Ruby on Rails Security Audit Checklist Template

Date of Audit: [Insert Date]
Auditor: [Insert Name]

---

1. Framework and Dependencies

- □ Is the Ruby version up-to-date with the latest stable release?
- □ Is the Rails version up-to-date with the latest stable release?
- □ Have all gems been audited for vulnerabilities using tools like `bundler-audit` or `Gemnasium`?
- □ Are unused gems removed from the `Gemfile`?

---

2. Input Validation

- □ Are all user inputs validated using Rails strong parameters?
- □ Are appropriate format validations applied to models?

---

3. Password Security

- □ Is the `bcrypt` gem being used for password hashing?
- □ Are passwords validated for minimum length and complexity?

---

4. HTTPS and Secure Connections

- □ Is HTTPS enforced in production (`config.force_ssl = true`)?
- □ Is HSTS (HTTP Strict Transport Security) enabled via headers?

---

5. CSRF Protection

- □ Is CSRF protection enabled (`protect_from_forgery` in `ApplicationController`)?
- □ Are all forms including authenticity tokens?

---

6. HTTP Security Headers

- □ Are secure headers configured using the `secure_headers` gem or custom middleware?
- □ Is a Content Security Policy (CSP) implemented?

---

7. SQL Injection Prevention

- □ Are queries written using Active Record methods to prevent SQL injection?

- □ Is all input sanitized before database interactions?

---

8. Cross-Site Scripting (XSS)

- □ Is user-generated content sanitized before rendering?
- □ Is the `html_safe` method used cautiously and only when necessary?

---

9. Mass Assignment Protection

- □ Are strong parameters used to whitelist allowed attributes?

---

10. Data Encryption

- □ Are sensitive data fields encrypted in the database (e.g., using `attr_encrypted`)?
- □ Are environment secrets stored securely (e.g., using `Rails.credentials`)?

---

11. Brute Force Protection

- □ Is account lockout implemented after a certain number of failed login attempts?
- □ Are CAPTCHAs or similar tools used for critical actions?

---

12. Cookie Security

- □ Are cookies marked as `secure` and `HttpOnly`?
- □ Are Rails encrypted cookies being used?

---

13. Admin Panel Security

- □ Is access to the admin panel restricted by authentication and IP filtering?
- □ Are admin accounts regularly reviewed for active status?

---

14. File Upload Security

- □ Are file uploads restricted to specific MIME types?
- □ Is file size validation implemented?
- □ Are uploaded files scanned for malware?

---

15. Logging and Monitoring

- □ Are logs monitored for suspicious activity?
- □ Is sensitive data excluded from logs?

---

16. Error Reporting

- □ Are stack traces and debugging information disabled in production?

---

17. Rate Limiting

- □ Is rate limiting implemented using middleware like `rack-attack`?

---

18. Role-Based Access Control

- □ Are user roles clearly defined?
- □ Are authorization checks performed for all sensitive actions?

---

19. Security Scanning

- □ Has the application been scanned for vulnerabilities using tools like `Brakeman`?
- □ Are results from scans reviewed and addressed promptly?

---

20. Backup Strategy

- □ Are data backups encrypted?
- □ Are backups stored in secure, offsite locations?
- □ Are backups tested periodically for integrity?

---

21. Security Training

- □ Have developers received training on secure coding practices?
- □ Is security awareness training conducted for all team members?

---

22. External Integrations

- □ Are API keys and secrets securely stored (e.g., environment variables)?
- □ Are third-party integrations reviewed for security vulnerabilities?

---

23. Final Checks

- □ Are all staging and development credentials removed before deploying to production?
- □ Is a final penetration test conducted before release?

Audit Summary

Overall Security Grade: [A/B/C/D/F]
Immediate Actions Required: [List key actions]

Next Scheduled Audit: [Insert Date]