**SECURE CODING REVIEW**

The bandit scan identified a high-severity issue related to running the Flask application with debug=True. This configuration exposes the Flask debugger, allowing arbitrary code execution through the Werkzeug debugger interface, which poses a critical risk in a production environment. To mitigate this vulnerability and ensure secure coding practices, consider the following recommendations:

* Avoid Using debug=True in Production: Never set debug=True in the app.run() method for production deployments. Use app.run() or app.run(debug=False) to disable debug mode.
* Use Environment Variables: Control debug mode using environment variables, making it easier to switch between development and production settings without modifying the source code directly.
* Restrict Access to the Debugger Interface: If debug mode is necessary during development, restrict access by setting the host parameter to 127.0.0.1 to limit exposure.
* Conduct Periodic Code Reviews: Regularly run static code analysis tools like bandit and perform manual reviews to ensure debug mode is not enabled before deployment.
* Implement Logging and Monitoring: Utilize comprehensive logging and monitoring to detect unexpected behavior and potential security issues early.
* Follow Flask’s Deployment Recommendations: Adhere to Flask’s official guidelines for secure application deployment to minimize security risks.

Top of Form

Bottom of Form