שאלה 2 – ניתוח סיכונים בפרוטוקול על פי התבנית

Risk 1: Insecure Default Settings

- **Threat**: Usage of Default IP Address and Port
- **Affected Component**: Server startup process
- **Module Details**: **server/server.py** (the part where IP and port are set)
- **Vulnerability Class**: Misconfiguration
- **Description**: The server uses a default IP and port if the **port.info** file is missing or cannot be read. This could make the server predictable and easier to target for attacks.
- **Result**: Attackers might easily guess the server location and attempt various network attacks.
- **Prerequisites**: Attacker knows or guesses that the default settings are used.
- **Business Impact**: Using easily guessable settings could lead to unauthorized access attempts, overloading the server, or other network-based attacks.
- **Proposed Remediation**: Ensure robust error handling and logging for missing configuration files. Consider requiring explicit administrator confirmation before starting with default settings.
- **Risk Assessment**:
    - Damage Potential: 4
    - Reproducibility: 9
    - Exploitability: 3
    - Affected Users: All
    - Discoverability: 8
    - **Overall**: 6

Risk 2: Lack of Input Validation on Client Registration

- **Threat**: Injection and Spoofing Attacks
- **Affected Component**: Client registration process
- **Module Details**: **server/handleclient.py** (**processRegisterRequest** method)
- **Vulnerability Class**: Injection
- **Description**: There's no evidence of validating the client's name upon registration. Malicious users could potentially inject harmful or misleading content into the database.

- **Result**: Database corruption, misleading user information, or script injections could occur if the client names are used in insecure ways.
- **Prerequisites**: Attacker is able to send crafted registration requests to the server.
- **Business Impact**: Could lead to data integrity issues, unauthorized actions, or cross-site scripting (if data is used in web contexts).
- **Proposed Remediation**: Implement stringent input validation and sanitization on all user-submitted data, especially during the registration process.
- **Risk Assessment**:
    - Damage Potential: 6
    - Reproducibility: 7
    - Exploitability: 5
    - Affected Users: All
    - Discoverability: 5
    - **Overall**: 5.75

Risk 3: Unencrypted Key Exchange

- **Threat**: Exposure of Encryption Keys
- **Affected Component**: Key exchange mechanism
- **Module Details**: **server/handleclient.py** (**processSendingPubkeyRequest** method)
- **Vulnerability Class**: Information Disclosure
- **Description**: Public keys and potentially AES keys are exchanged over unsecured connections, making them susceptible to interception.
- **Result**: An attacker capturing these keys could decrypt sensitive information or impersonate the server/client in communication.
- **Prerequisites**: Ability to intercept network traffic between the client and server.
- **Business Impact**: Compromise of data confidentiality and integrity, leading to potential data breaches.
- **Proposed Remediation**: Implement secure channels (like TLS) for all key exchanges and sensitive data transmission.
- **Risk Assessment**:
    - Damage Potential: 8
    - Reproducibility: 4
    - Exploitability: 6

    - Affected Users: All
    - Discoverability: 6
    - **Overall**: 6

Risk 4: Hardcoded Encryption Key Generation

- **Threat**: Predictable AES Key Generation
- **Affected Component**: AES key management
- **Module Details**: **server/handleclient.py** (**HandleClient** class initialization)
- **Vulnerability Class**: Cryptographic Issues
- **Description**: AES keys are generated in a potentially predictable manner using **os.urandom(16)** without further entropy or security considerations.
- **Result**: Weakly generated keys could be more susceptible to prediction or brute-force attacks.
- **Prerequisites**: Detailed knowledge of the server's key generation mechanism and substantial computational resources.
- **Business Impact**: If encryption keys are compromised, attackers could decrypt sensitive data transferred between the client and server.
- **Proposed Remediation**: Utilize a more robust key generation mechanism, potentially leveraging additional entropy sources and cryptographic libraries designed for secure key generation.
- **Risk Assessment**:
    - Damage Potential: 7
    - Reproducibility: 3
    - Exploitability: 4
    - Affected Users: All
    - Discoverability: 4
    - **Overall**: 4.5