Agentic AI Security: Applying the OWASP Top 10:2025

Protect your Agentic implementation from the most common vulnerabilities.

REMOTEWINNERS.COM

Anjana Silva

1

Broken Access Control

Risk: Agent deletes customer database after a prompt indirectly triggers an unconstrained administrative tool.

Scenario: An agent interprets a "cleanup" request by accessing an admin-level API it shouldn't have permission to use.

Mitigation: Deny access by default, enforce PoLP, and ensure agents strictly inherit the specific permissions of the active user.

REMOTE WINNERS

REMOTEWINNERS.COM

Anjana Silva

2

Security Misconfiguration

Risk: Agent operates with unnecessary default features enabled, such as unmonitored "admin" tools or active debugging consoles.

Scenario: An attacker accesses a sample administrative tool left enabled on the agent, gaining full control over the model's environment.

Mitigation: Disable unused tools, remove default accounts, and use automated scripts to ensure secure configurations across all environments.

REMOTE WINNERS

REMOTEWINNERS.COM

Anjana Silva

3

Software Supply Chain Failures

Risk: Agent executes malicious code embedded in an untrusted third-party library or a compromised model weight.

Scenario: A developer integrates a popular open-source agent tool that contains a hidden backdoor for data exfiltration.

Mitigation: Maintain a verified SBOM, scan all dependencies for vulnerabilities, and only source components from trusted, signed repositories.

REMOTE WINNERS

REMOTEWINNERS.COM

Anjana Silva

4

Cryptographic Failures

Risk: Agent leaks PII because it uses weak MD5 hashes or predictable random numbers for session keys.

Scenario: An attacker predicts the agent's session tokens because they were generated using a non-cryptographic, weak random number generator.

Mitigation: Use Argon2 for password hashing and ensure all random values use cryptographically secure generators (CSPRNG).

REMOTE WINNERS

REMOTEWINNERS.COM

Anjana Silva

5

Injection

Risk: Agent executes malicious shell commands or database queries because untrusted input is concatenated into its tool-calling logic.

Scenario: An attacker inputs a prompt that tricks the agent into running `*rm -rf*` via its terminal execution tool.

Mitigation: Use parameterised interfaces, enforce strict input validation, and avoid direct concatenation of agent outputs into system commands.

REMOTEWINNERS.COM

Anjana Silva

6

Insecure Design

Risk: Agent architecture lacks a "human-in-the-loop" design for irreversible business logic, enabling autonomous financial or data loss.

Scenario: An agent is designed to autonomously process refunds without a maximum threshold or secondary verification step.

Mitigation: Use threat modeling, implement secure design patterns, and integrate mandatory approval gates for high-impact autonomous actions.

REMOTE WINNERS

REMOTEWINNERS.COM

Anjana Silva

7

Authentication Failures

Risk: Agent uses hard-coded API keys or lacks multi-factor authentication for accessing sensitive enterprise services.

Scenario: An attacker exploits the agent's reliance on a single, long-lived API token to gain persistent account access.

Mitigation: Implement multi-factor authentication, use short-lived tokens, and avoid hard-coding credentials in agent configurations or prompts.

REMOTEWINNERS.COM

Anjana Silva

8

Software or Data Integrity Failures

Risk: Agent processes tampered serialised memory objects or unverified plugin updates, leading to unauthorised code execution.

Scenario: An attacker modifies an agent's serialised "long-term memory" file to inject malicious instructions during the next session.

Mitigation: Sign all data artefacts, verify plugin integrity with digital signatures, and use secure deserialisation libraries only.

REMOTE WINNERS

REMOTEWINNERS.COM

Anjana Silva

9

Security Logging & Alerting Failures

Risk: Agent's autonomous tool calls and decision-making logic are not logged, preventing detection of malicious behaviour or model drift.

Scenario: An attacker performs a prompt injection to exfiltrate data, but the agent's internal reasoning and API calls aren't recorded.

Mitigation: Log all agent tool calls, reasoning steps, and input/output failures; set real-time alerts for suspicious autonomous activity patterns.

REMOTE WINNERS

REMOTEWINNERS.COM

Anjana Silva

10

Mishandling of Exceptional Conditions

Risk: Agent reveals internal system prompts or API keys in error messages when a tool execution fails.

Scenario: A tool fails due to a timeout, causing the agent to output a raw stack trace containing backend credentials.

Mitigation: Implement global exception handlers, sanitise agent outputs for sensitive data, and ensure the system always fails securely.

REMOTEWINNERS.COM

Anjana Silva

Summary

OWASP Item	Risk	Scenario	Mitigation
01 Broken Access Control	Agent triggers admin tools via unauthorised user prompts.	Agent deletes database using an unconstrained administrative API.	Enforce PoLP and user-level session scoping for tools.
02 Security Misconfiguration	Unnecessary features or debug consoles remain active in production.	Attacker controls model through an enabled sample admin tool.	Disable unused tools and use automated hardening scripts.
03 Supply Chain Failures	Agent runs malicious code from compromised third-party dependencies.	Malicious plugin exfiltrates data via a hidden backdoor.	Verify SBOMs and use only trusted, signed repositories.
04 Cryptographic Failures	Sensitive data leaked due to weak hashes or entropy.	Attacker predicts tokens from weak random number generators.	Use Argon2 hashing and secure CSPRNG for all values.
05 Injection	Untrusted input concatenated directly into executable tool commands.	Attacker runs "rm -rf" through a terminal execution tool.	Use parameterised interfaces and strict input validation logic.
06 Insecure Design	Lacks human-in-the-loop for irreversible or high-impact actions.	Agent issues massive refunds without secondary human approval.	Integrate mandatory approval gates for sensitive autonomous tasks.
07 Authentication Failures	Hard-coded keys or missing MFA for sensitive service access.	Attacker uses stolen long-lived tokens for persistent access.	Require MFA and use short-lived, dynamic authentication tokens.
08 Integrity Failures	Agent processes tampered memory or unverified plugin updates.	Attacker modifies serialised memory to inject malicious instructions.	Sign artifacts and verify integrity with digital signatures.
09 Logging/Alerting Failures	Autonomous actions are not logged, hiding malicious behavior.	Prompt injection occurs without recording the agent's internal reasoning.	Log all tool calls and alert on suspicious activity.
10 Exceptional Conditions	Error messages leak system prompts or sensitive API keys.	Stack trace reveals backend credentials after a tool timeout.	Sanitise outputs and implement secure global exception handling.

REMOTE WINNERS

REMOTEWINNERS.COM

Anjana Silva

How Many of These Vulnerabilities Exist in Your Agentic Implementation?

Don't wait for a breach to take security seriously.

I help remote/ SaaS organisations build security-first architectures from day one.

Send me a message through RemoteWinners.com/contact-us or LinkedIn for a free discovery call.

REMOTEWINNERS.COM

Anjana Silva