Cyber Security Assessment Report

of

PPMIS,

Department Of Finance(PMU),

Govt. of AP

25/09/2019

by

Andhra Pradesh Technology Services

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

Vijayawada – 520 010. Andhra Pradesh

Contents

[1. Executive Summary 3](#_Toc20330329)

[1.1. Introduction 3](#_Toc20330330)

[1.2. Engagement Specific Details 3](#_Toc20330331)

[1.3. Scope Details 3](#_Toc20330332)

[1.3.1. Inclusion 3](#_Toc20330333)

[1.3.2. Exclusion 3](#_Toc20330334)

[1.4. Approach & Methodology 4](#_Toc20330335)

[1.4.1. Information Gathering: 4](#_Toc20330336)

[1.4.2. Automated & Manual Scanning: 4](#_Toc20330337)

[1.4.3. Analyse results and reporting: 4](#_Toc20330338)

[1.5. Risk Categorization 4](#_Toc20330339)

[1.6. Vulnerability Summary 6](#_Toc20330340)

[1.6.1. Distribution of Observation 6](#_Toc20330341)

[2. Detailed Observation 7](#_Toc20330342)

[2.1. Web Application Security Assessment & Penetration Testing 7](#_Toc20330343)

[3. Scanned Items 40](#_Toc20330344)

[4. Limitations 41](#_Toc20330345)

1. Executive Summary

## Introduction

Create a shelf of projects and recommend approval of suitable projects for implementation  on PPP route. Assist in preparing the pre-feasibility reports. Help in appointing / selecting consultants to develop the projects. Ensures transparent tendering processes. Ensures through legal or regulatory means, the Government minimizes financial obligations  through prudent use of guarantees. Develop internal evaluation guidelines in consultation with respective Departments to  evaluate and assess the projects. Nodal agency to conduct / recommend exposure visits and trainings.

Andhra Pradesh Technology Services (hereon referred as APTS) performed the Cyber Security Assessment of PPMIS Application for Finance Department(PMU) 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. 18/09/2019 |
| 1. 2. | 1. Submission date of initial report | 1. 25/09/2019 |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 1. **S. No** | **Area** | **Review Performed By** | **Application SPOC** | **Department Name** |
| 1. 1. | 1. Application Security Assessment | 1. APTS Security Team | 1. Sudha J | 1. Department Of Finance(PMU) |

|  |  |  |  |
| --- | --- | --- | --- |
| 1. **S. No** | **Date** | **Version Number** | 1. **Remarks** |
| 1. 1. | 1. 25/09/2019 | 1. v1.0 | 1. Initial Review |

## Scope Details

### Inclusion

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

Application Name: PPMIS

Application URL: http://testpppmis.cgg.gov.in

Environment: UAT

Version Number [or] Latest Compilation Timestamp: Not Applicable

Type of Review: Greybox

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 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 | 9 | 5 |
| **Total** |  | | **17** |

### Distribution of Observation

1. Detailed Observation

## Web Application Security Assessment & Penetration Testing

|  |  |  |
| --- | --- | --- |
| 1. **Vulnerability Name** | **Malicious File Upload** | **Risk Rating**: High |
| **Description** | The application fails to restrict the file types that the user uploads. The application accepted the files with the double extension when tried to upload. If the uploaded file contains any malicious content such as macros it may cause an adversary result in the server. | |
| **Affected Path(s)** | http://testpppmis.cgg.gov.in/TransactionForms/UploadDocuments.aspx | |
| **Impact** | An attacker can upload the malicious files that can be used as the backdoor for the later attacks in an attempt to compromise the whole server. | |
| **Evidence/Proof of Concept:**  **Step 1:** Access the Url http://testpppmis.cgg.gov.in/TransactionForms/UploadDocuments.aspx and uploaded a file with extension aspx even it’s showing to upload a valid files to upload but not validating the extension server side as shown in below image.  File Upload Shell.JPG  **Step 2**: By accessing file Url we can access the shell without authentication as show in below image.  File Upload Shell 2.JPG | | |
| **Recommendation** | 1. If there is no need to have Unicode characters, it is highly recommended to only accept alpha-numeric characters and only one dot as an input for the file name and the extension.  2. Don't rely only on client-side validation only.  3. Use the white list of extensions to verify the uploaded files  4. Use magic quotes for file upload validation  Reference Links:  https://stackoverflow.com/questions/2780191/how-to-restrict-file-type-in-fileupload-control  https://www.codeproject.com/Questions/886361/Prevent-double-file-Extension-file-upload-in-cshar | |
| **Management Comments** |  | |

|  |  |  |
| --- | --- | --- |
| 1. **Vulnerability Name** | **Cross Site Request Forgery(CSRF)** | **Risk Rating**: High |
| **Description** | Cross-Site Request Forgery (CSRF) is an attack that forces an end user to execute unwanted actions on a web application in which they're currently authenticated. CSRF attacks specifically target state-changing requests, not theft of data, since the attacker has no way to see the response to the forged request. With a little help of social engineering (such as sending a link via email or chat) | |
| **Affected Path(s)** | http://testpppmis.cgg.gov.in/TransactionForms/ProjectInception.aspx | |
| **Impact** | An attacker may trick the users of a web application into executing actions of the attacker's choosing. If the victim is a normal user, a successful CSRF attack can force the user to perform state changing requests like transferring funds, changing their email address, and so forth. If the victim is an administrative account, CSRF can compromise the entire web application. | |
| **Evidence/Proof of Concept**  **Step1**: select any project detail and click on modify as shown in image.  CSRF1.png  **Step 2:** Modify any fields and click on save as shown in image.  CSRF2.png  **Step 3**: Capture the Request and generate the CSRF poc as shown in below image.  CSRF3.png  **Step 4**: Modified the project name field from AAAAAAAAA to EXCUSEME in the csrf request as shown in below image.  CSRF4.png  **Step 5:** Now logged into new session and submitted the csrf request the data is updated successfully as shown in below image.  CSRF5.png | | |
| **Recommendation** | The following recommendation is applicable to all form fields in the application.  It is recommended to implement the Anti-CSRF token or page token for every request and validate both on client side and server side.  Reference link:  https://www.owasp.org/index.php/Cross-Site\_Request\_Forgery\_(CSRF) | |
| **Management Comments** |  | |

|  |  |  |
| --- | --- | --- |
|  | **Broken Access Control** | **Risk Rating**: High |
| **Description** | The application allows an unauthenticated user to access the pages that should be accessible to the administrator only. This happens due to the improper implementation of access controls set by the application. | |
| **Affected Path(s)** | http://testpppmis.cgg.gov.in/PPPMIS/ | |
| **Impact** | Attackers acting as users or administrators, or users using privileged functions have the ability of creating, accessing, updating or deleting every record. | |
| **Evidence/Proof of Concept**  **Step 1:** The web pages that are accessible to the authenticated users that can be directly accessed without any authentication as shown in image.  **Broken Access Control.png** | | |
| **Recommendation** | The default should always be denial. Everyone should be denied access to everything, and then every specific role can be explicitly granted access for each function needed. It is recommended to log failed attempts to access features to make sure everything is configured correctly. | |
| **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://testpppmis.cgg.gov.in/UserLogin.aspx | |
| **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**  **Step 1:** In the login page of the application the captcha is not implemented as shown below, the attacker can take advantage of this and can bruteforce the login credentials  **Captcha not implemented.png** | | |
| **Recommendation** | It is recommended to implement captcha.  Reference Links:  [http://www.captcha.net](http://www.captcha.net/) | |
| **Management Comments** |  | |

|  |  |  |
| --- | --- | --- |
| 1. **Vulnerability Name** | **Missing account lockout security feature** | **Risk Rating**: Medium |
| **Description** | A web application uses an account lockout policy to protect against clients attempting to log into accounts not belonging to them. | |
| **Affected Path(s)** | http://testpppmis.cgg.gov.in/UserLogin.aspx | |
| **Impact** | An attacker may attempt to discover a password by systematically trying every possible combination of letters, numbers and symbols until you discover the one correct combination that works. | |
| **Evidence/Proof of Concept**  **Step1:** Web application doesn’t have account lockout policy which leads to compromise user accounts by brute force  Account Lock out.png | | |
| **Recommendation** | It’s recommended that implement the account lockout policy after 3 unsuccessful attempts OR CAPTCHA should be implemented. | |
| **Management Comments** |  | |

|  |  |  |
| --- | --- | --- |
| 1. **Vulnerability Name** | **User Credentials are Sent in clear text** | **Risk Rating**: Medium |
| **Description** | User credentials are transmitted over an unencrypted channel. This information should always be transferred via an encrypted channel (HTTPS) to avoid being intercepted by malicious users. | |
| **Affected Path(s)** | http://testpppmis.cgg.gov.in/UserLogin.aspx | |
| **Impact** | A third party may be able to read the user credentials by intercepting an unencrypted HTTP connection | |
| **Evidence/Proof of Concept**  **Step1:**By capturing the request it is observed that the user credentials are sent in plain text as shown in image  Clear Text transmission of credentials.png | | |
| **Recommendation** | It is recommended to:  The application should use transport-level encryption (SSL or TLS) to protect all sensitive communications passing between the client and the server. Communications that should be protected include the login mechanism and related functionality, and any functions where sensitive data can be accessed or privileged actions can be performed. These areas of the application should employ their own session handling mechanism, and the session tokens used should never be transmitted over unencrypted communications.  Reference Link-  https://www.owasp.org/index.php/Testing\_for\_Credentials\_Transported\_over\_an\_Encrypted  \_Channel | |
| **Management Comments** |  | |

|  |  |  |
| --- | --- | --- |
| 1. **Vulnerability Name** | **Last Login Time Not Implemented** | **Risk Rating**: Medium |
| **Description** | This will give the user a message when they logon about the last time they logged on. | |
| **Affected Path(s)** | http://testpppmis.cgg.gov.in/UserLogin.aspx | |
| **Impact** | The logon attempts for service and scheduled job users are tracked in the same way when this feature is activated. The feature assumes interactive users are only used for interactive logon access. | |
| **Evidence/Proof of Concept**  Step1: In the application the last login time is not implemented as shown in image  Last Login time.png | | |
| **Recommendation** | It is recommended to implement logout button. | |
| **Management Comments** |  | |

|  |  |  |
| --- | --- | --- |
| 1. **Vulnerability title** | **Unencrypted View state Parameter** | **Risk Rating**: Medium |
| **Description** | The \_\_VIEWSTATE parameter is not encrypted. To reduce the chance of someone intercepting the information stored in the ViewState, it is good design to encrypt the ViewState. | |
| **Affected Url(s)** | /(Web Server) | |
| **Impact** | Attackers might be getting sensitive information through this parameter. | |
| **Evidence/Proof of Concept**  **Step-1:** Unencrypted viewstate.  **Viewstate unencrypted.png** | | |
| **Recommendation** | It is recommended to encrypt the view state parameter.  To do this, set the machineKey validation type to AES. This instructs ASP.NET to encrypt the ViewState value using the Advanced Encryption Standard. | |
| **Management Comments** |  | |

|  |  |  |
| --- | --- | --- |
| 1. **Vulnerability Name** | **Weak Password Policy** | **Risk Rating**: Medium |
| **Description** | The password policy implemented by the application is not as per the standard security practices recommended by the NIST. As per the NIST rules, the password must contain at least 10 characters wherein the application there is no policy to change the password. | |
| **Affected Path(s)** | http://testpppmis.cgg.gov.in/TransactionForms/ChangePassword.aspx | |
| **Impact** | Not implementing a proper password policy, could allow the users to set very easy password that help the attackers to perform the dictionary attacks and gain access to the user accounts. | |
| **Evidence/Proof of Concept**  **Step-1:** Changed the password to 123456  No Password Policy.png | | |
| **Recommendation** | It is recocommended to maintain strong usernames and passwords  Reference Links:  https://www.owasp.org/index.php/Testing\_for\_default\_credentials\_(OTG-AUTHN-002)  https://www.owasp.org/index.php/Testing\_for\_Weak\_password\_policy\_(OTG-AUTHN-007)  https://www.owasp.org/index.php/Testing\_for\_Weak\_or\_unenforced\_username\_policy\_(OTG-IDENT-005)  https://www.us-cert.gov/ncas/alerts/TA13-175A | |
| **Management Comments** |  | |

|  |  |  |
| --- | --- | --- |
| 1. **Vulnerability Name** | **Concurrent Logins Possible** | **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://125.17.121.166:8080/MSMETracker/Default.aspx | |
| **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**  Step1: The same user can login in two different browsers (locations) at same time as shown in image.  Concurrent logins.png | | |
| **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** | **Possible Session Fixation** | **Risk Rating**: Medium |
| **Description** | When authenticating a user, the application doesn't assign a new session ID, making it possible to use an existent session ID. The session id before login and after login remains unchanged which is a setback for the security of an application. | |
| **Affected Path(s)** | http://testpppmis.cgg.gov.in/UserLogin.aspx | |
| **Impact** | An attacker can craft a link making a user to authenticate himself with a known session ID, and then hijacking the user-validated session by the knowledge of the used session ID. | |
| **Evidence/Proof of Concept**  **Step 1:** the applicationsession id before login.  Before Login.png  **Step 2**: the application session id after login  **After Login.png**  **Step 3:**The supplication session id after Logout  After Logout.png | | |
| **Recommendation** | Web applications must ignore any session ID provided by the user's browser at login and must always generate a new session to which the user will log in if successfully authenticated.  Reference links:   1. https://www.hacksplaining.com/prevention/session-fixation   https://www.checkmarx.com/knowledge/knowledgebase/session-fixation | |
| **Management Comments** |  | |

|  |  |  |
| --- | --- | --- |
| 1. **Vulnerability Name** | **Security Headers** | **Risk Rating**: Medium |
| **Description** | It is observed that the application is missing CSP directives in the response header. Content Security Policy (CSP) is an important standard by the W3C that is aimed to prevent a broad range of content injection attacks such as cross-site scripting (XSS). 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)** | /(WebServer) | |
| **Impact** | There is no direct impact of not implementing CSP 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**  **Step 1:** the security headers are not implemented as shown in below image.  security headers.png | | |
| **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://www.owasp.org/index.php/Content\_Security\_Policy\_Cheat\_Sheet  https://stackoverflow.com/questions/37992225/config-your-iis-server-to-use-the-content-security-policy-header  https://content-security-policy.com  https://www.ryadel.com/en/iis-web-config-secure-http-response-headers-pass-securityheaders-io-scan/ | |
| **Management Comments** |  | |

|  |  |  |
| --- | --- | --- |
| 1. **Vulnerability Name** | **Technology/version Disclosure** | **Risk Rating**: Low |
| **Description** | The HTTP responses returned by this web application include headers named Server. 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. | |
| **Affected Path(s)** | /(Web Server) | |
| **Impact** | The disclosed information can be used by the attacker to launch further attacks. | |
| **Evidence/Proof of Concept**  **Step 1:** the response is displaying the server and technology version as shown in image  Technology & Version Disclosure.png | | |
| **Recommendation** | Apply the following changes to the web.config file to prevent ASP.NET version disclosure:  <System.Web><httpRuntime enableVersionHeader="false" /></System.Web> | |
| **Management Comments** |  | |

|  |  |  |
| --- | --- | --- |
| 1. **Vulnerability Name** | **Vulnerable JavaScript Library** | **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.  Consult Attack details and Web References for more information about the affected library and the vulnerabilities that were reported | |
| **Affected Path(s)** | http://testpppmis.cgg.gov.in/ppp-exports/jquery-1.4.2.js | |
| **Impact** | Lack of input validation causes a selector to be interpreted as html. The ability to exploit and impact depends on the component(s) and usage of the vulnerable libraries by the web application. | |
| **Evidence/Proof of Concept**  **Step 1:**the application is using the outdated jquery version as shown in image  Jquery older version.png | | |
| **Recommendation** | Upgrade to latest secure version. | |
| **Management Comments** |  | |

|  |  |  |
| --- | --- | --- |
| 1. **Vulnerability Name** | **Click Jacking** | **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 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)** | /(WebServer) | |
| **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 stylesheets, 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**  **Step 1:** Embed the website URL in the iframe src as shown in below image.  **Clickjacking code.png**  **Step 2:** The website is loaded within the frame as shown in image.  Clickjacking.png | | |
| **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 | |
| **Management Comments** |  | |

|  |  |  |
| --- | --- | --- |
| 1. **Vulnerability Name** | **Email Address Found** | **Risk Rating**: Low |
| **Description** | During assessment, we found that a lot of email ids in the contact us page which is not a good practice suggested by the OWASP community. The mail ids displayed here are not any generic ones (ex: helpdesk@gmail.com) rather they belong to the respective individuals based on their designation. The majority of spam comes from email addresses harvested off the internet. The spam-bots (also known as email harvesters and email extractors) are programs that scour the internet looking for email addresses on any website they come across. Spambot programs look for strings like myname@mydomain.com and then record any addresses found. | |
| **Affected Path(s)** | http://testpppmis.cgg.gov.in/TransactionForms/ContactUs.aspx | |
| **Impact** | Disclosing the individual mail ids sometimes lead to social engineering attacks and often affected with the spam mails. However, email addresses of developers and other individuals (whether appearing on-screen or hidden within page source) may disclose information that is useful to an attacker; for example, they may represent usernames that can be used at the application's login. | |
| **Evidence/Proof of Concept**  **Step 1**: E-mail address disclosure in the web application as shown in image.  Email Address Disclosure.png | | |
| **Recommendation** | 1. Obfuscate email address  2. Spell out email addresses | |
| **Management Comments** |  | |

|  |  |  |
| --- | --- | --- |
| 1. **Vulnerability Name** | **Password type input with auto-complete enabled** | **Risk Rating**: Low |
| **Description** | If the function is enabled, then credentials entered by the user are stored on their local computer and retrieved by the browser on future visits to the same application. The stored credentials can be captured by an attacker who gains control over the user's computer. | |
| **Affected Path(s)** | http://testpppmis.cgg.gov.in/UserLogin.aspx | |
| **Impact** | Possible sensitive information disclosure. | |
| **Evidence/Proof of Concept**  **Step 1**: in the application It is observed that the input fields are enabled with auto complete as shown in below image.  Autocomplete off not enabled.png | | |
| **Recommendation** | To prevent browsers from storing credentials entered into HTML forms, include the attribute **auto complete="off"** within the FORM tag (to protect all form fields) or within the relevant INPUT tags (to protect specific individual fields).  Reference Link: http://www.owasp.org/index.php/Testing\_for\_Vulnerable\_Remember\_Password\_(OTG-AUTHN-005) | |
| **Management Comments** |  | |

## Scanned Items

/

/AccessDenied.aspx

/JavaScript

/JavaScript/BiddingDetails.js

/JavaScript/BiddingDeveloper.js

/JavaScript/Validations.js

/JavaScript/Validations1.js

/Master

/Master/Welcome.aspx

/MasterForms

/MasterForms/AdministractiveDepartment.aspx

/MasterForms/DelayReason.aspx

/MasterForms/LoginDetails.aspx

/MasterForms/NewUserLogin.aspx

/MasterForms/ParticipateUptoLevel.aspx

/ppp-export

/ppp-export/jquery-1.4.2.js

/ppp-export/ui

/ppp-export/ui/jquery.ui.accordion.js

/ppp-export/ui/jquery.ui.core.js

/ppp-export/ui/jquery.ui.widget.js

/ScriptResource.axd

/TransactionForms

/TransactionForms/BidderConsultantDetails.aspx

/TransactionForms/BidderDeveloperDetails.aspx

/TransactionForms/BidderIndEngDetails.aspx

/TransactionForms/BiddingConsulDetails.aspx

/TransactionForms/BiddingDeveloperDetails.aspx

/TransactionForms/BiddingIndEngDetails.aspx

/TransactionForms/ChangePassword.aspx

/TransactionForms/ContactUs.aspx

/TransactionForms/DPRFeasibilityStudy.aspx

/TransactionForms/EstimatedPaymentsRevenues.aspx

/TransactionForms/FinancialClosure.aspx

/TransactionForms/FinancialForm.aspx

/TransactionForms/GovtSupport.aspx

/TransactionForms/LOI-ContractAward.aspx

/TransactionForms/PDFCentralDetails.aspx

/TransactionForms/PhysicalTargets.aspx

/TransactionForms/ProjectApprovals.aspx

/TransactionForms/ProjectAssistance.aspx

/TransactionForms/ProjectInception.aspx

/TransactionForms/ProjectLitigations.aspx

/TransactionForms/ProjectRisks.aspx

/TransactionForms/ProjectSubName.aspx

/TransactionForms/VGFCentralDetails.aspx

/TransactionForms/VGFStateDetails.aspx

/UnauthorizedAccess.aspx

/UserLogin.aspx

/WebResource.axd

## Limitations

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