For Mammoth – security check for Confluence

Table of contents:

1. What is this app
2. How to use it
3. Support contacts
4. Data security
5. Mitigations explained
What is this app?

This app was designed with all modern security threats affecting Confluence instances in mind from an enterprise security engineer point of view.

There are many places in Confluence where a single misconfiguration can lead to such vulnerabilities as SSRF, XSS, CSRF, RCE, etc.

I decided to make a plugin which takes care for an admin and tells him where things can go wrong in current configuration, so he could be aware of it and respond in time.
How to use it?

Once you install the plugin from Marketplace and set up license, navigate straight to /plugins/servlet/security-auditor.

There you will see a tabs with security checks and its’ results.
Data security

This app in order to perform security checks, does accept following information via HTTP:

- Base url of Confluence installation
- Emails of users of confluence administrator group

However, this data is not being stored and is not being a subject for a processing in any other ways than described.
Mitigations explained

Currently we have 3 mitigations methods. These are:

1. CSRF protection based on Origin/Referer check
2. XSS protection for /rest/* endpoints with CSP
3. Authentication layer

They work pretty much simple yet effective.

CSRF protection first checks if request is POST, then it tries to compare Origin or Referer header to Confluence instance base url. If no Origin or Referer headers passed or values are not matching base url, the check flow looks for X-* headers in request, like X-Test: test or checks if Authorization header was passed, if any of these two matches, then request considered complicated and CSRF check passed. Otherwise X-CSRF-Violation header is returned in response and current authentication data getting reset, preventing CSRF.

XSS protection is based on https://developer.mozilla.org/en-US/docs/Web/HTTP/Headers/Content-Security-Policy/sandbox and only working for /rest/* endpoints where scripts are not suppose to be executed anyway.
Authentication layer prevents 3rd party plugin vulnerabilities when developer forgot to check permissions. Any unauthenticated access will be denied for endpoints that plugins use (/rest/* and /plugins/*).
Support

If any feature requests or bug report, feel free to contact me via email: ivan.rumak57@gmail.com