Context Matters: Repository-Aware Security Analysis of the Agent Skill Ecosystem

Florian Holzbauer
Interdisciplinary Transformation University (IT:U)
Linz, Austria
florian.holzbauer@it-u.at

David Schmidt
University of Vienna, CDL AsTra
Faculty of Computer Science
Vienna, Austria
d.schmidt@univie.ac.at

Gabriel K. Gegenhuber
Interdisciplinary Transformation University (IT:U)
Linz, Austria
gabriel.gegenhuber@it-u.at

Sebastian Schrittwieser
University of Vienna, CDL AsTra
Faculty of Computer Science
Vienna, Austria
sebastian.schrittwieser@univie.ac.at

Johanna Ullrich
Interdisciplinary Transformation University (IT:U)
Linz, Austria
johanna.ullrich@it-u.at

Abstract

Agent skills extend local AI agents, such as Claude Code and OpenClaw, with additional functionality. Their growing popularity has led to dedicated marketplaces resembling mobile app stores, as well as automated scanners that assess whether skills are benign or malicious. However, scanner reports from individual marketplaces classify up to 46.8% of skills as malicious, raising concerns about false positives. We present the largest empirical security analysis of the AI agent skill ecosystem to date. We collect 238,180 unique skills from three major distribution platforms and GitHub, and analyze their contents, behavior, and repository context. Unlike existing scanner-based assessments, which evaluate skills largely in isolation, our repository-aware analysis checks whether a flagged skill is consistent with its surrounding GitHub project. This context substantially reduces the number of suspicious skills: only 0.52% remain suspicious after repository-aware analysis. Our results show that existing scanners can substantially overestimate maliciousness when repository context is ignored. At the same time, we identify previously undocumented real-world attack vectors, including the hijacking of skills hosted in abandoned GitHub repositories. Overall, our findings provide a more robust view of the agent-skill ecosystem's current risk surface and highlight the need for context-aware security evaluation.

CCS Concepts: • **Security and privacy**;

Keywords: Security, Privacy, Agent Skills, Skill Classification

Agent Skills '26, San José, CA

ACM Reference Format:
Florian Holzbauer, David Schmidt, Gabriel K. Gegenhuber, Sebastian Schrittwieser, and Johanna Ullrich. 2026. Context Matters: Repository-Aware Security Analysis of the Agent Skill Ecosystem. In *Agent Skills '26 Workshop: ACM Conference on AI and Agentic Systems, May 26, 2026, San José, CA*. ACM, New York, NY, USA, 10 pages.

1 Introduction

Autonomous AI agents, such as Claude Code [3] or OpenClaw [34] extend large language models (LLMs) from standalone text generation systems into truly autonomous, closed-loop assistants that can plan, act, and learn complex tasks. In a nutshell, the LLM interprets user requests, invokes adequate routines to serve them, and eventually integrates the results into subsequent reasoning steps. A key concept are skills, which are reusable, modular components that extend an agent's capabilities like access to an external API, code execution, or data retrieval. Following a standardized and open format [2], skills consist of natural language descriptions, informing the user and the LLM about its capabilities, in combination with executable logic implementing the functionality. Skills are found on dedicated skill markets, e.g., ClawHub [33], Skill.sh [35], and SkillDirectory [31], or traditional repositories like Github. Agents might even discover them on their own, e.g., when reading on moltbook.com, a reddit-like platform for bots.

On the one hand, the tight integration of LLM reasoning with execution capabilities poses unique risks as anecdotal evidence of undesired email deletion emphasizes [9]. A recurring pattern, on the other hand, are supply chain risks by integrating external resources for a system's extended functionality from market places or other sources. Examples are machine images in compute clouds [8], docker hub for containers [30], mobile applications for iPhones [27], or packet managers like npm and PyPI [37], and nowadays, it appears, also skills for AI agents. Among others, malicious skills attempted to steal private information from macOS [23]

or redirect cryptocurrency assets [19]. In consequence, skill markets nowadays automatically scan the provided skills for security, and provide the results for orientation to their users. The total share of malicious skills, however, varies significantly among market places – 46.8% (ClawHub), 23.0% (Skills.sh), and 6.0% (SkillsDirectory).

In this paper, we present the largest empirical security study of the AI agent skill ecosystem, collecting and analyzing 238,180 skills from three distribution platforms and GitHub. Our study revisits the high maliciousness rates reported by existing marketplaces and asks whether these classifications remain meaningful once skill and repository context are considered. We structure the analysis around the following research questions.

RQ1 *What skills are shared on marketplaces, and which new attack vectors emerge from the skill ecosystem?*

RQ2 *How do marketplaces and scanners classify skills as malicious?*

RQ3 *Can repository context improve existing security classifications of skills?*

We first characterize the collected skills across marketplaces and GitHub, including their scripts, embedded artifacts, and distribution structure. We then analyze how marketplace scanners and the Cisco Skill Scanner [10] classify skills as malicious and compare the consistency of their detections. Finally, we reevaluate scanner-flagged skills by incorporating the surrounding GitHub repository context and measuring whether the repository documentation and code align with the skill specification. This repository-aware analysis shows that isolated skill scanning substantially overestimates the ecosystem's risk. At the same time, our broader analysis uncovers structural weaknesses in skill distribution platforms, including repository hijacking risks that allow adversaries to take over references to existing skills. Summarizing, our paper contributes the following aspects:

- **Large-scale ecosystem measurement.** With 238,180 unique skills, we construct the largest cross-platform dataset of agent skills to date by collecting skills from three official marketplaces as well as GitHub repositories. The dataset not only facilitates the analysis at hand, but also provides a basis for future longitudinal studies of the AI skill ecosystem.
- **Repository-aware skill analysis.** Existing security scanners classify a large share of offered skills as malicious. We conduct a semantic analysis that incorporates not only the skill description, but also the surrounding repository context. Among 2,887 scanner-flagged skill and repository combinations, only 15 remain associated with suspicious repositories, corresponding to 0.52%. This substantially reduces the number of likely false positives and provides a more contextualized view of the ecosystem's risk surface.
- **Discovery of new attack vectors.** We identify previously undocumented attack vectors in the AI skill ecosystem. In particular, we demonstrate repository hijacking risks for seven abandoned repositories referenced by skill indexes, affecting 121 skills. One affected skill has more than 1,000 recorded installations, a significant number considering the recency of the ecosystem.

To enable reproducibility and future work, we publish our code: https://github.com/holzsec/repository-context-agentskills/.

2 Background and Related Work

Agent and Skills. AI agents autonomously execute tasks in interaction with external services, and operate in a reasoning-action loop. The LLM interprets a user request, selects and invokes an adequate capability, and eventually incorporates the results into subsequent reasoning. Many agents support modular extensions, such as API access, code execution, or data retrieval, that are referred to as *skills*. Skills typically come in a repository, combining a specification file (`SKILL.md`) with optional scripts, configuration files, or static assets. The specification file describes the skill's capabilities and its invocation context in natural language, enabling the autonomous agent to decide, while the latter represent the executable logic. For interoperability, Anthropic recently specified a skill packaging format [22].

Skill Marketplaces. Agent skills are distributed over dedicated marketplaces such as ClawHub [33], Skills.sh [35], and SkillsDirectory [31]. As of March 9, 2026, they provide 18,412, 86,800, and 36,109 skills, respectively. Yet, the platforms differ, leading to different levels of control for the operators: ClawHub curates and reviews uploaded skills, and hosts them itself. Skills.sh adopts an open Git-based distribution model and indexes skills in external repositories. SkillsDirectory also refers to external repositories, but moderates submissions and performs rule-based security scanning.

Security on Marketplaces. Malicious skills are known to manipulate agent behavior and after reports on their distribution over marketplaces [23, 19, 5, 28], marketplaces nowadays automatically scan the offered skills for security. The results, typically a classification of whether a skill is benign or malicious in combination with a short explanation on the reasoning, are provided as metadata to their users. Therefore, ClawHub relies on VirusTotal [36] and a custom LLM-based detection system, Skills.sh integrates several third-party scanners, and SkillsDirectory reports the use of more than 50 rule-based detection mechanisms. The share of skills reported as suspicious varies substantially across marketplaces, namely 46.8% (ClawHub), 23% (Skills.sh), and 6% (SkillsDirectory). Across all marketplaces, the share remains

high, which indicates either a large number of malicious skills or a high false positive rate.

Empirical Studies on the Skill Ecosystem. Studies by third parties come to a similar share of malicious skills. An analysis of 3,984 skills on ClawHub and skills.sh [5] found 13.4% of them having a critical-level security flaw like malware distribution, or prompt injection, and 36.82% show (more minor) security pitfalls like hard-coded API keys or insecure credential handling. Another analysis investigated 31,132 skills [21] from skills.rest and skillsmp.com, and found that 26.1% of the skills contain a security vulnerability such as prompt injection, and data exfiltration. The largest study by now investigated 40,285 skills from skills.sh [20]. While predominantly focusing on their publication behavior over time as well as prompt length, the author also assessed their security and concluded 9% of them having critical flaws. In this context, also multiple skill scanners emerged, e.g., SkillScan [21], SkillFortify [7], Snyk [6], and the Cisco skill scanner [10]. Also, multiple works promise annotated data sets of skills for security benchmarking [7, 21]. However, upon inspection, we were unable to find them, impeding a direct comparison of our analysis with those from previous work on the same data sets.

Security of AI Agents. Instead of skills, vulnerabilities might also directly affect the agent. (Meanwhile fixed) *ClawJacked* enabled an attacker to gain control over Open Claw instances using a web socket to localhost [19]. Alternatively, attackers could use web sockets to modify log files that the AI agents eventually rely on for troubleshooting [32]. Via prompt injection, OpenClaw was persuaded to reveal private keys [11]. In the manner of social engineering, moltbook appears to be exploited to extract metadata for reconnaissance. The programming agent Claude Code was meanwhile also vulnerable to remote code execution [12], and revealed API keys [13]. Finally, Shodan currently discovers 55,561 of such OpenClaw instances on the Internet [29], and a honeypot provides a glimpse into the attackers' current strategies [16].

Scientific Literature. Due to the recency of the topic, most reports appear in non-scientific venues, e.g., blogs or as unaccepted preprints. Yet, scientific literature already discusses autonomous AI agents [1, 25, 4]. Researchers consider security both a potential application of these systems and a major challenge for them. On the one hand, autonomous AI can continuously monitor malicious activities and immediately block them, which can improve security. On the other hand, these systems can themselves pose security threats because they combine autonomy with access to large and potentially sensitive datasets. A dedicated survey on security [14] classified the threat landscape of AI agents, also including supply chain threats.

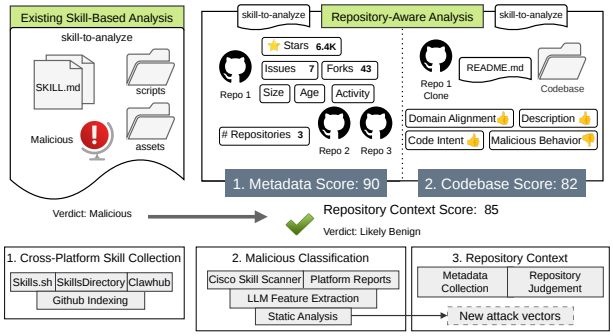

Figure 1. Overview of our repository-aware skill analysis to reduce the high number of false positives. Our approach consists of three stages, encompassing skill collection, malicious classification, and repository context analysis.

3 Methodology

We study the security of agent skills using a three-stage measurement pipeline, illustrated in Figure 1. Our methodology is designed to capture the breadth of the emerging skill ecosystem, compare how existing tools assess skill risk, and evaluate whether repository context can help interpret scanner alerts. Rather than treating any individual scanner as ground truth, we use scanner outputs as security signals that require contextual interpretation.

3.1 Cross-Platform Skill Collection

To answer RQ1, we collect agent skills from multiple sources, including public skill marketplaces and GitHub repositories. This cross-platform collection allows us to study both curated or indexed marketplace skills and skills that are published independently by developers. Since marketplaces differ in how they host and reference skills, we normalize all collected artifacts into a common representation based on the skill directory and its associated files.

To extend coverage beyond known marketplaces, we search public GitHub activity data for repositories likely to contain skill definitions. Candidate repositories are cloned and scanned for `SKILL.md` files, which serve as the entry point for identifying skill artifacts. We apply resource limits during collection to ensure scalability and deduplicate skills using content hashes to avoid counting identical artifacts multiple times.

After collection, we perform static analysis over each skill artifact. We enumerate contained files, record file types and directory structure, and extract ecosystem-level properties such as the presence of executable scripts. We further scan skill artifacts for embedded secrets and validate detected credentials where possible. This analysis provides the basis for characterizing the structure, content, and potential exposure risks of the skill ecosystem.

3.2 Scanner-Based Skill Assessment

To answer RQ2, we evaluate how existing security mechanisms classify agent skills. We collect scanner reports exposed by skill marketplaces and complement them with platform-independent analyses. This includes an open-source skill scanner and an LLM-based behavioral feature extraction pipeline.

The scanner-based analysis serves two purposes. First, it allows us to measure how frequently skills are flagged by different tools. Second, it enables us to compare the consistency of these tools by analyzing overlap between their detections. Because the ecosystem lacks a reliable ground-truth dataset of malicious skills, we interpret scanner results as alerts rather than definitive labels. This distinction is important because different scanners rely on different assumptions, rules, and behavioral models, which can lead to inconsistent classifications.

Our LLM-based analysis extracts structured behavioral indicators from each skill. Because the full skill contents exceeded the LLM's context window, the analysis used a LLM-based auxiliary script to extract relevant information from the skill files. The prompt evaluates whether a skill exhibits properties associated with risky behavior, such as system interaction, network communication, credential handling, persistence, or abuse potential. These features complement deterministic scanner outputs and provide a platform-independent basis for comparing skills across marketplaces and repositories.

3.3 Repository-Aware Contextual Analysis

We analyze whether skills flagged by scanner-based methods remain suspicious when evaluated in the context of their surrounding repositories to answer RQ3. This step is motivated by the fact that scanners typically inspect skills in isolation, although many skills are embedded in larger projects whose documentation, source code, and development history can explain their behavior.

We therefore perform repository-aware analysis for high-risk scanner alerts where GitHub repository context is available. For each selected skill, we collect repository metadata and relevant codebase context. We then derive two complementary signals. The first captures codebase alignment, i.e., whether the skill's described functionality is consistent with the surrounding repository documentation and implementation. The second captures repository maturity, based on metadata such as project age, activity, size, and popularity.

We combine these signals into a repository-context score. The score is intended as a contextual trust and triage signal rather than a definitive maliciousness label. A low score indicates that a flagged skill is weakly supported by its repository context, while a high score indicates that the repository provides evidence that the flagged behavior may be expected or benign in context. For skills appearing in multiple repositories, we aggregate context scores across representative repositories to account for the fact that the same skill may appear in different environments.

Finally, we validate the repository-aware analysis through manual inspection of a sample of flagged repositories. This validation assesses whether repositories identified by scanners as suspicious appear benign or suspicious to independent reviewers and whether the automatically derived context scores are consistent with human judgments.

Table 1. Overview of collected agent skills from ClawHub [33], SkillDirectory [31], Skills.sh [35], and GitHub. Retrieved denotes successfully downloaded skills retained for analysis; Added denotes skills retained after cross-source deduplication. For Skills.sh, the crawl retrieved 55,366 listed skills and 77,456 additional skills extracted from referenced repositories.

Skill Metric	ClawHub	SkillsDir.	Skills.sh	GitHub
Indexed	16,755	32,896	79,735	142,824
Retrieved	16,755	17,611	125,928	136,095
Added	16,755	17,611	112,231	91,583
Owners	n/a	709	7,950	14,197
Repositories	n/a	766	9,431	16,413
#Total 238,180 (distinct analyzed skills)				

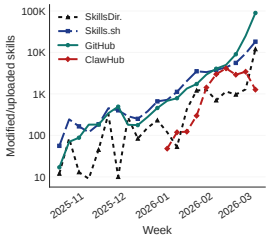

Figure 2. All platforms show an increase in number of weekly updated agent skills over time.

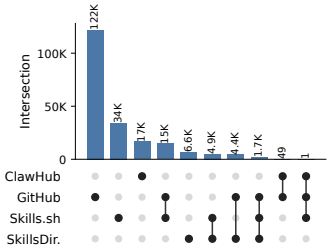

Figure 3. Overlap of skills retrieved from the marketplaces.

4 RQ1: Cross-platform Skill Analysis

We first present results on the differences between the marketplaces and the additional skills that we discovered on GitHub. We then provide an overview of the content of the published skills, and demonstrate how attackers can hijack skills referenced in two marketplaces.

4.1 Collected Skill Dataset

Table 1 summarizes the skills that we crawled from the different marketplaces and the subset that we successfully downloaded and analyzed. In total, we indexed 16,755 skills from ClawHub, 79,735 marketplace-listed skills from Skills.sh,

32,896 skills from SkillDirectory, and 142,822 skills referenced on GitHub. For Skills.sh, 55,366 of the indexed marketplace entries could be retrieved directly. In addition, we extracted 77,456 further skill folders from repositories referenced by Skills.sh, resulting in 125,928 analyzed Skills.sh skills in total.

During collection, we encountered several issues that limited the number of retrievable skills. For SkillDirectory, a large repository subdirectory that effectively represents a separate skill collection was omitted, resulting in 13,304 skills not being retrieved. In addition, changes in repository structures caused skill paths referenced by the marketplaces to become invalid. Specifically, 21,800 skills referenced by marketplace indexes were no longer present in the repositories, and another 3,991 repositories no longer contained skills referenced by *Skills.sh*. Furthermore, 188 repositories had become private or required authentication at the time of collection. These issues highlight the dynamic nature of the ecosystem and the challenges of reliably archiving skill datasets. Figure 2 further shows the number of skills added to these platforms each week since November 2025.

In total, we successfully downloaded and analyzed 16,755 skills from ClawHub, 125,928 from Skills.sh, 17,611 from SkillDirectory, and 136,095 from GitHub. In Figure 3, we show the overlap of skills across the platforms. The figure illustrates that a substantial fraction of skills appears on multiple marketplaces, indicating that many platforms reference the same underlying repositories. In particular, GitHub acts as the primary hosting platform for most skills, while the marketplaces serve as discovery layers. At the same time, each marketplace contributes skills not listed in the others, reflecting differences in indexing scope and update frequency. Overall, aggregating multiple marketplaces increases coverage of the skill ecosystem and results in a final dataset of 238,180 cross-marketplace unique skills for further analysis.

The marketplaces also differ in their ecosystem structure. Skills.sh references 7,950 owners and 9,431 repositories, while the GitHub dataset spans 14,197 owners and 16,413 repositories. SkillDirectory is comparatively smaller, with 709 owners and 766 repositories. These numbers indicate that many repositories host multiple skills, suggesting that developers frequently group related skills within a single project rather than publishing them individually.

Skill Content. We further analyzed which scripts skills contain and provided a table in our artifact [18]. Across all marketplaces, Python scripts appear most frequently, followed by shell scripts, JavaScript, and TypeScript. However, the share of skills that include at least one script differs across marketplaces. While Skills.sh, SkillsDirectory, and GitHub have a similar range of skills containing scripts (11.8% to 15.7%), ClawHub shows a substantially higher share of skills including at least one script (44.1%). One explanation for this difference could be that ClawHub more specifically targets OpenClaw instead of general agents.

We further evaluate whether scripts reside in a directory named `scripts/`, as defined by the specification [22]. Again, ClawHub represents an outlier. Among skills that include scripts from ClawHub, 13.2% lack a `scripts/` directory. In contrast, only 2.9% to 3.4% of skills from the other marketplaces contain scripts without the corresponding directory.

Secrets. We further analyzed whether skills contain valid tokens and credentials. In total, we discovered 12 functional credentials, including four for the NVIDIA API, three for ElevenLabs, two Gemini tokens, two MongoDB credentials, and one credential each for Amplify, Postgres, and X AI. Attackers could abuse these credentials to access third party services and perform actions on behalf of the credential owner. For example, NVIDIA, ElevenLabs, Gemini, and xAI token allow access to AI services, which attackers could use to issue requests that incur costs for the owner. One reason for the relatively small number of discovered secrets may be that most skills are hosted on GitHub. In contrast to mobile apps [26] or accessible storage buckets [15], developers are likely more aware that the code is publicly visible and that attackers could access any embedded secret.

4.2 Skill Provisioning

For the security of distributed skills, similar considerations as for dependency management systems apply. Attackers can hijack dependencies if the system does not host the dependency itself and the URL hosting it can be taken over, for example because a username on GitHub was renamed [26]. In addition, the authentication mechanism used to publish a skill plays an important role, as weak or missing authentication can enable attackers to hijack existing dependencies or skills. We therefore looked into the currently implemented authentication mechanisms for publishing skills and their distribution. For authentication, ClawHub and SkillsDirectory rely on GitHub authentication, while Skills.sh provides no authentication. Instead, the marketplace adds skills when users download them using the command line tool with telemetry enabled. In this sense, the system resembles Go modules, which also do not implement a separate authentication mechanism but cache dependencies [17]. However, instead of caching or redistributing the skills, Skills.sh directly downloads them from GitHub. This design can enable attackers to hijack existing repositories if the previous owner renames their account and the repository has not yet reached the required download threshold that would cause GitHub to retire the repository name [27]. The same issue also affects SkillsDirectory. Although SkillsDirectory provides the option to download skills from its website, the command line tool currently attempts to download the skill from GitHub. In contrast, ClawHub directly distributes the skill. This design

reduces the dependency on third-party URL management and therefore decreases the risk of repository hijacking.

Skills Vulnerable to Hijacking. To test whether GitHub mitigates repository hijacking for vulnerable skills, we created test accounts using the associated usernames and entered the repository names without creating the repositories. This approach keeps existing redirects functional while revealing whether an attacker could recreate the repository under the same name [27]. To prevent attackers from hijacking vulnerable skills, we keep the account names associated with vulnerable skills reserved. We performed this step for all identified repositories with five or more stars, as the process of registering GitHub accounts cannot be automated.

Overall, we discovered 121 skills that forward to seven vulnerable repositories. Among them, 77 skills indexed by Skills.sh reference five vulnerable repositories, while 44 skills listed on SkillsDirectory reference two additional vulnerable repositories. One hijackable repository referenced by SkillsDirectory has 159 stars, whereas the maximum star count among vulnerable repositories referenced by Skills.sh is 48. Using the download statistics provided by Skills.sh, we further assessed how frequently hijackable skills were downloaded. The median number of downloads is 25, while the most often downloaded skill reached 2,032 downloads.

We responsibly disclosed this attack vector to the affected platforms and recommended switching to a direct distribution model similar to OpenClaw.

New Ecosystems, Old Issues. Based on the source code of ClawHub [24], we implemented a crawler for skill and security reports. During this process, we discovered that the associated endpoint returns additional owner metadata. In particular, the API exposes the email address associated with each user's GitHub account. This information is not shown by default on GitHub profiles and is also not visible through the ClawHub website. Therefore, we did not expect the ClawHub API to disclose this data.

5 RQ2: Malicious Classification

To answer (RQ2), we study how existing security scanners classify agent skills and evaluate the consistency of their maliciousness assessments. We compare scanner reports from skill marketplaces with the results of our own analysis pipeline, which includes the Cisco Skill Scanner and an LLM-based behavioral classifier. This comparison allows us to quantify how different scanning approaches interpret the same skills and to identify potential overclassification of malicious behavior. Understanding these differences is important because high false positive rates may reduce trust in the ecosystem, confuse end-users, and motivate the need for repository-aware analysis.

Malicious Classification Rates. We compare the malicious classification rates reported by existing marketplace scanners with the results of our own analysis pipeline. Across both platforms, we observe substantial differences between scanners. On Clawhub, the OpenClaw scanner flags up to 41.93% of skills as suspicious, while VirusTotal reports a similar rate of 36.20%. Our GPT-5.3 based approach produces comparable results, classifying 38.8% of skills as malicious. In contrast, the Cisco Skill Scanner reports a significantly lower fail rate of 16.74%. These differences highlight that the perceived security of the skill ecosystem strongly depends on the chosen scanning approach.

Table 2. Comparison of security scanners used in the skill ecosystem. The table contrasts scanners deployed on ClawHub and Skills.sh (highlight in gray) with Cisco's skill scanner and our LLM-based feature set.

	Scanner	Scanned	Pass	Fail	Fail Rate
Clawhub	VirusTotal	12,213	7,792	4,421	36.20%
	OpenClaw Scanner	14,244	8,271	5,973	41.93%
	GPT 5.3-based	16,424	10,050	6,374	38.8%
	Cisco Skill Scan	16,745	13,941	2,804	16.74%
Skills.sh	agent-trust-hub	62,163	53,611	8,552	13.76%
	snyk	46,414	42,843	3,571	7.69%
	socket	56,695	54,544	2,151	3.79%
	GPT 5.3-based	52,577	38,234	14,343	27.28%
	Cisco Skill Scan	52,577	45,196	7,381	14.04%

The discrepancy between scanners is more pronounced when comparing the two marketplaces. On Clawhub, fail rates range from 16.7% to 41.9%, suggesting that a substantial fraction of skills may exhibit potentially suspicious behavior. In contrast, scanners deployed on Skills.sh report much lower fail rates, ranging from 3.79% to 13.76%. Our own analysis tools show similar trends: the GPT-5.3-based analysis classifies 27.28% of Skills.sh skills as suspicious, while the Cisco scanner flags 14.04%. These discrepancies indicate that existing scanners produce inconsistent classifications when they analyze skills in isolation. In particular, scanners that rely on behavioral heuristics or language model reasoning flag substantially larger portions of the ecosystem as suspicious. Such elevated fail rates can reduce trust in the ecosystem and suggest that many skills are misclassified as malicious. This observation motivates the need for additional context to enable more accurate classification of skill behavior.

Cross-Scanner Agreement. To evaluate how consistently scanners classify skills as malicious, we compare the results of five scanners on the subset of 27,111 Skills.sh skills analyzed by all tools. Figure 4 shows the conditional overlap between scanners, expressed as the probability that a skill flagged by scanner A is also flagged by scanner B. Overall, agreement between scanners is low and often asymmetric. For example, 33% of skills flagged by the Cisco Skill Scanner

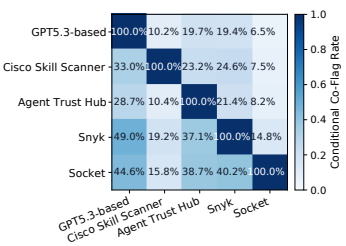

Figure 4. Conditional scanner agreement on Skills.sh common skills.

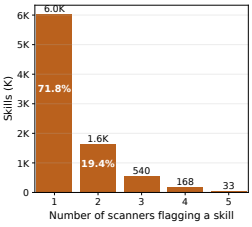

Figure 5. Number of Skills.sh common skills flagged by exactly $k \in \{1, \ldots, 5\}$ scanners.

Table 3. Skill.sh malicious detection rates for skills and repositories (skill flagged if at least one scanner flagged, repository malicious if at least one skill is malicious)

Category	Total	Flagged	Rate
Skills (overall)	62,219	12,004	19.29%
Skills (repo stars > 1000)	7,725	1,656	21.44%
Skills (installs > 1000)	755	122	16.16%
Repositories (overall)	8,451	3,878	45.89%
Repositories (stars > 1000)	528	268	50.76%
Repositories (installs > 1000)	171	105	61.40%

are also flagged by the GPT-5.3-based analysis, whereas only 10.2% of GPT-5.3 detections overlap with Cisco. Similar patterns appear across other scanner pairs, indicating that scanners frequently identify different sets of skills as suspicious. Figure 5 illustrates the distribution of detections across scanners. Among the 8,402 skills flagged by at least one scanner, most are flagged by a single scanner, 6,032 skills. In contrast, 1,629 skills are flagged by two scanners and 540 by three scanners. Only 168 skills are flagged by four scanners, and just 33 skills are flagged by all five scanners. The limited overlap shows that scanner consensus is rare and that most detections lack corroboration by other tools.

Repository-level Classification Rates. To better understand how scanner results translate from individual skills to repositories, we aggregate skill-level detections at the repository level. Table 3 shows the resulting malicious classification rates for both skills and repositories on Skills.sh. A skill is considered malicious if at least one scanner flags it, while a repository is classified as malicious if any of its contained skills is flagged. Overall, 19.29% of skills are flagged as malicious. When focusing on popular skills, the rates remain comparable: 21.44% for skills hosted in repositories with more than 1,000 stars and 16.16% for skills with more than 1,000 installs. However, the picture changes when aggregating these detections at the repository level. Nearly half of all repositories (45.89%) contain at least one flagged skill. This

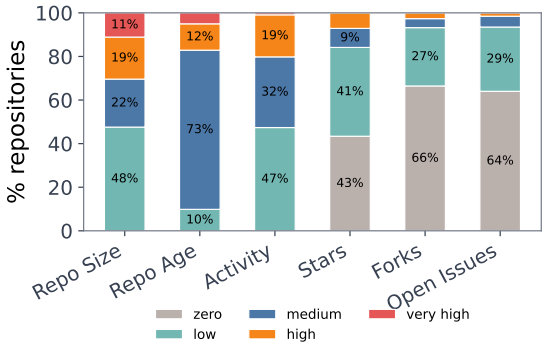

Figure 6. Metadata scores for unique repositories.

proportion increases further for popular repositories, reaching 50.76% for repositories with more than 1,000 stars and 61.40% for repositories associated with highly installed skills. These results suggest that even well-known repositories are frequently classified as malicious when applying strict aggregation rules. The effect is driven by repositories containing multiple skills: as the number of skills per repository increases, the probability that at least one skill is flagged also increases. Consequently, repository-level aggregation can substantially amplify skill-level detections and may overstate the prevalence of malicious repositories in the ecosystem.

6 RQ3: Repository-Aware Analysis

To answer RQ3, we analyze whether scanner flagged skills remain suspicious when evaluated in the context of their surrounding GitHub repositories. We focus on skills flagged by both the Cisco Skill Scanner, with severity high or critical, and our GPT-5.3 based analyzer, with a score greater than 3. This yields 8,153 flagged skill and repository combinations.

We exclude ClawHub skills because they lack GitHub repository context, and skills located in repository roots because they only allow metadata based analysis. From the remaining set, we randomly sample 3,000 skill and repository combinations and collect repository metadata together with full repository clones. Cloning failed in 113 cases, leaving 2,887 combinations for codebase evaluation.

Metadata Score. We first analyze repository metadata, including size, age, activity, popularity, and issue activity. Figure 6 shows that repositories containing flagged skills are typically small and have limited popularity: 47.6% are smaller than 2 MB, 43.4% have no stars, 66.5% have no forks, and 64.0% have no open issues. At the same time, many repositories remain active, with 47.4% updated within the last week. Compared to a random set of 1,500 repositories with matching marketplace distribution, repositories containing flagged skills do not exhibit distinct metadata characteristics. This suggests that metadata differences are driven more by marketplace composition than by suspicious skills.

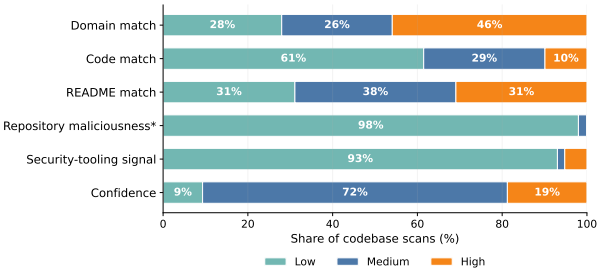

Figure 7. Codebase score category results. Categories marked with * indicate that lower values are better.

Codebase Score. To evaluate whether a flagged skill aligns with its repository, we use an evidence based prompt that considers domain alignment, code similarity, README consistency, support signals, and repository level maliciousness. To control analysis cost, we limit the repository context to up to 200 lines from the `SKILL.md` and README files and up to three repository files with 100 lines each.

Most repositories provide sufficient context: 94.1% contain a README, 65.7% contain code, and 61.9% contain both. Repository context often supports the skill purpose: domain matching is high for 45.9% of skills and medium for 26.1%, meaning that roughly 72% show at least moderate thematic alignment. Direct code alignment is lower, with 9.9% high and 28.6% medium similarity. Repository level maliciousness is rare, with 98.0% of repositories in the lowest maliciousness category and only two repositories showing high maliciousness signals. These results suggest that many scanner alerts are false positives caused by analyzing skills without repository context.

Repository Context Score. We combine the codebase score and metadata score into a repository context score, weighted 70% and 30%, respectively. As shown in Figure 8, the combined score has a mean of 58.5. The codebase score is higher, with a mean of 65.1, while the metadata score is lower, with a mean of 42.9, reflecting the young and low popularity repositories in the ecosystem.

We divide the repository context score into three categories. Only 121 cases (4.2%) fall below 40, indicating weak repository embedding or low repository maturity. The largest groups are the intermediate category from 40 to below 60, with 1,373 cases (47.6%), and the high category above or equal to 60, with 1,393 cases (48.3%). Overall, most scanner flagged skills show moderate to strong repository linkage.

Suspicious Repositories. After repository aware analysis, only 15 skill and repository combinations remain suspicious. These cases correspond to repositories where the skill aligns with the codebase, but the repository itself appears suspicious and is not categorized as a security tool. This represents 0.52% of the 2,887 evaluated combinations. These

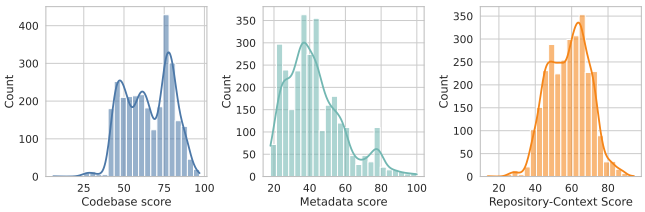

Figure 8. Repository context score and its weighted subcomponents, with 70% codebase score and 30% metadata score.

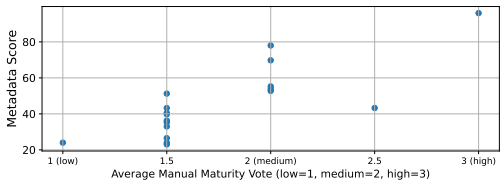

Figure 9. Comparison of maturity scores assigned by manual reviewers and our repository aware scoring approach.

repositories have similar metadata scores to the full set, 41.1 versus 42.9, but lower codebase scores, 51.5 versus 65.2. This shows that repository context substantially reduces the suspicious set while still separating suspicious environments from benign scanner alerts.

Validation. Because no public ground truth dataset labels repositories as malicious with respect to agent skills, we validate our results through manual inspection. Two independent researchers reviewed 20 randomly sampled repositories from the flagged set. Of the 18 repositories that were still available, both reviewers classified all as benign based on documentation, code structure, and functionality. This supports our interpretation that many scanner flagged skills do not appear malicious once their repository context is considered.

Reviewers also assessed repository maturity. Their judgments were more heterogeneous, but the averaged manual maturity scores show a positive relationship with our metadata score in Figure 9. This indicates that the metadata component captures repository maturity signals that are also visible to human reviewers, although it should not be interpreted as a definitive maliciousness label.

7 Discussion

Skill Marketplaces. Our cross-platform analysis shows that agent skill marketplaces largely act as discovery layers on top of GitHub rather than as independent distribution channels. This design increases ecosystem coverage and makes skill publication lightweight, but it also inherits risks from the underlying hosting platform. Broken repository references, renamed accounts, deleted repositories, and private repositories already affected our data collection. More

importantly, skills that are referenced rather than mirrored remain vulnerable to repository hijacking when abandoned GitHub namespaces can be re-registered. These findings show that the security of skill marketplaces depends not only on skill contents, but also on how skills are provisioned, authenticated, and archived.

Skill Scanners. The scanner results in RQ2 show that malicious classification rates vary substantially across tools and marketplaces. Some scanners classify large fractions of skills as suspicious, yet cross-scanner agreement is low and most flagged skills are detected by only one tool. This indicates that scanner outputs should be interpreted as alerts rather than ground-truth labels. Strictly aggregating these alerts at the repository level further amplifies the problem: repositories containing multiple skills are more likely to be classified as malicious simply because they provide more opportunities for at least one skill to trigger a scanner. This effect can overstate the prevalence of malicious repositories, including for popular projects.

Repository Context. The repository-aware analysis in RQ3 addresses this limitation by evaluating scanner-flagged skills within their surrounding project context. Many skills that appear suspicious in isolation are embedded in repositories whose documentation, codebase, and stated purpose align with the skill functionality. As a result, only 15 skill–repository combinations remain suspicious after repository-aware analysis, corresponding to 0.52% of the evaluated sample. This does not imply that scanners are unnecessary; rather, it shows that users could be provided with more context than isolated skill scanning. Repository context provides an additional signal that can distinguish suspicious behavior from legitimate functionality.

New Ecosystem, Familiar Risks. Although agent skills are a new distribution format, many of the observed risks mirror earlier software supply-chain problems. Similar to package managers, container registries, and mobile app stores, skill ecosystems face challenges around authentication, dependency ownership, abandoned projects, and embedded secrets. We found functional credentials in published skills and identified marketplace designs that can enable repository hijacking. These results suggest that the agent skill ecosystem should adopt established supply-chain defenses early, including immutable skill snapshots, stronger publisher authentication, namespace retirement, secret scanning, and clearer provenance metadata.

Practicality of Repository-Aware Scanning. Repository-aware analysis is more expensive than scanning a single `SKILL.md` file, but our cost measurements show that it is practical for periodic marketplace scans. In our sample, repository-context analysis cost around $0.0097 per skill–repository pair without caching and $0.0021 with prompt caching. The full sample of 3,000 skills cost approximately $24 including retries. This makes repository-aware scanning feasible for marketplace operators, especially when combined with caching, incremental scans, and prioritization of newly added or widely installed skills.

Limitations. Our repository-aware analysis does not establish a definitive ground truth of maliciousness. It reduces false positives by incorporating repository context, but sophisticated attackers could still craft repositories whose documentation and code appear benign while hiding malicious behavior elsewhere. Our manual validation is limited to a small sample and focuses on visible repository evidence. Moreover, unavailable repositories, failed clones, and dynamic marketplace references limit reproducibility. These limitations reinforce our central conclusion: scanner results should be treated as risk signals that require contextual interpretation, not as definitive labels.

8 Conclusion

We presented the largest empirical security analysis of the AI agent skill ecosystem to date, covering 238,180 unique skills from three marketplaces and GitHub. Our analysis revealed embedded secrets, marketplace weaknesses, and substantial inconsistencies between existing scanners. Malicious classification rates ranged from 3.8% to 41.9%, while agreement was limited: only 0.12% of commonly analyzed skills were flagged by all five tested scanners. To address this inconsistency, we proposed a repository aware scanning approach that adds GitHub repository context to skill assessment. Re-evaluating 2,887 scanner flagged skill and repository combinations, only 0.52% remained associated with suspicious repositories. Repository context improved interpretability, as most flagged skills were embedded in repositories whose documentation and code matched the skill functionality, while only 4.2% showed weak repository linkage. Finally, we uncovered structural weaknesses in skill marketplaces. We identified repository hijacking risks affecting 121 skills and found that ClawHub exposed sensitive developer metadata, including email addresses associated with GitHub accounts. These findings show that agent skill security cannot be assessed from skill descriptions alone, but requires repository context, stronger provenance guarantees, and established supply chain safeguards.

Acknowledgments

The financial support by the Austrian Federal Ministry of Economy, Energy and Tourism, the National Foundation for Research, Technology and Development, and the Christian Doppler Research Association is gratefully acknowledged.

References

[1] D. B. Acharya, K. Kuppan, and B. Divya. 2025. Agentic AI: Autonomous Intelligence for Complex Goals—A Comprehensive Survey. *IEEE Access*, 13. doi:10.1109/ACCESS.2025.3532853.

[2] Agent Skills. 2025. Agent Skills: A Simple, Open Format for Giving Agents New Capabilities. https://agentskills.io/home. Accessed: 2026-03-07.

[3] Anthropic. 2026. GitHub – Claude Code. https://github.com/anthropics/claude-code. Accessed: 2026-03-07.

[4] M. Basu. 2026. OpenClaw AI chatbots are running amok — these scientists are listening in. *Nature*, 650.

[5] L. Beurer-Kellner, A. Kudrinskii, M. Milanta, K. B. Nielsen, H. Sarkar, and L. Tal. 2026. Snyk Finds Prompt Injection in 36%, 1467 Malicious Payloads in a ToxicSkills Study of Agent Skills Supply Chain Compromise. Accessed: 2026-02-26. https://snyk.io/blog/toxicskills-malicious-ai-agent-skills-clawhub/.

[6] L. Beurer-Kellner, A. Kudrinskii, M. Milanta, K. B. Nielsen, H. Sarkar, and L. Tal. 2026. Technical Report: Exploring the Emerging Threats of the Agent Skill Ecosystem. https://github.com/snyk/agent-scan/blob/main/.github/reports/skills-report.pdf.

[7] V. P. Bhardwaj. 2026. Formal Analysis and Supply Chain Security for Agentic AI Skills. *arXiv preprint arXiv:2603.00195*.

[8] S. Bugiel, S. Nürnberger, T. Pöppelmann, A.-R. Sadeghi, and T. Schneider. 2011. AmazonIA: when elasticity snaps back. In *Proc. of ACM CCS*. doi:10.1145/2046707.2046753.

[9] H. Chandonnet. 2026. Meta AI alignment director shares her OpenClaw email-deletion nightmare:'I had to RUN to my MAC mini'. Accessed: 2026-03-08. https://www.businessinsider.com/meta-ai-alignment-director-openclaw-email-deletion-2026-2.

[10] Cisco AI Defense. 2026. Skill Scanner: Security Scanner for Agent Skills. https://github.com/cisco-ai-defense/skill-scanner. Accessed: 2026-03-07, V2.0.1.

[11] J. Cruz. 2026. OpenClaw (ex-Moltbot (ex-Clawdbot)): The AI Butler With Its Claws On The Keys To Your Kingdom. Accessed: 2026-03-04. https://www.bitsight.com/blog/openclaw-ai-security-risks-exposed-instances.

[12] CVE-2025-59536. 2025. Claude Code' startup trust dialog could lead to Command Execution attack. Accessed: 2026-03-08. https://www.cve.org/CVERecord?id=CVE-2025-59536.

[13] CVE-2026-21852. 2026. Claude Code Leaks Data via Malicious Environment Configuration Before Trust Confirmation. Accessed: 2026-03-08. https://www.cve.org/CVERecord?id=CVE-2026-21852.

[14] Z. Deng, Y. Guo, C. Han, W. Ma, J. Xiong, S. Wen, and Y. Xiang. 2025. AI Agents Under Threat: A Survey of Key Security Challenges and Future Pathways. *ACM Comput. Surv.*, 57, 7. doi:10.1145/3716628.

[15] S. El Yadmani, O. Gadyatskaya, and Y. Zhauniarovich. 2025. The File That Contained the Keys Has Been Removed: An Empirical Analysis of Secret Leaks in Cloud Buckets and Responsible Disclosure Outcomes. In *Proc of the Symposium on S&P*. doi:10.1109/SP61157.2025.00009.

[16] A. Fogel and E. Cohen. 2026. Caught in the Wild: Real Attack Traffic Targeting Exposed Clawdbot Gateways. Accessed: 2026-03-07. https://www.pillar.security/blog/caught-in-the-wild-real-attack-traffic-targeting-exposed-clawdbot-gateways.

[17] Y. Gu, L. Ying, Y. Pu, X. Hu, H. Chai, R. Wang, X. Gao, and H. Duan. 2023. Investigating Package Related Security Threats in Software Registries. In *Proc of the Symposium on S&P*. doi:10.1109/SP46215.2023.10179332.

[18] F. Holzbauer, D. Schmidt, G. Gegenhuber, S. Schrittwieser, and J. Ullrich. 2026. GitHub – Skill Scripts Table. https://github.com/holzsec/repository-context-agentskills/blob/main/tables/skill_content.pdf.

[19] R. Lakshmanan. 2026. ClawJacked Flaw Lets Malicious Sites Hijack Local OpenClaw AI Agents via WebSocket. Accessed: 2026-03-04. https://thehackernews.com/2026/02/clawjacked-flaw-lets-malicious-sites.html.

[20] G. Ling, S. Zhong, and R. Huang. 2026. Agent Skills: A Data-Driven Analysis of Claude Skills for Extending Large Language Model Functionality. *arXiv preprint arXiv:2602.08004*.

[21] Y. Liu, W. Wang, R. Feng, Y. Zhang, G. Xu, G. Deng, Y. Li, and L. Zhang. 2026. Agent Skills in the Wild: An Empirical Study of Security Vulnerabilities at Scale. *arXiv preprint arXiv:2601.10338*.

[22] Mintlify. Specification – Agent Skills. Accessed: 2026-03-04. https://agentskills.io/specification.

[23] A. Oliveira, B. Tancia, D. Fiser, P. Lin, and R. Reyers. 2026. Malicious OpenClaw Skills Used to Distribute Atomic macOS Stealer. Accessed: 2026-03-08. https://www.trendmicro.com/en_us/research/26/b/openclaw-skills-used-to-distribute-atomic-macos-stealer.html.

[24] openclaw. Skill Directory for OpenClaw. https://github.com/openclaw/clawhub. Accessed: 2026-03-07.

[25] A. K. Pati. 2025. Agentic AI: A Comprehensive Survey of Technologies, Applications, and Societal Implications. *IEEE Access*, 13. doi:10.1109/ACCESS.2025.3585609.

[26] D. Schmidt, S. Schrittwieser, and E. Weippl. 2025. Leaky Apps: Large-scale Analysis of Secrets Distributed in Android and iOS Apps. In *Proc. of ACM CCS*. doi:10.1145/3719027.3765033.

[27] D. Schmidt, S. Schrittwieser, and E. Weippl. 2026. Supply Chain Insecurity: Exposing Vulnerabilities in iOS Dependency Management Systems. arXiv: 2601.20638.

[28] D. Schmotz, L. Beurer-Kellner, S. Abdelnabi, and M. Andriushchenko. 2026. Skill-Inject: Measuring Agent Vulnerability to Skill File Attacks. *arXiv preprint arXiv:2602.20156*.

[29] Shodan. Shodan Search Enging – OpenClaw. Accessed: 2026-03-07; Archived at: https://archive.ph/3TNgq. https://www.shodan.io/search/report?query=product:openclaw.

[30] R. Shu, X. Gu, and W. Enck. 2017. A Study of Security Vulnerabilities on Docker Hub. In *Proc. of the ACM on Conference on Data and Application Security and Privacy*. doi:10.1145/3029806.3029832.

[31] Skills Directory. Agent Skills Directory. Accessed: 2026-03-04. https://www.skillsdirectory.com/.

[32] P. Steinberger. 2026. OpenClaw log poisoning (indirect prompt injection) via WebSocket headers. Accessed: 2026-03-04. https://github.com/openclaw/openclaw/security/advisories/GHSA-g27f-9qjv-22pm.

[33] P. Steinberger. ClawHub, the skill dock for sharp agents. Accessed: 2026-02-26. https://clawhub.ai/.

[34] P. Steinberger. OpenClaw – Personal AI Assistant. https://openclaw.ai/. Accessed: 2026-03-07.

[35] Vercel Labs. The Agent Skills Directory. Accessed: 2026-02-26. https://skills.sh.

[36] VirusTotal. https://www.virustotal.com. Accessed: 2026-03-09.

[37] J. Zhang, K. Huang, Y. Huang, B. Chen, R. Wang, C. Wang, and X. Peng. 2025. Killing Two Birds with One Stone: Malicious Package Detection in NPM and PyPI using a Single Model of Malicious Behavior Sequence. *ACM Transactions on Software Engineering and Methodology*, 34, 4. doi:10.1145/3705304.