

AI Jailbreak

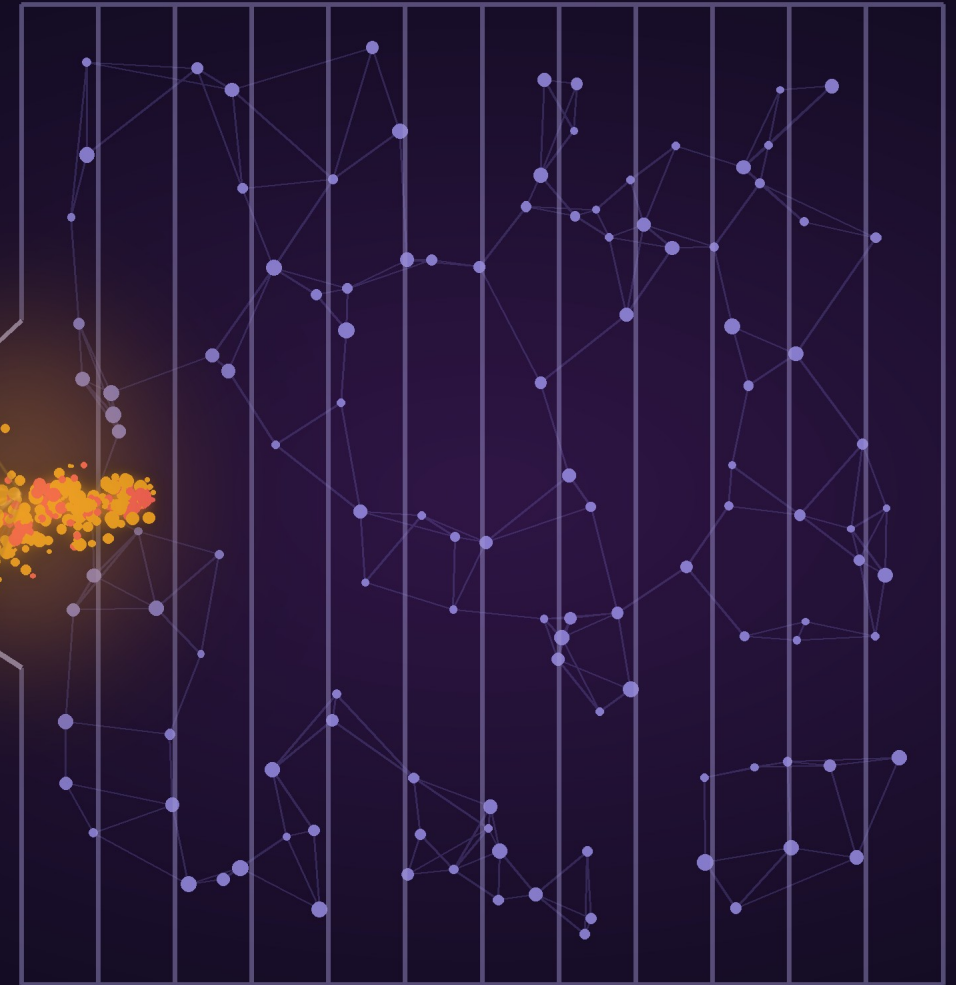
From the iPhone to language models: mechanisms, attacks, defenses, and governance

Aridio Silva

Independent Researcher

Creator of SGAEIA

Data Science and Technology • GRC and Information Security



What is AI jailbreaking

Definition

Any technique that gets a language model (LLM) to produce a response that its developers trained or configured it to refuse.

The term was born in 2007, when hackers "broke the iPhone out of jail" to install unauthorized software. With ChatGPT (2022), it came to describe prompts that make the AI ignore its safety rules.

This study walks through 7 steps, from fundamentals to continuous operation, with a focus on GRC.

1

No master key

On a phone, jailbreaking exploits code flaws. In AI, the weapon is language itself.

2

Probabilistic control

Trained safety lowers the risk of a harmful response but does not eliminate it.

3

Risk No. 1

Prompt injection is LLM01 in the OWASP Top 10 for LLM applications (2025).

From the iPhone to AI: the origin of jailbreaking

SMARTPHONES

2007

Weeks after launch, the iPhone Dev Team opens up the file system; in August, George Hotz performs the first carrier unlock.

2008

Jay Freeman (saurik) launches Cydia, an alternative app store for jailbroken devices.

2010

The U.S. Library of Congress, on the Copyright Office's recommendation, exempts smartphone jailbreaking from the DMCA.

2019

The checkm8 flaw in the bootrom of several iPhones cannot be fixed by a software update.

ARTIFICIAL INTELLIGENCE

2022

Simon Willison coins "prompt injection" (Sept.); after ChatGPT (Nov.), the "DAN" prompts appear.

2023

Wei et al. explain the failures; GCG automates attacks; Greshake et al. define indirect injection.

2024

Attacks escalate: many-shot, Crescendo, and Best-of-N; HarmBench and JailbreakBench appear.

2025

Defenses via classifiers and architecture: Constitutional Classifiers (Anthropic) and CaMeL (Google DeepMind).

1

STEP 1/7 · Fundamentals: why it works

OBJECTIVE

Understand the root cause of LLM security failures before attacking or defending.

WHY

Without the root cause, defenses treat symptoms: every point fix is bypassed by the next attack variation.

INPUTS

- LLM architecture and training phases
- Alignment techniques: RLHF, Constitutional AI
- Studies: Wei 2023, Qi 2025, Arditì 2024



WHAT IS DONE

- Analyze how alignment builds on pretraining
- Identify the 4 failure mechanisms
- Treat security as a probabilistic control



OUTPUTS

- 4 failure hypotheses: competing objectives, incomplete generalization, shallow alignment, refusal as a single direction

IMPORTANCE It is the foundation of the study: dangerous knowledge remains in the model, and trained safety reduces the risk but does not eliminate it.

2

STEP 2/7 · Taxonomy of attacks

OBJECTIVE

Organize known attacks by the attacker's level of access and by attack vector.

WHY

You cannot defend against what you do not know: the taxonomy guides threat modeling and control prioritization.

INPUTS

- Failure mechanisms (Step 1)
- Catalogs of real-world prompts (Shen et al., 2024)
- Papers: GCG, PAIR, TAP, many-shot



WHAT IS DONE

- Classify access: black-box, gray-box, or white-box
- Group the vectors into 8 families
- Link each family to the mechanism it exploits



OUTPUTS

- Attack taxonomy
- Access × vector map
- Scenarios for red teaming

IMPORTANCE Reveals the evolution from the handcrafted (prompts on forums) to the industrial (automated, scalable attacks), which changes the attacker profile.

3

STEP 3/7 · Prompt injection and agents

OBJECTIVE

Assess the risk when the AI reads third-party content and acts on real systems.

WHY

In agents, the victim stops being the vendor's policy and becomes the user and the organization: data leaks and improper actions occur.

INPUTS

- Attack taxonomy (Step 2)
- Inventory of agents, tools, and permissions
- External sources: web, email, attachments



WHAT IS DONE

- Distinguish jailbreak from indirect injection
- Map data and instruction flows
- Apply the lethal trifecta test



OUTPUTS

- High-risk agents
- Data exfiltration scenarios
- Secure architecture requirements

IMPORTANCE It is the most serious risk for businesses and still has no solution equivalent to the parameterized queries used against SQL injection.

4

STEP 4/7 · Layered defenses**OBJECTIVE**

Reduce the chance of attack success and limit the damage when attacks happen.

WHY

No single defense solves it: each layer covers the failures of the others (defense in depth).

INPUTS

- Prioritized risks (Steps 2 and 3)
- Defense techniques from the literature
- Utility and cost requirements

**WHAT IS DONE**

- Harden the model: adversarial training
- Filter input and output with classifiers
- Isolate untrusted data (CaMeL)
- Operate: monitor, limit, ban

**OUTPUTS**

- 4-layer defense architecture
- Controls prioritized by risk
- Deployment plan

IMPORTANCE Swaps the question "how do we stop every attack?" for "what can a successful attack do?", and that one has an answer.

5

STEP 5/7 · Evaluation and measurement**OBJECTIVE**

Measure the system's real robustness with metrics that are comparable over time.

WHY

Defenses tested only against fixed attacks look better than they are, because real attackers adapt.

INPUTS

- System with defenses (Step 4)
- Benchmarks: HarmBench, JailbreakBench, StrongREJECT, AgentDojo
- Red teaming team

**WHAT IS DONE**

- Measure ASR under fixed and adaptive attacks
- Measure unwarranted refusals (over-refusal)
- Measure cost: latency and compute

**OUTPUTS**

- Robustness report
- Gaps to fix (back to Step 4)
- Baseline for comparison

IMPORTANCE Without measurement, security becomes opinion. Measuring balances protection and utility and avoids a model that refuses everything.

6

STEP 6/7 · Governance and GRC

OBJECTIVE

Integrate jailbreak risk into risk management and compliance programs.

WHY

Regulators already demand it: since August 2025, the EU AI Act (Art. 55) requires adversarial testing of general-purpose models with systemic risk.

INPUTS

- Evaluation results (Step 5)
- OWASP LLM, MITRE ATLAS, NIST AI RMF, ISO/IEC 42001
- The organization's risk appetite

**WHAT IS DONE**

- Inventory and classify AI systems
- Map risks to controls and frameworks
- Define owners and evidence

**OUTPUTS**

- Controls matrix
- Policies and procedures
- Audit evidence

IMPORTANCE Turns technical findings into traceable business decisions, with an owner, a deadline, and evidence.

7

STEP 7/7 · Continuous operation**OBJECTIVE**

Keep protection effective over time in the face of new attacks.

WHY

It is an arms race: new techniques keep emerging, and today's robustness does not guarantee tomorrow's.

INPUTS

- Prompt and response logs
- Alerts and reports (bug bounty)
- New attack publications

**WHAT IS DONE**

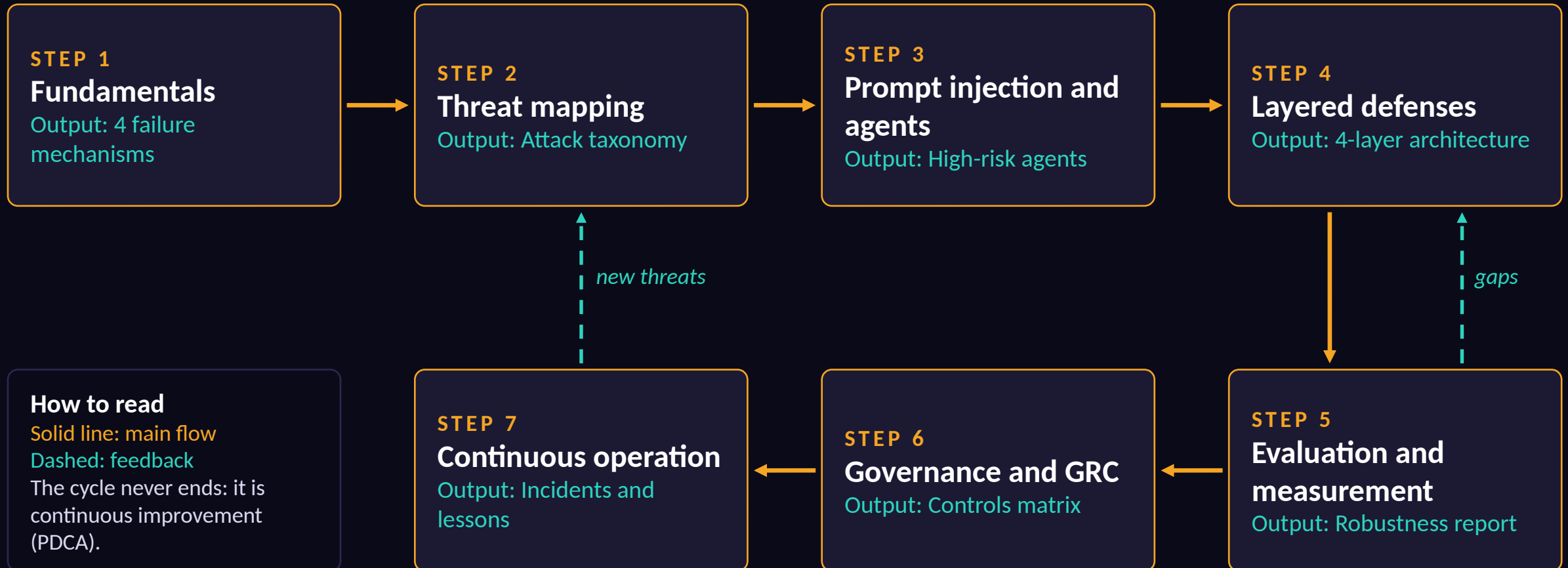
- Monitor abuse patterns
- Treat jailbreaks as security incidents
- Analyze the root cause and update defenses

**OUTPUTS**

- Incidents handled and documented
- Lessons learned
- Updates (back to Steps 2 and 4)

IMPORTANCE Closes the loop: the study stops being a one-off project and becomes a continuous improvement process, in the spirit of PDCA.

Study workflow: the 7 steps in sequence



Attack techniques: 8 families

1 Social engineering

Fictional personas, false authority, and hypothetical or literary framing.

Shen et al., 2024

2 Obfuscation

The request is hidden in encodings, ciphers, or underrepresented languages.

Yong et al., 2023

3 Multi-turn

Gradual escalation, with no explicit harmful request in any single message.

Russinovich et al., 2025

4 Context saturation

Many fake "compliance" dialogues; effectiveness grows with volume.

Anil et al., 2024

5 Gradient (GCG)

An algorithm generates suffixes with no apparent meaning, transferable across models.

Zou et al., 2023

6 Attacker LLM

One model generates and refines attempts against the target, with no human intervention.

Chao et al., 2023; Mehrotra et al., 2024

7 Mass sampling

Many random variations of the request until one gets through (Best-of-N).

Hughes et al., 2025

8 Multimodal

Adversarial instructions or perturbations embedded in images and audio.

Qi et al., 2024

Defense techniques: 4 layers

1. Model

make the model more resistant

Adversarial training · Instruction hierarchy (Wallace et al., 2024) · Circuit breakers (Zou et al., 2024) · Deep alignment (Qi et al., 2025)

2. Filtering

detect outside the model

Llama Guard (Inan et al., 2023) · Constitutional Classifiers (Sharma et al., 2025) · Perplexity filter (Jain et al., 2023) · SmoothLLM (Robey et al., 2025)

3. Architecture

limit the possible damage

Spotlighting (Hines et al., 2024) · Dual LLM (Willison, 2023) · CaMeL (Debenedetti et al., 2025) · Least privilege and human confirmation

4. Operations

detect and respond

Monitoring and alerts · Rate limiting and account banning · Continuous red teaming · Bug bounty

Controls matrix for GRC

Risk	Recommended control	OWASP LLM 2025	MITRE ATLAS	NIST AI RMF	ISO/IEC 42001
Direct jailbreak	Input and output classifiers; usage policy; abuse monitoring	LLM01	AML.T0054	MEASURE 2.7; MANAGE	A.6.2.6
Indirect prompt injection	Separate instructions from data (spotlighting, Dual LLM, CaMeL); sanitize external content	LLM01	AML.T0051.001	MAP; MEASURE 2.7	A.6.2.4
Agent with lethal trifecta	Least privilege; allowed destinations; human confirmation for sensitive actions	LLM06; LLM02	AML.T0057	MAP; MANAGE	A.5.2; A.9
System prompt leakage	No secrets or authorization rules in the prompt; controls outside the model	LLM07	AML.T0056	MEASURE	A.6.2.4
Harmful downstream output	Treat LLM output as untrusted input; validate and encode	LLM05	—	MANAGE	A.6.2.4
Undetected attack	Prompt and response logs; alerts; continuous red teaming; incident response	LLM01	AML.T0054	MANAGE 4.1	A.6.2.8; A.8.4

Codes verified in Oct 2026 against OWASP 2025, MITRE ATLAS, NIST AI RMF 1.0, and ISO/IEC 42001 (via secondary sources).

Conclusion

AI jailbreaking is not an isolated defect but a structural property of current LLMs.

Trained safety is probabilistic, and mixing instructions and data in the same channel keeps the door open to manipulation.

1

Assume the attack

Design systems knowing that some jailbreak will work.

2

Limit the damage

Architecture decides what a successful attack can do.

3

Always measure

Robustness and utility, against fixed and adaptive attacks.

4

Govern with rigor

OWASP, ATLAS, NIST, and ISO 42001 already cover the topic: apply them.

OPEN QUESTIONS

Responsible disclosure of attacks

Release of open-weight models

Who defines what is harmful

Is total robustness possible?

Glossary of technical terms (1/2)

AI agent

An LLM that uses tools and takes actions (reading email, browsing, sending data) on the user's behalf.

Alignment

Training that teaches the model to be helpful, honest, and to refuse harmful requests.

ASR (Attack Success Rate)

Share of attack attempts that produce a harmful response.

Black-box / white-box

Attack through the conversation only (black) or with access to the model's weights (white).

Bootrom

Code burned into the chip that boots the device; flaws in it cannot be fixed by an update.

Constitutional AI

Anthropic's method that aligns the model from a written set of principles.

Context window

All the text the model considers at once: instructions, conversation, and documents.

Defense in depth

Independent layers of control, so that the failure of one does not compromise everything.

Exploit

A technique or code that takes advantage of a vulnerability to gain unauthorized access.

Jailbreak

Bypassing the restrictions imposed by a system's manufacturer or developer.

Glossary of technical terms (2/2)

Lethal trifecta

Private data + untrusted content + external communication: critical leak risk.

LLM

Large language model, trained to predict text from enormous volumes of data.

Over-refusal

Unwarranted refusal: the model denies a legitimate request out of excess caution.

Prompt injection

Malicious instructions in the processed text; indirect when they come from third-party content.

Red teaming

Simulated, authorized attacks to find flaws before criminals do.

RLHF

Reinforcement learning from human feedback: people rate responses and the model learns from them.

Root

Maximum administrator privilege; on Android, the equivalent of jailbreaking.

Safety classifier

Auxiliary model that analyzes input and output to block harmful content.

Sandbox

Isolation that prevents an app or process from accessing the rest of the system.

System prompt

Developer instructions that define the model's rules and behavior in the application.

Verified bibliography (1/2)

ORIGINS IN SMARTPHONES

- MILLER, C. et al. iOS Hacker's Handbook. Wiley, 2012.
- LEVIN, J. *OS Internals, v. III: Security & Insecurity. Technogeeks, 2016 (2nd ed. 2019).
- U.S. COPYRIGHT OFFICE. Section 1201 Rulemaking: exemption for smartphone jailbreaking. 2010.

FOUNDATIONS

- WEI, A.; HAGHTALAB, N.; STEINHARDT, J. Jailbroken: How Does LLM Safety Training Fail? NeurIPS, 2023.
- QI, X. et al. Fine-tuning Aligned Language Models Compromises Safety. ICLR, 2024.
- QI, X. et al. Safety Alignment Should Be Made More Than Just a Few Tokens Deep. ICLR, 2025.
- ARDITI, A. et al. Refusal in Language Models Is Mediated by a Single Direction. NeurIPS, 2024.

ATTACKS

- PEREZ, F.; RIBEIRO, I. Ignore Previous Prompt. NeurIPS ML Safety Workshop, 2022.
- GRESHAKE, K. et al. Not What You've Signed Up For (indirect prompt injection). ACM AISec, 2023.
- ZOU, A. et al. Universal and Transferable Adversarial Attacks on Aligned LLMs. arXiv, 2023.
- CHAO, P. et al. Jailbreaking Black Box LLMs in Twenty Queries (PAIR). arXiv:2310.08419, 2023.
- YONG, Z.-X. et al. Low-Resource Languages Jailbreak GPT-4. NeurIPS Workshop SoLaR, 2023.
- SHEN, X. et al. "Do Anything Now": In-The-Wild Jailbreak Prompts. ACM CCS, 2024.
- ANIL, C. et al. Many-shot Jailbreaking. NeurIPS, 2024.
- MEHROTRA, A. et al. Tree of Attacks. NeurIPS, 2024.
- RUSSINOVICH, M. et al. The Crescendo Multi-Turn LLM Jailbreak Attack. USENIX Security, 2025.
- HUGHES, J. et al. Best-of-N Jailbreaking. NeurIPS, 2025.
- QI, X. et al. Visual Adversarial Examples Jailbreak Aligned LLMs. AAAI, 2024.

Verified bibliography (2/2)

DEFENSES

- INAN, H. et al. Llama Guard. Meta, 2023.
- JAIN, N. et al. Baseline Defenses for Adversarial Attacks Against Aligned LLMs. arXiv, 2023.
- ROBEY, A. et al. SmoothLLM. TMLR, 2025.
- WALLACE, E. et al. The Instruction Hierarchy. OpenAI, 2024.
- HINES, K. et al. Defending Against Indirect Prompt Injection With Spotlighting. Microsoft, 2024.
- ZOU, A. et al. Improving Alignment and Robustness with Circuit Breakers. NeurIPS, 2024.
- SHARMA, M. et al. Constitutional Classifiers. Anthropic, arXiv:2501.18837, 2025.
- DEBENEDETTI, E. et al. Defeating Prompt Injections by Design (CaMeL). arXiv:2503.18813, 2025.
- WILLISON, S. The Dual LLM pattern (Apr. 25, 2023); The lethal trifecta for AI agents (Jun. 16, 2025). simonwillison.net.

EVALUATION

- MAZEIKA, M. et al. HarmBench. ICML, 2024.
- CHAO, P. et al. JailbreakBench. NeurIPS, 2024.
- SOULY, A. et al. A StrongREJECT for Empty Jailbreaks. NeurIPS, 2024.
- DEBENEDETTI, E. et al. AgentDojo. NeurIPS, 2024.
- SCHULHOFF, S. et al. Ignore This Title and HackAPrompt. EMNLP, 2023.

GOVERNANCE

- OWASP. Top 10 for LLM Applications, 2025 version (Nov. 18, 2024).
- MITRE. ATLAS: Adversarial Threat Landscape for AI Systems. atlas.mitre.org.
- NIST. AI 100-1 (AI RMF 1.0), 2023; AI 600-1, 2024; AI 100-2e2025 (Mar. 2025).
- ISO/IEC 42001:2023. AI Management System.
- EUROPEAN UNION. Regulation (EU) 2024/1689 (AI Act). 2024.