aikido

When Is AI Pentesting Safe?

Minimum Safety Requirements for Autonomous Security Testing

Why this document exists

AI pentesting introduces automated offensive capability against live systems using AI agents. Unlike traditional security tools, these systems operate autonomously, execute real actions, and adapt based on responses, creating a fundamentally different risk profile.

This document establishes the **safety baseline** for AI pentesting. It defines the minimum enforceable technical requirements that must be met before autonomous security testing systems can be operated responsibly. These requirements are vendor-neutral and represent the baseline for safe deployment.

Security testing is one of the first domains where AI operates autonomously in adversarial, production-like environments. While organizations such as **OWASP** have documented the risks of agentic AI broadly, this document defines the minimum safety standard for one of its most sensitive applications.

Executive summary

- AI pentesting systems act autonomously against live applications and infrastructure
- Without technical safeguards, they introduce loss of control, misuse, and unintended impact
- This document defines the minimum safety requirements for operating such systems responsibly
- These requirements are technical, enforceable, and vendor-agnostic
- Anything below this bar is unsafe for autonomous security testing

Why AI Pentesting Requires a Higher Safety Bar

Unlike scanners or instruction-following systems, agentic AI pentesting systems:

- Make autonomous decisions
- Execute real tools and commands
- Interact with live systems
- Adapt behavior based on feedback

Instruction-following alone is insufficient. Safety must be enforced technically, across multiple layers, and independently of agent behavior.

Minimum safety requirements

These requirements assume systems operating at scale, with multiple concurrent agents, real network traffic, and continuous execution rather than single, point-in-time tests.

These are minimum requirements for operating safely at scale.

1. Abuse prevention and ownership validation

Agentic pentesting systems must ensure they are used only against assets the operator owns or is explicitly authorized to test.

At a minimum:

- Target ownership must be verified before testing begins
- Authorization must be enforced technically, not through user declarations alone

In Aikido Attack, ownership is validated through explicit verification steps such as DNS records or static files hosted on the target. This ensures the platform can only be used for authorized defensive testing.

2. Enforced scope control at the network level

Agentic systems must not rely on prompts or instructions to remain in scope.

Minimum requirements include:

- Programmatic inspection of every outbound request
- Hard enforcement of approved targets
- Automatic blocking of all non-authorized destinations

Scope violations must be prevented by design, not detected after the fact.

In worst-case scenarios, execution must remain fully contained.

3. Isolation between reasoning and execution

Agentic pentesting systems execute real tools, which introduces execution risk.

Minimum requirements include:

- Strict separation between agent reasoning and tool execution
- Sandboxed execution environments
- Isolation between agents and between customers

In worst-case scenarios, execution must remain fully contained.

4. Full observability and emergency controls

Agentic systems must not operate as black boxes.

Operators must be able to:

- Inspect every action taken by agents
- Monitor behavior in real time
- Immediately halt all activity

Emergency stop mechanisms are a baseline requirement, not an optional safeguard.

5. Data residency and processing guarantees

Agentic pentesting systems often handle sensitive application data.

Minimum requirements include:

- Clear guarantees on where data is processed and stored
- Regional isolation where required
- No cross-region data leakage by default

6. Prompt injection containment

Any agent interacting with untrusted application content must be assumed vulnerable to prompt injection.

Minimum requirements include:

- Restricting agent access to arbitrary third-party data sources
- Preventing outbound data exfiltration paths
- Isolating agent execution so injected instructions cannot escape scope

Prompt injection should be expected and contained, not treated as an edge case.

What this does not promise

No agentic AI pentesting system is perfect.

Such systems will:

- Miss some issues
- Occasionally misinterpret behavior
- Require validation
- Benefit from human oversight

The goal is not perfection.

The goal is to surface materially exploitable risk faster, more safely, and at greater scale than existing models.

Why this matters now

Agentic AI pentesting is moving from theory to practice.

Without shared minimum standards, the industry risks unsafe automation, misleading claims, and erosion of trust in a powerful new capability. Establishing a clear safety baseline is a prerequisite for responsible adoption.

These requirements represent the minimum bar.

Anything less is not safe AI pentesting.

Final note

This document is intentionally vendor-neutral.

It defines what safe agentic AI pentesting must look like, regardless of implementation. Operators, vendors, and buyers should use these requirements as a baseline for evaluation and accountability.

As autonomous systems become more common across software delivery, finance, and operations, the controls defined here are likely to become relevant well beyond security testing.