Cyber Security Assessment Report

of

APRDCAPP,

Andhra Pradesh Residential Educational Institutions Society,

Govt. of AP

04/02/2020

by

Andhra Pradesh Technology Services

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

Vijayawada – 520 010. Andhra Pradesh

Contents

[1. Executive Summary 3](#_Toc34222216)

[1.1. Introduction 3](#_Toc34222217)

[1.2. Engagement Specific Details 3](#_Toc34222218)

[1.3. Scope Details 3](#_Toc34222219)

[1.3.1. Inclusion 3](#_Toc34222220)

[1.3.2. Exclusion 3](#_Toc34222221)

[1.4. Approach & Methodology 4](#_Toc34222222)

[1.4.1. Information Gathering: 4](#_Toc34222223)

[1.4.2. Automated & Manual Scanning: 4](#_Toc34222224)

[1.4.3. Analyse results and reporting: 4](#_Toc34222225)

[1.5. Risk Categorization 4](#_Toc34222226)

[1.6. Vulnerability Summary 6](#_Toc34222227)

[1.6.1. Distribution of Observation 6](#_Toc34222228)

[2. Detailed Observation 7](#_Toc34222229)

[2.1. Web Application Security Assessment & Penetration Testing 7](#_Toc34222230)

[3. Scanned Items 21](#_Toc34222231)

[4. Limitations 21](#_Toc34222232)

1. Executive Summary

## Introduction

In order to do social and economic justice to the poor and talented rural children, Government of Andhra Pradesh established A.P. Residential Educational Institutions Society in the year 1972 as an autonomous body with the Secretary to Government, Education Department as Chairman.

Andhra Pradesh Technology Services (hereon referred as APTS) performed the Cyber Security Assessment of APRDC Application for APREIS Department 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. 03/03/2020 |
| 1. 2. | 1. Submission date of initial report | 1. 04/03/2020 |

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

|  |  |  |  |
| --- | --- | --- | --- |
| 1. **S. No** | **Date** | **Version Number** | 1. **Remarks** |
| 1. 1. | 1. 04/03/2020 | 1. v1.0 | 1. Initial Review |

## Scope Details

### Inclusion

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

Application Name: APRDCAPP

Application URL: https://test.apcfss.in/APRDCAPP2020/

Environment: UAT

Version Number [or] Latest Compilation Timestamp: Not Provided

Type of Review: Blackbox

Hash of Zipped Source Code (SHA512): Not Provided

### 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 an black-box automated & Manual scanning 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

## 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.

## 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** | 1 | 2 | 5 |
| **Total** |  | | **8** |

### Distribution of Observation

1. Detailed Observation

## Web Application Security Assessment & Penetration Testing

|  |  |  |
| --- | --- | --- |
| 1. **Vulnerability Name** | **Reflected XSS using Host Header Injection** | **Risk Rating**: High |
| **Description** | 1. This Web site is possibly vulnerable to Cross Site Scripting (XSS) attacks via the Host header. Cross site scripting (also referred to as XSS) is a vulnerability that allows an attacker to send malicious code (usually in the form of JavaScript) to another user. Because a browser cannot know if the script should be trusted or not, it will execute the script in the user context allowing the attacker to access any cookies or session tokens retained by the browser. | |
| **Affected Path(s)** | https://test.apcfss.in/APRDCAPP2020/aprdcapplication030220201153.aprjdc | |
| **Impact** | 1. Reflected cross-site scripting vulnerabilities arise when data is copied from a request and echoed into the application's immediate response in an unsafe way. An attacker can use the vulnerability to construct a request in such a way that the malicious JavaScript code will be executed in user’s browser which can access the cookies and local storage of user. | |
| **Evidence/Proof of Concept**  **Step 1:** Capture the request of URL https://test.apcfss.in/APRDCAPP2020/aprdcapplication030220201153.aprjdc using the proxy service as shown in image.  A screenshot of a social media post  Description automatically generated  **Step 2:** Modify the host parameter with the payload (<script>alert (1) </script>) and submit the request.  A screenshot of a social media post  Description automatically generated  **Step 3:** In the response from server the injected payload is executed as shown in image.  A screenshot of a social media post  Description automatically generated  A screenshot of a cell phone  Description automatically generated | | |
| **Recommendation** | 1. The best defence in this case is sanitization. Developers has to make sure that user input does not make it into server generated HTML headers unless it is necessary and in these situations, developers can implement filters that sanitize this content and eliminates any scripting tags, thus removing the possibility of exploiting potential entry points. | |
| **Management Comments** |  | |

|  |  |  |
| --- | --- | --- |
| 1. **Vulnerability Name** | **Improper captcha Validation** | **Risk Rating**: Medium |
| **Description** | The captcha will be implemented in order to prevent the automation but here the captcha is implemented but the same captcha can be reused multiple times which will leads to the brute force the request. | |
| **Affected Path(s)** | https://test.apcfss.in/APRDCAPP2020/aprdcapplication030220201153.aprjdc | |
| **Impact** | 1. It is a business requirement to have CAPTCHA on certain functionalities. If CAPTCHAs are insecure, then this can lead to extraction of sensitive data using tools, attack on authentication, DOS on websites. | |
| **Evidence/Proof of Concept**  **Step 1:** access the URL https://test.apcfss.in/APRDCAPP2020/aprdcapplication030220201153.aprjdc and capture the request after filling the details in the form  A screenshot of a social media post  Description automatically generated  **Step 2:** Same request can be submitted multiple time with different details but with same captcha as shown in image | | |
| **Recommendation** | Captcha should be validated server side and once validated captcha value should not be reusable and for every new request new value should be generated. | |
| **Management Comments** |  | |

|  |  |  |
| --- | --- | --- |
| 1. **Vulnerability Name** | **Host Header Injection** | **Risk Rating**: Medium |
| **Description** | The host header specifies which website or web application should process an incoming HTTP request. The web server uses the value of this header to dispatch the request to the specified website or web application. Each web application hosted on the same IP address is commonly referred to as a virtual host. So what constitutes a host header attack. | |
| **Affected Path(s)** | https://test.apcfss.in/APRDCAPP2020/aprdcapplication030220201153.aprjdc | |
| **Impact** | if the application accepts the host header, the risk lays there. There may occur:   * + URL Redirection issues   + Username and password credentials may get stolen   + Password Reset Poisoning   + Session Hijacking Attack   + Financial loss may occur   + Cross site scripting | |
| **Evidence/Proof of Concept**  **Step1:** First, go to the URL of https://test.apcfss.in/APRDCAPP2020/aprdcapplication030220201153.aprjdc And intercept the request of the URL using Proxy Tool.  A screenshot of a social media post  Description automatically generated  **Step 2: modify the host parameter and**  host given in request is reflected in response as shown in image.  A screenshot of a social media post  Description automatically generated  **Step 3:** Spider the host to get the 302-redirect page and intercept the request.  A screenshot of a computer  Description automatically generated  **Step 4:** Change the hostname from test.apcfss.in to demo.testfire.net in response the location is changed, and the site is redirected successfully to <https://demo.testfire.net>.  A screenshot of a cell phone  Description automatically generated  A screenshot of a social media post  Description automatically generated | | |
| **Recommendation** | 1. Remove the redirection function from the application and replace links to it with direct links to the relevant target URLs. Maintain a server-side list of all URLs that are permitted for redirection. Instead of passing the target URL as a parameter to the redirector, pass an index into this list. | |
| **Management Comments** |  | |

|  |  |  |
| --- | --- | --- |
| 1. **Vulnerability Name** | **Improper Error Handling** | **Risk Rating**: Low |
| **Description** | Improper handling of errors can introduce a variety of security problems for a web site. The most common problem is when detailed internal error messages such as stack traces, database dumps, and error codes are displayed to the user (hacker). These messages reveal implementation details that should never be revealed. Such details can provide hackers important clues on potential flaws in the site and such messages are also disturbing to normal users. | |
| **Affected Path(s)** | / (Web Server) | |
| **Impact** | An attacker makes use of the details found to perform the concentrated attacks based on the vulnerabilities present in the current version. | |
| **Evidence/Proof of Concept**    **Figure:** Error page disclosing the server version. | | |
| **Recommendation** | Ensure that the pages which is disclosing error or warning messages should properly configure to log errors to a file instead of displaying the error to the user. | |
| **Management Comments** |  | |

|  |  |  |
| --- | --- | --- |
| 1. **Vulnerability Name** | **Version Disclosure** | **Risk Rating**: Low |
| **Description** | The HTTP responses returned by this web application include a header named Server reveals apache version. This information is ignored by most people, with the exception of hackers, who use this information to launch targeted attacks against your web server and version. It is not necessary for production sites and should be disabled. | |
| **Affected Path(s)** | / (Web Server) | |
| **Impact** | The HTTP header may disclose sensitive information. This information can be used to launch further attacks. | |
| **Evidence/Proof of Concept**  **Step 1:** The version of server that is being used are disclosed as shown in image.  A screenshot of a cell phone  Description automatically generated | | |
| **Recommendation** | Remove these headers from the response messages by editing the web server configuration file and make sure that the technology /version number being used is not disclosed in any manner. | |
| **Management Comments** |  | |

|  |  |  |
| --- | --- | --- |
| 1. **Vulnerability Name** | **Vulnerable JQUERY Version** | **Risk Rating**: Low |
| **Description** | The application is using multiple vulnerable JavaScript libraries that have the known public exploits. Using these libraries may affect the application’s security. | |
| **Affected Path(s)** | https://test.apcfss.in/APRDCAPP2020/jQuery/jquery-1.10.1.js  https://test.apcfss.in/APRDCAPP2020/jQuery/jquery-ui.js | |
| **Impact** | The vulnerabilities caused by the used vulnerable libraries could help the attacker to perform cross site scripting attacks that result in client-side attacks that affect the end users. | |
| **Evidence/Proof of Concept**  **Step 1:** The Web sites is using the old versions of jQuery as shown in the below image.  A screenshot of a computer screen  Description automatically generated | | |
| **Recommendation** | Upgrade to the latest version of the jQuery libraries 3.0 or higher. | |
| **Management Comments** |  | |

|  |  |  |
| --- | --- | --- |
| 1. **Vulnerability Title** | **Port Misconfiguration** | **Risk Rating**: Low |
| **Description** | Website is running on both http and https protocols. | |
| **Affected Path(s)** | <http://test.apcfss.in/APRJCAPPL20/>  [https://test.apcfss.in/ APRDCAPP2020/](https://test.apcfss.in/APRJCAPPL20/) | |
| **Impact** | Attacker can sniff the data while website running on http protocol. | |
| **Evidence/Proof of Concept:**  **A screenshot of a computer  Description automatically generated**  **Figure:** Website is running on both http and https protocols. | | |
| **Recommendation** | It is recommended to redirect to https site. | |
| **Management Comment** |  | |

|  |  |  |
| --- | --- | --- |
| 1. **Vulnerability Name** | **Clickjacking** | **Risk Rating**: Low |
| **Description** | Clickjacking is a malicious technique of tricking a Web user into clicking on something different from what the user perceives they are clicking on, thus potentially revealing confidential information or taking control of their computer while clicking on seemingly innocuous web pages. | |
| **Affected Path(s)** | / (Web Server) | |
| **Impact** | An attacker can host this domain in other evil site by using iframe and if a user fills the given field it can directly redirect as logs to attacker and after its redirect to your web server. Leading to steal user information too and use that host site as phishing of your site its CSRF and Clickjacking. | |
| **Evidence/Proof of Concept**  **Step 1:** Embed the website URL in the iframe src as shown in below image.  A screenshot of a computer screen  Description automatically generated  **Step 2:** The website is loaded within the frame as shown in image.  A screenshot of a social media post  Description automatically generated | | |
| **Recommendation** | 1. Sites can use X-Frame-Options to avoid clickjacking attacks, by ensuring that their content is not embedded into other sites. It is recommended to perform the following: 2. Use the X-FRAME Options in response header set to DENY or Same Origin or ALLOW-FROM a specified URL? 3. X-Frame-Options: This header works with modern browsers and can be used to prevent framing of the page. | |
| **Management Comments** |  | |

## Scanned Items

1. /
2. /aprdcapplication030220201153.aprjdc
3. /javascript
4. /javascript/applications
5. /javascript/applications/aadharverification.js
6. /javascript/applications/aprdcapplValidation.js
7. /javascript/calender
8. /javascript/calender/datetimepicker.js
9. /javascript/commonJSfunction.js
10. /javascript/tooltip
11. /javascript/tooltip/simple-tooltip.js
12. /jQuery
13. /jQuery/jquery-1.10.1.js
14. /jQuery/jquery-ui.js
15. /jQuery/jquery.maskedinput-1.2.2.js

## Limitations

1. The report has been prepared based on the information given by AP Residential Educational Institutions Society and is accordingly, given for the specific purpose of internal use by the AP Residential Educational Institutions Society. 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 AP Residential Educational Institutions Society, being the express addressee to this document. AP Residential Educational Institutions Society 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) AP Residential Educational Institutions Society, to the extent agreed in the relevant contract for the matter to which this report relates (if any), or (ii) as expressly agreed by AP Residential Educational Institutions Society in writing in advance.
3. Without prior permission of AP Residential Educational Institutions Society, 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 AP Residential Educational Institutions Society 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 AP Residential Educational Institutions Society. Further, any quotation, citation or attribution of this report, or any extract from it, is strictly prohibited without AP Residential Educational Institutions Society’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.