MessageApp Vulnerability and Risk Table

Threat	**Weak Authentication & Brute Force Login**
Affected Component	User authentication & session management
Module Details	Authentication system (Involving request 120 - Member list, and the me.info file)
Vulnerability Class	Authentication bypass & user impersonation
Description	Users can retrieve UUIDs and usernames using request 120 (member list). By manually editing their local me.info file, they can re-enter the system impersonating another user. While private key passwords remain unaffected, an attacker can still send and receive messages as a different user. The protocol does not enforce verification of private keys stored on the server.
Result	Attackers can impersonate any user whose UUID/username has been retrieved. Unauthorized users can send and receive messages under false identities. This can lead to fraud, phishing.
Prerequisites	The attacker must retrieve a valid UUID and username using request 120. The attacker must modify their local me.info file. No authentication check is performed against stored private keys.
Business Impact	High risk of identity fraud: Users can be impersonated easily. Loss of trust in the messaging system due to unauthorized access.
Proposed Remediation	Implement UUID & private key verification before allowing login. Encrypt me.info or store authentication tokens securely. Restrict request 120 to authorized users only.
Risk	Damage Potential: 7/10 Reproducibility: 9/10 Exploitability: 6/10 Affected Users: 8/10 Discoverability: 9/10 Overall Score: 7.8/10

Threat **Database Overload - Unrestricted Large File Uploads**

Affected Component Message storage & database management

Module Details File storage within the messages table (BLOB field)

Vulnerability Class Resource Exhaustion (Denial of Service)

Description The protocol allows file attachments up to 4GB without any enforced size restrictions. Attackers can exploit this by sending excessively large files, consuming database storage and server disk space, potentially causing performance degradation or system crashes.

Result Database storage exhaustion, causing system slowdowns or failures.
Service disruption, preventing users from sending or receiving messages.

Prerequisites The attacker must be able to send file attachments.
No server-side validation exists to restrict file amounts / limit files per user.

Business Impact High risk of service downtime due to storage overload.
Operational costs increase due to excessive storage use.
Potential compliance violations if resource limits are exceeded.

Proposed Remediation Enforce strict file size limits (e.g., max 10MB per file, adjustable per user level).
Implement server-side validation to reject oversized files before storage.
Periodically remove old or unused large files to free up space.

Risk

Damage Potential:	8/10
Reproducibility:	9/10
Exploitability:	7/10
Affected Users:	8/10
Discoverability:	8/10
Overall Score:	7.8/10

Threat	**Database Overload - Unrestricted Large File Uploads**
Affected Component	Message storage & database management
Module Details	File storage within the messages table (BLOB field)
Vulnerability Class	Resource Exhaustion (Denial of Service)
Description	The protocol allows file attachments up to 4GB without any enforced size restrictions. Attackers can exploit this by sending excessively large files, consuming database storage and server disk space, potentially causing performance degradation or system crashes.
Result	Database storage exhaustion, causing system slowdowns or failures. Service disruption, preventing users from sending or receiving messages.
Prerequisites	The attacker must be able to send file attachments. No server-side validation exists to restrict file amounts / limit files per user.
Business Impact	High risk of service downtime due to storage overload. Operational costs increase due to excessive storage use. Potential compliance violations if resource limits are exceeded.
Proposed Remediation	Enforce strict file size limits (e.g., max 10MB per file, adjustable per user level). Implement server-side validation to reject oversized files before storage. Periodically remove old or unused large files to free up space.
Risk	Damage Potential: 8/10 Reproducibility: 9/10 Exploitability: 7/10 Affected Users: 8/10 Discoverability: 8/10 Overall Score: 7.8/10

Threat	**Distributed Denial of Service (DDoS) - No Rate Limiting**
Affected Component	Server request handling (selector-based server)
Module Details	Connection handling mechanism (server-side selector)
Vulnerability Class	Denial of Service (DoS)
Description	The server lacks a rate-limiting mechanism, allowing attackers to flood it with excessive requests. Since the server uses a selector-based architecture, it can be overwhelmed with multiple connections, exhausting resources and preventing legitimate users from accessing services.
Result	Attackers can send a high number of requests to consume server bandwidth and CPU. The server might become unresponsive, affecting all users. Service disruption for legitimate clients.
Prerequisites	The attacker must have access to a network capable of sending a large number of concurrent requests. The server must lack rate-limiting mechanisms.
Business Impact	High downtime risk, leading to loss of service availability. Reputational damage due to recurring outages. Potential financial loss if messaging services are mission-critical.
Proposed Remediation	Implement rate limiting per client (e.g., requests per second). Use CAPTCHA or challenge-response authentication to prevent bot-driven attacks. Deploy firewall rules or traffic filtering mechanisms to mitigate DDoS attempts. Monitor incoming request patterns (e.g., user sending repetitive requests from one specific type).
Risk	Damage Potential: 7/10 Reproducibility: 9/10 Exploitability: 7/10 Affected Users: 8/10 Discoverability: 7/10 Overall Score: 7.6/10

Threat **Lack of End-to-End Encryption**

Affected Component Message confidentiality

Module Details Server-managed encryption keys

Vulnerability Class Cryptographic Weakness

Description Messages are encrypted using symmetric encryption, but the server has full access to decryption keys, allowing it to decrypt and inspect user messages. This violates the principles of End-to-End Encryption (E2EE), where only the sender and recipient should have access to plaintext messages.

Result Server-side attackers or insiders can read user messages.
If the server is compromised, all stored messages can be decrypted in bulk.
Users may falsely assume their conversations are fully private, leading to trust issues.

Prerequisites The server must handle both encryption and decryption keys.
The attacker must have access to the server or database storing message keys.

Business Impact Loss of user trust due to lack of real privacy.
Potential lawsuits or reputation damage if data is leaked or misused.

Proposed Remediation Implement true E2EE by ensuring only clients generate and manage encryption keys.
Store only encrypted messages on the server, without access to decryption keys.
Educate users about the encryption model to set proper expectations.

Risk

Damage Potential:	9/10
Reproducibility:	9/10
Exploitability:	7/10
Affected Users:	9/10
Discoverability:	9/10
Overall Score:	8.6/10

Threat	**Man-in-the-Middle (MITM) - Weak Symmetric Key Protection**
Affected Component	Message encryption & key exchange
Module Details	RSA-based key exchange, AES-CBC encryption
Vulnerability Class	Cryptographic Weakness
Description	The protocol uses RSA encryption to exchange symmetric keys, but the RSA encryption is weak and can be exploited. Additionally, symmetric keys are reused and not refreshed between sessions, allowing attackers to analyze repeated encrypted exchanges to deduce the key. Furthermore, the AES-CBC encryption mode always uses an IV set to 0, making it predictable and vulnerable to cryptanalysis attacks.
Result	Attackers can intercept and decrypt message exchanges. Symmetric keys can be cracked due to repeated encryption patterns. Data confidentiality is compromised, leading to unauthorized access.
Prerequisites	The attacker must be able to intercept communications. The attacker can analyze predictable IVs to exploit AES-CBC vulnerabilities.
Business Impact	Severe data leakage due to weak encryption practices. Loss of user trust in the messaging system's security.
Proposed Remediation	Implement strong RSA encryption for key exchange. Refresh symmetric keys periodically to prevent key reuse attacks. Use AES-GCM instead of AES-CBC to eliminate IV predictability issues. Generate a secure random IV for each message encryption session.
Risk	Damage Potential: 9/10 Reproducibility: 8/10 Exploitability: 7/10 Affected Users: 9/10 Discoverability: 9/10 Overall Score: 8.4/10