|  |
| --- |
|  |
|  |

Application Security Assessment Report

Of

Cottage Donor Management System (CDMS),

Tirumala Tirupati Devasthanams (TTD), Govt. of AP

Dated 06/11/2019

by

Andhra Pradesh Technology Services Ltd

3rd Floor, R&B Building, M.G. Road, Labbipet,

Vijayawada – 520010. Andhra Pradesh

Contents

[1. Executive Summary 3](#_Toc23947106)

[1.1. Introduction 3](#_Toc23947107)

[1.2. Engagement Specific Details 3](#_Toc23947108)

[1.3. Scope Details 3](#_Toc23947109)

[1.3.1. Inclusion 3](#_Toc23947110)

[1.3.2. Exclusion 3](#_Toc23947111)

[1.4. Approach & Methodology 3](#_Toc23947112)

[1.4.1. Information Gathering: 4](#_Toc23947113)

[1.4.2. Automated & Manual Scanning: 4](#_Toc23947114)

[1.4.3. Analyse results and reporting: 4](#_Toc23947115)

[2. Detailed Observation 7](#_Toc23947116)

[2.1. Web Application Security Assessment & Penetration Testing 7](#_Toc23947117)

[3. Scanned Items 30](#_Toc23947118)

[4. Limitations 31](#_Toc23947119)

1. Executive Summary

## Introduction

CDMS Application provides Timely, Transparent and Hassle free Accommodation Services. It is SMS and Email based communication to the Cottage Donors. There is end to end facilitation of Cottage Donors. Donors can also seek Operator Assistance for availing services as needed.

Andhra Pradesh Technology Services (hereon referred as APTS) performed the Application Security Assessment of CDMS Application for Tirumala Tirupati Devasthanams to determine, if any weakness exist in the application.

## Engagement Specific Details

|  |  |  |
| --- | --- | --- |
| 1. **S. No.** | **Activity** | 1. **Date** |
| 1. 1. | 1. Start date of engagement | 1. 31/10/2019 |
| 1. 2. | 1. Submission date of initial report | 1. 06/11/2019 |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 1. **S. No** | **Area** | **Review Performed By** | **Application SPOC** | **Department Name** |
| 1. 1. | 1. Application Security Assessment | 1. APTS Security Audit Team |  | 1. Tiumala Tirupati Devasthanams |

|  |  |  |  |
| --- | --- | --- | --- |
| 1. **S. No** | **Date** | **Version Number** | 1. **Remarks** |
| 1. 1. | 1. 06/11/2019 | 1. v1.0 | 1. Initial Review |

## Scope Details

### Inclusion

1. **Web Application Security Assessment & Penetration Testing**

Application Name: CDMS

Application URL: https://cdms-uat.ttdsevaonline.com

Environment: Staging Server

Version Number [or] Latest Compilation Timestamp: Not provided

Type of Review: Grey box

Hash of Zipped Source Code (SHA512): Not Provided

User Accounts Tested: D000001902(Role-Donor),D000004300(Role-SuperAdmin)

### Exclusion

1. Server Vulnerability Assessment
2. Secure Code Review
3. Process Review
4. Secure Network Architecture Review

## Approach & Methodology

1. The web application security assessment was conducted in line with the leading security standards and guidelines for web application security such as OWASP.
2. The approach followed for the security assessment is detailed below:

### Information Gathering:

We conducted a walkthrough of the web application to assess the scope of the security assessment and obtain the following information to identify the potential attack vectors:

* 1. Functionalities available in the web application
  2. Entry points for the web application
  3. Web application is custom developed or off-the-shelf application
  4. Protocols used by the web application
  5. Back-end technology including web server, framework, and development language
  6. Conduct search engine discovery and reconnaissance
  7. Banner grabbing (finger printing) to identify the running version of web server / application server and framework
  8. Enumerate application on web server to identify other applications running on the server
  9. View source of the web application to review the comments and metadata
  10. Map functionalities and data flow to identify attack vectors

### Automated & Manual Scanning:

We performed a Grey box automated & Manual scanning (with the knowledge of user credentials) of the web application URL using commercial and open source tools. The scanning was conducted to identify any known vulnerabilities in the subjected application.

### Analyse results and reporting:

We then analysed the results from manual inspection to identify the vulnerabilities applicable to the web application. The risk classification for each of these vulnerabilities was identified based on the likelihood of occurrence, impact, and level of access required to exploit these vulnerability as per the risk classification methodology detailed in 1.5 of the report.

1. An exception based detailed report is prepared with the following:
2. Description of the vulnerability
3. Risk Rating
4. Impact & Root Cause
5. Recommendation including reference links
   1. **Risk Categorization**

The risk ratings assigned to each finding in this report are based on 3 dimensions – Likelihood, Impact, and Level of access required. These are defined below.

|  |  |  |
| --- | --- | --- |
| **Likelihood** | High | Attacker can use existing tools to exploit the vulnerability by following prescriptive instructions and without knowledge of coding/platforms. Target can be exploited directly. Finding assists with exploitation of or is linked to other high or critical risk findings. |
| Medium | Attacker must have knowledge of coding/platforms and may require customisation of tools (e.g. batch scripts, shell scripts, Metasploit module customization) to exploit the vulnerability.  Exploitation of target may require setup of additional infrastructure or processes. |
| Low | High level of skill required to exploit. Attacker must develop their own tools or processes (e.g. custom written exploit code) to successfully exploit the vulnerability.  Publicly available exploits were not identified.  Exploitation of target requires setup of additional infrastructure or processes (e.g. Spear Phishing). |
| **Impact** | Severe | Vulnerability may lead to widespread administrator access to multiple materially sensitive systems (e.g. Enterprise Administrator), or access to the internal network from the Internet. |
| Major | Vulnerability may lead to immediate access to sensitive or materially sensitive data, or highly privileged access to critical business systems, or a severe and extended disruption to critical business systems or operations, with impact to many users or sites. |
| Moderate | Vulnerability may lead to access to sensitive data, or privileged access to critical business systems, or partial disruption to critical business systems or operations, with impact to some users or sites. |
| Minor | Vulnerability may lead to:  Access to non-sensitive data, or  Access to non-critical business systems, or  Disruption to non-critical business systems or operations, with limited impact to users/sites. |
| Insignificant | Information disclosure of non-sensitive enticement information (e.g. IP addresses, hostnames, system information) with no direct impact to availability. |
| **Level of access required** | Privileged | Privileged user (e.g. administrator). |
| Non-privileged | General user (e.g. domain user). |
| Internal Anonymous | Unauthenticated user with access to the internal network. |
| External Anonymous | Unauthenticated Internet user (includes web applications that allow self-registration). |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Consequence**  **Likelihood** | **Small** | **Moderate** | **Severe** | **Catastrophic** |
| **Low** | Info | Low | Medium | Medium |
| **Moderate** | Low | Medium | Medium | High |
| **High** | Low | Medium | High | High |
| **Very High** | Medium | High | High | High |

The final risk ratings are defined as follows:

|  |  |
| --- | --- |
| High | Urgent action should be taken to address findings. |
| Medium | Action should be taken to address findings in a timely manner.  Out of cycle change and compensating controls may be required. |
| Low | No immediate action required. Remediation items can be implemented during the next scheduled change window. |
| Information | No immediate risks to the environment were identified as part of the testing. Findings are informational only. |

Note: The above matrices are intended to be used as a guide only in determining the appropriate risk rating for a particular vulnerability. Other factors may need to be considered when weighing up the final risk rating, such as the number of servers/applications affected by the vulnerability, nature of system’s affected (e.g. Production, Development, and Test), and nature of data accessed or disclosed.

* 1. **Vulnerability Summary**

Below is the summary of open vulnerabilities that still exist in the application.

|  |  |  |  |
| --- | --- | --- | --- |
| **Review Area** | **Initial Review** | | |
| **High** | **Medium** | **Low** |
| **Web Application Security Assessment** | 4 | 3 | 9 |
| **Total** |  |  | **16** |

**Distribution of Observation**

1. Detailed Observation

## Web Application Security Assessment & Penetration Testing

|  |  |  |
| --- | --- | --- |
| 1. **Vulnerability Title** | **Broken Authentication** | **Risk Rating**: High |
| **Description** | User tries to access other user account with the help of username without knowing the password. This happens due to improper input validation at server side. | |
| **Affected URL(s)** | <https://cdms-uat.ttdsevaonline.com>  (Applicable to all internal URLs in the application) | |
| **Impact** | If user can guess the username, he/she can tamper the original response and login to the application. | |
| **Evidence/Proof of Concept**  **Step-1:**User trying to login with guessable username  idor_1.jpg  **Step-2:**Request is captured through proxy tool after clicking on login button  **idor_2.jpg**  **Step-3:**Response is intercepted  **idor_3.jpg**  **Step-4:**Response is tampered with success response  **idor_3_2.jpg**  **Step-5:**Successfully logged into the application  **idor_4.jpg** | | |
| **Recommendation** | It is recommended to implement random tokens either in cookie header or in body to verify user’s authenticity.  Reference Link:  <https://www.owasp.org/index.php/Top_10-2017_A2-Broken_Authentication> | |
| **Management Comments** |  | |

|  |  |  |
| --- | --- | --- |
| 1. **Vulnerability Title** | **Malicious File Upload** | **Risk Rating**: High |
| **Description** | The "unrestricted file upload" term is used in vulnerability databases and elsewhere, but it is insufficiently precise. The phrase could be interpreted as the lack of restrictions on the size or number of uploaded files, which is a resource consumption issue.  We noticed that ,in the given URL we are giving different file types including executable and bat files which are vulnerable | |
| **Affected Path(s)** | <https://cdms-uat.ttdsevaonline.com>  (Applicable to all upload functionalities) | |
| **Impact** | Unrestricted File Upload vulnerability is the ability to upload any type of file through an upload form into a website. This is a very high-security risk because a hacker can upload script/executable files ,which can then execute remotely and perform various actions, such as explore the file structure, retrieve sensitive data or even cause damage to the website . | |
| **Evidence/Proof of Concept**  **Step-1:** Login to the application and go to “Update Profile” Tab. Capture the request through proxy while submitting the request and change the Extension parameter value from image to html. In response the uploaded image path is displayed.  upload_1.png  **Step-2:** Alert message is reflected by accessing image uploaded path  upload_2.png | | |
| **Recommendation** | It is recommended to validate the file while uploading and reject the files with other formats (.exe,.bat. etc..,)  It is also recommended to validate multiple file extension for the same file  Reference Link:  https://www.datasprings.com/Help/DNN-Tutorials/artmid/535/articleid/65/secure-programming-tips-handling-file-uploads/AspxAutoDetectCookieSupport/1  https://www.owasp.org/index.php/Unrestricted\_File\_Upload  https://www.computerweekly.com/answer/File-upload-security-best-practices-Block-a-malicious-file-upload | |
| **Management Comments** |  | |

|  |  |  |
| --- | --- | --- |
| 1. **Vulnerability Title** | **No Session Management** | **Risk Rating**: High |
| **Description** | In this application there is no maintenance of session so the internal URLs are accessible even after logging out from the application. | |
| **Affected URL(s)** | https://cdms-uat.ttdsevaonline.com | |
| **Impact** | It helps the attacker access the internal URL directly and understands the schema of the application and some URLs are crawled through web page source and can be submitted. He/She can gain privileges by changing the role id. | |
| **Evidence/Proof of Concept**  force_1.png  Fig. Internal Page accessed without authentication | | |
| **Recommendation** | It is recommended to maintain proper session management.  It is recommended to maintain session after authentication and terminate the session after logout.  It is recommended not to reuse the session id.  It is recommended to maintain authorisation token in order to mitigate user access violation issues.  It is recommended to maintain proper session time out.  Reference Link.  <https://cheatsheetseries.owasp.org/cheatsheets/Session_Management_Cheat_Sheet.html>  https://cheatsheetseries.owasp.org/cheatsheets/Cross-Site\_Request\_Forgery\_Prevention\_Cheat\_Sheet.html | |
| **Management Comments** |  | |

|  |  |  |
| --- | --- | --- |
| 1. **Vulnerability Title** | **OTP Bypass** | **Risk Rating**: High |
| **Description** | In this application sensitive information like OTP is shown in response. | |
| **Affected URL(s)** | https://cdms-uat.ttdsevaonline.com | |
| **Impact** | Attacker can directly use the OTP to login to the application and make changes to it. | |
| **Evidence/Proof of Concept**  **Step-1:** For Forgot Password and Change Password page, the OTP i.e. generated in order to change password in disclosed with base64 encoded in the response.  **Screenshot (842).png** | | |
| **Recommendation** | It is recommended not to disclose OTP in response. | |
| **Management Comments** |  | |

|  |  |  |
| --- | --- | --- |
| 1. **Vulnerability Title** | **Password return in later Response** | **Risk Rating**: Medium |
| **Description** | Application returns passwords submitted to the application in clear form in later response. | |
| **Affected URL(s)** | https://cdms-uat.ttdsevaonline.com | |
| **Impact** | It increases the risk that users' passwords will be captured by an attacker. Many types of vulnerability, such as weaknesses in session handling, broken access controls, and cross-site scripting, could enable an attacker to leverage this behavior to retrieve the passwords of other application users. This possibility typically exacerbates the impact of those other vulnerabilities, and in some situations can enable an attacker to quickly compromise the entire application. | |
| **Evidence/Proof of Concept**  pswd_later_resp.png  Fig. Password return in later response | | |
| **Recommendation** | It is recommended not to return password in response data.  Reference Link:  <https://cwe.mitre.org/data/definitions/204.html> | |
| **Management Comments** |  | |

|  |  |  |
| --- | --- | --- |
| 1. **Vulnerability Title** | **Weak Password Policy** | **Risk Rating**: Medium |
| **Description** | Password policy includes password length of minimum 8 characters, password complexity where all types of characters (Uppercase, Lowercase and Special characters).Password is getting validated at the client side. | |
| **Affected URL(s)** | https://cdms-uat.ttdsevaonline.com | |
| **Impact** | An attacker can easily guess passwords facilitating unauthorized access to the website | |
| **Evidence/Proof of Concept**  **weak_pswd.png**  Fig. Weak password | | |
| **Recommendation** | It is recommended to validate new password and confirm password at server side.  Password should contain min of 8 characters length which includes 1 special char, 1 upper case letter, 1 lower case letter and one number.  Reference Link:  https://www.owasp.org/index.php/Testing\_for\_Weak\_password\_policy\_(OTG-AUTHN-007) | |
| **Management Comments** |  | |

|  |  |  |
| --- | --- | --- |
| 1. **Vulnerability Title** | **Insufficient Anti-Automation** | **Risk Rating**: Medium |
| **Description** | Captcha and Account Lockout policy is not implemented for login page | |
| **Affected URL(s)** | https://cdms-uat.ttdsevaonline.com  (Applicable to all forms present before login) | |
| **Impact** | Bots that try to automatically harvest email addresses or try to automatically sign up for or make use of Web sites. | |
| **Evidence/Proof of Concept**  INSUF-ANTI.png  Fig. Captcha not implemented | | |
| **Recommendation** | Captcha should be implemented for login page and user registration page.  Username and Password should be validated along with Captcha at server side.  It is recommended to implement account lockout policy  Reference Links:  <https://www.owasp.org/index.php/Testing_for_Captcha_(OWASP-AT-008)>  <https://www.owasp.org/index.php/Testing_for_Weak_lock_out_mechanism_(OTG-AUTHN-003)> | |
| **Management Comments** |  | |

|  |  |  |
| --- | --- | --- |
| 1. **Vulnerability Title** | **Clickjacking** | **Risk Rating**: Low |
| **Description** | Click jacking can be performed by framing the target site. An attack can trick the user into clicking on the link by framing the original page and showing a layer on top of it with dummy buttons. Upon executing the html file, we can see the entire website is been framed. | |
| **Affected URL(s)** | https://cdms-uat.ttdsevaonline.com | |
| **Impact** | Tricking the user to click on the link by framing the original page and showing a layer on top of it with dummy buttons which leads to Phishing. | |
| **Evidence/Proof of Concept**  click_code.JPG  Fig. Html Code  **click_igrs.png**  Fig. Clickjacking | | |
| **Recommendation** | The X-Frame-Options HTTP response header can be used to indicate whether or not a browser should be allowed to render a page in a <frame> or <iframe>. Sites can use this to avoid Click jacking attacks, by ensuring that their content is not embedded into other sites. Set the X-Frame-Options header for all responses containing HTML content. The possible values are "DENY", "SAMEORIGIN", or "ALLOW-FROM Uri"  Reference Link:  https://www.owasp.org/index.php/Testing\_for\_Clickjacking\_(OTG-CLIENT-009) | |
| **Management Comments** |  | |

|  |  |  |
| --- | --- | --- |
| 1. **Vulnerability Title** | **OTP Flooding** | **Risk Rating**: Low |
| **Description** | In this application there is no restriction for OTP generation for forgot password and change password functionality | |
| **Affected URL(s)** | https://cdms-uat.ttdsevaonline.com | |
| **Impact** | Attackers can rack up phone verification bill by requesting for OTPs with no intention of use. We term this as a resource exhaustion attack. They can rest in comfort that defenders will not shut out the phone number as that will deny the legitimate phone number access to the defender’s web service, culminating in ‘Denial-of-Service’, which is as costly. | |
| **Evidence/Proof of Concept**  **Step-1:** Attacker can generate OTP multiple times in order to identify the pattern of OTPs generated and also affecting the cost of SMS services.  **Screenshot (842).png** | | |
| **Recommendation** | It is recommended to send maximum 10 OTPs per day to the user’s mobile number to limit OTP Flooding. | |
| **Management Comments** |  | |

|  |  |  |
| --- | --- | --- |
| 1. **Vulnerability Title** | **Cross Origin Resource Sharing** | **Risk Rating**: Low |
| **Description** | Cross Origin Resource Sharing or CORS is a mechanism that enables a web browser to perform "cross-domain" requests using the XMLHttpRequest L2 API in a controlled manner. In the past, the XMLHttpRequest L1 API only allowed requests to be sent within the same origin as it was restricted by the same origin policy. | |
| **Affected URL(s)** | https://cdms-uat.ttdsevaonline.com | |
| **Impact** | XMLHttpRequest L2 (or XHR L2) introduces the possibility of creating a cross-domain request using the XHR API for backwards compatibility. This can introduce security vulnerabilities that in XHR L1 were not present. Interesting points of the code to exploit would be URLs that are passed to XMLHttpRequest without validation, especially if absolute URLS are allowed because that could lead to code injection. Likewise, other part of the application that can be exploited is if the response data is not escaped and we can control it by providing user-supplied input. | |
| **Evidence/Proof of Concept**  **cors.jpg**  Fig. CORS -Fisheries | | |
| **Recommendation** | It is recommended to allow only trusted origins  Reference Link:  https://www.owasp.org/index.php/Test\_Cross\_Origin\_Resource\_Sharing\_(OTG-CLIENT-007) | |
| **Management Comments** |  | |

|  |  |  |
| --- | --- | --- |
| 1. **Vulnerability Title** | **Security Headers not enabled** | **Risk Rating**: Low |
| **Description** | Web Browser XSS Protection is nor enabled, or is disabled by the configuration of X-XSS – Protection HTTP response header on the web server Content Security Policy (CSP) is an effective "defence in depth" technique to be used against content injection attacks. It is a declarative policy that informs the user agent what are valid sources to load from. | |
| 1. **Affected URL(s)** | https://cdms-uat.ttdsevaonline.com | |
| 1. **Impact** | There is no direct impact of not implementing CSP and XSS on your website. However, if your website is vulnerable to a Cross-site Scripting attack CSP can prevent successful exploitation of that vulnerability. | |
| **Evidence/Proof of Concept**  **sec_head.jpg**  Fig. Missing Security Headers | | |
| **Recommendation** | Add the X-XSS-Protection header with a value of "1; mode= block".  Enable CSP on your website by sending the Content-Security-Policy in HTTP response headers that instruct the browser to apply the policies you specified.  It is recommended to implement Strict Transport Security Header  Reference Links:  <https://www.keycdn.com/blog/http-security-headers> | |
| **Management Comments** |  | |

|  |  |  |
| --- | --- | --- |
| 1. **Vulnerability Title** | **Technology/Version Disclosure** | **Risk Rating**: Low |
| **Description** | The HTTP responses returned by this web application include a header named X-AspNet-Version. The value of this header is used by Visual Studio to determine which version of ASP.NET is in use. It is not necessary for production sites and should be disabled. | |
| **Affected URL(s)** | https://cdms-uat.ttdsevaonline.com | |
| **Impact** | This information might help an attacker gain more information and potentially focus on the development of further attacks for the target system. | |
| **Evidence/Proof of Concept**  tech_ver_new.jpg  Fig. Server and Application Version Disclosure | | |
| **Recommendation** | It is recommended to apply changes on your web.config file to prevent information leakage by applying custom error pages.  Remove unwanted HTTP Headers  Reference Link:  <https://www.owasp.org/index.php/Fingerprint_Web_Server_(OTG-INFO-002)>  <https://www.owasp.org/index.php/Fingerprint_Web_Application_Framework_(OTG-INFO-008)>  <https://www.owasp.org/index.php/Fingerprint_Web_Application_(OTG-INFO-009)> | |
| **Management Comments** |  | |

|  |  |  |
| --- | --- | --- |
| 1. **Vulnerability Title** | **Email Id Disclosure** | **Risk Rating**: Low |
| **Description** | During assessment, we found the email ids in web page and documents which is not a good practice suggested by the OWASP community. The mail ids displayed here are not any generic ones (ex: helpdesk@gmail.com) rather they belong to the respective individuals based on their designation.  The majority of spam comes from email addresses harvested off the internet. The spam-bots (also known as email harvesters and email extractors) are programs that scour the internet looking for email addresses on any website they come across. Spam bot programs look for strings like myname@mydomain.com and then record any addresses found. | |
| **Affected URL(s)** | https://cdms-uat.ttdsevaonline.com | |
| **Impact** | Disclosing the individual mail ids sometimes lead to social engineering attacks and often affected with the spam mails. However, email addresses of developers and other individuals (whether appearing on-screen or hidden within page source) may disclose information that is useful to an attacker; for example, they may represent usernames that can be used at the application's login. | |
| **Evidence/Proof of Concept**  email.png  Fig. Email Address disclosure | | |
| **Recommendation** | 1. Obfuscate email address or Spell out email addresses(Please enclose the email address with example [at] gmail [dot] com)   Refer:  https://stackoverflow.com/questions/748780/best-way-to-obfuscate-an-e-mail-address-on-a-website  https://stackoverflow.com/questions/11563283/why-write-at-and-dot-in-email-rather-than-and  https://academia.stackexchange.com/questions/55612/why-do-people-in-academia-tend-to-write-their-email-address-with-dot-at  https://stackoverflow.com/questions/483212/effective-method-to-hide-email-from-spam-bots | |
| **Management Comments** |  | |

|  |  |  |  |
| --- | --- | --- | --- |
| 1. **Vulnerability title** | | **Last Login Time not Implemented** | **Risk Rating**: Low |
| **Description** | | This will give the user a message when they logon about the last time they logged on. | |
| **Affected URL(s)** | | https://cdms-uat.ttdsevaonline.com | |
| **Impact** | | The logon attempts for service and scheduled job users are tracked in the same way when this feature is activated. The feature assumes interactive users are only used for interactive logon access. | |
| **Evidence/Proof of Concept**  **Step-1:** Login into any account and verify last login time not implemented to user in home screen  LASTLOGIN.png | | | |
| **Recommendation** | It is recommended to implement last login time on the home screen | | |
| **Management Comments** |  | | |

|  |  |  |
| --- | --- | --- |
| 1. **Vulnerability Title** | **Concurrent Login Enabled** | **Risk Rating**: Low |
| **Description** | It is the web application design decision to determine if multiple simultaneous logons from the same user are allowed from the same or from different client IP addresses. If the web application does not want to allow simultaneous session logons, it must take effective actions after each new authentication event, implicitly terminating the previously available session, or asking the user (through the old, new or both sessions) about the session that must remain active. | |
| **Affected URL(s)** | https://cdms-uat.ttdsevaonline.com | |
| **Impact** | An attacker after gaining credentials can login from a different location simultaneously and make some modifications which are not known to actual user. | |
| **Evidence/Proof of Concept**  **cncrnt.png**  Fig. Login from Mozilla browser and Chrome browser | | |
| **Recommendation** | It is recommended to disable concurrent logins.  Reference Link:  https://stackoverflow.com/questions/38098629/how-we-prevent-second-person-login-when-already-first-person-is-logged-in-same | |
| **Management Comments** |  | |

|  |  |  |
| --- | --- | --- |
| 1. **Vulnerability Title** | **Improper Input Validation** | **Risk Rating**: Low |
| **Description** | Application is accepting malicious scripts in the input field in submitting the forms. | |
| **Affected URL(s)** | <https://cdms-uat.ttdsevaonline.com>  (Applicable to all the forms containing input fields) | |
| **Impact** | Attacker is able to craft the input in a form that is not expected by the rest of the application. This will lead to parts of the system receiving unintended input, which may result in altered control flow, arbitrary control of a resource, or arbitrary code execution. | |
| **Evidence/Proof of Concept**  **input_valid.png**  Fig. Script injected in suggestions field of feedback form | | |
| **Recommendation** | It is recommended to validate the user input and don’t accept special characters through input fields such as ( > < )’ “ , if not required. | |
| **Management Comments** |  | |

## Scanned Items

/

/api

/api/Bookings

/api/Bookings/Booking

/api/Feedback

/api/Feedback/SaveFeedback

/api/Login

/api/Login/LoginDetails

/api/Login/SendOtp

/api/Login/SendOtp\_DonorCode

/api/Login/UpdatePassword

/api/Master

/api/Master/MasterDetails

/api/Redemption

/api/Redemption/GetSearchResults

/api/Search

/api/Search/GetUserData

/api/summary

/api/summary/GetDonationHistory

/api/summary/getdonordetailsbyuserid

/api/summary/GetWFRequestHistory

/api/UpdateDonorProfile

/api/UpdateDonorProfile/GetPrevDataFromDatabase

/api/UpdateDonorProfile/GetPrevDataFromDatabase?TransactionId=1901

/api/UpdateDonorProfile/SaveDonorData

/api/UpdateDonorUserProfile

/api/UpdateDonorUserProfile/DonorUserDetails

/api/UpdateDonorUserProfile/UpdateDonorUserDetails

/DonationImages

/DonationImages/354aa7b5-fc2c-4124-93a8-78884590297c.html

/DonationImages/64fe5942-979d-4c8f-8834-7d2fff2f3a31.jpg

## Limitations

1. The report has been prepared based on the information given by TTD and is accordingly, given for the specific purpose of internal use by the TTD. Our conclusions are based on the completeness and accuracy of the stated facts and assumptions; which if not entirely complete or accurate, should be communicated to us immediately, as the inaccuracy or incompleteness could have a material impact on our conclusions.
2. This report has been prepared Solely for TTD, being the express addressee to this document. TTD, does not accept or assume any liability, responsibility or duty of care for any use of or reliance on this report by anyone, other than (i) TTD, to the extent agreed in the relevant contract for the matter to which this report relates (if any), or (ii) as expressly agreed by TTD, Govt of Andhra Pradesh in writing in advance.
3. Without prior permission of TTD, the contents of this report may not be quoted in whole or in part or otherwise referred to in any documents. The report is for the sole information of TTD and APTS accepts no responsibility to any other party.
4. This report (and any extract from it) may not be copied, paraphrased, reproduced, or distributed in any manner or form, whether by photocopying, electronically, by internet, within another document or otherwise, without the prior written permission of TTD. Further, any quotation, citation or attribution of this report, or any extract from it, is strictly prohibited without TTD’s prior written permission.
5. This report makes recommendations based on the initial information. However, corrective action must be taken by the respective owners by performing a root cause analysis for each of the observations highlighted as part of this report.

|  |
| --- |
|  |