# Vulnerability 1: Insufficient Client Authentication (Identity Spoofing)

# Vulnerabilities in the protocol

Threat: An attacker could impersonate a legitimate client by using a stolen UUID.

Affected component: Client-Server communication during registration and reconnection.

Module details: Registration, Reconnection.

Vulnerability class: Authentication Bypass.

Description: The protocol does not verify the authenticity of a client's identity beyond the UUID and client name. An attacker could use this information to impersonate a valid client and perform unauthorized actions.

Prerequisites: The attacker needs to obtain or guess the UUID of a valid client.

Business impact: Unauthorized access to sensitive data or systems, leading to potential data breaches or service disruptions.

Proposed remediation: Implement stronger client authentication mechanisms such as digital certificates or two-factor authentication to ensure that the client is who they claim to be.

## Risk

Damage potential: 8

Reproducibility: 7

Exploitability: 7

Affected Users: 9

Discoverability: 7

Overall: 8

# Vulnerability 2: Man-in-the-Middle (MitM) Attack on Key Exchange

Threat: An attacker could intercept the public key exchange and replace the client's public key with their own.

Affected component: Public Key Exchange.

Module details: Sending Public Key, AES key exchange.

Vulnerability class: Man-in-the-Middle Attack.

Description: The protocol does not include measures to verify the integrity of the public key being exchanged between the client and server. An attacker intercepting the communication could replace the client’s public key with their own, allowing them to decrypt sensitive AES keys and data.

Prerequisites: The attacker must be able to intercept and modify the communication between the client and server.

Business impact: Exposure of sensitive information, including AES keys, which can lead to unauthorized access to encrypted file content.

Proposed remediation: Implement integrity checks for key exchanges, such as using digital signatures or certificates to ensure that the key comes from a trusted source.

## Risk

Damage potential: 8

Reproducibility: 7

Exploitability: 6

Affected Users: 8

Discoverability: 7

Overall: 7

# Vulnerablity 3: Constant IV in AES Encryption

Threat: Predictable encryption, replay attacks

Affected component: AES encryption module

Module details: File transfer encryption

Vulnerability class: Weak cryptographic initialization

Description: The protocol uses AES encryption, but the Initialization Vector (IV) is constantly set to 0. This significantly weakens the encryption process, as the IV is meant to provide randomness to prevent patterns in the ciphertext, making the encryption predictable and vulnerable to replay attacks.

Prerequisites: The attacker must be able to capture the encrypted communication between the client and the server to exploit the predictable IV.

Business impact: This vulnerability allows an attacker to replay captured communications, potentially allowing unauthorized file access or manipulation. In some cases, it may lead to data exposure.

Proposed remediation: Use a securely generated random IV for each encryption operation to ensure better security

## Risk

Damage potential: 7

Reproducibility: 6

Exploitability: 5

Affected Users: 8

Discoverability: 7

Overall: 7

# Vulnerability 4: Replay Attack on Registeration and Reconnection

Threat: An attacker could capture and replay registration or reconnection messages to gain unauthorized access.

Affected component: Registration, Reconnection.

Module details: Request headers, client UUIDs.

Vulnerability class: Replay Attack.

Description: The protocol does not prevent replay attacks, where an attacker could intercept and resend a legitimate registration or reconnection message to gain unauthorized access. Without mechanisms such as timestamps, the servr has no way to determine if a message has already been used before.

Prerequisites: The attacker needs to capture registration or reconnection messages.

Business impact: Unauthorized access, potentially allowing an attacker to take over an existing session or impersonate a client.

Proposed remediation: Implement nonce-based or timestamp-based mechanisms to detect and prevent replayed messages.

## Risk

Damage potential: 6

Reproducibility: 7

Exploitability: 6

Affected Users: 7

Discoverability: 7

Overall: 6

# Vulnerability 5: Use of CRC32 for Integrity Verfication

Threat: Weak integrity check, undetected tampering

Affected component: File transfer validation

Module details: File transfer integrity verification

Vulnerability class: Insecure integrity check

Description: The protocol uses CRC32 as the integrity verification method for the transferred file. CRC32 is not a cryptographic hash function, and it is prone to collisions, meaning an attacker can tamper with the data and reverse engineer a different file that produces the same CRC32 hash, bypassing integrity checks.

Prerequisites: The attacker needs access to the file transmission process to modify the file in such a way that it results in the same CRC32 value.

Business impact: The lack of a secure integrity check could allow an attacker to replace or alter the file being transferred without detection, leading to possible data corruption or insertion of malicious content.

Proposed remediation: Replace CRC32 with a cryptographically secure hash function, such as SHA-256, to ensure reliable integrity verification.

## Risk

Damage potential: 6

Reproducibility: 5

Exploitability: 4

Affected Users: 7

Discoverability: 6

Overall: 6