Securing the SmartOffice Solution

Actions Taken to Secure the Solution

1. ### Hashed Passwords
   Passwords for the client-facing website are hashed and salted, effectively mitigating dictionary attacks even in the event of a database breach.
2. ### Encrypted HTTPS Connections
   Both client and server websites are secured via HTTPS, ensuring that data cannot be intercepted or read during transmission. Node-RED is also secured, and Nginx is configured with TLS encryption.
3. ### Secured Public Broker T
   he public broker, test.mosquitto.org, is secured using TLS through generated certificates.
4. ### Docker Isolation
   Docker is employed to isolate applications within containers, limiting potential damage in case of an incident. We use Docker to containerize two MongoDB databases.
5. ### Restricted Node-RED Port
   Access to the Node-RED UI port is blocked externally, requiring users to authenticate via Flask through Nginx.

Addressing the Top 10 IoT Vulnerabilities

1. ### Weak, Guessable, or Hardcoded Passwords
   Currently, password complexity is not fully addressed. User passwords are simple; however, future improvements will include enforcing complex password requirements (minimum of 20 characters with various character types) and regular changes, particularly for VNC. Users will receive guidance and awareness training on secure password practices.
2. ### Insecure Network Services
   The Oracle Linux server implements explicit port openings, mandatory SSH connections via private keys, and firewall rules that deny all incoming connections unless explicitly authorized.
3. ### Insecure Ecosystem Interfaces
   An authentication system is in place for both administrators and users, but deeper vulnerability management remains to be implemented.
4. ### Lack of Secure Update Mechanisms
   While we have ensured that critical components such as MongoDB, Flask, LoRa ESP32, and Python libraries are up-to-date, comprehensive monitoring for all installed libraries and components is pending.
5. ### Use of Insecure or Outdated Components
   This concern aligns with point 4, where updating and verifying all components will be addressed in due time.
6. ### Insufficient Privacy Protection
   Private data is protected through HTTPS for web communications, TLS for the public broker, and hashed passwords for user authentication.

7. ## Insecure Data Transfer and Storage
   Data transfer is encrypted using HTTPS for web communications and TLS for MQTT. This ensures data confidentiality during transit.
8. ## Lack of Device Management
   Device management is currently facilitated through VNC.
9. ## Insecure Default Settings
   Default settings, such as default passwords for MongoDB and Node-RED, have been changed. Further improvements will focus on identifying and modifying additional default configurations to enhance security.
10. ## Lack of Physical Hardening
    The Ubuntu server hosting the solution is located in a secure facility, providing protection against physical attacks.