QSM® The Intelligence behind Successful Software Projects

SLIM-Collaborate 4.0 Application Security

SLIM-Collaborate employs multiple practices to ensure the security of data stored in the application. This document describes the key methods used in the product.

Authentication and Passwords

Editing or viewing project data in SLIM-Collaborate requires a user to log into Collaborate with a username and password. SLIM-Collaborate provides an internal authentication method to verify login credentials. Customers may optionally use a corporate directory server or a custom authentication method to store and verify passwords.

Passwords are not saved in plain text in the SLIM-Collaborate user database nor displayed on the screen during user logins. If the username and password do not match, a single error message that does not give any indication of whether the username was found will appear.

If the customer uses the SLIM-Collaborate internal authentication, encrypted passwords are stored in the SLIM-Collaborate database. This method uses SHA-512 encryption and the “salted hash” mechanism to protect against attempts by an intruder to guess a password. Each user is responsible for choosing or changing their password via the “My Preferences” menu item on the main Collaborate site.

Site Administrators may deactivate a user’s profile to prevent that user from logging in. Only site administrators can deactivate or reactivate a user.

Users who forget their password can enter their username and request a password reset from the link provided on the login page. If the username matches an existing user profile, a password reset link is sent to the email address in the user’s profile. The password reset link is valid for 30 minutes and can be used just once during that period. To discourage unauthorized reset requests, users who mistype their usernames when requesting a password reset are not informed of the error. This helps prevent unauthorized users from gaining information useful for stealing usernames. The Site Administrator can also reset user passwords directly in the Admin site.

If all Site Administrators are locked out of the system during the valid license period, QSM can supply an authorized administrator a one-time use key to log in and correct the problem. Once used, the key becomes invalid so other users cannot use it later. Before sending this “unlock” key, QSM reserves the right to employ all reasonable methods to verify that the request is legitimate and authorized. This may include contacting multiple from the requesting company for verification.

Customizing Application Security

Clients can add additional security to SLIM-Collaborate through plug-ins that add multi-factor authentication or enhance password integrity using customizable rules. Please contact QSM support for more information on existing or new custom plug-ins.

Authorization

Each SLIM-Collaborate user profile has a default role with permissions set by the Site Admin. Only projects which the user has permission to access are visible to that user. Site Administrators have permission to all projects except for Private Projects, to which the Admin user has not been granted explicit access. Actions available on a project depend on the user's role and permissions as well as project-level access settings. See the SLIM-Collaborate 4.0 online documentation for further details.

Cookies

All cookies have a secure attribute, ensuring that requests are sent over secure channels. The HttpOnly attribute has been added to necessary cookies to prevent attacks. The SameSite attribute has been set on all cookies to mitigate the risk of cross-site request forgery.

Web Vulnerabilities

SLIM-Collaborate follows industry standards to protect against web vulnerabilities such as cross-site scripting, cross-site request forgery, SQL injection, and LDAP injection.

Cross-site Scripting

All user data is encoded before it is displayed in the browser, preventing malicious users from executing scripts that are not part of the SLIM-Collaborate code. This prevents cross-site scripting attacks from any source, including data entered within SLIM-Collaborate, data imported into SLIM-Collaborate by other authorized techniques, or any other way data is entered into the SLIM-Collaborate database.
Data and text entered directly in SLIM-Collaborate is also validated using Microsoft IIS page validation to check for malicious scripts before it is saved. Users who attempt to enter invalid data are redirected to an error page.

Cross-site request forgery

All requests that change data are checked to be sure the request came from SLIM-Collaborate. SLIM-Collaborate uses two hidden, encoded tokens that must match site and session specific cookie values. Requests that do not match are rejected. Also, the SameSite attribute has been set for all cookies.

SQL Injection

To mitigate threats from SQL injection, SLIM-Collaborate uses a data layer framework that generates parameterized SQL queries.

LDAP Injection

If SLIM-Collaborate is configured to use the optional LDAP user authentication, the password authentication is checked through a bind call rather than a search for the password value, so it is not susceptible to LDAP injection.

Known Vulnerabilities

SLIM-Collaborate release 4.0 contains no known vulnerabilities. All frameworks, libraries, modules, and components have been scanned for known vulnerabilities at the time of release.

Log4j Vulnerability

SLIM-Collaborate release 4.0 does not use Java or the log4j software library. Neither Java nor log4j are installed in the SLIM-Collaborate production environment, so SLIM-Collaborate is not vulnerable to Apache log4j RCE (remote code execution) exploits.

Privacy

Personally Identifiable Information (PII) is encrypted, at rest, in the SLIM-Collaborate database. To protect user privacy and support the European Union’s General Data Protection Regulation (GDPR), the following fields are encrypted in the SLIM-Collaborate database:

- Login name
- E-mail address
- First Name, Middle Initial, Last Name (and the derived Display Name)
- Phone Number

Existing PII is automatically encrypted whenever the SLIM-Collaborate application is upgraded to a new version.

Custom user data stored by a custom authentication plugin may be updated in bulk when the plugin is first loaded. For example, if the current version of a custom authentication plugin stores PII that is not encrypted, and a new version of the authentication plugin is installed that encrypts PII, the plugin will encrypt all PII when SLIM-Collaborate is restarted and the new plugin is loaded for the first time.

A custom authentication plugin can store and read private configuration data that is not exposed in the Admin site. When SLIM-Collaborate is started and the plugin is first loaded, the plugin can add or modify private configuration and other data. For example, the plugin can store a version number to allow the plugin to detect when a new version is installed and - based on the version number - update custom user data once, then store the new version number.

QSM Hosted Sites

To secure sensitive communications between the user's computer and SLIM-Collaborate, all exchanges in transit are protected over secured sockets layer version 3.0 high-grade encryption (TLS_RSA_WITH_AES_128_CBC_SHA, 128-bit keys) and SHA-256 with RSA encryption.

The hosted environment is secured with logging and monitoring systems as well as managed firewalls, a threat management system, and DDoS mitigation services. In the event a website requires data restoration, full database remote backups are performed on a nightly basis through a secure and encrypted gateway.

Client (On Premise) Hosting

Connection security will depend on the configuration of the client's chosen hosting environment, however the https protocol is required. For Collaborate sites hosted by the client, the access and encryption settings needed to protect data are custom configurable.

OWASP Top 10 Threat Prevention

Risk	Vulnerabilities[1]	Prevention
Broken Access Control	Restrictions on what authenticated users are allowed to do are often not properly enforced. Attackers can exploit these flaws to access unauthorized functionality and/or data.	Access control checks are implemented for each user-accessible object.
Cryptographic Failures (formerly Sensitive Data Exposure)	Sensitive data may be compromised without extra protection, such as encryption at rest or in transit, and requires special precautions when exchanged with the browser.	All sensitive data is encrypted at rest and in transit.
Injection	Injection flaws, such as SQL, NoSQL, OS, and LDAP injection, occur when untrusted data is sent to an interpreter as part of a command or query. Cross-site Scripting is now part of this category. XSS flaws occur whenever an application includes untrusted data in a new web page without proper validation or escaping or updates an existing web page with user-supplied data using a browser API that can create HTML or JavaScript.	All incoming requests are checked to determine their trustworthiness. A Content Security Policy is in place, response headers are sent, outputted data is encoded, and input data is filtered.
Insecure Design	This category focuses on risks related to design flaws and incorporates threat modeling, secure design patterns/principles, and reference architectures.	Design and coding practices have been reviewed and revised to make application design more secure. SLIM-Collaborate employs secure development lifecycle and design patterns and separates application tiers.

Security Misconfiguration	This is commonly a result of insecure default configurations, incomplete or ad hoc configurations, open cloud storage, misconfigured HTTP headers, and verbose error messages containing sensitive information. XML External Entities is now part of this category.	There is secure separation between components and tenants with segmentation.
Vulnerable and Outdated Components	If a vulnerable component is exploited, such an attack can facilitate serious data loss or server takeover. Applications and APIs using components with known vulnerabilities may undermine application defenses and enable various attacks and impacts.	No components with known vulnerabilities are in use.
Identification/Authentication Failures (formerly Broken Authentication)	Functions related to authentication and session management are often implemented incorrectly, allowing attackers to compromise passwords, keys, or session tokens, or to exploit other implementation flaws to assume other users' identities temporarily or permanently.	Strong authentication and session management controls are in place. Where additional security is desired, clients can customize application security through plug-ins or adjustment to internal authentication settings.
Software and Data Integrity Failures	This category focuses on assumptions related to software updates, critical data, and CI/CD pipelines without verifying integrity. Insecure Deserialization is now part of this category. Insecure deserialization often leads to remote code execution that can be used to perform replay attacks, injection attacks, and privilege escalation attacks.	SLIM-Collaborate ensures libraries and dependencies are trusted and that a review process for code changes is in place. Certain types of objects are restricted from being deserialized and untrusted objects are not deserialized at all.
Security Logging and Monitoring Failures (formerly Insufficient Logging and Monitoring)	Insufficient logging and monitoring, coupled with missing or ineffective integration with incident response, allows attackers to further attack systems, maintain persistence, pivot to more systems, and tamper, extract, or destroy data.	Detailed logging is in place to monitor attacks in progress.

Server-Side Request Forgery	SSRF flaws occur whenever a web application fetches a remote resource without validating the user-supplied URL. It allows an attacker to coerce the application to send a crafted request to an unexpected destination, even when protected by a firewall, VPN, or another type of network access control list (ACL).	All requests that change data are checked to ensure the request came from within SLIM-Collaborate. SLIM-Collaborate uses two hidden, encoded tokens which must match site and session specific cookie values and requests that do not match are rejected. Also, the SameSite attribute has been set for all cookies.

References:

[1] OWASP. OWASP Top Ten. Retrieved from https://owasp.org/Top10// on June 24, 2022.