Application Security Assessment Report

Of

Department of

APCFSS,

A.P Residential Educational Institutions Society (APREIS)

Dated 04/03/2020

By

Andhra Pradesh Technology Services

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

Vijayawada – 520 010. Andhra Pradesh

Contents

[1. Executive Summary 3](#_Toc34214593)

[1.1. Introduction 3](#_Toc34214594)

[1.2. Engagement Specific Details 3](#_Toc34214595)

[1.3. Scope Details 3](#_Toc34214596)

[1.3.1. Inclusion 3](#_Toc34214597)

[1.3.2. Exclusion 3](#_Toc34214598)

[1.4. Approach & Methodology 3](#_Toc34214599)

[1.4.1. Information Gathering 4](#_Toc34214600)

[1.4.2. Automated & Manual Scanning 4](#_Toc34214601)

[1.4.3. Analyse Results & Reporting 4](#_Toc34214602)

[1.5. Risk Categorization 4](#_Toc34214603)

[1.6. Vulnerability Summary 6](#_Toc34214604)

[1.6.1. Distribution of Observation 6](#_Toc34214605)

[2. Detailed Observation 7](#_Toc34214606)

[2.1. Web Application Security Assessment & Penetration Testing 7](#_Toc34214607)

[3. Appendix 29](#_Toc34214608)

[3.1. OWASP Checklist 29](#_Toc34214609)

[3.2. Network Reconnaissance 32](#_Toc34214610)

[4. Scanned items 33](#_Toc34214611)

[5. Limitations 34](#_Toc34214612)

1. Executive Summary

## Introduction

The Andhra Pradesh State Council of Higher Education, the first of its kind in the country, set up as per the recommendations of the National Education Policy 1986, is primarily a coordinating body between the University Grants Commission (UGC), the State Government and the Universities.

Andhra Pradesh Technology Services (hereon referred as APTS) performed the Application Security Assessment of APSCHE Application for Higher Education Department Govt of Andhra Pradesh to determine, if any weakness exist in the application.

## Engagement Specific Details

|  |  |  |
| --- | --- | --- |
| **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. 04/03/202 |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **S. No** | **Area** | **Review Performed By** | **Application SPOC** | **Department Name** |
| 1. 1. | 1. Application Security Assessment | APTS TEAM | Sri Rayudu | APCFSS |
| 1. 2. | 1. Server Vulnerability Assessment | APTS TEAM | Sri Rayudu | APCFSS |

|  |  |  |  |
| --- | --- | --- | --- |
| **S.No** | **Date** | **Version Number** | 1. **Remarks** |
| 1. 1. | 1. 04/03/2019 | V1.0 | 1. Application Security Assessment |

## Scope Details

### Inclusion

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

Application Name: A.P Residential Educational Institutions Society (APREIS)

Application URL: <https://test.apcfss.in/APRJCAPPL20/>

<https://test.apcfss.in/APRDCAPP2020/>

Environment: Staging Server

Version Number [or] Latest Compilation Timestamp: v1.0

Type of Review: Blackbox

Hash of Zipped Source Code (SHA512): Not Provided

### Exclusion

1. Secure Code Review
2. Process Review
3. Secure Network Architecture Review
4. Committee Member login

## Approach & Methodology

The web application security assessment was conducted in line with the leading security standards and guidelines for web application security such as OWASP. 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 authenticated/ 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 & 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** | 3 | 4 | 6 |
| **Server Vulnerability assessment** | 0 | 0 | 0 |
| **Total** | **3** | **4** | **6** |
|  |  | **13** |

### Distribution of Observation

1. Detailed Observation

## Web Application Security Assessment & Penetration Testing

|  |  |  |
| --- | --- | --- |
| 1. **Vulnerability Title** | **Blind SQL Injection** | **Risk Rating**: High |
| **Description** | When an attacker executes SQL Injection attacks sometimes the server responds with error messages from the database server complaining that the SQL Queries syntax is incorrect. Blind SQL injection is identical to normal SQL Injection except that when an attacker attempts to exploit an application rather than getting a useful error message they get a generic page specified by the developer instead. This makes exploiting a potential SQL Injection attack more difficult but not impossible. An attacker can still steal data by asking a series of True and False questions through SQL statements. | |
| **Affected Path(s)** | <https://test.apcfss.in/APRJCAPPL20/aprjcapplication030220201042.aprs>  https://test.apcfss.in/APRDCAPP2020/aprdcapplication030220201153.aprjdc  **Parameters**- journalno, dateofpayment, dob | |
| **Impact** | Attacker can Retrieve internal data and information in data base by using malicious payload in web application. Attacker may modify and view detail in data base. | |
| **Evidence/Proof of Concept**  **Step-1:** Access the **URL:** <https://test.apcfss.in/APRJCAPPL20/aprjcapplication030220201042.aprs> and submit data.  sql.png  **Step-2:** Capture the above request . In this request “journalno, dateofpayment, dob” parameters is vulnerable SQL injection.  sql2.jpg  **Step-3:** Fuzzing the journalno, dateofpayment, dob parameters for SQL Injection.  sql1.png  **Step-4:** We retrieved the details of a student.  sql3.jpg | | |
| **Recommendation** | Depending on the backend database, the database connection settings, and the operating system, an attacker can mount one or more of the following attacks successfully:   * Reading, updating and deleting arbitrary data or tables from the database * Executing commands on the underlying operating system   Reference link:  <https://www.owasp.org/index.php/Blind_SQL_Injection> | |
| **Management Comments** |  | |

|  |  |  |
| --- | --- | --- |
| 1. **Vulnerability Title** | **Cross Site Scripting(XSS)** | **Risk Rating**: High |
| **Description** | The application is affected with this vulnerability which occurs at the client side due to the improper input validation supplied by the user. This is a code injection that includes client-side code injection attack wherein an attacker can execute malicious scripts into a legitimate website or web application. | |
| **Affected Path(s)** | <https://test.apcfss.in/APRJCAPPL20/aprjcapplication030220201042.aprs>  https://test.apcfss.in/APRDCAPP2020/aprdcapplication030220201153.aprjdc | |
| **Impact** | An attacker can insert malicious JavaScript code that has access to cookies and local storage, which are often used to store session tokens. If an attacker can obtain a user's session cookie, they can then impersonate that user. | |
| **Evidence/Proof of Concept**  **Step 1:** Access the <https://test.apcfss.in/APRJCAPPL20/aprjcapplication030220201042.aprs> and capture the request.  and inject the xss payload in Host Header.  xss.jpg  xss1.jpg  **Figure:** javascript code is seen resulting in a pop up showing xss vulnerability. | | |
| **Recommendation** | * Every User input should be validated as strictly as possible on arrival, given the kind of content that it is expected to contain * User input should be HTML-encoded at any point where it is copied into application responses. All HTML meta characters, including < > " ' and =, should be replaced with the corresponding HTML entities (&lt; &gt; etc). * Refer: https://www.owasp.org/index.php/XSS\_(Cross\_Site\_Scripting)\_Prevention\_Cheat\_Sheet https://www.checkmarx.com/2017/10/09/3-ways-prevent-xss/ | |
| **Management Comments** |  | |

|  |  |  |
| --- | --- | --- |
| 1. **Vulnerability Title** | **Sensitive Data Disclosure** | **Risk Rating**: High |
| **Description** | Information disclosure is when an application fails to properly protect sensitive information from parties that are not supposed to have access to such information in normal circumstances. These type of issues are not exploitable in most cases, but are considered as web application security issues because they allows attackers to gather information which can be used later in the attack lifecycle, in order to achieve more than they could if they didn’t get access to such information. | |
| **Affected Path(s)** | <https://test.apcfss.in/APRJCAPPL20/aprjcapplication030220201042.aprs>  https://test.apcfss.in/APRDCAPP2020/aprdcapplication030220201153.aprjdc | |
| **Impact** | An attacker gains access to sensitive information (such as Andhra Card, passwords or social security numbers) which may be used to launch further exploits against the application, or which may be of direct value. | |
| **Evidence/Proof of Concept**  **Step-1:** Sensitive data stored in plain text after submission of the valid student details.  sensitive data disclosure.png  **Figure :** User’s sensitive data (unmasked Aadhar number) | | |
| **Recommendation** | High sensitive (Aadhaar, Account no and other PII) information to be masked at first 8 digits.  Reference link:  https://www.owasp.org/index.php/Top\_10-2017\_A3-Sensitive\_Data\_Exposure | |
| **Management Comments** |  | |

|  |  |  |
| --- | --- | --- |
| 1. **Vulnerability Title** | **Port Misconfiguration** | **Risk Rating**: Medium |
| **Description** | Website is running on both http and https protocols. | |
| **Affected Path(s)** | <http://test.apcfss.in/APRJCAPPL20/>  <https://test.apcfss.in/APRJCAPPL20/>  http://test.apcfss.in/APRDCAPP2020/aprdcapplication030220201153.aprjdc  https://test.apcfss.in/APRDCAPP2020/aprdcapplication030220201153.aprjdc | |
| **Impact** | Attacker can sniff the data while website running on http protocol. | |
| **Evidence/Proof of Concept:**  **port misconfiguration.png**  **Figure:** Website is running on both http and https protocols. | | |
| **Recommendation** | It is recommended to redirect to https site. | |
| **Management Comment** |  | |

|  |  |  |
| --- | --- | --- |
| 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/APRJCAPPL20/aprjcapplication030220201042.aprs>  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 Title** | **Host Header Attack** | **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. | |
| **Affected Path(s)** | 1. /(Web Server) | |
| **Impact** | Vulnerable to Cross Site Scripting attack and an attacker can manipulate the Host header as seen by the web application and cause the application to behave in unexpectedways. | |
| **Evidence/Proof of Concept**  **host header attack.jpg**  **Figure:** Given Host header request is reflected inside an Link tag | | |
| **Recommendation** | * The web application should use the SERVER\_NAME instead of the Host header.   Reference links-   * https://stackoverflow.com/questions/43941048/prevent-host-header-attack | |
| **Management Comments** |  | |

|  |  |  |
| --- | --- | --- |
| 1. **Vulnerability Title** | **Session id in URL** | **Risk Rating**: Medium |
| **Description** | The application submits user session id using the GET method. Sensitive information should not be passed via the URL. URLs could be logged or leaked via the referrer header. | |
| **Affected Path(s)** | https://test.apcfss.in/APRJCAPPL20/aprjcapplication030220201042.aprs  https://test.apcfss.in/APRDCAPP2020/aprdcapplication030220201153.aprjdc | |
| **Impact** | Possible sensitive information disclosure. | |
| **Evidence/Proof of Concept**  **Step-1**: Access the URL: <https://test.apcfss.in/APRJCAPPL20/aprjcapplication030220201042.aprs;jsessionid=F8441E846D02C5F03CA56B467F121A8D>  **session id in url.png** | | |
| **Recommendation** | It is recommended to always use POST method for submitting the sensitive data. | |
| **Management Comment** |  | |

|  |  |  |
| --- | --- | --- |
| 1. **Vulnerability title** | **Clickjacking** | **Risk Rating**: Low |
| **Description** | Click jacking, also known as a "UI redress attack", is when an attacker uses multiple  Transparent or opaque layers to trick a user into clicking on a button or link on another page when they were intending to click on the top level page. Thus, the attacker is "hijacking" clicks  Meant for their page and routing them to another page, most likely owned by another application, domain, or both. When the URL is encoded with the URL encoding then this attack is possible in this scenario. | |
| **Affected Path(s)** | <https://test.apcfss.in/APRJCAPPL20/aprjcapplication030220201042.aprs>  https://test.apcfss.in/APRDCAPP2020/aprdcapplication030220201153.aprjdc | |
| **Impact** | Clickjacking is when an attacker uses multiple transparent or opaque layers to trick a user into clicking on a button or link on a framed page when they were intending to click on the top level page. Thus, the attacker is "hijacking" clicks meant for their page and routing them to other another page, most likely owned by another application, domain, or both. Using a similar technique, keystrokes can also be hijacked. With a carefully crafted combination of style sheets, iframes, and text boxes, a user can be led to believe they are typing in the password to their email or bank account, but are instead typing into an invisible frame controlled by the attacker. | |
| **Evidence/Proof of Concept**  **clickjacking1.png**  **Figure:** HTML Code for click jacking  ***clickjacking.png***  **Figure:** POC of click jacking | | |
| **Recommendation** | Configure your web server to include an X-Frame-Options header. Consult Web references for more information about the possible values for this header. Reference:  https://developer.mozilla.org/en-US/docs/HTTP/X-Frame-Options  http://en.wikipedia.org/wiki/Clickjacking  https://www.owasp.org/index.php/Clickjacking | |
| **Management Comments** |  | |

|  |  |  |
| --- | --- | --- |
| 1. **Vulnerability Title** | **Technology/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. | |
| 1. **Affected Path(s)** | <https://test.apcfss.in/APRJCAPPL20/aprjcapplication030220201042.aprs>  https://test.apcfss.in/APRDCAPP2020/aprdcapplication030220201153.aprjdc | |
| 1. **Impact** | The HTTP header may disclose sensitive information. This information can be used to launch further attacks. | |
| **Evidence/Proof of Concept**  **Step1:** Below response header revealing the server banner and technology information.  version disclosure.JPG   1. **Figure :** POC for Apache version Detected | | |
| **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.  Reference Links:  https://www.saotn.org/remove-iis-server-version-http-response-header/ | |
| **Management Comments** |  | |

|  |  |  |
| --- | --- | --- |
| 1. **Vulnerability Title** | **Outdated jQuery version** | **Risk Rating**: Low |
| **Description** | jQuery is a fast, small, and feature-rich JavaScript library. It makes things like HTML document traversal and manipulation, event handling, animation, and Ajax much simpler with an easy-to-use API that works across a multitude of browsers. This application is using a vulnerable JavaScript version. One or more vulnerabilities were reported for this version of the JavaScript library | |
| **Affected Path(s)** | /(Web Server) | |
| **Impact** | Affected versions of this package are vulnerable to DOM based Cross-site Scripting (XSS) attack. | |
| **Evidence/Proof of Concept:**  **Step-1**: By Crawling the website we can able to see the java script library version.  **JQuery disclosure.png**  **Figure :** Outdated jquery version detected | | |
| **Recommendation** | Recommended to upgrade jQuery & Bootstrap to the latest version | |
| **Management Comment** |  | |

|  |  |  |
| --- | --- | --- |
| 1. **Vulnerability Title** | **Improper Error Handling** | **Risk Rating**: Low |
| **Description** | The application is not configured to display any generic error message rather it displays the application version details. Application error or warning messages may expose sensitive information about an application's internal workings to an attacker. | |
| **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**  **error msg.jpg**  **Figure :** Error page disclosing the server version details  **error msg1.jpg** | | |
| **Recommendation** | Verify that this page is disclosing error or warning messages and properly configure the application to log errors to a file instead of displaying the error to the user.  https://www.c-sharpcorner.com/blogs/handling-verbose-error-messageimproper-errorhandling-in-webconfig https://stackoverflow.com/questions/4363941/asp-net-customerrors-mode-off-error-when-tryingto-access-working-webpage | |
| **Management Comment** |  | |

|  |  |  |
| --- | --- | --- |
| 1. **Vulnerability Title** | **Content Security Policy not implemented** | **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. | |
| **Affected Path(s)** | /(Web Server) | |
| **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  CROSS.png  **Figure:** CSP header not implemented | | |
| **Recommendation** | * 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.   Reference Links:  https://developer.mozilla.org/en-US/docs/Web/HTTP/CSP | |
| **Management Comment** |  | |

|  |  |  |
| --- | --- | --- |
| 1. **Vulnerability Title** | **Strict transport security not enforced** | **Risk Rating**: Low |
| **Description** | HTTP Strict Transport Security (HSTS) is a web server directive that informs user agents and web browsers how to handle its connection through a response header sent at the very beginning and back to the browser. This sets the Strict-Transport-Security policy field parameter. It forces those connections over HTTPS encryption, disregarding any script's call to load any resource in that domain over HTTP. HSTS is but one arrow in a bundled sheaf of security settings for your web server or your web hosting service. | |
| **Affected Path(s)** | /(Web Server) | |
| **Impact** | An attacker who can perform XSS could insert malicious script such as: When the client loads and executes this script, it makes a request to the attacker-controlled web site. The attacker can then log the request and steal the cookie. | |
| **Evidence/Proof of Concept**  **CROSS.png**  **Figure:** No HSTS flag has been implemented | | |
| **Recommendation** | * Set the HSTS flag for this cookie.   Reference links-   * <http://www.valencynetworks.com/kb/session-cookie-found-without-secure-flag-set.html> * <https://www.owasp.org/index.php/SecureFlag> | |
| **Management Comments** |  | |

1. Appendix

## OWASP Checklist

The Application Security Assessment has been evaluated as per Open Web Application Security Project Testing guide v4.0 as follows:

| **Ref. No.** | **Category** | **Test Name** | **Safe?** | **Remarks** |
| --- | --- | --- | --- | --- |
| 1.1 | **Information Gathering** | | | |
| 1.1.1 | OTG-INFO-001 | Conduct Search Engine Discovery and Reconnaissance for Information Leakage | Yes | Tested, Not vulnerable |
| 1.1.2 | OTG-INFO-002 | Fingerprint Web Server | Yes | Tested, Not vulnerable |
| 1.1.3 | OTG-INFO-003 | Review Webserver Metafiles for Information Leakage | Yes | Tested, Not vulnerable |
| 1.1.4 | OTG-INFO-004 | Enumerate Applications on Webserver | Yes | Tested, Not vulnerable |
| 1.1.5 | OTG-INFO-005 | Review Webpage Comments and Metadata for Information Leakage | Yes | Tested, Not vulnerable |
| 1.1.6 | OTG-INFO-006 | Identify application entry points | NA | Not Applicable |
| 1.1.7 | OTG-INFO-007 | Map execution paths through application | NA | Not Applicable |
| 1.1.8 | OTG-INFO-008 | Fingerprint Web Application Framework | Yes | Tested, Not vulnerable |
| 1.1.9 | OTG-INFO-009 | Fingerprint Web Application | Yes | Tested, Not vulnerable |
| 1.1.10 | OTG-INFO-010 | Map Application Architecture | NA | Not Applicable |
| 1.2 | **Configuration and Deploy Management Testing** | | | |
| 1.2.1 | OTG-CONFIG-001 | Test Network/Infrastructure Configuration | NA | Not Applicable |
| 1.2.2 | OTG-CONFIG-002 | Test Application Platform Configuration | NA | Not Applicable |
| 1.2.3 | OTG-CONFIG-003 | Test File Extensions Handling for Sensitive Information | NA | Not Applicable |
| 1.2.4 | OTG-CONFIG-004 | Backup and Unreferenced Files for Sensitive Information | NA | Not Applicable |
| 1.2.5 | OTG-CONFIG-005 | Enumerate Infrastructure and Application Admin Interfaces | NA | Not Applicable |
| 1.2.6 | OTG-CONFIG-006 | Test HTTP Methods | Yes | Tested, Not vulnerable |
| 1.2.7 | OTG-CONFIG-007 | Test HTTP Strict Transport Security | Yes | Tested, Not vulnerable |
| 1.2.8 | OTG-CONFIG-008 | Test RIA cross domain policy | NA | Not Applicable |
| 1.3 | **Identity Management Testing** | | | |
| 1.3.1 | OTG-IDENT-001 | Test Role Definitions | NA | Not Applicable |
| 1.3.2 | OTG-IDENT-002 | Test User Registration Process | NA | Not Applicable |
| 1.3.3 | OTG-IDENT-003 | Test Account Provisioning Process | NA | Not Applicable |
| 1.3.4 | OTG-IDENT-004 | Testing for Account Enumeration and Guessable User Account | NA | Not Applicable |
| 1.3.5 | OTG-IDENT-005 | Testing for Weak or unenforced username policy | NA | Not Applicable |
| 1.3.6 | OTG-IDENT-006 | Test Permissions of Guest/Training Accounts | NA | Not Applicable |
| 1.3.7 | OTG-IDENT-007 | Test Account Suspension/Resumption Process | NA | Not Applicable |
| 1.4 | **Authentication Testing** | | | |
| 1.4.1 | OTG-AUTHN-001 | Testing for Credentials Transported over an Encrypted Channel | NA | Not Applicable |
| 1.4.2 | OTG-AUTHN-002 | Testing for default credentials | NA | Not Applicable |
| 1.4.3 | OTG-AUTHN-003 | Testing for Weak lock out mechanism | NA | Not Applicable |
| 1.4.4 | OTG-AUTHN-004 | Testing for bypassing authentication schema | NA | Not Applicable |
| 1.4.5 | OTG-AUTHN-005 | Test remember password functionality | NA | Not Applicable |
| 1.4.6 | OTG-AUTHN-006 | Testing for Browser cache weakness | NA | Not Applicable |
| 1.4.7 | OTG-AUTHN-007 | Testing for Weak password policy | NA | Not Applicable |
| 1.4.8 | OTG-AUTHN-008 | Testing for Weak security question/answer | NA | Not Applicable |
| 1.4.9 | OTG-AUTHN-009 | Testing for weak password change or reset functionalities | NA | Not Applicable |
| 1.4.10 | OTG-AUTHN-010 | Testing for Weaker authentication in alternative channel | NA | Not Applicable |
| 1.5 | **Authorization Testing** | | | |
| 1.5.1 | OTG-AUTHZ-001 | Testing Directory traversal/file include | NA | Not Applicable |
| 1.5.2 | OTG-AUTHZ-002 | Testing for bypassing authorization schema | NA | Not Applicable |
| 1.5.3 | OTG-AUTHZ-003 | Testing for Privilege Escalation | NA | Not Applicable |
| 1.5.4 | OTG-AUTHZ-004 | Testing for Insecure Direct Object References | NA | Not Applicable |
| 1.6 | **Session Management Testing** | | | |
| 1.6.1 | OTG-SESS-001 | Testing for Bypassing Session Management Schema | NA | Not Applicable |
| 1.6.2 | OTG-SESS-002 | Testing for Cookies attributes | NA | Not Applicable |
| 1.6.3 | OTG-SESS-003 | Testing for Session Fixation | NA | Not Applicable |
| 1.6.4 | OTG-SESS-004 | Testing for Exposed Session Variables | NA | Not Applicable |
| 1.6.5 | OTG-SESS-005 | Testing for Cross Site Request Forgery | NA | Not Applicable |
| 1.6.6 | OTG-SESS-006 | Testing for logout functionality | NA | Not Applicable |
| 1.6.7 | OTG-SESS-007 | Test Session Timeout | NA | Not Applicable |
| 1.6.8 | OTG-SESS-008 | Testing for Session puzzling | NA | Not Applicable |
| 1.7 | **Data Validation Testing** | | | |
| 1.7.1 | OTG-INPVAL-001 | Testing for Reflected Cross Site Scripting | Yes | Tested, Not vulnerable |
| 1.7.2 | OTG-INPVAL-002 | Testing for Stored Cross Site Scripting | Yes | Tested, Not vulnerable |
| 1.7.3 | OTG-INPVAL-003 | Testing for HTTP Verb Tampering | Yes | Tested, Not vulnerable |
| 1.7.4 | OTG-INPVAL-004 | Testing for HTTP Parameter pollution | Yes | Tested, Not vulnerable |
| 1.7.5 | OTG-INPVAL-005 | Testing for SQL Injection | No | Tested unsafe refer  vulnerability in 2.1 |
| 1.7.5.1 |  | Oracle Testing | NA | Not Applicable |
| 1.7.5.2 |  | MySQL Testing | NA | Not Applicable |
| 1.7.5.3 |  | SQL Server Testing | NA | Not Applicable |
| 1.7.5.4 |  | Testing PostgreSQL | NA | Not Applicable |
| 1.7.5.5 |  | MS Access Testing | NA | Not Applicable |
| 1.7.5.6 |  | Testing for NoSQL injection | NA | Not Applicable |
| 1.7.6 | OTG-INPVAL-006 | Testing for LDAP Injection | NA | Not Applicable |
| 1.7.7 | OTG-INPVAL-007 | Testing for ORM Injection | NA | Not Applicable |
| 1.7.8 | OTG-INPVAL-008 | Testing for XML Injection | NA | Not Applicable |
| 1.7.9 | OTG-INPVAL-009 | Testing for SSI Injection | NA | Not Applicable |
| 1.7.10 | OTG-INPVAL-010 | Testing for XPath Injection | NA | Not Applicable |
| 1.7.11 | OTG-INPVAL-011 | IMAP/SMTP Injection | NA | Not Applicable |
| 1.7.12 | OTG-INPVAL-012 | Testing for Code Injection | NA | Not Applicable |
| 1.7.12.1 |  | Testing for Local File Inclusion | NA | Not Applicable |
| 1.7.12.2 |  | Testing for Remote File Inclusion | NA | Not Applicable |
| 1.7.13 | OTG-INPVAL-013 | Testing for Command Injection | NA | Not Applicable |
| 1.7.14 | OTG-INPVAL-014 | Testing for Buffer overflow | NA | Not Applicable |
| 1.7.14.1 |  | Testing for Heap overflow | NA | Not Applicable |
| 1.7.14.2 |  | Testing for Stack overflow | NA | Not Applicable |
| 1.7.14.3 |  | Testing for Format string | NA | Not Applicable |
| 1.7.15 | OTG-INPVAL-015 | Testing for incubated vulnerabilities | No | Tested unsafe refer  vulnerability in 2.1 |
| 1.7.16 | OTG-INPVAL-016 | Testing for HTTP Splitting/Smuggling | Yes | Tested, Not vulnerable |
| 1.8 | **Error Handling** | | | |
| 1.8.1 | OTG-ERR-001 | Analysis of Error Codes | No | Tested unsafe refer  vulnerability in 2.1 |
| 1.8.2 | OTG-ERR-002 | Analysis of Stack Traces | No | Tested unsafe refer  vulnerability in 2.1 |
| 1.9 | **Cryptography** | | | |
| 1.9.1 | OTG-CRYPST-001 | Testing for Weak SSL/TSL Ciphers, Insufficient Transport Layer Protection | No | Tested, Not vulnerable |
| 1.9.2 | OTG-CRYPST-002 | Testing for Padding Oracle | NA | Not Applicable |
| 1.9.3 | OTG-CRYPST-003 | Testing for Sensitive information sent via unencrypted channels | NA | Not Applicable |
| 1.1 | **Business Logic Testing** | | | |
| 1.10.1 | OTG-BUSLOGIC-001 | Test Business Logic Data Validation | NA | Not Applicable |
| 1.10.2 | OTG-BUSLOGIC-002 | Test Ability to Forge Requests | NA | Not Applicable |
| 1.10.3 | OTG-BUSLOGIC-003 | Test Integrity Checks | NA | Not Applicable |
| 1.10.4 | OTG-BUSLOGIC-004 | Test for Process Timing | Yes | Tested, Not Applicable |
| 1.10.5 | OTG-BUSLOGIC-005 | Test Number of Times a Function Can be Used Limits | Yes | Tested, Not Applicable |
| 1.10.6 | OTG-BUSLOGIC-006 | Testing for the Circumvention of Work Flows | Yes | Tested, Not vulnerable |
| 1.10.7 | OTG-BUSLOGIC-007 | Test Defenses Against Application Mis-use | NA | Not Applicable |
| 1.10.8 | OTG-BUSLOGIC-008 | Test Upload of Unexpected File Types | NA | Not Applicable |
| 1.10.9 | OTG-BUSLOGIC-009 | Test Upload of Malicious Files | NA | Not Applicable |
| 1.11 | **Client Side Testing** | | | |
| 1.11.1 | OTG-CLIENT-001 | Testing for DOM based Cross Site Scripting | NA | Not Applicable |
| 1.11.2 | OTG-CLIENT-002 | Testing for JavaScript Execution | NA | Not Applicable |
| 1.11.3 | OTG-CLIENT-003 | Testing for HTML Injection | NA | Not Applicable |
| 1.11.4 | OTG-CLIENT-004 | Testing for Client Side URL Redirect | NA | Not Applicable |
| 1.11.5 | OTG-CLIENT-005 | Testing for CSS Injection | NA | Not Applicable |
| 1.11.6 | OTG-CLIENT-006 | Testing for Client Side Resource Manipulation | NA | Not Applicable |
| 1.11.7 | OTG-CLIENT-007 | Test Cross Origin Resource Sharing | NA | Not Applicable |
| 1.11.8 | OTG-CLIENT-008 | Testing for Cross Site Flashing | NA | Not Applicable |
| 1.11.9 | OTG-CLIENT-009 | Testing for Clickjacking | No | Tested unsafe refer  vulnerability in 2.1 |
| 1.11.10 | OTG-CLIENT-010 | Testing WebSockets | NA | Not Applicable |
| 1.11.11 | OTG-CLIENT-011 | Test Web Messaging | NA | Not Applicable |
| 1.11.12 | OTG-CLIENT-012 | Test Local Storage | NA | Not Applicable |

## Network Reconnaissance

PORT STATE SERVICE

1. open http

443 open https

8080 open ssh

1. Scanned items

https://test.apcfss.in/APRJCAPPL20

https://test.apcfss.in/APRJCAPPL20/

https://test.apcfss.in/APRJCAPPL20/aprjcapplication030220201042.aprs

https://test.apcfss.in/APRJCAPPL20/aprjcapplication030220201042.aprs;jsessionid=561344B249A60E8F82F0D44E17099512

https://test.apcfss.in/APRJCAPPL20/faces

https://test.apcfss.in/APRJCAPPL20/javascript

https://test.apcfss.in/APRJCAPPL20/javascript/aadharverification.js

https://test.apcfss.in/APRJCAPPL20/javascript/applications

https://test.apcfss.in/APRJCAPPL20/javascript/calender

https://test.apcfss.in/APRJCAPPL20/javascript/calender/datetimepicker.js

https://test.apcfss.in/APRJCAPPL20/javascript/commonJSfunctions.js

https://test.apcfss.in/APRJCAPPL20/javascript/tooltip

https://test.apcfss.in/APRJCAPPL20/javascript/tooltip/simple-tooltip.js

https://test.apcfss.in/APRJCAPPL20/jQuery/jquery-1.10.1.js

https://test.apcfss.in/APRJCAPPL20/jQuery/jquery-ui.js

https://test.apcfss.in/APRJCAPPL20/jQuery/jquery.maskedinput-1.2.2.js

https://test.apcfss.in/APRJCAPPL20/Notification

https://test.apcfss.in/APRJCAPPL20/Notification/Instructions.pdf

https://test.apcfss.in/APRJCAPPL20/themes

https://test.apcfss.in/APRDCAPP2020

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

https://test.apcfss.in/APRDCAPP2020/aprdcapplication030220201153.aprjdc

https://test.apcfss.in/APRDCAPP2020/aprdcapplication030220201153.aprjdc;jsessionid=A942217F0C21B3B45AD3AF598D384D52

https://test.apcfss.in/APRDCAPP2020/javascript/applications/aprdcapplValidation.js

https://test.apcfss.in/APRDCAPP2020/javascript/calender

https://test.apcfss.in/APRDCAPP2020/javascript/calender/datetimepicker.js

https://test.apcfss.in/APRDCAPP2020/javascript/commonJSfunction.js

https://test.apcfss.in/APRDCAPP2020/jQuery

https://test.apcfss.in/APRDCAPP2020/jQuery/jquery-1.10.1.js

https://test.apcfss.in/APRDCAPP2020/jQuery/jquery-ui.js

https://test.apcfss.in/APRDCAPP2020/jQuery/jquery.maskedinput-1.2.2.js

## 

1. Limitations
2. The report has been prepared based on the information given by APCFSS, Govt of Andhra Pradesh and is accordingly, given for the specific purpose of internal use by the APCFSS, Govt of Andhra Pradesh. 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.
3. This report has been prepared solely for APCFSS, Govt of Andhra Pradesh, being the express address to this document. APCFSS, Govt of Andhra Pradesh 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) APCFSS, Govt of Andhra Pradesh, to the extent agreed in the relevant contract for the matter to which this report relates (if any), or (ii) as expressly agreed by APCFSS, Govt of Andhra Pradesh in writing in advance.
4. Without prior permission of APCFSS, Govt of Andhra Pradesh, 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 APCFSS, Govt of Andhra Pradesh and APTS accepts no responsibility to any other party.
5. 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 APCFSS, Govt of Andhra Pradesh. Further, any quotation, citation or attribution of this report, or any extract from it, is strictly prohibited without APCFSS, Govt of Andhra Pradesh prior written permission.
6. 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.