VULNERABILITY ASSESSMENT AND PENETRATION TESTING (VAPT) REPORT

SUBMITTED TO

AGRICULTURAL DEVELOPMENT BANK LIMITED
RAMSHAHAPATH, KATHMANDU

SUBMITTED BY

PEACE NEPAL DOT COM
KUPONDOLE, LALITPUR

Date: October, 2024

CONTENTS

INTRODUCTION

INTRODUCTION TO VAPT

Vulnerability Assessment and Penetration Testing (VAPT) is a comprehensive security testing process used to assess the security posture of a system or application. It combines vulnerability assessment, which identifies weaknesses and vulnerabilities that could be exploited by attackers, with penetration testing, which simulates real-world attacks to evaluate the effectiveness of security controls.

The primary goal of VAPT is to proactively identify and mitigate potential security risks, ensuring the confidentiality, integrity, and availability of critical assets and data. By conducting a thorough assessment of vulnerabilities and performing controlled penetration tests, organizations can gain valuable insights into their security vulnerabilities, make informed decisions for risk mitigation, and enhance their overall security defenses.

OBJECTIVE

The primary objective of Vulnerability Assessment and Penetration Testing (VAPT) is to conduct a thorough security audit to identify and address potential vulnerabilities and weaknesses within a system or application. By performing VAPT, organizations aim to proactively assess their security measures, detect possible loopholes, and prioritize the implementation of appropriate measures to mitigate risks. The objective is to ensure that the systems and applications are adequately protected against cyber-attacks and data breaches. VAPT helps in evaluating and improving security controls, applying necessary security patches and updates, reviewing security policies and procedures, and continuously enhancing the overall security posture of the organization. Ultimately, the goal is to establish a robust and resilient security framework that safeguards critical assets and data, fostering a secure environment for the organization and its stakeholders.

SCOPE

Scope Details	**Description**
Tested Operating System	Windows 11
Tested Product Version	1
Tested Environment	UAT Server 1. Public Facing Website: https://adblbank.peacenepal.com/

	2. Content Management System (CMS): https://adblbackend.peacenepal.com/
Tool Used	Burp Suite
Report Prepared By	Quality Assurance Team

The scope of the technical assessment included the following work streams:

1. Web Vulnerability Assessment
2. Web Application Penetration Testing
3. API Security Testing
4. Input Validation Testing
5. Authentication and Session Management Testing
6. Cross-Site Scripting (XSS) Testing
7. SQL Injection Testing
8. Cross-Site Request Forgery (CSRF) Testing
9. Configuration Review for Laravel and Node.js
10. Dependency and Library Vulnerability Analysis

The results summarized in this document are based upon a collection of methodologies and tests interacting at a single point in time, utilizing tools and frameworks recognized in the industry. Given the nature of technology, which is continually changing and becoming ever more complex, any projection of the future findings contained in this document is subject to risks associated with change. As such, they may no longer accurately portray the system or environment in existence at that time.

Additionally, the information gathered is subject to inherent limitations. There may be weaknesses, errors, or irregularities that occur and go undetected during the assessment. It is important to consider this when interpreting the results, and stakeholders are encouraged to conduct regular security assessments and maintain a proactive security posture to adapt to evolving threats.

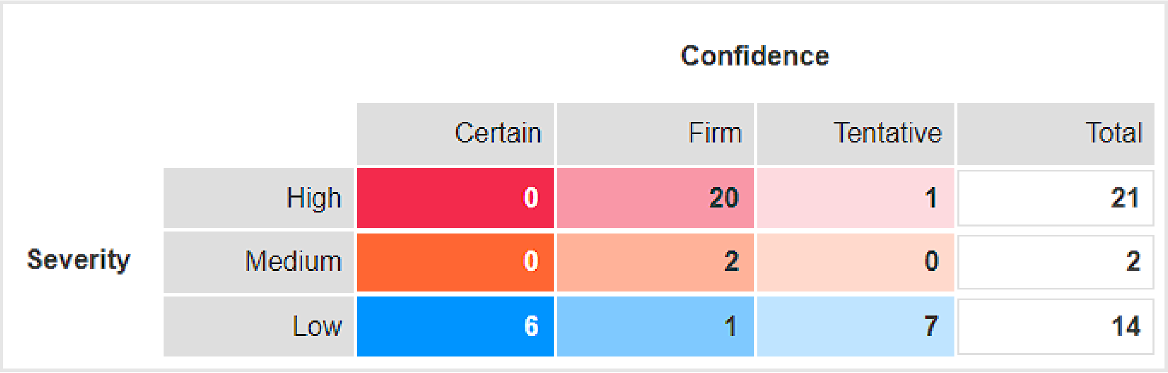
SUMMARY OF FINDING

We scan the system using **Burp suite Pro,** and during the engagement, findings were identified where the design or operating effectiveness of a security control did not meet industry leading practice. Each finding was qualitatively assigned a risk rating based on one of four categories: High, Medium, Low, and Information.

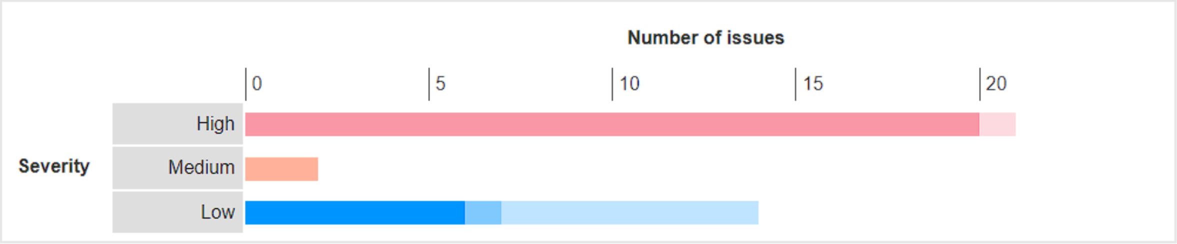
The categories are defined in the table below:

RISK LEVEL	**VULNERABILITY DEFINITION**
HIGH	High-risk vulnerabilities are those that can be easily exploited by attackers, potentially leading to severe consequences such as unauthorized data access or complete system compromise. These issues demand immediate attention and remediation efforts, as they pose a significant threat to the organization's security and data integrity.
MEDIUM	Medium-risk vulnerabilities require specific conditions or a certain level of attacker skill to exploit but still present a considerable threat. While they may not be as urgent as high-risk vulnerabilities, they can lead to unauthorized access or data compromise if left unaddressed. Remediation should be prioritized after high-risk issues are resolved.
LOW	Low-risk vulnerabilities are typically harder to exploit and have minimal impact on the system or data. Although they may not pose an immediate threat, it's advisable to address them during routine maintenance or development cycles to strengthen overall security. Ignoring these vulnerabilities can lead to accumulated risk over time.
INFORMATION	Informational issues provide insights or recommendations that enhance security practices without being direct vulnerabilities themselves. They often highlight best practices, compliance gaps, or areas for improvement. While they do not require immediate action, addressing these can contribute to a more robust security posture and proactive risk management.

The table below shows the numbers of issues identified in different categories. Issues are classified according to severity as High, Medium, and Low. This reflects the likely impact of each issue for a typical organization. Issues are also classified according to confidence as Certain, Firm or Tentative. This reflects the inherent reliability of the technique that was used to identify the issue.



		Confidence			
		Certain	Firm	Tentative	Total
Severity	High	0	20	1	21
	Medium	0	2	0	2
	Low	6	1	7	14



FINDING OVERVIEW

Only some informational issues were discovered during the vulnerability assessment of Website which cause no security threat to the system. No any high-risk vulnerabilities were identified. These findings underscore the need for prompt remediation of critical security risks, as well as addressing the identified vulnerabilities to strengthen the overall security resilience of the system/application.

Public Facing Website and Content Management System (CMS)				
S. N	**Vulnerability**	**Severity**	**Alert Count**	**Status**
1	SQL injection	High	2	Resolved
2	Cross-site scripting (DOM-based)	High	18	Resolved
3	Client-side desync	High	1	Resolved
4	TLS cookie without secure flag set	Medium	1	Resolved
5	Web cache poisoning	Medium	1	Resolved
6	Vulnerable JavaScript dependency	Low	4	Resolved
7	Open redirection (reflected)	Low	1	Resolved
8	Open redirection (DOM-based)	Low	3	Resolved
9	Password field with autocomplete enabled	Low	2	Resolved
10	Client-side HTTP parameter pollution (reflected)	Low	1	Resolved
11	Strict transport security not enforced	Low	3	Resolved

SUMMARY OF RECOMMENDATIONS

The following recommendations are provided based on the findings of the VAPT (Vulnerability Assessment and Penetration Testing) report. These recommendations aim to address the identified vulnerabilities and strengthen the overall security posture of the system/application. They are grounded in common best practices and industry standards for web application security, specifically aligned with the vulnerabilities typically identified in assessments.

1. **High-Risk Vulnerability Remediation**
   Immediate attention should be given to addressing any high-risk vulnerabilities identified in the report. This may involve patching the system, updating software components, or implementing additional security controls to effectively mitigate the risk.

2. **Medium and Low Vulnerability Mitigation**
   Medium and low vulnerabilities should be remediated promptly to minimize potential security risks. Actions may include applying security patches, configuring access controls, and enhancing authentication mechanisms.

3. **Informational Vulnerability Awareness**
   While informational vulnerabilities may not pose immediate risks, they should not be overlooked. Reviewing and addressing these vulnerabilities—such as improving documentation and implementing best practices—will enhance overall security.

4. **Regular Patching and Updates**
   Establish a systematic process for applying security patches and updates to ensure that software components remain current and protected against the latest threats.

5. **Strengthen Authentication Mechanisms**
   Enhance authentication practices by implementing strong password policies, multi-factor authentication (MFA), and account lockout mechanisms to prevent unauthorized access.

6. **Secure Input Validation**
   Implement robust input validation techniques to prevent injection attacks, such as SQL injection and XSS, by ensuring that only properly formatted data is processed.

7. **Improve Session Management**
   Adopt secure session handling practices, including the use of secure cookies and appropriate session timeouts, to prevent session hijacking and fixation.

8. **Implement CSRF Protections**

Use anti-CSRF tokens to ensure that state-changing requests are intentional and authorized, effectively protecting against CSRF attacks.

9. **Continuous Monitoring and Security Testing**
   Establish a regular security monitoring and testing program to proactively identify and address emerging vulnerabilities. Periodic VAPT assessments and security audits are essential for ongoing protection.

10. **Review and Update Configurations**
    Conduct regular configuration reviews to identify and rectify misconfigurations that could expose the application to risks, particularly those related to the Security Misconfiguration category in the OWASP Top Ten.

11. **Monitor Third-Party Dependencies**
    Regularly update and monitor third-party libraries and dependencies to mitigate risks associated with known vulnerabilities.

12. **Implement Proper Error Handling**
    Ensure that error handling mechanisms minimize information disclosure, preventing attackers from gaining insights into the application's vulnerabilities.

13. **Develop a Security Awareness Program**
    Provide comprehensive security training for employees and stakeholders to promote best practices and create a culture of security awareness.

14. **Establish Incident Response Procedures**
    Develop a well-defined incident response plan to ensure effective response to security incidents, minimizing damage and recovery time.

These recommendations are grounded in established security frameworks, such as OWASP, NIST, and general industry best practices, and aim to create a robust security posture for web applications. They address common vulnerabilities identified during assessments and reflect a proactive approach to security management.

APPENDIX

This includes the resources utilized in detail to find out the vulnerabilities into the system.

APPENDIX A: TOOLS LEVERAGED FOR SECURITY TESTS

Name	Manufacturer	Use
Burp Suite Professional	PortSwigger	An HTTP proxy used to perform manual testing of web applications Burp Suite is used in web application security testing to intercept and analyze HTTP/S requests and responses. Security professionals leverage it to manipulate parameters, identify vulnerabilities like injection flaws and XSS, and assess session management and authentication mechanisms. Its suite of tools allows for both manual testing and automated vulnerability scanning, enabling thorough evaluation of an application's security posture. HTTP proxy allows user to trap and monitor requests sent to web server, gain a better understanding of hidden functionality and attempt to tamper with web site parameters.

APPENDIX B: PROOF OF CONCEPT

For further details regarding the vulnerability assessment results, please refer the below attached document:

1. Security Scan Report
   https://drive.google.com/drive/folders/1zQvNwwAjW1tHqspavJlqsNQSK4TNYvHc?usp=drive_link