Is your AI feature ready for a security review?

For software companies shipping chatbots, AI assistants, RAG search and agents. Part 1 is five questions an enterprise buyer's security review will ask. Part 2 is ten technical checks for your engineering team, mapped to the OWASP Top 10 for LLM Applications 2025.

Part 1: about 20 minutes **Part 2:** for your engineers **Only test** systems you own

PART 1 · BEFORE THE QUESTIONNAIRE ARRIVES

Can you answer these today?

Write the answers down. If an answer is "I think so", treat it as "not yet".

- [ ] 01 **Which customer data reaches the model, and which provider handles it?**

  List every place customer data goes into a prompt, a retrieval index, a log or a fine-tuning set. Name the model provider for each, and what your contract and account settings say about how long they keep it and whether they may train on it.

- [ ] 02 **Can one customer's data show up in another customer's answer?**

  Think about shared retrieval indexes, caches, conversation history and fine-tuned models. Know where tenant separation is enforced, and whether it's in code or only in the prompt.

- [ ] 03 **What can the AI do on its own?**

  List every tool, API and action the feature can call, with what it can read, write, send or delete. Know which actions need a person to confirm first.

- [ ] 04 **What happens when someone tells it to ignore its instructions?**

  Including instructions hidden in a document, email, web page or support ticket the feature reads. Have you tried it, and do you have the results?

- [ ] 05 **How do you know a model or prompt change didn't break anything?**

  A saved set of test prompts, including attacks, that you rerun before each release, plus logs that let you investigate a bad answer after the fact.

PART 2 · FOR YOUR ENGINEERING TEAM

Ten technical checks

Each one names the OWASP Top 10 for LLM Applications 2025 risk it covers, so you can point a reviewer to the standard. Only test systems you own or are authorized to test.

- [ ] 01 **Treat everything the model reads as untrusted**

OWASP LLM01 · PROMPT INJECTION

Protects: your data and actions, from instructions planted in user input or in content the feature reads.

Prompt injection can come straight from a user or hide inside retrieved documents, emails, web pages and file uploads. The system prompt isn't a security boundary. Test both kinds: ask the feature to ignore its rules, and plant an instruction in a document it will retrieve, then see what it does.

- [ ] 02 **Enforce permissions before retrieval, not in the prompt**

OWASP LLM08 · VECTOR AND EMBEDDING WEAKNESSES

Protects: each customer's documents, from other customers and from users without access.

Filter every retrieval query by tenant and user permissions in your code, on the server. A shared index with a "please only use this customer's data" instruction is not separation. Test with two tenants: put a unique phrase in tenant A's documents and try to get it back as tenant B.

- [ ] 03 **Give tools and agents the least privilege**

OWASP LLM06 · EXCESSIVE AGENCY

Protects: against the model taking actions the user couldn't take, or didn't mean to.

Each tool should run with the signed-in user's permissions, not a shared admin key, and check authorization in code. Remove tools the feature doesn't need. Require a person to confirm anything that sends, pays, deletes or changes records.

- [ ] 04 **Handle model output like user input**

OWASP LLM05 · IMPROPER OUTPUT HANDLING

Protects: your app and users, from code, links and queries the model writes.

Escape output before rendering it as HTML or Markdown, and never pass it straight to SQL, a shell, `eval` or a URL fetch. Watch for Markdown images and links that point at outside addresses: they can carry data out of a chat in the address itself.

- [ ] 05 **Keep secrets out of the system prompt**

OWASP LLM07 · SYSTEM PROMPT LEAKAGE

Protects: keys, internal details and access rules, from anyone who coaxes the prompt out.

Assume the system prompt can be extracted. No API keys, passwords, internal URLs or customer names in it, and no access rules that exist only there. Enforce those in code.

- [ ] 06 **Know what you send to the model provider**

OWASP LLM02 · SENSITIVE INFORMATION DISCLOSURE

Protects: customer and personal data, in prompts, logs and model answers.

Send only the data the task needs, and mask or remove what it doesn't. Check your provider's data-retention and training settings for the account and API you actually use, and have a data processing agreement in place. Check that answers can't repeat other users' personal data.

- [ ] 07 **Inventory your models, SDKs, plugins and MCP servers**

OWASP LLM03 · SUPPLY CHAIN

Protects: against a changed or compromised component you didn't know you depended on.

Keep a list of every model (with version), SDK, plugin, MCP server and third-party dataset the feature uses, who maintains it, and what it can access. Pin versions, and review a new or updated component before it ships.

- [ ] 08 **Control what goes into your knowledge base and training data**

OWASP LLM04 · DATA AND MODEL POISONING

Protects: the feature's answers, from data someone planted to steer them.

Know who and what can write to the documents, index or fine-tuning data the feature learns from, including customer uploads, synced tools and web pages. Review or restrict sources, and be able to trace and remove a bad document.

- [ ] 09 **Cap usage per user, tenant and day**

OWASP LLM10 · UNBOUNDED CONSUMPTION

Protects: your AI bill and availability, from abuse and runaway loops.

Set rate limits per user and tenant, a maximum input and output size, limits on agent steps and tool calls, and a spending alert with the provider. A single script or a stuck agent loop shouldn't be able to run up your bill.

- [ ] 10 **Log, evaluate and re-test on every change**

OWASP LLM09 · MISINFORMATION, PLUS LOGGING

Protects: customers from confident wrong answers, and you from changes that quietly undo a fix.

Keep a saved set of test prompts, including attack prompts and questions with known right answers, and rerun it whenever the model, prompt, tools or data change. Log prompts, tool calls and answers with personal data masked, so you can investigate. For high-impact answers, show sources or add a human review.

YOUR RESULT

How did you do?

You can answer all 5 and confirm all 10

A strong position for a security review. Keep the evidence (test results, settings, the inventory) where your sales team can find it.

Some answers are "I think so"

Those are the questions an enterprise security review will ask. Write down what you know and test what you don't.

No one has tried to break it yet

That's normal for a new AI feature, and it's the gap to close before a big customer asks. Have someone outside the build test it.

Want an independent review before the buyer asks?

Our **AI Security Assessment** tests your AI feature against these risks and more. You get a report your engineers can act on, and a summary you can share with customers.

- We never change your code. We find the issues, explain them in plain English, recommend fixes, and re-test after your team makes the changes.
- Testing is read-only by default, and nothing is tested until you've signed a written authorization.
- For software companies in Canada (outside Quebec) and the United States.

Book a free 30-minute scoping call → axieo.ca/contact

General guidance, not a full security assessment. Only test systems you own or are authorized to test. OWASP is a trademark of the OWASP Foundation; this checklist is not affiliated with or endorsed by OWASP.