

Security Test Using ZAP

3/17/2021

Joseph Sepe

After first run of my ZAP tests I had a few Medium Alerts. The rest were low and I believe some had to do with the IIS web server Visual Studio is running. The low ones that had to deal with IIS don't really seem necessary to fix since I'm not running IIS on my server when I'm hosting my project. Here's the first run showing my medium x-frame-options-header not set.

The screenshot displays the OWASP ZAP 2.10.0 application window. The top menu bar includes File, Edit, View, Analyse, Report, Tools, Import, Online, and Help. The main interface is divided into several panes:

- Left Pane:** Contains 'Contexts' (Default Context) and 'Sites' (https://update.googleapis.com, https://localhost:44380, https://accounts.google.com).
- Top Right Pane:** Shows the 'Quick Start' tab with 'Request' and 'Response' sub-tabs. The 'Response' tab is active, displaying an HTTP/1.1 200 OK response from localhost:44380. The response headers include Content-Type: text/html; charset=utf-8, Server: Microsoft-IIS/10.0, X-Powered-By: ASP.NET, Date: Wed, 17 Mar 2021 19:31:42 GMT, and Content-Length: 3124. The body shows an HTML document with a title 'MADWORDS - madwords' and several links to CSS files.
- Bottom Left Pane:** Displays a list of alerts. The 'Alerts (8)' section is expanded, showing 'X-Frame-Options Header Not Set (5)' as the primary alert. Other alerts include 'Cookie Without Secure Flag', 'Incomplete or No Cache-control and Pragma HTTP Header Set (10)', 'Server Leaks Information via "X-Powered-By" HTTP Response Header Field(s)', 'X-Content-Type-Options Header Missing (12)', 'Information Disclosure - Suspicious Comments (2)', 'Loosely Scoped Cookie (3)', and 'Timestamp Disclosure - Unix (2)'.
- Bottom Right Pane:** Provides details for the selected alert, 'X-Frame-Options Header Not Set'. It shows the URL as https://localhost:44380/, the risk as Medium, and the confidence as Medium. The parameter is X-Frame-Options, and the attack is described as 'X-Frame-Options header is not included in the HTTP response to protect against "ClickJacking" attacks.' The solution suggests setting the X-Frame-Options header on all web pages returned by the site to protect against ClickJacking attacks. A reference link is provided: https://developer.mozilla.org/en-US/docs/Web/HTTP/Headers/X-Frame-Options.

The status bar at the bottom indicates 'Alerts 0 1 4 3 Primary Proxy: localhost:8080'.

Following the guide linked below, I set the x-frame-options as SAMEORIGIN inside my startup.cs in the configuration method. This prevents ClickJacking attacks.

<https://dotnetcoretutorials.com/2017/01/08/set-x-frame-options-asp-net-core/>

Here's the run after adding the SAMEORIGIN code

The screenshot displays the OWASP ZAP 2.10.0 interface. The top menu bar includes File, Edit, View, Analyse, Report, Tools, Import, Online, and Help. The main toolbar contains icons for various actions like adding sites, contexts, and launching the browser. The left sidebar shows a tree view with 'Contexts' (Default Context) and 'Sites' (https://update.googleapis.com, https://localhost:44380, https://accounts.google.com). The right pane shows the 'Quick Start' tab with a 'URL to explore' field set to 'https://localhost:44380', an 'Enable HUD' checkbox checked, and a dropdown for 'Explore your application' with options 'Launch Browser' and 'Chro...'. The bottom section shows the 'Alerts' list with 7 alerts, including 'Cookie Without Secure Flag', 'Incomplete or No Cache-control and Pragma HTTP Header Set (9)', 'Server Leaks Information via "X-Powered-By" HTTP Response Header Field(s) (21)', 'X-Content-Type-Options Header Missing (12)', 'Information Disclosure - Suspicious Comments (2)', 'Loosely Scoped Cookie (5)', and 'Timestamp Disclosure - Unix (2)'. The status bar at the bottom indicates 'Alerts 0 0 4 3' and 'Primary Proxy: localhost:8080'.

OWASP ZAP - OWASP ZAP 2.10.0

File Edit View Analyse Report Tools Import Online Help

Standard Mode

Sites +

Quick Start Request Response +

<

This screen allows you to launch the browser of your choice so that you can explore your application.

The ZAP Heads Up Display (HUD) brings all of the essential ZAP functionality into your browser.

URL to explore:

Enable HUD: ☒

Explore your application:

You can also use browsers that you don't launch from ZAP, but will need to configure them to work with ZAP.

History Search Alerts Output WebSockets Active Scan +

Alerts (7)

- Cookie Without Secure Flag
 - GET: https://localhost:44380/Account/Login
- Incomplete or No Cache-control and Pragma HTTP Header Set (9)
 - GET: https://localhost:44380/
 - GET: https://localhost:44380/Account/Login
 - GET: https://localhost:44380/api/MadwordTemplate
 - GET: https://localhost:44380/css/site.css
 - GET: https://localhost:44380/lib/bootstrap/dist/css/bootstrap.min.css
 - GET: https://localhost:44380/Madword
 - GET: https://localhost:44380/Madword/Comment?madwordId=5
 - GET: https://localhost:44380/Madword/Create
 - GET: https://localhost:44380/Madword/TopRated
- Server Leaks Information via "X-Powered-By" HTTP Response Header Field(s) (21)
- X-Content-Type-Options Header Missing (12)
- Information Disclosure - Suspicious Comments (2)
- Loosely Scoped Cookie (5)
- Timestamp Disclosure - Unix (2)

Alerts 0 0 4 3 Primary Proxy: localhost:8080

One thing I did notice was that my form is validating data on the client side. This has to do with the way my program is working at the moment. Somebody can circumvent the required fields and type in whatever they want in the form to post whatever text they want in the create a MADWORD form. I understand this is a problem in terms of breaking my game, but I do not see any security threats here since ASP.net still prevents hackers from getting in through that form. I plan to fix this bug in a future release of my game. Here's an example

