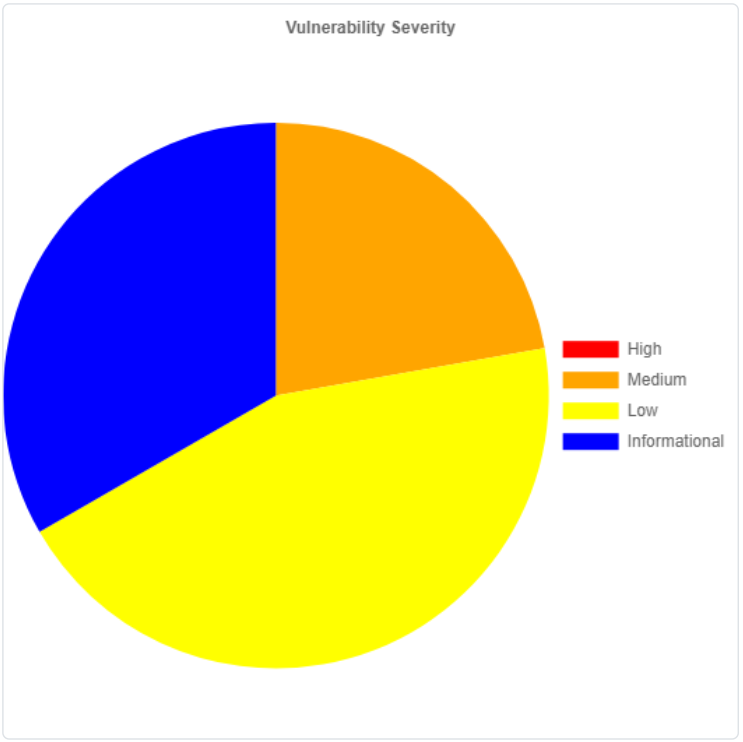
ZAP Scanning Report - polri.go.id

Generated on Thu, 9 Feb 2023 12:47:58

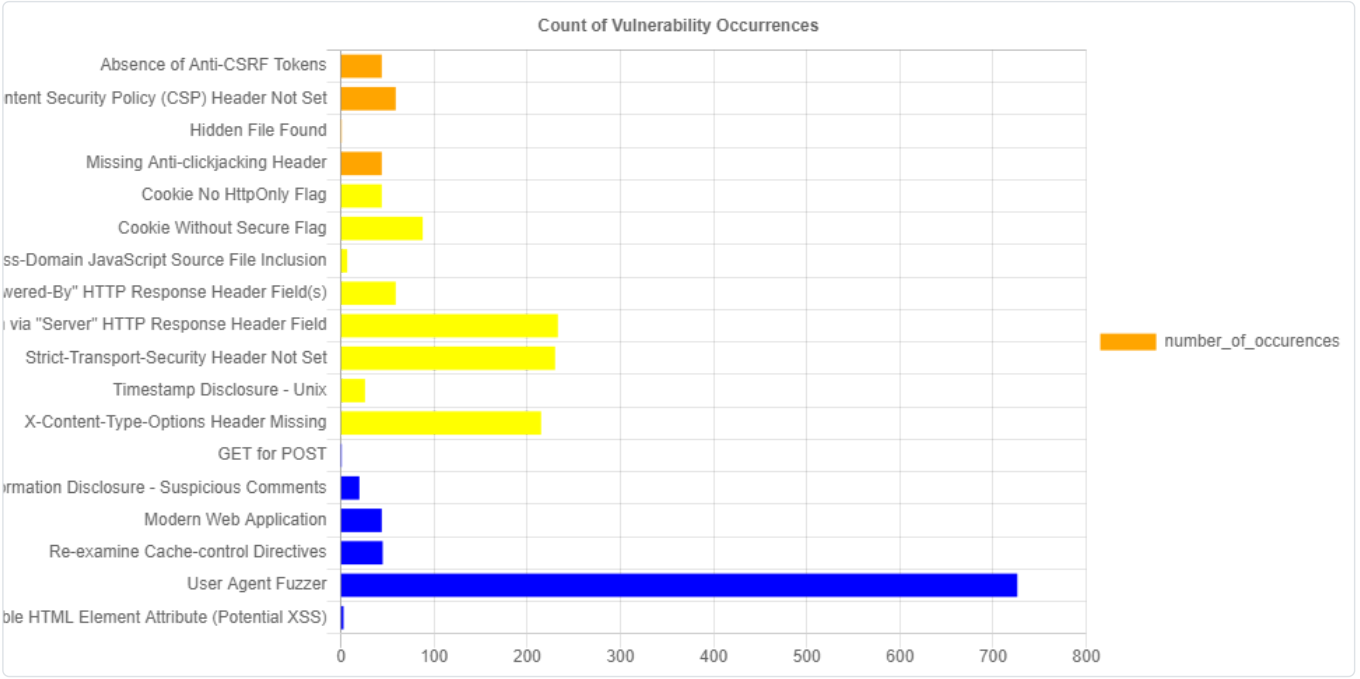
Most Severe Alert

Medium



Most Common Bug

User Agent Fuzzer (726)



Vulnerability Impact

#	Name	Impact

1	Absence of Anti-CSRF Tokens [1] [2]	No Anti-CSRF tokens were found in a HTML submission form.
2	Content Security Policy (CSP) Header Not Set [1] [2] [3] [4] [5] [6] [7]	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 that page — covered types are JavaScript, CSS, HTML frames, fonts, images and embeddable objects such as Java applets, ActiveX, audio and video files.
3	Hidden File Found [1]	A sensitive file was identified as accessible or available. This may leak administrative, configuration, or credential information which can be leveraged by a malicious individual to further attack the system or conduct social engineering efforts.
4	Missing Anti-clickjacking Header [1]	The response does not include either Content-Security-Policy with 'frame-ancestors' directive or X-Frame-Options to protect against 'ClickJacking' attacks.
5	Cookie No HttpOnly Flag [1]	A cookie has been set without the HttpOnly flag, which means that the cookie can be accessed by JavaScript. If a malicious script can be run on this page then the cookie will be accessible and can be transmitted to another site. If this is a session cookie then session hijacking may be possible.
6	Cookie Without Secure Flag [1]	A cookie has been set without the secure flag, which means that the cookie can be accessed via unencrypted connections.
7	Cross-Domain JavaScript Source File Inclusion	The page includes one or more script files from a third-party domain.
8	Server Leaks Information via "X-Powered-By" HTTP Response Header Field(s) [1] [2]	The web/application server is leaking information via one or more "X-Powered-By" HTTP response headers. Access to such information may facilitate attackers identifying other frameworks/components your web application is reliant upon and the vulnerabilities such components may be subject to.
9	Server Leaks Version Information via "Server" HTTP Response Header Field [1] [2] [3] [4]	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.
10	Strict-Transport-Security Header Not Set [1] [2] [3] [4] [5]	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.
11	Timestamp Disclosure - Unix [1]	A timestamp was disclosed by the application/web server - Unix
12	X-Content-Type-Options Header Missing [1] [2]	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.
13	GET for POST	A request that was originally observed as a POST was also accepted as a GET. This issue does not represent a security weakness unto itself, however, it may facilitate simplification of other attacks. For example if the original POST is subject to Cross-Site Scripting (XSS), then this finding may indicate that a simplified (GET based) XSS may also be possible.
14	Information Disclosure - Suspicious Comments	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.
15	Modern Web Application	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.
16	Re-examine Cache-control Directives [1] [2] [3]	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.

17	User Agent Fuzzer [1]	Check for differences in response based on fuzzed User Agent (eg. mobile sites, access as a Search Engine Crawler). Compares the response statuscode and the hashcode of the response body with the original response.
18	User Controllable HTML Element Attribute (Potential XSS) [1]	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.

Vulnerability Descriptions