Task 4: Web Application Penetration Testing with OWASP ZAP

HAFIZ NAVEED UDDIN

KAIRIZ CYBER TECHNOLOGIES

Vulnerability Scan Complete Report:

1. ABSENCE OF ANTI CSRF-TOKENS (83)

- **URL:** https://www.google.com
- **Risk:** Medium
- **Confidence:** Low
- **Parameter:**
- **Attack:**
- **Evidence:** `<form action="/search" autocomplete="off" method="GET" role="search">`
- **CWE ID:** 352
- **WASC ID:** 9
- **Source:** Passive (10202 - Absence of Anti-CSRF Tokens)
- **Input Vector:**
- **Description:** No known Anti-CSRF token [anticsrf, CSRFToken, etc.] was found in the following HTML form.
- **Solution:**
    - **Architecture and Design:** Use vetted libraries or frameworks, anti-CSRF packages such as OWASP CSRFGuard.
    - **Implementation:** Ensure the application is free of cross-site scripting issues, generate a unique nonce for each form, check the HTTP Referer header.
- **Reference:** https://cheatsheetseries.owasp.org/cheatsheets/Cross-Site_Request_Forgery_Prevention_Cheat_Sheet.html, https://cwe.mitre.org/data/definitions/352.html

2. CSP-Wildcard Directive (313)

- **URL:** https://www.google.com/search
- **Risk:** Medium
- **Confidence:** High
- **Parameter:** Content-Security-Policy
- **Attack:**
- **Evidence:** `object-src 'none';base-uri 'self';script-src 'nonce-WJmpei9QRAupI8ho1Aq3nw' 'strict-dynamic' 'report-sample' 'unsafe-eval' 'unsafe-inline' https: http:;report-uri https://csp.withgoogle.com/csp/gws/fff`
- **CWE ID:** 693
- **WASC ID:** 15
- **Source:** Passive (10055-CSP)
- **Alert Reference:** 10055-4
- **Input Vector:**
- **Description:** CSP provides a set of standard HTTP headers that allow website owners to declare approved sources of content.

- **Solution:** Ensure web server, application server, load balancer, etc. is configured to set the Content-Security-Policy header.
- **Reference:** https://www.w3.org/TR/CSP/

---

3. CSP: Script-src unsafe-eval (154)

- **URL:** https://www.google.com/search
- **Risk:** Medium
- **Confidence:** High
- **Parameter:** Content-Security-Policy
- **Attack:**
- **Evidence:** `object-src 'none';base-uri 'self';script-src 'nonce-WJmpei9QRAupI8ho1Aq3nw' 'strict-dynamic' 'report-sample' 'unsafe-eval' 'unsafe-inline' https: http:;report-uri https://csp.withgoogle.com/csp/gws/fff`
- **CWE ID:** 693
- **WASC ID:** 15
- **Source:** Passive (10055 - CSP)
- **Alert Reference:** 10055-10
- **Input Vector:**
- **Description:** CSP provides a set of standard HTTP headers that allow website owners to declare approved sources of content.
- **Solution:** Ensure web server, application server, load balancer, etc. is configured to set the Content-Security-Policy header.
- **Reference:** https://www.w3.org/TR/CSP/

---

4. CSP: Script-src unsafe-inline (156)

- **URL:** https://www.google.com/gwt/
- **Risk:** Medium
- **Confidence:** High
- **Parameter:** Content-Security-Policy
- **Attack:**
- **Evidence:** `require-trusted-type-for ‘script’`
- **CWE ID:** 693
- **WASC ID:** 15
- **Source:** Passive (10055 - CSP)
- **Alert Reference:** 10055-5
- **Input Vector:**
- **Description:** CSP provides a set of standard HTTP headers that allow website owners to declare approved sources of content.
- **Solution:** Ensure web server, application server, load balancer, etc. is configured to set the Content-Security-Policy header.

5. CSP: Style-src Unsafe-inline (313)

- **URL:** https://www.google.com/search
- **Risk:** Medium
- **Confidence:** High
- **Parameter:** Content-Security-Policy
- **Attack:**
- **Evidence:** `object-src 'none';base-uri 'self';script-src 'nonce-WJmpei9QRAupI8ho1Aq3nw' 'strict-dynamic' 'report-sample' 'unsafe-eval' 'unsafe-inline' https: http:;report-uri https://csp.withgoogle.com/csp/gws/fff`
- **CWE ID:** 693
- **WASC ID:** 15
- **Source:** Passive (10055 - CSP)
- **Alert Reference:** 10055-6
- **Input Vector:**
- **Description:** CSP provides a set of standard HTTP headers that allow website owners to declare approved sources of content.
- **Solution:** Ensure web server, application server, load balancer, etc. is configured to set the Content-Security-Policy header.
- **Reference:** https://www.w3.org/TR/CSP/, https://caniuse.com/#search=content+security+policy, https://content-security-policy.com/, https://github.com/HtmlUnit/htmlunit-csp, https://developers.google.com/web/fundamentals/security/csp#policy_applies_to_a_wide_variety_of_resources

6. Content Security Policy (CSP) Header Not Set

- **URL:** https://www.google.com
- **Risk:** Medium
- **Confidence:** High
- **Parameter:** hi
- **Attack:**
- **Evidence:**
- **CWE ID:** 693
- **WASC ID:** 15
- **Source:** Passive (10038 – Content Security Policy Header Not Set)
- **Alert Reference:** 10038-1
- **Input Vector:**
- **Description:** CSP helps to detect and mitigate certain types of attacks, including Cross-Site Scripting (XSS) and data injection attacks.
- **Solution:** Ensure web server, application server, load balancer, etc. is configured to set the Content-Security-Policy header.

- **Reference:** https://www.w3.org/TR/CSP/, https://w3c.github.io/webappsec-csp/, https://web.dev/articles/csp, https://caniuse.com/#feat=contentsecuritypolicy

---

7. Cross-Domain Misconfiguration (3)

- **URL:** https://www.google.com/maps/api/staticmap
- **Risk:** Medium
- **Confidence:** Medium
- **Parameter:**
- **Attack:**
- **Evidence:** `Access-Control-Allow-Origin: *`
- **CWE ID:** 264
- **WASC ID:** 14
- **Source:** Passive (10098- Cross-Domain Misconfiguration)
- **Alert Reference:** 10055-5
- **Input Vector:**
- **Description:** Web browser data loading may be possible due to a CORS misconfiguration on the web server.
- **Solution:** Ensure sensitive data is not available in an unauthenticated manner, configure "Access-Control-Allow-Origin" HTTP header to a more restrictive set of domains.
- **Reference:** https://vulncat.fortify.com/en/detail?id=desc.config.dotnet.html5_overly_permissive_cors_policy

---

8. Missing Anti-clickjacking Header (70)

- **URL:** https://www.google.com/landing/signout.html
- **Risk:** Medium
- **Confidence:** Medium
- **Parameter:** x-frame-options
- **Attack:**
- **Evidence:**
- **CWE ID:** 1021
- **WASC ID:** 15
- **Source:** Passive (10020- Anti-clickjacking Header)
- **Alert Reference:** 10020-1
- **Input Vector:**
- **Description:** The response does not include either Content-Security-Policy with 'frame-ancestors' directive or X-Frame-Options to protect against 'ClickJacking' attacks.
- **Solution:** Ensure Content-Security-Policy and X-Frame-Options HTTP headers are set.
- **Reference:** https://developer.mozilla.org/en-US/docs/Web/HTTP/Headers/X-Frame-Options

9. CSP: Notice (307)

- **URL:** https://www.google.com/Search
- **Risk:** Low
- **Confidence:** High
- **Parameter:** Content-Security-Policy
- **Attack:**
- **Evidence:** `object-src 'none';base-uri 'self';script-src 'nonce-WJmpei9QRAupI8ho1Aq3nw' 'strict-dynamic' 'report-sample' 'unsafe-eval' 'unsafe-inline' https: http:;report-uri https://csp.withgoogle.com/csp/gws/fff`
- **CWE ID:** 693
- **WASC ID:** 15
- **Source:** Passive (10055 - CSP)
- **Alert Reference:** 10055-3
- **Input Vector:**
- **Description:** CSP helps to detect and mitigate certain types of attacks, including Cross-Site Scripting (XSS) and data injection attacks.
- **Solution:** Ensure web server, application server, load balancer, etc. is configured to set the Content-Security-Policy header.
- **Reference:** https://www.w3.org/TR/CSP/

10. Non-storable Content (33)

- **URL:** https://www.google.com
- **Risk:** Low
- **Confidence:** Medium
- **Parameter:** Cache-Control
- **Attack:**
- **Evidence:** `no-cache, no-store, max-age=0, must-revalidate`
- **CWE ID:**
- **WASC ID:**
- **Source:** Passive (10050 - Non-storable Content)
- **Input Vector:**
- **Description:**
- **Solution:** Ensure Cache-Control HTTP header is set to public and max-age is set to a large value.
- **Reference:** https://developer.mozilla.org/en-US/docs/Web/HTTP/Headers/Cache-Control

11. Cookie with SameSite Attribute None (12)

- **URL:** https://www.google.com
- **Risk:** Low
- **Confidence:** Medium
- **Parameter:** NID
- **Attack:**
- **Evidence:** `Set-Cookie: NID`
- **CWE ID:** 1275
- **WASC ID:** 13
- **Source:** Passive (10054 - Cookie without SameSite Attribute)
- **Alert Reference:** 10054-2
- **Input Vector:**
- **Description:** A cookie has been set with its SameSite attribute set to "none", which means that the cookie can be sent as a result of a 'cross-site' request. The SameSite attribute is an effective counter measure to cross-site request forgery, cross-site script inclusion, and timing attacks.
- **Solution:** Ensure that the SameSite attribute is set to either 'lax' or ideally 'strict' for all cookies.
- **Reference:** SameSite Cookies

12. Cookie Without SameSite Attribute

- **URL:** https://www.google.com/_/ShoppingUi/gen204/?e&ei=6TGOZpGoBMSqmLAPqMiqiAg&evt=view
- **Risk:** Low
- **Confidence:** Medium
- **Parameter:** expor
- **Attack:**
- **Evidence:** `Set-Cookie: expor`
- **CWE ID:** 1275
- **WASC ID:** 13
- **Source:** Passive (10054 - Cookie without SameSite Attribute)
- **Alert Reference:** 10054-1
- **Input Vector:**
- **Description:** A cookie has been set without the SameSite attribute, which means that the cookie can be sent as a result of a 'cross-site' request. The SameSite attribute is an effective counter measure to cross-site request forgery, cross-site script inclusion, and timing attacks.
- **Solution:** Ensure that the SameSite attribute is set to either 'lax' or ideally 'strict' for all cookies.
- **Reference:** SameSite Cookies

13. Cross-Domain JavaScript Source File Inclusion

- **URL:** https://www.google.com/map/reserve
- **Risk:** Low
- **Confidence:** Medium
- **Parameter:** https://www.google-analytics.com/analytics.js
- **Attack:**
- **Evidence:** `<script async src="https://www.google-analytics.com/analytics.js" nonce="pyM8mQJTXmQcTHBdj1onSg"></script>`
- **CWE ID:** 829
- **WASC ID:** 15
- **Source:** Passive (10017 - Cross-Domain JavaScript Source File Inclusion)
- **Input Vector:**
- **Description:** The page includes one or more script files from a third-party domain.
- **Solution:** Ensure JavaScript source files are loaded from only trusted sources, and the sources can't be controlled by end users of the application.

14. Server Leaks Version Information via “Server” HTTP Response Header Field

- **URL:** https://www.google.com/books/
- **Risk:** Low
- **Confidence:** High
- **Parameter:**
- **Attack:**
- **Evidence:** `OFE/0.1`
- **CWE ID:** 200
- **WASC ID:** 13
- **Source:** Passive (10036 - HTTP Server Response Header)
- **Input Vector:**
- **Description:** The web/application server is leaking version information via the "Server" HTTP response header. Access to such information may facilitate attackers identifying other vulnerabilities your web/application server is subject to.
- **Solution:** Ensure that your web server, application server, load balancer, etc. is configured to suppress the "Server" header or provide generic details.
- **Reference:**
  - Apache ServerTokens Directive
  - Microsoft Documentation
  - Troy Hunt's Blog

15. Strict-Transport-Security Header Not Set (58)

- **URL:** https://www.google.com/sitemap.xml

- **Risk:** Low
- **Confidence:** High
- **Parameter:**
- **Attack:**
- **Evidence:**
- **CWE ID:** 319
- **WASC ID:** 15
- **Source:** Passive (10035 - Strict-Transport-Security Header)
- **Alert Reference:** 10035-1
- **Input Vector:**
- **Description:** HTTP Strict Transport Security (HSTS) is a web security policy mechanism whereby a web server declares that complying user agents (such as a web browser) are to interact with it using only secure HTTPS connections (i.e. HTTP layered over TLS/SSL). HSTS is an IETF standards track protocol and is specified in RFC 6797.
- **Solution:** Ensure that your web server, application server, load balancer, etc. is configured to enforce Strict-Transport-Security.
- **Reference:**
    - OWASP HTTP Strict Transport Security Cheat Sheet
    - OWASP Security Headers
    - Wikipedia on HSTS
    - Can I Use - HSTS
    - RFC 6797
    - Troy Hunt's Blog

16. Timestamp Disclosure – Unix

- **URL:** https://www.google.com/maps/reserve
- **Risk:** Low
- **Confidence:** Low
- **Parameter:**
- **Attack:**
- **Evidence:** `1732584193`
- **CWE ID:** 200
- **WASC ID:** 13
- **Source:** Passive (10096 - Timestamp Disclosure)
- **Input Vector:**
- **Description:** A timestamp was disclosed by the application/web server - Unix
- **Solution:** Manually confirm that the timestamp data is not sensitive, and that the data cannot be aggregated to disclose exploitable patterns.
- **Reference:** CWE-200: Information Exposure

17. X-Content-Type-Options Header Missing (33)

- **URL:** https://www.google.com
- **Risk:** Low
- **Confidence:** Medium
- **Parameter:**
- **Attack:**
- **Evidence:**
- **CWE ID:** 693
- **WASC ID:** 15
- **Source:** Passive (10021 - X-Content-Type-Options Header Missing)
- **Input Vector:**
- **Description:** The Anti-MIME-Sniffing header X-Content-Type-Options was not set to 'nosniff'. This allows older versions of Internet Explorer and Chrome to perform MIME-sniffing on the response body, potentially causing the response body to be interpreted and displayed as a content type other than the declared content type. Current (early 2014) and legacy versions of Firefox will use the declared content type (if one is set), rather than performing MIME-sniffing.
- **Solution:** Ensure that the application/web server sets the Content-Type header appropriately, and that it sets the X-Content-Type-Options header to 'nosniff' for all web pages. If possible, ensure that the end user uses a standards-compliant and modern web browser that does not perform MIME-sniffing at all, or that can be directed by the web application/web server to not perform MIME-sniffing.
- **Reference:**
  - Microsoft Documentation
  - OWASP Security Headers

---

18. Content Security Policy (CSP) Report-Only Header Found

- **URL:** https://www.google.com
- **Risk:** Informational
- **Confidence:** High
- **Parameter:**
- **Attack:**
- **Evidence:**
- **CWE ID:** 693
- **WASC ID:** 15
- **Source:** Passive (10038 - Content Security Policy (CSP) Header Not Set)
- **Alert Reference:** 10038-3
- **Input Vector:**
- **Description:** The response contained a Content-Security-Policy-Report-Only header, this may indicate a work-in-progress implementation, or an oversight in promoting pre-Prod to Prod, etc.

Content Security Policy (CSP) is an added layer of security that helps to detect and mitigate certain types of attacks, including Cross Site Scripting (XSS) and data injection attacks. These attacks are used for everything from data theft to site defacement or distribution of malware. CSP provides a set of standard HTTP headers that allow website owners to declare approved sources of content that browsers should be allowed to load on.

Other Info:

Solution:

Ensure that your web server, application server, load balancer, etc. is configured to set the Content-Security-Policy header.

Reference: https://www.w3.org/TR/CSP2/

https://w3c.github.io/webappsec-csp/

https://caniuse.com/#feat=contentsecuritypolicy

https://content-security-policy.com/

https://cwe.mitre.org/data/definitions/200.html.

19. Information Disclosure - Sensitive Information in URL

- **URL:** https://www.google.com/citations?user
- **Risk:** Informational
- **Confidence:** Medium
- **Parameter:** user
- **Attack:**
- **Evidence:** user
- **CWE ID:** 200
- **WASC:** 13
- **Source:** Passive (10024 - Information Disclosure - Sensitive Information in URL)
- **Input Vector:**
- **Description:** The request appeared to contain sensitive information leaked in the URL. This can violate PCI and most organizational compliance policies. You can configure the list of strings for this check to add or remove values specific to your environment.
- **Other Info:** The URL contains potentially sensitive information. The following string was found via the pattern: user.
- **Solution:** Do not pass sensitive information in URIs.

20. Information Disclosure – Suspicious Comments

- **URL:** https://www.google.com
- **Risk:** Informational
- **Confidence:** Low
- **Parameter:** user

- **Attack:**
- **Evidence:** fn
- **CWE ID:** 200
- **WASC:** 13
- **Source:** Passive (10027 - Information Disclosure - Suspicious Comments)
- **Input Vector:**
- **Description:** The response appears to contain suspicious comments which may help an attacker. Note: Matches made within script blocks or files are against the entire content not only comments.
- **Other Info:** The following pattern was used: \bDB\b and was detected in the element starting with: `<script nonce="U_A_AJoukFUaONjkD92taQ">(function(){window.google.erd={jsr:1,bv:2040,sd:true,de:true};})();(function(){var sdo=fa` - see evidence field for the suspicious comment/snippet.
- **Solution:** Remove all comments that return information that may help an attacker and fix any underlying problems they refer to.

21. Loosely Scoped Cookie

- **URL:** https://www.google.com
- **Risk:** Informational
- **Confidence:** Low
- **Parameter:**
- **Attack:**
- **Evidence:**
- **CWE ID:** 565
- **WASC:** 15
- **Source:** Passive (90033 - Loosely Scoped Cookie)
- **Input Vector:**
- **Description:** Any subdomain of nottrusted.com can access the cookie. Loosely scoped cookies are common in mega-applications like google.com and live.com. Cookies set from a subdomain like app.foo.bar are transmitted only to that domain by the browser. However, cookies scoped to a parent-level domain may be transmitted to the parent, or any subdomain of the parent.
- **Other Info:** The origin domain used for comparison was: www.google.com
- **Cookies:**
  - AEC=AVYB7cpBHvHRuPnfoek9ikhJJGJaFdkf2Lrm_qU0Fg0pSgfx8S_N9_s2HA
  - NID=515=dpNCJBCNkbI7o7zHYFA75B1MofA0RxGH5tMkLOxdkKHfD82cDlyQPwV3yRqVMEogwJfZhBqrW543oJ96xNxmgkXaIr4pprsAdnQ2eRetJBHB42dCEhkVofAgH8fvUjGg8xJaz1BCn8-xlD4u6aoRyKKNLIIN1F4C-D14XEHPgRk.
- **Solution:** Always scope cookies to a FQDN (Fully Qualified Domain Name).

22. Modern Web Application

- **URL:** https://www.google.com/maps/dir/

- **Risk:** Informational
- **Confidence:** Medium
- **Parameter:**
- **Attack:**
- **Evidence:** <noscript> <div id="XvQR9b"> <div class="wSgKnf"> <div> When you have eliminated the <strong>JavaScript</strong>, whatever remains must be an empty page. </div> <a class="hl4GXb" href="https://support.google.com/maps/?hl=en&amp;authuser=0&amp;p=no_javascript" target="_blank"> Enable JavaScript to see Google Maps. </a> </div> </div> </noscript>
- **CWE ID:** 200
- **WASC:** 13
- **Source:** Passive (10109 – Modern Web Application)
- **Input Vector:**
- **Description:** The application appears to be a modern web application. If you need to explore it automatically then the Ajax Spider may well be more effective than the standard one.
- **Other Info:** A <noscript> tag has been found, which is an indication that the application works differently with JavaScript enabled compared to when it is not.
- **Solution:** This is an informational alert and so no changes are required.

23. Re-examine Cache-control Directives

- **URL:** https://www.google.com
- **Risk:** Informational
- **Confidence:** Low
- **Parameter:** cache-control
- **Attack:**
- **Evidence:** private, max-age = 0
- **CWE ID:** 525
- **WASC:** 13
- **Source:** Passive (10015 - Re-examine Cache-control Directives)
- **Input Vector:**
- **Description:** The cache-control header has not been set properly or is missing, allowing the browser and proxies to cache content. For static assets like css, js, or image files this might be intended, however, the resources should be reviewed to ensure that no sensitive content will be cached.
- **Solution:** For secure content, ensure the cache-control HTTP header is set with "no-cache, no-store, must-revalidate". If an asset should be cached consider setting the directives "public, max-age, immutable".

24. Retrieved from Cache

- **URL:** https://www.google.com/books/
- **Risk:** Informational
- **Confidence:** Medium
- **Parameter:**

- **Attack:**
- **Evidence:** Age: 50586
- **CWE ID:**
- **WASC:** 13
- **Source:** Passive (10050 - Retrieved from Cache)
- **Alert Reference:** 10050-2
- **Input Vector:**
- **Description:** The content was retrieved from a shared cache. If the response data is sensitive, personal or user-specific, this may result in sensitive information being leaked. In some cases, this may even result in a user gaining complete control of the session of another user, depending on the configuration of the caching components in use in their environment. This is primarily an issue where caching servers such as "proxy" caches are configured on the local network. This configuration is typically found in corporate or educational environments, for instance.
- **Other Info:** The presence of the 'Age' header indicates that a HTTP/1.1 compliant caching server is in use.
- **Solution:** Validate that the response does not contain sensitive, personal or user-specific information. If it does, consider the use of the following HTTP response headers, to limit, or prevent the content being stored and retrieved from the cache by another user: `Cache-Control: no-cache, no-store, must-revalidate, private`.

25. Session Management Response Identified

- **URL:** https://www.google.com
- **Risk:** Informational
- **Confidence:** Medium
- **Parameter:** NID
- **Attack:**
- **Evidence:** 515=dpNCJBCNkbI7o7zHYFA75B1MofA0RxGH5tMkLOxdkKHfD82cDlyQPwV3yRqVMEogwJfZhBqrW543oJ96xNxmgkXaIr4pprsAdnQ2eRetJBHB42dCEhkVofAgH8fvUjGg8xJaz1BCn8-xlD4u6aoRyKKNLIIN1F4C-D14XEHPgRk
- **CWE ID:** 200
- **WASC:** 13
- **Source:** Passive (10112 - Session Management Response Identified)
- **Input Vector:**
- **Description:** The given response has been identified as containing a session management token. The 'Other Info' field contains a set of header tokens that can be used in the Header Based Session Management Method. If the request is in a context which has a Session Management Method set to "Auto-Detect" then this rule will change the session management to use the tokens identified.
- **Other Info:** Cookies: `NID`, `AEC`.
- **Solution:** This is an informational alert rather than a vulnerability and so there is nothing to fix.

26. User Controllable HTML Element Attribute (Potential XSS)

- **URL:** https://www.google.com/imghp?hl=en&ogbl
- **Risk:** Informational
- **Confidence:** Low
- **Parameter:** hi
- **Attack:**
- **Evidence:**
- **CWE ID:** 20
- **WASC:** 20
- **Source:** Passive (10031 - User Controllable HTML Element Attribute (Potential XSS))
- **Input Vector:**
- **Description:** This check looks at user-supplied input in query string parameters and POST data to identify where certain HTML attribute values might be controlled. This provides hot-spot detection for XSS (cross-site scripting) that will require further review by a security analyst to determine exploitability.
- **Other Info:** User-controlled HTML attribute values were found. Try injecting special characters to see if XSS might be possible. The page at the following URL: `https://www.google.com/imghp?hl=en&ogbl` appears to include user input in: a(n) `[html]` tag `[lang]` attribute. The user input found was: `hl=en`.
- **Solution:** Validate all input and sanitize output it before writing to any HTML attributes.