Report

Vulnerability Report of Website: http://zero.webappsecurity.com/

My intentions were only to detect vulnerabilities in the website rather than to harm it. There are some issues in your website which i have found and are listed as below:

- **Out-of-dated version of Apache Tomcat, OpenSSL and version disclosure:** The website is using outdated versions of server (Apache) and OpenSSL which is as:

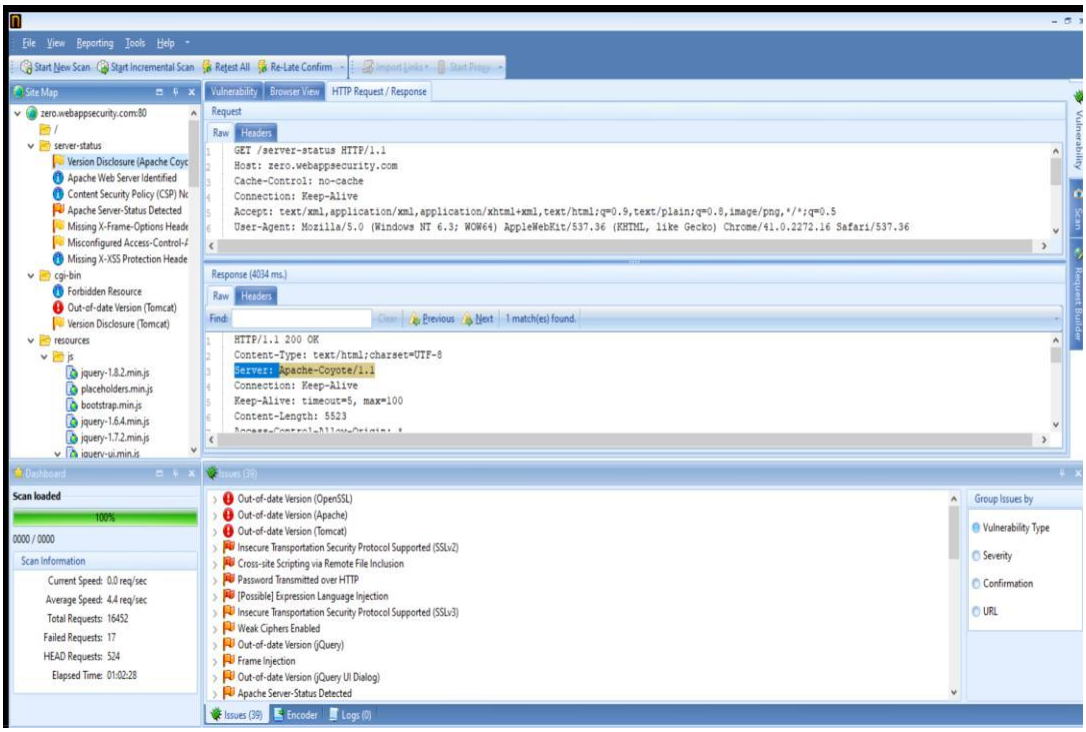
  Server: Apache/2.2.6 (Win32) mod_ssl/2.2.6 OpenSSL/0.9.8e mod_jk/1.2.40

  Version: Apache-Coyote/1.1

  Impact: An attacker might use the disclosed information to harvest specific security vulnerabilities for the version identified.

  Solution: Configure your web server to prevent information leakage from the server header of its HTTP response.

Netsparker Screenshot:



- **Content Security Policy (CSP) Not Implemented**: CSP is an added layer of security that helps to mitigate cross-site scripting attacks.

CSP can be enabled instructing the browser with a Content-Security-Policy directive in a response header:

Script loading example using meta tag:

Script: <meta http-equiv="Content-Security-Policy" content="script-src 'self';">

Using the script listed above, you can restrict script loading only to the same domain. It will also restrict inline script executions both in the element attributes and the event handlers.

Impact: There is no direct impact of not implementing CSP on your website. However, if your website is vulnerable to a Cross-site Scripting attack CSP can prevent successful exploitation of that vulnerability. By not implementing CSP you'll be missing out this extra layer of security.

Solution: Enable CSP on your website by sending the `Content-Security-Policy` in HTTP response headers that instruct the browser to apply the policies you specified.

- **SameSite Cookie Not Implemented**: Cookies are typically sent to third parties in cross origin requests. This can be abused to do CSRF attacks. Recently a new cookie attribute named *SameSite* was proposed to disable third-party usage for some cookies, to prevent CSRF attacks.

Same-site cookies allow servers to mitigate the risk of CSRF and information leakage attacks by asserting that a particular cookie should only be sent with requests initiated from the same registrable domain.

Solution: The server can set a same-site cookie by adding the SameSite=... attribute to the Set-Cookie header:

Set-Cookie: key=value; SameSite=strict

There are two possible values for the same-site attribute:

- Lax
- Strict

In the strict mode, the cookie is not sent with any cross-site usage even if the user follows a link to another website. Lax cookies are only sent with a top-level get request.

- **X-XSS Protection Header is Missing**: There is a missing X-XSS-Protection header which means that this website could be at risk of a Cross-site Scripting (XSS) attacks.

Solution: Add the X-XSS-Protection header with a value of "1; mode= block".

→ `X-XSS-Protection: 1; mode=block`