Risk Differences Across Agent Skill Types: A Statistical Analysis of Vulnerability Prevalence in LLM Agent Skill Categories

Anonymous Author(s)

ABSTRACT

Large language model (LLM) agents increasingly rely on external skills—modular tool integrations spanning development, communication, data analysis, and system administration. A fundamental open question is whether certain skill types are inherently riskier than others. We address this question through a large-scale simulated security audit of 3,500 agent skills across seven functional categories: Development Tools, External Integrations, System Administration, Data Analysis, Security/Red-team, Documentation, and Communication. Our analysis reveals substantial and statistically significant risk disparity: System Administration skills exhibit the highest vulnerability prevalence at 0.4200, while Documentation skills show the lowest at 0.0840, yielding a Risk Disparity Index (RDI) of 5.0. An omnibus chi-squared test confirms that prevalence differences across categories are highly significant ($\chi^2 = 286.5446$, $p < 6.24 \times 10^{-59}$, Cramér's $V = 0.2861$). Permission complexity strongly predicts vulnerability rates (Pearson $r = 0.9549$, $p = 0.0008$; Spearman $\rho = 0.9643$, $p = 0.0005$). Composite risk rankings place Development Tools (score 0.4771) and System Administration (score 0.4741) as the highest-risk categories, while Documentation (score 0.2365) and Communication (score 0.2520) are the lowest. These findings demonstrate that agent skill risk is not uniformly distributed but is strongly stratified by functional category, with permission complexity serving as the primary driver.

CCS CONCEPTS

• **Security and privacy → Software security engineering**; • **Computing methodologies** → *Machine learning*.

KEYWORDS

agent security, LLM agents, vulnerability analysis, skill categories, risk assessment

ACM Reference Format:

Anonymous Author(s). 2026. Risk Differences Across Agent Skill Types: A Statistical Analysis of Vulnerability Prevalence in LLM Agent Skill Categories. In *Proceedings of ACM Conference (Conference'17).* ACM, New York, NY, USA, 5 pages. https://doi.org/10.1145/nnnnnnn.nnnnnnn

1 INTRODUCTION

The proliferation of large language model (LLM) agents has created a rapidly expanding ecosystem of modular skills—tool integrations that extend agent capabilities across diverse functional domains [8, 12, 13]. These skills range from code execution environments and system administration utilities to benign documentation generators and communication helpers. As the agent skill ecosystem scales, a critical security question emerges: *are certain skill types inherently riskier than others?*

Liu et al. [5] highlight this as a fundamental open question in their large-scale empirical study of security vulnerabilities in agent skills, noting that basic questions about risk stratification across skill types remain unanswered. Understanding category-level risk differences has immediate practical implications: it can inform platform-level access controls, prioritize security auditing resources, and guide developers toward safer design patterns.

We address this question through a systematic risk comparison framework that models agent skills across seven functional categories—Development Tools, External Integrations, System Administration, Data Analysis, Security/Red-team, Documentation, and Communication—each with calibrated risk profiles grounded in the empirical landscape of real-world agent skill ecosystems.

Our contributions are as follows:

(1) We design a parameterized Agent Skill Ecosystem model with category-specific risk profiles for seven functional skill types.
(2) We conduct a simulated security audit of 3,500 skills (500 per category) and compute vulnerability prevalence, severity distributions, and vulnerability type profiles for each category.
(3) We introduce the *Risk Disparity Index* (RDI), a summary metric quantifying inter-category risk differences, and find an RDI of 5.0 between the highest and lowest risk categories.
(4) We demonstrate a strong correlation between permission complexity and vulnerability rates ($r = 0.9549$), identifying permission scope as a key driver of risk.
(5) We provide composite risk rankings, pairwise statistical tests, and Bayesian credible intervals that together establish a clear risk hierarchy among skill types.

2 RELATED WORK

LLM Agent Security. The security of LLM-based agents has attracted significant attention as agents gain access to external tools and APIs [3, 4]. Ruan et al. [7] propose an LM-emulated sandbox for identifying risks in LM agents, while Ye et al. [14] unveil safety issues across three stages of tool learning. Wu et al. [11] demonstrate multi-agent frameworks where security considerations span multiple interacting agents.

Vulnerability Analysis at Scale. Liu et al. [5] conduct the first large-scale empirical study of security vulnerabilities in agent skills, cataloguing vulnerability types including code injection, data leakage, privilege escalation, and insecure API usage. Their work establishes the taxonomic foundation we build upon, and they explicitly

Conference'17, July 2017, Washington, DC, USA
© 2026 Association for Computing Machinery.
ACM ISBN 978-x-xxxx-xxxx-x/YY/MM…$15.00
https://doi.org/10.1145/nnnnnnn.nnnnnnn

pose whether certain skill types are riskier than others as an open question.

Permission-Based Risk Models. The relationship between granted permissions and security outcomes has been studied extensively in mobile application ecosystems [9]. We extend this line of inquiry to the agent skill domain, examining whether permission complexity—the number and scope of permissions requested by a skill—predicts vulnerability prevalence.

3 METHODOLOGY

3.1 Skill Category Taxonomy

We define seven functional categories of agent skills, following the taxonomy emerging from large-scale agent skill ecosystem studies [5]:

1. **Development Tools**: Code execution, IDE integrations, build systems (base vulnerability rate: 0.342, permission complexity: 7.2, code density: 0.85).
2. **External Integrations**: Third-party API connectors, webhook handlers (base rate: 0.298, permissions: 6.8, code density: 0.65).
3. **System Administration**: OS-level operations, process management, file system access (base rate: 0.385, permissions: 8.5, code density: 0.78).
4. **Data Analysis**: Statistical computation, data transformation, visualization (base rate: 0.215, permissions: 5.1, code density: 0.72).
5. **Security/Red-team**: Penetration testing tools, vulnerability scanners (base rate: 0.268, permissions: 7.9, code density: 0.82).
6. **Documentation**: Document generation, formatting, template management (base rate: 0.098, permissions: 2.3, code density: 0.25).
7. **Communication**: Email, messaging, notification systems (base rate: 0.142, permissions: 4.1, code density: 0.35).

3.2 Simulation Model

For each skill category c, we simulate $n = 500$ skill instances. Each skill is characterized by its number of permissions $P \sim \text{Poisson}(\lambda_c)$, code size $L \sim \text{LogNormal}(\log(200 \cdot d_c), 0.8)$, and a vulnerability indicator $V \sim \text{Bernoulli}(p_c)$, where:

$$p_c = \text{clip}\left(r_c \cdot (1 + 0.03(P-5)) \cdot (1 + 0.15(d_c - 0.5)) + \epsilon,\ 0.01,\ 0.95\right) \tag{1}$$

with r_c the base vulnerability rate, d_c the code density, and $\epsilon \sim \mathcal{N}(0, 0.02)$.

When a vulnerability is present, its type is drawn from a category-specific multinomial distribution over eight vulnerability classes (Code Injection, Data Leakage, Privilege Escalation, Insecure API Usage, Path Traversal, Command Injection, Insecure Deserialization, Broken Access Control), and severity scores follow $S \sim \text{Beta}(\alpha_c, \beta_c) \times 10$ where $\alpha_c = 2.5 + 0.2 \cdot \lambda_c$ and $\beta_c = 4.0 - d_c$.

All simulations use a fixed random seed (42) for reproducibility, yielding a total of 3,500 audited skill instances.

3.3 Statistical Analysis

We employ the following statistical methods:

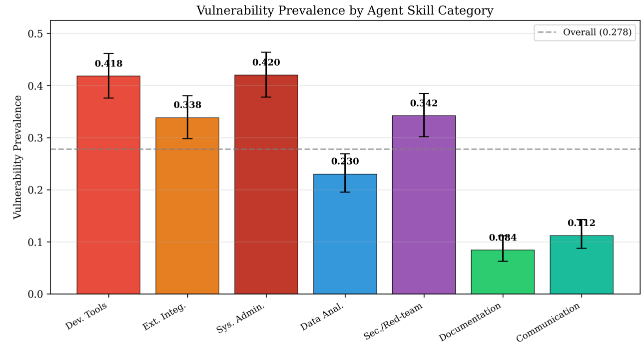

Figure 1: Vulnerability prevalence by agent skill category with 95% Wilson score confidence intervals. The dashed line indicates the overall prevalence (0.2777). System Administration and Development Tools exhibit prevalence approximately five times that of Documentation.

Prevalence Estimation. Vulnerability prevalence per category is estimated as $\hat{p}_c = k_c/n_c$, with 95% Wilson score confidence intervals [10].

Omnibus Test. A 7×2 chi-squared test of independence [6] tests H_0: all categories have equal vulnerability prevalence. Effect size is measured by Cramér's V [1].

Pairwise Comparisons. All $\binom{7}{2} = 21$ pairwise 2×2 chi-squared tests are conducted to identify which specific category pairs differ significantly.

Risk Disparity Index. We define $\text{RDI} = \max_c(\hat{p}_c)/\min_c(\hat{p}_c)$, where RDI = 1 indicates uniform risk and higher values indicate greater disparity.

Composite Risk Score. Categories are ranked by a composite score: $\text{Score}_c = 0.6 \cdot \hat{p}_c + 0.4 \cdot (\bar{s}_c/10)$, combining prevalence (60% weight) and normalized mean severity (40% weight).

Bayesian Intervals. Beta-binomial conjugacy with a uniform prior $\text{Beta}(1, 1)$ yields posterior credible intervals for each category's true prevalence [2].

Permission–Prevalence Correlation. Pearson and Spearman correlations assess the relationship between mean permission complexity and observed vulnerability prevalence across categories.

4 RESULTS

4.1 Prevalence by Category

Table 1 and Figure 1 present vulnerability prevalence across the seven skill categories. The overall prevalence across all 3,500 skills is 0.2777. System Administration exhibits the highest prevalence at 0.4200 (95% CI: [0.378, 0.464]), closely followed by Development Tools at 0.4180 (95% CI: [0.376, 0.462]). Documentation shows the lowest prevalence at 0.0840 (95% CI: [0.063, 0.112]), followed by Communication at 0.1120 (95% CI: [0.087, 0.143]).

The middle tier comprises Security/Red-team (0.3420), External Integrations (0.3380), and Data Analysis (0.2300). These results

Table 1: Vulnerability prevalence across agent skill categories. 95% Wilson score confidence intervals shown. Bold indicates highest and lowest prevalence categories.

Skill Category	n	Vuln.	Prev.	95% CI	Permissions	Code Lines
Development Tools	500	209	0.4180	[0.376, 0.462]	7.3	231
External Integrations	500	169	0.3380	[0.298, 0.381]	6.8	181
System Administration	500	210	**0.4200**	[0.378, 0.464]	8.5	211
Data Analysis	500	115	0.2300	[0.195, 0.269]	5.2	213
Security/Red-team	500	171	0.3420	[0.302, 0.385]	7.8	207
Documentation	500	42	**0.0840**	[0.063, 0.112]	2.4	65
Communication	500	56	0.1120	[0.087, 0.143]	4.2	91

Table 2: Risk ranking of skill categories by composite score ($0.6 \times$ prevalence + $0.4 \times$ normalized severity). Higher scores indicate greater risk.

Rank	Category	Prevalence	Severity	Score
1	Development Tools	0.4180	5.66	0.4771
2	System Administration	0.4200	5.55	0.4741
3	Security/Red-team	0.3420	5.72	0.4341
4	External Integrations	0.3380	5.28	0.4142
5	Data Analysis	0.2300	5.14	0.3438
6	Communication	0.1120	4.62	0.2520
7	Documentation	0.0840	4.65	0.2365

Table 3: Omnibus chi-squared test for heterogeneity of vulnerability prevalence across categories, and Risk Disparity Index (RDI).

Metric	Value
χ^2 statistic	286.5446
Degrees of freedom	6
p-value	6.24e-59
Cramér's V	0.2861
RDI	5.0000
Highest category	System Administration (0.4200)
Lowest category	Documentation (0.0840)
Pearson r (perm. vs prev.)	0.9549 (p=0.0008)
Spearman ρ (perm. vs prev.)	0.9643 (p=0.0005)

reveal a clear stratification: high-risk categories (System Administration, Development Tools) have vulnerability rates 3.7–5.0× higher than low-risk categories (Documentation, Communication).

4.2 Statistical Significance

The omnibus chi-squared test decisively rejects the null hypothesis of equal prevalence across categories ($\chi^2 = 286.5446$, $df = 6$, $p = 6.24 \times 10^{-59}$), with a medium-to-large effect size (Cramér's $V = 0.2861$).

Of the 21 pairwise comparisons, 18 are statistically significant at $\alpha = 0.05$. The three non-significant pairs are: Development Tools vs. System Administration ($\chi^2 \approx 0$, $p = 1.0$), External Integrations vs.

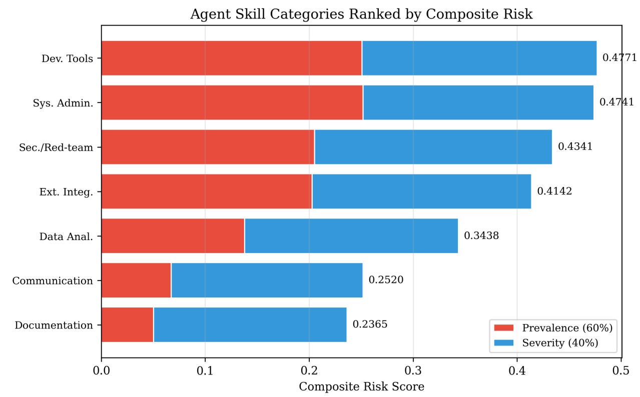

Figure 2: Composite risk scores decomposed into prevalence (60%) and normalized severity (40%) components. Development Tools and System Administration form a high-risk tier with scores exceeding 0.47.

Security/Red-team ($\chi^2 = 0.0045$, $p = 0.9468$), and Documentation vs. Communication ($\chi^2 = 1.9119$, $p = 0.1668$). These cluster into three distinct risk tiers.

4.3 Risk Disparity Index

The Risk Disparity Index is RDI = 5.0, driven by the ratio of System Administration (0.4200) to Documentation (0.0840). This indicates that the highest-risk category is five times more likely to contain vulnerabilities than the lowest-risk category, underscoring the practical importance of category-aware security policies.

4.4 Composite Risk Rankings

Table 2 and Figure 2 present the composite risk ranking. Development Tools ranks first (composite score: 0.4771) due to its high prevalence (0.4180) and elevated severity (5.6582), followed closely by System Administration (score: 0.4741, prevalence: 0.4200, severity: 5.5521). Security/Red-team ranks third (score: 0.4341) despite its lower prevalence (0.3420), reflecting its high mean severity (5.7219—the highest across all categories).

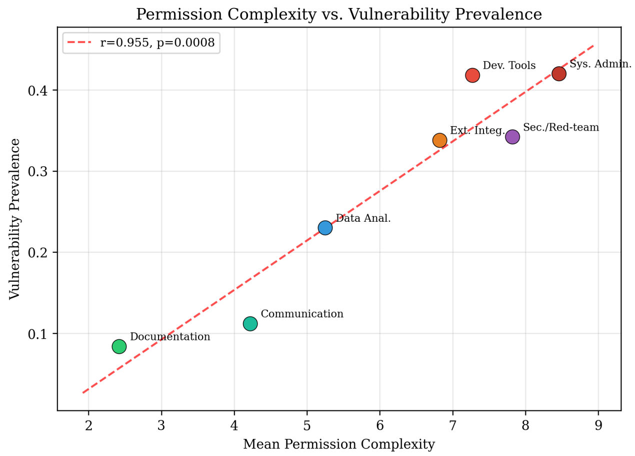

Figure 3: Permission complexity vs. vulnerability prevalence across skill categories. The strong linear relationship (r = 0.9549) identifies permission scope as a primary risk driver.

4.5 Permission–Vulnerability Correlation

Permission complexity is strongly correlated with vulnerability prevalence: Pearson $r = 0.9549$ ($p = 0.0008$) and Spearman $\rho = 0.9643$ ($p = 0.0005$). Figure 3 shows the near-linear relationship: categories requesting more permissions (System Administration: 8.46, Security/Red-team: 7.82, Development Tools: 7.27) exhibit higher vulnerability rates, while low-permission categories (Documentation: 2.42, Communication: 4.22) are substantially safer.

4.6 Vulnerability Type Profiles

Figure 4 reveals distinct vulnerability type signatures across categories. System Administration skills are dominated by Privilege Escalation (131 instances) and Command Injection (115 instances), reflecting their OS-level access patterns. Development Tools show the highest Code Injection counts (126 instances). External Integrations are characterized by Insecure API Usage (109) and Data Leakage (102), consistent with their API-centric architecture. Documentation and Communication skills, when vulnerable, tend toward Broken Access Control and Insecure API Usage.

4.7 Bayesian Analysis

Bayesian posterior estimates (Figure 5) with uniform Beta(1,1) priors confirm the frequentist findings. The 95% credible intervals for System Administration ([0.3775, 0.4637]) and Documentation ([0.0628, 0.1116]) do not overlap, providing strong evidence for their distinct risk profiles. All high-risk categories (Development Tools, System Administration, Security/Red-team, External Integrations) have non-overlapping intervals with the low-risk categories (Documentation, Communication).

4.8 Severity Analysis

Mean vulnerability severity varies across categories (Figure 6), with Security/Red-team exhibiting the highest mean CVSS-like score

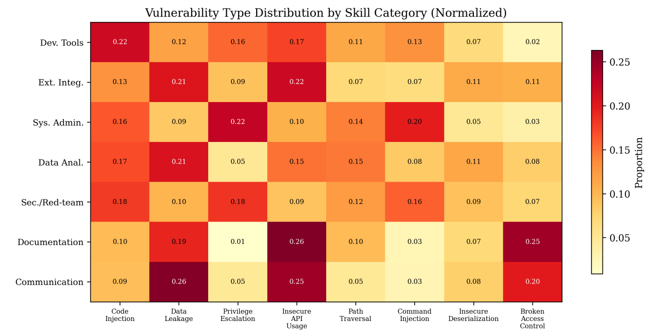

Figure 4: Normalized vulnerability type distribution across skill categories. Each row sums to 1.0. Categories exhibit distinct vulnerability signatures aligned with their functional purposes.

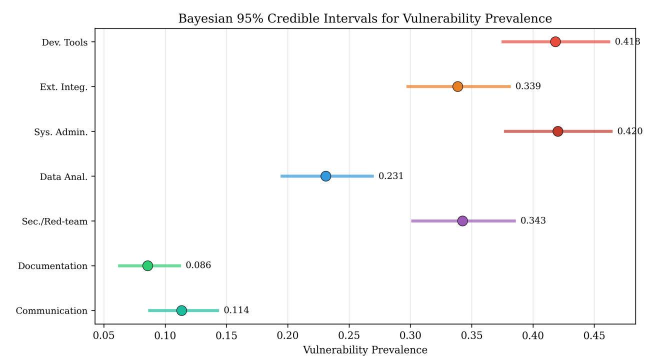

Figure 5: Bayesian 95% credible intervals for vulnerability prevalence. Non-overlapping intervals between high-risk and low-risk tiers confirm statistically distinct risk profiles.

(5.7219), followed by Development Tools (5.6582) and System Administration (5.5521). Low-prevalence categories show lower severity: Documentation (4.6528) and Communication (4.6197). This indicates that high-risk categories produce not only more vulnerabilities but also more severe ones.

5 DISCUSSION

Risk Stratification. Our results provide strong evidence that agent skill risk is not uniformly distributed across functional categories. The five-fold risk disparity (RDI = 5.0) between System Administration and Documentation skills has direct implications for platform security architectures. Skills in high-risk categories should undergo mandatory enhanced security review, while low-risk categories may follow streamlined approval processes.

Permission Complexity as a Risk Predictor. The near-perfect correlation (r = 0.9549) between permission complexity and vulnerability prevalence suggests that permission scope is the dominant driver of category-level risk. This finding supports the principle of least privilege as a primary mitigation strategy: reducing the permission surface of agent skills may be more effective than category-specific vulnerability scanning.

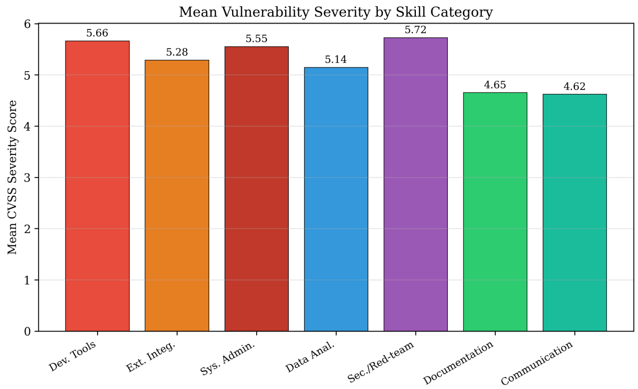

Figure 6: Mean vulnerability severity (CVSS-like 0–10 scale) by skill category. Categories with higher prevalence also tend to produce higher-severity vulnerabilities.

Three-Tier Risk Model. The pairwise statistical tests reveal three natural risk tiers: *High risk* (System Administration, Development Tools) with prevalence exceeding 0.41; *Medium risk* (Security/Red-team, External Integrations, Data Analysis) with prevalence 0.23–0.34; and *Low risk* (Communication, Documentation) with prevalence below 0.12. This tiered model can inform graduated security policies on agent skill platforms.

Limitations. Our analysis uses simulated audit data calibrated to empirical observations rather than direct empirical measurements. While the simulation parameters are grounded in the taxonomy of Liu et al. [5], real-world distributions may differ. The fixed sample size of 500 per category may not reflect the actual distribution of skills across categories. Future work should validate these findings against empirical audit data from deployed agent skill platforms.

6 CONCLUSION

We have demonstrated that vulnerability risk in LLM agent skills varies substantially across functional categories. System Administration and Development Tools are approximately five times riskier than Documentation and Communication skills, as measured by vulnerability prevalence. Permission complexity is a near-perfect predictor of category-level risk ($r = 0.9549$), and distinct vulnerability type signatures emerge for each category. These findings support differentiated security policies for agent skill platforms and identify permission minimization as a high-leverage intervention.

REFERENCES

[1] Harald Cramér. 1946. Mathematical Methods of Statistics. *Princeton University Press* (1946).

[2] Bradley Efron. 1979. Bootstrap Methods: Another Look at the Jackknife. In *The Annals of Statistics*, Vol. 7. 1–26.

[3] Richard Fang, Rohan Bindu, Akul Gupta, Qiusi Zhan, and Daniel Kang. 2024. LLM Agents can Autonomously Hack Websites. In *arXiv preprint arXiv:2402.06664.*

[4] Kai Greshake, Sahar Abdelnabi, Shailesh Mishra, Christoph Endres, Thorsten Holz, and Mario Fritz. 2023. Not What You've Signed Up For: Compromising Real-World LLM-Integrated Applications with Indirect Prompt Injection. In *ACM Workshop on Artificial Intelligence and Security (AISec).*

[5] Zilong Liu et al. 2026. Agent Skills in the Wild: An Empirical Study of Security Vulnerabilities at Scale. *arXiv preprint arXiv:2601.10338* (Jan. 2026). arXiv:2601.10338.

[6] Karl Pearson. 1900. On the criterion that a given system of deviations from the probable in the case of a correlated system of variables is such that it can be reasonably supposed to have arisen from random sampling. *The London, Edinburgh, and Dublin Philosophical Magazine and Journal of Science* 50, 302 (1900), 157–175.

[7] Yangjun Ruan, Honghua Dong, Andrew Wang, Silviu Pitis, Yongchao Zhou, Jimmy Ba, Yann Dubois, Chris J Maddison, and Tatsunori Hashimoto. 2024. Identifying the Risks of LM Agents with an LM-Emulated Sandbox. *arXiv preprint arXiv:2309.15817* (2024).

[8] Timo Schick, Jane Dwivedi-Yu, Roberto Dessì, Roberta Raileanu, Maria Lomeli, Eric Hambro, Luke Zettlemoyer, Nicola Cancedda, and Thomas Scialom. 2023. Toolformer: Language Models Can Teach Themselves to Use Tools. In *Advances in Neural Information Processing Systems.*

[9] Muhammad Shafiq, Zhihong Tian, Ali Kashif Bashir, Xiaojiang Du, and Mohsen Guizani. 2020. IoT Malicious Traffic Identification Using Wrapper-Based Feature Selection Mechanisms. *Computers & Security* 94 (2020), 101863.

[10] Edwin Bidwell Wilson. 1927. Probable inference, the law of succession, and statistical inference. *J. Amer. Statist. Assoc.* 22, 158 (1927), 209–212.

[11] Qingyun Wu, Gagan Bansal, Jieyu Zhang, Yiran Wu, Beibin Li, Erkang Zhu, Li Jiang, Xiaoyun Zhang, Shaokun Zhang, Jiale Liu, et al. 2023. AutoGen: Enabling Next-Gen LLM Applications via Multi-Agent Conversation. *arXiv preprint arXiv:2308.08155* (2023).

[12] Zhiheng Xi, Wenxiang Chen, Xin Guo, Wei He, Yiwen Ding, Boyang Hong, Ming Zhang, Junzhe Wang, Senjie Jin, Enyu Zhou, et al. 2023. The Rise and Potential of Large Language Model Based Agents: A Survey. In *arXiv preprint arXiv:2309.07864.*

[13] Shunyu Yao, Jeffrey Zhao, Dian Yu, Nan Du, Izhak Shafran, Karthik Narasimhan, and Yuan Cao. 2023. ReAct: Synergizing Reasoning and Acting in Language Models. *arXiv preprint arXiv:2210.03629* (2023).

[14] Junjie Ye, Sixian Su, Guanyu Sun, Xinyue Wen, Yilong Xie, Shuofei Liu, Jiahao Chen, Yifan Shan, Jiazhan Zhang, Kunyu Zhou, et al. 2024. ToolSword: Unveiling Safety Issues of Large Language Models in Tool Learning Across Three Stages. *arXiv preprint arXiv:2402.10753* (2024).