On the Ethics of Using LLMs for Offensive Security

Andreas Happe
andreas.happe@tuwien.ac.at
TU Wien
Vienna, Austria

Jürgen Cito
juergen.cito@tuwien.ac.at
TU Wien
Vienna, Austria

ABSTRACT

Large Language Models (LLMs) have rapidly evolved over the past few years and are currently evaluated for their efficacy within the domain of offensive cyber-security. While initial forays showcase the potential of LLMs to enhance security research, they also raise critical ethical concerns regarding the dual-use of offensive security tooling.

This paper analyzes a set of papers that leverage LLMs for offensive security, focusing on how ethical considerations are expressed and justified in their work. The goal is to assess the culture of AI in offensive security research regarding ethics communication, highlighting trends, best practices, and gaps in current discourse.

We provide insights into how the academic community navigates the fine line between innovation and ethical responsibility. Particularly, our results show that 13 of 15 reviewed prototypes (86.6%) mentioned ethical considerations and are thus aware of the potential dual-use of their research. Main motivation given for the research was allowing broader access to penetration-testing as well as preparing defenders for AI-guided attackers.

KEYWORDS

Ethics, Offensive Security, Large Language Models, LLM, Dual-Purpose

1 INTRODUCTION

Large Language Models (LLMs) have rapidly evolved, demonstrating remarkable capabilities in tasks ranging from natural language understanding [22] to code generation [18, 25]. In the realm of cybersecurity, these models are increasingly being applied to offensive security tasks, including vulnerability discovery, penetration testing, and automated exploitation [3, 6, 14, 19, 26, 27, 39, 41, 42, 45]. While these applications showcase the potential of LLMs to enhance security research, they also raise critical ethical concerns regarding the dual-use of offensive security tooling [36].

This paper analyzes a set of papers that leverage LLMs for offensive security, how they address ethical considerations of their work and the reasoning given to justify or mitigate potential risks.

The goal of this paper is to assess the culture of AI for offensive security research papers in communicating ethics in their papers. We aim to highlight emerging trends, best practices, and gaps in current discourse. This analysis contributes to the ongoing discussion [2, 23] about the responsible use of LLMs in security research and provides insights into how the academic community navigates the fine line between innovation and ethical responsibility. Particularly, we found that authors are aware of the potential dual-use of their research but believe that the positive outcomes overweight negative ones. Reasons given for their research include making penetration testing more accessible and preparing defenders for future LLM-based attackers.

2 BACKGROUND

We provide a brief overview of how ethical considerations are expressed within traditional security tooling, and the different established vulnerability disclosure practices as relevant background information.

2.1 Ethics Within Traditional Security Tooling

Security tooling is inherently dual-natured as it can be used both by ethical white-hat hackers for improving defenses as well as by black-hat hackers to attack their victims. We fully agree with the analysis by Zhang et al. [44], stating:

> *For instance, Silic [36] surveys practitioners and finds that empirically practitioners agree that dual-use technology has both benefits and harms, as malicious attackers can use them for harm but good actors can use them for defense. Rad [30] argues that while such technology can be used for harm, restrictions can hinder the benefits of the technology more than the harms, as malicious actors may simply obtain equivalent technology through alternative means such as black markets that are not available to law-abiding actors.*

To the best of our knowledge, there is no academic research on the ethics of open-source penetration testing tooling. Within the security industry, releasing penetration testing tools as open source is common practice [36]. These tools often target a single attack technique, not single exploits, e.g., *sqlmap*[1] targets SQL injection attacks or *gobuster*[2] targeting web URL fuzzing.

Releasing specific exploits for found vulnerabilities, e.g. for *PrintNightmare*[3] or *EternalBlue*[4], are often released utilizing a responsible disclosure process (Section 2.2). A special case can be made for command-and-control (C2) tooling. These tools are typically used for Red-Teaming undercover assignments and could be abused by ransomware or APT groups. Commercial C2 frameworks such as *CobaltStrike*[5] often perform a background check of potential buyers, while open-source tools such as *slither*[6], *mythic*[7], or *havoc*[8] are publicly available as open-source tools.

2.2 Disclosure Practices in Pen-Testing

One of the ethical hackers' goals is to improve security [11]. After a vulnerability has been identified, it is typically reported to the

[1] https://sqlmap.org/
[2] https://github.com/OJ/gobuster
[3] https://en.wikipedia.org/wiki/PrintNightmare
[4] https://de.wikipedia.org/wiki/EternalBlue
[5] https://www.cobaltstrike.com/
[6] https://github.com/BishopFox/sliver
[7] https://github.com/its-a-feature/Mythic
[8] https://github.com/HavocFramework/Havoc

respective author so that remediation can be implemented before releasing information about the vulnerability. Releasing a vulnerability increases the state-of-the-art of security as well as creates prestige for the ethical hacker (which is a part of hackers' motivation as the research is often not monetarily compensated).

When using Coordinated Disclosure [1], the ethical hacker only releases information after a security patch has been released by the software author or vendor. Some vendors argue that there should be an additional disclosure deadline until a released patch is applied by the users of the respective software.

As coordinated disclosure policies are abusable by vendors to delay or prevent the publication of vulnerabilities, many ethical researchers have adopted strict deadlines. For example, Google Project Zero [43] switched to a 90+30 policy in 2021. A vendor has 90 days to release a security update, after a security update has been released Google waits for 30 days until disclosure of the vulnerability. An additional 14 day grace period is allowed upon request. If the vulnerability is already abused in the wild, i.e., is a 0-day, a 7-day disclosure policy replaces the 90-day policy. The Zero-Day Initiative (ZDI), an initiative that reports security research findings to software vendors, utilizes a 120 day disclosure policy [16].

An alternative approach is Full-Disclosure in which security researchers directly report a vulnerability without reporting to the vendor first. The use of full-disclosure arose in the early 2000s as software vendors were often delaying disclosure of vulnerabilities or even threatening to sue security researchers [33]. Full-disclosure often results from bad experiences with prior coordinated disclosure procedures [11]. Originally, coordination disclosure procedures were named Responsible Disclosure. To prevent other forms of disclosure from appearing not responsible compared to responsible disclosure, the term coordinated disclosure is now preferred by the industry.

3 METHODOLOGY

To gather the state-of-the-art on using LLMs for offensive security, we analyzed recent survey papers [3, 6, 14, 19, 26, 27, 39, 41, 42, 45] and identified English papers that were using LLMs to perform offensive security in a penetration-testing context. We analyzed their references to backward-reference the initial papers that utilized LLMs for offensive security research, resulting in both *wintermute* [11] and *pentestGPT* [5]. Using Google Scholar, we performed forward-referencing by adding papers that cited either *pentestGPT* or *wintermute* and fit our initial selection criteria. This process led to the final selection of the 16 papers detailing 15 prototypes utilizing LLMs for offensive security.

Using the selected papers, we performed a thematic analysis [4, 32]. During the initial phase, often referred to as "Familiarizing yourself with the data", all selected papers were read by the authors and initial themes (codes) emerged. In the next phase, we compared the themes derived from the individual authors and integrated them into our final themes seen in Table 1. Finally, all papers were re-read and analyzed using the unified themes. This data is the base for the subsequent analysis presented in this paper.

Threats to Validity. Any literature-based study faces the threat of selection bias (internal threat). To counteract this, we performed forward referencing. Another potential bias would be experimenter bias (internal threat). To reduce the risk, all the data collected was analyzed separately by the different authors and their respective labeling results were compared for differences, and ambiguities were discussed and resolved.

4 RESULTS

The results of our thematic analysis can be seen in Table 1. 13 of the 15 reviewed prototypes (86.6%) contained a mention of ethical considerations. We detected explicit ethics sections, ethical limitation sections, as well as dedicated paragraphs that mention ethics or the potential abuse of the respective attack prototype. A single paper [40] mentioned involvement of an ethics board/IRB.

10 of the 15 papers (66.6%) released their source code or artifacts. Of the five papers that did not provide their artifacts, 3 provided example prompts within their papers or their appendix. One paper [37] additionally mentioned that full sources will be released upon publication. Three papers neither released prompts nor code (also see Section 5.5).

The ubiquitous motivation for using LLMs for offensive security was that penetration tests are fundamental for establishing a good security posture while being costly and resource-expensive. The analysis of LLMs' capabilities for offensive security automation was presented as a potential solution to this problem.

The development of offensive tooling is inherently dual-use and can be problematic when black-hats utilize those tools for illegal activities. As shown in Table 1, 66.6% of publications provided additional justification for their research, with the most common one being "helping defenders prepare" (60%). 20% of papers saw potential for automated penetration-testing using LLMs but claimed in their ethics section that their current prototype is not sophisticated enough for real-world usage (see Section 5.1).

Other reasons given were "*continuation of existing work*" by either other papers or traditional security tooling (two papers), "*empirical evidence that attackers are already using LLMs*" (two papers), or "*providing transparency*" to defenders or decision-makers (two papers, see Section 5.5).

Papers gave mitigations to reduce the risk of LLM-guided penetration-testing. 53% of papers detailed their sandboxed testing environment that prevents performing unintentional harmful operations by the used LLM (see Section 5.1).

Three papers gave concrete recommendations for the detection of LLM-guided penetration-testing or created remediation suggestions within their prototype. Three additional papers mentioned ongoing monitoring of their LLM's activities as remediation. AI regulation was mentioned by three papers (either to provide information to create potential regulation or use regulation as mitigation), although regulation was also noted as being slow to adopt [44].

5 DISCUSSION

5.1 Balance and Consistency

All prototypes use a form of virtualization or sandboxing for their test environment. It is assumed that this was influenced at least partially by safety considerations. These mechanisms typically protect the security researchers' infrastructure, and is not necessarily about protecting others. If a paper details protection mechanisms for its own experiment infrastructure, we would expect an ethics section

Publication	Availability		Motivation			Argument for Publication						Mitigation			
Publication	Ethics Statement	Availability	Capability Evaluation	Education & Training	Preparing Defenders	Continuation of trad. Tooling	Other Paper Released Prototypes	Not Sophisticated enough	Attackers already use LLMs	Defenders must Prepare	Transparency	Protection of Test Environments	Remediation Given	Monitoring Suggested	Regulation
Getting pwned by AI [11]	✓	full	✓	✓	✓				✓	✓					✓
LLMs as Hackers [13]	✓	full	✓		✓			✓	✓	✓		✓			
Autonomously Hack Websites [8]	✓	upon request	✓		✓					✓		✓			
Autonomously Exploit One-day Vulns. [9]	✓	upon request	✓		✓					✓		✓			
Exploit Zero-Day Vulnerabilities [9]		none	✓												
PenHeal [15]	✓	prompts	✓									✓	✓	✓	
AUTOPENBENCH [10]	✓	full	✓			✓		✓				✓			
HackSynth [28]	✓	full	✓		✓					✓					
Vulnbot [24]		full	✓												
Multistage Network Attacks [37]	✓	prompts	✓	✓	✓			✓		✓					
pentestGPT [5]	✓	full	✓											✓	
Can LLMs hack Enterprise Networks? [12]	✓	full	✓		✓					✓	✓	✓	✓		
Towards automated penetration testing [17]	✓	full	✓	✓	✓					✓					
AutoAttacker [40]	✓	prompts	✓		✓					✓		✓	✓		
CyBench [44]	✓	full	✓			✓	✓			✓	✓				✓
NYU CTF Dataset[34, 35]	✓	full	✓												✓

Table 1: Publications using LLMs to perform offensive security. We categorize availability as "full" if the sources and prompts are publicly released, "prompts" if the publication includes example prompts but no sources were released, "none" if neither sources or prompts are included or released, and "upon request" if a publication mentions that sources will be made available upon request (or for reviewers).

detailing the potential abuse of the written prototype (Muszai et al. [28] mention the dual-nature of security tooling within single sentence while detailing securing the experiment setup).

A slight inconsistency occurs if a paper mentions ethical concerns within its background section but does not contain an ethics section itself [24].

We noted inconsistencies between the evaluation and ethics sections. For example, Singer et al. [37] state in their results that "*Using Incalmo, LLMs can autonomously and fully succeed at multistage attack in 5 out of 10 environments and partially succeed in 9 out of 10 environments*". Their experiment scenarios are "*inspired from a mix of public reports of real-world attacks, common topologies, or used in prior work*" and includes simulated critical infrastructure. These substantial improvements and findings are in contrast to the paper's ethics section stating "*Incalmo only has a limited number of attacker capabilities (e.g., only has five exploits), limiting the harm that it could cause in practice*".

5.2 Disclosure of Findings vs. Disclosure of Tooling

Two papers mentioned ongoing disclosure as a reason for not releasing artifacts. Singer et al. [37] (also see Section 5.1) state that they will eventually release artifacts. As they were using a synthetic testbed, there is a question where they disclose the finding to. Fang et al. [8] used their prototype against a curated collection of 50 web-sites. They found a single XSS vulnerability and were not able to contact the creator of the website. This questions the quality of the bug-bounty program within which they were operating their prototype in and how they got the approval for the testing in the first place.

We highlighted the ethics of traditional tooling within our Background section (Section 2). We would argue that LLM-driven tools are similar to generic attack tools, do not fall into the "undercover" C2 area, and do not produce single exploits for vulnerabilities. We highly recommend performing responsible disclosure of found vulnerabilities (Section 2.2) while making the case that generic tooling as LLM-harnesses and prompts can be released similarly to traditional security tooling.

The increased agency of autonomous systems increases the potential fallout, but we would argue that this calls for mitigation measures (such as keeping humans in the loop) instead of non-disclosure of tooling.

5.3 Humans in the Loop

We were investigating papers that utilize LLMs for performing attacks, typically in an autonomous way to allow for comparative benchmarking. In a real-world scenario we would assume that humans would be kept in the loop, i.e., humans have to acknowledge potential destructive operations to prevent unintentional harm. The two earliest papers either keep the human in the loop (for error-correcting purposes, *pentestGPT*) or mention using Human-in-the-Loop techniques for safety reasons (*wintermute*) while later publications do not explicitly mention this. We assume that their focus on measuring and increasing the efficacy of using LLMs for penetration testing mandates autonomous use and real-life deployments of the prototypes will have human oversight added.

5.4 Ethics of using CTF for Education

The analyzed papers contained both papers that focused on the attack prototype as well as papers that focused on creating benchmarks that were evaluated using LLM-driven attack tooling. The latter were primarily benchmark papers and stated common knowledge about CTF ethical usage [34]:

> *Furthermore, the misuse of LLMs to launch sophisticated attacks raises concerns around malicious use [38]. However, the benefit of CTFs in cybersecurity education is well-accepted [20, 21].*

5.5 Safety vs. Transparency in Artifact Disclosure

We see different opinions with regard to artifact disclosure within our reviewed papers. Fang et al. [7–9] propose non-disclosure of artifacts, stating e.g. [8] (emphasis added):

> *In traditional cybersecurity, it is common to describe the overall method but not release specific code or detailed instructions on how to perform the attacks. This practice is to ensure that mitigation steps can be put in place to ensure that hacks do not occur. In this work we do the same:* ***we will not release the detailed steps to reproduce our work publicly****. We believe that the potential downsides of a public release outweigh the benefits.*

This contrasts with disclosure procedures shown in our background section (Section 2.2). Other authors propose transparency and artifact disclosure, e.g., Zhang et al. [44] state:

> *Finally, as scientific researchers, we believe that reproducibility and transparency are central to the AI ecosystem [29, 31]. The reproducibility crisis affecting the sciences has affected machine learning as well, owing to mistakes and/or even fraud and fabrication [29, 31]. While transparency in code, data, and methods is not sufficient to guarantee reproducibility (as mistakes can, of course, occur in the research process), obscurity can ensure irreproducibility. Additionally, releasing our code allows the community to build on our work, helping accelerate scientific progress.*

6 CONCLUSION AND RECOMMENDATIONS

Our analysis has shown that authors are aware of the dual nature of offensive tool research.

We recommend including an ethics section in future papers that clearly states the motivation for the research as well as its potential impact. The impact analysis should be consistent with the evaluation performed within the respective paper. If feasible, mitigations or guidance for detection and monitoring should be given analogous to Indicator-of-Compromise (IoC) often published during attack analysis. If vulnerabilities are found during the research, they should be reported using responsible disclosure mechanisms.

All authors have to decide for themselves on the topic of releasing artifacts (see Section 2.2). We lean towards transparency and would like to finish this paper with a quote from Happe and Cito [12]:

> *Open security tooling ultimately enhances collective cybersecurity.*

REFERENCES

[1] Coordinated vulnerability disclosure. https://en.wikipedia.org/wiki/Coordinated_vulnerability_disclosure. Accessed: 2025-02-19.

[2] Large language models in vulnerability research: Opportunities and responsibilities. https://c3.unu.edu/blog/large-language-models-in-vulnerability-research-opportunities-and-responsibilities. Accessed: 2025-02-20.

[3] Mohamed Boukhlif, Nassim Kharmoum, and Mohamed Hanine. Llms for intelligent software testing: A comparative study. In *Proceedings of the 7th International Conference on Networking, Intelligent Systems and Security*, NISS '24, New York, NY, USA, 2024. Association for Computing Machinery.

[4] Virginia Braun and Victoria Clarke. Using thematic analysis in psychology. *Qualitative research in psychology*, 3(2):77–101, 2006.

[5] Gelei Deng, Yi Liu, Víctor Mayoral-Vilches, Peng Liu, Yuekang Li, Yuan Xu, Tianwei Zhang, Yang Liu, Martin Pinzger, and Stefan Rass. {PentestGPT}: Evaluating and harnessing large language models for automated penetration testing. In *33rd USENIX Security Symposium (USENIX Security 24)*, pages 847–864, 2024.

[6] Rohit Dube. Large language models in information security research: A january 2024 survey. *ResearchGate preprint RG*, 2(20107.26404), 2024.

[7] Richard Fang, Rohan Bindu, Akul Gupta, and Daniel Kang. Llm agents can autonomously exploit one-day vulnerabilities, 2024.

[8] Richard Fang, Rohan Bindu, Akul Gupta, Qiusi Zhan, and Daniel Kang. Llm agents can autonomously hack websites, 2024.

[9] Richard Fang, Rohan Bindu, Akul Gupta, Qiusi Zhan, and Daniel Kang. Teams of llm agents can exploit zero-day vulnerabilities, 2024.

[10] Luca Gioacchini, Marco Mellia, Idilio Drago, Alexander Delsanto, Giuseppe Siracusano, and Roberto Bifulco. Autopenbench: Benchmarking generative agents for penetration testing, 2024.

[11] Andreas Happe and Jürgen Cito. Getting pwn'd by ai: Penetration testing with large language models. In *Proceedings of the 31st ACM Joint European Software Engineering Conference and Symposium on the Foundations of Software Engineering*, pages 2082–2086, 2023.

[12] Andreas Happe and Jürgen Cito. Can llms hack enterprise networks? autonomous assumed breach penetration-testing active directory networks. *arXiv preprint arXiv:2502.04227*, 2025.

[13] Andreas Happe, Aaron Kaplan, and Juergen Cito. Llms as hackers: Autonomous linux privilege escalation attacks. *arXiv preprint arXiv:2310.11409*, 2024.

[14] Mohammed Hassanin and Nour Moustafa. A comprehensive overview of large language models (llms) for cyber defences: Opportunities and directions, 2024.

[15] Junjie Huang and Quanyan Zhu. Penheal: a two-stage llm framework for automated pentesting and optimal remediation. In *Proceedings of the Workshop on Autonomous Cybersecurity*, pages 11–22, 2023.

[16] Zero Day Initiative. Disclosure policy. https://www.zerodayinitiative.com/advisories/disclosure_policy/. Accessed: 2025-02-19.

[17] Isamu Isozaki, Manil Shrestha, Rick Console, and Edward Kim. Towards automated penetration testing: Introducing llm benchmark, analysis, and improvements. *arXiv preprint arXiv:2410.17141*, 2024.

[18] Juyong Jiang, Fan Wang, Jiasi Shen, Sungju Kim, and Sunghun Kim. A survey on large language models for code generation. *arXiv preprint arXiv:2406.00515*, 2024.

[19] Haolin Jin, Linghan Huang, Haipeng Cai, Jun Yan, Bo Li, and Huaming Chen. From llms to llm-based agents for software engineering: A survey of current, challenges and future, 2024.

[20] Zack Kaplan, Ning Zhang, and Stephen V Cole. A capture the flag (ctf) platform and exercises for an intro to computer security class. In *Proceedings of the 27th ACM Conference on on Innovation and Technology in Computer Science Education Vol. 2*, pages 597–598, 2022.

[21] Stylianos Karagiannis, Elpidoforos Maragkos-Belmpas, and Emmanouil Magkos. An analysis and evaluation of open source capture the flag platforms as cybersecurity e-learning tools. In *IFIP World Conference on Information Security Education*, pages 61–77. Springer, 2020.

[22] Nikitas Karanikolas, Eirini Manga, Nikoletta Samaridi, Eleni Tousidou, and Michael Vassilakopoulos. Large language models versus natural language understanding and generation. In *Proceedings of the 27th Pan-Hellenic Conference on Progress in Computing and Informatics*, pages 278–290, 2023.

[23] Wafaa Kasri, Yassine Himeur, Hamzah Ali Alkhazaleh, Saed Tarapiah, Shadi Atalla, Wathiq Mansoor, and Hussain Al-Ahmad. From vulnerability to defense: The role of large language models in enhancing cybersecurity. *Computation*, 13(2):30, 2025.

[24] He Kong, Die Hu, Jingguo Ge, Liangxiong Li, Tong Li, and Bingzhen Wu. Vulnbot: Autonomous penetration testing for a multi-agent collaborative framework. *arXiv preprint arXiv:2501.13411*, 2025.

[25] Feng Lin, Dong Jae Kim, et al. When llm-based code generation meets the software development process. *arXiv preprint arXiv:2403.15852*, 2024.

[26] Harindra S. Mavikumbure, Victor Cobilean, Chathurika S. Wickramasinghe, Devin Drake, and Milos Manic. Generative ai in cyber security of cyber physical systems: Benefits and threats. In *2024 16th International Conference on Human System Interaction (HSI)*, pages 1–8, 2024.

[27] Farzad Nourmohammadzadeh Motlagh, Mehrdad Hajizadeh, Mehryar Majd, Pejman Najafi, Feng Cheng, and Christoph Meinel. Large language models in cybersecurity: State-of-the-art, 2024.

[28] Lajos Muzsai, David Imolai, and András Lukács. Hacksynth: Llm agent and evaluation framework for autonomous penetration testing, 2024.

[29] National Academies of Sciences, Policy, Global Affairs, Board on Research Data, Information, Division on Engineering, Physical Sciences, Committee on Applied, Theoretical Statistics, Board on Mathematical Sciences, et al. *Reproducibility and replicability in science*. National Academies Press, 2019.

[30] Tiffany S Rad. The sword and the shield: Hacking tools as offensive weapons and defensive tools. *Geo. J. Int'l Aff.*, 16:123, 2015.

[31] David B Resnik and Adil E Shamoo. Reproducibility and research integrity. *Accountability in research*, 24(2):116–123, 2017.

[32] Collin Robson. Real world research, 2002.

[33] Bruce Schneier. Schneier: Full disclosure of security vulnerabilities a 'damned good idea'. https://www.schneier.com/essays/archives/2007/01/schneier_full_disclo.html, 2007. Accessed: 2025-02-19.

[34] Minghao Shao, Boyuan Chen, Sofija Jancheska, Brendan Dolan-Gavitt, Siddharth Garg, Ramesh Karri, and Muhammad Shafique. An empirical evaluation of llms for solving offensive security challenges, 2024.

[35] Minghao Shao, Sofija Jancheska, Meet Udeshi, Brendan Dolan-Gavitt, Haoran Xi, Kimberly Milner, Boyuan Chen, Max Yin, Siddharth Garg, Prashanth Krishnamurthy, Farshad Khorrami, Ramesh Karri, and Muhammad Shafique. Nyu ctf dataset: A scalable open-source benchmark dataset for evaluating llms in offensive security, 2024.

[36] Mario Silic. Dual-use open source security software in organizations–dilemma: help or hinder? *Computers & Security*, 39:386–395, 2013.

[37] Brian Singer, Keane Lucas, Lakshmi Adiga, Meghna Jain, Lujo Bauer, and Vyas Sekar. On the feasibility of using llms to execute multistage network attacks. *arXiv preprint arXiv:2501.16466*, 2025.

[38] Jan Vykopal, Valdemar Švábenskỳ, and Ee-Chien Chang. Benefits and pitfalls of using capture the flag games in university courses. In *Proceedings of the 51st ACM Technical symposium on computer science education*, pages 752–758, 2020.

[39] Hanxiang Xu, Shenao Wang, Ningke Li, Kailong Wang, Yanjie Zhao, Kai Chen, Ting Yu, Yang Liu, and Haoyu Wang. Large language models for cyber security: A systematic literature review, 2024.

[40] Jiacen Xu, Jack W Stokes, Geoff McDonald, Xuesong Bai, David Marshall, Siyue Wang, Adith Swaminathan, and Zhou Li. Autoattacker: A large language model guided system to implement automatic cyber-attacks. *arXiv preprint arXiv:2403.01038*, 2024.

[41] Yifan Yao, Jinhao Duan, Kaidi Xu, Yuanfang Cai, Zhibo Sun, and Yue Zhang. A survey on large language model (llm) security and privacy: The good, the bad, and the ugly. *High-Confidence Computing*, 4(2):100211, 2024.

[42] Yagmur Yigit, William J Buchanan, Madjid G Tehrani, and Leandros Maglaras. Review of generative ai methods in cybersecurity, 2024.

[43] Google Project Zero. Vulnerability disclosure policy. https://googleprojectzero.blogspot.com/p/vulnerability-disclosure-policy.html. Accessed: 2025-02-19.

[44] Andy K Zhang, Neil Perry, Riya Dulepet, Joey Ji, Justin W Lin, Eliot Jones, Celeste Menders, Gashon Hussein, Samantha Liu, Donovan Jasper, et al. Cybench: A framework for evaluating cybersecurity capabilities and risks of language models. *arXiv preprint arXiv:2408.08926*, 2024.

[45] Jie Zhang, Haoyu Bu, Hui Wen, Yu Chen, Lun Li, and Hongsong Zhu. When llms meet cybersecurity: A systematic literature review, 2024.