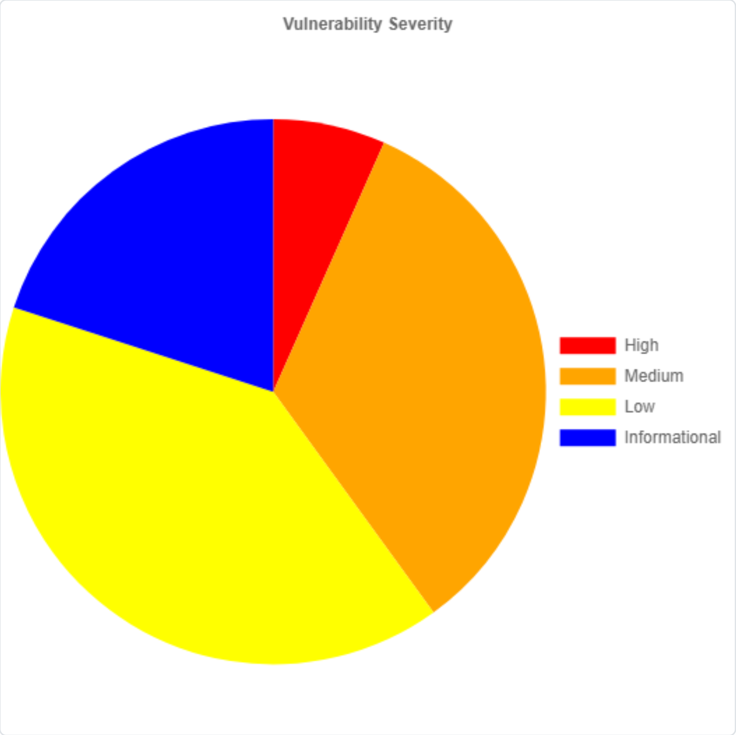
ZAP Scanning Report - ojk.go.id

Generated on Thu, 9 Feb 2023 11:47:42

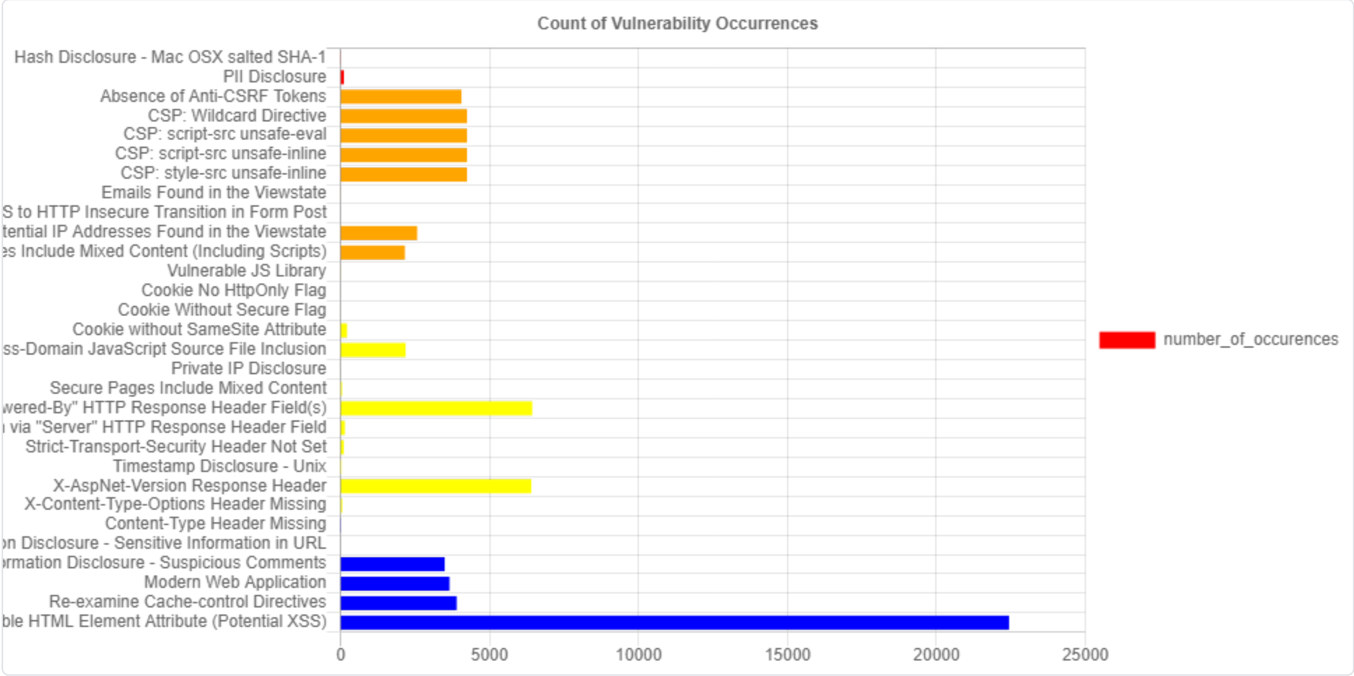
Most Severe Alert

High



Most Common Bug

User Controllable HTML Element Attribute (Potential XSS) (22435)



Vulnerability Impact

#	Name	Impact

1	Hash Disclosure - Mac OSX salted SHA-1 [1] [2]	A hash was disclosed by the web server. - Mac OSX salted SHA-1
2	PII Disclosure	The response contains Personally Identifiable Information, such as CC number, SSN and similar sensitive data.
3	Absence of Anti-CSRF Tokens [1] [2]	No Anti-CSRF tokens were found in a HTML submission form.
4	CSP: Wildcard Directive [1] [2] [3] [4] [5] [6]	Content Security Policy (CSP) is an added layer of security that helps to detect and mitigate certain types of attacks. Including (but not limited to) 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.
5	CSP: script-src unsafe-eval [1] [2] [3] [4] [5] [6]	Content Security Policy (CSP) is an added layer of security that helps to detect and mitigate certain types of attacks. Including (but not limited to) 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.
6	CSP: script-src unsafe-inline [1] [2] [3] [4] [5] [6]	Content Security Policy (CSP) is an added layer of security that helps to detect and mitigate certain types of attacks. Including (but not limited to) 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.
7	CSP: style-src unsafe-inline [1] [2] [3] [4] [5] [6]	Content Security Policy (CSP) is an added layer of security that helps to detect and mitigate certain types of attacks. Including (but not limited to) 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.
8	Emails Found in the Viewstate	The following emails were found being serialized in the viewstate field:
9	HTTPS to HTTP Insecure Transition in Form Post	This check identifies secure HTTPS pages that host insecure HTTP forms. The issue is that a secure page is transitioning to an insecure page when data is uploaded through a form. The user may think they're submitting data to a secure page when in fact they are not.
10	Potential IP Addresses Found in the Viewstate	The following potential IP addresses were found being serialized in the viewstate field:
11	Secure Pages Include Mixed Content (Including Scripts) [1]	The page includes mixed content, that is content accessed via HTTP instead of HTTPS.
12	Vulnerable JS Library [1] [2] [3] [4] [5] [6] [7] [8] [9] [10] [11]	The identified library jquery, version 1.8.3 is vulnerable.
13	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.
14	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.
15	Cookie without SameSite Attribute [1]	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.

16	Cross-Domain JavaScript Source File Inclusion	The page includes one or more script files from a third-party domain.
17	Private IP Disclosure [1]	A private IP (such as 10.x.x.x, 172.x.x.x, 192.168.x.x) or an Amazon EC2 private hostname (for example, ip-10-0-56-78) has been found in the HTTP response body. This information might be helpful for further attacks targeting internal systems.
18	Secure Pages Include Mixed Content [1]	The page includes mixed content, that is content accessed via HTTP instead of HTTPS.
19	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.
20	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.
21	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.
22	Timestamp Disclosure - Unix [1]	A timestamp was disclosed by the application/web server - Unix
23	X-AspNet-Version Response Header [1] [2]	Server leaks information via "X-AspNet-Version"/"X-AspNetMvc-Version" HTTP response header field(s).
24	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.
25	Content-Type Header Missing [1]	The Content-Type header was either missing or empty.
26	Information Disclosure - Sensitive Information in URL	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.
27	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.
28	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.
29	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.
30	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