Cyber Security Assessment Report

Of

ATTENDANCE,

General Administration Department (GAD),

Govt. of AP

03/09/2019

by

Andhra Pradesh Technology Services

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

Vijayawada – 520 010. Andhra Pradesh

Contents

[1. Executive Summary 3](#_Toc527051578)

[1.1. Introduction 3](#_Toc527051579)

[1.2. Engagement Specific Details 3](#_Toc527051580)

[1.3. Scope Details 4](#_Toc527051581)

[1.3.1. Inclusion 4](#_Toc527051582)

[1.3.2. Exclusion 4](#_Toc527051583)

[1.4. Approach & Methodology 5](#_Toc527051584)

[1.4.1. Information Gathering: 5](#_Toc527051585)

[1.4.2. Automated & Manual Scanning: 5](#_Toc527051586)

[1.4.3. Analyse results and reporting: 5](#_Toc527051587)

[1.5. Risk Categorization 6](#_Toc527051588)

[1.6. Vulnerability Summary 7](#_Toc527051589)

[1.6.1. Distribution of Observation 7](#_Toc527051590)

[2. Detailed Observation 8](#_Toc527051591)

[2.1. Web Application Security Assessment & Penetration Testing 8](#_Toc527051592)

[2.2. Vulnerability Summary 8](#_Toc527051593)

[3. Appendix 9](#_Toc527051594)

[3.1. OWASP Checklist 9](#_Toc527051595)

[3.2. Network Reconnaissance 12](#_Toc527051596)

[4. SSL Test 12](#_Toc527051597)

[5. Scanned Items 13](#_Toc527051598)

[6. Limitations 13](#_Toc527051599)

1. Executive Summary

## Introduction

This is the official attendance portal for General Administration Department of Government of Andhra Pradesh. The Office of the Nodal Authority, General Administration Department manages this site. Though all efforts have been made to ensure the accuracy and currency of the content on this website, the same should not be construed as a statement of law or used for any legal purposes.

Andhra Pradesh Technology Services (hereon referred as APTS) performed the Cyber Security Assessment of Attendance Application for APGAD 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. DD/MM/YYYY |
| 1. 2. | 1. Submission date of initial report | 1. DD/MM/YYYY |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 1. **S. No** | **Area** | **Review Performed By** | **Application SPOC** | **Department Name** |
| 1. 1. | 1. Application Security Assessment | APTS TEAM |  | 1. GAD |

|  |  |  |  |
| --- | --- | --- | --- |
| 1. **S. No** | **Date** | **Version Number** | 1. **Remarks** |
| 1. 1. | 1. 03/09/2019 | 1. v1.0 | 1. Application Security Assessment |

## Scope Details

### Inclusion

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

Application Name: attendance

Application URL: uat.attendance.ap.gov.in

Environment: UAT

Version Number [or] Latest Compilation Timestamp: NA

Type of Review: Greybox

Hash of Zipped Source Code (SHA512):Not Available

### 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 unauthenticated 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** | 3 | 2 | 3 |
| **Total** |  | | **08** |

### Distribution of Observation

1. Detailed Observation

## Web Application Security Assessment & Penetration Testing

|  |  |  |
| --- | --- | --- |
| 1. **Vulnerability Name** | Sql Injection | **Risk Rating**: High |
| **Description** | SQL injection (SQLi) refers to an injection attack wherein an attacker can execute malicious SQL statements that control a web application's database server. | |
| **Affected Path(s)** | http://www.uat.attendance.ap.gov.in/ | |
| **Impact** | An attacker can use SQL injection it to bypass a web application's authentication and authorization mechanisms and retrieve the contents of an entire database. SQLi can also be used to add, modify and delete records in a database, affecting data integrity. Under the right circumstances, SQLi can also be used by an attacker to execute OS commands, which may then be used to escalate an attack even further. | |
| **Evidence/Proof of Concept**  **Step 1:**access the above url and capture the request and inject the sql injection payloads which will generate the my sql error as shown in below image  sql error.JPG | | |
| **Recommendation** | Use parameterized queries when dealing with SQL queries that contain user input. Parameterized queries allows the database to understand which parts of the SQL query should be considered as user input, therefore solving SQL injection  Reference links:  https://dzone.com/articles/aspnet-preventing-sql-injectio  https://www.aspsnippets.com/Articles/SQL-Injection-Attack-its-examples-and-Prevention-mechanisms-and-Techniques-in-ASPNet.aspx  https://stackoverflow.com/questions/305044/how-can-i-avoid-sql-injection-attacks-in-my-asp-net-application | |
| **Management Comments** |  | |
| 1. **Vulnerability Name** | **Sensitive Data Disclosure** | **Risk Rating**: High |
| **Description** | The application displays the users’ sensitive information like Aadhar number in a clear format. It is insecure to display the information to the public without masking the part of the information. | |
| **Affected Path(s)** | http://www.uat.attendance.ap.gov.in/Employee\_edit\_Reports.aspx | |
| **Impact** | Possible sensitive information disclosure. | |
| **Evidence/Proof of Concept**  **Step 1:** By accessing the above URL it is observed that the sensitive information like Aadhar number, phone number is not masked as shown in below image.  **aadhaar.JPG** | | |
| **Recommendation** | Mask part of the data with any arbitrary characters from the complete disclosure. | |
| **Management Comments** |  | |

|  |  |  |
| --- | --- | --- |
| 1. **Vulnerability Name** | **Concurrent logins** | **Risk Rating**: Medium |
| **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 Path(s)** | http://www.uat.attendance.ap.gov.in/ | |
| **Impact** | In case the user accessed from the cyber cafe and forgot to logout from the system, an attacker who has access to the same system would be able to continue the session and conduct malicious activities. Later, if the user logged in using his laptop from his home, the session being accessed by the attacker will not be terminated which is a serious issue. | |
| **Evidence/Proof of Concept**  **Step 1:** In the application it is observed that the same user can login at different places as shown in below image.  **concurent.JPG** | | |
| **Recommendation** | It is recommended for web applications to add user capabilities that allow checking the details of active sessions at any time, monitor and alert the user about concurrent logons, provide user features to remotely terminate sessions manually, and track account activity history (logbook) by recording multiple client details such as IP address, User-Agent, login date and time, idle time, etc  **Reference Links:**  https://www.owasp.org/index.php/Session\_Management\_Cheat\_Sheet#Simultaneous\_  Session Logons  https://security.stackexchange.com/questions/34880/is-it-safe-to-allow-users-multiple-login-at-different-browsers-computers  https://stackoverflow.com/questions/17515716/only-one-concurrent-login-per-user-in-asp-net  http://geekswithblogs.net/Frez/archive/2010/05/17/preventing-a-user-from-having-multiple-concurrent-sessions.aspx | |
| **Management Comments** |  | |

|  |  |  |
| --- | --- | --- |
| 1. **Vulnerability Name** | **Insufficient Anti-Automation** | **Risk Rating**: Medium |
| **Description** | Insufficient Anti-automation is when a web site permits an attacker to automate a process that should only be performed manually. Certain web site functionalities should be protected against automated attacks. | |
| **Affected Path(s)** | http://www.uat.attendance.ap.gov.in | |
| **Impact** | Attackers could repeatedly exercise web site functionality attempting to exploit or defraud the system. An automated robot could potentially execute thousands of requests a minute, causing potential loss of performance or service. | |
| **Evidence/Proof of Concept**  **Step1**: in the login pages the captcha is not implemented as shown in below image.  Captcha.JPG | | |
| **Recommendation** | It is recommended to implement captcha and implemented captcha should not be bypassed or should not be readable by ocr scanners  Reference links:  http://www.captcha.net | |
| **Management Comments** |  | |

|  |  |  |
| --- | --- | --- |
| 1. **Vulnerability Name** | Stack trace enabled | **Risk Rating**: Medium |
| **Description** | The application responds with stack traces that are not managed which could reveal information useful to attackers. Providing debugging information as a result of operations that generate errors is considered a bad practice due to multiple reasons. For example, it may contain information on internal workings of the application such as relative paths of the point where the application is installed or how objects are referenced internally. | |
| **Affected Path(s)** | http://www.uat.attendance.ap.gov.in/\* | |
| **Impact** | An attacker can obtain information such as:  • ASP.NET version.  • Physical file path of temporary ASP.NET files.  • Information about the generated exception and possibly source code, SQL queries, etc.  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**  **Step 1:**the application is not configured to handle errors properly ,if we give any malicious inputs the it is generating starck trace errors as shown in below image  Stack Trace.JPG | | |
| **Recommendation** | Disable the trace in the web.config file.  Reference Links:  https://forums.asp.net/t/1729685.aspx?How+to+disable+Stack+Trace+in+Asp+net+  Web Application  https://dotnetstories.wordpress.com/2007/10/13/the-worst-5-mistakes-in-the-webconfig-file/ | |
| **Management Comments** |  | |

|  |  |  |
| --- | --- | --- |
| 1. **Vulnerability Name** | **click jacking** | **Risk Rating**: Low |
| **Description** | Click jacking 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)** | http://www.uat.attendance.ap.gov.in | |
| **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 Click jacking. | |
| **Evidence/Proof of Concept**  **Step 1**:the website can be loaded in iframe as shown in the below image  click.JPG | | |
| **Recommendation** | Sites can use X-Frame-Options to avoid click jacking attacks, by ensuring that their content is not embedded into other sites.  It is recommended to perform the following: ? Use the X-FRAME Options in response header set to DENY or Same Origin or ALLOW-FROM a specified URL ? X-Frame-Options: This header works with modern browsers and can be used to prevent framing of the page. | |
| **Management Comments** |  | |

|  |  |  |
| --- | --- | --- |
| 1. **Vulnerability Name** | **Vulnerable JavaScript library** | **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)** | http://www.uat.attendance.ap.gov.in | |
| **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 application is using the vulnerable jquery version as shown in below image  js.JPG | | |
| **Recommendation** | Upgrade to the latest version of the JQuery libraries 3.0 or higher. | |
| **Management Comments** |  | |

|  |  |  |
| --- | --- | --- |
| 1. **Vulnerability Name** | Internal file path disclosure | **Risk Rating**: Low |
| **Description** | The application software generates an error message that includes sensitive information about its environment. In this case, the server discloses the unnecessary information like the source file path. | |
| **Affected Path(s)** | http://www.uat.attendance.ap.gov.in/\* | |
| **Impact** | The information might help an attacker gain more information and potentially focus on the development of further attacks for the target system such as the file location on the server. | |
| **Evidence/Proof of Concept**  **Step 1**:By accessing some of pages in the web application it is observed that internal file path of server is disclosed as shown in below image  Path.JPG | | |
| **Recommendation** | It is recommend to display a custom error pages | |
| **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 | No | Tested unsafe refer  vulnerability in 2.1 |
| 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 | Yes | Tested, Not vulnerable |
| 1.1.7 | OTG-INFO-007 | Map execution paths through application | NA | Not Applicable |
| 1.1.8 | OTG-INFO-008 | Fingerprint Web Application Framework | No | Tested unsafe refer  vulnerability in 2.1 |
| 1.1.9 | OTG-INFO-009 | Fingerprint Web Application | No | Tested unsafe refer  vulnerability in 2.1 |
| 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 | No | Tested unsafe refer  vulnerability in 2.1 |
| 1.2.3 | OTG-CONFIG-003 | Test File Extensions Handling for Sensitive Information | Yes | No information found |
| 1.2.4 | OTG-CONFIG-004 | Backup and Unreferenced Files for Sensitive Information | Yes | Tested, Not vulnerable |
| 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 DefaultUser Account | Yes | Tested, Not vulnerable |
| 1.3.5 | OTG-IDENT-005 | Testing for Weak or unenforced username policy | No | Tested unsafe refer  vulnerability in 2.1 |
| 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 | No | Tested, Not vulnerable |
| 1.4.2 | OTG-AUTHN-002 | Testing for default credentials | No | Tested, Not vulnerable |
| 1.4.3 | OTG-AUTHN-003 | Testing for Weak lock out mechanism | No | Tested, Not vulnerable |
| 1.4.4 | OTG-AUTHN-004 | Testing for bypassing authentication schema | Yes | Tested, Not vulnerable |
| 1.4.5 | OTG-AUTHN-005 | Test remember password functionality | NA | Not Applicable |
| 1.4.6 | OTG-AUTHN-006 | Testing for Browser cache weakness | Yes | Tested, Not vulnerable |
| 1.4.7 | OTG-AUTHN-007 | Testing for Weak password policy | No | Tested unsafe refer  vulnerability in 2.1 |
| 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 | Yes | Tested, Not vulnerable |
| 1.5.2 | OTG-AUTHZ-002 | Testing for bypassing authorization schema | Yes | Tested, Not vulnerable |
| 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 | Yes | Tested, Not vulnerable |
| 1.6 | **Session Management Testing** | | | |
| 1.6.1 | OTG-SESS-001 | Testing for Bypassing Session Management Schema | Yes | Tested, Not vulnerable |
| 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 | No | Tested Not vunlerable |
| 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 | No | Tested not  vulnerability |
| 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 | Yes | Tested, vulnerable |
| 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 | Yes | Tested, Not vulnerable |
| 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 | Yes | Tested, Not vulnerable |
| 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 | Yes | Tested, Not vulnerable |
| 1.8.2 | OTG-ERR-002 | Analysis of Stack Traces | NA | Not Applicable |
| 1.9 | **Cryptography** | | | |
| 1.9.1 | OTG-CRYPST-001 | Testing for Weak SSL/TSL Ciphers, Insufficient Transport Layer Protection | NA | Not Applicable |
| 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 | No | Tested unsafe refer  vulnerability in 2.1 |
| 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 | NA | Not Applicable |
| 1.10.5 | OTG-BUSLOGIC-005 | Test Number of Times a Function Can be Used Limits | NA | Not Applicable |
| 1.10.6 | OTG-BUSLOGIC-006 | Testing for the Circumvention of Work Flows | NA | Not Applicable |
| 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 | Yes | Tested, Not vulnerable |
| 1.11.2 | OTG-CLIENT-002 | Testing for JavaScript Execution | No | Tested Vulnerable refer  vulnerability in 2.1 |
| 1.11.3 | OTG-CLIENT-003 | Testing for HTML Injection | Yes | Tested, Not vulnerable |
| 1.11.4 | OTG-CLIENT-004 | Testing for Client Side URL Redirect | Yes | Tested, Not vulnerable |
| 1.11.5 | OTG-CLIENT-005 | Testing for CSS Injection | Yes | Tested, Not vulnerable |
| 1.11.6 | OTG-CLIENT-006 | Testing for Client Side Resource Manipulation | Yes | Tested, Not vulnerable |
| 1.11.7 | OTG-CLIENT-007 | Test Cross Origin Resource Sharing | Yes | Tested, Not vulnerable |
| 1.11.8 | OTG-CLIENT-008 | Testing for Cross Site Flashing | NA | Not Applicable |
| 1.11.9 | OTG-CLIENT-009 | Testing for Clickjacking | Yes | Tested vulnerable |
| 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 | Yes | Tested, Not vulnerable |

## Scanned Items

## Limitations

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