Security Requirements Status

How to understand the *NEW!* security requirements table

The new security requirement table will be broken down into the following parts:

- "*Type of requirement*" indicates the requirement category, such as 'data protection' or 'application security.' This is meant to indicate the security areas or types of security practices that developers should focus on.
- "*Security Requirement*" lists the actual requirement.
- The next four columns – "*Connect*," "*Forge with data egress*," "*Forge*," and "*Power-Ups*" – apply this requirement to these various app types. In these sections, we list implementation details on how apps **must** or **must not** apply these requirements to their apps.
  - As you can see, all "*Forge with data egress*," "*Forge*," and "*Power-Ups*" are tagged as NEW since we've never explicitly listed requirements for these types of apps before.

Table Key

Security Requirements

- UNCHANGED = This is an existing security requirement
- UPDATED = This is an existing security requirement with new language
- NEW = This is a new security requirement
- FRAMEWORK SUPPORTED = This requirement is supported by either the ACE or ACSB frameworks provided by Atlassian.
  - **NOTE:** Framework Supported does not mean that these requirements are automatically met if you are using these frameworks.

App Types

- APPLICABLE = This security requirement applies to these types of apps
- PLATFORM PROVIDED = Atlassian's Forge platform covers for this security requirement
- NOT YET APPLICABLE = This security requirement is not yet applicable to these types of apps.

***NEW!* Security Requirements Table**

Security Requirements					
TYPE OF REQUIREMENT	**SECURITY REQUIREMENT**	**CONNECT**	**FORGE WITH DATA EGRESS** NEW	**FORGE** NEW	**TRELLO APPS (POWER-UPS)** NEW
Authentication & Authorization	UPDATED 1. An application must authenticate and authorize every request on all endpoints exposed. *Anonymous access to application endpoints and resources can be allowed in scenarios where it is needed.*	APPLICABLE	APPLICABLE 1. An application must default to using `asUser()` when performing an operation on behalf of the user. 2. Before making calls using `asApp()`, you must verify the expected permissions (for example, from product context) with the permissions REST APIs.	APPLICABLE 1. An application must default to using `asUser()` when performing an operation on behalf of the user. 2. Before making calls using `asApp()`, you must verify the expected permissions (for example, from product context) with the permissions REST APIs.	APPLICABLE

		1. NEW FRAMEWORK SUPPORTED An application must always use JWT as the authentication type to validate the application identity and the integrity of the request. a. Read about implementing authentication here: https://developer.atlassian.com/cloud/confluence/understanding-jwt/ b. *Exception: It is permissible to set authentication type to* `None` *in the app descriptor if all of the endpoints in the descriptor serve only static content.* 2. UPDATED FRAMEWORK SUPPORTED A JWT token must be validated on the server-side for every authenticated request. Validate all user permissions to ensure that only permitted users can execute actions within an application. a. Read about implementing authorization here: https://developer.atlassian.com/developer-guide/connect-app-authorization/#approaches-for-authorization. 3. NEW FRAMEWORK SUPPORTED An application must always check for JWT token expiration time. Expired tokens must be rejected.			1. Secure the communication between the Power-Up and your server by using `t.jwt()` to include JWT token with requests. a. Read about implementing authentication here: https://developer.atlassian.com/cloud/trello/power-ups/client-library/t-jwt/ 2. JWT tokens must be properly validated on the server-side to ensure that the JWT signature is (a) valid, (b) not expired, (c) it comes from your Power-Up, and (d) is a particular Trello user. 3. When creating an API key for use with Trello's REST API, set the appropriate allowed origins for your application. a. Read about allowed origins here: https://developer.atlassian.com/cloud/trello/guides/rest-api/authorization/#allowed-origins

		4. NEW FRAMEWORK SUPPORTED An application must always validate install/uninstall lifecycle requests. 5. NEW An application must not accept context JWTs in module or lifecycle endpoints.			
Data Protection	UPDATED 1. Any Atlassian End User Data: • stored by an application **outside of the Atlassian product or users' browser** must ensure full disk encryption at-rest. • accessed by an application or a service should be authenticated and authorized appropriately. For more information about Atlassian End User data, reference our Atlassian Developer Terms: https://developer.atlassian.com/platform/marketplace/atlassian-developer-terms/#atlassian-developer-terms or https://developer.atlassian.com/cloud/trello/developer-terms/	APPLICABLE	APPLICABLE	PLATFORM PROVIDED	APPLICABLE

UPDATED An application must use TLS version 1.2 (or higher) to encrypt all of its traffic, and enable HSTS with a minimum age of one year. *We highly recommend only allowing clients to connect using ciphersuites listed under the "Intermediate compatibility" section of the latest Mozilla's Server Side TLS guidance . These ciphersuites provide a good balance of security and compatibility with older clients. If you choose to only implement a subset of the ciphersuites, you should thoroughly test any production changes to avoid customer impact.*	**APPLICABLE** 1. **NEW** When possible, add the 'includeSubDomain ' directive in the HSTS policy definition.	**APPLICABLE**	**PLATFORM PROVIDED**	**APPLICABLE** 1. When possible, add the 'includeSubDomain ' directive in the HSTS policy definition.
NEW 1. An application must follow the "Principle of Least Privilege," when requesting app scopes. This means that an application should only request scopes required to perform its intended functionality, and nothing more.	**APPLICABLE** Read about Connect scopes here: https://developer.atlassian.com/cloud/jira/platform/scopes-for-connect-apps/	**APPLICABLE** Read about Forge scopes here:https://developer.atlassian.com/platform/forge/manifest-reference/permissions/#product-scopes	**APPLICABLE** Read about Forge scopes here:https://developer.atlassian.com/platform/forge/manifest-reference/permissions/#product-scopes	**APPLICABLE** 1. When making use of t.set() from the Power-Up client library, the `shared` scope should *never* be used to store data that should be kept secret. Please note that data stored in the `shared` scope is readable by any user who can view the board. Read about Power-Up scopes here: https://developer.atlassian.com/cloud/trello/power-ups/client-library/getting-and-setting-data/#scopes

	UPDATED 1. An application must securely store and manage secrets, which include JWTs, OAuth tokens, Trello tokens, sharedSecret, API keys, and encryption keys. They cannot be stored in places that are easily accessible. Example of places include but not limited to: • Source code • URL strings • Referer headers • Application logs • Code repositories, such as Bitbucket and Github	**APPLICABLE** 1. **NEW** An application must never store secrets or authorization information in Entity properties / Content Properties.	**PLATFORM PROVIDED** 1. Use encrypted environment variables and storage.setSecret to store secrets in your app. Read about secure data storage on Forge here: https://developer.atlassian.com/platform/forge/storage/	**PLATFORM PROVIDED** 1. Use encrypted environment variables and storage.setSecret to store secrets in your app. Read about secure data storage on Forge here: https://developer.atlassian.com/platform/forge/storage/	**APPLICABLE** 1. Use `t.storeSecret` to store secrets such as oauth token. Read about secure data storage with Power-Ups here: https://developer.atlassian.com/cloud/trello/power-ups/client-library/managing-secrets/#t-storesecret-key--data-
Application Security	**UPDATED** 1. An application must maintain and securely configure domains where the application is hosted.	**APPLICABLE** The following needs to be true for the `baseUrl` listed in an application's app descriptor: 1. **UNCHANGED** An application must maintain control of each domain. 2. **NEW** An application owner must maintain valid TLS certificates of the domains where an application is hosted, and the domain must be signed by a trusted Certificate Authority. 3. **UNCHANGED** An application's DNS configuration for subdomains must reference services that are in use.	**APPLICABLE** The following needs to be true when an application makes calls to domains owned by the application owner: 1. An application must maintain control of each domain. 2. An application owner must maintain valid TLS certificates of the domains where an application is hosted, and the domain must be signed by a trusted Certificate Authority. 3. An application's DNS configuration for subdomains must reference services that are in use.	**NOT APPLICABLE**	**APPLICABLE** The following needs to be true for the domain listed in an application's `iFrameConnectorURL`: 1. An application must maintain control of each domain. 2. An application owner must maintain valid TLS certificates of the domains where an application is hosted, and the domain must be signed by a trusted Certificate Authority. 3. An application's DNS configuration for subdomains must reference services that are in use.

UPDATED 1. When applicable, an application must enable security headers and cookie security attributes.	**APPLICABLE** 1. **NEW** An application must set Content Security Policy (CSP) header. `Content-Security-Policy: script-src <source>` a. Do not use un`safe-inline` or un`safe-eval` directives in s`cript-src` when possible. This will make the policy ineffective against XSS vulnerabilities. 2. **UNCHANGED** An application must implement the Referrer-Policy header. a. The header must not be configured to `no-referrer-when-downgrade` or `unsafe-url`. We recommend using `no-referrer` or `strict-origin-when-cross-origin`. 3. **UNCHANGED** An application must disable caching on all HTTPS pages that contain sensitive data by using “no-cache” and “no-store” instead of “private” in the cache control header. 4. **UNCHANGED** For session-related cookies, an application must set `HttpOnly` and `Secure` attributes when sending Set-Cookie headers.	**PLATFORM PROVIDED**	**PLATFORM PROVIDED**	**APPLICABLE** 1. An application must set Content Security Policy (CSP) header. `Content-Security-Policy: script-src <source>` a. Do not use un`safe-inline` or un`safe-eval` directives in s`cript-src` when possible. This will make the policy ineffective against XSS vulnerabilities. 2. An application must implement the Referrer-Policy header. a. The header must not be configured to `no-referrer-when-downgrade` or `unsafe-url`. We recommend using `no-referrer` or `strict-origin-when-cross-origin`. 3. An application must disable caching on all HTTPS pages that contain sensitive data by using “no-cache” and “no-store” instead of “private” in the cache control header. 4. For session-related cookies, an application must set `HttpOnly` and `Secure` attributes when sending Set-Cookie headers.

	NEW 1. An application must validate and sanitize all untrusted data and treat all user input as unsafe to mitigate injection-related vulnerabilities. Untrusted data is any input that can be manipulated to contain a web attack payload.	**APPLICABLE** 1. **UNCHANGED** **FRAMEWORK SUPPORTED** An application must validate the **q sh** to prevent URL tampering. 2. **NEW** An application using template engines must not use dangerous functions or modules that lead to arbitrary code execution. When there is a business requirement to use these functions, sandbox it inside an isolated, locked-down environment.	**APPLICABLE**	**APPLICABLE**	**APPLICABLE** 1. You may either use provided `t.safe(rawHTML)` or your framework of choice to sanitize the content before insertion into the DOM.
	UPDATED 1. An application must not use versions of third-party libraries and dependencies with known critical or high vulnerabilities. When vulnerabilities in these libraries and dependencies are discovered, an application owner must remediate them as quickly as possible.	**APPLICABLE**	**APPLICABLE**	**APPLICABLE**	**APPLICABLE**

Privacy	**UPDATED** 10. An application must not collect or store credentials belonging to Atlassian user accounts such as user passwords or user API tokens. *If an app is currently requesting or storing Atlassian API tokens in order to access a REST API that does not currently support authentication from apps, the app developer can notify Atlassian and receive a temporary waiver for this requirement while Atlassian makes the proper changes to our API to support authenticated requests from apps. Once the API supports approved authentication methods, the app developer will be given a reasonable amount of time to migrate away from using Atlassian API tokens. This requirement does not prohibit apps from storing credentials used to access non-Atlassian applications*.	**APPLICABLE**	**APPLICABLE**	**APPLICABLE**	**APPLICABLE**
Vulnerability Management	**UPDATED** 11. You must know, understand, and follow our Security Bug Fix Policy. The following page explains Atlassian's Security Bug Fix Policy for Marketplace Apps:https://developer.atlassian.com/platform/marketplace/security-bugfix-policy/	**APPLICABLE**	**APPLICABLE**	**APPLICABLE**	**APPLICABLE**
	NEW 12. You must notify Atlassian of all security incidents via DEVHELP. The following guide explains how to handle a security incident: https://developer.atlassian.com/platform/marketplace/app-security-incident-management-guidelines/	**APPLICABLE** 1. **NEW** If the security incident involves sharedSecret leakage, immediately notify Atlassian and request to rotate the sharedSecret through DEVHELP within 24 hours.	**APPLICABLE**	**APPLICABLE**	**APPLICABLE**

	NEW 13. You must identify at least one email as a security contact and have them create an account on **ecosystem.atlassian.net** so that they are notified about vulnerabilities in the app via Atlassian Marketplace Security (AMS) tickets. The following guide explains how to get access to AMS: https://developer.atlassian.com/platform/marketplace/vulnerability-review-practices-for-atlassian-partners/ *Please note that an admin can also be listed as a security contact.*	**APPLICABLE**	**APPLICABLE**	**APPLICABLE**	**NOT YET APPLICABLE**