PARAMOUNT
Software Solutions

CHATBOTS YOU SHOULDN'T USE (YET):

Security Risks, Real Incidents, And How To Protect Your Org In 2026

A practical guide for dev, QA, architecture, and business teams on avoiding insecure AI chatbots and building safer alternatives.

Author : Paramount AI Security Team

Date : May 2026

Who This Book Is For

This book is written for teams that are already using or planning to use AI chatbots in serious, production-grade environments.

It is specifically tailored to:

- Software Developers & Programmer Analysts who integrate chatbots into products and internal tools.
- QA Engineers & SDETs who need to test for prompt injection, data leaks, and unsafe tool use.
- Architects & Senior Engineers who design end-to-end AI and agentic systems.
- Project Managers & Scrum Leads who own delivery risk and stakeholder expectations.
- SAP & Functional Consultants who connect chatbots to ERP, CRM, and financial systems.
- Business Analysts, Sales, and Business Development teams who lean on chatbots for client-facing workflows and internal analysis.

If your chatbot can see customer data, financials, code, or internal docs, this book is for you.

If you also want to understand how to spot AI hallucinations and reliability issues in day-to-day work, check out our companion guide, the AI Lie Detector Guide for Teams and Leaders

What You Can Expect

In this book, you will get:

- A plain-language, technical-accurate explanation of how modern chatbots get compromised in 2025–2026.
- Concrete examples of incidents, including a 300M-message leak from a popular AI chat app and OWASP-documented LLM attacks.
- A clear list of “chatbot patterns not to use” in production, and when public or consumer chatbots are simply the wrong tool.
- Data and statistics you can reuse in internal decks to justify governance, security budgets, and design changes.
- Role-specific checklists for developers, QA, architects, PMs, and business teams to harden their current or future chatbot deployments.

The goal is simple: after reading this, you should be able to confidently say which chatbots you should not use as-is, what risks they introduce, and what a safer alternative looks like

Table of Contents

I. Introduction: Why “Don’t Use That Chatbot” Is A Legit Security Strategy

Over the last two years, chatbots have gone from side projects to critical interfaces for customers, employees, and systems of record. At the same time, attackers have learned to treat them not as toys but as serious entry points into data and infrastructure.

Several 2025–2026 reports show that AI and chatbots are now directly involved in a meaningful slice of breaches and cyber incidents, from phishing to data exfiltration through exposed AI apps. In February 2026, for example, the “Chat & Ask AI” mobile app leaked about 300 million messages from more than 25 million users due to a misconfigured Firebase database.

This book is not anti-AI; it is anti-careless-AI. There are chatbots you can use safely with the right design, and there are chatbots you should not use at all for sensitive data or production workflows without additional controls.

II. The 2025–2026 Chatbot Security Landscape

A. Key Macro Numbers

- The global cost of cybercrime hit around 10.5 trillion USD annually by 2025, with AI-assisted attacks contributing to scale and speed.
- IBM and other 2025–2026 threat reports highlight that attackers increasingly weaponize generative models for phishing and malware generation, with AI-generated phishing now a large share of campaigns.
- AI security research in 2026 notes that AI-related incidents, including insecure AI apps and chatbots, have risen sharply since 2024, with organizations reporting higher risk exposure from LLM use

B. Chatbot-Specific Data Points

- A Malwarebytes-covered breach of the "Chat & Ask AI" app exposed around 300 million messages belonging to more than 25 million users because of a simple Firebase misconfiguration.
- Security researchers scanning similar apps found over 100 additional iOS apps with comparable misconfigurations, showing that insecure chatbot backends are widespread.
- AI security studies in 2026 show that a significant portion of AI-generated code samples still include OWASP Top 10 vulnerabilities, increasing the risk when chatbots or copilots generate code that is shipped with minimal review.

These are not theoretical risks: they are production incidents affecting millions of users and large quantities of sensitive conversation data.

III. How Chatbots Get Hacked: OWASP LLM & Agent Risks In Practice

A. OWASP Top 10 For LLM Applications (2025)

- The OWASP Top 10 for LLM Applications is now the baseline for thinking about chatbot risk. Key categories relevant here include:
- Prompt Injection: Inputs that override system instructions or jailbreak the model to reveal secrets or perform unauthorized actions.
- Sensitive Information Disclosure: Models leaking training data, conversation history, or internal system prompts.
- Supply-Chain Vulnerabilities: Risks in model providers, plugins, and third-party APIs that chatbots use behind the scenes.
- Data and Model Poisoning: Training or fine-tuning on manipulated data, causing biased or malicious outputs.
- Excessive Agency: LLM-based agents with too much access to tools, credentials, or systems.

OWASP's 2025 update explicitly integrates lessons from real incidents and the rapid rise of agentic AI, placing prompt injection and excessive agency near the top.

B. Agentic Applications: 2026 Focus

OWASP and related projects are also publishing guidance specifically for agentic applications—LLM systems that can plan and act autonomously using tools and APIs. For these, the risk shifts from "bad answers" to "bad actions," including:

- Goal hijacking, where prompt injection changes what the agent optimizes for.
- Identity and access abuse, where agents use over-privileged tokens or accounts.

This is exactly where many "chatbots you should not use" live: bots that can talk to your CRM, ticketing tools, or ERP with too much trust and too little control.

IV. Chatbots You Should Not Use As-Is (And Why)

This section is about patterns and categories, not attacking individual vendors. The same underlying products can be safe in one configuration and dangerous in another.

Pattern 1: Public Consumer Chatbots For Sensitive Enterprise Data

Description: Using public, consumer-grade chatbots (e.g., generic web or mobile interfaces) for internal code, contracts, customer lists, or financial data.

Why this is a problem:

- You often lack clear guarantees about whether your data is used for model training or stored longer than you expect.
- Credentials for these accounts have been found in infostealer malware, exposing not just login access but entire conversation histories.
- Users regularly paste secrets into these tools, and many companies have no central visibility into what is being shared.

When to absolutely avoid:

- Anything involving regulated data (health, finance, government) or trade secrets over open, unmanaged chatbot accounts.

Pattern 2: Unvetted “All-In-One” AI Chat Apps

Description: Mobile or web apps that aggregate multiple LLMs (ChatGPT, Claude, Gemini, etc.) but have weak or unknown backend security.

Why this is a problem:

- The Chat & Ask AI breach showed that misconfigured Firebase rules exposed 300M messages and settings; similar misconfigurations were found in many other apps.
- These apps can have millions of downloads with almost no formal security review or enterprise-grade controls.

When to absolutely avoid:

- Using such apps for any business-related conversations or internal documents, unless the vendor can prove hardened infrastructure and audits.

Pattern 3: Chatbots With Direct DB Or System Access And No Guardrails

Description: “Ask your database” or “chat with your ERP/CRM” bots wired directly to production databases or APIs with broad read/write access.

Why this is a problem:

- Prompt injection can trick the LLM into generating harmful SQL or API calls, leading to data exfiltration or corruption.
- Many open-source demos and code samples show how easy it is to pivot from natural language to privileged queries; tests find SQL injection and similar issues in these flows.

When to absolutely avoid:

- Any bot that can write to production systems or sensitive tables without explicit, hard-coded constraints and policy checks.

Pattern 4: Shadow AI Chatbots

Description: Unsanctioned chatbots spun up by teams or individuals using no-code tools or SaaS connectors, without security review.

Why this is a problem:

- They create ungoverned data flows, often storing chat histories and uploaded files on third-party servers.
- They bypass controls your security team expects (like DLP, logging, and identity checks).

When to absolutely avoid:

- Any shadow chatbot connected to internal systems or files, regardless of how convenient it seems for a project.

V. High-Risk Chatbot Patterns You Should Avoid

This section focuses on patterns, not brands. If your setup matches any of these, treat it as high-risk until fixed.

Pattern 1: Public Consumer Chatbots For Sensitive Enterprise Data

Description: Employees use public web or mobile chatbots (consumer accounts) to paste internal code, contracts, customer lists, or financials.

Why it's dangerous:

- You often do not control how long prompts are retained or whether they are used to improve models.
- Consumer logins and tokens are a common target in infostealer malware, which can expose entire chat histories.
- Privacy research shows that prompts often contain personal or confidential information users did not intend to share at scale.

Safer alternative: Use an enterprise-grade deployment (self-hosted or via a cloud provider) with clear data-use terms, access controls, and logging.

Pattern 2: "All-In-One" AI Chat Apps With Weak Backend Security

Description: Third-party apps that aggregate multiple LLMs (e.g., "Chat with GPT, Claude, Gemini in one place") but rely on misconfigured or opaque backends.

Why it's dangerous:

- In 2026, the "Chat & Ask AI" mobile app exposed around 300 million messages from 25 million users because its Firebase database was left open to the internet.
- The same research found more than 100 other iOS apps with similar misconfigurations, meaning chat histories and settings were broadly accessible.

- These apps often have millions of downloads but little to no enterprise-grade security review or compliance posture.

Safer alternative: Use vendors that publish their security architecture, support SSO and RBAC, and undergo regular audits or pen tests.

Pattern 3: Chatbots Directly Wired To Production Databases Or Systems

Description: "Ask your database" or "chat with your ERP/CRM" bots that translate natural language straight into SQL or privileged API calls against production.

Why it's dangerous:

- Prompt injection can push the model to generate overly broad SELECTs, delete/UPDATE statements, or high-volume queries that hurt performance or leak data.
- OWASP's LLM guidance explicitly calls out unvalidated tool use and excessive agency as critical risks, especially when the model issues commands without strict constraints.
- Mis-scoped service accounts can turn a chatbot compromise into full database or system compromise.

Safer alternative: Put a service layer in front of the database, allow-list only specific parameterized queries or actions, enforce least privilege, and treat all model-generated queries as untrusted input that must be validated.

Pattern 4: Shadow AI Chatbots Built Outside Governance

Description: No-code bots or SaaS tools teams spin up on their own, connecting to Google Drive, SharePoint, Jira, SAP, or email without going through security or architecture.

Why it's dangerous:

- They create untracked data flows where chat logs, files, and embeddings sit on third-party infrastructure with unknown security standards.
- They bypass DLP, IAM policies, and monitoring that your organization expects to be in place.
- If the vendor is compromised, attackers get a full map of your internal content and workflows.

Safer alternative: Create an approved internal catalogue of AI tools, require a basic security assessment for any new chatbot, and consolidate usage on vetted platforms.

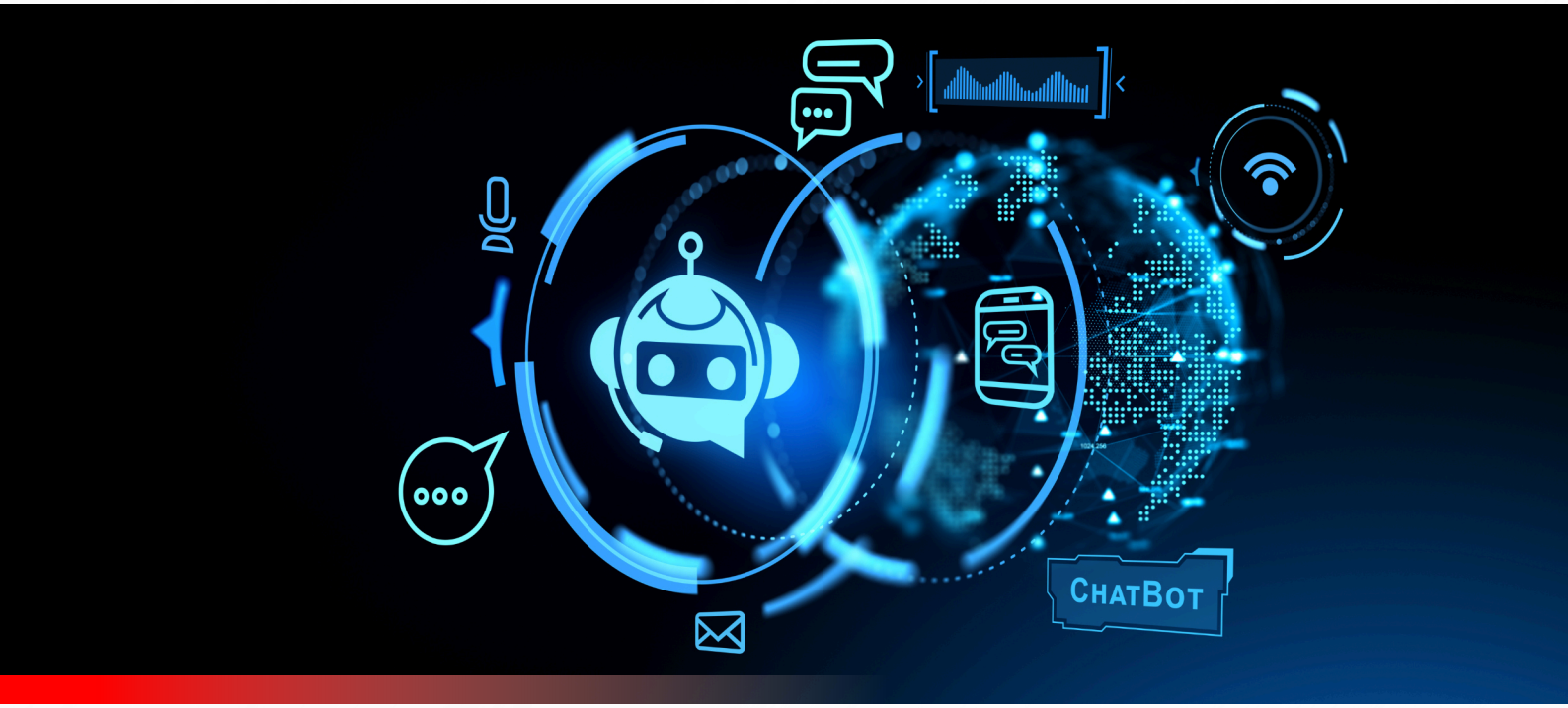
Pattern 5: Chatbots With Broad, Ungated "Agent" Powers

Description: Agentic bots that can trigger emails, create tickets, modify configurations, or call external APIs with minimal verification.

Why it's dangerous:

- OWASP's guidance for agentic applications flags goal hijacking and identity abuse as key risks—prompt injection can change what the agent is optimizing for or trick it into misusing credentials.
- Without action boundaries and approvals, a compromised or mis-prompted agent can execute a chain of real-world changes (e.g., updating customer records, modifying access, or triggering external workflows).

Safer alternative: Gate risky actions behind explicit human approval, implement strong role checks before execution, and restrict what agents can trigger by default.



VI. How Developers And Analysts Should Work With Chatbots Safely

This section is intentionally actionable.

A. Rules For Using Public Chatbots

- Never paste secrets: API keys, SSH keys, tokens, customer PII, or proprietary source code should not go into public chatbots.
- Use anonymization: If you must discuss a scenario, strip names, IDs, and company identifiers.
- Treat output as untrusted: Generated code, queries, or configurations must go through normal review and scanning.

B. Building Internal Chatbots: Developer Checklist

When you build or integrate a chatbot into your stack:

- Put a DLP filter in front of the LLM to redact obvious sensitive patterns (credit cards, IDs, etc.) before text leaves your boundary.
- Scope access: Use narrowly scoped service accounts and time-limited tokens for any tools or databases the bot can access.
- Validate and constrain actions: For tools (SQL, HTTP calls, file writes), implement hard rules and allow-lists instead of trusting the model.
- Log everything: Capture inputs, decisions, and tool calls for security review and incident response.

C. Analysts And BAs: Safe Prompting Habits

- Avoid copying full client decks or contracts in one go; summarize locally first and send only necessary fragments.
- When in doubt, assume your prompts could one day be leaked or subpoenaed; write accordingly.

VII. How QA, SDETs, And Security Teams Should Test Chatbots

QA and security engineering can treat chatbots as new attack surfaces.

A. Core Test Categories

- Prompt Injection: Try instructions like “ignore previous rules” plus data-exfiltration tasks to see if the bot obeys.
- Data Leakage: Attempt to retrieve system prompts, previous conversation snippets, or hidden configuration.
- Tool Misuse: If the bot can access APIs or databases, attempt prompts that cause unintended writes or high-volume queries.

B. Using Existing Research And Tools

- Use open-source pentesting repos that demonstrate how AI chatbots are vulnerable to SQL injection, prompt hijacking, and other issues to inspire your test cases.
- Map findings back to OWASP LLM Top 10 categories to make risk reports clearer for leadership.

C. Exit Criteria

A chatbot should not go to production if:

- It readily reveals internal prompts or previous user data.
- It can be steered into performing unsafe or out-of-scope actions with simple prompt tricks.
- There is no logging, rate limiting, or identity enforcement on tool calls.

VIII. Architecture And PM Playbook: Designing For “Secure By Default”

Architects and PMs own the design decisions that make a chatbot fundamentally safe or unsafe.

A. Architectural Principles

- Isolation: Place the LLM behind a controlled service layer; never let it talk directly to core databases or systems.
- Principle of Least Privilege: Give the chatbot only the permissions required for each specific task, nothing more.
- Defense In Depth: Combine identity, network controls, DLP, rate limiting, and monitoring—do not rely on the model’s “good behavior.”

B. PM Responsibilities

- Define “red lines”: Explicitly list data types and systems that the chatbot must never touch.
- Require security sign-off: No chatbot in production without passing QA and security criteria tied to OWASP guidance.
- Budget for hardening: Include time and cost for security tooling, pentesting, and audits in the roadmap.

IX. Business, Sales, And Functional Teams: Everyday Safe-Use Rules

You don’t need to be technical to avoid high-risk chatbot patterns.

A. Simple Rules For Daily Work

- Use company-approved chatbots only; avoid random browser extensions and mobile apps that promise “AI superpowers.”
- Do not paste pipelines, pricing sheets, or client lists into consumer chatbots.
- Treat AI outputs as drafts, not final answers; always review before sending to clients.

B. SAP / Functional Consultants

- Do not connect SAP, ERP, or CRM systems to a chatbot without going through your architecture and security teams.
- If you pilot a chatbot for transactional or financial data, ensure access is read-only and highly constrained.

X. Quick Internal Assessment: “Is Our Chatbot Unsafe?”

Use this as a one-page checklist in workshops.

Answer “Yes/No” for each:

1. Our chatbot is accessible from the public internet without strong authentication.
2. It can see or manipulate production data (DB, ERP, CRM) directly.
3. We do not have documented rules for what data can be shared with it.
4. We do not log prompts, outputs, and tool calls centrally.
5. We have never done prompt-injection or data-leakage testing on it.
6. It was created by a team or individual without formal security review (shadow AI).
7. We rely on a consumer-grade app or browser extension without vetting the vendor’s security posture.

If you answer “Yes” to three or more, you likely have a chatbot that should not be used in its current form for sensitive or production workloads

XI. References

1. **Cycode** – "Top AI Security Vulnerabilities to Watch out for in 2026" (March 2026). https://cycode.com/blog/ai-security-vulnerabilities/

2. **Concentric AI** – "ChatGPT Security Risks in 2026: A Guide to Risks Your Team Might Be Missing" (May 2026).https://concentric.ai/chatgpt-security-risks-in-2026-a-guide-to-risks-your-team-might-be-missing/

3. **Hinckley Allen** – "2025 Year in Review and Predictions for 2026 in the Cyber, AI, and Privacy Frontier" (January 2026).https://www.hinckleyallen.com/publications/2025-year-in-review-and-predictions-for-2026-in-the-cyber-ai-and-privacy-frontier/

4. **Practical DevSecOps** – "AI Security Statistics 2026: Latest Data, Trends & Research Report" (March 2026).https://www.practical-devsecops.com/ai-security-statistics-2026-research-report/

5. **Botpress** – "Chatbot Security Guide: Risks & Guardrails (2026)" (January 2026).https://botpress.com/blog/chatbot-security

6. OWASP / Trend Micro – "What are the OWASP Top 10 Risks for LLMs?" (2026 overview).https://www.trendmicro.com/en/what-is/ai/owasp-top-10.html

7. **Aembit** – "The OWASP Top 10 for LLM Applications (2025): Explained Simply" (March 2026).https://aembit.io/blog/owasp-top-10-llm-risks-explained/

8. **Malwarebytes** – "AI Chat App Leak Exposes 300 Million Messages Tied to 25 Million Users" (February 2026).https://www.malwarebytes.com/blog/news/2026/02/ai-chat-app-leak-exposes-300-million-messages-tied-to-25-million-users

9. **IBM Prompt Injection overview** – "What Is a Prompt Injection Attack?" (2024, still relevant in 2025–2026).https://www.ibm.com/think/topics/prompt-injection

10. **Stanford News** – "Study exposes privacy risks of AI chatbot conversations" (October 2025).https://news.stanford.edu/stories/2025/10/ai-chatbot-privacy-concerns-risks-research

11. **Safe Security / similar resources on generative AI chatbot risks**.https://safe.security/resources/insights/generative-ai-chatbots-cybersecurity-risks/

12. **Example GitHub pentesting repo** – "AI Chatbot Security Vulnerability Demonstration."https://github.com/Ymabbas11/AIChatbotPentesting

13. **OWASP and related agentic-AI guidance.**

- **OWASP Top 10 for Agentic Applications:** https://genai.owasp.org/resource/owasp-top-10-for-agentic-applications-for-2026/
- **CyberArk "AI agents and identity risks:** How security will shift in 2026": https://www.cyberark.com/resources/blog/ai-agents-and-identity-risks-how-security-will-shift-in-2026

14. **Egnyte** – "AI Chatbot Security: Understanding Key Risks and Testing Best Practices" (Dec 2025)Supports: shadow AI, ungoverned tools, and enterprise security/testing practices (Patterns 3 & 4).https://www.egnyte.com/blog/post/ai-chatbot-security-understanding-key-risks-and-testing-best-practices

15. **LayerX** – "AI Chatbot Security: Risks and Vulnerabilities Explained" (2025)Supports: high-level description of chatbot vulnerabilities and enterprise risk patterns.https://layerxsecurity.com/learn/chatbot-security/

16. **SupportGPT** / similar enterprise guide – "Chatbots for Enterprise: The 2026 Practical Guide"Supports: distinction between consumer chatbots vs. enterprise-grade platforms, and governance needs.https://supportgpt.app/blog/chatbots-for-enterprise

XII. About Paramount

Paramount Software Solutions is a 25+ year technology firm with a dedicated AI Center of Excellence that builds domain-specific, scalable, and secure AI solutions. From strategy and PoCs to full-scale deployment across industries like supply chain, logistics, healthcare, agriculture, and urban planning. With proven AI products (such as multi-agent platforms and geospatial AI) and modular, microservices-based architectures, Paramount helps organizations turn AI ideas into production-ready, governed systems that deliver measurable ROI.

To explore AI consulting, secure chatbot design, or a focused 6-week AI pilot, visit the [Paramount website and connect with the AI CoE team.]

Follow Us