**23. MIS Application is vulnerable to Cross-Site Request Forgery (CSRF)**

**Observation**

We observed that an application is vulnerable to CSRF attacks while executing sensitive operations like add, delete, modify, and password reset since it lacks a CSRF token.

Cross-site request forgery (also known as CSRF) is a web security vulnerability that allows an attacker to induce users to perform actions that they do not intend to perform. It allows an attacker to partly circumvent the same origin policy, which is designed to prevent different websites from interfering with each other.

**Recommendation:**

It is recommended to include a [CSRF token](https://portswigger.net/web-security/csrf/tokens) within relevant requests. The token should be

* Unpredictable with high entropy, as for session tokens in general.
* Tied to the user's session.
* Strictly validated in every case before the relevant action is executed.

**Status: Resolved**

CSRF token has been generated and passed through every request, and the requests with only valid token values are allowed to process the request.

**26. MIS Application is vulnerable to Broken Access Control**

**Observation**

The read only user has limited access to employee information than the superadmin. Changing the numeric value of search parameter will allow visiting and editing the employee’s details resulting into broken access control.

Broken access control prevails in an application when accesses are not properly restricted, allowing low-privileged users to do high-privileged user functions.

**Recommendation**

* Apart from resources to be shared, denying by default.
* Reusing same access control mechanisms that are used to restrict other functions of the application.

Logging access control failures, alerting admin when repeated failures

**Status: Resolved**

Restrictions have been set not allowing the unauthorized users to access the system.

**27. MIS Application is vulnerable to IDOR (Insecure Direct Object References)**

We observed IDOR vulnerability in the MIS (Management Information System) application which arises when an application uses user-supplied input to access objects directly. Visiting and editing memo can only be accessed by admin user but changing the numeric value of search parameter will allow an attacker visit and edit the disbursement memo resulting to IDOR.

**Recommendation**

* **Use per user or session indirect object references**. This prevents attackers from directly targeting unauthorized resources.
* **Check access.** Each use of a direct object reference from an untrusted source must include an access control check to ensure the user is authorized for the requested object.

**Status: Resolved**

Vulnerable access has been restricted by adding the access control to the editing feature of the disbursement memo.

**28. SSL/TLS not implemented in MIS (Management Information System)**

We observed that the SSL/TLS is not implemented in TDF Management Information System (MIS) which allow an attacker to see passwords in clear text and sensitive data sniffing the network traffic.

SSL stands for Secure Sockets Layer, and it's the standard technology for keeping an internet connection secure and securing any sensitive data passed between two systems, preventing cybercriminals from reading and altering any data transferred, including potentially personal information.

TLS (Transport Layer Security) is a commonly used protection technology for facilitating privacy and data security in Internet interactions. Encrypting communication between web applications and servers, such as web browsers loading a website, is a primary use case of TLS.

**Recommendation**

* Use SSL for all connections that are authenticated or transmitting sensitive or value data.
* Ensure that communications between infrastructure elements, such as between web servers and database systems, are appropriately protected via the use of transport layer security (TLS) or protocol level encryption for credentials and valuable data.

**29. MIS Application is using vulnerable version of jQuery**

* We observed that the TDF Management Information System application was using vulnerable version of jQuery 1.7.2.

**Recommendation**

For better Security, It is recommended to upgrade the JQuery to latest version where above issues are patched.

**30. MIS Application is vulnerable to SQL injection attack**

We observed SQL Injection vulnerability in the MIS application at the time of editing URL. The web application displays error messages from the database complaining that the SQL Query’s syntax is incorrect.

SQL injection is a web security vulnerability that allows an attacker to interfere with the queries that an application makes to its database. It generally allows an attacker to view data that they are not normally able to retrieve.

**Recommendation**

* Validate user-supplied input for expected data types, including input fields like drop-down menus or radio buttons, not just fields that allow users to type in input.
* Configure proper error reporting and handling on the web server and in the code so that database error messages are never sent to the client web browser. Attackers can leverage technical details in verbose error messages to adjust their queries for successful exploitation.

**31. MIS web application is vulnerable to Information Disclosure**

We found information disclosure files available publicly in above mentioned application.

Information disclosure, also known as information leakage, is when MIS (Management Information System) web application unintentionally reveals sensitive information to its users. Depending on the context, web application may leak all kinds of information to a potential attacker, including: error logs, web server Config files, full paths of the web server, environment variables.

Information disclosure vulnerabilities arise when development teams fails to delete internal information from the public contents.

**Recommendation**

Make sure that everyone involved in producing the website is fully aware of what information is considered sensitive. Sometimes seemingly harmless information can be much more useful to an attacker than people realize. Highlighting these dangers can help make sure that sensitive information is handled more securely in general by your organization.

* Audit any code for potential information disclosure as part of your QA or build processes. It should be relatively easy to automate some of the associated tasks, such as stripping developer comments.
* Use generic error messages as much as possible. Don't provide attackers with clues about application behavior unnecessarily.
* Double-check that any debugging or diagnostic features are disabled in the production environment.

Make sure you fully understand the configuration settings, and security implications, of any third-party technology that you implement. Take the time to investigate and disable any features and settings that you don't actually need.

**Status: Resolved**

**33. MIS web application is vulnerable to Reflected Cross-Site Scripting**

Because the MIS (Management Information System) application’s transaction start date and end date field are vulnerable to reflected Cross-Site Scripting user input is displayed without any Input/Output sanitization.

Reflected cross-site scripting (or XSS) arises when an application receives data in an HTTP request and includes that data within the immediate response in an unsafe way.

* **Filter input on arrival.** At the point where user input is received, filter as strictly as possible based on what is expected or valid input**.**
* **Encode data on output.** At the point where user-controllable data is output in HTTP responses, encode the output to prevent it from being interpreted as active content. Depending on the output context, this might require applying combinations of HTML, URL, JavaScript, and CSS encoding.
* **Use appropriate response headers.** To prevent XSS in HTTP responses that aren't intended to contain any HTML or JavaScript, you can use the Content-Type and X-Content-Type-Options headers to ensure that browsers interpret the responses in the way you intend.
* **Content Security Policy.** As a last line of defense, you can use Content Security Policy (CSP) to reduce the severity of any XSS vulnerabilities that still occur.

**Status: Resolved**

**34. Disclosure of Bash History**

We observed that the bash history of the MIS (Management Information System) application can be viewed easily. We found a bash history file in MIS application which contains command history of starting, viewing status of drop box and Unix basic commands.

The file that contains all commands that you have executed in the command line is known as bash history.

**Recommendation**

* Disable recording bash history.
* Delete bash history.

**Status: Resolved**