VULNERABILITY ASSESSMENT & PENETRATION TESTING REPORT

Application Security Assessment Deliverable

Target Organization / Client	[CLIENT NAME]
Target Asset / Application	[CLIENT APPLICATION]
Assessment Type	Vulnerability Assessment & Penetration Testing
Assessment Environment	Production
Assessment Period	[DD-MM-YYYY] to [DD-MM-YYYY]
Report Release Date	[DD-MM-YYYY]
Document Version & Classification	v1.0 | CONFIDENTIAL

CONFIDENTIAL DOCUMENT

Not to be circulated or reproduced without appropriate authorization.

DISCLAIMER: ESCOSS LLP logo and trademarks are the exclusive property of ESCOSS LLP. All other trademarks and brand names are property of their respective owners. This report contains proprietary security analysis and recommendations.

1. Document Control & Governance

1.1 Document Summary

Document Title	Vulnerability Assessment & Penetration Testing Report
Target Organization	[CLIENT NAME]
Classification	Confidential
Document Type	Security Assessment Report
Application / Asset Name	[CLIENT APPLICATION]
Assessment Type	Vulnerability Assessment & Penetration Testing
Assessment Environment	Production
Application / System Type	Web Application
Assessment Period	[DD-MM-YYYY] to [DD-MM-YYYY]
Report Release Date	[DD-MM-YYYY]
Document Version	v1.0

1.2 Document Revision History

Version	Release Date	Author / Assessor	Technical Reviewer and Approver	Summary of Changes
v1.0	[DD-MM-YYYY]	[REDACTED NAME]	[REDACTED NAME]	Initial report release

1.3 Engagement & Reporting Contacts

Role / Representation	Stakeholder Name	Designation / Title	Email Address	Organization
Client	[REDACTED NAME]	Co-Founder	[REDACTED CONTACT]	[CLIENT NAME]
Lead Assessor	[REDACTED NAME]	Cyber Security Analyst	[REDACTED CONTACT]	ESCOSS LLP
Technical Reviewer	[REDACTED NAME]	Chief Executive Officer	[REDACTED CONTACT]	ESCOSS LLP

Table of Contents

1. Document Control & Governance **2**
1.1 Document Summary 2
1.2 Document Revision History 2
1.3 Engagement & Reporting Contacts 2
2. Executive Summary **5**
2.1 Engagement Overview 5
2.2 Overall Security Posture & Rating 5
2.3 Prioritized Management Action Plan 5
3. Assessment Scope & Parameters **6**
3.1 In-Scope Assets 6
3.2 Out-of-Scope Boundaries & Excluded Systems 6
3.3 Testing Limitations & Excluded Attack Types 6
4. Assessment Methodology & Approach **7**
4.1 Industry Standards & Framework Alignment 7
4.2 Phased Assessment Workflow 7
4.3 Assessment Tooling & Environment 7
5. Risk Rating Methodology & Assessment Criteria **8**
5.1 ESCOSS Risk Rating Matrix 8
5.2 Impact & Exploitability Criteria Definitions 8
6. Findings Summary & Mitigation Tracker **9**
6.1 Vulnerability Overview by Severity 9
6.2 Remediation & Mitigation Tracker 10
6.3 Vulnerability Graph 11
7. Detailed Technical Findings **12**
ESC-VAPT-001: Insufficient Rate Limiting on WordPress Authentication Endpoint 12
ESC-VAPT-002: WordPress Username Disclosure via REST API 14
ESC-VAPT-003: Username Enumeration via Differential Authentication Responses 15
ESC-VAPT-004: Clickjacking via Missing Anti-Framing Controls 17
ESC-VAPT-005 Directory Listing Enabled on Public Upload Directory 18
ESC-VAPT-006: Missing HttpOnly and Secure Parameters on Cookies 20
ESC-VAPT-007: Missing Security Response Headers 22
8. Security Framework & Control Coverage Matrices **24**
8.1 OWASP Top 10 (2025) Assessment Matrix 24
8.2 SANS / CWE Top 25 Most Dangerous Software Weaknesses 25
9. Secure Coding Guidelines & Industry Best Practices **26**
10. Assessment Conclusion & Retesting Protocol **27**

10.1 Engagement Summary & Next Steps 27
10.2 Retesting Protocol & Verification Procedure 27
11. Appendices & Legal Terms 28
11.1 Terms of Engagement & Limitation of Liability 28
11.2 Corporate Information & Engagement Sign-Off 28

2. Executive Summary

2.1 Engagement Overview

[CLIENT NAME] engaged **ESCOSS LLP** to perform a comprehensive Vulnerability Assessment & Penetration Testing (VAPT) on the target asset: **[CLIENT APPLICATION]**. The assessment was executed between [DD-MM-YYYY] and [DD-MM-YYYY] under a Black Box testing model.

The primary objective of this penetration test was to identify potential security vulnerabilities, evaluate the resilience of the target against realistic threat vectors, determine the extent to which sensitive business assets could be compromised, and provide actionable, prioritized remediation guidance to strengthen the organization's defensive posture.

2.2 Overall Security Posture & Rating

Assessed Security Rating	Executive Posture Evaluation & Business Risk Summary
MODERATE	The assessed environment demonstrates baseline security controls; however, 3 Medium severity vulnerabilities and 4 Low severity vulnerabilities were identified. The findings primarily relate to authentication hardening, username exposure, browser framing controls, public directory exposure, cookie attribute configuration, and security response headers. No Critical or High severity findings were identified.

2.3 Prioritized Management Action Plan

Priority Tier	Target Resolution Window	Key Strategic Actions & Focus Areas
Immediate (Critical / High)	**24 – 72 Hours**	No Critical or High severity findings were identified during this engagement.
Short-Term (Medium)	**7 – 14 Business Days**	**Address:** ESC-VAPT-001 (Insufficient Rate Limiting on WordPress Authentication Endpoint); ESC-VAPT-002 (WordPress Username Disclosure via REST API); ESC-VAPT-003 (Username Enumeration via Differential Authentication Responses).
Strategic / Long-Term (Low / Informational)	**30 – 60 Days**	**Address**: ESC-VAPT-004 (Clickjacking via Missing Anti-Framing Controls); ESC-VAPT-005 (Directory Listing Enabled on Public Upload Directory); ESC-VAPT-006 (Missing HttpOnly and Secure Attributes on Cookies); ESC-VAPT-007 (Missing Security Response Headers).

3. Assessment Scope & Parameters

3.1 In-Scope Assets

The following target assets were explicitly authorized and designated as in-scope for this penetration testing engagement:

Asset ID	Target Name / URL / Hostname	Asset Type	IP / CIDR Range	Environment	Testing Approach
[CONFIDENTIAL INFORMATION REMOVED]	[REDACTED URL]	Web Application	NA	Production	Black Box

3.2 Out-of-Scope Boundaries & Excluded Systems

Any asset, IP address, subdomain, third-party integration, payment gateway API, or cloud infrastructure not explicitly listed in Section 3.1 was strictly considered **OUT OF SCOPE**. Testing activities were restricted to authorized target endpoints to prevent collateral disruption.

3.3 Testing Limitations & Excluded Attack Types

To ensure operational continuity and protect production data integrity, the following testing constraints and exclusions were maintained throughout the engagement:

- **Denial of Service (DoS / DDoS):** Flooding attacks, resource exhaustion exploits, and distributed volumetric tests were strictly prohibited to avoid disruption to business services.
- **Social Engineering & Phishing:** Spear-phishing, employee pretexting, credential harvesting campaigns, and physical unauthorized access attempts were excluded from this engagement scope.
- **Destructive Payload Execution:** Exploits causing permanent data loss, database corruption, or irreversible system modification were not executed. Verification was conducted non-destructively.
- **Third-Party Hosted Services:** Direct attacks on shared cloud providers, multi-tenant infrastructure, or third-party SaaS vendors outside client ownership were omitted.

4. Assessment Methodology & Approach

ESCOSS LLP conducts security assessments utilizing industry-recognized penetration testing standards, ensuring systematic, repeatable, and thorough vulnerability discovery. Our methodology harmonizes manual security expertise with automated tooling to deliver deep technical validation with zero false positives.

4.1 Industry Standards & Framework Alignment

- **OWASP Top 10 (2025):** Evaluation against the latest consensus standard for critical web application security risks.
- **OWASP Application Security Verification Standard (ASVS v5.0):** Granular technical security controls verification for web applications and web services.
- **NIST SP 800-115:** Technical Guide to Information Security Testing and Assessment principles and methodologies.
- **PTES (Penetration Testing Execution Standard):** End-to-end framework covering scoping, intelligence gathering, threat modeling, exploitation, and reporting.
- **SANS / CWE Top 25:** Verification against the most dangerous, prevalent software weaknesses and implementation flaws.

4.2 Phased Assessment Workflow

Engagement Phase	Key Technical Activities & Objectives
Phase 1: Planning & Scoping	Define assessment scope, establish rules of engagement, obtain formal testing authorization, confirm the approved testing window, and establish communication and escalation channels.
Phase 2: Reconnaissance & OSINT	Perform domain enumeration, target surface mapping, technological stack profiling, server banner identification, and endpoint discovery.
Phase 3: Threat Modeling & Vulnerability Discovery	Analyze application business logic, map access control boundaries, conduct automated scanning, and enumerate input vectors across all functional tiers.
Phase 4: Controlled Verification & Exploitation	Manually validate potential flaws, execute controlled proof-of-concept exploits, evaluate privilege escalation chains, and eliminate false positives.
Phase 5: Impact Analysis & Risk Scoring	Determine realistic business and technical impact, calculate standardized CVSS v3.1, and establish tailored remediation steps.
Phase 6: Reporting & Retesting	Deliver comprehensive executive and technical findings, conduct technical debrief with stakeholders, and execute verification retesting on patched vulnerabilities.

4.3 Assessment Tooling & Environment

Assessments are conducted by experienced security consultants utilizing enterprise-grade commercial and open-source security tooling, including Burp Suite, OWASP ZAP, Postman, Nmap, Nuclei and proprietary ESCOSS security automation scripts. All automated results undergo strict manual validation to eliminate false positives.

5. Risk Rating Methodology & Assessment Criteria

ESCOSS LLP assigns risk ratings based on a qualitative and quantitative assessment of the potential business impact and the ease with which a vulnerability can be exploited by an adversary. This ensures risk prioritization aligns with real-world threat models.

5.1 ESCOSS Risk Rating Matrix

Access	Ease of Exploitation	High Impact	Medium Impact	Low Impact
Remote / Public (Unauthenticated)	Easy	**CRITICAL**	**HIGH**	**MEDIUM**
Remote / Public (Unauthenticated)	Moderate	**HIGH**	**MEDIUM**	**LOW**
Remote / Public (Unauthenticated)	Difficult	**MEDIUM**	**LOW**	**INFORMATIONAL**
Authenticated (Standard User)	Easy	**HIGH**	**MEDIUM**	**LOW**
Authenticated (Standard User)	Moderate	**MEDIUM**	**LOW**	**INFORMATIONAL**
Privileged / Local Access	Difficult	**LOW**	**INFORMATIONAL**	**INFORMATIONAL**

Note: *The ESCOSS Risk Rating Matrix provides a qualitative risk-prioritization framework based on access requirements, exploitability, and business impact. Final vulnerability severity is determined using the applicable CVSS assessment together with the contextual business and technical impact observed during the engagement. Accordingly, the qualitative matrix should be considered a supporting risk-prioritization reference and not as a direct replacement for the reported CVSS-based severity rating.*

CVSS Alignment Note: *CVSS Base Score provides a standardized assessment of technical severity but does not fully capture application-specific functionality, business context, asset criticality, or the actual business impact demonstrated during the engagement. Accordingly, the* ***ESCOSS severity rating is determined independently using the applicable CVSS assessment, the ESCOSS Risk Rating Matrix, applicable industry standards, and the contextual business and technical impact observed during testing. Therefore, the ESCOSS severity rating and CVSS score should not be assumed to be identical.***

5.2 Impact & Exploitability Criteria Definitions

- **High Business Impact:** Direct unauthorized access to sensitive financial or PII data, complete system take-over, arbitrary code execution on backend servers, or severe regulatory non-compliance.
- **Medium Business Impact:** Partial compromise of data integrity, unauthorized access to non-critical internal functions, elevated operational cost exposure, or limited information disclosure.
- **Low Business Impact:** Minor information disclosure (software versions, non-sensitive metadata) with minimal operational impact and negligible likelihood of direct exploitation.
- **Easy Exploitability:** Vulnerability can be exploited remotely with minimal skill using automated or standard public exploit techniques without authentication.
- **Moderate Exploitability:** Requires valid authenticated access, specific system conditions, user interaction (e.g. CSRF/phishing), or custom exploit adaptation.
- **Difficult Exploitability:** Requires advanced domain expertise, non-default system configurations, complex exploit chaining, or physical/local access.

6. Findings Summary & Mitigation Tracker

Critical	High	Medium	Low	Informational
0	0	3	4	0

Total Count
7

6.1 Vulnerability Overview by Severity

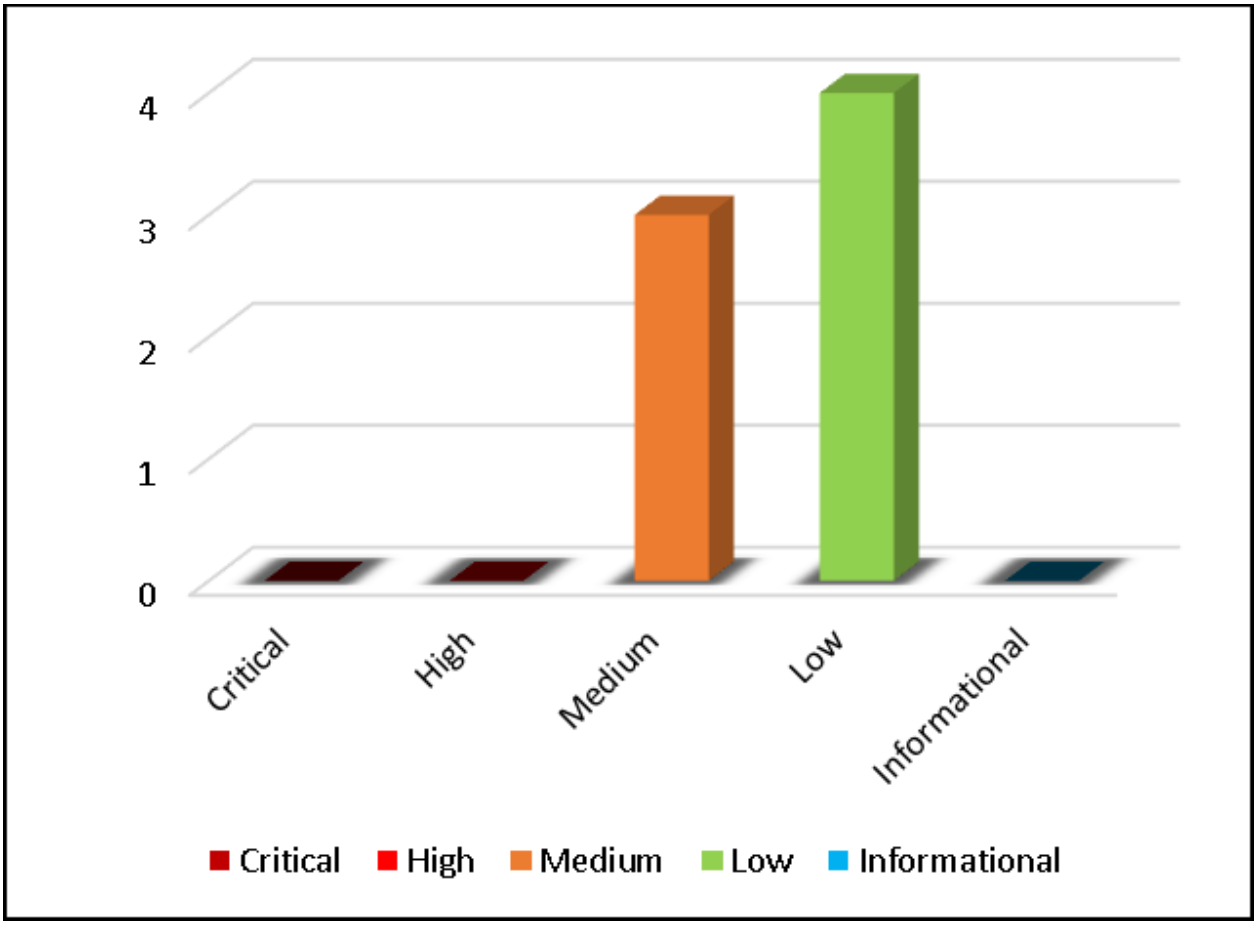
#	Finding ID	Vulnerability Name	Severity	CVSS	Affected Asset	Status
1	ESC-VAPT-001	Insufficient Rate Limiting on WordPress Authentication Endpoint	MEDIUM	5.3	[REDACTED URL]	OPEN
2	ESC-VAPT-002	WordPress Username Disclosure via REST API	MEDIUM	5.3	[REDACTED URL]	OPEN
3	ESC-VAPT-003	Username Enumeration via Differential Authentication Responses	MEDIUM	5.3	[REDACTED URL]	OPEN
4	ESC-VAPT-004	Clickjacking via Missing Anti-Framing Controls	LOW	4.3	[REDACTED URL]	OPEN
5	ESC-VAPT-005	Directory Listing Enabled on Public Upload Directory	LOW	5.3	[REDACTED URL]	OPEN
6	ESC-VAPT-006	Missing HttpOnly and Secure Attributes on Cookies	LOW	3.1	[REDACTED URL]	OPEN
7	ESC-VAPT-007	Missing Security Response Headers	LOW	4.3	[REDACTED URL]	OPEN

6.2 Remediation & Mitigation Tracker

The mitigation tracker establishes operational ownership and accountability for remediation actions. It should be updated continuously as patches are implemented and verified during retesting cycles.

Finding ID	Vulnerability Name	Severity	Remediation Owner	Priority	Target Date	Retest Status	Closure Date
ESC-VAPT-001	Insufficient Rate Limiting on WordPress Authentication Endpoint	MEDIUM	[REDACTED NAME]	Short-Term	[DD-MM-YYYY]	PENDING	—
ESC-VAPT-002	WordPress Username Disclosure via REST API	MEDIUM	[REDACTED NAME]	Short-Term	[DD-MM-YYYY]	PENDING	—
ESC-VAPT-003	Username Enumeration via Differential Authentication Responses	MEDIUM	[REDACTED NAME]	Short-Term	[DD-MM-YYYY]	PENDING	—
ESC-VAPT-004	Clickjacking via Missing Anti-Framing Controls	LOW	[REDACTED NAME]	Strategic	[DD-MM-YYYY]	PENDING	—
ESC-VAPT-005	Directory Listing Enabled on Public Upload Directory	LOW	[REDACTED NAME]	Strategic	[DD-MM-YYYY]	PENDING	—
ESC-VAPT-006	Missing HttpOnly and Secure Attributes on Cookies	LOW	[REDACTED NAME]	Strategic	[DD-MM-YYYY]	PENDING	—
ESC-VAPT-007	Missing Security Response Headers	LOW	[REDACTED NAME]	Strategic	[DD-MM-YYYY]	PENDING	—

6.3 Vulnerability Graph



7. Detailed Technical Findings

This section provides comprehensive technical documentation for each identified vulnerability, including vulnerability classification, CVSS metrics, affected endpoints, detailed technical observation, business and technical impact analysis, step-by-step reproducible proof of concept (PoC), and actionable remediation guidance.

ESC-VAPT-001: Insufficient Rate Limiting on WordPress Authentication Endpoint

Severity Rating	**MEDIUM**
CVSS v3.1 Base Score	**5.3 CVSS:3.1/AV:N/AC:L/PR:N/UI:N/S:U/C:N/I:N/A:L**
Classification Taxonomy	CWE-307 - Improper Restriction of Excessive Authentication Attempts | A07:2025-Authentication Failure
Affected Asset / URL	[REDACTED URL]
Affected Component / Parameter	[REDACTED DOMAIN]/wp-login.php
Authentication / Access Level	None
Ease of Exploitation	Easy

Vulnerability Description & Root Cause

The WordPress authentication endpoint accepts repeated invalid password attempts against the same username without effective server-side throttling or temporary blocking. During testing, 25 password-guessing requests were submitted for the username “[REDACTED USERNAME]” and the endpoint continued processing the attempts without CAPTCHA, account lockout, progressive delay, temporary blocking, or equivalent throttling controls.

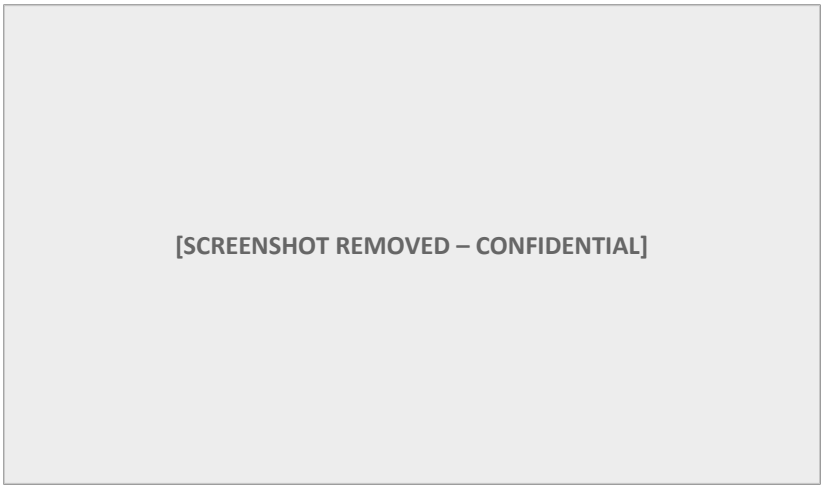
Business & Technical Impact Analysis

The absence of rate limiting increases the feasibility of automated password-guessing and credential attacks against known usernames. The observed behavior did not establish successful account compromise; however, it materially reduces an important defensive barrier around the authentication function.

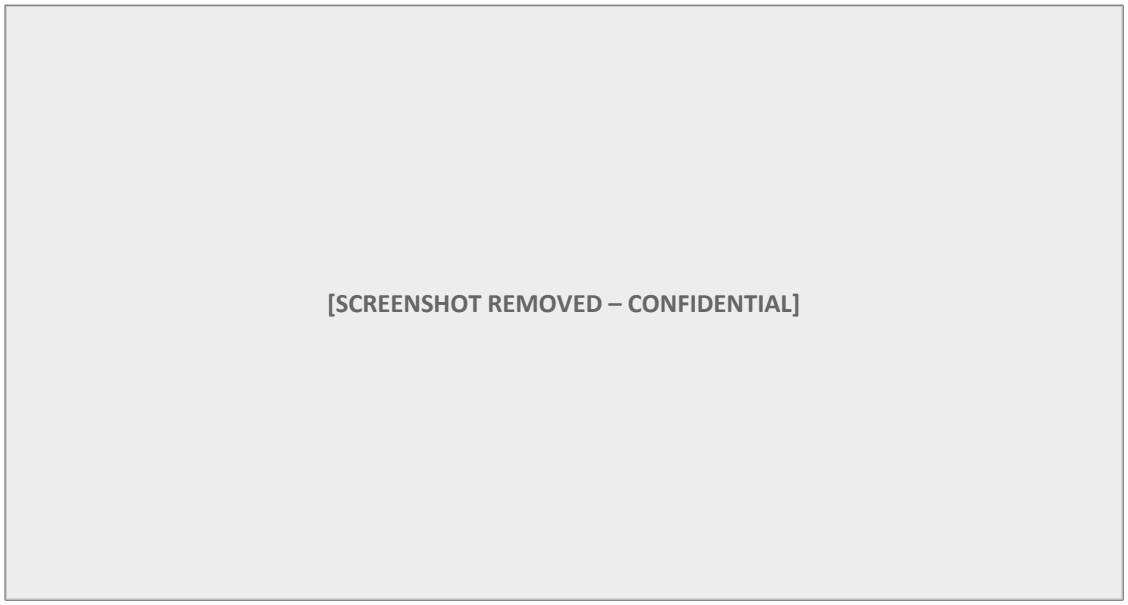
Step-by-Step Proof of Concept & Evidence

Step-1:

As shown in the image below, this is the target login page at wp-login.php



Step-2:
The POC demonstrates the absence of effective rate-limiting controls, as repeated authentication requests continued to receive HTTP 200 OK responses with consistent response content, indicating that the requests were processed without effective throttling



Remediation & Secure Coding Guidance

Implement server-side rate limiting for authentication attempts using layered controls such as per-account and per-IP throttling, progressive delays, temporary lockouts, CAPTCHA or equivalent abuse protections, and monitoring for repeated failed authentication patterns. Ensure controls are enforced at the application/backend layer and cannot be bypassed by manipulating client-side requests.

Remediation Verification & Retest Status

Retest Status	Retest Date	Verified By	Retest Finding Result
PENDING	—	NA	NA

References & Vulnerability Taxonomies

- **CWE-307:** https://cwe.mitre.org/data/definitions/307.html

ESC-VAPT-002: WordPress Username Disclosure via REST API

Severity Rating	**MEDIUM**
CVSS v3.1 Base Score	**5.3 CVSS:3.1/AV:N/AC:L/PR:N/UI:N/S:U/C:L/I:N/A:N**
Classification Taxonomy	CWE-200 – Exposure of Sensitive Information to an Unauthorized Actor | A02:2025 – Security Misconfiguration
Affected Asset / URL	[REDACTED URL]
Affected Component / Parameter	[REDACTED DOMAIN]/wp-json/
Authentication / Access Level	None
Ease of Exploitation	Easy

Vulnerability Description & Root Cause

The public WordPress REST API root exposes information that discloses a valid WordPress username. The assessed endpoint returned REST API content without requiring authentication, and the username associated with the WordPress login function was identified as "[REDACTED USERNAME]"

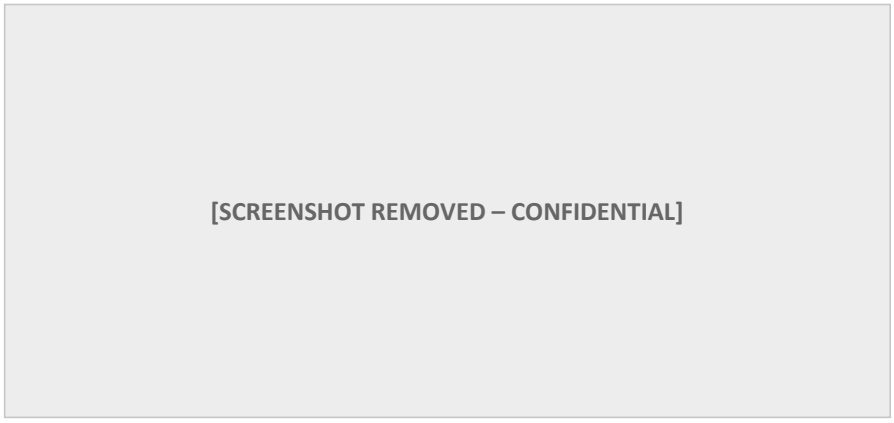
Business & Technical Impact Analysis

Direct disclosure of a valid username reduces the attacker's uncertainty about the authentication surface and can support targeted credential-guessing or username-enumeration activity. The issue becomes more actionable when combined with the observed absence of effective authentication rate limiting.

Step-by-Step Proof of Concept & Evidence

Step–1:

The public WordPress REST API root was requested at /wp-json/, and the WordPress username "[REDACTED USERNAME]" was identified within the exposed information



Remediation & Secure Coding Guidance

Review WordPress REST API exposure and restrict unnecessary user-identifying information from unauthenticated responses. Where user enumeration is not required, disable or limit public user discovery and ensure authentication-related identity data is not unnecessarily exposed through public API responses. Validate the effectiveness of changes using unauthenticated requests.

Remediation Verification & Retest Status

Retest Status	Retest Date	Verified By	Retest Finding Result
PENDING	—	NA	NA

References & Vulnerability Taxonomies

- **CWE-200:** https://cwe.mitre.org/data/definitions/200.html

ESC-VAPT-003: Username Enumeration via Differential Authentication Responses

Severity Rating	**MEDIUM**
CVSS v3.1 Base Score	**5.3 CVSS:3.1/AV:N/AC:L/PR:N/UI:N/S:U/C:L/I:N/A:N**
Classification Taxonomy	CWE-203 – Observable Discrepancy | A07:2025 – Authentication Failures
Affected Asset / URL	[REDACTED URL]
Affected Component / Parameter	[REDACTED DOMAIN]/wp-login.php
Authentication / Access Level	None
Ease of Exploitation	Easy

Vulnerability Description & Root Cause

The WordPress login interface returns distinguishable error messages for different username conditions. By comparing the response generated for a valid username with the response generated for an invalid username, an unauthenticated requester can determine whether a username exists.

Business & Technical Impact Analysis

Username enumeration narrows the set of identifiers an attacker needs to target during credential attacks and can be used as a reconnaissance step before password guessing. The observed behavior becomes more significant when combined with the separately identified authentication rate-limiting weakness.

Step-by-Step Proof of Concept & Evidence

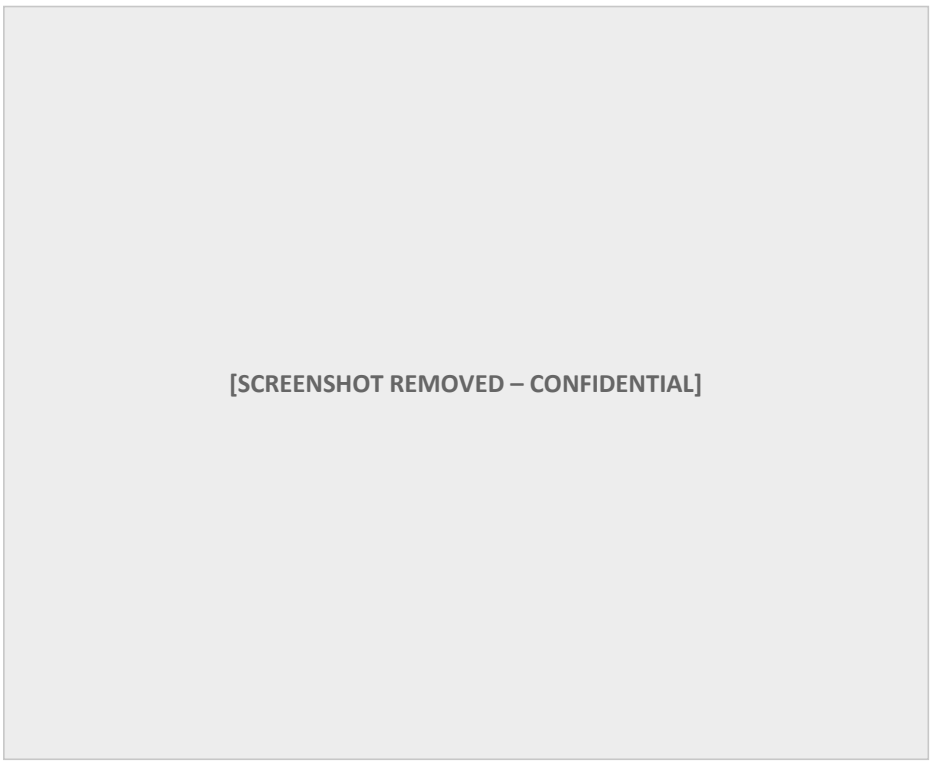
Step-1:

As demonstrated, as we submit known-invalid username to wordpress, it gives an informative error saying that username is invalid.

[SCREENSHOT REMOVED – CONFIDENTIAL]

Step-2:

Here, we have submitted valid known username to wordpress as it gives a informative error saying that password for the entered username is invalid.



Remediation & Secure Coding Guidance

Return a uniform authentication failure response regardless of whether the supplied username exists. Keep error message content, status handling, and other observable response characteristics as consistent as practical so that account existence cannot be reliably inferred from unauthenticated responses.

Remediation Verification & Retest Status

Retest Status	Retest Date	Verified By	Retest Finding Result
PENDING	—	NA	NA

References & Vulnerability Taxonomies

- **CWE-203:** https://cwe.mitre.org/data/definitions/203.html

ESC-VAPT-004: Clickjacking via Missing Anti-Framing Controls

Severity Rating	**LOW**
CVSS v3.1 Base Score	**4.3 CVSS:3.1/AV:N/AC:L/PR:N/UI:R/S:U/C:N/I:L/A:N**
Classification Taxonomy	CWE-1021 – Improper Restriction of Rendered UI Layers or Frames | A02:2025 – Security Misconfiguration
Affected Asset / URL	[REDACTED URL]
Affected Component / Parameter	[REDACTED URL]
Authentication / Access Level	None
Ease of Exploitation	Moderate

Vulnerability Description & Root Cause

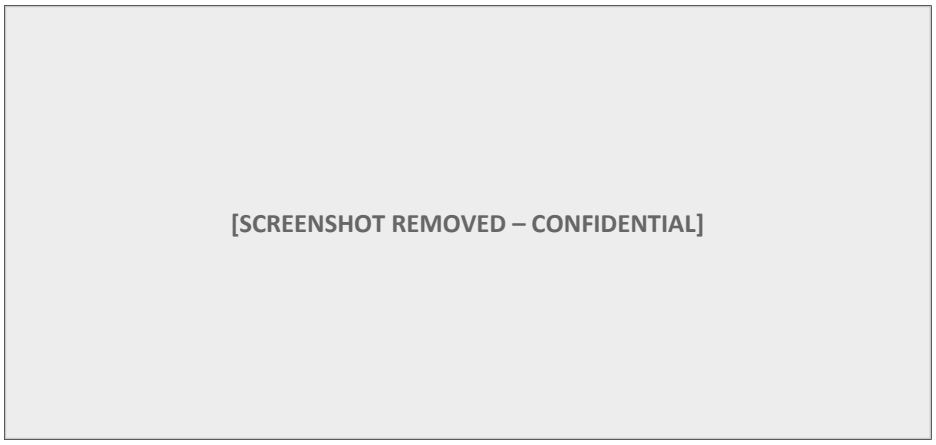
The public homepage can be embedded within an attacker-controlled iframe and remains visibly rendered and usable. The observed response does not implement effective anti-framing controls such as X-Frame-Options or CSP frame-ancestors for the affected page.

Business & Technical Impact Analysis

An attacker-controlled page may visually embed the target website and use deceptive overlays or interface positioning to induce a victim to interact with the framed content. The provided evidence demonstrates framing of the public homepage; no specific sensitive authenticated action was established during testing.

Step-by-Step Proof of Concept & Evidence

Step-1:
As Shown in the POC below, the page was loaded inside an attacker-controlled iframe.



.

Remediation & Secure Coding Guidance

Configure anti-framing protections at the application/server layer. Use Content-Security-Policy with an appropriate frame-ancestors directive and/or X-Frame-Options (for example, DENY or SAMEORIGIN as required by application functionality). Validate the response headers on all relevant pages.

Remediation Verification & Retest Status

Retest Status	Retest Date	Verified By	Retest Finding Result
PENDING	—	NA	NA

References & Vulnerability Taxonomies

- CWE-1021: https://cwe.mitre.org/data/definitions/1021.html

ESC-VAPT-005 Directory Listing Enabled on Public Upload Directory

Severity Rating	**LOW**
CVSS v3.1 Base Score	**5.3 CVSS:3.1/AV:N/AC:L/PR:N/UI:N/S:U/C:L/I:N/A:N**
Classification Taxonomy	CWE-548 – Exposure of Information Through Directory Listing | A02:2025 – Security Misconfiguration
Affected Asset / URL	[REDACTED URL]
Affected Component / Parameter	[REDACTED DOMAIN]/wp-content/uploads/
Authentication / Access Level	None
Ease of Exploitation	Easy

Vulnerability Description & Root Cause

The public upload directory is browsable without authentication and returns an index listing of directories and uploaded content. The directory response exposes the underlying upload structure and makes files within the web-accessible storage location directly discoverable. No sensitive files or confidential information were identified in the observed listing.

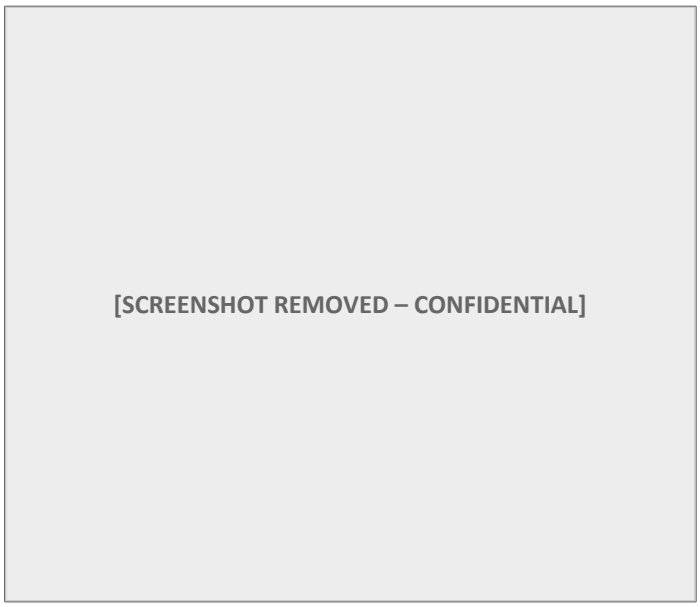
Business & Technical Impact Analysis

Public directory indexing increases the discoverability of uploaded content and application storage structure. Even where the currently exposed files are non-sensitive, the condition can facilitate access to stale, unintended, or subsequently uploaded content that was not designed for unrestricted directory discovery.

Step-by-Step Proof of Concept & Evidence

Step-1:

As shown below, The listing exposed directories and actual uploaded files/content.



Remediation & Secure Coding Guidance

Disable directory indexing for the web-accessible uploads directory and configure the web server to return a suitable response when an index resource is absent. Review existing upload locations and ensure uploaded files are accessible only through intended application workflows where appropriate.

Remediation Verification & Retest Status

Retest Status	Retest Date	Verified By	Retest Finding Result
PENDING	—	NA	NA

References & Vulnerability Taxonomies

- **CWE-548**: https://cwe.mitre.org/data/definitions/548.html

ESC-VAPT-006: Missing HttpOnly and Secure Attributes on Cookies

Severity Rating	**LOW**
CVSS v3.1 Base Score	**3.1 CVSS:3.1/AV:N/AC:H/PR:N/UI:R/S:U/C:L/I:N/A:N**
Classification Taxonomy	CWE-16 – Configuration | A02:2025 - Security Misconfiguration
Affected Asset / URL	[REDACTED URL]
Affected Component / Parameter	Cookies
Authentication / Access Level	None
Ease of Exploitation	Moderate

Vulnerability Description & Root Cause

The cookies are set without the HttpOnly and Secure attributes. The cookie observed in the browser is a non-authentication / analytics cookie, so the demonstrated condition does not directly expose an authenticated session token. The missing attributes nevertheless reduce browser-side cookie hardening and defense in depth.

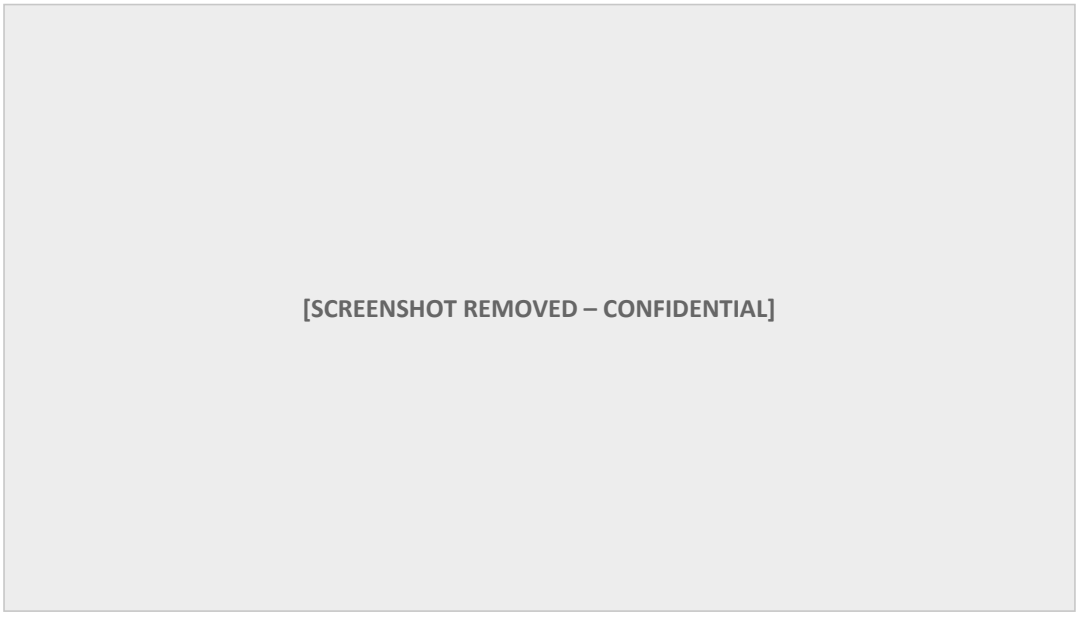
Business & Technical Impact Analysis

Without HttpOnly, client-side scripts can access the cookie value. Without Secure, the browser is not explicitly instructed to restrict the cookie to secure transport. The practical impact is limited by the identified non-authentication nature of the cookie, but the missing controls can increase exposure if the cookie value becomes useful to an attacker through a secondary browser-side or transport-layer weakness.

Step-by-Step Proof of Concept & Evidence

Step-1:

The evidence below demonstrates that, cookies are implemented without HttpOnly and Secure Attribute.



Remediation & Secure Coding Guidance

Apply the Secure attribute to cookies that should only be transmitted over HTTPS and apply HttpOnly to cookies that do not require client-side script access. Review all cookies issued by the application, classify them by function, and apply an appropriate SameSite policy and restrictive scope in addition to Secure/HttpOnly where applicable.

Remediation Verification & Retest Status

Retest Status	Retest Date	Verified By	Retest Finding Result
PENDING	—	NA	NA

References & Vulnerability Taxonomies

- **CWE-16:** https://cwe.mitre.org/data/definitions/16.html

ESC-VAPT-007: Missing Security Response Headers

Severity Rating	**LOW**
CVSS v3.1 Base Score	**4.3 CVSS:3.1/AV:N/AC:L/PR:N/UI:R/S:U/C:N/I:L/A:N**
Classification Taxonomy	CWE-693: Protection Mechanism Failure | A02:2025 – Security Misconfiguration
Affected Asset / URL	[REDACTED URL]
Affected Component / Parameter	[REDACTED URL]
Authentication / Access Level	None
Ease of Exploitation	Easy

Vulnerability Description & Root Cause

The HTTP response observed for the public homepage does not include several commonly deployed browser security response headers. The provided evidence shows an HTTP 200 response with the following headers absent: Strict-Transport-Security, Content-Security-Policy, X-Frame-Options, X-Content-Type-Options, Referrer-Policy, and Permissions-Policy.

Business & Technical Impact Analysis

The absence of these headers weakens browser-side defense in depth and removes controls that can mitigate or constrain classes of attacks such as clickjacking, MIME-type sniffing, excessive referrer disclosure, and unwanted browser feature use. The finding is treated separately from the clickjacking finding; the response-header issue itself represents a broader configuration weakness.

Step-by-Step Proof of Concept & Evidence

Step-1:

The evidence below demonstrates that, website lacks many security response headers.

[SCREENSHOT REMOVED – CONFIDENTIAL]

Remediation & Secure Coding Guidance

Deploy an appropriate security header baseline for the application and server stack. Configure Content-Security-Policy, Strict-Transport-Security, X-Frame-Options or CSP frame-ancestors, X-Content-Type-Options: nosniff, Referrer-Policy, and Permissions-Policy according to the application's functional requirements. Verify headers on representative application responses rather than on a single page only.

Remediation Verification & Retest Status

Retest Status	Retest Date	Verified By	Retest Finding Result
PENDING	—	NA	NA

References & Vulnerability Taxonomies

- **CWE-693 -** https://cwe.mitre.org/data/definitions/693.html

8. Security Framework & Control Coverage Matrices

8.1 OWASP Top 10 (2025) Assessment Matrix

The table below details the target application's compliance status against the OWASP Top 10 (2025) security benchmark categories:

Category ID	Category Name	Assessment Status	Mapped Findings
A01:2025	Broken Access Control	SAFE	N/A
A02:2025	Security Misconfiguration	UNSAFE	ESC-VAPT-002, ESC-VAPT-004, ESC-VAPT-005, ESC-VAPT-006, ESC-VAPT-007
A03:2025	Software Supply Chain Failures	SAFE	N/A
A04:2025	Cryptographic Failures	SAFE	N/A
A05:2025	Injection	SAFE	N/A
A06:2025	Insecure Design	SAFE	N/A
A07:2025	Authentication Failure	UNSAFE	ESC-VAPT-001, ESC-VAPT-003
A08:2025	Software or Data Integrity Failures	SAFE	N/A
A09:2025	Security Logging and Alerting Failures	SAFE	N/A
A10:2025	Mishandling of Exceptional Conditions	SAFE	N/A

8.2 SANS / CWE 2025 Most Dangerous Software Weaknesses

Rank	CWE ID	Weakness Name & Taxonomy Description	Assessment Status
1	CWE-79	Improper Neutralization of Input During Web Page Generation ('Cross-site Scripting')	SAFE
2	CWE-89	Improper Neutralization of Special Elements used in an SQL Command ('SQL Injection')	SAFE
3	CWE-352	Cross-Site Request Forgery (CSRF)	SAFE
4	CWE-862	Missing Authorization	SAFE
5	CWE-787	Out-of-bounds Write	SAFE
6	CWE-22	Improper Limitation of a Pathname to a Restricted Directory ('Path Traversal')	SAFE
7	CWE-416	Use After Free	SAFE
8	CWE-125	Out-of-bounds Read	SAFE
9	CWE-78	Improper Neutralization of Special Elements used in an OS Command ('OS Command Injection')	SAFE
10	CWE-94	Improper Control of Generation of Code ('Code Injection')	SAFE
11	CWE-120	Buffer Copy without Checking Size of Input ('Classic Buffer Overflow')	SAFE
12	CWE-434	Unrestricted Upload of File with Dangerous Type	SAFE
13	CWE-476	NULL Pointer Dereference	SAFE
14	CWE-121	Stack-based Buffer Overflow	SAFE
15	CWE-502	Deserialization of Untrusted Data	SAFE
16	CWE-122	Heap-based Buffer Overflow	SAFE
17	CWE-863	Incorrect Authorization	SAFE
18	CWE-20	Improper Input Validation	SAFE
19	CWE-284	Improper Access Control	SAFE
20	CWE-200	Exposure of Sensitive Information to an Unauthorized Actor	UNSAFE
21	CWE-306	Missing Authentication for Critical Function	SAFE
22	CWE-918	Server-Side Request Forgery (SSRF)	SAFE
23	CWE-77	Improper Neutralization of Special Elements used in a Command ('Command Injection')	SAFE
24	CWE-639	Authorization Bypass Through User-Controlled Key	SAFE
25	CWE-770	Allocation of Resources Without Limits or Throttling	SAFE

9. Secure Coding Guidelines & Industry Best Practices

To foster long-term cyber resilience and prevent systemic vulnerabilities, development and operations teams should incorporate the following secure coding principles into the organization's Secure Software Development Lifecycle (SSDLC):

- **Input Validation & Context-Aware Output Encoding:** Validate all user inputs against strict allowlists (data types, lengths, character sets). Encode all dynamic output based on the rendering context (HTML, JS, CSS, SQL).
- **Parameterized Queries & ORM Defense:** Strictly utilize parameterized SQL queries, prepared statements, or hardened ORMs to completely prevent SQL and query injection attacks.
- **Strong Authentication & Multi-Factor Enforcement:** Implement MFA across all administrative and user portals. Enforce robust password complexity, secure password hashing (Argon2id, bcrypt), and progressive throttling.
- **Strict Authorization & Object-Level Access Control:** Enforce server-side authorization checks on every request. Validate tenant boundaries and object ownership (prevent IDOR/BOLA).
- **Cryptographic Protection for Data at Rest & in Transit:** Enforce TLS 1.3/1.2 with strong cipher suites across all endpoints. Encrypt sensitive data at rest using AES-256 or authenticated encryption (GCM).
- **Secure Session Lifecycle Management:** Generate cryptographically random session tokens. Invalidate sessions upon logout or inactivity timeouts. Enforce SameSite, Secure, and HttpOnly cookie attributes.
- **Hardened Security Headers & Content Security Policy (CSP):** Deploy comprehensive defensive HTTP headers: Content-Security-Policy, Strict-Transport-Security (HSTS), X-Content-Type-Options: nosniff, and X-Frame-Options: DENY.
- **Dependency Management & Software Bill of Materials (SBOM):** Regularly audit third-party libraries, packages, and frameworks. Automate dependency vulnerability scanning within CI/CD build pipelines.
- **Comprehensive Security Logging & Real-Time Alerting:** Log security-relevant events (authentication failures, access control denials, administrative modifications) without recording sensitive credentials or PII.
- **Regular Security Assurance & Automated Scanning:** Conduct continuous automated SAST/DAST testing during development and periodic manual penetration testing prior to major production releases.

10. Assessment Conclusion & Retesting Protocol

10.1 Engagement Summary & Next Steps

The security assessment conducted by ESCOSS LLP on [CLIENT APPLICATION] provided valuable insights into the current security posture of [CLIENT NAME]'s assets.

ESCOSS LLP recommends that [CLIENT NAME] prioritize patches according to the Management Action Plan in Section 2.3. Upon completion of development fixes, a formal retesting cycle should be scheduled to verify that all vulnerabilities have been remediated effectively without introducing secondary security regressions.

10.2 Retesting Protocol & Verification Procedure

1. Remediation Notification: Client technical team notifies ESCOSS LLP with details of applied patches and deployed environments.
2. Retesting Execution: ESCOSS security assessors re-execute the specific PoC steps for each reported finding.
3. Verification Status: Findings are classified as 'Verified Fixed', 'Partially Remediated', or 'Open / Not Resolved'.
4. Final Report Release: An updated final report version (e.g. V2 Retest) is issued with updated findings and mitigation tracker.

11. Appendices & Legal Terms

11.1 Terms of Engagement & Limitation of Liability

This security assessment reflects the state of the target system during the designated testing window: [DD-MM-YYYY] to [DD-MM-YYYY]. Due to the dynamic nature of computer security and continuous evolution of attack techniques, no security assessment can guarantee the total absence of vulnerabilities.

ESCOSS LLP exercised reasonable professional skill, care, and diligence conforming to recognized cybersecurity consulting standards. ESCOSS LLP shall not be liable for any indirect, incidental, or consequential damages resulting from the use of or reliance on this report, nor for vulnerabilities emerging after the conclusion of the assessment window.

11.2 Corporate Information & Engagement Sign-Off

Prepared & Submitted By	Technical Review & Quality Approval
ESCOSS LLP	**Quality & Technical Review Board Technical Reviewer and Approver: [REDACTED NAME]** **Title: Chief Executive Officer** **Email:** **[REDACTED CONTACT]**

[CONFIDENTIAL INFORMATION REMOVED]