AI Red Teaming in Healthcare: Testing Clinical and Patient-Facing AI for Safety and Security

A practical guide to red teaming healthcare AI systems for privacy, security, and safety risks.

Prepared by Jacobian Engineering | February 9, 2026

This guide is for informational purposes only and does not constitute legal advice.

Executive Summary

AI is moving quickly into healthcare workflows. Patient-facing chatbots answer questions. Clinical decision support tools help clinicians prioritize care. Revenue cycle teams use models to detect anomalies and improve billing accuracy. These systems can improve outcomes, but they also introduce new failure modes. Traditional security testing does not fully cover model behavior, data leakage risks, and unsafe outputs.

AI red teaming is a structured testing approach that looks for ways an AI system can fail, be misused, or be manipulated. In healthcare, failures can affect privacy, safety, and trust. This guide explains how to plan and execute AI red teaming for clinical and patient-facing systems, including how to test for data leakage, prompt injection, bias, and operational weaknesses. If your AI system behaved unexpectedly tomorrow, would you have logs and controls that let you understand what happened and limit impact?

What AI red teaming means in healthcare

Red teaming is often associated with penetration testing, where a tester tries to break into a system. AI red teaming is related, but the target is broader. You test the AI model, the prompts, the integrations, the data pipeline, and the surrounding controls. The goal is to discover ways the system can produce unsafe outputs, leak sensitive data, or be manipulated by users or attackers.

In healthcare, "unsafe" is not only a security concern. It includes patient harm, clinical misinformation, inappropriate recommendations, and biased outcomes that affect care. That is why red teaming should involve clinical stakeholders, not only engineering and security teams.

Red teaming vs model evaluation

Many teams already run model evaluation for accuracy. Red teaming is different. It focuses on adversarial behavior and misuse. It asks how the system behaves when users are confused, malicious, or persistent. It also asks how the surrounding system behaves when the model produces unexpected output.

Common healthcare AI use cases that benefit from red teaming

- **Patient support and navigation:** Chatbots that answer benefit questions, appointment questions, or medication guidance.
- **Clinical decision support:** Tools that summarize records, suggest diagnoses, or highlight abnormal patterns.
- **Operational automation:** Models that classify claims, route tickets, or extract information from documents.
- **Medical imaging assistance:** AI that supports radiology or pathology review workflows.
- **Provider productivity tools:** Note summarization, coding suggestions, and documentation support.

Threats and failure modes unique to healthcare AI

AI systems introduce risks that traditional applications do not. Red teaming makes these risks concrete by turning them into test cases. What could a user do that you did not anticipate? What could a malicious actor do through your AI interface?

Data leakage and privacy exposure

Healthcare AI systems often touch sensitive data such as PHI and clinical notes. Leakage can happen through prompts, through integrations, through logs, or through model behavior. Red teaming tests whether the system can be coaxed into revealing data it should not reveal. It also tests whether staff can accidentally expose PHI by pasting sensitive content into tools that store prompts for training or analytics.

Prompt injection and tool abuse

Many healthcare AI systems use tool integrations such as search, ticketing, or EHR APIs. Prompt injection attacks try to override instructions and manipulate tool calls. If your AI can access internal systems, can a user trick it into pulling data from the wrong record or exposing internal information?

Hallucinations and unsafe recommendations

Generative models can produce confident but incorrect answers. In healthcare, that can create safety risk. Red teaming should test how the system responds when it does not know something, when a user asks for medical advice outside intended scope, or when data is incomplete. The goal

is not perfect answers. The goal is safe behavior under uncertainty.

Bias and unequal outcomes

Healthcare models can produce biased outputs if training data reflects historical inequities or if evaluation is incomplete. Red teaming can include bias testing to identify whether recommendations or classifications differ across demographic groups in ways that are not clinically justified. Bias testing also includes checking whether language or tone changes in ways that may alienate or harm users.

Supply chain and configuration risk

AI systems often depend on third-party model providers, vector databases, prompt libraries, and plugin ecosystems. Changes in any of those dependencies can change model behavior. Red teaming should include tests for dependency misuse and misconfiguration, such as overly broad tool permissions or unsafe default settings.

Planning a healthcare AI red team

Red teaming works best when it is planned like a project. Define scope, success criteria, and safety boundaries. Decide what data is allowed in the test environment and how findings will be handled. In healthcare, it is often appropriate to use de-identified or synthetic data for testing where possible.

Define the system boundary

- **Model and prompts:** Which model is used and what instructions govern it.
- **Data sources:** What data the model can access and how access is controlled.
- **Integrations:** Tools, APIs, and services the AI can call.
- **User roles:** Who can use the system and what permissions they have.
- **Logging and monitoring:** What telemetry exists to investigate issues.
- **Human oversight:** Where clinicians or reviewers approve or override output.

Choose the right testing environment

Testing in production is rarely appropriate for healthcare AI. A safer pattern is a staged environment with controlled access and carefully selected data. If de-identified data is used, define the de-identification method and verify the test set does not contain hidden identifiers. If synthetic data is used, ensure it still exercises realistic edge cases.

When live PHI must be involved, restrict access, log all activity, and define retention for prompts and outputs. Treat the test environment like a regulated system, not a sandbox.

Define test objectives

- **Privacy objectives:** Confirm the system does not expose PHI through outputs or logs.
- **Security objectives:** Confirm the system cannot be manipulated to access unauthorized data or actions.
- **Safety objectives:** Confirm the system refuses or escalates when asked for unsafe medical guidance.
- **Reliability objectives:** Confirm the system behaves predictably under stress and unusual inputs.
- **Compliance objectives:** Confirm evidence exists to support due diligence, such as audit logs and access controls.

AI red teaming techniques for healthcare systems

Adversarial prompting and jailbreak testing

Test whether users can override system instructions. This includes attempts to reveal system prompts, bypass guardrails, or elicit restricted content. In healthcare, jailbreak testing should include attempts to obtain medical advice, dosage recommendations, or diagnoses when the system is not intended for that purpose.

PHI extraction tests

Test whether the system can be manipulated to reveal PHI from its context window, connected tools, or logs. This includes attempts to access other patient records, infer identifiers, or leak data through summarization tasks. The test should also examine whether the system echoes input content when it should summarize or redact.

Tool misuse and authorization testing

If the AI can call tools, test whether a user can influence tool calls. For example, can a user prompt the AI to search for a different patient record, download an attachment, or query an internal database? Tool misuse is one of the highest risk areas in AI systems because it bridges model behavior with real actions.

Authorization testing should also validate server-side controls. AI guardrails cannot be the only protection. Sensitive actions should require server-side permission checks and, for some workflows, explicit human approval.

Bias and fairness evaluation

Bias testing requires careful design. Define clinically relevant outcomes, select representative test cases, and evaluate whether outputs differ in problematic ways. The goal is not to demand identical outcomes in all cases. The goal is to detect patterns that are not clinically justified and that could create harm or unequal access.

Robustness and stress testing

Test how the system behaves under load, with malformed inputs, or with ambiguous context. Healthcare AI systems should fail safely. That means refusing or escalating rather than guessing when uncertainty is high. Robustness testing also includes ensuring the model does not degrade into unsafe behavior when context windows are long or when users paste unstructured records.

Prompt and output logging review

Even when a model behaves safely, logs can create exposure if they capture sensitive data without controls. Red team exercises should include a review of how prompts, outputs, and tool calls are stored. Who can access logs? How long are they retained? Are they included in analytics? These questions are often overlooked until a partner asks.

Hardening and operational controls after findings

Red teaming is useful only if findings lead to improvements. In healthcare, improvements often involve both model changes and system controls. Control improvements may include tightening access, adding approval steps for sensitive actions, improving logging, and adding monitoring for

abnormal use.

- **Access controls:** Limit who can use the system and what data it can access.
- **Guardrails:** Implement content filters, refusal patterns, and escalation paths.
- **Human oversight:** Require human review for high-impact outputs.
- **Logging:** Log prompts, tool calls, and outputs in a way that supports investigations without exposing PHI unnecessarily.
- **Monitoring:** Monitor for prompt injection attempts, data extraction patterns, and abuse signals.
- **Change control:** Treat prompt and model updates as changes that require review, testing, and rollback plans.

Documenting results for trust and accountability

Healthcare organizations rarely red team for curiosity alone. They red team to reduce risk and to prove discipline to partners. A clear red team report helps internal teams prioritize work and helps external reviewers understand your approach.

Useful reports typically include scope, test objectives, test cases used, findings with severity, remediation actions, and retest results. They also include operational recommendations such as logging improvements and monitoring rules. Documentation is also the bridge between AI testing and broader security programs like HIPAA safeguards and HITRUST control requirements.

> *"The most valuable outcome of AI red teaming is not a list of clever attacks. It is a safer release process and better visibility into how the system behaves in production." - Jacobian Engineering AI Security Team*

Implementation Methodology

Phase 1: Scoping and test plan design

Define use cases, system boundaries, and safety constraints. Identify which data is allowed for testing and how results will be stored and reviewed. Build a threat model that includes privacy, security, and safety risks. Agree on who signs off on fixes and what the release gate is.

Phase 2: Execute red team testing and remediation

Run adversarial tests, document findings, and prioritize remediation. Validate fixes through retesting. If findings involve third-party models or tools, define compensating controls such as tighter permissions and stronger monitoring. Ensure remediation includes both model-side and system-side controls.

Phase 3: Ongoing monitoring and continuous testing

AI systems change over time as prompts, models, and integrations evolve. Establish a cadence for red team retesting, monitoring, and incident response. Treat AI red teaming as part of the release lifecycle, not a one-time project. As new features are added, update the test library so coverage grows with the system.

Common pitfalls in healthcare AI testing

- **Testing only the model:** Many failures happen in integrations and access control, not in the model alone.
- **No clinical involvement:** Safety criteria require clinical context, especially for patient-facing systems.
- **Uncontrolled prompt storage:** Prompts and outputs are retained indefinitely without access controls.
- **Guardrails without server-side authorization:** The model is asked to enforce permissions it cannot reliably enforce.
- **One-time testing:** Model and prompt updates change behavior, so testing must be repeated.

Business benefits of AI red teaming in healthcare

- **Reduced privacy risk:** Testing for leakage and access flaws reduces the likelihood of PHI exposure.
- **Safer deployments:** Guardrails and evaluation reduce unsafe outputs.
- **Higher partner confidence:** Demonstrating disciplined testing supports due diligence and procurement reviews.
- **Better operational readiness:** Logging and monitoring improvements help teams investigate issues quickly.

Frequently Asked Questions

Is AI red teaming the same as penetration testing?

They overlap, but they are not the same. Penetration testing focuses on application and infrastructure vulnerabilities. AI red teaming focuses on model behavior, prompt injection, data leakage, and misuse of AI integrations. Many healthcare organizations benefit from both.

How often should we red team an AI system?

At a minimum, test before initial production release and after major changes such as model upgrades, new tools, or new data sources. Many teams adopt a periodic cadence, especially for high-impact clinical or patient-facing systems.

Can we red team a third-party model?

You may not be able to change the underlying model, but you can still test your system integration, prompts, access controls, and guardrails. Red teaming often focuses on the combined system, not only the model.

How do we avoid exposing PHI during testing?

Prefer de-identified or synthetic data when feasible. Restrict access to test environments, log activity, and define retention for prompts and outputs. If PHI must be involved, treat test assets with the same access and monitoring controls you would use in production.

How can Jacobian Engineering help?

Jacobian Engineering provides AI red teaming services that evaluate model behavior, integrations, data leakage risk, and operational controls. The team also offers penetration testing for web apps, mobile apps, and APIs, along with cloud security architecture and monitoring services that help healthcare organizations deploy AI systems more safely.

Conclusion

AI red teaming turns abstract AI risks into concrete test cases and measurable improvements. In healthcare, that discipline protects privacy, supports patient safety, and builds trust with partners. Define the system boundary, test for data leakage and misuse, fix findings, and monitor continuously as the system evolves.

If you want help scoping a healthcare AI red team, designing test cases, or integrating red teaming into your release lifecycle, Jacobian Engineering can help you build a testing program that supports safe AI deployment.