Deployment Guide

Security Testing with ZAProxy

4/16/2025

Security Testing with ZAProxy

Contents

Genesys performs security testing with OWASP Zed Attack Proxy (ZAProxy) to make sure the Genesys Co-browse solution is invincible to known attacks.

> **Tip**
>
> Genesys aims to test against the latest version of ZAProxy available at the time of a release. For your convenience, this version is shipped together with the Co-browse solution. For instructions on how to obtain and use the ZAProy, see ZAProxy

> **Important**
>
> Some issues reported by ZAProxy security are false positives. See ZAProxy False Positives.

ZAP Overview

The ZAProxy is an easy-to-use, integrated penetration testing tool for finding vulnerabilities in websites and web applications.

Among others, ZAProxy supports the follow methods for penetration security testing:

- passive scan
- active scan

Genesys uses both methods.

Passive Scan Overview

ZAP is an Intercepting Proxy. It allows you to see all of the requests made to a website/web app and all of the responses received from it. For example, you can see AJAX calls that might not otherwise be obvious.

Once set up, ZAP automatically passively scans all of the requests to and responses from the web application being tested.

While mandatory use cases for the application that is being tested are followed (either manually or automatically), ZAProxy analyzes the requests to verify the usual operations are safe.

Active Scan Overview

Active scanning attempts to find potential vulnerabilities by using known web attacks against the selected targets. Active scanning is an attack on those targets. ZAProxy emulates known attacks when active mode is used.

Through active scanning, Genesys Co-browse is verified against the following types of attacks:

- **Spider attack** — Automatically discovers all URL links found on a web resource, sends requests, and analyzes results (including src attributes, comments, low-level information disclosure, and so on).
- **Brute browsing** (based on the Brute Force technique) — Systematically makes requests to find secure resources based on known (commonly used) rules. For example, backup, configuration files, temporary directories, and so on.
- **Active scan** — Attempts to perform a predefined set of attacks on all resources available for the web resource. You can find the default set of rules here.
- **Ajax spider** — Automatically discovers web resources based on presumed rules of AJAX control (JS scripts investigation, page events, common rules, dynamic DOM, and so on).

Important

Requests to other web applications must be excluded from scanning in order to see a report for a particular web application.

Important

Web applications that are being tested should be started on the local box because some types of verification (like active scanning) can be forbidden by network administrators.

False Positives

Some issues reported by ZAProxy security testing are not actual vulnerabilities:

- High risk security alert "Remote file inclusion".

  To allow certain types of dynamic content synchronization, Co-browse may proxy some of the website's static assets (for example, CSS files). The response content is only interpreted by browsers as a corresponding asset (like CSS), because it is retrieved through according means (for example, in case of CSS through stylesheet links). Illegible assets are skipped. The source for a proxied asset is always the web site page itself, which is by definition a legitimate resource. To limit access to Co-browse resource proxy mechanism, use the `allowedExternalDomains` option.

- Medium risk security alert "Secure page browser cache" and Low risk security alert "Incomplete or no cache-control and pragma HTTPHeader set".

  As mentioned above, Co-browse may proxy some of the website's static assets (like CSS). Browser side caching of such resources is determined by original servers that own those resources and are responsible for proper caching. In other words, Co-browse will

just serve the same headers that the original website serves. In most cases, such static assets do not contain any sensitive information and can be safely cached. However, if you decide to disable all caching for these resources, you can do this on Co-browse side using the `dislabeCaching` option, without the need to modify headers on your website's side. Note that this may increase traffic load for both end users (if they opt to Co-browse), and agents.

- Medium level security alert "X-Frame-Options header not set".

  Co-browse JavaScript by design works as a third-party add-on to another web site. Moreover, it can be (and usually is) loaded from another domain and must operate in iframes while the X-Frame-Options header is specifically designed to disallow that. To prevent other websites from using your Co-browse deployment, use the `allowedOrigins` configuration option (it will not remove the "X-Frame-Options" alert).

- Low risk security alert "Cookie set without HttpOnly flag".

  This security alert means that the cookie can be accessed by JavaScript, which is a security issue if the cookie is a session cookie that can be used to hijack the session. However, the reported cookie is not such a cookie and will not allow anyone to hijack a Co-browse session.

References

If you want to examine your website against vulnerabilities in a similar way, refer to the OWASP Zed Attack Proxy Project or additional documentation to learn about security testing with ZAP.