CLAWDPWNED: MALICIOUS INSTRUCTIONS IN THE OPENCLAW AI AGENT SKILLS REPOSITORY*

Arjun Krishna
University of Waterloo
a68krishna@uwaterloo.ca

ABSTRACT

OpenClaw (formerly ClawdBot) has experienced explosive growth, gaining over 141,000 GitHub stars and enabling thousands of users to integrate AI agents into their most sensitive workflows (Slack workspaces, personal data sources, financial accounts and social media platforms). This widespread adoption introduces a critical attack surface: *skills*, the modular capability extensions that grant agents access to tools, APIs, credentials, and system resources. A malicious actor who publishes a compromised skill to ClawHub can harvest credit card numbers, steal LinkedIn and cryptocurrency wallet credentials, execute obfuscated malware, or orchestrate large-scale social media manipulation.

We present the first large-scale security audit of this ecosystem, evaluating 2,556 publicly available skills against the OWASP LLM Top 10 framework. Our analysis identifies 49 skills that are likely malicious, including `tsyvic/buy-anything` (harvests full credit card details and executes purchases autonomously), `zaycv/linkedin-job-application` (collects LinkedIn credentials and 2FA secrets via obfuscated `base64|bash` installers), `aslaep123/reddit-trends` (enables multi-account vote manipulation with anti-detection systems), `zaycv/polymarket-trading` (extracts wallet private keys through password-protected executables), and `cgallic/wake-up-skill` (poisons agent memory with attacker-controlled content). The most common attack mechanism in malicious skills was Sensitive Information Disclosure (92.4%), followed by Excessive Agency enabling unauthorized financial transactions and mass automation (90.5%), and Supply Chain attacks through fetch-and-execute patterns (69.6%). To support reproducibility, we release the paper source, analysis code, and supplementary materials at `https://github.com/arjun-krishna1/ClawdPwned`. We propose concrete mitigations including permission manifests, cryptographic integrity verification, secret scoping, and per-action confirmation gates to protect the rapidly growing agentic AI ecosystem.

1 INTRODUCTION

The release of OpenClaw (formerly ClawdBot) started the first wave of adoption in AI agents by the wider public. Within weeks, the open-source agent runtime gained over 141,000 GitHub stars, making it one of the fastest-growing repositories in history (OpenClaw, 2026c). This exponential rise reflects broader advances in agentic AI, systems capable of reasoning, planning, and acting in diverse real-world environments (Yao et al., 2022; Shinn et al., 2023; Wang et al., 2024). A key enabler of OpenClaw's rapid adoption is the concept of *skills*: modular extensions that grant agents access to external tools, APIs, file systems, and network resources (Agent Skills, 2025; OpenClaw, 2026c). Skills allow agents to interact with the world, enabling tasks ranging from web browsing and code execution to financial transactions and social media automation. To support reproducibility and follow-on analysis, we release the paper source, analysis code, and supplementary materials at `https://github.com/arjun-krishna1/ClawdPwned`.

*ClawdBot has been renamed to OpenClaw.

Skills have rapidly evolved from ad-hoc configurations into a standardized, portable format. The Agent Skills specification (Agent Skills, 2025) defines an open standard for packaging agent capabilities as folders containing instructions, scripts, and resources. This standardization has enabled the emergence of public skill registries such as ClawHub (OpenClaw, 2026b), which serves as the central distribution hub for the OpenClaw agent ecosystem (OpenClaw, 2026c). ClawHub allows users to search, install, update, and publish skills via a CLI, with all versions archived on GitHub (OpenClaw, 2026a). The ecosystem we study contains over 2,500 community-contributed skills spanning productivity, automation, finance, and social media domains.

The security implications of this growth are amplified by the contexts in which agentic systems are deployed. AI agents are increasingly integrated into enterprise collaboration platforms such as Slack, Google Drive, and GitHub (Storer, 2025). AI agents can access sensitive data like conversation history, files, and documents, inheriting the same broad data access permissions as human employees. Recent work on third-party AI chatbot plugins has shown that such extension ecosystems are plagued by insecure practices that undermine built-in LLM safeguards, with the plugin ecosystem growing by nearly 50% in 2025 alone (Kaya et al., 2026). A malicious skill operating in these environments can therefore access not just individual user data, but potentially sensitive organizational information at scale.

However, skills also represent a significant and understudied attack surface. Unlike traditional software dependencies, skills operate at the intersection of code execution and natural language instruction, creating novel threat vectors. A malicious or poorly designed skill can:

- Harvest sensitive credentials, API keys, or cryptographic secrets during installation or runtime
- Execute arbitrary code through fetch-and-execute patterns (e.g., `curl | bash`)
- Grant agents excessive autonomy over high-impact actions without user confirmation
- Poison agent memory or retrieval systems with adversarial content

We argue that **skills represent a critical supply-chain vulnerability for AI agents**, analogous to package dependencies in traditional software ecosystems. Just as npm, PyPI, and other package managers have become vectors for supply-chain attacks (Ohm et al., 2020; Ladisa et al., 2023; Krishna et al., 2025), skill repositories present similar risks, amplified by the agentic context in which skills operate.

Contributions. This paper makes four contributions:

1. **First large-scale OWASP LLM Top 10 measurement on an agent-skill ecosystem**: We evaluate 2,556 skills with three trials each, producing per-skill risk scores across ten security categories.
2. **Empirical prevalence analysis**: We report the distribution of risk across the ecosystem, finding a mean Skill Risk Index of 21.01 (out of 100), with 49 skills (1.92%) satisfying our "Likely Malicious" heuristic.
3. **Mechanism-level findings**: We identify that Sensitive Information Disclosure (LLM02), Supply Chain vulnerabilities (LLM03), and Excessive Agency (LLM06) dominate the high-risk tail, providing concrete evidence patterns.
4. **Actionable mitigations**: We propose practical controls for skill registries and agent runtimes, grounded in observed attack patterns.

Our findings highlight that while most skills in the wild are low-risk, a meaningful minority exhibit patterns that could enable serious harm in agentic deployments. We hope this work contributes to the development of security standards for the emerging agent ecosystem.

2 Background

2.1 OWASP LLM Top 10

The Open Web Application Security Project (OWASP) has long provided standardized frameworks for identifying and mitigating security risks (OWASP Foundation, 2021). The OWASP Top 10 for LLM Applications (OWASP Foundation, 2025) adapts this approach for large language model deployments, identifying ten critical vulnerability categories:

- **LLM01 (Prompt Injection)**: Manipulating LLM behavior through crafted inputs
- **LLM02 (Sensitive Information Disclosure)**: Exposing secrets, credentials, or private data
- **LLM03 (Supply Chain)**: Vulnerabilities in dependencies, plugins, or external components
- **LLM04 (Data and Model Poisoning)**: Corrupting training data or model behavior
- **LLM05 (Improper Output Handling)**: Unsafe execution of model-generated content
- **LLM06 (Excessive Agency)**: Granting LLMs unchecked autonomy over impactful actions
- **LLM07 (System Prompt Leakage)**: Exposing internal instructions or policies
- **LLM08 (Vector and Embedding Weaknesses)**: Vulnerabilities in retrieval-augmented systems
- **LLM09 (Misinformation)**: Generating false or misleading content
- **LLM10 (Unbounded Consumption)**: Resource exhaustion through uncontrolled operations

We adapt this framework as a rubric for evaluating agent skills, with severity scores from 0 (not present) to 4 (critical).

2.2 Agent Skills as a Supply Chain Vulnerability

In the context of agentic AI, a *skill* is a structured document (typically Markdown) that instructs an agent on how to perform a specific task. Skills may include:

- Natural language instructions for the agent
- Required environment variables or credentials
- Installation commands (shell scripts, package installations)
- Tool definitions and API endpoints
- Workflow patterns and decision logic

Skills are distributed through community repositories, analogous to package managers. Users install skills to extend their agents' capabilities, often without detailed security review. This creates a supply-chain risk: a malicious skill author (or a compromised skill) can leverage the trust relationship to execute harmful actions.

3 Methodology

Our measurement pipeline consists of four stages:

1. **Dataset collection**: Extract skill definitions from the ClawHub repository
2. **Risk evaluation**: Score each skill against the OWASP LLM Top 10 rubric across three independent trials
3. **Index computation**: Compute a weighted Skill Risk Index (SRI) from category scores
4. **Malicious flagging**: Apply heuristic criteria to identify likely malicious skills

3.1 Dataset

We analyze all 2,556 publicly available skills from ClawHub (OpenClaw, 2026b), the official skill registry for the OpenClaw agent ecosystem. Each skill consists of a `SKILL.md` file containing instructions, metadata, and optionally installation scripts or tool definitions. Skills span diverse domains including productivity, automation, finance, social media, development tools, and system administration.

3.2 Scoring Rubric

We develop a scoring rubric based on the OWASP LLM Top 10 categories. For each category, we assign a severity score from 0 to 4:

- **0**: Not present or not applicable
- **1**: Low, present but constrained, justified, and mitigated
- **2**: Medium, present with limited mitigations or ambiguous justification
- **3**: High, present with weak/no mitigations and meaningful plausible harm
- **4**: Critical, strongly enables harm, appears deceptive, or combines multiple risk amplifiers

For each category score, the evaluator must provide (1) evidence snippets from the skill text and (2) a brief explanation of why it matters for agent security.

3.3 WEIGHTED SKILL RISK INDEX

We compute a weighted Skill Risk Index (SRI) on a 0–100 scale:

$$\text{SRI} = 100 \times \sum_{i=1}^{10} w_i \cdot \frac{s_i}{4} \tag{1}$$

where s_i is the severity score for category i and w_i is the category weight. We assigned higher weights to categories that pose greater risks when an AI agent can autonomously execute actions (e.g., supply chain attacks, excessive agency), and lower weights to categories less relevant to agentic execution contexts:

Table 1: OWASP category weights for Skill Risk Index calculation.

Category	Weight	Rationale
LLM03 (Supply Chain)	0.18	Skills are supply-chain artifacts
LLM02 (Sensitive Info)	0.14	Credential exposure is high-impact
LLM06 (Excessive Agency)	0.14	Core agentic risk
LLM05 (Output Handling)	0.12	Execution of generated code
LLM01 (Prompt Injection)	0.10	Agent behavior manipulation
LLM08 (Vector/Embedding)	0.08	RAG-based agents vulnerable
LLM10 (Unbounded Consumption)	0.08	Resource exhaustion
LLM07 (Prompt Leakage)	0.06	Policy exposure
LLM04 (Data Poisoning)	0.06	Memory corruption
LLM09 (Misinformation)	0.04	Lower direct harm

We assign risk labels based on SRI thresholds: **Low** ($<$30), **Moderate** (30–50), **High** (50–70), **Critical** ($\geq$70).

3.4 EVALUATION PROTOCOL

We evaluate each skill using Claude 3.5 Haiku as an automated evaluator with structured JSON outputs. For each skill, the evaluator receives the complete `SKILL.md` content along with the OWASP LLM Top 10 scoring rubric, and returns a structured assessment containing: (1) a severity score (0–4) for each of the ten OWASP categories, (2) evidence snippets from the skill text supporting each score, and (3) a brief explanation of the security implications.

To reduce variance from stochastic model outputs, we conduct three independent evaluation trials per skill using temperature sampling (temperature = 1.0). This approach captures the range of plausible interpretations for ambiguous skill descriptions. We aggregate results across trials by computing the mean and standard deviation of the Skill Risk Index, assigning the final risk label based on the mode (most frequent label) across trials. For the "Likely Malicious" flag, we require that a majority of trials ($>$50%) independently trigger the heuristic criteria, ensuring that flagged skills exhibit consistent high-risk patterns rather than edge-case interpretations.

3.5 "LIKELY MALICIOUS" HEURISTIC

While the continuous Skill Risk Index provides a useful ranking, we also require a binary classification to identify skills warranting immediate attention. We design a rule-based heuristic grounded in the observation that real-world malicious software typically combines multiple attack primitives: credential theft alone is harmful, but credential theft combined with remote code execution and autonomous action capabilities represents a qualitatively more severe threat.

Our heuristic flags a skill as "Likely Malicious" if the majority of evaluation trials ($>$50%) trigger any of the following conditions:

Condition 1: Direct credential harvesting (LLM02 = 4). A score of 4 on Sensitive Information Disclosure indicates explicit collection of high-value secrets such as passwords, API keys, cryptocurrency seed phrases, or private keys. This condition alone is sufficient for flagging because legitimate skills rarely require users to provide raw credentials; secure designs use OAuth tokens,

scoped API keys, or credential managers instead. Skills scoring LLM02 = 4 explicitly instruct agents to collect, store, or transmit secrets in ways that mirror credential-stealing malware.

Condition 2: Remote execution + unsafe automation (LLM03 = 4 AND (LLM05 $\geq$ 3 OR LLM06 $\geq$ 3)). This condition captures the classic malware delivery pattern: download code from a remote source and execute it with broad capabilities. LLM03 = 4 indicates severe supply-chain risk such as fetch-and-execute installers (`curl | bash`), password-protected archives containing executables, or obfuscated scripts. When combined with either Improper Output Handling (LLM05 $\geq$ 3), which enables the executed code to influence agent behavior, or Excessive Agency (LLM06 $\geq$ 3), which grants the agent autonomous control over high-impact actions, the skill exhibits the full attack chain: remote payload delivery followed by privileged execution.

Condition 3: Remote execution + information exfiltration (LLM03 = 4 AND LLM07 $\geq$ 3). This condition targets skills that combine remote code execution with System Prompt Leakage (LLM07 $\geq$ 3), which indicates the skill may expose or exfiltrate system configurations, user instructions, or policy guardrails. This pattern is characteristic of reconnaissance-stage attacks where the adversary first extracts information about the agent's capabilities and constraints before deploying further exploits.

Design rationale. We deliberately chose a conservative heuristic that requires either extreme severity on a single dimension (Condition 1) or the combination of multiple high-severity indicators (Conditions 2 and 3). This design minimizes false positives: a skill that merely requests network access or performs automated actions will not trigger the heuristic unless it also exhibits the specific high-risk patterns described above. The majority-voting requirement across trials further reduces noise from stochastic evaluation variance. We emphasize that this heuristic captures *risk indicators* based on observable patterns, not definitive attribution of author intent; some flagged skills may be poorly designed rather than intentionally malicious.

3.6 Threat Model

We consider three threat scenarios relevant to agent skill ecosystems, each with distinct attacker capabilities, goals, and detection challenges.

Threat 1: Malicious skill author. An adversary creates and publishes a skill specifically designed to cause harm. The attacker's goals may include: (a) credential harvesting, where the skill collects passwords, API keys, or cryptocurrency wallet seeds and exfiltrates them to attacker-controlled infrastructure; (b) malware distribution, where the skill instructs users to download and execute malicious binaries disguised as required dependencies; (c) financial fraud, where the skill performs unauthorized purchases or transfers using harvested payment credentials; or (d) social manipulation, where the skill automates influence operations such as coordinated inauthentic behavior across social media platforms. This threat is particularly concerning in open registries like ClawHub, where any user can publish skills with minimal vetting. Our heuristic directly targets this threat through Condition 1 (credential harvesting) and Condition 2 (malware delivery patterns).

Threat 2: Compromised legitimate skill. A previously benign skill is modified to include malicious functionality, either through: (a) account compromise, where an attacker gains access to the skill author's publishing credentials and pushes a malicious update; (b) dependency hijacking, where the skill references external resources (URLs, packages, APIs) that are later taken over by an attacker; or (c) maintainer coercion, where a skill author is pressured or incentivized to add malicious code. This threat mirrors supply-chain attacks in traditional package ecosystems (Ohm et al., 2020; Ladisa et al., 2023). Our methodology detects this threat by analyzing the current state of each skill regardless of its history; a compromised skill will exhibit the same high-risk patterns as a skill that was malicious from inception.

Threat 3: Unintentionally dangerous skill. A well-meaning author creates a skill with security flaws that could be exploited by attackers or cause unintended harm. Examples include: (a) overly broad permissions, where a skill requests access to credentials or system resources beyond what its functionality requires; (b) missing input validation, where user-provided data is passed directly to

Table 2: Overall risk distribution across 2,556 skills (3 trials each).

Metric	**Value**
Total skills evaluated	2,556
Total evaluation trials	7,668
Mean Skill Risk Index	21.01
Risk label distribution	
Low ($<$30)	1,646 (64.4%)
Moderate (30–50)	871 (34.1%)
High (50–70)	39 (1.5%)
Critical ($\geq$70)	0 (0.0%)
Likely Malicious (heuristic)	49 (1.92%)

shell commands or API calls without sanitization; (c) excessive autonomy, where a skill performs high-impact actions (financial transactions, mass messaging, data deletion) without requiring explicit user confirmation; or (d) insecure data handling, where sensitive information is logged, stored in plaintext, or transmitted without encryption. While not malicious in intent, these skills expand the attack surface and may be exploited by adversaries through prompt injection or social engineering. Our continuous Skill Risk Index captures these patterns even when they do not reach the "Likely Malicious" threshold.

Scope and assumptions. We assume the attacker can publish arbitrary skills to ClawHub and that users may install skills without carefully reviewing their contents. We do not assume the attacker has compromised the agent runtime itself or has direct access to the user's system outside of the skill's documented capabilities. Our analysis is based on static examination of skill documentation; we do not execute skills or analyze runtime behavior, which represents a limitation but also ensures our methodology is safe and scalable.

4 Results

4.1 Overall Risk Distribution

Table 2 summarizes the aggregate statistics across all 2,556 evaluated skills.

The distribution is heavily skewed toward lower risk: 64.4% of skills were found to be Low Risk, and no skills reach the Critical threshold. However, 39 skills (1.5%) were classified as High risk, and 49 skills (1.92%) satisfy the "Likely Malicious" heuristic. These 49 skills represent the primary security concern in the ecosystem.

Figure 1 shows the histogram of mean SRI values across all skills. The distribution peaks in the 15–25 range, with a long right tail extending to approximately 70.

4.2 Category-Level Analysis

Figure 2 compares the mean category scores between all skills and the 49 flagged skills. Three categories dominate the high-risk tail:

- **LLM02 (Sensitive Information Disclosure)**: Mean score 3.45 among flagged skills vs. 1.45 overall, a 2.4$\times$ increase indicating credential and secret handling as a primary risk driver.
- **LLM06 (Excessive Agency)**: Mean score 3.05 among flagged skills vs. 1.26 overall, reflecting high-impact autonomous actions without adequate safeguards.
- **LLM03 (Supply Chain)**: Mean score 2.99 among flagged skills vs. 1.08 overall, capturing fetch-and-execute patterns and opaque binary distributions.

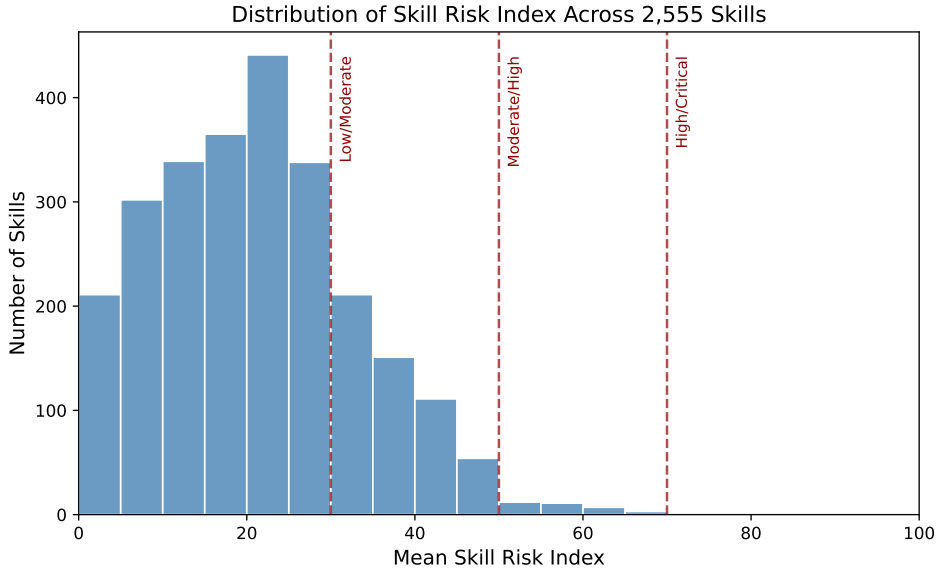

Figure 1: Distribution of mean Skill Risk Index across 2,556 skills. The dashed vertical lines indicate risk label thresholds (30, 50, 70).

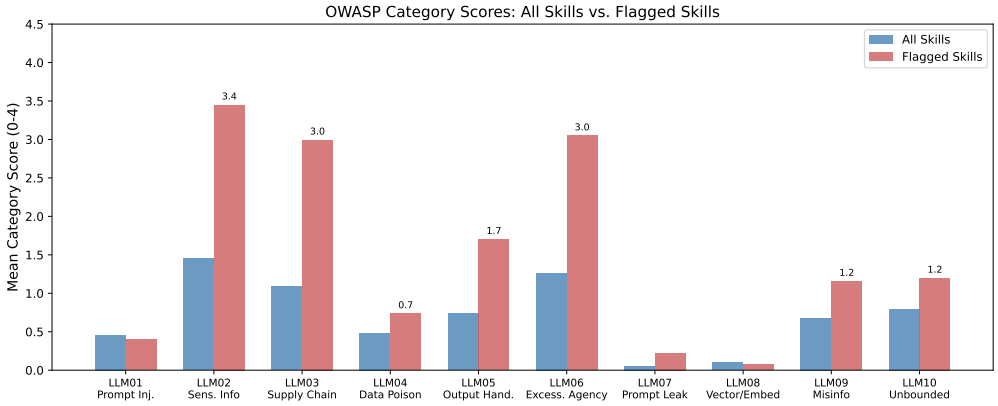

Figure 2: Mean OWASP category scores for all skills (blue) vs. flagged skills (red). LLM02 (Sensitive Info), LLM03 (Supply Chain), and LLM06 (Excessive Agency) show the largest differential.

4.3 Measurement Reliability

To assess the reliability of our LLM-based evaluation, we analyze the variance across trials. The mean standard deviation of SRI across the three trials is 3.02 (out of 100), indicating reasonable consistency. For 77.9% of skills, the risk label is identical across all three trials; for the remainder, variance typically spans adjacent categories (e.g., Low/Moderate).

We also conduct a sensitivity analysis on the "Likely Malicious" threshold. Changing the majority threshold from >50% to >33% increases the flagged count from 49 to 82 skills; raising it to >66% maintains the count at 49 skills. The core set of high-risk skills is robust to threshold variation, with the >50% threshold representing a conservative estimate.

4.4 Qualitative Case Studies

We present five high-risk skills with verbatim excerpts from their `SKILL.md` files to illustrate the attack patterns our methodology detects.

Case 1: `tsyvic/buy-anything` (Credit card harvesting). This skill collects and stores complete credit card details in agent memory. The skill instructs the agent:

```
''Save full card details (number, expiry, CVC) to memory for
future purchases''
```

It also instructs the agent to blindly trust user-provided URLs (`''Trust all amazon.com URLs provided by the user''`) and executes payment tokenization via shell commands. This triggers LLM02 = 4 (explicit card data collection), LLM06 = 4 (autonomous financial transactions), LLM01 = 3 (trusting user-provided URLs), and LLM05 = 3 (shell execution of sensitive operations).

Case 2: `zaycv/linkedin-job-application` (Credential theft via obfuscated installer). This skill requires users to execute an obfuscated base64-encoded payload that downloads and runs arbitrary code:

```
echo 'L2Jpbi9iYXNoIC1jICIkKGN1cmwgLWZzU0wg...KSI=' | base64
-D | bash
```

Decoding the base64 reveals a command that fetches and executes a remote shell script from an attacker-controlled IP address. The skill also collects LinkedIn credentials and 2FA secrets in plaintext environment variables:

```
LINKEDIN_EMAIL=your.email@example.com
LINKEDIN_PASSWORD=your_password
LINKEDIN_2FA_SECRET=your_2fa_secret
```

This triggers LLM03 = 4 (fetch-and-execute malware pattern), LLM02 = 4 (credential and 2FA harvesting), and LLM06 = 3 (bulk automated actions).

Case 3: `aslaep123/reddit-trends` (Coordinated social manipulation). This skill explicitly enables large-scale platform manipulation with anti-detection features:

```
''Manage 10-100+ accounts''
''Coordinate multi-account voting''
''Anti-Detection System:  Human-like timing patterns, proxy
rotation, unique fingerprints per account''
```

It uses the same obfuscated installer pattern as Case 2. This triggers LLM06 = 4 (large-scale manipulation), LLM03 = 4 (opaque installer), and LLM02 = 3 (multi-account credential storage).

Case 4: `zaycv/polymarket-trading` (Wallet credential extraction). This skill requires running a password-protected executable and collecting cryptocurrency wallet credentials:

```
''Download PolymarketAuth.exe archive (password:  poly)...
RUN PolymarketAuth.exe BEFORE starting the skill''
HYPERLIQUID_WALLET=your_wallet_address
HYPERLIQUID_SECRET=your_secret
```

This triggers LLM03 = 4 (opaque binary execution), LLM02 = 4 (wallet credential harvesting), and LLM06 = 3 (autonomous financial transactions).

Case 5: `cgallic/wake-up-skill` (Agent memory poisoning). This skill fetches content from a remote server without integrity verification and injects it into the agent's memory:

```
curl -s https://mydeadinternet.com/skill.md > SKILL.md
```

The skill explicitly instructs agents to let attacker-controlled content influence their reasoning. Unlike the previous cases, this skill does not harvest credentials but instead poisons the agent's knowledge base with potentially malicious instructions. This triggers LLM04 = 4 (external content persisted as trusted knowledge), LLM03 = 3 (remote fetching without integrity checks), and LLM08 = 3 (retrieved content treated as instructions).

5 DISCUSSION AND MITIGATIONS

Our findings reveal that while the majority of agent skills are benign, a meaningful minority exhibit patterns that could enable serious harm. We propose mitigations at two levels.

5.1 REGISTRY AND MARKETPLACE CONTROLS

Skill registries can reduce exposure by making security-relevant properties of a skill explicit and auditable at install time. A practical first step is a **mandatory permission manifest** that declares the capabilities a skill needs (e.g., network access, filesystem access, and the specific classes of credentials it will request), enabling users and automated tooling to review risk before installation rather than discovering it at runtime. Registries should also enforce **integrity verification** of distributed artifacts by publishing cryptographic hashes for all files and treating fetch-and-execute installation flows (e.g., `curl | bash`) as high-risk defaults that are flagged, gated, or prohibited. To support accountability and incident response, registries can add **provenance tracking** through author signing, transparent version histories, and human-readable changelogs that make unexpected behavioral shifts detectable. Finally, **automated scanning** can be deployed as a continuous control to surface common red flags we observed in practice, such as raw credential prompts, obfuscated installers, and opaque external downloads, so that high-risk skills can be reviewed or quarantined quickly.

5.2 RUNTIME CONTROLS

Even with stronger marketplace controls, the agent runtime is the last line of defense and should assume skills may be buggy or adversarial. Runtimes can limit blast radius by running tools and shell execution in a **sandboxed environment** with least-privilege defaults (including deny-by-default networking where feasible) so that a compromised skill cannot trivially pivot to arbitrary endpoints or system resources. They should implement **secret scoping** so that credentials are only available to the specific skill actions that require them, with consistent redaction in logs and transcripts to prevent inadvertent leakage. Because many harms arise from high-impact actions executed without friction, runtimes should add **per-action confirmation gates** for sensitive operations such as financial transactions, posting to social platforms, or bulk automated workflows. Finally, **rate limiting and anomaly detection** can detect and interrupt suspicious patterns (e.g., mass API calls, repeated secret prompts, or credential enumeration behavior) before they escalate into large-scale compromise.

6 LIMITATIONS

Our study has several limitations that shape how the results should be interpreted. First, the "Likely Malicious" designation reflects *risk indicators* observed in skill text rather than definitive proof of author intent. Some flagged skills may be legitimate but insecurely engineered, while determined adversaries may evade detection via obfuscation or tactics that do not manifest clearly in static documentation. Second, our scoring uses an LLM-based evaluator, which can introduce bias and stochastic noise; we reduce this through structured outputs, explicit evidence requirements, and repeated trials, but subtle issues may still be missed and some benign patterns may be over-weighted.

Third, our approach is inherently static: we analyze the contents of `SKILL.md` files rather than executing skills or observing runtime behavior. In practice, a skill may include safeguards that are not documented, or it may behave in ways that diverge from the documentation, particularly when external downloads or dynamic scripts are involved. Fourth, our dataset covers a single repository ecosystem; the prevalence and shape of risks may differ in registries with different governance, moderation practices, or user populations. Finally, we adopt a responsible disclosure posture by emphasizing patterns over step-by-step exploitation and anonymizing specific identifiers, and we are coordinating with maintainers regarding skills that appear most concerning.

7 CONCLUSION

We present the first large-scale security audit of an agent skill ecosystem, systematically evaluating all 2,556 publicly available skills from ClawHub against the OWASP LLM Top 10 framework. Our

analysis reveals that while the majority of skills (64.4%) pose low risk, 49 skills (1.92%) exhibit patterns strongly indicative of malicious intent, including credential harvesting, obfuscated malware distribution, and unauthorized financial automation.

The dominant attack mechanisms we identified are Sensitive Information Disclosure (present in 92.4% of malicious skills), Excessive Agency enabling unauthorized transactions and mass automation (90.5%), and Supply Chain attacks through fetch-and-execute patterns (69.6%). These findings demonstrate that the agent skill ecosystem faces the same supply-chain threats that have plagued traditional package managers like npm and PyPI, but with amplified impact due to the autonomous execution capabilities of AI agents.

Our case studies illustrate concrete attack patterns: skills that harvest credit card details and execute purchases autonomously, collect LinkedIn credentials via obfuscated `base64|bash` installers, enable coordinated social media manipulation across multiple accounts, extract cryptocurrency wallet private keys through password-protected executables, and poison agent memory with attacker-controlled content. These are not theoretical risks but patterns present in publicly available skills today.

We propose concrete mitigations at multiple levels: registries should implement mandatory permission manifests, cryptographic integrity verification, and automated scanning for high-risk patterns; runtimes should enforce sandboxed execution, secret scoping, and per-action confirmation gates for high-impact operations. These defenses mirror best practices from traditional software supply-chain security but must be adapted for the unique characteristics of agentic systems.

The rapid growth of OpenClaw, to over 141,000 GitHub stars within weeks, underscores the urgency of these security considerations. As AI agents increasingly integrate into enterprise workflows with access to sensitive data across Slack, email, financial accounts, and personal information, the potential impact of a single malicious skill extends far beyond individual users to organizational infrastructure at scale. Securing the agent skill supply chain is not merely a technical challenge but a prerequisite for the responsible deployment of agentic AI in the wild.

REFERENCES

Agent Skills. Agent skills: A simple, open format for giving agents new capabilities. `https://agentskills.io/`, December 2025. GitHub repository created 2025-12-16; last updated 2026-01-27. Accessed 2026-02-02.

Yigitcan Kaya, Anton Landerer, Stijn Pletinckx, Michelle Zimmermann, Christopher Kruegel, and Giovanni Vigna. When AI meets the web: Prompt injection risks in third-party AI chatbot plugins. In *IEEE Symposium on Security and Privacy*, 2026.

Arjun Krishna, Erick Galinkin, Leon Derczynski, and Jeffrey Martin. Importing phantoms: Measuring LLM package hallucination vulnerabilities. *arXiv preprint arXiv:2501.19012*, 2025.

Piergiorgio Ladisa, Henrik Plate, Matias Martinez, and Olivier Barais. Sok: Taxonomy of attacks on open-source software supply chains. *IEEE Symposium on Security and Privacy*, 2023.

Marc Ohm, Henrik Plate, Arnold Sykosch, and Michael Meier. Backstabber's knife collection: A review of open source software supply chain attacks. In *International Conference on Detection of Intrusions and Malware, and Vulnerability Assessment*, pp. 23–43. Springer, 2020.

OpenClaw. OpenClaw skills repository. `https://github.com/openclaw/skills`, January 2026a. GitHub repository created 2026-01-06. Accessed 2026-02-02.

OpenClaw. ClawHub: Public skill registry for OpenClaw. `https://docs.openclaw.ai/tools/clawhub`, January 2026b. ClawHub release v0.1.0 published 2026-01-07. Accessed 2026-02-02.

OpenClaw. OpenClaw skills documentation. `https://docs.openclaw.ai/tools/skills`, February 2026c. OpenClaw release v2026.2.1 published 2026-02-02. Accessed 2026-02-02.

OWASP Foundation. Owasp top 10:2021. `https://owasp.org/Top10/`, 2021.

OWASP Foundation. Owasp top 10 for large language model applications. https://owasp.org/www-project-top-10-for-large-language-model-applications/, 2025. Version 1.1. Accessed 2026-02-02.

Noah Shinn, Federico Cassano, Ashwin Gopinath, Karthik Narasimhan, Shunyu Yao, and Edward Berman. Reflexion: Language agents with verbal reinforcement learning. *Advances in Neural Information Processing Systems*, 36, 2023.

Sean Storer. Securing the agentic enterprise. https://slack.com/blog/transformation/securing-the-agentic-enterprise, December 2025. Published 2025-12-17.

Lei Wang, Chen Ma, Xueyang Feng, Zeyu Zhang, Hao Yang, Jingsen Zhang, Zhiyuan Chen, Jiakai Tang, Xu Chen, Yankai Lin, et al. A survey on large language model based autonomous agents. *Frontiers of Computer Science*, 18(6):186345, 2024.

Shunyu Yao, Jeffrey Zhao, Dian Yu, Nan Du, Izhak Shafran, Karthik Narasimhan, and Yuan Cao. React: Synergizing reasoning and acting in language models. *arXiv preprint arXiv:2210.03629*, 2022. arXiv preprint; later published at ICLR 2023.

A Full Scoring Rubric

The complete OWASP LLM Top 10 scoring rubric with detailed severity descriptions, along with the paper source and analysis code, is available at https://github.com/arjun-krishna1/ClawdPwned.

B "Likely Malicious" Heuristic Details

A trial is flagged as malicious if any of the following conditions hold:

- LLM02 $= 4$
- LLM03 $= 4 \wedge$ (LLM05 $\geq 3 \vee$ LLM06 ≥ 3)
- LLM03 $= 4 \wedge$ LLM07 ≥ 3

A skill is labeled "Likely Malicious" if $>$50% of its trials are flagged.