A.I.G (AI-Infra-Guard) Technical Report

Tencent Zhuque Lab

https://github.com/Tencent/AI-Infra-Guard

Abstract

AI agents and agentic workflows are evolving beyond single model calls into systems that retrieve knowledge, invoke tools, execute tasks, and coordinate across external systems. The security boundary has expanded accordingly: model services, AI framework components, MCP Servers, Agent Skills, tool permissions, context data, plugin dependencies, and runtime infrastructure now all participate in the attack surface. Traditional SAST, DAST, vulnerability scanning, and jailbreak evaluation cover only fragments of this risk landscape; they do not, by themselves, provide a unified way to assess vulnerable agent frameworks, poisoned tool and Agent Skills supply chains, and uncontrolled agent behavior.

A.I.G (AI-Infra-Guard) is an open-source AI red-team security assessment platform from Tencent Zhuque Lab. It provides multi-layer security self-assessment capabilities for enterprise AI systems. A.I.G V4 covers five assessment scenarios: AI Infra Scan, MCP/Skill Scan, Agent Scan, ClawScan, and Jailbreak Eval. The platform maintains rule and knowledge assets for AI infrastructure, Agent Skills, agent workflows, and model safety evaluation, helping users identify component vulnerabilities, supply-chain risks, permission-boundary issues, and agent-behavior risks.

The engineering value of A.I.G is its integration of rule-based asset identification, model-assisted semantic analysis, and agent workflow risk assessment into one platform. Deterministic risks are verified through reviewable rules. MCP, Agent Skills, and agent risks are explained through contextual analysis and evidence-based reporting. The scanning capabilities can be integrated into different workflows through the Web UI, REST API, self-hosted services, EdgeOne ClawScan, EdgeOne Skill Scanner, and the `aig-scanner` Skill. This report summarizes the architecture, capability boundaries, practical use cases, deployment paths, integration model, and operational boundaries of A.I.G based on the V4 snapshot.

This technical report is maintained as a living document and will continue to be updated as A.I.G evolves and as new external feedback arrives. Some descriptions may contain omissions, inaccuracies, or implementation drift; issues, corrections, and suggestions are welcome through the project repository or community channels.

Keywords: AI Security; Agent Security; MCP; Skill; AI Red Teaming; AI Infrastructure Scan

Introduction

AI systems are moving from model-centric applications toward AI agents and agentic workflows: systems that retrieve knowledge, invoke tools, operate external systems, and continue working across task steps. Enterprise adoption, private deployments, the MCP and Agent Skills ecosystem, and open-source model supply chains are expanding at the same time. As a result, the security boundary now spans the full engineering stack: model services, application frameworks, RAG data, agent runtimes, tool permissions, plugin dependencies, cloud resources, and continuous operations. Adoption rates, incident growth, Shadow AI, agent reliability gaps, and weak AI access controls reported by authoritative sources all point to the same conclusion: AI security is no longer a single-point jailbreak problem. It is a systems engineering problem that requires asset discovery, rule validation, semantic evaluation, and evidence-based closure.

Existing security programs do not fully cover this shift. Traditional SAST, DAST, host-security, and cloud-security tools are effective for code, interfaces, ports, and configuration checks, but they usually lack AI component fingerprints, model-framework vulnerability knowledge, MCP Server semantics, Agent Skills behavior and metadata, tool-call chain analysis, context contamination detection, and indirect prompt-injection reasoning. Model safety evaluation can measure harmful outputs and jailbreak resilience, but it usually cannot explain how a real agent workflow creates compound risk across permissions, memory, plugins, external APIs, and execution results. Security assessment for the AI engineering stack therefore needs three capabilities at once: reproducible scanning for deterministic assets and vulnerabilities, context-aware analysis of agent behavior and tool chains, and durable rules, evidence, and remediation guidance.

A.I.G (AI-Infra-Guard) is an open-source AI red-team security assessment platform from Tencent Zhuque Lab. Its goal is to provide deployable, integrable, and extensible AI security self-assessment capabilities for enterprises, research teams, and security platforms. A.I.G V4 uses a Server-Agent distributed architecture and implements five assessment engines around AI Infra Scan, MCP/Skill Scan, Agent Scan, ClawScan, and Jailbreak Eval. On one side, it identifies AI infrastructure and supply-chain risks through asset identification, vulnerability knowledge, configuration rules, and risk taxonomies. On the other side, it evaluates agentic AI

and LLM scenarios through agent-behavior assessment and model safety evaluation. The platform can be used through the Web UI, CLI, REST API, Agent Skills, Docker Compose, EdgeOne ClawScan, and EdgeOne Skill Scanner.

This report focuses on the implementation and practical use of A.I.G V4. It discusses the platform's technical design, rule assets, runtime architecture, evaluation observations, and deployment/integration model for AI engineering-stack security assessment. The technical contributions summarized in this report are:

1. A systematic AI engineering-stack security assessment framework: AI infrastructure, MCP and Agent Skills, agent workflows, the OpenClaw ecosystem, and model jailbreak evaluation are treated as one assessment space rather than reducing AI security to model-output testing.
2. A maintainable rule and knowledge-asset system: component identification, vulnerability knowledge, supply-chain risk taxonomies, agent risk patterns, and model-evaluation data are used as core assets for continuous extension and evidence-based output.
3. A stronger Agentic AI risk assessment capability: typical risks around agent goals, tools, permissions, context, and data movement are mapped to the OWASP Agentic Applications 2026 taxonomy to make findings easier to explain and review.
4. An open-source implementation for real workflows: the Server-Agent architecture, unified event stream, externalized rule data, and multiple integration paths allow A.I.G to support personal research, enterprise self-assessment, security platform integration, and general agent workflows.

1 Background and Challenges

1.1 From Model Risk to System Risk

Earlier AI security discussions focused mainly on the model itself: jailbreaks, hallucination, bias, and privacy leakage. As Skills, RAG, MCP, agent workflows, and private deployments become mainstream, the attack surface has expanded across the complete AI engineering chain. At the model and prompt layer, risks include prompt injection, sensitive information disclosure, and system-prompt leakage. At the data and knowledge-base layer, risks include data poisoning, vector/embedding weaknesses, and training/runtime data leakage. At the agent and tool layer, risks include excessive delegation, tool misuse, cross-tool data movement, and indirect injection. At the supply-chain layer, poisoned models, dependencies, plugins, Skills, and MCP Servers may introduce backdoors or vulnerable versions. At the infrastructure layer, exposed AI services, component CVEs, weak authentication, and cloud misconfigurations become entry points.

The OWASP Top 10 for Agentic Applications 2026 expands agent risk into goal hijacking, tool misuse, identity and permission abuse, agentic supply chain, unintended code execution, memory and context poisoning, cross-agent communication, cascading failures, human-agent trust exploitation, and malicious agent behavior. This taxonomy provides a direct reference point for A.I.G's agent risk categorization and evidence organization [3].

1.2 Risk Signals in Authoritative Reports

Authoritative reports provide several mutually reinforcing signals. Organizational AI adoption reached 88%; documented AI incidents rose from 233 in 2024 to 362; and AI agents improved on real computer-task benchmarks such as OSWorld from roughly 12% to roughly 66% success, while still failing about one-third of structured tasks [4]. Agentic AI risk is moving from experimentation into production: about one in eight reported AI security incidents involves agentic systems, and 31% of organizations cannot confirm whether they experienced an AI security incident in the past year. Seventy-six percent of organizations consider Shadow AI a definite or likely problem; 93% use open-weight models from public repositories, but fewer than half continuously scan inbound models [6]. Cloud and enterprise governance also show gaps: 81% of organizations use managed AI services, 90% run self-hosted models, 57% deploy self-hosted AI agents, and 80% use MCP Servers [7]. Thirteen percent of surveyed organizations reported AI model or application breaches, and 97% of those lacked proper AI access controls [5]. Only 15% of organizations reported that their network was fully ready for AI, and only 24% could enforce guardrails and real-time monitoring for agent behavior [8]. CISA, NSA, FBI, and partner agencies also emphasize that data protection, data supply chains, and tamper resistance must be incorporated throughout the AI system lifecycle [9].

1.3 Gaps in Existing Security Programs

Existing security solutions cover parts of the risk surface, but they do not directly cover the full AI engineering stack. Traditional SAST, DAST, and vulnerability scanners are strong for code, web interfaces, and infrastructure, yet they provide limited coverage of agent context, tool invocation, and model behavior. Model safety evaluation can assess jailbreaks and harmful output, but it is not designed to cover plugins, knowledge bases, MCP, agent workflows, and runtime environments. Cloud-security and host-security tools

find cloud-resource and host-level risks, but they usually lack AI component, model-asset, and AI workflow semantics. Manual review remains flexible, but the rapid growth of models, applications, plugins, MCP Servers, and agent workflows makes manual-only review difficult to scale.

A.I.G is designed to address this gap by building an AI security assessment platform that covers models, AI agents, tools, supply chains, and infrastructure. It aims to assess AI infrastructure, agent workflows, the MCP and Agent Skills ecosystem, and LLM applications in a scalable, reproducible, and extensible way.

2 Core Technology

2.1 Design Principles

A.I.G emphasizes engineering feasibility, extensible rules, and explainable results. Deterministic scenarios are handled first with rule-based validation, while complex semantic scenarios use model-assisted judgment. Architecturally, Web services, task scheduling, scanning engines, and knowledge assets are layered and decoupled. At the data layer, maintainable assets cover component identification, vulnerability knowledge, supply-chain risks, and model evaluation. At the integration layer, the Web UI, CLI, REST API, and Agent Skills support different operational contexts.

2.2 Conceptual Model

To avoid treating A.I.G as a loose stack of scanning features, it can be modeled as a four-layer security assessment framework. Targets are the evaluated objects, such as AI services, model APIs, MCP Servers, agent workflows, and Skills. Knowledge Bases provide detection and evaluation references, including component knowledge, vulnerability knowledge, risk taxonomies, and evaluation data. Scanners / Evaluators execute AI Infra Scan, MCP Scan, Agent Scan, ClawScan, and Jailbreak Eval tasks. Evidence & Reports produce risk summaries, evidence descriptions, taxonomy mappings, remediation recommendations, and API/report exports.

A.I.G does not attempt to reduce AI security to a fixed score or a single benchmark. It provides a continuously updated security assessment framework that helps security teams discover, verify, and govern AI-system risks.

2.3 System Architecture

A.I.G uses a Server-Agent distributed architecture. The Web Server handles user interaction, task orchestration, result display, and APIs. Agent Workers execute scanning tasks and return progress and results through a unified event stream. The overall design follows four principles: unified access layer, centralized orchestration, pluggable scanning engines, and externalized rule data.

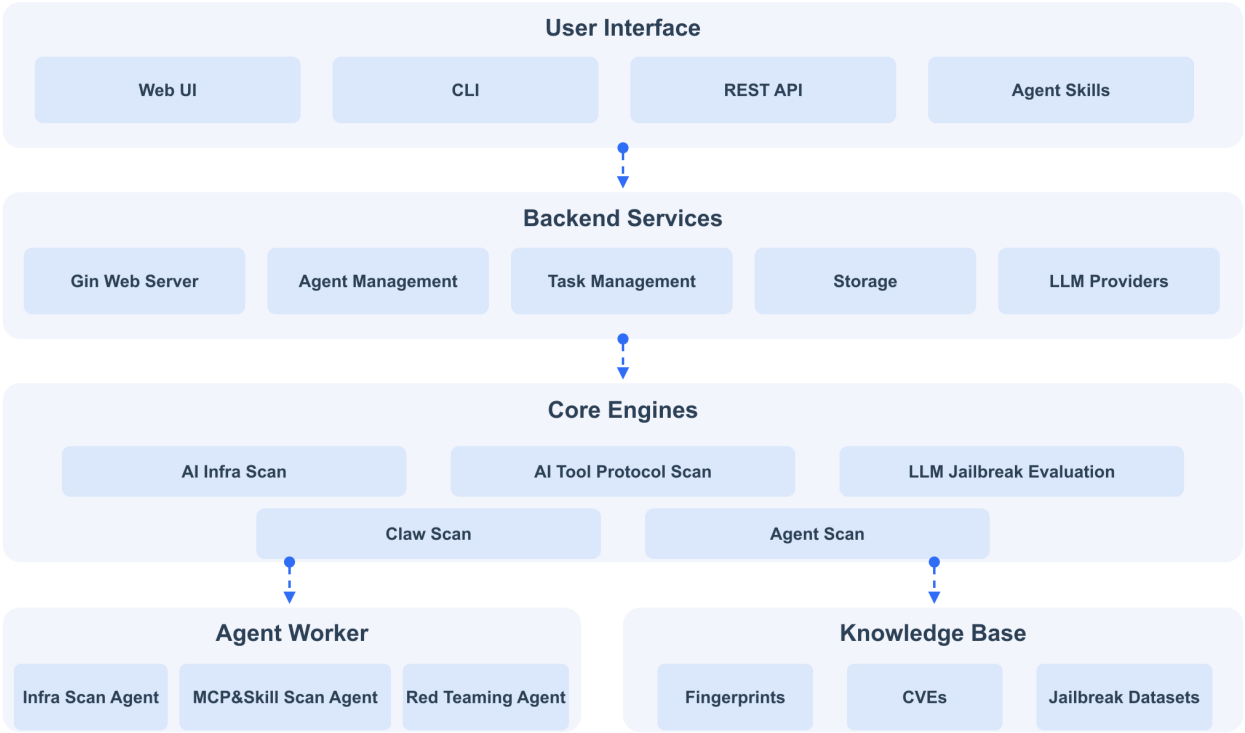

Figure 1: A.I.G Server-Agent architecture.

Runtime responsibilities can be summarized in five layers. The access layer receives scan targets, model

configuration, and task parameters through the Web UI, CLI, REST API, and ClawHub Skills. The orchestration layer manages task lifecycle, worker scheduling, session state, and real-time events. The execution layer runs AI Infra Scan, MCP/Skill Scan, Agent Scan, ClawScan, and Jailbreak Eval. The knowledge layer reads external rules and evaluation assets. The reporting layer generates Web reports, JSON, PDF/API exports, and Agent Security Reports.

Agent Scan is a key extension in the V4 phase. Its goal is not merely to decide whether an agent gives a ”safe” answer. Instead, it evaluates execution risk in real workflows by combining capability boundaries, contextual interaction, and runtime behavior.

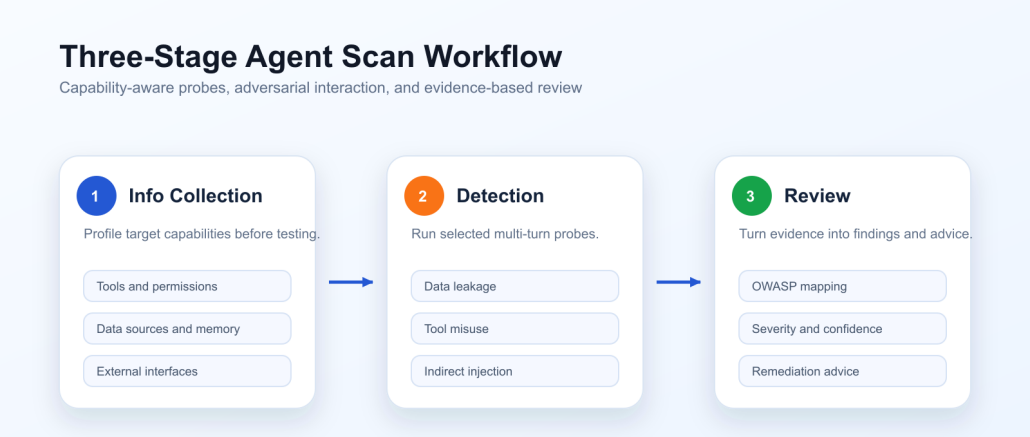

Figure 2: Three-stage Agent Scan assessment workflow.

2.4 Detection Capabilities

A.I.G V4 contains five core detection engines. AI Infra Scan identifies AI-service components and vulnerability risks. MCP & Skills Scan analyzes MCP Servers and Agent Skills for supply-chain risks in tools, permissions, configurations, dependencies, and behavior. Agent Scan evaluates Dify, Coze, Knot, and custom agent workflows for permission-boundary, data-flow, indirect-injection, and tool-use risks. ClawScan focuses on the OpenClaw ecosystem, including configuration risks, Skill risks, and component vulnerabilities. Jailbreak Eval tests model safety resilience against evaluation data for OpenAI-compatible model APIs.

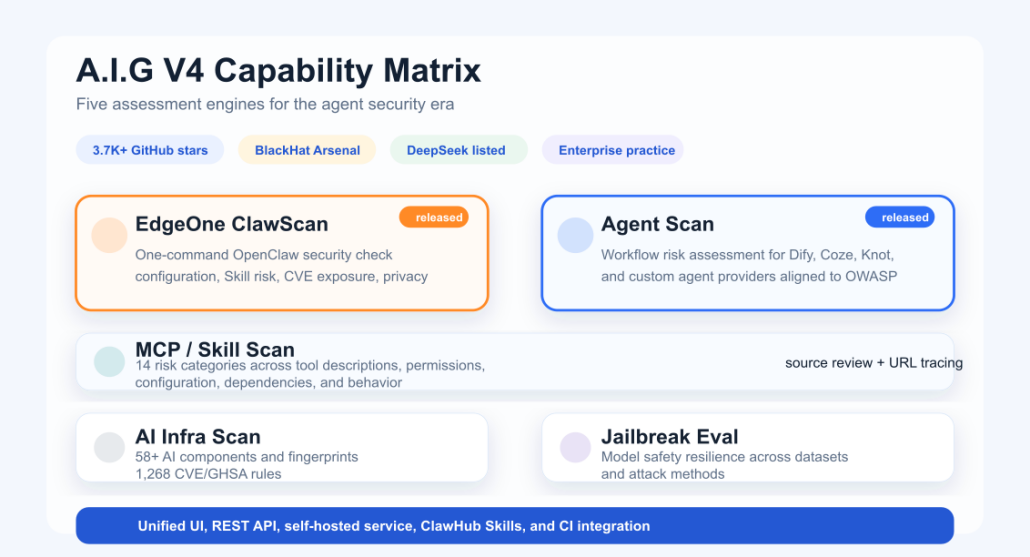

Figure 3: A.I.G V4 capability matrix across ClawScan, Agent Scan, MCP/Skill Scan, AI Infra Scan, and Jailbreak Eval.

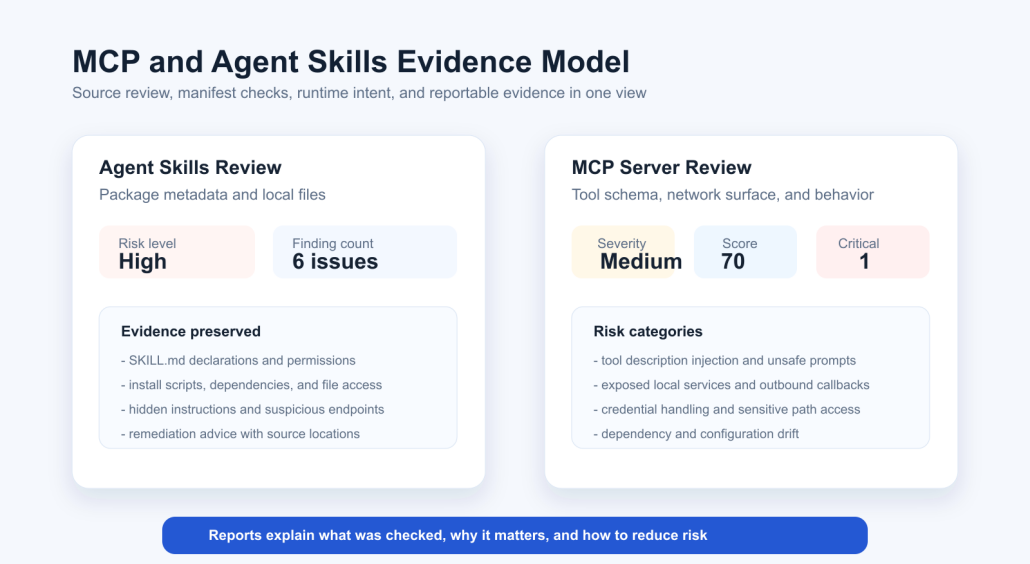

Figure 4: MCP and Agent Skills security scanning report examples.

2.5 Product Form Factors and Boundaries

The open-source A.I.G platform and the lightweight online entry points serve different user groups. The full A.I.G platform is suitable for security teams that need self-hosted batch scanning, API integration, model configuration, and rule extension. EdgeOne ClawScan is a one-click security check Skill for OpenClaw users, focusing on configuration, Skill supply-chain risk, OpenClaw version vulnerabilities, and privacy exposure. EdgeOne Skill Scanner is a lightweight scanning entry point for the general Agent Skills ecosystem, emphasizing local static review, declaration consistency, high-risk behavior indicators, and sensitive-data/instruction risk warnings. The `aig-scanner` Skill connects self-hosted A.I.G capabilities for AI Infra, MCP/Agent Skills, Agent Scan, and Jailbreak scanning into OpenClaw conversation workflows [13][14].

These entry points should not be conflated. The open-source platform provides extensible deep assessment and enterprise integration. ClawScan lowers the barrier for ordinary OpenClaw users to run security self-checks. Skill Scanner provides pre-installation or pre-use security review for Skills. Public pages and Skill metadata both make the boundary clear: scan results are diagnostic aids and do not guarantee discovery of all unknown threats. ClawScan defaults to risk discovery and recommendations; it should not automatically change configuration, upgrade versions, or delete plugins without user assessment. If policy does not allow any external lookup, teams can use a self-hosted A.I.G instance or disable cloud lookup [13][14].

2.6 Technology Stack

A.I.G uses Go + Gin for backend services and task orchestration. Python is used for intelligent analysis and evaluation. Task state is stored in SQLite, and progress/results are returned through WebSocket and SSE. Deployment supports Docker Compose and local binaries. Model access uses OpenAI-compatible APIs.

3 Knowledge Base and Rule System

3.1 Rule Asset Layers

A.I.G maintains rule assets by assessment scenario, covering AI infrastructure identification, vulnerability knowledge, MCP and Agent Skills risk taxonomies, agent workflow risks, and model safety evaluation data. This public report describes asset types and capability boundaries only. It does not disclose internal directory organization, rule fields, prompt templates, or matching logic. Concrete contribution instructions should follow the latest GitHub repository documentation.

Figure 5: A.I.G data and rule asset layers.

3.2 Rule Design Principles

A.I.G follows four rule-design principles. First, deterministic vulnerabilities should be described through reviewable versions, configuration evidence, or public advisories where possible. Second, supply-chain and agent risks should use risk taxonomies and contextual evidence to support explanation instead of relying only on opaque model judgment. Third, evaluation data and rule assets should be maintained continuously as public

vulnerabilities, ecosystem behavior, and community feedback evolve. Fourth, report outputs should support user understanding and governance workflows without exposing internal detection details.

For the public report, rule capabilities can be grouped into three categories: asset and component identification rules, vulnerability and configuration-risk rules, and agent/Agent Skills risk-assessment rules. The first two categories are closer to deterministic validation and are suitable for reproduction and prioritization. The third category is more context-oriented and is designed to support human review and platform governance.

3.3 Rule Update Mechanism

Rules and data evolve through local updates, version releases, and community contributions. The project validates new rules for format and quality and publishes them in versioned releases, allowing security teams to obtain reproducible assessment results in self-hosted, offline, and CI/CD environments. Concrete update entry points and contribution workflows should follow the latest project documentation.

4 Performance and Evaluation Observations

The following section is not a cross-tool leaderboard. It is a protocol-oriented description for reproducible security assessment. A.I.G results are better used for risk discovery, governance prioritization, and human review. When used in research papers or enterprise evaluation, the source version, rule snapshot, target set, model interface, concurrency parameters, and random seeds should be fixed.

4.1 Evaluation Protocol

Table 1: Evaluation inputs, metrics, and evidence forms by scan engine.

Engine	Input Object	Core Metrics	Evidence Output
AI Infra Scan	Running AI services, network ranges, target lists	Asset identification, vulnerability matching, scan efficiency	Component and vulnerability summaries, risk level, remediation guidance
MCP/Skill Scan	MCP Server source code, Agent Skills, remote URLs	Risk taxonomy coverage, valid finding rate, review cost	Risk summary, key evidence description, governance recommendations
Agent Scan	Dify, Coze, Knot, or custom Agent Providers	Agent-risk coverage, valid finding rate, review cost	Risk type, severity, OWASP Agentic classification, remediation guidance
ClawScan	OpenClaw configuration, Skills, components, and runtime environment	Configuration risk, Skill risk, component vulnerabilities, privacy risk	Layered risk summaries, user-readable reports, governance recommendations
Jailbreak Eval	OpenAI-compatible model APIs and evaluation datasets	Safety resilience, scenario coverage, cost and latency	Evaluation summary, risk classification, model-side recommendations

4.2 Coverage Snapshot

This report uses statistics from a local V4 source snapshot. The public edition keeps only the capability-coverage view and does not disclose internal rule counts, directory splits, or per-asset scale, avoiding confusion between rule-maintenance status and absolute detection capability.

Core V4 coverage includes asset identification and vulnerability knowledge for mainstream AI infrastructure components, supply-chain risk taxonomies for MCP and Agent Skills, model safety evaluation data for LLM scenarios, and workflow risk assessment for agent platforms such as Dify, Coze, and Knot. Additional platforms can be supported through Agent Provider/Adapter integrations.

4.3 Performance Observations

For AI infrastructure, a single-target scan is usually completed in seconds and is mainly affected by target response time and network conditions. Batch scans support network ranges and target lists; runtime depends on concurrency, target count, and timeout policy. MCP and Agent Skills security scanning and agent workflow evaluation usually operate at minute-level latency. Their bottlenecks include codebase size, dependency complexity, model response speed, and assessment depth. LLM jailbreak evaluation can range from minutes to hours, depending on dataset size, evaluation methods, and model rate limits.

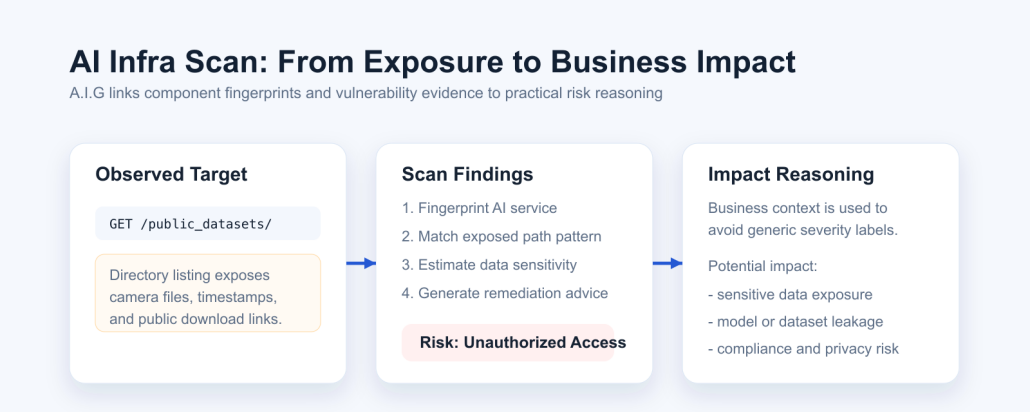

Figure 6: AI Infra Scan impact assessment example for unauthorized access and directory exposure.

4.4 Large-Scale Practice

Public practice has mainly covered three directions. First, Zhuque Lab scanned nearly 50,000 Skills from ClawHub and identified ecosystem-level risks around configuration, permissions, external connectivity, and high-risk behavior. These findings indicate that Agent Skills risk is not only about individual malicious samples; it is an ecosystem problem created by supply, permissions, distribution, and usage flows [15]. Second, A.I.G has been used for OpenClaw exposure checks, bringing public exposure, access control, vulnerable versions, Skill risks, and privacy mistakes into end-user self-assessment [18]. Third, A.I.G supports rapid rule-based response for AI infrastructure supply-chain events. For example, after the publicly disclosed LiteLLM poisoning incident, A.I.G added detection capabilities to help users identify affected environments and remediate in time [16].

4.5 Result Interpretation Boundaries

Intelligent analysis tasks are affected by model capability, context-window size, target-agent stability, and assessment design. A.I.G output is best treated as reviewable risk evidence rather than an opaque final verdict. High-risk findings should preserve necessary evidence and rule rationale. Medium- and low-risk findings should be further confirmed against business permissions, data sensitivity, and exploitability.

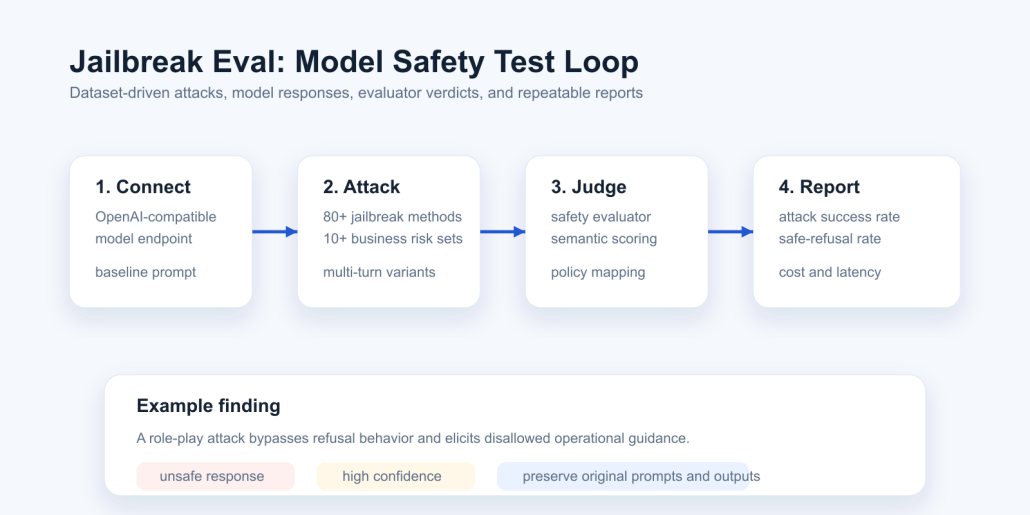

Figure 7: Jailbreak Eval workflow and model safety case example.

5 Practical Use Cases

This section describes four representative scenarios: OpenClaw Skill ecosystem scanning, one-click OpenClaw security checks, MCP and agentic workflow risk identification, and local security review with the general Skill Scanner. Together, these cases cover platform governance, end-user security checks, pre-release developer assessment, and pre-installation risk warning.

5.1 OpenClaw Skill Ecosystem Risk Scanning

Zhuque Lab conducted a full scan of nearly 50,000 ClawHub Skills and found that risk in the AI agent ecosystem is no longer limited to obviously malicious code. It is increasingly shaped by combinations of per-

missions, remote connectivity, sensitive-path access, platform ranking mechanisms, and automated installation behavior.

A.I.G brings Skill declarations, permissions, dependencies, network behavior, and runtime configuration into a unified risk profile, then aggregates point signals into user-readable conclusions. Reports focus on why a finding matters and how it can be handled, without exposing matching fields, rule weights, or decision templates.

Public cases show that traditional rules and antivirus scanning can block obvious malicious code, but they are less reliable against multi-stage supply-chain attacks that span files, dependencies, and external resources. A.I.G differs by focusing on whether observed behavior can form an exploitable risk chain, while evaluating ecosystem distribution, permission declarations, and runtime behavior in one governance view [15].

Skill scan results can therefore support platform listing review, ranking governance, pre-installation warnings, and ecosystem-level baselining for Agent Skills. For ordinary users, reports should translate permission reasonableness, external dependencies, sensitive-resource access, and high-risk behavior signals into understandable conclusions rather than only outputting code fragments or abstract scores.

5.2 One-Click OpenClaw Security Check

In OpenClaw scenarios, A.I.G supports security checks triggered through natural language. It reviews baseline configuration, installed Skills, component vulnerabilities, and privacy risks in layers, then produces a plain-language report for ordinary users.

In this workflow, a user starts the security check from OpenClaw. A typical entry prompt is:

```
Fetch https://matrix.tencent.com/clawscan/skill.md, install it, and then run edgeone-clawscan for a
↪  security check.
```

A.I.G translates the natural-language request into a ClawScan task and performs layered detection for configuration, Skills, component vulnerabilities, and privacy exposure. The report preserves explainable sources for each risk type instead of returning only an abstract risk score. This lowers the barrier for AI agent security checks and allows security capability to be embedded into everyday agent workflows [13][18].

EdgeOne ClawScan's four-layer check model can be summarized as follows. Baseline checks review public exposure and access control. Skill checks review declaration consistency and high-risk behavior. Vulnerability checks cover OpenClaw framework and dependency risks. Privacy checks evaluate sensitive-data access boundaries. Current Skill metadata further clarifies privacy and outbound boundaries: default cloud lookup is used only for supply-chain intelligence and version-vulnerability matching and does not upload Skill source code, chat logs, workspace files, or credentials. If `AIG_CLOUD_LOOKUP=off` is set, all A.I.G HTTPS lookup is skipped and the check falls back to local review [13].

Figure 8: EdgeOne ClawScan one-click OpenClaw security check: command entry, automated check, and report output.

5.3 MCP / Agent Workflow Risk Identification

For MCP Server and agent workflow scenarios, A.I.G helps developers identify risks in tool descriptions, permission boundaries, external content ingestion, sensitive-data handling, and tool-use chains.

A.I.G does not evaluate MCP and agent systems by checking only a single file or API. It combines capability boundaries, permission configuration, external content handling, and runtime behavior for risk analysis. Agent Scan reports provide risk type, severity, OWASP Agentic category, remediation advice, and enough key evidence for human review. For risks such as permission bypass, indirect injection, tool misuse, and data leakage, evidence-based assessment is closer to real agent usage than one model response.

Figure 9: Agent security scan report example: risk level, score, finding count, and OWASP Top 10 distribution.

5.4 General SkillScan Local Security Review

EdgeOne Skill Scanner targets the broader Agent Skills ecosystem rather than only OpenClaw. Its public page positions it as a security check entry point for the general Agent Skills ecosystem. It is already adapted to WorkBuddy, Qclaw, CodeBuddy, OpenClaw, Cursor, Windsurf, Claude Code, and other mainstream platforms, with broader compatibility under continuous development [14]. Its core approach is not to execute Skills in a remote sandbox. Instead, it performs static review over local files, making it better suited for pre-installation checks, enterprise endpoint self-assessment, and built-in security hints in agent platforms.

Its detection framework can be summarized in three layers. The first layer evaluates declaration consistency between Skill descriptions and actual behavior. The second layer identifies high-risk behavior such as over-privileged, destructive, malicious-execution, and permission-abuse patterns. The third layer analyzes sensitive-data and instruction risks that may affect user data, system boundaries, or agent decision-making. Compared with ClawScan, SkillScan emphasizes ”read-only” and ”local-first” behavior. The public FAQ states that source code, API keys, environment variables, and chat logs are not uploaded; the tool does not modify Skills or environment configuration without permission; and scan results are decision-support signals rather than an absolute safety guarantee [14].

Figure 10: EdgeOne Skill Scanner local review framework: contract consistency, high-risk behavior, and sensitive-data analysis.

6 Comparison with Related Tools

6.1 MCP / Skill Security Scanning

A.I.G MCP/Agent Skills scanning covers MCP Servers, Agent Skills, and runtime configuration. It combines rules, contextual analysis, and model-assisted judgment, and supports private deployment, batch scanning,

platform governance, and Web/API/JSON output. Other MCP scanning tools usually focus on a subset of MCP risks and often rely on static rules or cloud analysis. Manual review remains flexible, but coverage depends heavily on reviewer experience, costs are high, and scaling is difficult.

6.2 AI Infrastructure Vulnerability Scanning

A.I.G infrastructure scanning is optimized for AI frameworks and AI services. It focuses on AI infrastructure CVE/GHSA coverage, component fingerprints, and version identification, making it usable out of the box in AI scenarios. General-purpose vulnerability scanners have broader vulnerability databases, but they are usually centered on web, host, and network components. Security teams often need to configure AI asset-identification logic and interpret AI-specific results themselves.

6.3 Agent Security Assessment

A.I.G Agent Scan does not replace security checklists. It turns typical risks around agent goals, tools, permissions, context, and data movement into reviewable assessment results. Compared with generic DAST, Agent Scan focuses more on agent-platform adaptation, runtime behavior, and data permissions rather than only observing HTTP request behavior.

7 Deployment and Integration

7.1 System Requirements

A.I.G recommends Docker 24.0+, 8 GB RAM, 20 GB disk, and a 4-core CPU. The minimum environment is Docker 20.10+, 4 GB RAM, 10 GB disk, and a 2-core CPU. MCP/Agent Scan and Jailbreak Eval depend on model inference, so actual resource and cost requirements also depend on target scale, model interface, and concurrency.

7.2 Deployment Options

A.I.G supports prebuilt images, one-click scripts, source builds, and local binaries. Production environments typically use Docker Compose image deployment. Research and development environments can use source builds or local CLI mode. A typical image deployment uses:

```
git clone https://github.com/Tencent/AI-Infra-Guard.git && cd AI-Infra-Guard
docker compose -f docker-compose.images.yml up -d
```

7.3 Integration Paths

A.I.G can be integrated through several paths. OpenClaw workflows can install `aig-scanner` and set `AIG_BASE_URL` to a self-hosted A.I.G instance. CI/CD pipelines can integrate through the REST API and JSON output. Enterprise security platforms can integrate through the Swagger API. Agent workflows can trigger scanning through ClawHub Skills such as `edgeone-clawscan` and `edgeone-skill-scanner`.

Typical integration commands include:

```
clawhub install aig-scanner

Fetch https://matrix.tencent.com/clawscan/skill.md, install it, and then run edgeone-clawscan for a
↪  security check.

Fetch https://matrix.tencent.com/skillscan/skill.md, install it, and use edgeone-skill-scanner to check
↪  all installed Skills.
```

If enterprise policy does not allow default public-network lookup, the integration should point to a self-hosted A.I.G instance or set `AIG_CLOUD_LOOKUP=off`.

7.4 API Overview

- Swagger documentation: `localhost:8088/docs/index.html`
- Four task-creation API groups: AI Infra Scan / MCP Scan / Agent Scan / Jailbreak Eval
- WebSocket real-time event stream

8 Limitations and Assumptions

8.1 Known Limitations

The current version has seven known limitations. First, the open-source platform has no built-in authentication and is intended by default for intranet or controlled-environment deployment; it should not be directly exposed to the public Internet. Second, MCP/Agent Scan quality depends on the model's code-understanding

and context capabilities. Third, scanning large projects can introduce high inference costs and should be optimized through scope control and task splitting. Fourth, intelligent analysis tasks are usually slower than pure rule scanning and are better suited for deep assessment and prioritized targets. Fifth, model-assisted judgment can produce false positives and false negatives, so human review and secondary validation are still required. Sixth, Agent Scan currently focuses on Dify, Coze, and Knot, with more platforms planned through the Adapter architecture. Seventh, ClawScan and SkillScan are risk-scanning and analysis-assistance tools; by default, they should not automatically harden systems, upgrade components, delete plugins, or modify permissions.

8.2 Assumptions

- Target AI services are reachable over the network for AI Infra Scan.
- MCP source code is available, or remote URLs are reachable for MCP Scan.
- LLM API endpoints are configured and available for MCP/Agent/Jailbreak tasks.
- Evaluation datasets reflect the current threat landscape and may require periodic updates.

8.3 Operational Boundaries

A.I.G is an authorized security assessment and defensive governance tool, not a production traffic-protection system. Scan results represent reviewable risk evidence and do not directly equal final vulnerability conclusions. In production environments, target scope, concurrency strategy, data retention, cloud lookup settings, and retesting workflows should be determined by security teams in business context. For Agent Skills scenarios, "all checks passed" only means that known risks were not found; it does not prove that unknown threats are absent. "Needs attention" does not necessarily mean malicious; it usually indicates elevated permissions or behavior that should be reviewed against the intended business use.

9 Ecosystem and Community

9.1 Academic Collaboration

- Peking University Future Network Laboratory (Prof. Hui Li)
- Fudan University (Prof. Zhemin Yang's team)

9.2 Academic Citations (17 Papers)

Representative papers include MCPGuard: Automatically Detecting Vulnerabilities in MCP Servers (PKU, 2025), When MCP Servers Attack: Taxonomy, Feasibility, and Mitigation (2025), SkillAttack: Automated Red Teaming of Agent Skills (2026), MCP-38: A Comprehensive Threat Taxonomy for MCP Systems (2026), and From Component Manipulation to System Compromise (Fudan, 2026).

9.3 Industry Recognition

- Selected for Black Hat Europe 2025 Arsenal
- First compliance-detection support for the OWASP Agentic Apps / Skills risk list
- Listed by Awesome DeepSeek Integration
- The project team has helped NVIDIA, Hugging Face, OpenClaw, PyTorch, Langflow, and other open-source AI framework/component projects discover and fix multiple high-risk security vulnerabilities, receiving acknowledgements.

9.4 Community

Community channels include the GitHub project page, Discord, and zhuque@tencent.com.

A.I.G welcomes more security teams, engineering teams, academic groups, independent researchers, and practitioners to participate in co-building detection rules, evaluation datasets, benchmarks, product integrations, documentation, and real-world case studies.

9.5 AI-Infra-Guard Pro (A.I.G Pro) Early Access

To provide core users with more professional and forward-looking AI red-team security testing capabilities, Tencent Zhuque Lab is introducing A.I.G Pro, the professional edition of A.I.G. The project invites selected teams, researchers, and practitioners to join the first internal beta through `https://aigsec.ai/`.

Compared with the open-source edition, A.I.G Pro provides the following dedicated experience:

- Early access to new capabilities: beta users can evaluate features that have not yet been open-sourced, including expanded jailbreak-algorithm support, an AI agent lens for agent-level security visibility, LLM security leaderboards, and other advanced assessment modules.
- Dedicated technical support: feedback and suggestions submitted by beta users through email or other beta channels receive prioritized handling.
- Convenient public access: users can register and log in directly on the public service through invitation codes, without complex local installation and configuration.
- Built-in LLM APIs: the platform integrates mainstream LLM APIs and provides free trial quota for internal beta users, making it easier to start security assessment quickly.

Community participation remains central to A.I.G. During the internal beta, teams and users who provide valuable suggestions, contribute code, or publicly share beta experience will receive a Tencent-themed gift and public acknowledgement in the GitHub project documentation.

9.6 Contribution Guide

Rules and code contributions mainly cover component identification, vulnerability knowledge, MCP/Agent Skills risk, model safety evaluation data, and core code. The project welcomes community pull requests for rules, datasets, documentation, and product integrations. Concrete directories, format requirements, and validation commands should follow the latest contribution guide in the GitHub repository.

10 Project Team and Contributions

10.1 Tencent Zhuque Lab

Tencent Zhuque Lab is a leading AI security laboratory established by Tencent Security Platform Department in 2019. The lab focuses on practical offensive and defensive security work and frontier research in AI security, with research directions covering LLM security, AI agent security, AI for security, and AI-generated content detection.

The team has repeatedly helped well-known vendors such as NVIDIA, Google, and Microsoft, as well as open-source communities including OpenClaw, Linux, and Hugging Face, fix a large number of high-risk vulnerabilities and has received official public acknowledgements. It has launched AI security products including A.I.G (AI-Infra-Guard), an open-source AI red-team security testing platform, and Zhuque AI Detection Assistant. Its research results have been published at leading international security and AI venues including Black Hat, DEF CON, ICLR, CVPR, NeurIPS, and ACL, and the team has published the book *AI Security: Technology and Practice*.

10.2 Project Team and Contributions

Role	Member	Contribution
Head of Tencent Security Platform Department	Yong Yang	Initiated A.I.G and guided its expansion from AI infrastructure vulnerability scanning to AI agent security assessment, tool-risk governance, and permission-boundary analysis.
Head of Tencent Zhuque Lab	Xing Zheng	Guided A.I.G's overall direction, key technical roadmap, and rule-knowledge system, supporting continued evolution across AI infrastructure security and AI agent security.
Project Lead	Nicky	Frontier security research, product planning, technical-route decisions, internal and external collaboration, and communications.
Technical Lead	Python	Overall architecture design, core module development, and version iteration.
Core Contributor	Zona	Frontend interaction, product experience, community operations, and user-feedback loop.
Core Contributor	Fyoung	AI Infra vulnerability component fingerprint updates and Benchmark system construction.
Core Contributor	Robert	LLM safety assessment and jailbreak-evaluation strategy operations.
Core Contributor	Zoe	LLM safety assessment, jailbreak evaluation, and model-integration module development.

Role	Member	Contribution
Core Contributor	Xiangfan	Security capability development for Skill risks and agent loss-of-control scenarios.
Contributor	Ronin	Participated in AI agent security scanning development.
Contributor	Rsin	Participated in community operations and campaign communications.

11 Related Standards and Tool Positioning

A.I.G is positioned as a multi-layer AI security assessment platform rather than a single-point model-testing tool. OWASP Top 10 for Agentic Applications 2026 [3] provides a framework for general agent security scanning, tool-call risk, permission boundaries, and runtime behavior evaluation. NIST AI RMF and the CISA/NSA/FBI AI Data Security Guidance [9][10] provide references for governance, data security, and lifecycle security. MITRE ATLAS and AVID [11][12] provide references for attack techniques, incidents, and vulnerability knowledge. Compared with model red-team tools such as garak, Giskard, and NeMo Guardrails, A.I.G covers a broader AI engineering stack. Compared with SAST, DAST, vulnerability scanners, and cloud-security platforms, A.I.G is better treated as a complementary assessment layer for AI assets, agent workflows, and model-behavior risk.

12 Conclusion and Future Work

By V4, A.I.G has evolved from an early AI infrastructure vulnerability scanner into a multi-engine AI red-team security assessment platform covering AI Infra Scan, MCP/Skill Scan, Agent Scan, ClawScan, and Jailbreak Eval. Its core value is not replacing existing SAST, DAST, cloud-security, or model-evaluation tools. Its value is filling new risk areas in the AI engineering stack: component fingerprints and CVE rules, MCP and Agent Skills tool supply chains, agent workflow permissions and context flows, and LLM behavior safety assessment.

Future work will focus on agent security scanning. As general-purpose agents move from conversational assistants into executable workflows, risk also moves from generating unsafe text to tool use, data access, permission boundaries, and cross-system collaboration. A.I.G will continue to improve OWASP Top 10 risk scanning for general agents.

At the integration layer, A.I.G will improve support for user-built agents and mainstream agent products, lowering the cost of connecting different platforms to security assessment capabilities. At the risk-coverage layer, A.I.G will expand detection around agent goals, tools, permissions, supply chains, context, and data movement while staying aligned with OWASP Top 10 for Agentic Applications. At the assessment layer, A.I.G will improve capability-aware evaluation, evidence quality, and false-positive control so reports are more useful for audit, retesting, and regression testing. At the governance layer, scan results will be mapped to risk levels, remediation advice, and retesting workflows, and integrated with CI/CD, security operations platforms, model-governance platforms, and agent release processes.

From a technical-evolution perspective, the next stage of A.I.G is not simply to add more rules. The priority is to improve agent-risk coverage, evidence quality, and governance closure. Under a unified risk taxonomy, A.I.G will continue connecting risk identification, evidence description, remediation advice, and retest results, helping enterprises move AI security assessment from one-time scanning to continuous governance [17].

References

[1] Tencent Zhuque Lab, "AI-Infra-Guard: A Comprehensive, Intelligent, and Easy-to-Use AI Red Teaming Platform," GitHub repository, 2025. [Online]. Available: `https://github.com/Tencent/AI-Infra-Guard`. Accessed: May 9, 2026.

[2] Tencent Zhuque Lab, "AI-Infra-Guard README, CHANGELOG, and API documentation," GitHub repository documentation. [Online]. Available: `https://github.com/Tencent/AI-Infra-Guard/tree/main`. Accessed: May 9, 2026.

[3] OWASP GenAI Security Project, "OWASP Top 10 for Agentic Applications for 2026," Dec. 9, 2025. [Online]. Available: `https://genai.owasp.org/resource/owasp-top-10-for-agentic-applications-for-2026/`. Accessed: May 9, 2026.

[4] Stanford Institute for Human-Centered Artificial Intelligence, "AI Index Report 2026," 2026. [Online]. Available: `https://hai.stanford.edu/ai-index/2026-ai-index-report`. Accessed: May 9, 2026.

[5] IBM Newsroom, "IBM Report: 13% Of Organizations Reported Breaches Of AI Models Or Applications, 97% Of Which Reported Lacking Proper AI Access Controls," Jul. 30, 2025. [Online]. Available: `https://newsroom.ibm.com/2025-07-30-ibm-report-13-of-organizations-reported-breaches-of-ai-models-or-applications%2C-97-of-which-reported-lacking-proper-ai-access-controls`. Accessed: May 9, 2026.

[6] HiddenLayer, "2026 AI Threat Landscape Report," 2026. [Online]. Available: `https://www.hiddenlayer.com/report-and-guide/threatreport2026`. Accessed: May 9, 2026.

[7] Wiz Research, "State of AI in the Cloud 2026: How AI Adoption, Autonomy, and Attacker Innovation Are Reshaping Cloud Security," 2026. [Online]. Available: https://www.wiz.io/reports/state-of-ai-in-the-cloud-2026. Accessed: May 9, 2026.

[8] Cisco, "AI Readiness Index 2025," 2025. [Online]. Available: https://www.cisco.com/c/m/en_us/solutions/ai/readiness-index.html. Accessed: May 9, 2026.

[9] CISA, NSA, FBI, and international partners, "AI Data Security: Best Practices for Securing Data Used to Train & Operate AI Systems," May 2025. [Online]. Available: https://www.cisa.gov/resources-tools/resources/ai-data-security-best-practices-securing-data-used-train-operate-ai-systems. Accessed: May 9, 2026.

[10] National Institute of Standards and Technology, "Artificial Intelligence Risk Management Framework (AI RMF 1.0)," NIST AI 100-1, Jan. 2023. [Online]. Available: https://www.nist.gov/itl/ai-risk-management-framework. Accessed: May 9, 2026.

[11] MITRE, "MITRE ATLAS: Adversarial Threat Landscape for Artificial-Intelligence Systems," knowledge base. [Online]. Available: https://atlas.mitre.org/. Accessed: May 9, 2026.

[12] AVID, "AI Vulnerability Database," vulnerability knowledge base. [Online]. Available: https://avidml.org/. Accessed: May 9, 2026.

[13] Tencent Zhuque Lab and Tencent Cloud EdgeOne, "EdgeOne ClawScan / OpenClaw Security Check," 2026. [Online]. Available: https://matrix.tencent.com/clawscan/. Accessed: May 9, 2026.

[14] Tencent Zhuque Lab and Tencent Cloud EdgeOne, "EdgeOne Skill Scanner," 2026. [Online]. Available: https://matrix.tencent.com/clawscan/skillscan. Accessed: May 9, 2026.

[15] Tencent Zhuque Lab, "We Scanned 50,000 Skills and Found That the Risk Still Exists," Tencent Security, 2026. [Online]. Available: https://security.tencent.com/index.php/blog/msg/224?from_tab=security. Accessed: May 9, 2026. [in Chinese]

[16] Tencent Zhuque Lab, "AI Infrastructure Security Lessons from the LiteLLM Poisoning Incident Affecting 480 Million Downloads," Tencent Security, 2026. [Online]. Available: https://security.tencent.com/index.php/blog/msg/214?from_tab=security. Accessed: May 9, 2026. [in Chinese]

[17] Tencent Zhuque Lab, "Securing AI Agents in the Agent Era: Core Risks, Protection Strategies, and A.I.G Practice," Tencent Academy / Black Hat Arsenal material, 2026. [Online]. Available: https://github.com/Tencent/AI-Infra-Guard/blob/main/Arsenal-BHEU2025-AI-Infra-Guard.pdf. Accessed: May 9, 2026. [in Chinese]

[18] Tencent Zhuque Lab, "AI-Infra-Guard V4 / ClawScan release material," GitHub release, 2026. [Online]. Available: https://github.com/Tencent/AI-Infra-Guard/releases. Accessed: May 9, 2026. [in Chinese]