cross site request forgery (CSRF)?

All data-alerting actions are validated using a CSRF token referred to internally as the sesskey. See require_sesskey() and confirm_sesskey() functions.

### Which security measures are taken to prevent SQL-Injection?

All user-supplied data is filtered prior to use, then only used in SQL queries via parameters. The database layer handles escaping of parametrised data - either internally escaping data or using bound parameters as implemented by the database driver.

### Which security measures are taken against automated attempts to gain access? (e.g. bots)

There are optional account lockout settings to lock accounts after excessive failed login attempts. When combined with password policies this makes it very difficult to brute force user accounts. When self-registration is enabled Totara Learn offers optional ReCaptcha integration to prevent automated sign-ups. Totara Learn also includes IP blocking functionality with support for whitelisting and blacklisting of IP ranges.

## Session-Handling

### Which security measures are taken to prevent Session-Hijacking?

Session-ID is accepted from session cookies only. Session-ID is reset after each login and logout.

Totara Learn sets the 'X-Content-Type-Options: nosniff' and 'X-Frame-Options: deny' headers by default when requested resources may be files uploaded by a user. Sessions are only used when required (e.g. no session is used when transferring public static resources such as CSS/JS.

The 'X-XSS-Protection: 1; mode=block' header is currently set in limited scenarios. However, support for this header is inconsistent across our supported browsers, with some choosing to deprecate the related functionality. Primary protection against XSS is by cleaning user input as appropriate.

### How are Session-IDs generated?

Session-IDs are generated by php in a standard way. See the PHP built-in session_id() function.

### When are Session-IDs generated?

Session-ID is generated on first access, login, logout and session timeout.

### How does the application ensure that users only have access to certain data?

Totara Learn provides a fine-grained hierarchical permissions model based on roles, contexts and capabilities. Individual system actions are controlled by specific capabilities, which can be granted or denied to roles in specific contexts. Users are assigned one or more roles which can be granted both across the entire site and to specific content within the site, such that a user can be given access to only the content and actions they need have access to.

### Are Session-IDs part of the URL?

The session ID is only ever passed as a cookie via HTML headers, it is never passed via any other means.

### How long is the session timeout?

The session timeout is configurable, the default setting is 2 hours.

## Shortly describe session handling after logout.

Session data is destroyed, then Session-ID is regenerated.

## Which data stays in the session after logout?

No session data remains after logout. There is a separate cookie for storing the user's username for convenience, if the user checks the "Remember username" checkbox.

## Which data is saved in the session?

1. Basic current user information (excluding password hash, description, etc.)
2. Session key for CSRF prevention
3. Some other caches
4. Other access control and auxiliary user preference data

## Are there further identification features in the session? (e.g. once-only token)

No.

# Cookies

## Which type of cookies is used?

Totara uses three cookies. The session cookie is required while the other two are optional and controllable via configuration settings.

### Session cookie

Totara uses a cookie to track the users session.

This cookie is created for every user who browses any page on the site and Totara will not function if this cookie is blocked.

- In 2.7 and below this cookie is given the name MoodleSessionXXX where XXX is determined by $CFG->sessioncookie.
- In 2.9 and above this cookie is given the name TotaraSessionXXX where XXX is determined by $CFG->sessioncookie.

### Remember Username cookie

Totara uses a cookie to remember the users username if the equivalent option is checked on the login page.

This cookie is created by the login page after the user logs in if the remember username functionality is set to On, or if the functionality is set to optional and the user has checked the remember username box on the login screen.

The remember username functionality can be disabled browsing to Site administration  Security  Site policies and setting Remember username (rememberusername) to No.

- In 2.7 and below this cookie is given the name MOODLEID1_XXX where XXX is determined by $CFG->sessioncookie.
- In 2.9 and above this cookie is given the name TOTARAID_XXX where XXX is determined by $CFG->sessioncookie.

### Shibboleth

If the Totara site has been configured to use a Shibboleth provider for authentication then the Shibbotleth authentication plugin will create a cookie to facilitate its functionality.

This cookie is only created if the Shibboleth pluggin has been enabled and configured for use.

In all versions of Totara this cookie is given the name _saml_idp.

## Which flags are used to secure cookies? (e.g. httponly)

This is configurable in Totara Learn - all parameters of session_set_cookie() are set including path, domain, secure, httponly.

## Which data is saved in cookies?

Session-ID, optionally last logged in user name.

# HTTP(S) forms

## Which data is saved/submitted in hidden fields?

Totara Learn is a large application, listing all uses of hidden fields would be impractical.

### Which method is used to submit forms? (e.g. HTTP-POST)

All forms that submit data use HTTP-POST. There are a small number of forms that are used for navigation purposes only which still rely on HTTP-GET.

### How does the application ensure that SSL encrypted sites cannot be cached?

All pages provided by Totara Learn and content mediated by Totara Learn are sent with caching headers appropriate to their nature and content.

Dynamic content, authentication restricted content, and permission restricted content are all sent with headers requesting servers handling the request do not cache this page.

Pages which are static, contain no user specific or sensitive data, and which are of benefit to all users of the site may be sent with caching headers. A good example of where this happens is with CSS and JavaScript libraries which get mediated via Totara Learn.

It is worth noting that SSL is highly recommended as you cannot always trust the servers and systems that handle requests as they pass between the server and client.

### Which data is encrypted on transfer?

We rely on SSL for encryption of data transfer between the user and the LMS. All data is encrypted if site is configured to use SSL.

### Which encryption method is used for data transfer?

Clients access encryption depends on web server configuration, this is not part of our codebase.

The LMS code may contact other servers via curl extension which is configured to validate certificates. Again this depends on server configuration.

### How long is the key?

See above.

## Logging

### Which of the following events are not logged?

- Successful and failed logins and logouts?
- Errors on file access or access of sensitive data (unauthorized access)
- Authorized access of sensitive files, data or processes
- Access of security relevant parameters and tables, especially user profiles and capabilities and security policies

The level of logging is configurable within the application. However by default all of the above are logged.

### *Which data is logged?*

- Date and time of event: Yes
- Event ID or event type: Yes
- User-ID: Yes
- Network-Address (e.g. IP-Address): Yes

### Where is the logged data stored?

Log storage is configurable within the application. The default log storage is in a database table.

## Data storage

### Which information is encrypted on storage?

Data is not encrypted. Totara Learn relies on application security and server security to ensure data is secure. Passwords are hashed using a per-user salted bcrypt hash using the built-in PHP password hashing functionality.

### Which encryption method is used for data transfer?

See above.

### How long is the key?

See above.

# Third party libraries

## What third party libraries are used in Totara?

Information about all third party libraries, including version numbers, can be found within the product by navigating to *Site administration* > *Development* > *Third party libraries* as a site administrator.

## What is the policy for updates to third party libraries?

Minor updates to third party libraries are made within Totara stable releases in order to get security fixes, providing the update is backwards-compatible. If the update is not backwards-compatible then Totara will cherry-pick the security fix where possible, or otherwise address product code to ensure the security concern is not exploitable within Totara.

It is important to note that minor updates are not taken unless required for security or because of broken functionality within Totara.

Major upgrades to third party libraries are only made on the next major release of a Totara product.

## Support for third-party libraries

Totara maintains and supports the third party libraries used within Totara.

We will follow the above policy and ensure that any security issues are addressed completely and thoroughly.
Where possible we will ensure that security vulnerabilities within third party libraries are fixed at their source, either through upstream fixes or bespoke solutions we develop ourselves.
In the rare situation an issue cannot be fixed at the source we will find solutions for any and all individual uses of the third-party library, and put safeguards in place for any future uses, including in come cases removing functionality from the third party library.

## What third-party libraries are sometimes raised in security scans and penetration tests?

The libraries listed below may be raised as having known vulnerabilities. The development team regularly reviews the product to ensure any known vulnerabilities cannot be exploited.

- jQuery (including related modules such as jquery-ui and jquery-ui-dialog)
- YUILib
- Handlebars

Predominantly, vulnerabilities from the above libraries rely upon unsanitised user input being supplied to various library methods. Any such input will have been sanitised by the server. As described in "Support for third-party libraries" above, we will also backport fixes where appropriate to provide a further layer of security.

# Miscellaneous

## How are security measures tested?

Our partners and subscribers regularly submit our application to security penetration testing. In addition we commission an independent third party security review ourselves every other major release.

## Are connected applications required to authenticate each other?

There are many different systems and services that Totara Learn hooks into. By default Totara Learn does not communicate with any systems or services that do not provide public information, therefore there is no need to authenticate.

In situations where Totara Learn is configured to connect to and communicate with external systems and services it will use a means of authentication that is appropriate to that service or system.

Importantly Totara Learn can be configured to publish its own services. When doing so an authentication token is required by Totara Learn in order to authenticate the incoming request.
This token must be known and pre-approved in Totara Learn.

## Which comments are there in the HTML markup?

Totara is a large application developed over many years. Whilst HTML comments are rare is it impossible for us to empirically state the nature of any HTML comments that do exist, other than to say that they should relate to specific decision in the user interface and provide only insight into design direction.

## What documentation about the application is available?

Our main feature help site is at https://help.totaralearning.com/. Security specific information is on the Security page of our policy documents area.