

✦ TEMPLATE · TEMPLATES

Security Review Checklist

A pre-launch security review form for AI features: identity, data, prompt injection, tool permissions, model supply chain, logging and operations — with findings and sign-off.

FORMAT	LEVEL	READ TIME	UPDATED
Template	Beginner	Fill-in	Oct 2026

OVERVIEW

What this template covers

AI features inherit every classic application risk and add new ones: instructions hidden in documents, models that repeat what they were shown, and agents holding write access. This form is the review we run before an AI feature goes live. Work through each area, record findings with severity and owners, and sign off only when blocking findings are closed.

KEY TAKEAWAYS

- Treat every document, email and web page the model reads as untrusted input.
- Enforce permissions in code at retrieval and tool level — never rely on the prompt.
- Log enough to reconstruct any answer, without logging more personal data than needed.
- Rate limits and cost caps are security controls for AI features.
- A finding without an owner and a date is not a finding, it is a hope.

WHO IT'S FOR

- Security engineers and reviewers
- AI / ML and platform engineers
- Engineering managers preparing a launch
- Data protection officers

HOW TO USE IT

The feature team completes section 01 and self-assesses; a reviewer verifies, logs findings and signs off. Every box shows a worked example — a support assistant for a fictional broadband provider. Replace the examples with your own.

CONTENTS

01 System under review
02 Identity, access and data
03 Prompt injection, output and tools
04 Supply chain, logging and resilience
05 Findings log
06 Review decision

01 / SCOPE

System under review

Describe what is being reviewed precisely enough that a reviewer who has never seen it can follow the data.

Feature, version and launch date

01

e.g. “Customer support assistant v1.2, web and mobile, planned launch [date].”

WORKED EXAMPLE

Support Assistant v1.2 for a fictional broadband provider — web chat and mobile app, answers plan and billing questions and can book technician callbacks. Review held 12 Oct 2026; planned launch 3 Nov 2026.

Architecture and data flow

02

User → app → model provider / retrieval / tools. Attach a diagram; note every external service.

WORKED EXAMPLE

- Customer signs in → app backend → orchestration service
- Orchestrator retrieves from the help-center index and the customer’s own tickets
- Hosted model API generates the answer; one write tool: `create_callback`
- External services: model provider, vector database, ticketing system, SMS gateway

Data classes processed

03

Public, internal, confidential, personal, sensitive personal (health, financial, children’s data).

WORKED EXAMPLE

- Public: help-center articles and plan sheets
- Internal: agent runbooks (staff only — must never reach customers)
- Personal: name, phone, email, address, account and ticket history
- No payment card data — billing questions show amounts only

Model providers and hosting 04

Provider, region, data retention and training-use terms, self-hosted components.

WORKED EXAMPLE

Hosted model API in an India region under an enterprise agreement: no training on our data, 30-day abuse-monitoring retention. Vector database and orchestrator are self-hosted in our own cloud account (Mumbai).

✦ 02 / ACCESS

Identity, access and data

Tick only what you have verified, not what is intended.

IDENTITY AND ACCESS

- ☐ Every request is authenticated and tied to a user or service identity
- ☐ Retrieval filters by the caller's permissions inside the index query
- ☐ Admin and evaluation consoles are behind SSO and role checks
- ☐ Service credentials are scoped, stored in a secrets manager and rotated

DATA PROTECTION

- ☐ Data sent to model providers is minimized to what the task needs
- ☐ Provider terms reviewed: retention, training use, region
- ☐ Personal data in prompts and logs is masked or retention-limited
- ☐ Deletion requests propagate to indexes, caches and logs
- ☐ Encryption in transit and at rest for indexes, caches and logs

✦ 03 / MODEL RISKS

Prompt injection, output and tools

The risks specific to language models and agents.

PROMPT INJECTION

- [ ] Untrusted content is clearly delimited from instructions
- [ ] Injection test cases (direct and indirect, via documents) are in the test suite
- [ ] Write tools are unavailable while untrusted content is in context, or require approval
- [ ] System prompts contain no secrets

OUTPUT HANDLING

- [ ] Model output is encoded before rendering (no raw HTML / script injection)
- [ ] Output is never executed as code or SQL without validation
- [ ] Links and file paths in output are validated
- [ ] Content filters cover the harms relevant to this audience

TOOLS AND AGENTS

- [ ] Each tool has the narrowest permission that works
- [ ] Irreversible actions require human approval
- [ ] Step, time and spend limits are enforced in code
- [ ] Tool arguments are validated server-side

✦ 04 / OPERATIONS

Supply chain, logging and resilience

What keeps the feature safe after launch day.

SUPPLY CHAIN

- [ ] Model versions are pinned and changes go through review
- [ ] Third-party libraries and model weights come from verified sources
- [ ] Dependencies scanned for known vulnerabilities

LOGGING AND MONITORING

- [ ] Each request logs user, input, retrieved sources, tool calls and output
- [ ] Logs are access-controlled and retention-limited
- [ ] Alerts on abuse patterns, error spikes and cost spikes

RESILIENCE AND ABUSE

- [ ] Per-user rate limits and per-tenant cost caps
- [ ] Graceful behavior when the model provider is down
- [ ] Kill switch to disable the feature tested
- [ ] Incident runbook names an on-call owner

✦ 05 / FINDINGS

Findings log

Record every gap. Severity guides the launch decision below.

Severity scale

SEVERITY	MEANING	LAUNCH IMPACT
Critical	Exploitable now; exposes data or allows unauthorized actions	Blocks launch
High	Likely exploitable or a missing core control	Blocks launch unless risk accepted by named owner
Medium	Weakness with limited impact or needing unusual conditions	Fix within agreed period after launch
Low	Hardening or hygiene improvement	Track in backlog

Findings

#	FINDING	SEVERITY	OWNER	DUE / STATUS
01	Index query had no audience filter — staff runbooks were retrievable by customers	Critical	K. Iyer, Platform	10 Oct · closed, retested 13 Oct
02	Instructions hidden in ticket attachments can call `create_callback` without customer confirmation	High	A. Rao, AI engineering	28 Oct · in progress
03	Prompt logs keep full phone numbers and emails for 180 days	Medium	S. Pillai, Data engineering	31 Oct · open
04	No per-user rate limit on the chat endpoint; only a global cap	High	K. Iyer, Platform	15 Nov · risk accepted
05	Web widget rendered raw HTML from model output	High	M. Das, Frontend	14 Oct · closed
06	Model called via a floating “latest” alias, not a pinned version	Low	A. Rao, AI engineering	30 Nov · open

Accepted risks

01

Any High finding launched with, the reason, compensating controls, and who accepted it.

WORKED EXAMPLE

#04 — no per-user rate limit. Accepted by A. Rao (AI engineering lead) until 15 Nov 2026; compensated by a global spend cap, a 3× volume alert and a tested kill switch.

✦ 06 / SIGN-OFF

Review decision

Sign only when every Critical finding is closed and every High finding is closed or formally accepted.

Decision

01

Approved / approved with conditions / not approved. List conditions and the re-review trigger.

WORKED EXAMPLE

Approved with 2 conditions: close and retest #02 before launch; close #03 by 31 Oct. Re-review if a new write tool or data source is added.

Security reviewer — name, signature, date

02

WORKED EXAMPLE

R. Menon, Head of Security · approved with 2 conditions · 16 Oct 2026

Feature owner — name, signature, date

03

WORKED EXAMPLE

J. Fernandes, Product Manager, Digital Support · signed · 16 Oct 2026

Data protection officer (if personal data) — name, signature, date

04

WORKED EXAMPLE

P. Kulkarni, Data Protection Officer · signed, subject to #03 · 17 Oct 2026

GET IN TOUCH

Want us to run this with your team?

We run pre-launch security reviews of RAG systems and agents, including prompt-injection testing, permission audits and tool hardening.

- Pre-launch AI security review
- Prompt-injection test suites
- Agent permission hardening

BOOK AN AI STRATEGY SESSION →

EMAIL
hello@splendenslabs.com

WEB
splendenslabs.com

OFFICE
Sona Incubation Foundation, Salem