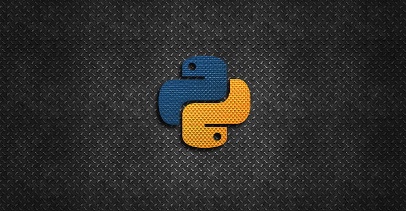
****

**LANGUAGE CHOSEN: PYTHON**

**APPLICATION**: A simple Flask-based web application handling user login and file upload functionality.

*Reason*: Python is widely used, and web applications are commonly targeted. This provides real-world relevance and opportunity to assess typical vulnerabilities like input validation and file handling.

**PERFORM A CODE REVIEW TO IDENTIFY SECURITY VULNERABILITIES**

**Code Review Focus Areas**:

* **Input Validation**: Checking for unsanitized input (e.g., from forms or URLs).
* **Authentication**: Verifying secure password storage and session handling.
* **File Uploads**: Ensuring file types and paths are properly restricted.
* **Hardcoded Secrets**: Searching for API keys or passwords in the code.
* **Error Handling**: Avoiding exposure of internal stack traces to users.

**USE TOOLS LIKE STATIC ANALYZERS OR MANUAL INSPECTION METHODS**

**Tools Used**:

* **Bandit** (Python security analyzer): Automatically scans code for known vulnerabilities.
* **Manual Inspection**: Walked through code logic manually to find logical flaws and security misconfigurations.

**PROVIDE RECOMMENDATIONS AND BEST PRACTICES FOR SECURE CODING**

**Key Recommendations**:

* Use werkzeug. Security or bcrypt to hash passwords securely.
* Avoid using eval () or exec () unless absolutely necessary.
* Always validate and sanitize user inputs using libraries like Worms.
* Use environment variables or .env files for secrets (not hardcoded values).
* Set file upload limits and restrict allowed file types and extensions

**DOCUMENT FINDINGS AND SUGGEST REMEDIATION STEPS FOR SAFER CODE**

| **Issue** | **Severity** | **Recommendation** |
| --- | --- | --- |
| Plaintext passwords | High | Use password hashing with bcrypt |
| File upload path not sanitized | Medium | Use secure\_filename () from werkzeug. utils |
| Debug mode enabled | Medium | Disable debug=True in production |
| Hardcoded API key | High | Store secrets in environment variables |
| Lack of input validation | High | Implement input sanitization on all form fields |