1. Authentication Protocol

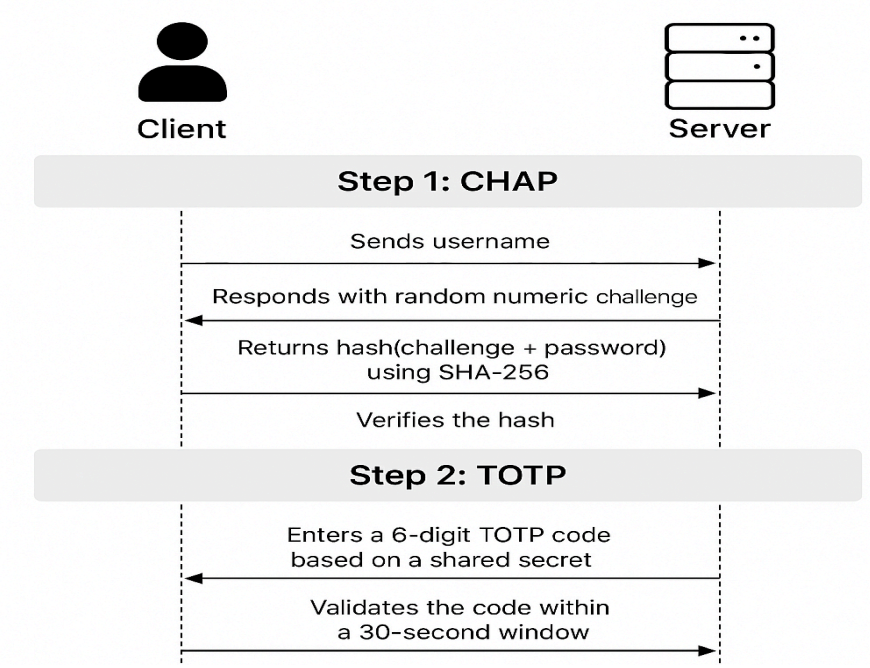
Step 1: CHAP

- The client sends its username.
- The server responds with a random numeric challenge.
- The client returns *hash(challenge + password)* using SHA-256.
- The server verifies the hash.

Step 2: TOTP

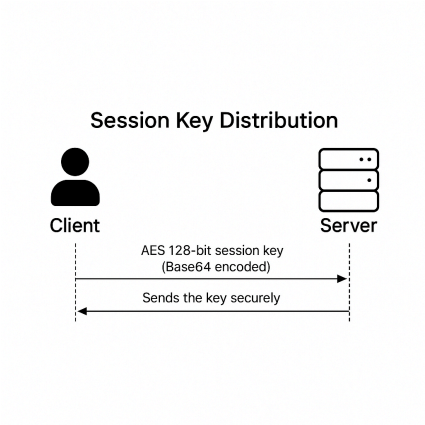
- The client enters a 6-digit TOTP code based on a shared secret.
- The server validates the code within a 30-second window.

If both steps pass, the session is authenticated.



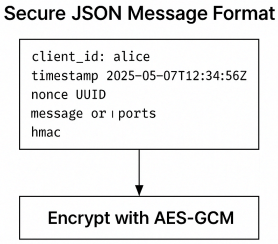
2. Session Establishment

- The server generates an AES 128-bit session key.
- It sends the key (Base64 encoded) securely to the client.
- This key is used to encrypt all subsequent messages using AES/GCM/NoPadding.



3. Secure Communication

- Messages are converted into structured JSON objects.
- Each message includes:
    - *client_id*
    - *message* or *ports*
    - *timestamp*
    - *nonce* (UUID)
    - *hmac* (signed using the session key)
- The server decrypts, validates the HMAC, and checks for nonce reuse.



4. Logging and Analysis

- All alerts are logged with tamper-evident hashing:
    - Each log line includes `||` *`hash(previous_entry + current_entry)`*
    - Stored in *`secure_log.txt`* and *`port_report_log.txt`*
- Port scan results are analyzed and flagged if they:
    - Are high-numbered ports
    - Do not match known safe ports/processes
    - Appear to be a port flood (15+ ports in < 60 seconds)
- Analysis verdict is sent back to client in JSON format.