|  |
| --- |
| Governance of FMIS operations  Ver 1.0, 17th August 2022 |

Contents

[1 Version History 3](#_Toc111679927)

[2 Introduction 4](#_Toc111679928)

[3 Salient Features 5](#_Toc111679929)

[4 Application Architecture 6](#_Toc111679930)

[4.1 Security Architecture 6](#_Toc111679931)

[4.1.1 Application Security 7](#_Toc111679932)

[4.1.2 Database Security 8](#_Toc111679933)

[4.1.3 Network Security 9](#_Toc111679934)

[4.1.4 Other Security Aspects 10](#_Toc111679935)

[4.1.5 Audit Trial 11](#_Toc111679936)

[4.1.6 Security Audit 12](#_Toc111679937)

[5 Notes 13](#_Toc111679938)

# Version History

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| VER NO | CHANGE SUMMARY | DATE | PREPARED BY | REVIEWED BY | APPROVED BY |
| 1.0 | First Version | 17/08/2022 | Abhijith R V | Manoj V | Maya V T |

# Introduction

Government of Assam has deployed a comprehensive tool called **FinAssam** for managing the FMIS operations of the state with the support of World Bank under the Assam State Public Finance Institutional Reforms (ASPIRe) Program. **FinAssam** is a Single Source of Truth that enables prompt and efficient access to reliable financial data and helps strengthen government's financial controls, Improvise the provision of Govt. services, raising the budget process to higher levels of transparency and accountability and expediting government operations.

FinAssam comprises of over 40 modules covering all the essential aspects of a comprehensive Financial Management System. The key groupings of the modules are listed below.

|  |  |
| --- | --- |
| * Budget Planning & Preparation * Budget Allocation & Distribution * Administrative Approval / Technical Sanction/ Financial Sanction * Accounting & Reconciliation * RIDF Loan Processing * Debt Management * Employee & Payroll * Bill Creation * Expenditure Processing & Reporting * Receipt Management * Cash Planning & Management * Reports Generation * Asset Registry * Audit * Stock Management * Help Desk Management |  |

# Salient Features

* Centralized & Web Based Solution
* Built on Open Technologies
* Fully Integrated with over 40 Modules
* Customizable to meet State Specific Requirements
* Workflow Enabled
* Highly Intuitive
* Multilingual Support
* Unicode Compatible
* Security & GUI Audited
* Responsive GUI
* Easy to Administer and Manage
* External integrations
* Audit trail enabled
* DSC enabled
* Two Factor Authentication (Optional)
* No Third-Party Licenses Required
* Unlimited No. of Users

# Application Architecture

FinAssam is developed fully on opensource technologies. Below image depicts some of the major tools/ technologies used in FinAssam.

|  |
| --- |
|  |

FinAssam is a Java based web application and the core application is a monolith developed using Struts2+Spring Framework. Hibernate is the ORM tool used for interacting with the backend MariaDB database.

New modules in FinAssam are developed using latest Spring Boot and Spring Security Frameworks.

FinAssam application deployment consists of mainly 3 servers

* Web server – is SSL enabled and handles all the requests to the site and pass it to application server after initial validation through reverse proxy.
* Application Server – which contains the FinAssam application
* Database Server – where the database is hosted

The communication between these 3 servers is established through local IP

## Security Architecture

Security Architecture describes how the security controls are positioned and how they relate to the overall systems architecture. These controls serve the purpose to maintain the system’s quality attributes such as confidentiality, integrity and availability.

### Application Security

Application security is the process of developing, adding, and testing security features within applications to prevent security vulnerabilities against threats such as unauthorized access and modification. Application security is one of several levels of security used to protect systems.

Some of the application security features used are listed below

**Authentication**: Authentication procedures are built into the application to ensure that only authorized users gain access to the services. This is accomplished by prompting the user to provide a valid username and password for accessing secure contents. In addition to this CAPTCHA validation is also enables to prevent bots from automatically submitting the forms.

* **Authorization**: After a user has been successfully authenticated, the user may be authorized to access and use the application. The system can validate that a user has permission to access the contents by comparing the user’s identity with a list of pre-set roles and privileges.
* **Encryption**: Sensitive data in transit are encrypted using AES algorithm. AES is widely adopted and supported in both hardware and software. Additionally, AES has built-in flexibility of key length, which allows a degree of ‘future-proofing’ against progress in the ability to perform exhaustive key searches.
* **Hashing**: The user passwords are hashed and stored in Database using BCRYPT password hashing algorithm.
* **Session Management:** User sessions are properly validated, and a predefined timeout is also enabled. Idle user sessions are monitored and terminated automatically.

**File upload** facilities in the application can also be a threat to the security of the application.

To avoid file upload attacks, the following checks and best practices are used

* **Only allow specific file extensions** – whitelist of allowed file extensions are used to avoid executables, scripts and other potentially malicious content from being uploaded to the site.
* **Verify file types** – In addition to white listing, it is important to ensure that no files are masked as white listed file types. For instance, if an attacker were to rename an .exe to .docx, it would seem like a Word document but in fact is not. Therefore, it is important to verify file types before allowing them to be uploaded. Server-side validations are used for verifying the actual file types.
* **Set a maximum name length and maximum file size** – Maximum name length and file size are in place in order to prevent a Denial-of-Service attack.
* **Randomize uploaded file names** – uploaded file names are altered to application specific names so that attackers cannot try to access the file with the file name they have uploaded.
* **Store uploaded files outside web-root** - The directory to which files are uploaded are maintained outside of the application’s base directory so that the attackers cannot execute the file via a web URL.
* **Use of simple error messages** – When displaying file upload errors, we are not including directory paths, server configuration settings or other critical information that attackers could potentially misuse.

### Database Security

Database security is necessary for protecting and securing the database against intentional or accidental threats.

Key mechanisms followed to ensure database level security are listed below

* **Whitelisted IPs**: Access to the database server is restricted to only whitelisted IPs
* **SSH private key file**: Access to database is secured with SSH private key file (pem) to ensure more security.
* **Changing default port**: The default port 3306 for accessing the database is changed, to prevent various type of DDoS and Sniffing attacks on the database.
* **Restricting root access:** The root database access is restricted only to the database administrator to avoid misuse.
* **Privilege based access:** Critical operations such as DROP, TRUNCATE, DELETE etc. are allowed only to authorized users.
* **Disabling database shell history:** The database shell history file is disabled to avoid the risk of sensitive data being compromised, if an attacker manages to gain access to the history file.
* **Data at rest encryption:** Encrypting data in binary log and relay log files, to protect sensitive data contained in these files from being misused by attackers. MariaDB has built-in feature for this functionality known as ‘Transparent Data Encryption’.
* **Automatic data backups and replication**: Realtime data replication enabled to ensure data recovery and business continuity. Automatic backups are scheduled to prevent data loss.

### Network Security

Network security is the process of taking physical and software preventative measures to protect the underlying networking infrastructure from unauthorized access, misuse, malfunction, modification, destruction or improper disclosure, thereby creating a secure platform for computers, users and programs to perform their permitted functions within a secure environment.

* **Access control:** Direct SSH or SFTP access to the servers are only allowed from whitelisted Ips which are further validated using pem key files.
* **Behavioural analytics**: Data Centre is equipped to monitor the normal network behaviour of the applications so that they can spot any anomalies or breaches as and when they happen.
* **Firewalls**: Firewalls are the first line of defence in network security. Appropriate firewall rules are configured in the Data Centre. In this way the attacks including DoS/ DDoS are blocked before it even reaches to the web server
* **Establishing Secure Connection**: SSH (Secure Shell) Protocol is the best way to establish a protected connection. By default, SSH uses port 22. However, changing the port number is an easy way to reduce the chances of hackers attacking the server. Therefore, the best practice is to use port numbers between 1024 and 32,767.
* **SSH Keys Authentication**: Instead of using plain text passwords, the SSH server authentication is using a pair of SSH keys, a better alternative to traditional logins. The keys carry many more bits than a password and are uncrackable by most modern computers. The popular RSA 2048-bit encryption is equivalent to a 617-digit password.
* **Secure File Transfer Protocol**: To transfer files to and from the server without danger of hackers compromising or stealing data, it is vital to use Secure File Transfer Protocol (SFTP). This encrypts data files and the authentication information, which adds another layer of security.
* **Secure Sockets Layer Certificates**: Secure Socket Layer (SSL) Certificate is enabled for the domain as it will protect the information passed between two systems via internet. SSL can be used both in server-client and in server-server communication. SSL scrambles data so that sensitive information is not stolen in transit.
* **Private Networks and VPNs**: For some connections we are using VPNs such as OpenVPN, for server administration activities to ensure extra level of security and secure communication.

### Other Security Aspects

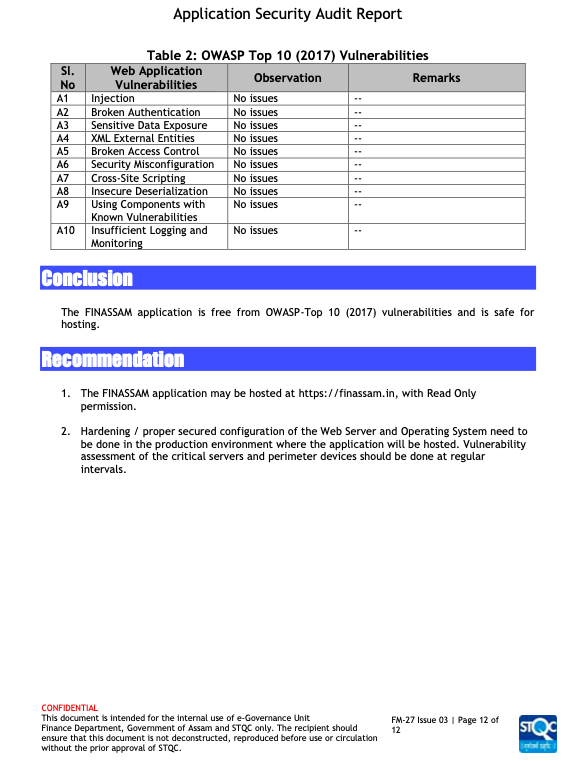
* System also supports Two factor authentication i.e., User ID + Password + Captcha validation and Mobile/ Email OTP verification
* The system uses mechanisms, based on industry standards, which assist the users in retrieving the forgotten credentials (e.g., passwords). Such practices include not disclosing the forgotten passwords online in the portal to the users, mailing the forgotten password to the e-mail address indicated by the user in his profile etc.
* System logs are written using RollingFileAppender mechanism where the roll over can be configured based on datetime or file size.
* Only authorized personnel have access to the system logs
* Only the application is having the write privileges in log files. All other access to the log files is configured as read-only
* System supports assigning privileges to the user on behalf of the spending units. The users will be able to perform transactions in the system only based on their assigned privileges.
* System follows a standard password creation guideline which will ensure the password will have minimum of 8 Characters with at least 1 Uppercase Alphabet, 1 Lowercase Alphabet, 1 Digit and 1 Special Character. The password will not be accepted if it is same as username or same as previous password. Every month system will prompt the user to reset their password to ensure security.
* All data which are available in the App Sever syncs every 2 hours to DR location. Live database replication is enabled between primary and DR server which will ensure the instant data availability in the Remote DR Database instance. Database backup is taken every 24 hours in the primary server and every 2 hours from the DR server.
* Zabbix tool is used by the CSP to identify and monitor system bottlenecks, service down events, network, security etc. The tool will send notifications through email which will helps to take necessary action immediately.
* Content Security Policy (CSP) headers are configured in the application which is an added layer of security that helps to detect and mitigate certain types of attacks, including Cross-Site Scripting (XSS) and data injection attacks.
* Annotations are used for Method validation to ensure only whitelisted HTTP methods are used to access the application methods.
* Proper timestamping is followed for records and its updation in the database.
* Use of Vault: Application secrets such as API keys, passwords, certificates, etc. are being moved to a Vault which provides a unified interface to any secret, while providing tight access control and recording a detailed audit log.

### Audit Trial

System captures detailed audit trail for the actions performed in the system and the same is accessible through the Audit Trail module. Authorized users can access this module and they will able to view / filter all the audit trail captured in the system which includes the before state and after state of the transaction. Based on the log analysis, respective users can mark their findings against each log item and they will also be able to generate reports using multiple parameters.

### Security Audit

The FinAssam application is Security Audited by STQC on 2019 and certified that the application is free from OWASP Top 10 vulnerabilities. Please see the screenshot of the report below.



The next security audit of the application is currently ongoing.

# Notes