Vulnerability Assessment And Penetration Testing

PREPARED FOR:

Career path orientation

Confidential

DOCUMENT VERSION CONTROL

DOCUMENT VERSION CONTROL

Data Classification *– **Client Confidential***

Client Name	
Project name	VAPT
Authors	
Approved by	
Version	
Submission Date	

Section 1: Executive Summary

This report: documents the findings after testing (irsheidat.softether.net). A series of tests were conducted against the targeted scope, using testing tools and where appropriate, manual testing techniques, to establish the presence of actual or potentially exploitable security vulnerabilities, which if exploited could result in direct or indirect damage to Customer Name. These key findings (classified as "High", "Medium" or "Low") have been highlighted in this report. Please refer to the Detailed Penetration Testing Results of the report for in depth details the key findings, their associated risks, and recommended actions.

The following represents the definition and the description of each severity rate.

Impact	Description
High	A vulnerability that can be exploited by the attacker and cause huge damage to application components and data.
Medium	A vulnerability that can be exploited by the attacker and cause moderate damage to application components and data.
Low	A vulnerability that can be exploited by the attacker to understand application underlying technologies and versions which can be utilized in further attacks.

1.1 Scope Details

The scope of evaluation and testing covered the following assets:

#	Host	Platform
1	Irsheidat.softether.net	

Section 2: Detailed VAPT Results

2.1 Introduction

Target domain (irseidat. Softether.net)

The following is the detailed report for our findings during the Penetration testing process.

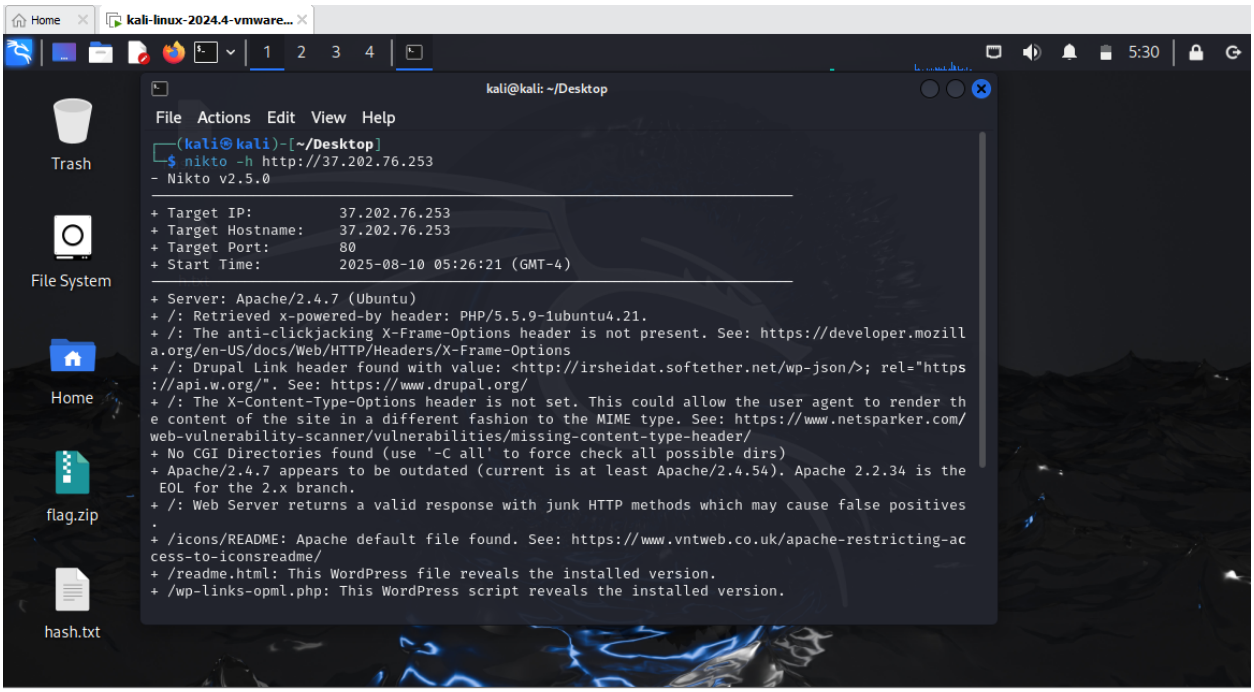
2.2 Restrictions

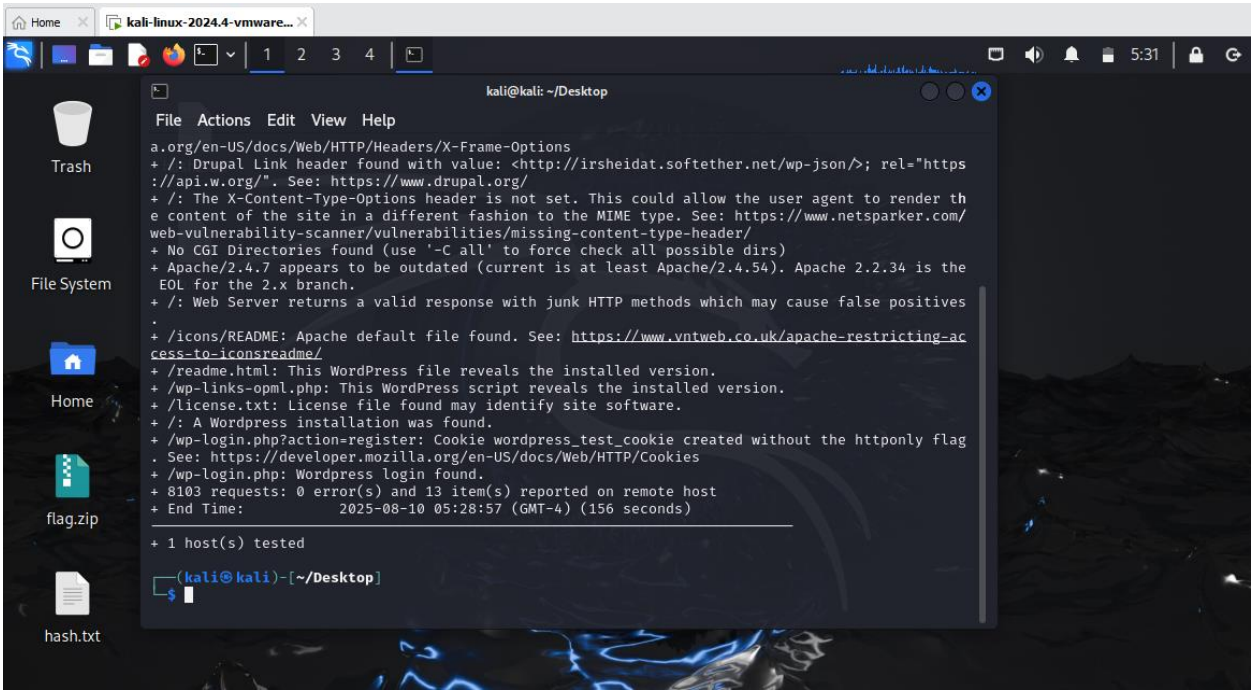
1. Cookie without HttpOnly flag on the registration page:

It was found that the wordpress_test_cookie is created without the HttpOnly attribute on the registration page (/wp-login.php?action=register). Lack of the HttpOnly flag makes the cookie accessible to client-side scripts, increasing the risk of theft via Cross-Site Scripting (XSS) attacks. It is recommended to set the HttpOnly attribute on all sensitive cookies to prevent access by JavaScript

2. Exposed WordPress login page:

The WordPress login page is accessible at /wp-login.php without additional protections like rate limiting or two-factor authentication (2FA). This exposure allows attackers to perform brute force attacks on user accounts. It is recommended to implement protections such as 2FA and rate limiting to mitigate this risk.





2.3 Recommendations

1. Set HttpOnly flag for all sensitive cookies: Ensure that cookies like wordpress_test_cookie and any session cookies have the HttpOnly attribute set. This will prevent client-side scripts from accessing these cookies, reducing the risk of theft through XSS attacks.
2. Implement strong login protections: Protect the WordPress login page by enabling mechanisms such as: • Rate limiting to block or slow down repeated login attempts. • Two-factor authentication (2FA) to add an extra layer of security beyond passwords. • CAPTCHA or similar anti-bot measures to prevent automated brute force attacks. • Monitoring and alerting on suspicious login activities.
3. Regularly test for XSS vulnerabilities: Since cookies without HttpOnly can be stolen if XSS exists, conduct frequent security testing and code reviews to identify and fix XSS issues.
4. Consider hiding or restricting access to /wp-login.php: If possible, restrict access to the login page by IP whitelisting, or move it behind additional authentication layers or custom URLs to reduce exposure.

CONCLUSION

The assessment identified medium-risk vulnerabilities related to cookie security and login page exposure on the WordPress site. The absence of the HttpOnly flag on cookies increases the risk of session theft through potential XSS attacks, while the publicly accessible login page without adequate protections leaves the site vulnerable to brute force attempts. By implementing the recommended security controls—such as setting proper cookie flags, enabling multi-factor authentication, rate limiting, and regular vulnerability testing—the overall security posture of the WordPress installation can be significantly improved, reducing the likelihood of unauthorized access and potential compromise.

END OF THE REPORT