# Security Requirements

## GitHub App Authentication & Authorization

- Use GitHub App authentication with OAuth or GitHub's JWT-based authentication.
- Follow least privilege principle, granting only necessary permissions (e.g., read PRs, write comments).
- Implement webhook validation to verify event authenticity.

## Secure API Interactions

- Communicate with GitHub's API over TLS 1.2/1.3.
- Rate-limit API calls to prevent abuse and avoid exceeding GitHub's API limits.
- Store API credentials securely using GitHub Secrets or an encrypted environment variable store.

## Input Validation & Code Execution Security

- Validate and sanitize PR data to prevent injection attacks.
- Use sandboxed environments for code analysis to mitigate risks from untrusted code.

## LLM Security & Bias Prevention

- Fine-tune or prompt LLMs to avoid insecure or biased recommendations.
- Implement a manual verification step for high-risk security suggestions before applying fixes.

## Audit Logging & Monitoring

- Maintain logs of PR reviews, API calls, and security events.
- Set up GitHub Security Alerts & monitoring for suspicious activities.

---

# Security Planning

## 1. GitHub App Deployment Security

- Deploy as a GitHub App, ensuring it follows GitHub's security best practices.
- Use GitHub's webhook signature verification to authenticate incoming events.
- Restrict API scopes to only necessary permissions to prevent overreach.

# 1. Threat Modeling & Risk Assessment

- Identify threats like webhook tampering, API abuse, and adversarial LLM inputs.
- Conduct regular security audits and penetration testing of the GitHub integration.

## 2. Incident Response & Mitigation

- Implement a fallback mechanism in case of system errors (e.g., allow manual reviews if the AI fails).
- Set up automated revocation of compromised API keys or leaked tokens.