­



***C-AD-WEB-Q1-2025***

­

**{{client\_name}}**

Web Application VAPT

**© 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.

Table of Content

**[Document Control](#_Toc131542981)** [3](#_Toc131542981)

**[RISK LEVEL & DESCRIPTION](#_Toc131542982)** [4](#_Toc131542982)

**[SCAN Type](#_Toc131542983)** [5](#_Toc131542983)

**[List of Tools.](#_Toc131542984)** [6](#_Toc131542984)

**[Overall Findings](#_Toc131542985)** [7](#_Toc131542985)

**[Example Corp Risk Assessment Analysis of the Entire Framework](#_Toc131542986)** [7](#_Toc131542986)

**[Test Case](#_Toc131542987)** [8](#_Toc131542987)

**[Vulnerabilities Found](#_Toc131542988)** [9](#_Toc131542988)

**[URL](#_Toc131542989)** [10](#_Toc131542989)

[1) https://examplecorp.com 10](#_Toc131542990)

**[SUMMARY OF FINDINGS & CONCLUSION:](#_Toc131542991)** [22](#_Toc131542991)

# **Document Control**

|  |  |
| --- | --- |
| **Document Name** | {{client\_name}} Web Application VAPT |
| **Document ID** | C-EC-WEB-Q1-2025 |
| **Security Classification** | External |

|  |  |
| --- | --- |
| **Authorization** | |
| Reviewed By | Authorized By |
| Mr. George | Mr. Ray |

|  |  |  |  |
| --- | --- | --- | --- |
| **Amendment Log** | | | |
| Version | Date | Reviewed By | Brief description of the change |
| **V1.0** | 13/12/2025 | Mr. George | Change in impact |
| **V1.1** | 16/12/2025 | Mr. George | Severity changes |
| **V1.2** | 25/12/2025 | Mr. George | Revalidation status |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

# **RISK LEVEL & DESCRIPTION**

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



# **SCAN Type**

A picture containing screenshot, diagram, circle, graphics

Description automatically generated



The application security assessment is done:

**{{assessment\_scope}}**

The security vulnerabilities were reported for the following quarter on:

**{{date\_range}}**

The Scope of the Assessment only focuses on Applications provided by Client Team.

# **List of Tools.**



# **Overall Findings**

# **{{client\_name}} Risk Assessment Analysis of the Entire Framework**

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Sr. No.** | **Hostname** | **Instant purpose** | **VAPT Status** | **Critical** | **High** | **Medium** | **Low** | **Informational** | **Total** |
| **1.** | **https://examplecorp.com** | Web App | Completed | **1** | **3** | **3** | **3** | **1** | **11** |
| **Overall Findings** | | | | **1** | **3** | **3** | **3** | **1** | **11** |

**Chart

Description automatically generated**

# 

# **Test Case**

|  |  |  |
| --- | --- | --- |
| **Sr. No.** | **Test Case** | **Status** |
| 1 | Business Logic Testing | **Completed** |
| 2 | Service Disruption Analysis | A picture containing drawing  Description automatically generated**Completed** |
| 3 | OWASP Top 10 | A picture containing drawing  Description automatically generated**Completed** |
| 4 | Framework Based Vulnerability | A picture containing drawing  Description automatically generated**Completed** |
| 5 | Database Pentest | A picture containing drawing  Description automatically generated**Completed** |

**A picture containing drawing

Description automatically generated**

# **Vulnerabilities Found**

# **URL**

# https://examplecorp.com

|  |  |  |
| --- | --- | --- |
| **Sl.no** | **Vulnerability Name** | **Vulnerability Risk Type** |
| 1. | **Unauthenticated file upload leads to Command Injection** | **Critical** |
| 2. | **Remote Code Execution** | **High** |
| 3. | **HTML INJECTION** | **Medium** |
| 4. | **Outdated SSL being Used** | **Low** |
| 5. | **Directory listing** | **Informational** |

|  |  |
| --- | --- |
| **EC-001** | Unauthenticated file upload leads to  Command Injection |
| **URL** | https://examplecorp.com/uploads |
| **Vulnerable Parameter :** /uploads | |
| **CVSS:** **9.1 - CVSS:3.1/AV:N/AC:L/PR:N/UI:N/S:U/C:H/I:H/A:N** | |
| **Severity**: Critical | |
| **Vulnerability Description:**  The remote application/IP address is vulnerable to unauthenticated file upload functionality which can be further leads to gain Command Injection vulnerability. And also one can be able to get the remote access of the particular application. | |
| **Vulnerability Impact:**  If an attacker can control a script that is executed in the victim's browser, then they can typically fully compromise that user. Amongst other things, the attacker can:   1. Perform any action within the application that the user can perform. 2. View any information that the user is able to view. 3. Modify any information that the user is able to modify. 4. Initiate interactions with other application users, including malicious attacks that will appear to originate from the initial victim user. | |
| **Recommendation:**  We recommend using the following steps The following steps should be taken:  Since the application is using the vulnerable version of NVR – Network Video Recorder, it is  recommended to upgrade to the patched version.  *[Unrestricted File Upload | OWASP Foundation](https://owasp.org/www-community/vulnerabilities/Unrestricted_File_Upload)*  *[How to Prevent File Upload Vulnerabilities (wordfence.com)](https://www.wordfence.com/learn/how-to-prevent-file-upload-vulnerabilities/)*  *[ClipBucket SQL Injection / Command Injection / File Upload ≈ Packet Storm (packetstormsecurity.com)](https://packetstormsecurity.com/files/146600/ClipBucket-SQL-Injection-Command-Injection-File-Upload.html)* | |
| **Proof of Concept:**  **Step 1:** The following URL of the application look like this.  **Step 2:** Upload function can only be access by an authenticated user.  **Step 3:** Now, getting the above request in the Burp Proxy, going to the repeater tab and changing the HTTP method from GET to POST and adding our simple PHP script in the upload option. Once hit send button, we can see that we are able to successfully add asdf.php file into the application.  **Step 4:** After uploading the file, now navigating to the asdf.php directory, passing the  argument into the CMD field and we got the response.  **A.**  **B.** | |

|  |  |  |
| --- | --- | --- |
| **EC-002** | Remote Code Execution | |
| **URL** | https://examplecorp.com/uploads | |
| **Vulnerable Parameters:** /vender | | |
| **CVSS:** **7.1 - CVSS:3.1/AV:L/AC:L/PR:L/UI:N/S:U/C:H/I:H/A:N** | | |
| **Rating**: Informational | | **Severity**: Critical |
| **Vulnerability Description:**  Arbitrary code execution. A program that is designed to exploit such a vulnerability is called an arbitrary code execution exploit. The ability to trigger arbitrary code execution over a network (especially via a wide-area network such as the Internet) is often referred to as remote code execution (RCE). | | |
| **Impact:**  Remote arbitrary code execution is most often aimed at giving a remote user administrative access on a vulnerable system. The attack is usually prefaced by an information gathering attack, in which the attacker uses some means such as an automated scanning tool to identify the vulnerable version of software. Once identified, the attacker executes the script against the program with hopes of gaining local administrative access on the host. | | |
| **Recommendation:**  Programming If your application calls out to the operating system, you need to be sure command strings are securely constructed, or else you risk having malicious instructions injected by an attacker. This section outlines a few approaches to protecting yourself. Modern programming languages have interfaces that permit you to read files, send emails, and perform other operation system functions. Use APIs wherever possible –only use shell commands where absolutely necessary. This will reduce the number of attack vectors in your application, and will also simplify your codebase.   * <https://www.rapid7.com/blog/post/2017/06/15/about-user-enumeration/> * <https://cwe.mitre.org/data/definitions/203.html> * <https://nvd.nist.gov/vuln/detail/CVE-2020-24008> | | |
| **Proof of Concept:**  **Step 1:** Filling all the necessary Details and submitting the details.  **Step 2:** Now uploading the file.  **Step 3:** Browsing the C99 shell and uploading it.  **Step 4:** Intercepting the request in Burp Proxy and analyzing the response. It tells that we have successfully uploaded the C99 shell.  **Step 5:** We have successfully uploaded our shell.  **Step 6:** Dirbuster reveals the /uploads directory.  **Step 7:** Navigating to /uploads directory reveals that our C99 shell is being uploaded. | | |

|  |  |  |
| --- | --- | --- |
| **EC-003** | **HTML INJECTION** | |
| **URL** | *https:/examplecorp.com/* | |
| **Vulnerable Parameter:** /imddir/ | | |
| **CVSS:** **5.8 - CVSS:3.1/AV:N/AC:L/PR:N/UI:N/S:U/C:N/I:N/A:L** | | |
| **Rating**: Medium | | **Severity**: Medium |
| **Vulnerability Description:**  The application is vulnerable to HTML Injection which can lead an attacker to inject | | |
| **Impact:**  If an attacker can control a script that is executed in the victim's browser, then they can typically fully compromise that user. | | |
| **Recommendation:**  The following steps should be taken:   1. Enable Browser XSS filter. 2. Validate HTML characters in the input fields.  * https://www.testorigen.com/html-injection-techniques-and-solution/ * https://blog.alertlogic.com/blog/client-side-injection-attacks/ * https://medium.com/@jamischarles/xss-aka-html-injection-attack-explained-538f46475f6c | | |
| **Proof of Concept:**  **Step 1:** Injecting HTML characters in input field.  **Step 2:** HTML Injection Found | | |

|  |  |  |
| --- | --- | --- |
| **EC-004** | **Outdated SSL being Used** | |
| **URL:** | https:examplecorp.com/uploads/ | |
| **Vulnerable Parameter: /** | | |
| **CVSS:** **3.5** | | |
| **Rating**: Low | | **Severity**: Low |
| **Vulnerability Description:**  *Vulnerabilities in SSL Certificate Expiry is a Medium risk vulnerability that is one of the most frequently found on networks around the world. This issue has been around since at least 1990 but has proven either difficult to detect, difficult to resolve or prone to being overlooked entirely.* | | |
| **Impact:**  Such certificates can have severe effects like phishing, financial loss, customer loss, etc. As we all know, an SSL certificate is the most significant tool that websites rely on to establish trust with their users. | | |
| **Recommendation:**  Generate a new certificate for the server, expired certificates pose a security threat as they prevent the user accessing your site from being able to properly evaluate the safety of your SSL certificates.   * <https://www.zdnet.com/article/google-reveals-major-flaw-in-outdated-but-widely-used-ssl->protocol/ * https://www.globalsign.com/en/blog/disable-tls-10-and-all-ssl-versions * https://github.com/ssllabs/research/wiki/SSL-and-TLS-Deployment-Best-Practices * https://www.venafi.com/education-center/ssl/common-ssl-attacks | | |
| **Proof of Concept:**  **Step 1:** nmap -sV --version-light -Pn --script ssl-poodle -p 443 117.000.00.00. | | |

|  |  |  |
| --- | --- | --- |
| **EC-005** | **Directory listing** | |
| **URL** | http://examplecorp.com/PHPWord-develop/ | |
| **Vulnerable Parameter:** PHPWord-develop/ | | |
| **CVSS:** **7.1 - CVSS:3.1/AV:L/AC:L/PR:L/UI:N/S:U/C:H/I:H/A:N** | | |
| **Rating**: Low | | **Severity**: Informational |
| **Vulnerability Description:**  *Web servers can be configured to automatically list the contents of directories that do not have an index page present. This can aid an attacker by enabling them to quickly identify the resources at a given path, and proceed directly to analysing and attacking those resources. It particularly increases the exposure of sensitive files within the directory that are not intended to be accessible to users, such as temporary files and crash dumps.* | | |
| **Impact:**  There is not usually any good reason to provide directory listings, and disabling them may place additional hurdles in the path of an attacker. This can normally be achieved in two ways:   * Configure your web server to prevent directory listings for all paths beneath the web root; * Place into each directory a default file (such as index.htm) that the web server will display instead of returning a directory listing. | | |
| **Recommendation:**  You should make sure the directory does not contain sensitive information or you may want to  restrict directory listings from the web server configuration.   * https://www.acunetix.com/blog/articles/directory-listing-information-disclosure/ * https://www.acunetix.com/vulnerabilities/web/directory-listing/ * https://portswigger.net/kb/issues/00600100\_directory-listing | | |
| **Proof of Concept:**  **Step1 :**Visit the vulnerable parameter | | |

# **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 assessmentwas 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.**