AGENT SKILLS WORKSHOP
CAIS 2026

SkillSpector: A Pre-Publication Security Control for Agent Skills

Nir Paz*, Keshav Pradeep*, Narendran Raghavan, Ashley Nikirk, Mohit Gupta

* Equal contribution

Treat skills as compound artifacts — scan instructions, metadata, permissions, dependencies, code, and semantic intent as one review target before publication or install.

64 rule patterns

16 risk categories

1,058 live CI/CD scans analyzed

34.1% of scans surface a semantic finding

Why scan agent skills?

26.1%

of 31,132 marketplace skills had a vulnerability.
Bundled-script skills were 2.12× more likely to be vulnerable than instruction-only skills.

Liu et al., "Agent Skills in the Wild," 2026.

A skill is not only code — it is agent context. Names, descriptions, triggers, parameter docs, and instructions are read by the model and used to select tools.

Six risk surfaces in one skill bundle

Instruction
Direct prompt injection, hidden comments, role-played authority, system-prompt extraction

Metadata / Trigger
Zero-width chars, homoglyphs, base64 payloads, BIDI controls (Unicode override), parameter overrides

Permission
Wildcards, undeclared sensitive capabilities, declared-vs-actual behavior mismatch

Code
exec / eval, subprocess, dynamic imports, env + file read → network sink

Dependency
Vulnerable packages (OSV.dev CVE feed), remote install scripts, unpinned versions, typosquats

Semantic
Declared purpose vs. actual behavior — surfaced as advisory review leads only

No existing scanner reasons across all six layers.

- SAST flags subprocess but misses YAML frontmatter.
- Prompt-injection detectors miss permission mismatch.
- Dependency scanners miss Unicode-hidden instructions and wildcard manifests.

Contributions

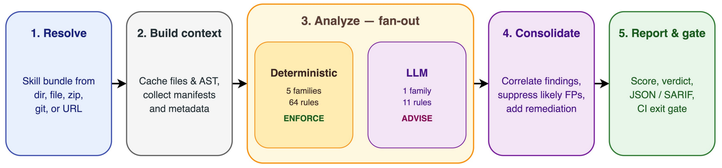
- A unified threat model that treats instruction, metadata, permission, code, and dependency risks as one review target.
- An open-source LangGraph-based scanner that unifies AST + taint analysis, OWASP-mapped static patterns, YARA malware rules, OSV CVE checks, MCP-protocol audits, and LLM-backed semantic review under one finding model.
- A canonical deployment profile that separates enforcement-grade from advisory evidence — and shows that split holding up under live CI/CD load.

Open Source Repository
github.com/nvidia/skillspector

Read the Paper
openreview.net/forum?id=rVAPXHmGHN

How SkillSpector works

LangGraph workflow chains AST + taint, OWASP-mapped patterns, YARA, OSV CVE, MCP, and LLM semantic analyzers over one shared skill context.



Six analyzer families

64 deterministic rule IDs unpack to 220+ regex patterns, plus 11 LLM advisory rules — all normalized into one finding model. Five enforce; the LLM family advises only.

Analyzer family	What it catches	Rules	Posture
Static patterns	Prompt injection, exfil triggers, system-prompt leaks, suspicious metadata	37	ENFORCE
Python AST + taint	exec / eval / subprocess; env → network / execution-sink flows	13	ENFORCE
MCP least privilege	Wildcard / missing permissions vs detected capabilities	4	ENFORCE
MCP tool poisoning	Hidden instructions, zero-width / BIDI, homoglyphs, param overrides	4	ENFORCE
Supply chain	OSV.dev CVE lookups, remote install scripts, likely typosquats	6	ENFORCE
LLM semantic + meta	Novel injection, gradual deception, description ↔ behavior mismatch	11	ADVISE

Grounded in external risk taxonomies

Rules and analyzers map to public AI-risk frameworks and feeds — SARIF output ships framework_tags for CI integration.

- OWASP LLM Top 10 (2025) · OWASP Agentic AI (2026)
- MITRE ATLAS — adversarial AI techniques
- OSV.dev — Google's open-source CVE feed
- MCP spec — tool poisoning, parameter overrides

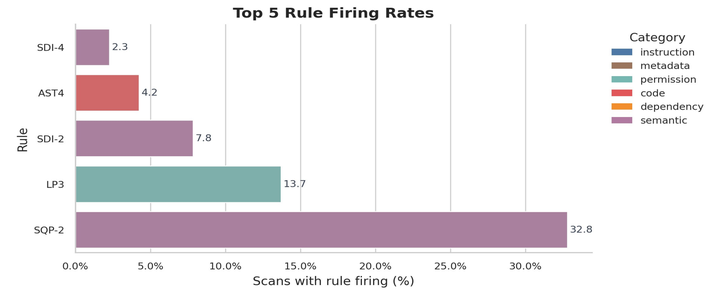
The judgment call

Deterministic = enforce. LLM = advise.

Skill content is adversarial input. An LLM should not become an uncalibrated security oracle.

Live CI/CD evidence

Deployed inside a multi-stage skill-publication gate. 1,058 LangSmith traces over a benign-skewed internal catalog.



The enforce/advisory split shows up in the data.
SQP-2 (advisory): 32.8% — missing user-facing warnings, not a security defect. Review-queue pressure, never blocks.
LP3 (enforcement): 13.7% — skill code uses shell / network / file / env but the manifest declares no permissions. Hard-fail.

Functional validation fixtures

Six representative fixtures, each exercising a distinct analyzer family. Confirms each family fires on the cases it owns — not a precision or recall claim.

Fixture	Scenario	Findings	Score	Verdict
safe-greeting	Benign docs-only skill	—	0	SAFE
metadata-poisoning	Hidden HTML instruction · Cyrillic homoglyph · param override	TP1, TP2, P1, P2	100	DO NOT INSTALL
underdeclared-agent	No permissions · reads env · invokes shell · posts to network	AST4, TT3, LP3, E1, E2, OH1	100	DO NOT INSTALL
env-exfiltration	Env-var collection sent to external endpoint	E1, E2	58	DO NOT INSTALL
vulnerable-dependencies	Remote install script · CVE'd pkgs · abandoned · typosquat	SC2, SC4, SC5, SC6, TM2	100	DO NOT INSTALL
wildcard-permissions	Wildcard permission + chmod + delete operations	AST4, AST5, LP2, TR2, TM1	100	DO NOT INSTALL

Limitations & next steps

Limitations
- No precision/recall claim — validation is fixture-based plus an unlabeled 178-skill field study.
- Intraprocedural taint engine (within-function only) misses helper, imported-module, and cross-file exfiltration flows.

Next
- Labeled corpus eval over 20,000+ public skills with synthetic malicious variants and adversarial stress tests.
- Sandboxed dynamic analysis for high-risk review.