­

***05 July 2025***



***C-G-WEB-Q3-2025***

­

**G**

Web Application VAPT

***C-G-WEB-Q3-2025***

**© Copyright 2025 CyberSmithSECURE Private Limited**

**Disclaimer**

All information contained in this document is confidential and proprietary to CyberSmithSECURE and Reproduction, disclosure or use of any information contained in this document by photographic, electronic or any other means, in whole or part, for any reason other than for the purpose of operations is strictly prohibited without written consent.

**Limitations on Disclosure and Use**

This document contains sensitive and confidential information concerning vulnerabilities of target applications. CyberSmithSECURE recommends that special precautions be taken to protect the confidentiality of the information contained in this report.

While the VAPT Team is confident that the major security vulnerabilities of the target applications have been identified, there can be no assurance that an assessment of this nature will identify all possible security exposures. Additionally, the findings and recommendations presented in this document are based on the technologies and known threats as of the date of this report. As technologies and risks change over time, the vulnerabilities and the recommendations associated with the target applications may also change.

***C-G-WEB-Q3-2025***

Table of Content

[**Document Control 3**](#_Toc24836)

[**RISK LEVEL & DESCRIPTION 4**](#_Toc30375)

[**SCAN Type 5**](#_Toc17755)

[**List of Tools. 6**](#_Toc14108)

[**Overall Findings 7**](#_Toc25134)

[**G Risk Assessment Analysis of the Entire Framework 7**](#_Toc21028)

[**Test Case 8**](#_Toc6190)

[**Vulnerabilities Found 9**](#_Toc2041)

[**URL 10**](#_Toc10192)

[**1) https://G.com 10**](#_Toc7126)

***C-G-WEB-Q3-2025***

# **Document Control**

|  |  |
| --- | --- |
| Document Name | G Web Application VAPT |
| Document ID | C-G-WEB-Q3-2025 |
| Security Classification | External |

|  |  |
| --- | --- |
| Authorization | |
| Reviewed By | Authorized By |
| Mr. George | Mr. Aldrin |

|  |  |  |  |
| --- | --- | --- | --- |
| Amendment Log | | | |
| Version | Date | Reviewed By | Brief description of the change |
|  | | | |
| V1.0 | 2025-12-26 | Mr. George | Changes in impact |
| V2.0 | 2025-12-15 | Mr. Clint | Severity changes |
| V3.0 | 2025-12-12 | Mr. George | Revalidation status |

***C-G-WEB-Q3-2025***

# **RISK LEVEL & DESCRIPTION**

The below vulnerability ranging risk pattern indicates the ratings of the vulnerability according to their respective CVSS3.1 Score.



***C-G-WEB-Q3-2025***

# **SCAN Type**

A picture containing screenshot, diagram, circle, graphics

Description automatically generated



The application security assessment is done:

**Black Box Penetration Testing**

The security vulnerabilities were reported for the following quarter on:

**11th March 2025 to 29th March 2025**

Sample Assessment Note

# **List of Tools.**

***C-G-WEB-Q3-2025***



# **Test Case**

***C-G-WEB-Q3-2025***

|  |  |  |
| --- | --- | --- |
| **Sr. No.** | **Test Case** | **Status** |
| 1 | PORT SCANNING | A picture containing drawing  Description automatically generated**Completed** |
| 2 | Banner Grabbing/OS Fingerprinting | **A picture containing drawing  Description automatically generatedCompleted** |
| 3 | Certificate expiry | **A picture containing drawing  Description automatically generatedCompleted** |
| 4 | Identify known vulnerabilities | **A picture containing drawing  Description automatically generatedCompleted** |

***C-G-WEB-Q3-2025***

# **Overall Findings**

**Risk Assessment Analysis of the IPs**

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Sr. No.** | **Hostname** | **Instant Purpose** | **VAPT Status** | **Critical** | **High** | **Medium** | **Low** | **Informational** | **Total** |
| 1 | 10.50.33.2 | Public IP | Completed | 1 | 2 | 2 | 0 | 0 | 5 |
| 2 | 10.50.33.21 | Public IP | Completed | 1 | 2 | 2 | 0 | 0 | 5 |
| **Overall Findings** | | | | **2** | **4** | **4** | **0** | **0** | **10** |

***C-G-WEB-Q3-2025***

# **Vulnerabilities Found**

\_VULNERABILITIES\_FOUND\_PLACEHOLDER\_

***C-G-WEB-Q3-2025***

**IP 10.50.33.2**

|  |  |  |
| --- | --- | --- |
| **Sr. No.** | **Vulnerability Name** | **Vulnerability Risk Type** |
| **1** | **Unauthenticated File Upload** | **Critical** |
| **2** | **SQL Injection in Login** | **High** |
| **3** | **Sensitive Data Exposure** | **High** |
| **4** | **Cross-Site Scripting (XSS)** | **Medium** |
| **5** | **Insecure Direct Object References** | **Medium** |

|  |
| --- |
| **VUL-001: Unauthenticated File Upload** |
| **IP: 10.50.33.2** |
| **Vulnerable URL:** /upload.php?file= |
| **Vulnerable Parameter:** file |
| **CVSS Score:** 9.8 |
| **Severity:** Critical |
| **Vulnerability Description:** This vulnerability allows unauthenticated attackers to upload malicious files to the server, potentially leading to remote code execution. The application fails to properly validate the uploaded file type, content, and size. |
| **Vulnerability Impact:** Critical impact as attackers can achieve complete system compromise by uploading and executing malicious code. This can lead to unauthorized access to all system resources, data theft, and establishing persistence on the server. |
| **Recommendation:** Implement strict file type validation using content inspection rather than relying on extensions. Set file size limits and store uploaded files outside the web root directory. Implement proper authentication before allowing uploads. |
| **Proof of Concept / Steps to Reproduce:**  Step 1: Access /upload.php    Step 2: Upload malicious PHP file    Step 3: Access the uploaded file to get shell access |

|  |
| --- |
| **VUL-002: SQL Injection in Login** |
| **IP: 10.50.33.2** |
| **Vulnerable URL:** /login.php?username= |
| **Vulnerable Parameter:** username |
| **CVSS Score:** 7.5 |
| **Severity:** High |
| **Vulnerability Description:** The login page is vulnerable to SQL injection attacks, allowing attackers to bypass authentication and access sensitive data. The application directly includes user input in SQL queries without proper sanitization. |
| **Vulnerability Impact:** High impact as attackers can bypass authentication mechanisms to access unauthorized data, potentially compromising all user accounts and sensitive information stored in the database. |
| **Recommendation:** Use parameterized queries or prepared statements instead of dynamic SQL queries. Apply input validation and sanitization for all user inputs used in database operations. |
| **Proof of Concept / Steps to Reproduce:**  Step 1: Access login page    Step 2: Input username=' OR 1=1 --    Step 3: Observe authentication bypass |

|  |
| --- |
| **VUL-003: Sensitive Data Exposure** |
| **IP: 10.50.33.2** |
| **Vulnerable URL:** /api/users/data |
| **Vulnerable Parameter:** None |
| **CVSS Score:** 8.2 |
| **Severity:** High |
| **Vulnerability Description:** The application transmits sensitive user data without proper encryption, exposing it to potential interception. API responses contain plaintext sensitive information. |
| **Vulnerability Impact:** High impact as sensitive data exposure can lead to identity theft, financial fraud, and regulatory compliance violations. Exposed data may include personal information, credentials, and financial details. |
| **Recommendation:** Implement proper encryption for all sensitive data in transit and at rest. Use HTTPS for all connections and proper hashing for stored passwords. |
| **Proof of Concept / Steps to Reproduce:**  Step 1: Login to application    Step 2: Capture network traffic    Step 3: Observe plaintext transmission of credentials |

|  |
| --- |
| **VUL-004: Cross-Site Scripting (XSS)** |
| **IP: 10.50.33.2** |
| **Vulnerable URL:** /search.php?q= |
| **Vulnerable Parameter:** q |
| **CVSS Score:** 6.1 |
| **Severity:** Medium |
| **Vulnerability Description:** The search functionality is vulnerable to XSS attacks, allowing attackers to inject and execute malicious scripts. User input from the search query is rendered without proper encoding. |
| **Vulnerability Impact:** Medium impact as attackers can execute arbitrary JavaScript in victims' browsers, potentially stealing session cookies, hijacking user sessions, or performing actions on behalf of victims. |
| **Recommendation:** Implement context-sensitive output encoding for all user-supplied data. Use Content Security Policy (CSP) headers to restrict execution of injected scripts. |
| **Proof of Concept / Steps to Reproduce:**  Step 1: Go to search page    Step 2: Input <script>alert("XSS")</script>    Step 3: Observe script execution |

|  |
| --- |
| **VUL-005: Insecure Direct Object References** |
| **IP: 10.50.33.2** |
| **Vulnerable URL:** /profile.php?id= |
| **Vulnerable Parameter:** id |
| **CVSS Score:** 5.5 |
| **Severity:** Medium |
| **Vulnerability Description:** The application does not properly validate user access to resources, allowing unauthorized access to protected data. The system relies solely on user-supplied IDs without proper authorization checks. |
| **Vulnerability Impact:** Medium impact as attackers can access information belonging to other users, violating confidentiality and potentially leading to privacy breaches and unauthorized data access. |
| **Recommendation:** Implement proper access control checks for all user-accessible resources. Use indirect references that are mapped server-side to actual resource identifiers. |
| **Proof of Concept / Steps to Reproduce:**  Step 1: Login as low-privilege user    Step 2: Access /profile.php?id=1    Step 3: Observe admin data access |

**IP 10.50.33.21**

|  |  |  |
| --- | --- | --- |
| **Sr. No.** | **Vulnerability Name** | **Vulnerability Risk Type** |
| **1** | **Unauthenticated File Upload** | **Critical** |
| **2** | **SQL Injection in Login** | **High** |
| **3** | **Sensitive Data Exposure** | **High** |
| **4** | **Cross-Site Scripting (XSS)** | **Medium** |
| **5** | **Insecure Direct Object References** | **Medium** |

|  |
| --- |
| **VUL-001: Unauthenticated File Upload** |
| **IP: 10.50.33.21** |
| **Vulnerable URL:** /upload.php?file= |
| **Vulnerable Parameter:** file |
| **CVSS Score:** 9.8 |
| **Severity:** Critical |
| **Vulnerability Description:** This vulnerability allows unauthenticated attackers to upload malicious files to the server, potentially leading to remote code execution. The application fails to properly validate the uploaded file type, content, and size. |
| **Vulnerability Impact:** Critical impact as attackers can achieve complete system compromise by uploading and executing malicious code. This can lead to unauthorized access to all system resources, data theft, and establishing persistence on the server. |
| **Recommendation:** Implement strict file type validation using content inspection rather than relying on extensions. Set file size limits and store uploaded files outside the web root directory. Implement proper authentication before allowing uploads. |
| **Proof of Concept / Steps to Reproduce:**  Step 1: Access /upload.php    Step 2: Upload malicious PHP file    Step 3: Access the uploaded file to get shell access |

|  |
| --- |
| **VUL-002: SQL Injection in Login** |
| **IP: 10.50.33.21** |
| **Vulnerable URL:** /login.php?username= |
| **Vulnerable Parameter:** username |
| **CVSS Score:** 7.5 |
| **Severity:** High |
| **Vulnerability Description:** The login page is vulnerable to SQL injection attacks, allowing attackers to bypass authentication and access sensitive data. The application directly includes user input in SQL queries without proper sanitization. |
| **Vulnerability Impact:** High impact as attackers can bypass authentication mechanisms to access unauthorized data, potentially compromising all user accounts and sensitive information stored in the database. |
| **Recommendation:** Use parameterized queries or prepared statements instead of dynamic SQL queries. Apply input validation and sanitization for all user inputs used in database operations. |
| **Proof of Concept / Steps to Reproduce:**  Step 1: Access login page    Step 2: Input username=' OR 1=1 --    Step 3: Observe authentication bypass |

|  |
| --- |
| **VUL-003: Sensitive Data Exposure** |
| **IP: 10.50.33.21** |
| **Vulnerable URL:** /api/users/data |
| **Vulnerable Parameter:** None |
| **CVSS Score:** 8.2 |
| **Severity:** High |
| **Vulnerability Description:** The application transmits sensitive user data without proper encryption, exposing it to potential interception. API responses contain plaintext sensitive information. |
| **Vulnerability Impact:** High impact as sensitive data exposure can lead to identity theft, financial fraud, and regulatory compliance violations. Exposed data may include personal information, credentials, and financial details. |
| **Recommendation:** Implement proper encryption for all sensitive data in transit and at rest. Use HTTPS for all connections and proper hashing for stored passwords. |
| **Proof of Concept / Steps to Reproduce:**  Step 1: Login to application    Step 2: Capture network traffic    Step 3: Observe plaintext transmission of credentials |

|  |
| --- |
| **VUL-004: Cross-Site Scripting (XSS)** |
| **IP: 10.50.33.21** |
| **Vulnerable URL:** /search.php?q= |
| **Vulnerable Parameter:** q |
| **CVSS Score:** 6.1 |
| **Severity:** Medium |
| **Vulnerability Description:** The search functionality is vulnerable to XSS attacks, allowing attackers to inject and execute malicious scripts. User input from the search query is rendered without proper encoding. |
| **Vulnerability Impact:** Medium impact as attackers can execute arbitrary JavaScript in victims' browsers, potentially stealing session cookies, hijacking user sessions, or performing actions on behalf of victims. |
| **Recommendation:** Implement context-sensitive output encoding for all user-supplied data. Use Content Security Policy (CSP) headers to restrict execution of injected scripts. |
| **Proof of Concept / Steps to Reproduce:**  Step 1: Go to search page    Step 2: Input <script>alert("XSS")</script>    Step 3: Observe script execution |

|  |
| --- |
| **VUL-005: Insecure Direct Object References** |
| **IP: 10.50.33.21** |
| **Vulnerable URL:** /profile.php?id= |
| **Vulnerable Parameter:** id |
| **CVSS Score:** 5.5 |
| **Severity:** Medium |
| **Vulnerability Description:** The application does not properly validate user access to resources, allowing unauthorized access to protected data. The system relies solely on user-supplied IDs without proper authorization checks. |
| **Vulnerability Impact:** Medium impact as attackers can access information belonging to other users, violating confidentiality and potentially leading to privacy breaches and unauthorized data access. |
| **Recommendation:** Implement proper access control checks for all user-accessible resources. Use indirect references that are mapped server-side to actual resource identifiers. |
| **Proof of Concept / Steps to Reproduce:**  Step 1: Login as low-privilege user    Step 2: Access /profile.php?id=1    Step 3: Observe admin data access |

# **SUMMARY OF FINDINGS & CONCLUSION:**

Finally, it must be remembered that security is an ongoing process, and that this report will provide an idea of the current vulnerabilities we were able to detect. There is no guarantee that new vulnerabilities will not be found and exploited in the future.

The assessment was only possible because **joint support & coordination** from the information security team of organization for **sharing & coordinating** during the assessment period. It is advised to refer the **Technical Report** for understanding in-depth of vulnerabilities that were discovered by technical team of **CyberSmithSECURE Pvt. Ltd.**

The Security Researchers of the **CyberSmithSECURE** performed Vulnerability Testing. We jointly recommend that all suggested measures in this document be performed to ensure the overall security of the target device. The following targeted sectors were identified by the security researchers for the scope of this testing.

**We thank internal Information Security team for their support & cooperation during the time of assessment.**