Static and Dynamic Analysis

ISTEC-Cyber Security

November 6, 2024

Description

This report contains static and dynamic analysis of the target. It uses Semgrep for static analysis and OWASP ZAP, Nmap, and SQLMap for dynamic analysis.

Provided by

1 Static Analysis

Details about static analysis...

2 Analysis Report

2.1 Risk Summary

Risk Level	Number of Findings
Low Risk	42
Medium Risk	28
High Risk	0
Critical Risk	0

Table 1: Summary of Risk Findings

2.2 Vulnerability Categories

- Category 1: Denial-of-Service (DoS) — 28
- Category 2: Improper Encoding — 4
- Category 3: Mass Assignment — 4
- Category 4: Cross-Site-Scripting (XSS) — 4
- Category 5: Cryptographic Issues — 29
- Category 6: Mishandled Sensitive Information — 1

2.3 Vulnerabilities by Page

Vulnerability 1	
Path: ScrapedFiles/index/main/analytics.js	
Vulnerability Class	['Denial-of-Service (DoS)']
Start	line: 10 col: 486
End	line: 10 col: 529
Message	RegExp() called with a 'a' function argument, this might allow an attacker to cause a Regular Expression Denial-of-Service (ReDoS) within your application as RegExP blocks the main thread. For this reason, it is recommended to use hardcoded regexes instead. If your regex is run on user-controlled input, consider performing input validation or use a regex checking/sanitization library such as https://www.npmjs.com/package/recheck to verify that the regex does not appear vulnerable to ReDoS.

Vulnerability 2	
Path: ScrapedFiles/index/main/analytics.js	
Vulnerability Class	['Denial-of-Service (DoS)']
Start	line: 27 col: 150
End	line: 27 col: 189
Message	RegExp() called with a 'a' function argument, this might allow an attacker to cause a Regular Expression Denial-of-Service (ReDoS) within your application as RegExP blocks the main thread. For this reason, it is recommended to use hardcoded regexes instead. If your regex is run on user-controlled input, consider performing input validation or use a regex checking/sanitization library such as https://www.npmjs.com/package/recheck to verify that the regex does not appear vulnerable to ReDoS.

Vulnerability 3	
Path: ScrapedFiles/index/main/analytics.js	
Vulnerability Class	['Denial-of-Service (DoS)']
Start	line: 28 col: 304
End	line: 28 col: 368
Message	RegExp() called with a 'a' function argument, this might allow an attacker to cause a Regular Expression Denial-of-Service (ReDoS) within your application as RegExP blocks the main thread. For this reason, it is recommended to use hardcoded regexes instead. If your regex is run on user-controlled input, consider performing input validation or use a regex checking/sanitization library such as https://www.npmjs.com/package/recheck to verify that the regex does not appear vulnerable to ReDoS.

Vulnerability 4	
Path: ScrapedFiles/index/main/analytics.js	
Vulnerability Class	['Improper Encoding']
Start	line: 37 col: 130
End	line: 37 col: 184
Message	'"https://www.google.%/ads/ga-audiences".replace' method will only replace the first occurrence when used with a string argument ("%"). If this method is used for escaping of dangerous data then there is a possibility for a bypass. Try to use sanitization library instead or use a Regex with a global flag.

3 Dynamic Analysis

Details about dynamic analysis...

4 Analysis Report

4.1 Risk Summary

Risk Level	Number of Findings
Low Risk	204
Medium Risk	130
High Risk	1
Critical Risk	0

Table 2: Summary of Risk Findings

4.2 Vulnerability Categories

- Category 1: Cross Site Scripting (DOM Based) — 1
- Category 2: Absence of Anti-CSRF Tokens — 40
- Category 3: Content Security Policy (CSP) Header Not Set — 47
- Category 4: Missing Anti-clickjacking Header — 43
- Category 5: Server Leaks Information via "X-Powered-By" HTTP Response Header Field(s) — 62
- Category 6: Server Leaks Version Information via "Server" HTTP Response Header Field — 74
- Category 7: X-Content-Type-Options Header Missing — 68
- Category 8: Authentication Request Identified — 1
- Category 9: Charset Mismatch (Header Versus Meta Content-Type Charset) — 31
- Category 10: Information Disclosure - Suspicious Comments — 1
- Category 11: Modern Web Application — 9
- Category 12: User Controllable HTML Element Attribute (Potential XSS) — 3

4.3 Vulnerabilities by Page

Site 1: http://testphp.vulnweb.com

Host: testphp.vulnweb.com, Port: 80, SSL: false

Vulnerability 1	
Risk Level	High (High)
Vulnerability Name	Cross Site Scripting (DOM Based)

Description	Cross-site Scripting (XSS) is an attack technique that involves echoing attacker-supplied code into a user's browser instance. A browser instance can be a standard web browser client, or a browser object embedded in a software product such as the browser within WinAmp, an RSS reader, or an email client. The code itself is usually written in HTML/JavaScript, but may also extend to VBScript, ActiveX, Java, Flash, or any other browser-supported technology.When an attacker gets a user's browser to execute his/her code, the code will run within the security context (or zone) of the hosting web site. With this level of privilege, the code has the ability to read, modify and transmit any sensitive data accessible by the browser. A Cross-site Scripted user could have his/her account hijacked (cookie theft), their browser redirected to another location, or possibly shown fraudulent content delivered by the web site they are visiting. Cross-site Scripting attacks essentially compromise the trust relationship between a user and the web site. Applications utilizing browser object instances which load content from the file system may execute code under the local machine zone allowing for system compromise.There are three types of Cross-site Scripting attacks: non-persistent, persistent and DOM-based.Non-persistent attacks and DOM-based attacks require a user to either visit a specially crafted link laced with malicious code, or visit a malicious web page containing a web form, which when posted to the vulnerable site, will mount the attack. Using a malicious form will oftentimes take place when the vulnerable resource only accepts HTTP POST requests. In such a case, the form can be submitted automatically, without the victim's knowledge (e.g. by using JavaScript). Upon clicking on the malicious link or submitting the malicious form, the XSS payload will get echoed back and will get interpreted by the user's browser and execute. Another technique to send almost arbitrary requests (GET and POST) is by using an embedded client, such as Adobe Flash.Persistent attacks occur when the malicious code is submitted to a web site where it's stored for a period of time. Examples of an attacker's favorite targets often include message board posts, web mail messages, and web chat software. The unsuspecting user is not required to interact with any additional site/link (e.g. an attacker site or a malicious link sent via email), just simply view the web page containing the code.

Instances	**URI**
Instance 1	http://testphp.vulnweb.com/#jaVasCript: /*-/*\unhbox\voidb@x\bgroup\let\unhbox\ voidb@x\setbox\@tempboxa\hbox{/\global\ mathchardef\accent@spacefactor\spacefactor}\ let\begingroup\let\typeout\protect\ begingroup\def\MessageBreak{Ω(Font)}\let\ protect\immediate\write\m@ne{LaTeXFontInfo: oninputline222.}\endgroup\endgroup\relax\ let\ignorespaces\relax\accent18/\egroup\ spacefactor\accent@spacefactor*\‘/*’/*"/ **/(/**/oNcliCk=alert(5397))//%0D%0A%0d%0a/ /</stYle/</titLe/</teXtarEa/</scRipt/--!>sVg/ <sVg/oNloAd=alert(5397)//>

Vulnerability 2	
Risk Level	Medium (Low)
Vulnerability Name	Absence of Anti-CSRF Tokens

Description	No Anti-CSRF tokens were found in a HTML submission form.A cross-site request forgery is an attack that involves forcing a victim to send an HTTP request to a target destination without their knowledge or intent in order to perform an action as the victim. The underlying cause is application functionality using predictable URL/form actions in a repeatable way. The nature of the attack is that CSRF exploits the trust that a web site has for a user. By contrast, cross-site scripting (XSS) exploits the trust that a user has for a web site. Like XSS, CSRF attacks are not necessarily cross-site, but they can be. Cross-site request forgery is also known as CSRF, XSRF, one-click attack, session riding, confused deputy, and sea surf.CSRF attacks are effective in a number of situations, including: * The victim has an active session on the target site. * The victim is authenticated via HTTP auth on the target site. * The victim is on the same local network as the target site.CSRF has primarily been used to perform an action against a target site using the victim's privileges, but recent techniques have been discovered to disclose information by gaining access to the response. The risk of information disclosure is dramatically increased when the target site is vulnerable to XSS, because XSS can be used as a platform for CSRF, allowing the attack to operate within the bounds of the same-origin policy.
Instances	**URI**
Instance 1	`http://testphp.vulnweb.com/`
Instance 2	`http://testphp.vulnweb.com/artists.php`
Instance 3	`http://testphp.vulnweb.com/artists.php?artist=1`
Instance 4	`http://testphp.vulnweb.com/artists.php?artist=2`
Instance 5	`http://testphp.vulnweb.com/artists.php?artist=3`
Instance 6	`http://testphp.vulnweb.com/cart.php`
Instance 7	`http://testphp.vulnweb.com/categories.php`
Instance 8	`http://testphp.vulnweb.com/disclaimer.php`
Instance 9	`http://testphp.vulnweb.com/guestbook.php`
Instance 10	`http://testphp.vulnweb.com/guestbook.php`

Instance 11	http://testphp.vulnweb.com/index.php
Instance 12	http://testphp.vulnweb.com/listproducts.php?artist=1
Instance 13	http://testphp.vulnweb.com/listproducts.php?artist=2
Instance 14	http://testphp.vulnweb.com/listproducts.php?artist=3
Instance 15	http://testphp.vulnweb.com/listproducts.php?cat=1
Instance 16	http://testphp.vulnweb.com/listproducts.php?cat=2
Instance 17	http://testphp.vulnweb.com/listproducts.php?cat=3
Instance 18	http://testphp.vulnweb.com/listproducts.php?cat=4
Instance 19	http://testphp.vulnweb.com/login.php
Instance 20	http://testphp.vulnweb.com/login.php
Instance 21	http://testphp.vulnweb.com/product.php?pic=1
Instance 22	http://testphp.vulnweb.com/product.php?pic=1
Instance 23	http://testphp.vulnweb.com/product.php?pic=2
Instance 24	http://testphp.vulnweb.com/product.php?pic=2
Instance 25	http://testphp.vulnweb.com/product.php?pic=3
Instance 26	http://testphp.vulnweb.com/product.php?pic=3
Instance 27	http://testphp.vulnweb.com/product.php?pic=4
Instance 28	http://testphp.vulnweb.com/product.php?pic=4
Instance 29	http://testphp.vulnweb.com/product.php?pic=5
Instance 30	http://testphp.vulnweb.com/product.php?pic=5
Instance 31	http://testphp.vulnweb.com/product.php?pic=6
Instance 32	http://testphp.vulnweb.com/product.php?pic=6
Instance 33	http://testphp.vulnweb.com/product.php?pic=7
Instance 34	http://testphp.vulnweb.com/product.php?pic=7
Instance 35	http://testphp.vulnweb.com/signup.php
Instance 36	http://testphp.vulnweb.com/signup.php
Instance 37	http://testphp.vulnweb.com/cart.php

Instance 38	http://testphp.vulnweb.com/guestbook.php
Instance 39	http://testphp.vulnweb.com/guestbook.php
Instance 40	http://testphp.vulnweb.com/search.php?test=query

Vulnerability 3	
Risk Level	Medium (High)
Vulnerability Name	Content Security Policy (CSP) Header Not Set
Description	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.
Instances	**URI**
Instance 1	http://testphp.vulnweb.com/
Instance 2	http://testphp.vulnweb.com/AJAX/index.php
Instance 3	http://testphp.vulnweb.com/artists.php
Instance 4	http://testphp.vulnweb.com/artists.php?artist=1
Instance 5	http://testphp.vulnweb.com/artists.php?artist=2
Instance 6	http://testphp.vulnweb.com/artists.php?artist=3
Instance 7	http://testphp.vulnweb.com/cart.php
Instance 8	http://testphp.vulnweb.com/categories.php
Instance 9	http://testphp.vulnweb.com/disclaimer.php
Instance 10	http://testphp.vulnweb.com/guestbook.php

Instance 11	http://testphp.vulnweb.com/high
Instance 12	http://testphp.vulnweb.com/hpp/
Instance 13	http://testphp.vulnweb.com/hpp/?pp=12
Instance 14	http://testphp.vulnweb.com/hpp/params.php?p=valid&pp=12
Instance 15	http://testphp.vulnweb.com/index.php
Instance 16	http://testphp.vulnweb.com/listproducts.php?artist=1
Instance 17	http://testphp.vulnweb.com/listproducts.php?artist=2
Instance 18	http://testphp.vulnweb.com/listproducts.php?artist=3
Instance 19	http://testphp.vulnweb.com/listproducts.php?cat=1
Instance 20	http://testphp.vulnweb.com/listproducts.php?cat=2
Instance 21	http://testphp.vulnweb.com/listproducts.php?cat=3
Instance 22	http://testphp.vulnweb.com/listproducts.php?cat=4
Instance 23	http://testphp.vulnweb.com/login.php
Instance 24	http://testphp.vulnweb.com/Mod_Rewrite_Shop/
Instance 25	http://testphp.vulnweb.com/Mod_Rewrite_Shop/BuyProduct-1/
Instance 26	http://testphp.vulnweb.com/Mod_Rewrite_Shop/BuyProduct-2/
Instance 27	http://testphp.vulnweb.com/Mod_Rewrite_Shop/BuyProduct-3/
Instance 28	http://testphp.vulnweb.com/Mod_Rewrite_Shop/Details/color-printer/3/
Instance 29	http://testphp.vulnweb.com/Mod_Rewrite_Shop/Details/network-attached-storage-dlink/1/
Instance 30	http://testphp.vulnweb.com/Mod_Rewrite_Shop/Details/web-camera-a4tech/2/
Instance 31	http://testphp.vulnweb.com/Mod_Rewrite_Shop/RateProduct-1.html

Instance 32	http://testphp.vulnweb.com/Mod_Rewrite_Shop/RateProduct-2.html
Instance 33	http://testphp.vulnweb.com/Mod_Rewrite_Shop/RateProduct-3.html
Instance 34	http://testphp.vulnweb.com/privacy.php
Instance 35	http://testphp.vulnweb.com/product.php?pic=1
Instance 36	http://testphp.vulnweb.com/product.php?pic=2
Instance 37	http://testphp.vulnweb.com/product.php?pic=3
Instance 38	http://testphp.vulnweb.com/product.php?pic=4
Instance 39	http://testphp.vulnweb.com/product.php?pic=5
Instance 40	http://testphp.vulnweb.com/product.php?pic=6
Instance 41	http://testphp.vulnweb.com/product.php?pic=7
Instance 42	http://testphp.vulnweb.com/robots.txt
Instance 43	http://testphp.vulnweb.com/signup.php
Instance 44	http://testphp.vulnweb.com/cart.php
Instance 45	http://testphp.vulnweb.com/guestbook.php
Instance 46	http://testphp.vulnweb.com/search.php?test=query
Instance 47	http://testphp.vulnweb.com/secured/newuser.php