Do Agent Skills Speak Safety in Every Language? A Cross-Lingual Security Analysis of the Skills Ecosystem

Anonymous Author(s)

Abstract

Agent Skills are structured `SKILL.md` packages that augment LLM agents at inference time. They have been adopted across Claude Code, Gemini CLI, OpenClaw, Codex, and other agent platforms [5]. The ToxicSkills audit found that 37% of 3,984 Skills contain security flaws [13], but no study has examined whether this rate is uniform across languages. We present the first cross-lingual security analysis of the Skills ecosystem, scanning **3,656** real-world Skills across **8** languages with a Skills-native scanner (Cisco Skill Scanner v2.0.9) and a general-purpose baseline (Bandit 1.8.0). We find that code-level security findings (command injection, hardcoded secrets, prompt injection) affect 2.3% of Skills with *no statistically significant cross-lingual difference*. However, behavioral flags—particularly social-engineering indicators such as vague or misleading descriptions—are 3–5× higher for non-English Skills ($p < 0.001$, Bonferroni-corrected), with Japanese Skills reaching 31.7% compared to 6.1% for English. Manual precision assessment reveals that these behavioral heuristics have low precision overall (EN: 10%, JA: 0% on 20-finding samples), but the false-positive rate is particularly acute for non-English content. We argue that this gap reflects scanner calibration bias and call for multilingual evaluation of Skills security tooling. We release the dataset and analysis scripts to support reproducible research.

CCS Concepts: • **Security and privacy** → **Software security engineering**; • **Software and its engineering** → *Software supply chain.*

Keywords: agent skills, cross-lingual safety, supply chain security, static analysis, LLM agents

1 Introduction

Agent Skills have rapidly become the dominant mechanism for extending LLM agent capabilities without model modification [5]. A Skill is a `SKILL.md` file—a structured package of instructions, code snippets, and tool references—adopted across multiple agent platforms [5]. Public registries now index over 100,000 Skills [1], and curated Skills improve agent pass rates by 16.2 percentage points on average [4].

This growth introduces a supply chain attack surface. The ClawHavoc campaign planted 1,184 malicious Skills on ClawHub [2], and Snyk's ToxicSkills audit found that 37% of 3,984 publicly listed Skills contain at least one security flaw [13]. In response, Skills-native security scanners have emerged: Cisco Skill Scanner [6] combines YARA signatures, behavioral dataflow analysis, and pipeline taint tracking; Snyk Agent Scan [12] adds API-based verification. Yet no study has examined whether the Skills security landscape is *uniform across languages.*

The multilingual safety alignment gap in LLMs is well documented [7, 15]. If this gap extends to Skills—or to the tools that scan them—then multilingual agent deployments face risks that monolingual audits cannot capture. We investigate this through the first cross-lingual security analysis of the Skills ecosystem.

Contributions.

- **[Ecosystem characterization]** We scan 3,656 real-world Skills across 8 languages with a Skills-native scanner, finding that code-level security findings affect only 2.3% of Skills—substantially lower than the 37% reported by ToxicSkills—while 87% lack a license declaration.
- **[Cross-lingual finding]** Code-level security findings show no significant cross-lingual difference ($p > 0.05$). However, behavioral flags (social-engineering indicators) are 3–5× higher for non-English Skills, suggesting that current scanning tools are insufficiently calibrated for multilingual content.
- **[Scanner evaluation]** We compare a Skills-native scanner against a general-purpose baseline on the same corpus, quantifying the coverage gap and identifying categories invisible to traditional SAST tools.

2 Methodology

2.1 Dataset Construction

We collect Skills from three source categories: community-curated registries (1,414 Skills), platform-official repositories from three major agent platforms (208 Skills), and GitHub code search for public `SKILL.md` files (1,868 Skills, plus 166 from targeted non-English queries). Inclusion criteria require ≥10 lines of content, public accessibility, and SHA-256 deduplication. Language detection uses `langdetect` [11] with CJK Unicode block correction for misclassification. Table 1 summarizes the dataset; all Skills are real-world artifacts from 1,380 independent repositories.

2.2 Security Scanner

We use Cisco Skill Scanner v2.0.9 [6], an open-source Skills-native scanner released in response to the ClawHavoc incident. It combines three analysis engines: (1) static analysis

Table 1. Dataset summary. All Skills are from public registries.

Language	Skills	%	Repos	Avg. lines
English	2,734	74.8	737	263
Chinese	330	9.0	257	407
Japanese	227	6.2	175	126
Spanish	163	4.5	87	104
Korean	113	3.1	89	—
Other (PT, DE, FR)	89	2.4	49	—
Total	**3,656**	100	**1,380**	—

with 900+ YARA signatures, (2) behavioral dataflow analysis that tracks pipeline taint (e.g., `curl | sh` patterns), and (3) compound pattern detection for multi-step attack chains. It outputs structured findings with severity levels (CRITICAL, HIGH, MEDIUM, LOW, INFO) and 13 threat categories aligned with the Agent Skills Threat Taxonomy.

Findings fall into three tiers:

- **Security**: command injection, prompt injection, hardcoded secrets, obfuscation, supply chain attacks.
- **Behavioral**: social engineering (vague/misleading descriptions), autonomy abuse, tool chaining abuse.
- **Policy**: missing license, invalid metadata, description quality.

As a general-purpose baseline, we run Bandit v1.8.0 on Python code blocks extracted from each `SKILL.md`. Snyk Agent Scan [12] requires API authentication and was not included in the batch comparison.

2.3 Cross-Lingual Experiment

We compute finding rates per language at each tier. For EN vs. non-EN and per-language pairwise comparisons, we use Fisher's exact test with Bonferroni correction for 18 tests (6 languages × 3 tiers, corrected $\alpha = 0.0028$).

2.4 Ground Truth

We conduct two annotation passes: (1) *Code-level*: two independent raters (one LLM, one human) label 50 random security-tier findings as TP or FP using CWE definitions. Cohen's $\kappa = 1.00$ (perfect agreement); code-level precision is 12.0% (6 TP, 44 FP), reflecting LOW/INFO-severity credential-detection rules triggered by documentation that *discusses* credentials. All 6 true positives are CRITICAL or HIGH severity (3 hardcoded database passwords, 2 daemon-privilege instructions, 1 NOPASSWD configuration); the 53 Skills classified as *unsafe* in Table 2 are drawn from this higher-precision tier. (2) *Behavioral*: we sample 20 EN and 20 JA Skills flagged by `SOCIAL_ENG_VAGUE_DESCRIPTION` and assess whether descriptions genuinely fail to communicate Skill functionality. EN behavioral precision is 10% (2/20 genuinely vague); JA behavioral precision is 0% (0/20—all flagged JA Skills have clear Japanese descriptions or are non-Skill files misincluded in the corpus). Description clarity is judged by whether the description contains ≥1 sentence describing the Skill's functionality in the Skill's primary language; a translation tool was used to verify Japanese descriptions.

3 Results

3.1 Overall Findings

Table 2 summarizes the Cisco scanner findings. Of 3,655 successfully scanned Skills, 53 (1.5%) are classified as *unsafe* (containing CRITICAL or HIGH findings), while 4,288 findings span 13 threat categories. Policy violations dominate (86% of findings), driven by 3,197 Skills lacking a license field. Excluding policy, 599 security and behavioral findings affect 387 Skills (10.6%). The gap between our 2.3% code-level rate and ToxicSkills' 37% reflects definitional scope: ToxicSkills counts all flagged patterns including overpermissions and documentation issues, while our code-level tier is restricted to findings with concrete exploit vectors. When both studies' broadest tiers are compared (our 10.6% security+behavioral vs. their 37%), the remaining difference reflects corpus composition and scanner methodology. This definitional divergence underscores the need for standardized severity tiers in Skills security reporting.

Table 2. Findings by threat category (Cisco Skill Scanner v2.0.9, $n = 3{,}655$). Categories with <3 findings omitted.

Tier	Category	Findings	Skills
Security	Command injection	151	75
	Prompt injection	19	17
	Hardcoded secrets	7	7
Behavioral	Social engineering	361	345
	Autonomy abuse	30	29
Policy	Policy violation	3,689	3,500
Unsafe (CRIT/HIGH)		—	**53 (1.5%)**

3.2 Cross-Lingual Analysis

Figure 1 presents finding rates by language across the three tiers.

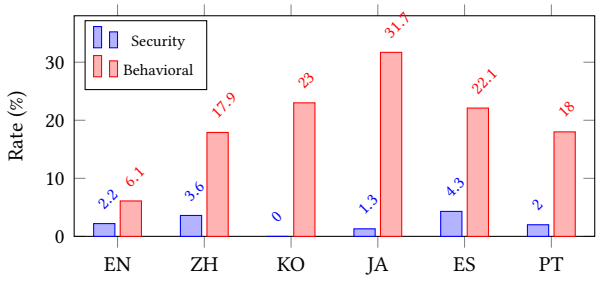

Figure 1. Finding rates by Skill language. Security findings show no significant cross-lingual difference. Behavioral findings are 3–5× higher for non-English Skills (unadjusted rates; see text for adjustment).

Security findings: no cross-lingual gap. Code-level security findings (command injection, prompt injection, hardcoded secrets) affect 2.2% of English and 2.0–4.3% of non-English Skills. No pairwise Fisher's exact test reaches significance after Bonferroni correction (smallest $p = 0.10$ for EN vs. ES).

Behavioral findings: significant gap. Behavioral flags—predominantly the `SOCIAL_ENG_VAGUE_DESCRIPTION` rule—are substantially higher for non-English Skills. Four of six pairwise comparisons survive Bonferroni correction ($\alpha = 0.0028$): JA (31.7%, $p < 0.001$), KO (23.0%, $p < 0.001$), ES (22.1%, $p < 0.001$), and ZH (17.9%, $p < 0.001$) vs. EN (6.1%).

Interpreting the behavioral gap. The `SOCIAL_ENG_VAGUE_DESCRIPTION` rule flags Skills whose descriptions do not clearly communicate functionality. To assess whether the cross-lingual gap reflects genuine quality differences or scanner bias, we manually evaluated precision on random samples of 20 EN and 20 JA flagged Skills (§2.4). EN behavioral precision is 10% (2 genuine findings: one Skill with description "build" and one Anthropic impersonation). JA behavioral precision is 0%: of 20 flagged JA Skills, 7 are not agent Skills at all (game documentation and portfolio pages included via filename matching), and the remaining 13 have clear Japanese-language descriptions flagged solely because the scanner's heuristics expect English-style documentation. We note that 23% of JA files in our dataset lack YAML frontmatter and are likely non-Skill files (game documentation, portfolio pages) included via filename matching. Restricting to files with valid frontmatter reduces the JA behavioral rate from 31.3% to 10.9% (EN: 5.7% to 1.8%; Fisher's $p < 0.001$). The gap persists after this adjustment, confirming a *scanner calibration bias* beyond dataset noise: behavioral heuristics have low precision overall, but the false-positive rate is particularly acute for non-English content.

Robustness. Within GitHub-search-only Skills (n = 1,868), security rates remain similar: EN 3.2%, ZH 3.7%, JA 1.3%. Non-English Skills are well-distributed across repositories (ZH: 257 repos, JA: 175, KO: 89), mitigating single-author clustering.

3.3 Baseline Comparison

Bandit 1.8.0 detects 116 findings in 46 Skills (1.3%) on extracted Python code blocks. The Cisco scanner detects 151 command-injection findings alone, plus 19 prompt-injection and 7 hardcoded-secret findings that Bandit misses entirely because they occur in Markdown prose and shell pipelines rather than in Python source. Semgrep 1.99.0 with default configuration detects zero findings on the same corpus. The comparison confirms that Skills-native scanners provide substantially broader coverage than general-purpose SAST tools.

3.4 Case Studies

Case 1: Pipeline taint in an offensive Skill. A privilege-escalation Skill (marked `risk: offensive`) contains `curl -L ... | sh` and `find . -exec /bin/sh` patterns. Cisco's pipeline analyzer flags both as CRITICAL command injection; Bandit detects neither because the commands appear in Markdown, not Python.

Case 2: Behavioral flag as scanner bias. A Japanese Skill for estimate generation has a detailed Japanese-language description and structured YAML frontmatter. Cisco flags it for `SOCIAL_ENG_VAGUE_DESCRIPTION` because its heuristics do not parse Japanese text as "clear documentation." This false positive illustrates the multilingual calibration gap.

Case 3: Well-structured Skill as positive example. A curated writing-style Skill with 21 rules and explicit safety constraints (e.g., "never overwrite files you own") triggers zero findings across all scanners, demonstrating that the `SKILL.md` format can encode safety discipline effectively.

4 Discussion and Related Work

The multilingual calibration gap. Our central finding is that the Skills security tool we evaluated exhibits a *multilingual calibration gap*: behavioral detection heuristics produce 3–5× more findings on non-English Skills, with manual verification confirming a 0% precision on Japanese behavioral flags vs. 10% on English. Code-level detection, by contrast, shows no cross-lingual bias. This parallels findings in multilingual LLM safety [7, 14, 15], where models exhibit weaker guardrails on non-English content. We extend this observation from model behavior to the *tooling layer*: the tools designed to secure the Skills ecosystem inherit the same English-centric assumptions. More broadly, any rule-based scanner whose documentation-quality heuristics are calibrated on English-language corpora will exhibit this pattern, because natural-language style heuristics (e.g., description clarity, specificity, length norms) do not transfer across languages without explicit multilingual support. Our finding is therefore not specific to a single tool but reflects a structural limitation of monolingual heuristic design in multilingual ecosystems.

Implications. Skills registries should (1) evaluate scanner precision across languages before deploying automated quality gates; (2) ensure documentation-quality heuristics support multilingual content; and (3) distinguish security findings from policy and behavioral flags in severity classification. The OWASP Agentic Security Initiative Top 10 [8] provides a taxonomy for agent-level risks; our work extends this to the Skill supply chain.

Related work. Cisco Skill Scanner [6] and Snyk Agent Scan [12] are the two Skills-native scanners; neither has been

evaluated for cross-lingual bias. General SAST tools such as Bandit [9] and Semgrep [10] do not model Skill-specific concepts. InjecAgent [16] and AgentHarm [3] benchmark runtime agent safety but do not address static Skill security.

Limitations. Our behavioral-bias finding rests on one scanner (Cisco v2.0.9); future work should replicate with Snyk and SkillRisk. The 12% precision on code-level annotation reflects a 50-finding sample dominated by credential-detection rules; a larger sample stratified by rule type would yield more informative precision estimates. Portuguese and German samples remain small ($n = 50$ and $n = 26$). Our dataset includes non-Skill files: 23% of Japanese entries lack YAML frontmatter (likely game documentation and portfolios reusing the `SKILL.md` filename), compared to <5% for English and Chinese. We report adjusted rates where this affects conclusions; the cross-lingual behavioral gap persists after adjustment (JA 10.9% vs. EN 1.8%, $p < 0.001$). Collection source and language are partially confounded for Portuguese (75% from one registry).

5 Conclusion

We presented the first cross-lingual security analysis of 3,656 real-world Agent Skills across 8 languages using a Skills-native scanner. Code-level security findings show no significant cross-lingual difference, but behavioral flags are 3–5× higher for non-English Skills due to scanner documentation heuristics calibrated for English. This *multilingual calibration gap*—confirmed by 0% behavioral precision on Japanese vs. 10% on English—means that non-English Skills face systematically higher false-positive rates in automated security gates, a form of tooling bias that registries should address. We release the dataset and scripts to support reproducible research.

References

[1] 2026. agentskill.sh: Agent Skills Directory. https://agentskill.sh 107,000+ Skills listed as of April 2026.

[2] 2026. ClawHavoc: 1,184 Malicious Skills Found on ClawHub. https://cyberpress.org/clawhavoc-poisons-openclaws-clawhub-with-1184-malicious-skills/ February 2026.

[3] Maksym Andriushchenko et al. 2025. AgentHarm: A Benchmark for Measuring Harmfulness of LLM Agents. In *ICLR*.

[4] Anonymous. 2026. SkillsBench: Benchmarking How Well Agent Skills Work Across Diverse Tasks. *arXiv preprint arXiv:2602.12670* (2026).

[5] Anthropic. 2025. Agent Skills Overview. https://agentskills.io/home Accessed April 2026.

[6] Cisco AI Defense. 2026. Cisco AI Defense Skill Scanner. https://github.com/cisco-ai-defense/skill-scanner v2.0.9. Open-source Skills security scanner with YARA, behavioral dataflow, and pipeline taint analysis..

[7] Yue Deng, Wenxuan Zhang, Sinno Jialin Pan, and Lidong Bing. 2024. Multilingual Jailbreak Challenges in Large Language Models. In *NAACL*.

[8] OWASP Foundation. 2025. OWASP Agentic AI Security Initiative – Top 10 Risks. https://genai.owasp.org/resource/agentic-ai-threats-and-mitigations/

[9] PyCQA. 2024. Bandit: Security Linter for Python. https://github.com/PyCQA/bandit

[10] Semgrep, Inc. 2024. Semgrep: Lightweight Static Analysis. https://semgrep.dev

[11] Nakatani Shuyo. 2014. langdetect: Language Detection Library. https://github.com/Mimino666/langdetect

[12] Snyk. 2026. Snyk Agent Scan: Security Scanner for AI Agents and Skills. https://github.com/snyk/agent-scan CLI with –skills flag for Skill scanning via API verification..

[13] Snyk Labs. 2026. ToxicSkills: Snyk Finds Prompt Injection in 36%, 1,467 Malicious Payloads in Agent Skills Supply Chain. https://snyk.io/blog/toxicskills-malicious-ai-agent-skills-clawhub/ February 5, 2026. Audit of 3,984 Skills..

[14] Wenxuan Wang et al. 2024. All Languages Matter: On the Multilingual Safety of Large Language Models. *arXiv preprint arXiv:2310.00905* (2024).

[15] Zheng-Xin Yong, Cristina Menghini, and Stephen H. Bach. 2024. Low-Resource Languages Jailbreak GPT-4. In *NAACL*.

[16] Qiusi Zhan et al. 2024. InjecAgent: Benchmarking Indirect Prompt Injections in Tool-Integrated LLM Agents. In *ACL Findings*.