Forewarned is Forearmed: A Survey on Large Language Model-based Agents in Autonomous Cyberattacks

MINRUI XU, Nanyang Technological University, Singapore
JIANI FAN, Nanyang Technological University, Singapore
XINYU HUANG, University of Waterloo, Canada
CONGHAO ZHOU, University of Waterloo, Canada
JIAWEN KANG, Guangdong University of Technology, China
DUSIT NIYATO, Nanyang Technological University, Singapore
SHIWEN MAO, Auburn University, USA
ZHU HAN, University of Houston, USA
XUEMIN (SHERMAN) SHEN, University of Waterloo, Canada
KWOK-YAN LAM, Nanyang Technological University, Singapore

With the continuous evolution of Large Language Models (LLMs), LLM-based agents have advanced beyond passive chatbots to become autonomous cyber entities capable of performing complex tasks, including web browsing, malicious code and deceptive content generation, and decision-making. By significantly reducing the time, expertise, and resources, AI-assisted cyberattacks orchestrated by LLM-based agents have led to a phenomenon termed Cyber Threat Inflation, characterized by a significant reduction in attack costs and a tremendous increase in attack scale. To provide actionable defensive insights, in this survey, we focus on the potential cyber threats posed by LLM-based agents across diverse network systems. Firstly, we present the capabilities of LLM-based cyberattack agents, which include executing autonomous attack strategies, comprising scouting, memory, reasoning, and action, and facilitating collaborative operations with other agents or human operators. Building on these capabilities, we examine common cyberattacks initiated by LLM-based agents and compare their effectiveness across different types of networks, including static, mobile, and infrastructure-free paradigms. Moreover, we analyze threat bottlenecks of LLM-based agents across different network infrastructures and review their defense methods. Due to operational imbalances, existing defense methods are inadequate against autonomous cyberattacks. Finally, we outline future research directions and potential defensive strategies for legacy network systems.

CCS Concepts: • **Networks** → *Network security*; • **Computing methodologies** → *Artificial intelligence*; • **General and reference** → Surveys and overviews.

Additional Key Words and Phrases: Large Language Models (LLMs), Cybersecurity, Autonomous Cyberattacks, Network Security.

1 Introduction

1.1 Background and Motivation

The evolving capabilities of large language models (LLMs) are rapidly transforming attack and defense operations in cybersecurity [80]. Major AI companies have begun to systematically evaluate these risks using the Cyber Kill Chain Framework [127, 161]. For instance, Google's Project Naptime team has demonstrated that frontier LLMs can

Authors' Contact Information: Minrui Xu, MINRUI001@e.ntu.edu.sg, Nanyang Technological University, Singapore; Jiani Fan, jiani001@e.ntu.edu.sg, Nanyang Technological University, Singapore; Xinyu Huang, x357huan@uwaterloo.ca, University of Waterloo, Waterloo, ON, Canada; Conghao Zhou, c89zhou@uwaterloo.ca, University of Waterloo, Waterloo, ON, Canada; Jiawen Kang, kavinkang@gdut.edu.cn, Guangdong University of Technology, Guangzhou, Guangdong, China; Dusit Niyato, dniyato@ntu.edu.sg, Nanyang Technological University, Singapore; Shiwen Mao, smao@ieee.org, Auburn University, USA; Zhu Han, hanzhu22@gmail.com, University of Houston, Houston, TX, USA; Xuemin (Sherman) Shen, sshen@uwaterloo.ca, University of Waterloo, Waterloo, ON, Canada; Kwok-Yan Lam, kwokyan.lam@ntu.edu.sg, Nanyang Technological University, Singapore.

autonomously assist in offensive security tasks with minimal human input, including code exploitation and vulnerability discovery [75]. Similarly, Anthropic has deployed red teams, i.e., offensive attackers, to test its Claude models against cybersecurity misuse scenarios, revealing new emergent risks in autonomous agent behavior [23]. These findings reinforce the concern that LLMs have significantly lowered the technical threshold and cost of multi-stage intrusions [175]. Leveraging LLMs equipped with perception, memory, reasoning, and action modules, LLM-based agents can conduct cyberattacks autonomously with minimal human intervention [47, 107]. Specifically, LLM-based agents introduce novel attack paradigms, e.g., jailbreak attack [170], and significantly amplify existing cyberattacks, e.g., vulnerability exploitation, malware generation, and social engineering [38]. LLM-based agents allow attackers with limited skills and resources to conduct complex cyberattacks with minimal human intervention. Therefore, LLM-based agents accelerate attack deployment, scale offensive activities, and erode traditional resource bottlenecks, resulting in Cyber Threat Inflation. This inflation in cybersecurity describes the drastic reduction in operational costs for launching cyberattacks alongside a significant increase in their scalability.

LLM-based agents can reduce time, expertise, and resource requirements across all stages of cyberattacks, e.g., vulnerability detection, customized exploitation, and persistent installation [161]. Therefore, cyberattacks that previously required months of labor and substantial expert involvement can now be accomplished within hours [157]. In addition to cost collapse, scale uplift manifests in three critical dimensions [18]. First, capability uplift refers to the automation of offensive tasks such as vulnerability scanning and social engineering, traditionally limited to skilled red-team experts. For instance, PentestGPT [52] demonstrates a 228.6% increase in task completion, and RapidPen [132] achieves shell access within 200–400 seconds at an estimated cost of $0.3–$0.6 per run, with a 60% success rate. Second, throughput uplift captures the ability of LLM-based agents to execute continuous and large-scale attacks in parallel. To generate next-packet predictions based on previous traffic context in unmanned aerial vehicle (UAV) networks, Net-GPT [151] achieves 95% packet-generation accuracy and maintains time man-in-the-middle (MitM) sessions for 30 min without expert intervention. Finally, autonomous risk emergence highlights how LLMs with reasoning abilities can dynamically adapt to defensive mechanisms. In satellite networks, PLLM-CS [85] autonomously interprets satellite telemetry to detect intent-based anomalies, signaling the rise of real-time, self-adjusting adversarial agents.

While advanced persistent threat (APT) groups often leverage sophisticated techniques such as advanced phishing [42], zero-day exploitation [222], and polymorphic malware [162], individual attackers have also demonstrated the ability to execute similar methods. However, the emergence of LLM-based agents will empower individual attackers to achieve sophisticated attacks. Through the integration of LLMs with tool APIs and accessible programming interfaces, organizations with limited technical capabilities are now able to orchestrate complex operations, encompassing systematic vulnerability assessment and coordinated exploit deployment. This transformation has effectively dismantled the traditional security asymmetry between attackers and defenders, as sophisticated attack vectors are no longer restricted to well-resourced threat actors. Furthermore, an LLM-based agent might probe systems outside of typical human working hours or adapt in real-time. Therefore, defenses should remain vigilant at all times to detect and respond to these autonomous intrusions. Consequently, the potential for widespread, cost-effective system compromises has expanded dramatically.

The cyber threat inflation has profound implications for legacy network infrastructures, including enterprise networks, cellular core networks, cloud platforms, and embedded systems. However, many applications of LLMs in cybersecurity still assume traditional threat models, where human attackers remain the principal adversaries [144]. While human attackers have traditionally posed the primary threat in cybersecurity, integrating LLMs now augments or even substitutes their expertise in many domains. Accordingly, LLMs are employed to enhance performance across established cybersecurity tasks and evaluation benchmarks.For instance, LLMCloudHunter [164] and AppPoet [218] present targeted solutions

Table 1. Related works on LLM Agents, cyberattacks, and network systems.

Ref.	Survey Focus	LLM agents	Cyber-attacks	Networks
[189]	Architecture, capabilities, applications, and evaluation of LLM-based agents	✓	×	×
[123]	The life-cycle of LLM agents including construction, collaboration, and evolution	✓	×	×
[97]	LLM applications in software engineering and evolution into agents	✓	×	×
[86]	LLM-based multi-agent systems for software engineering and human-in-the-loop	✓	×	×
[214]	LLMs for cybersecurity tasks like threat intelligence and vulnerability detection	×	✓	×
[65]	Benchmarking 42 LLMs on intrusion and malware detection tasks	×	✓	×
[221]	Evaluation of 37 LLMs for bug detection and patch generation	×	✓	×
[27]	LLMs for code security, strengths in simple flaws and weaknesses in complex issues	×	✓	×
[80]	Frontier AI's impact on cybersecurity landscapes	×	✓	×
[11]	LLMs for malware detection, task taxonomies, metrics, and countermeasure	×	✓	×
[95]	LLM usage in code analysis, malware detection, and reverse engineering	×	✓	×
[135]	LLM-specific threats and defense pipelines in 6G networks	×	✓	✓
[58]	Cyberattacks on cyber-physical systems; threat modeling and defense synthesis	×	✓	✓
[37]	ML-enabled attacks on IoT networks; evaluation challenges and defense gaps	×	✓	✓
[193]	Metaverse fundamentals, emerging security threats, and privacy challenges	×	✓	✓
Ours	Cyberattack capabilities of LLM-based agents across various network systems	✓	✓	✓

for cloud threat intelligence extraction and Android malware detection, respectively, yet lack a systematic analysis for LLM-based cyberattack agents across different types of networks.

1.2 Related Works

As summarized in Table 1, the capabilities of LLM-based agents have expanded from simple chatbots to sophisticated copilots in cybersecurity, although their deployment across diverse network environments is still under investigation. From an architectural perspective, Wang *et al.* [189] provide a comprehensive review of LLM-based autonomous agents, focusing on their construction, capabilities, applications, and evaluation. Adopting a life cycle perspective, Luo *et al.* [123] categorize LLM-based agents into three dimensions, i.e., construction, collaboration, and evolution, encompassing components from profile definition to deployment in real-world settings. With a domain-specific focus, Jin *et al.* [97] review LLM applications in software engineering across six domains, examine their evolution into autonomous agents, and propose a taxonomy of agent capabilities. Considering multi-agent systems, He *et al.* [86] investigate LLM-based multi-agent systems in software engineering, outlining a two-phase research agenda while emphasizing the critical role of human-in-the-loop approaches for system advancement.

LLM adaptation and evaluation for cybersecurity applications have recently been mapped out in several complementary surveys. From a methodological standpoint, Zhang *et al.* [214] survey adaptation techniques that repurpose foundation models for threat intelligence and vulnerability detection, and articulate open challenges and integration requirements. From an empirical perspective, Ferrag *et al.* [65] benchmark 42 LLMs across intrusion and malware-detection tasks. In vulnerability remediation, Zhou *et al.* [221] assess 37 LLMs on bug detection and patch generation, advocating for

higher-quality datasets and tighter workflow integration. At the code level, Basic *et al.* [27] show that while LLMs handle simple flaws well, they still struggle with complex security issues. Adopting a policy lens, Guo *et al.* [80] analyze frontier AI's security impact through case studies and controlled evaluations, provide recommendations for policymakers, and identify pressing research gaps. Focusing on malware, Al *et al.* [11] summarize core concepts, task taxonomies, and evaluation metrics, and propose a risk-mitigation framework that balances theory with practice. Finally, Jelodar *et al.* [95] review LLM-enabled code analyses for malware detection, compare fine-tuning strategies, and highlight challenges.

Security risks and defenses for network systems, from 6G to cyber-physical infrastructures and the metaverse, have been scrutinized in recent research. For 6G, Nguyen *et al.* [135] survey LLM threats, vulnerabilities, mitigation pipelines, and blockchain integration for secure deployment. For cyber-physical systems, Duo *et al.* [58] analyze attacks and defense techniques. For IoT, Bout *et al.* [37] examine ML-enabled attacks, their advantages, and defense gaps. For metaverse, Wang *et al.* [193] map security threats and propose safeguards across human, physical, and digital realms. Different from these surveys, we provide a network-centric review examining LLM-based agents, their cyberattack capabilities, and impact across network paradigms. By understanding the core challenges of LLM-based agents like hallucination and context windows, defenders can identify weaknesses.

1.3 Contributions

Conventional perspectives in cybersecurity often overlook that LLM-based autonomous agents can be both defenders and adversaries, contributing to Cyber Threat Inflation to legacy systems [161]. This oversight reveals a gap in current cybersecurity research. To fill this gap, we provide a comprehensive taxonomy and comparative analysis of LLM-based agents in autonomous cyberattacks. We emphasize that LLM-based agents are not just tools for defenders but can become the adversaries themselves. Blue teams, i.e., defensive protectors, defending against cyberattacks, should update threat models by considering LLM-based agents as potential attackers and recognizing novel threat dynamics. We categorize research across attack chains and examine manifestations in static, mobile, and infrastructure-free networks. Existing analysis shows how LLM-based agents reduce attack costs while creating new defense challenges through automation and operational asymmetries.

In this survey, we investigate LLM-based autonomous agents as cyber adversaries operating autonomously. We begin by decomposing each LLM-based agent into five fundamental modules, consisting of models, perception, memory, reasoning & planning, and actions. Then, we demonstrate how multiple agents can collaborate with humans and other agents to deliver end-to-end attacks autonomously. Additionally, we examine how cyberattacks become more cost-effective and scalable while generating new forms of autonomous risk across network systems with diverse infrastructure characteristics. By highlighting classic defense methods that struggle against LLM-driven attacks, the paper provides advice for blue teams on where to watch out. The main contributions of this survey can be summarized as follows.

(1) We present a novel unified architecture that abstracts the common design patterns of existing LLM-based cyberattack agents. This architecture comprises components for model selection, perception, memory, reasoning&planning, and tools&actions. We demonstrate that cooperative multi-agent orchestration can enable autonomous cyber operations.
(2) We present a taxonomy of eight representative cyberattack capabilities for LLM-based agents, and analyze the specific attack bottlenecks and limitations these agents face in executing each of these capabilities.
(3) We demonstrate how the cyberattack capabilities of LLM-based agents manifest across different network paradigms, including static infrastructure networks, mobile infrastructure networks, and infrastructure-free networks.

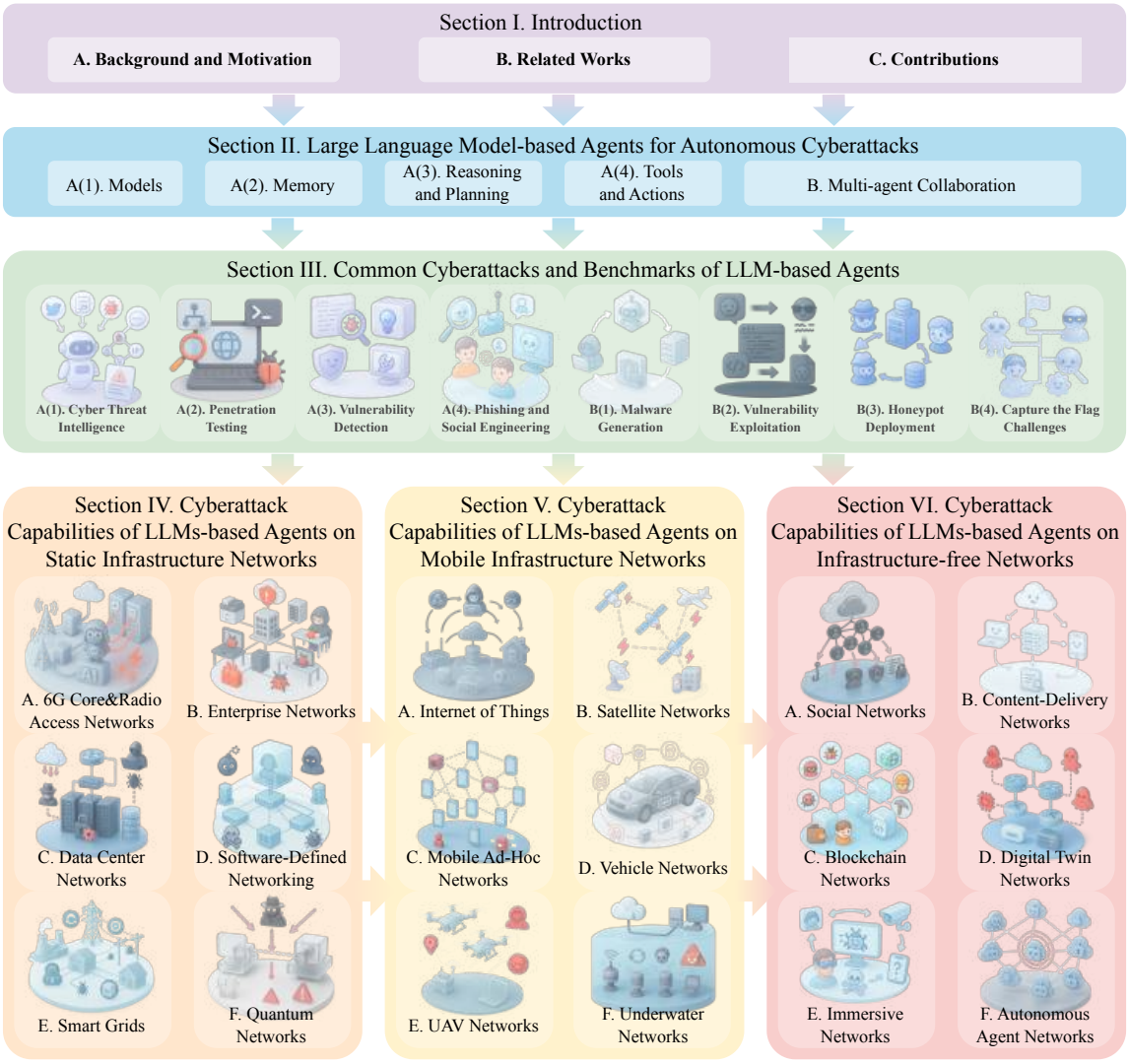

Fig. 1. The outline of this paper.

Section II deconstructs the construction and collaboration of LLM-based cyberattack agents. Section III presents common cyberattack capabilities of LLM-based agents and benchmarks. Sections IV, V, and VI then analyze how those capabilities manifest in three network paradigms, including static infrastructure networks, mobile infrastructure networks, and infrastructure-free networks, respectively. An overview of this survey's structure is shown in Fig. 1. With this survey, we provide a clear direction of how LLM-enabled adversaries evolve across capabilities and network systems. The analysis serves as a reference for blue-team defenders preparing defenses to track the state-of-the-art adversaries.

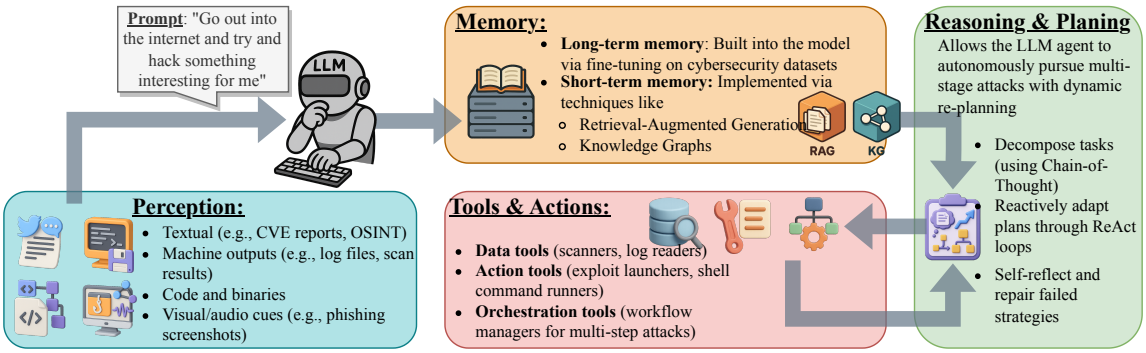

Fig. 2. LLM-based cyberattack agent construction. This architecture enables the agent to ingest diverse input types, store and retrieve contextual knowledge, adaptively plan multi-stage attacks, and interact with tools to perform cyberattacks.

2 Large Language Model-based Agents in Autonomous Cyberattacks

Cyberattack agents built on top of LLMs with external modules that map high-level natural-language objectives to concrete offensive actions [212]. Fig. 2 illustrates the modular architecture of LLM-based cyberattack agents, whose core module is an LLM, while perception, memory, reasoning, and actuation are provided by external APIs or tool wrappers.

2.1 LLM-based Agent Construction

2.1.1 Models. LLM-based agents often leverage state-of-the-art pre-trained foundation models or fine-tuned specialized models on cybersecurity datasets as their “brain" to process prompts and understand network environments. As listed in Table II, LLM-based cyberattack agents are typically equipped with state-of-the-art LLMs (e.g., GPT-3.5/4 or Llama) as their core due to these models’ generalized world knowledge and strong reasoning capabilities [25, 146, 194]. With the continuous improvement of pre-trained LLMs [25, 130, 155], in terms of larger context and better reasoning, more potent attacks can be performed by LLM-based agents. While cloud-based LLMs are commonly used, attackers may prefer running open-source models on local servers to evade detection via API logs from cloud data centers. To address the limitations of using external APIs or very large models, recent studies focus on fine-tuning smaller open-source LLMs for security-specific tasks. For example, Rigaki *et al.* [159] fine-tune a 7B-parameter model, named Hackphyr, as a local red-team agent for network security. The resulting model runs on a single GPU and matches GPT-4 while even outperforming OpenAI’s GPT-3.5-turbo on complex network intrusion scenarios, owing to its training on a purpose-built cybersecurity dataset. Likewise, Ahmed *et al.* [6] introduce AttackLLM for industrial control systems (ICS), demonstrating that LLM-generated attack patterns can exceed human-crafted ones in both quality and diversity. This method combines data-centric and design-centric methodologies to autonomously produce diverse and realistic attack scenarios without relying on expensive physical testbeds. However, LLMs have their respective limitations, such as context size, knowledge cutoff, and tendency to hallucinate, which can be estimated by using benchmarks and evaluation systems. After successfully identifying the LLMs, defenders can exploit these weaknesses. Table 2 compares leading LLMs in terms of architecture size, context window, inference speed, pricing, and MMLU benchmark performance.

Benchmarks and Evaluation: In recent research, benchmarks and evaluation frameworks have been developed to assess the performance and safety of LLM-based agents [22, 145, 150, 155]. Early studies focused on broad evaluations of model capabilities. These provided general insights but lacked task-level granularity. To address this gap, Yu *et al.* [209] propose

Table 2. Comparison of state-of-the-art LLMs (May 2025).
Context window in tokens, speed in tokens/second, prices in USD per million tokens [24].

Company	Model	Parameters	Context Window	Speed	Input Price	Output Price	MMLU
OpenAI	GPT-o3	—	1M	77	$10.00	$40.00	**0.853**
	GPT-4o	—	128k	164	$5.00	$15.00	0.803
Meta	Llama 4 Maverick	400B	1M	121	$0.20	$0.85	0.809
	Llama 3.3	70B	128k	110	$0.59	$0.70	0.713
Google	Gemini 2.5	—	1M	160	$1.25	$10.00	0.800
	Gemini 2.0	—	1M	205	$0.07	$0.30	0.724
Anthropic	Claude 3.7 Sonnet	—	200k	77	$3.00	$15.00	0.803
	Claude 3.5 Haiku	—	200k	66	$0.80	$4.00	0.634
Mistral AI	Mixtral 8×7B	56B	33k	80	$0.70	$0.70	0.387
DeepSeek	R1	671B	130k	24.6	$0.55	$2.219	0.844
xAI	Grok 3	2.7T	1M	49	$3.00	$15.00	0.799

CS-Eval. It includes eleven cybersecurity tasks, such as vulnerability management and penetration testing, covering knowledge, reasoning, and application. Shifting the focus toward safety and misuse, Andriushchenko *et al.* [22] introduce AgentHarm. It contains 110 harmful tasks grouped into eleven categories, such as fraud, cybercrime, and harassment. Their results show that even advanced models follow unsafe instructions. Mazeika *et al.* [126] extend this work with HarmBench. By including a wide array of harmful behaviors, both textual and multimodal, designed to violate laws or norms, they find that none of model is fully robust. This holds even with strong alignment techniques. Later, Yuan *et al.* [210] present R-Judge, which evaluates risk awareness in multi-step decisions.

2.1.2 Perception. Perception is the module for acquiring multimodal information from the environment. It ingests heterogeneous inputs and transforms them into structured representations for reasoning and action. In cyberattacks, the input stream extends beyond human text, where an autonomous cyberattack agent encounters at least four distinct sensory channels [214]: i) **Textual OSINT and Human Prose:** Including tweets, dark-web forum discussions, common vulnerabilities and exposures (CVE) advisories, and incident response blogs; ii) **Machine Traces:** Encompassing Nmap/Masscan scan banners, Nessus XML outputs, system log entries, and NetFlow or PCAP packet captures; iii) **Program Artefacts:** Such as source code snippets, abstract syntax tree or control flow graph fragments, disassembled binaries, and container manifests; iv) **Diagrammatic and Audiovisual Cues:** Including screenshots of phishing webpages, network topology diagrams, or VoIP samples encountered in vishing campaigns. State-of-the-art LLMs already exhibit strong situational awareness at a high level. For example, GPT-4 achieves an F1 score of approximately 0.94 when classifying cyber threat posts from Twitter feeds [115, 167]. Incoming artefacts are tokenized and embedded with the LLM encoder. Then, vectors enter the short-term buffer and are condensed into schema triples for the long-term store, enabling retrieval and planning.

2.1.3 Memory. LLM-based agents demand a well-structured module for maintaining both **long-term** memory and **short-term** memory. This dual-memory architecture, increasingly adopted in LLM-based agent designs [120, 189, 198], enables agents to consider both static cybersecurity knowledge and dynamic environmental information.

Long-term Memory: Long-term memory refers to the static repository of cybersecurity knowledge internalized by the agent during pretraining or fine-tuning stages. Such memory is critical for providing agents with foundational expertise

on vulnerabilities, exploits, attack vectors, and defensive protocols. Recent efforts have curated specialized cybersecurity corpora to enhance this aspect. PRIMUS [208] aggregates extensive open-source cybersecurity data, including vulnerability advisories, exploit scripts, and traffic captures, forming a comprehensive 18GB corpus designed for LLM pretraining. Similarly, ATTACKER [51] is a named-entity recognition benchmark for attribution tasks. SECQA [121] is a cybersecurity-focused Q&A corpus. CMDCALIPER [92] is a semantic mapping of command-line activities to enrich the agent's long-term memory base. Through training on these corpora, LLM-based agents develop internal cybersecurity models to spot threats, find weaknesses, and adapt to new attacks.

Short-term Memory: In addition to static knowledge, LLM-based agents must dynamically manage real-time information encountered during cyberattack operations. Limited by context windows of LLMs, agents can leverage short-term memory techniques, such as Retrieval-Augmented Generation (RAG) [72] systems and Knowledge Graphs (KGs) [141], to leverage external non-parametric knowledge during each single attack.

1) Retrieval-Augmented Generation. RAG lets agents access knowledge sources for LLM prompts. This enables operation on the latest threat intelligence without retraining. For instance, Daneshvar *et al.* [49] demonstrate that a RAG-enhanced vulnerability scanner can improve the accuracy of vulnerability detection by 70%.

2) Knowledge Graphs. KGs provide structured memory for agents, where nodes represent systems and vulnerabilities, and edges show relationships. LLM-assisted KG construction tools ATTACKG [215], CTI-KG [91], and CTI-NEXUS [44] extract threat knowledge graphs from reports. KGs can maintain operational coherence in multi-stage attacks.

RAG enables millisecond-level recall of short-term memory, while the KG provides triples for causal reasoning.

2.1.4 Reasoning and Planning. Unlike static bots, LLM-based agents can reason through failures and change tactics on the fly. State-of-the-art foundation models, e.g., GPT-4o and GPT-4-o3 variant, already expose latent chain-of-thought (CoT) traces when prompted appropriately, providing single-agent multi-step reasoning even before any task-decomposition scaffolding is applied. LLM-based agents need to execute multi-stage operations and adjust to defensive responses when conducting autonomous cyberattacks, which are accomplished through three core reasoning methods:

1) Task-decomposition Reasoning: Each agent is first prompted to expose its CoT [196] to perform multi-step reasoning for complex tasks. Dwight *et al.* in [59] show how repeated CoT prompting lets an LLM develop an attack tree, where each node is a prerequisite or sub-goal. Beyond CoT, **tree-/graph-of-thoughts** [31, 190, 202] prompting allows the agent to branch early and explore several candidate paths in parallel till the most promising one.

2) ReAct Planning: After an initial plan is drafted, the agent enters a *Reason-Act* loop [203], which enables dynamic re-planning. For instance, Paul *et al.* [149] report a marked increase in exploit success rate when every action is immediately scrutinised by the model's follow-up reasoning. As AI attackers plan based on the feedback they receive, feeding misleading or confusing information can derail their reasoning.

3) Self-reflection and Auto-repair: LLM-based agents further embed a light-weight "critic" that reviews the latest CoT or action log, flags contradictions or dead ends, and triggers a self-correction cycle [159, 219]. Crimson agent [98] can couple scenario simulation with rule-based sanity checks. In this way, an exploit that lands a low-privilege shell is automatically followed by privilege-escalation suggestions. Crimson develops a comprehensive CVE-to-ATT&CK Mapping dataset and implements Retrieval-Aware Training to improve model performance. Using a model fine-tuned with 7 billion parameters and the Low-Rank Adaptation (LoRA) technique [88], they achieve results comparable to GPT-4 while showing lower rates of hallucination and errors.

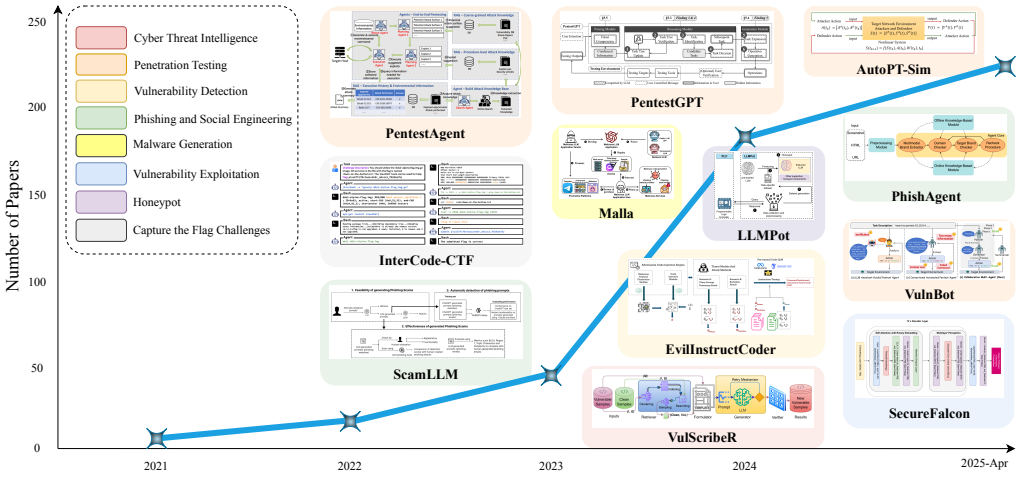

Fig. 3. The timeline of LLM-based agent development and their increasing capabilities in cyberattacks.

Task-decomposition reasoning creates a static attack tree of nested sub-goals through chain-of-thought prompting. ReAct planning then combines this reasoning with real-time feedback loops to refine each step. Finally, self-reflection layers act as an internal critic, iteratively fixing errors and eliminating dead ends in the evolving plan.

2.1.5 Action and Tools. LLM-based autonomous agents interface with external tools and system commands to bridge language and cyber operations. These agents can execute actions directly through tools for running commands, exploits, and scanning. To enable this functionality, developers standardize the interface between LLMs and actionable tools by organizing them into three categories [214]:

1) Data tools support passive information gathering and reconnaissance. Examples include file-system readers, port scanners, vulnerability enumerators, and HTTP request handlers.

2) Action tools enable active manipulation of the environment. These include file-system operations, network scans, exploit payload launches, authentication attempts, and other system-altering actions.

3) Orchestration tools coordinate complex workflows, allowing the agent to sequence multiple sub-actions or delegate subtasks to specialized routines, effectively building multi-stage attack chains.

LLM-based agents are provided with a predefined and controlled set of callable tools or APIs for execution [154]. Therefore, defenders can monitor the usage of powerful administrative and network tools, preventing unauthorized automated operations through implementing white lists or two-factor authentication.

For tool-using benchmarks, Ristea *et al.* in [160] propose the AI Cyber Risk Benchmark to test LLM agents' exploit capabilities in controlled environments. For instance, Fang *et al.* in [62] demonstrate an LLM-based agent with web tools that found and exploited vulnerabilities through attack stages. Granting LLM-based agents access to powerful tools also raises significant safety risks. Kim *et al.* [103] highlight the emerging dangers of web-enabled LLMs, warning that once agents can act on the open Internet, they can perform unintended or malicious operations. Consequently, strict controls are imposed on tool access, and agents are typically confined to isolated testbeds to mitigate real-world risk. Following this principle, Bhatt *et al.* [32, 33] developed the CyberSecEval suite, providing a standardized evaluation framework that

tests agents across a wide range of cybersecurity tasks while ensuring all actions occur within a controlled environment. The development roadmap of LLM-based cyberattacks is shown in Fig. 3.

2.2 Multi-agent Collaboration

Multiple LLM-based agents can collaborate, e.g., one does scanning, another exploits, another handles exfiltration, to perform a complex attack [19, 35, 105]. For example, the Audit-LLM framework for insider threat detection proposed by Song *et al.* in [177] employs three types of agents to analyze security logs, including planner agents, specialist agents, and analyst agents. Furthermore, multi-agent cyberattacks can also adopt adversarial or competitive roles as a form of collaboration. For instance, one agent might act as the attacker while another acts as a defender or as a cautious evaluator, effectively red-teaming each other's strategies. Wang *et al.* in [191] explore an RL-driven agent that autonomously attacks other LLM-based systems. In such scenarios, agents can iteratively improve offensive tactics and defensive countermeasures through simulation.

2.3 Lessons Learned for Blue Teams

(1) **Utilize Model Limitations:** While attackers will use state-of-the-art LLMs for attacks, each model has limitations, e.g., context length limits, knowledge cutoff dates, and tendency to hallucinate. If defenders know which specific LLM an attacker might use, they can exploit these weaknesses.
(2) **Designed Traps in Multi-Stage Attacks:** The multi-stage reasoning of LLM-based agents means that they can complete reconnaissance, exploitation, and post-exploitation faster than humans since they do not need pauses. Blue teams can implement defensive countermeasures such as setting up automated incident response tasks with specific reasoning times during the observe-orient-decide-act (OODA) loop to prevent LLM-based agents from fully executing their attack chain.
(3) **Leverage Multi-Agent Defense:** Blue teams can deploy multiple defensive LLM-based agents. One agent monitors networks, another watches files, and a third responds to threats. These agents work together by sharing data to counter various attacks.

3 Common Cyberattacks and Benchmarks of LLM-based Agents

As summarized in Table 3, each LLM ability maps differently across cyberattack types, with perception and memory dominating reconnaissance tasks while reasoning, planning, and tool orchestration drive exploitation workflows. The LLM-based agent frameworks for cyberattacks are listed in Table 4.

3.1 Threat Intelligence Gathering and Target Selection

LLMs process and synthesize intelligence by extracting data from diverse sources [184]. Then, LLM-based agents transform this data into actionable intelligence.

3.1.1 Cyber Threat Intelligence. The cyber-threat-intelligence (CTI) capabilities of LLM-based agents use a retrieval-reasoning-action framework with perceptual processing [68, 195]. Complementing extraction, Daneshvar *et al.* [49] introduce VulScribeR, a RAG-powered framework that mutates, injects, and extends code to generate realistic vulnerable samples, boosting deep-learning vulnerability-detector F1 scores by up to 69.9% at minimal cost. Moving from global to organization-specific intelligence, Mitra *et al.* [128] propose LocalIntel, which fuses public feeds with internal wikis and confidential reports; Qwen1.5-7B-Chat delivers 93% accurate contextualization across 58 zero-day triggers while slashing

Table 3. **Mapping of LLM-based agent capabilities to cyberattack categories.** *Legend:* **High** **Medium** **Low**

Cyberattack Type	Perception	Memory	Reasoning & Planning	Tool Invocation	Multi-agent Collaboration
Reconnaissance and Intelligence					
Threat-Intelligence Gathering	OSINT extraction, IOC mining, KG building	RAG-assisted CVE recall	Threat correlation and prioritisation	SIEM rule generation, API interfacing	Autonomous agent workflow
Exploitation and Payload Delivery					
Penetration Testing	Parsing scan/vuln outputs	Tracking enumeration progress	ReAct planning, attack-graph generation	Automated shell/Nmap/Metaspl calls	Role-decomposed collaboration
Vulnerability Detection	Semantic code/binary parsing	Knowledge-base integration	Cause localisation, patch suggestion	Selective tool orchestration	Cascaded single-agent detector
Malware Generation	Behaviour-to-code conversion	TTP/code pattern memory	Automated payload synthesis	Emit functional malware scripts/code	Autonomous agent swarms
One-/Zero-day Exploitation	Extract CVEs, logs, descriptions	Recall exploit modules/chains	CoT/Reflexive reasoning of paths	Dynamic exploit crafting and parameterisation	Shared roles for recon/exfiltration
Deception and Social Engineering					
Phishing & Social Engineering	Victim profiling from raw text	Contextual memory in dialogue	Psychologically tuned message crafting	Spear-phishing content generation & delivery	Individual agent-driven attack
Honeypot Deployment	Parse attacker input, emulate system response	Track session history, deception context	Adapt interaction strategy based on behaviour	Run realistic shell commands, mimic services	Multi-agent deception or response collaboration
Autonomous Challenge Solving					
Capture-the-Flag Challenges	Problem-statement parsing, flag-pattern recognition	State tracking for multi-step problems	CoT multi-hop reasoning, action planning	Basic decoding/scripting tools	ReAct & Plan single-agent template

analyst effort. Tseng *et al.* [184] push automation into the SOC by chaining GPT-4 tools that extract 2,300 validated IOCs, build relationship graphs, and autogenerate SIEM regexes with 97% accuracy, although post-processing mitigates occasional hallucinations. In the underground-economy domain, Clairoux *et al.* [45] harness GPT-3.5-turbo to summarize 700 cybercrime-forum threads and predict CTI variables with 96.2% overall accuracy, demonstrating LLM versatility with noisy multilingual text.

Benchmarks: Alam *et al.* [14] release CTIBench, a comprehensive APT and malware benchmark. Their evaluation shows GPT-4 leads overall performance while highlighting models' tendency to overestimate threats.

3.1.2 Penetration Testing. In LLM-based penetration-testing agents [104, 136, 171], dynamic reasoning lets agents adapt attack strategies based on discovered vulnerabilities. Initially, LLM-driven penetration testing that still keeps a human "red button" in the loop. Goyal *et al.* [76] first benchmark GPT-3.5-Turbo through GPT-4-Turbo inside pentest workflows, finding that the cheaper model is faster yet loses context in complex attacks. To impose discipline on that

Table 4. The list of LLM-based agent frameworks for cyberattacks. Attack-type abbreviations: CTI = Cyber Threat Intelligence; PT = Penetration Testing; VD = Vulnerability Detection; PSE = Phishing & Social Engineering; MG = Malware Generation; VE = Vulnerability Exploitation; HP = Honeypot Deployment; CTF = Capture the Flag.
Symbols: ✓ = Yes, × = No, △ = Partial, ○ = basic reasoning, ⊙ = advanced, ● = state-of-the-art chain-of-thought.

LLM-based Agents	Attack Type	Params	Context	Open	Multi	Reason	Tool use	Role
MAD-LLM [56]	CTI	varies	8 k	△	×	⊙	AutoGen debate	Purple
LLMCloudHunter [164]	CTI	GPT-4o-V	8 k	×	✓	⊙	Vision & rules	Blue
VulScribeR [49]	CTI	175 B & 7 B	8 k	△	×	○	RAG augmentation	Purple
Crimson [98]	CTI	70 B	16 k	✓	×	●	CVE to ATT&CK	Blue
PentestGPT [52]	PT	backend	16 k	✓	×	⊙	Metasploit CLI	Purple
RapidPen [132]	PT	GPT-4	32 k	×	×	●	RAG executor	Red
Breachseek [19]	PT	GPT-4	128 k	✓	×	⊙	LangGraph planner	Red
Hackphyr [159]	PT	7–13 B	4 k	✓	×	⊙	Internal cmds	Red
AttackLLM [6]	PT	GPT-4	8 k	△	×	⊙	Agent actions	Red
VulnBot [105]	PT	GPT-4o-mini	32 k	✓	×	⊙	Multi-agent	Red
AutoPT [197]	PT	GPT-4	32 k	×	×	⊙	FSM executor	Red
CIPHER [153]	PT	GPT-4	8 k	△	×	○	Function calls	Red
ARACNE [136]	PT	GPT-4	32 k	△	×	⊙	SSH tools	Red
PenHealNet [89]	PT	mixed	8 k	△	×	⊙	Remediation agents	Purple
PenHeal [90]	PT	mixed	8 k	△	×	○	Remediation chain	Purple
LProtector [173]	VD	GPT-4o	128 k	△	✓	●	RAG & CoT	Blue
EvilInstructCoder [87]	VD	7–16 B	4 k	✓	×	○	—	Purple
WitheredLeaf [43]	VD	mixed	8 k	△	×	⊙	Cascade detector	Blue
GRACE [122]	VD	GPT-4	8 k	✓	×	⊙	Graph-aug. prompts	Blue
PDBERT [142]	VD	110 M	512	✓	×	○	—	Blue
PhishAgent [39]	PSE	Otter-MM	4 k	✓	✓	⊙	Vision detector	Blue
ConvoSentinel [8]	PSE	GPT-4	8 k	△	×	⊙	Delegate agents	Blue
SE-OmniGuard [110]	PSE	GPT-4	8 k	△	×	⊙	Persona filter	Blue
WormGPT [70]	PSE	6 B	8 k	△	×	○	—	Red
SEAR [34]	PSE	GPT-4o	128 k	△	✓	⊙	AR interface	Red
AppPoet [218]	MG	GPT-4	8 k	△	×	○	—	Blue
GenTTP [216]	MG	mixed	8 k	✓	×	⊙	Agent parsing	Purple
RedCodeAgent [79]	MG	GPT-4o-mini	32 k	✓	×	⊙	Function calls	Red
SEVENLLM [96]	VE	13 B	8 k	✓	×	⊙	JSON tools	Blue
Net-GPT [151]	VE	hybrid	4 k	✓	×	○	MITM packet gen	Purple
RatGPT [28]	VE	ChatGPT	4 k	△	×	○	Bash shell	Red
AdbGPT [64]	VE	GPT-3.5/4	8 k	✓	×	⊙	ADB automation	Purple
Vul-RAG [57]	VE	GPT-4	32 k	△	×	⊙	RAG	Blue
CVE-LLM [73]	VE	7 B	8 k	✓	×	○	—	Blue
ShelLM [176]	VE	GPT-3.5/4	8 k	✓	×	○	—	Blue
CheatAgent [137]	VE	GPT-3.5/4	8 k	✓	×	⊙	Function calls	Red
ChatIoT [55]	VE	70 B	16 k	✓	×	⊙	RAG	Purple
hackingBuddyGPT [77]	VE	GPT-4	8 k	✓	×	○	Bug-bounty assist	Red
HackerGPT [186]	VE	13 B	4 k	△	×	○	OSINT tools	Red
HoneyLLM [60]	HP	mixed	128 k	×	✓	⊙	Function calls	Blue
LLMPot [187]	HP	4 B/L2/ByT5	8 k	✓	△	⊙	Honeypot sim	Blue
HackSynth [131]	CTF	GPT-4	8 k	△	×	⊙	Plan / summarise	Red
EnIGMA [1]	CTF	GPT-4o	128 k	✓	×	●	GDB / nc tools	Purple

reasoning gap, Wu *et al.* [197] frame each step as a Penetration-Testing State Machine, named AutoPT, which lifted task-completion rates over ReAct [203] while occasionally mis-generating shell commands. In parallel, Pratama *et al.*

[153] fine-tune CIPHER on write-ups for better exploit guidance. Al-Qurishi *et al.* [13] develop PenTest++, an automated framework requiring human oversight.

Happe *et al.* [83] first show GPT-3.5 can guide pentesting when paired with a vulnerable VM, though stability varies. Building on these insights, Deng *et al.* [52] develop PentestGPT, achieving 228.6% better task completion than GPT-3.5. The framework excels at tool usage and output interpretation but struggles with images, strategy selection, and knowledge accuracy. Its three self-interacting modules for penetration testing have gained wide recognition since being open-sourced. Pushing autonomy to enterprise scale, Happe *et al.* [84] fuse hackingBuddyGPT with PentestGPT to compromise an Active Directory lab without operator input, surpassing orchestrators such as MITRE Caldera. Nakatani *et al.* [132] develop RapidPen, a React-driven framework achieving shell access in 200-400s. Huang *et al.* [89] introduce PenHealNet, combining Pentest and Remediation agents to improve upon PentestGPT's capabilities.

Multi-agent penetration testing frameworks coordinate specialized roles for automated security assessments. PenHeal combines testing and remediation to improve coverage by 31% and reduce costs by 46% [90]. Breach-Seek implements a distributed architecture for autonomous scanning [19]. PENTEST-AI integrates MITRE ATT&CK with GPT-4 agents but requires reporting improvements [35]. Furthermore, VulnBot organizes reconnaissance, scanning, and exploitation agents via a penetration-task graph, achieving up to 69% task completion yet still struggles with non-text inputs [105].

Benchmarks: To benchmark the performance of LLM-based agents in automated penetration testing [192], Gioacchini *et al.* [74] introduce AUTOPENBENCH, a 33-task framework spanning access-control, web, network, and cryptography challenges. Complementing this closed-set study, Isozaki *et al.* [94] release an open benchmark driven by PentestGPT and show that LLMs still falter on end-to-end workflows, reinforcing the need for human oversight. Extending the evaluation landscape to CTF environments, Muzsai *et al.* [131] propose HackSynth, whose planner–summarizer architecture solves 41/120 PicoCTF tasks with GPT-4o.

3.1.3 Vulnerability Detection. LLM-based agents can detect vulnerabilities by integrating advanced language perception with structured reasoning and selective tool orchestration, enabling automated, high-fidelity triage across diverse codebases and binary artifacts [173]. Chen *et al.* in [43] propose WitheredLeaf, a cascaded detector that funnels alerts from lightweight language models to GPT-4; across 154 Python and C GitHub projects, it uncovers 123 previously unknown flaws, 45% exploitable, while GPT-4's 60% success on synthetic EIBs is bolstered by CodeBERT and Code Llama to sharpen recall and trim false positives. Building on this, Hossen *et al.* [87] present EvilInstructCoder, revealing that poisoning just 0.5% of instruction-tuning data for code LLMs yields 76–86% attack success, spotlighting urgent defence gaps as code LLMs permeate development pipelines. Complementing these insights, Akuthota *et al.* [10] report a 77% accuracy from GPT-3.5-Turbo on 2,740 snippets spanning SQLi, XSS, and command-injection.

Recent advancements in RAG and structure-aware LLM-based agents have demonstrated significant improvements in C/C++ and binary code vulnerability detection through enhanced accuracy, expanded training datasets, and optimized resource utilization. For instance, LProtector [173] demonstrates the effectiveness of integrating GPT-4o with RAG and CoT reasoning, achieving 89.68% accuracy and 33.49% F1 scores on 5,000 balanced Big-Vul samples, outperforming established tools while identifying limitations in plain text code processing. VulScribeR [49] presents an innovative approach to dataset enhancement, leveraging ChatGPT-3.5 and CodeQwen-1.5 with specialized prompting techniques to generate 1,000 vulnerability examples cost-effectively at US $1.88, resulting in F1 score improvements of up to 30.80% across multiple datasets, though effectiveness depends on prompt optimization. Additionally, GRACE [122] incorporates graph-based contextual demonstrations, demonstrating superior performance with a 28.65% F1 improvement across

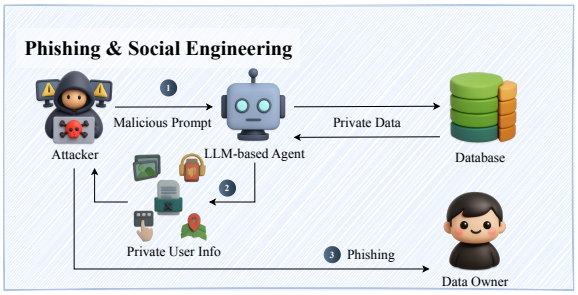

Fig. 4. LLM-based agents' cyberattack capabilities of phishing and social engineering.

comparable datasets, despite current C/C++ limitations. Furthermore, PDBERT by Panebianco *et al.* [142] reveals critical insights regarding model limitations.

3.1.4 Phishing and Social Engineering. As shown in Fig. 4, LLMs can craft convincing phishing emails, chats, and voice scripts, making social engineering harder to detect. Using victim-specific language and automated workflows, these agents transform manual campaigns into instant, personalized attacks at scale [71]. For instance, Alotaibi *et al.* [18] demonstrate that prompt-engineering can bypass safeguards to mass-produce phishing content cheaper than humans, while surveying countermeasures and deepfake risks. Furthermore, Begou *et al.* [29] show ChatGPT can deploy complete phishing kits in 10 minutes, noting token limits and withholding specifics. Roy *et al.* [163] analyze how attackers bypass ChatGPT, Claude, and Bard safeguards, proposing prompt-level detection. Finally, Chen *et al.* [42] introduce PEN using LLMs to synthesize novel phishing samples.

LLMs can automate and scale phishing creation, subsequent work pivots to testing the resilience of current defenses, probing new AI-mediated attack channels, and proposing multimodal countermeasures [4]. Figueiredo *et al.* [69] propose ViKing system that uses GPT and voice modules to persuade 52% of participants to divulge sensitive data, with 71.25% rating its replies effective. Cao *et al.* [39] design PhishAgent that achieves 94% detection accuracy while resisting brand-obfuscation attacks. Finally, Yang *et al.* [201] uncover fox8, a network of 1,140 ChatGPT-assisted Twitter bots whose interaction patterns defeat standard detectors, highlighting the need for models trained on real adversarial data.

In addition to phishing, LLM-enabled social engineering agents now focus on intent-driven conversational attacks based on psychological analysis and evaluation. Yu *et al.* [207] classify AI-driven social engineering through their taxonomy, reviewing 117 studies and developing a Markov process to measure penetration efficiency and costs.

Benchmarks: Evaluating these threats, Ai *et al.* [8] create SEConvo with 5,300 dialogues and ConvoSentinel pipeline, which increases F1 scores by 12% against LLM-generated attacks. Kumarage *et al.* [110] develop SE-VSim to simulate 1,350 persona-based conversations and SE-OmniGuard to improve detection by 8-15%.

3.2 Automated Weaponization

3.2.1 Malware Generation. As shown in Fig. 5, LLM-based agents in cybersecurity enable automated malware generation through code generation [86, 97]. Using natural language programming and script generation, agents convert behavioral descriptions to attack code, evade detection, and generate variant malware. This language-code fusion accelerates malware development while expanding attack potential with minimal human input. The authors in [70]

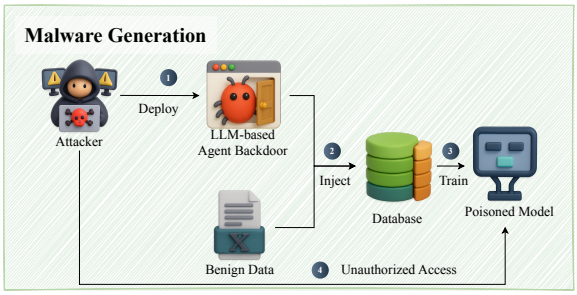

Fig. 5. LLM-based agents' cyberattack capabilities of malware generation.

present what they describe as the first peer-reviewed study of WormGPT, a black-hat LLM built on EleutherAI's GPT-J and trained on malware-related data, after an exhaustive literature search revealed no prior academic work on the subject. Charan *et al.* in [40] examine the malicious use of LLMs for generating cyberattack payloads, analyzing over 500,000 real-world malware samples from 2022 and systematically producing executable code for the top 10 MITRE Techniques. Their findings show that ChatGPT outperforms Bard in producing coherent, functional code and handling error resolution, thereby enabling the development of more sophisticated attack vectors. Transitioning from payload synthesis to behavioral analysis, Zhang *et al.* [216] introduce GENTTP for extracting TTPs from malware. The model was tested on labeled and real-world datasets. GPT-4 achieved 0.90 coverage and 0.99 sequence accuracy. GENTTP surpassed other LLMs in detecting behavioral patterns. Building on this foundation, Patsakis *et al.* [148] and Lin *et al.* [118] explore LLM performance in script-level deobfuscation, focusing on PowerShell samples from the Emotet malware campaign. Extending beyond individual samples.

Building upon prior concerns surrounding LLM-enabled cyber threats, Beckerich *et al.* in [28] examine LLMs as malware proxies. Their POC shows ChatGPT enabling covert command and control (C2) communication through plugin exploitation. Extending the discussion from offensive misuse to defensive innovation, Zhao *et al.* in [218] introduce AppPoet, a multi-view LLM-based Android malware detection system that leverages GPT-4 for generation and text-embedding-ada-002 for embedding tasks. AppPoet integrates static feature extraction with human-readable behavioral analysis and utilizes a DNN classifier to combine multi-view data. When tested on a dataset of 11,189 benign apps from AndroZoo and 12,128 malware samples verified via VirusTotal, the system achieved a detection accuracy of 97.15% and an F1 score of 97.21%.

Benchmarks: Transitioning to safe development environments, Guo *et al.* [78] created RedCode to test code agent safety. They evaluated multiple agents across thousands of tests in sandboxed environments. Their findings show GPT-4 produces more harmful code despite safeguards. Building on this, Guo *et al.* [79] developed RedCodeAgent, achieving 72.47% attack success rate, which highlights the need for better automated safety testing.

3.2.2 Vulnerability Exploitation: One-Day and Zero-Day Attacks. LLM-based agents with code analysis and reasoning abilities enable autonomous systems to detect and exploit software vulnerabilities dynamically. Through semantic analysis, exploit chain construction, and automated tool integration, these agents transform manual exploitation into rapid, adaptive workflows. Studies show their effectiveness across cloud, web, and mobile environments, demonstrating their ability to expand attack coverage with reduced expertise requirements. As shown in Fig. 6, Patil *et al.* [147] inaugurate this discourse on the defensive front by showing that LLM-powered anomaly detectors improve zero-day spotting in

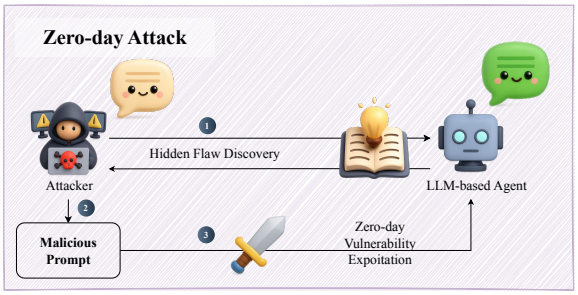

Fig. 6. LLM-based agents' cyberattack capabilities of Zero-day attacks.

cloud networks while proposing safeguards against hallucination and bias. Transitioning from defence to offence, Fang *et al.* [61] demonstrate that GPT-4 armed with publicly available CVE descriptions automatically reproduces 87% one-day exploits with performance dropping to 7% absent that auxiliary knowledge, thereby exposing both the promise and the limits of current models. Extending to the mobile domain, Feng *et al.* [64] present AdbGPT, which reproduces 81.3% of 88 Android bugs within 253.6 seconds and surpasses 90% accuracy in step-to-reproduce extraction through prompt engineering and CoT reasoning. Situating these advances within the broader tooling landscape, Ferrag *et al.* [66] critique the pattern dependence of traditional scanners and argue that deep-learning pipelines must reconcile formal-verification precision with real-time performance to scale beyond curated datasets. Building on LLM-assisted exploitation research, subsequent work focuses on vulnerability detection, progressing from code finetuning through knowledge retrieval to domain adaptation. Shestov *et al.* [174] finetune WizardCoder for Java vulnerability detection, framing the task as question-answering and mitigating a 20× class skew with curriculum learning, active sampling, focal loss, and sample weighting. Evaluated on 624 vulnerabilities from 205 OSS projects, their model surpasses CodeBERT-like baselines in both ROC-AUC and F1 on balanced and imbalanced test sets, albeit with noted sensitivity to noisy labels.

Benchmarks: Du *et al.* [57] introduce Vul-RAG, which constructs a knowledge base from 2,174 CVEs and matches candidate functions by semantic retrieval before prompting GPT-4 to reason about causes and fixes. On the new PairVul benchmark, which consists of 4,667 function pairs, Vul-RAG lifts overall accuracy by 12.96% and pairwise accuracy by 110% over prior art while acknowledging leakage risks and its Linux-kernel focus. Ghosh *et al.* [73] target the medical-device supply chain with CVE-LLM, combining domain-adaptive pre-training on regulatory notifications with a human-in-the-loop workflow. Over a two-month pilot, the system materially reduces analyst effort, though long-sequence handling and spurious text in Llama-2 variants remain open challenges.

3.2.3 Honeypot Deployment. Honeypots are controlled environments that mimic vulnerable systems to study adversarial behavior safely. LLM-based agents are deceptive frameworks that generate realistic system responses to attacker inputs. Through contextual analysis and response generation, these agents can simulate authentic behaviors from Linux shells to industrial protocols. Reti *et al.* [158] launch this trajectory by testing 210 prompt templates across GPT-3.5, GPT-4, Llama-2, and Gemini on 1.6 M leaked ClixSense credentials, showing that LLMs can craft honeywords and robots.txt tokens with a 56% indistinguishability rate. Building on automation, Fan *et al.* [60] introduce HoneyLLM, a Go-based medium-interaction honeypot whose GPT-4-Turbo, Claude 3 Opus, and Gemini 1.5 Pro back-ends successfully execute 21 of 25 Linux commands and log both network and system-level activity, as quantified by their ShellEval metric, while highlighting open challenges in latency and jailbreak resistance. Complementing proprietary models, Otal *et al.*

Table 5. Benchmarks for LLM-based cyberattack agents: Main advantages & limitations

Benchmark	Task Focus	Main Advantages	Key Limitations
Safety / Red-Teaming			
AgentHarm [22]	Harmful-instruction	Fully automated evaluation	Text-only prompts
HarmBench [126]	Unsafe behavior robustness	Per-class breakdowns	Focuses only on single-turn prompts
R-Judge [210]	Safety-risk awareness	Multi-step safety scoring	Small scale
Knowledge Q&A / Retrieval			
CS-Eval [209]	Cybersecurity Q&A	Separates knowledge vs reasoning	No interaction or action execution
SecQA [121]	Multiple-choice queries	Simple and fast diagnostic	Small MCQ set; lacks deep reasoning
CmdCaliper [92]	Command safety	Retrieval-based	Synthetic queries
PRIMUS [208]	Corpus assessment	Large-scale domain corpus	No downstream task linkage
CTIBench [14]	Threat intel from CTI reports	APT/malware alignment tasks	Expensive labeling
Pen-Testing / Exploitation			
CyberSecEval 1 [33]	ATT&CK tactics	Safe sandbox testing	No end-to-end chaining
CyberSecEval 2 [32]	Prompt injection	Targets specific exploit types	Limited kill-chain scope; static
AutoPT-Sim [192]	In simulated networks	FSM planning improves ASR	Shell error rates persist
AutoPenBench [74]	Containerized pen-test tasks	Diverse exploit goals	Requires expert setup
Breach-Seek [19]	Multi-agent coordination	Demonstrates role-based planning	Evaluation unclear
Vulnerability & Code Analysis			
Vul-RAG [57]	Function-level matching	Boosts patch accuracy	Limited to known CVEs
PairVul [57]	Code pair vulnerability	Strong pairwise matching	Potential overfitting; dataset-specific
RedCodeAgent [79]	Unsafe code generation	72% attack success	Shell-centric; lacks broader context
Social Engineering / Phishing			
PEN [42]	Phishing mail generation	Human realism evaluations	Only text; small scale
SE-OmniGuard [110]	Multi-turn SE detection	Persona-aware detection	Early-stage sim; unreleased dataset
Honeypot / Shell Evaluation			
ShellEval [60]	Shell realism and deception	Command match rate	Linux-only; limited in size
LLMPot [187]	ICS honeypot interaction	Byte-level metrics on protocol	Limited function length
Capture-the-Flag (CTF)			
HackSynth [131]	Autonomous CTF solving	Solves 34% of tasks	Lower performance on complex logic
InterCode-CTF [200]	Interactive CTF coding tasks	ReAct&Plan boosts solve rate to 95%	Gaps in binary/reversing domains

[139] fine-tune Llama-3-8B on 617 attacker commands, achieving cosine and Jaro-Winkler similarities of 0.695 and 0.599 to ground-truth outputs, yet note fingerprinting vulnerabilities and the need for adaptive rate-limiting. Human-subject validation follows in Sladič *et al.* [176], where shelLM (GPT-3.5-turbo-16k) deceives participants in 90% of 226 SSH-shell interactions, despite occasional hallucinations and response lag. Domain-specific generalisation is tackled by Vasilatos *et al.* [187], whose LLMPot emulates industrial-control protocols via GPT-4, Llama, and ByT5, attaining 93% Response-Validity Accuracy and 88% Byte-to-byte Comparison Accuracy, though unbounded-length functions remain problematic. Finally, Volkov *et al.* [188] demonstrate real-world viability over a three-month public deployment. Their LLM-augmented SSH honeypot records 8.13 M interaction attempts, which identifies eight autonomous prompt-injection attacks, and observes that LLM-based agents reply within a 1.7s median, which is far faster than humans.

Table 6. Comparison of representative LLM-Enabled cyberattack methods on static-infrastructure networks.

Ref.	Agent Architecture	Network Type	Attack Goal	Blue-team Impact
[175]	ReAct planner & multi-tool orchestration	6G Core & RAN	One-shot break, long-term persistence	Defences largely unaffected (legacy rules bypassed)
[84]	Role-split multi-agent (scan/exploit/privilege)	Enterprise Networks	Privilege escalation and lateral movement	Existing identity and segmentation measures can be bypassed
[147]	Log RAG & anomaly reasoning loops	Data Center Networks	Zero-day detection or abuse of control plane APIs	Alert fatigue decreased; detection improved
[180]	Tokenized flow-based classification with BERT	Software Define Networking	Flow rule manipulation, stealth DDoS	Signature-based IDSs evaded; new attack paths open
[93, 211]	Prompt completion & ICS payload synthesis	Smart Grid	False-data injection, phishing, system spoofing	Real-time model outputs bypass legacy sensors
[9]	Code generation & classical/quantum planning	Quantum Networks	Side-channel attacks on QKD, device layer threats	Control-plane defenses need upgrade

3.2.4 Capture the Flag Challenges. The results that LLM-based agents are being tested on Capture-the-Flag (CTF) challenges mean we can observe their problem-solving strengths and weaknesses. Early evaluations of LLMs for cybersecurity education were conducted by Tann *et al.* in [182]. Comparing ChatGPT, Google Bard, and Microsoft Bing on seven CTF exercises and three tiers of Cisco-level factual questions, they reported that ChatGPT solved 6 / 7 challenges and reached 82% accuracy on knowledge items. These results confirmed the pedagogical value of LLMs while exposing two structural weaknesses. Building on these insights, Turtayev *et al.* in [185] reframe CTF solving as an agentic process. Their ReAct&Plan template steers GPT-4o through up to 30 reasoning–action turns, yet most tasks were solved in only 1–2 turns. The approach pushed success on InterCode-CTF to 95%, eclipsing the prior 29% and 72% baselines. **Benchmarks:** To quantify those weaknesses more systematically, Yang *et al.* in [200] created InterCode-CTF, a 100-task PicoCTF-based benchmark. GPT-4 solved 40% of tasks, struggling with complex reverse-engineering and binary-exploitation. Tests with GPT-3.5, Vicuna-13B, and StarChat-16B showed similar limitations. Abramovich *et al.* in [1] developed EnIGMA, enhancing the SWE-agent with new tools and demonstrations. EnIGMA outperforms prior benchmarks but faces challenges in web-exploitation and data leakage protection.

Finally, the overview of cyberattack-oriented benchmarks for LLM-based agents is shown in Table 5.

3.3 Lessons Learned for Blue Teams

(1) **Frequent Defense Upgrade:** Defensive teams should implement regular updates to security controls and threat intelligence feeds, fix exposed ports, and misconfigurations. Multiple vulnerabilities signal system weakness, especially with automated scanning. AI malware shows distinct markers like machine-written code and unusual API calls. These identifiers help trace origins and assess threat levels.

(2) **Active Honeypot Deployment for LLM-based Agents:** Blue teams can enhance their defensive capabilities by deploying LLM-augmented honeypots to engage and monitor attackers at scale. These systems serve as valuable intelligence-gathering tools, providing data that helps update detection signatures and defensive playbooks with emerging attack patterns. To maintain effectiveness, teams must focus on keeping honeypots realistic through regular updates to their conversational models and system responses, preventing sophisticated attackers from detecting and evading these defensive measures.

4 Cyberattack Capabilities of LLMs-based Agents on Static-Infrastructure Networks

Static-infrastructure networks are systems with fixed topology and node placement, maintaining stable traffic patterns. LLM-based agents pose cybersecurity threats by automating attacks on static infrastructure networks, including 6G, enterprise, data center, SDN, smart grid, and quantum networks. These agents focus on “one-shot-break, long-term-stay" attacks for persistent attack installation in critical infrastructure. Table 6 summarizes representative LLM-enabled cyberattack methods across static-infrastructure networks, highlighting their architectures, system targets, attack goals, and defensive implications.

4.1 6G Core and Radio Access Networks

LLM-based agents can translate high-level intents into low-level network commands, potentially abusing 6G programmability to alter network behavior maliciously. Beginning with network management, Mani *et al.* [125] establish that state-of-the-art LLMs can translate natural-language directives into valid router, firewall, and orchestration code. Because these same routines can inject flows or deactivate security rules, their work frames LLM-mediated configuration as a dual-use capability. Shifting focus from functionality to vulnerability exposure, Nguyen *et al.* [135] enumerate attack surfaces unique to 6G and argue that LLM autonomy enables real-time, cross-domain exploit generation. Providing quantitative evidence, Singer *et al.* [175] demonstrate with the Incalmo abstraction framework that an LLM-based agent compromised nine of ten multi-host mobile-core testbeds (25–50 hosts each) by chaining reconnaissance, signaling-protocol exploits, and lateral movement. Finally, Andreoni *et al.* [21] and Yigit *et al.* [206] show that the “cost-collapse" of generative AI simultaneously lowers the barrier to sophisticated attacks and overwhelms legacy detection. At the edge layer of the 6G RAN, Rondanini *et al.* [162] propose an LLM-centric malware-detection architecture for resource-constrained edge nodes. The best GPT variant achieves 97% detection accuracy without exporting raw traffic centrally. Zhang *et al.* [213] show in-context learning matches fine-tuning in wireless-network IDSs, with GPT-4 reaching 95% accuracy across DDoS classes. Legashev *et al.* [114] develop a hybrid LLM-LSTM system for wireless backbones, where Gemma-7B achieves 0.89 F1 in malicious classification with 3% error from poisoning.

4.2 Enterprise Networks

In enterprise networks, valuable assets such as public-facing servers and critical internal services are frequent targets of distributed reconnaissance scans, lateral movement, privilege escalation, and distributed denial-of-service (DDoS) attacks [124]. Attackers typically exploit exposed services, misconfigured devices like internal DNS/NTP servers, and unmanaged mobile devices, which can either be direct victims or leveraged as attack amplifiers. The authors in [84] investigate whether LLMs can perform autonomous penetration testing in enterprise networks through a novel prototype designed for Active Directory environments. They develop and evaluate this prototype using two specific models of OpenAI in a realistic simulation environment, demonstrating that LLMs can effectively conduct Assumed Breach simulations by identifying access points and executing lateral movement. In enterprise networks where LLM-based agents can privilege escalation and lateral movement, blue teams should adopt a zero-trust mindset.

4.3 Data Center Networks

Data center networks usually rely on APIs and orchestration. LLM-based agents could exploit these control plane APIs if credentials or misconfigurations are found. For data center networks, Patil *et al.* [147] introduce an LLM system designed to continuously analyze cloud infrastructure logs and telemetry data for potential *zero-day* attack patterns, demonstrating

superior detection capabilities compared to conventional rule-based approaches across multiple historical breach scenarios. Blue teams should strictly enforce least privilege on API keys, rotate them frequently, and monitor API usage patterns.

4.4 Software-Defined Networking

The SDN controller is a high-value target, and LLM-based agents might launch sophisticated DDoS or traffic-manipulation attacks that standard threshold-based systems are not able to catch [7]. Foundational research by AlEroud *et al.* explores the implementation of inference-based intrusion detection systems for software-defined networking controllers [16]. LLM-based agents could reverse-engineer defenses to reprogram flow tables, enabling evasion and link-flooding attacks. Specht *et al.* show SDN architectures can mitigate industrial malware through network path reconfiguration [178]. This research reveals malicious LLMs could exploit southbound API interfaces for worm propagation. The authors in [180] use BERT-base-uncased to transform network flows into natural language for attack detection in SDN. Using the InSDN dataset with normal and attack flows, the system detects DDoS, DOS, Probe, U2R, BFA, and Web attacks through BERT tokenization and Random Forest Classification. The model achieves 99.96% accuracy with 0.9995 precision and recall scores for known and unseen attacks. Defending SDN infrastructures against LLM threats requires understanding their capabilities in reasoning, evasion, flow manipulation, and network telemetry perception. Traditional detection mechanisms risk obsolescence against these autonomous adversaries.

4.5 Smart Grids

Smart grids could face multi-vector attacks orchestrated by AI. In smart grid operations, LLM-based agents might attempt false data injection to mislead grid control systems [117, 138]. Modern simulation platforms such as *GridAttackSim* [112] and *GridAttackAnalyzer* [113] enable researchers to model and evaluate these attack scenarios in controlled environments. The emergence of LLM-based agents has dramatically accelerated this process by automatically generating sophisticated attack graphs for these testbed environments, reducing scenario development time from hours to mere seconds. The reinforcement learning-based detection system developed by Kurt *et al.* [111] exemplifies the evolving complexity of the adversarial landscape, particularly well-suited to advanced generative AI systems. Zaboli *et al.* [211] provide detailed documentation of ChatGPT's capability to generate convincing phishing campaigns using sector-specific terminology and craft targeted Modbus/TCP attack payloads. Ibrahim *et al.* in [93] conduct a comprehensive examination of large model applications in grid cyber-physical systems, highlighting particular concerns regarding prompt-injection attacks targeting control room assistant systems. Li *et al.* [116] present a systematic analysis of LLM-based risks across power generation, transmission infrastructure, and distributed energy resource orchestration systems, while Zhang *et al.* in [217] reveal critical vulnerabilities wherein LLM-generated code can compromise anomaly detection systems within IoT-enabled electrical substations. To defend against emerging threats, monitoring and limiting LLM capabilities in tool orchestration, prompt interpretation, code generation, and adversarial reasoning is crucial. Implementing model alignment, sandboxed execution, and anomaly detection can help prevent LLM-driven cyberattacks in smart grids.

4.6 Quantum Networks

Quantum communications might be theoretically secure in transmission, but the supporting classical infrastructure is still vulnerable to LLM-based agents. By combining pattern-completion, code-generation, and planning skills, LLMs can (i) automate the discovery of implementation-side channels in QKD devices, (ii) craft novel attack graphs that blend classical and quantum layers, and (iii) orchestrate large-scale post-quantum reconnaissance at machine speed. Ajimon and Kumar present the first systematic blueprint in which an LLM is coupled with quantum-protocol libraries to generate

Table 7. Comparison of representative LLM-based cyber-attack methods in mobile-infrastructure networks.

Ref.	Agent Framework / Example	Network Type	Primary Attack Vector
[6, 54, 55, 67, 169]	AttackLLM multi-agent pentester; LLMPot industrial honeypot; ChatIoT on-device assistant	Constrained edge / IIoT gateways	Automated scanning, firmware takeover, process hijack
[5, 85]	PLLM-CS telemetry analyser; LEO-SDN LLM-aided routing monitor	LEO constellation & ground segment	Telemetry spoofing, routing manipulation
[3, 12, 129]	Generative-replay IDS; compact-Transformer router monitor	Dynamic MANET / VANET clusters	Sybil node injection, route disruption
[30, 156, 168, 179, 186]	GenAI CAN-log anomaly detector; HackerGPT for automated exploitation; fine-tuned GPTs for CAN fuzzing; polymorphic malware generators bypassing rule-based gateways	6G-V2X communication links; in-vehicle CAN buses; ADAS sensors (e.g., LiDAR, GPS)	CAN message fuzzing to disable controls; sensor spoofing (e.g., fake GPS or LiDAR input to trigger emergency braking); SYN flood attacks
[106, 151, 166]	Net-GPT MITM for forged C2; Bayesian/LSTM hybrid IDS	UAV C2 links	Command hijack, GPS spoof, jamming
[2, 20, 99]	GPT-augmented anomaly IDS; ChatGPT-based toolkits	Acoustic & optical UWNs	Adaptive DoS floods, topology inference

proof-of-concept exploits, e.g., photon-number-splitting or detector-blinding scripts, against BB84 and decoy-state systems in real time [9]. In the future, attacks might target quantum repeaters, entanglement distribution systems, or even quantum routers, as full quantum networks develop.

4.7 Lessons Learned for Blue Teams

(1) **Use AI to Counter AI Threats:** Deploy LLM-based monitoring systems to detect and respond to attacks from LLM-based agents. This is particularly important for complex environments like 6G networks, where defensive LLMs can identify subtle malicious patterns that humans might miss in regular operations.

(2) **Implement Zero Trust Architecture:** In environments where LLM-based agents can automate reconnaissance and lateral movement, blue teams need to adopt zero-trust approaches that can continuously verify all users and actions, implement strict network segmentation, and never assume internal traffic is automatically trustworthy.

5 Cyberattack Capabilities of LLMs-based Agents on Mobile Infrastructure Networks

To systematically examine the threat landscape in mobile-infrastructure networks, we categorize representative scenarios according to their underlying network architectures, mobility patterns, and security challenges. In mobile infrastructure networks, LLM-based agents succeed by continually re-planning in response to wireless volatility and connectivity changes. Through tool-chaining, an agent processes telemetry, GNSS, spectrum, and LiDAR data to compose protocol-aware payloads that adjust channels in real time. This capability enables GNSS spoofing, MitM, and DDoS attacks, reducing time-to-impact from hours to milliseconds. We summarize LLM-based cyberattack capabilities across six mobile infrastructure network categories as shown in Table 7.

5.1 Internet of Things

The Internet of Things (IoT) often has constrained devices, and LLM-based agents might seek out weak links like unpatched IoT firmware or default credentials to take over devices in the IoT Supply Chain [54, 55, 169]. Ferrag *et al.* in [67] demonstrate that LLMs integrated with RAG pipelines effectively process heterogeneous telemetry and derive

threat indicators autonomously, reducing reconnaissance costs for potential attackers. The AttackLLM [6] implements an LLM-based multi-agent system for industrial attacks, outperforming human experts in water-treatment plant testing.

In vulnerability discovery, LLMs demonstrate significant capabilities. Binhulayyil *et al.* successfully fine-tune a distilled model using CVE descriptions, achieving state-of-the-art F_1 scores in identifying buffer-overflow and injection vulnerabilities within embedded firmware [36]. On the defensive front, Vasilatos *et al.* present LLMPOT, an innovative LLM-controlled honeypot that implements industrial protocols and simulates physical processes, effectively attracting autonomous adversaries while identifying their LLM signatures [187]. The integration of conversational agents within constrained devices represents an emerging trend. ChatIoT enables the transformation of open-weight models into on-device security assistants capable of managing scanning, patch generation, and real-time alert triage [55]. Additionally, BARTPREDICT combines a BART-based predictor with time-series embeddings to anticipate zero-day exploits 24 hours in advance across IIoT power grids [54]. These developments indicate a dual-use trajectory where enhanced generative capabilities simultaneously facilitate both system protection and exploitation. The authors in [53] propose an IoT cybersecurity framework combining LLMs with LSTM networks. The authors in [169] investigate security challenges between IoT devices and LLMs, focusing on adversarial attacks against Llama-2-7b. Their experiments achieve 76% ASR, bypassing alignment measures through prompt injection and gradient-guided search methods.

5.2 Satellite Networks

LLM-based agents could attempt to spoof or manipulate the unencrypted parts of satellite communications. Hassanin *et al.* have developed PLLM-CS, a domain-specific LLM that analyzes satellite telemetry and identifies kinetic-level anomalies in Low-Earth-Orbit constellations [85]. Agnew *et al.* demonstrate that integrating an LLM with a software-defined network controller enables preemptive detection of zero-day routing attacks in LEO mega-constellations through network metric prediction, achieving a 42% reduction in mean detection time [5]. While these implementations position LLMs as defensive tools, they also reveal the potential for adapting these capabilities for satellite-borne intrusions.

5.3 Mobile Ad-Hoc Networks

In Mobile Ad-Hoc Networks (MANETs), where there is no fixed infrastructure, a common threat is Sybil attacks or rogue nodes [129]. LLM-based agents can rapidly create or control multiple nodes to disrupt routing or eavesdrop. Mohandas *et al.* implement a compact transformer for routing anomaly classification in vehicular MANETs, demonstrating superior performance of LLM embeddings over traditional features in high-mobility scenarios [129]. Al-Rubaye and Turkben implement generative replay techniques to maintain lightweight LLM detection accuracy despite concept drift, advancing *continual* adversarial adaptation [12]. Notably, Addula *et al.* present a generative AI-enhanced IDS combining an LLM planner with reinforcement learning, achieving 97% neutralization of multi-vector attacks while generating adversarial traffic for network stress testing [3]. These developments indicate significant potential for autonomous red-teaming.

5.4 Vehicular Networks

Vehicular networks present unique challenges, combining critical latency requirements with extensive and heterogeneous attack surfaces. Therefore, LLM-based agents might exploit these channels for SYN flood DDoS or spoofing attacks. Sun *et al.* in [179] demonstrate that GenAI-driven detection systems effectively analyze vehicular CAN traffic and edge-compute logs, achieving 4.3 percentage points higher recall than CNN baselines in identifying SYN-flood and GPS-spoofing attacks. Begum *et al.* demonstrate LLM capabilities in creating sensor-spoofing payloads that effectively compromise LiDAR-based ADAS, achieving 82% success in triggering emergency braking within a 6G-V2X testbed [30].

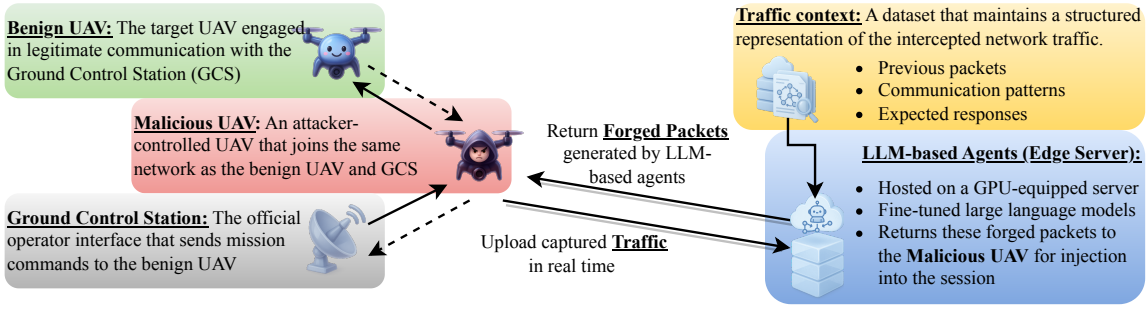

Fig. 7. LLM-based agents for main-in-the-middle attacks with UAV command-and-control.

Shafique *et al.* and Haddaji *et al.* analyze ML countermeasures, noting that prompt-injected LLMs generate polymorphic malware at rates exceeding rule-based gateway blacklisting capabilities [81, 168]. Rajapaksha's analysis of in-vehicle IDSs highlights risks from fine-tuned GPT agents in automated CAN fuzzing [156]. Aldhyani provides further evidence of deep-learning attack effectiveness against autonomous-vehicle perception systems [15]. These developments exemplify the ongoing competition between LLM-powered offensive and defensive capabilities. The authors in [186] present a comprehensive study on using LLMs for automotive cybersecurity research, developing a customized model called HackerGPT that generates exploitation scripts targeting vehicle systems.

5.5 UAV Networks

As shown in Fig. 7, UAV networks face cyber and kinetic risks through LLM-driven man-in-the-middle attacks. A malicious UAV inserts itself between a ground-control station and a benign UAV to capture TCP packets. An edge server stores traffic and uses LLM agents to predict legitimate packet fields [151].The edge server returns these forged packet templates to the malicious UAV, which then injects them back toward either the GCS or the benign UAV while optionally suppressing real packets. By repeating this capture-predict-inject loop in real time, the attacker can seamlessly impersonate either party, modify commands, and exfiltrate data without disrupting the appearance of normal communications. Common cyberattacks in UAV networks include GPS spoofing, C2 hijacking, jamming of communication links, and sensor data manipulation. Sedjelmaci *et al.* addresses routing misbehavior through Bayesian learning [166]. Current capabilities now extend to AI-automated spoofing, hijacking, and jamming tactics, as systematically surveyed by Kong [106]. LLM-based agents significantly amplify these threats by autonomously generating attack scripts [165]. Recent studies further highlight dual-use risks, as Dahiya*et al.* in [48] and Garg *et al.* in [46] demonstrate LLM capabilities in generating precise flight control modification scripts.

5.6 Underwater Networks

Underwater networks face unique challenges with bandwidth and latency constraints that were once thought to provide security benefits. These networks are actually susceptible to various security threats, including DoS attacks, spoofing, jamming, and routing attacks. LLM-based agents can now autonomously exploit these vulnerabilities through sophisticated techniques like adaptive DoS floods and automated topology inference. Altameemi *et al.* in [20] demonstrate enhanced anomaly detection capabilities through an SVM-RNN architecture augmented with GPT-generated features, achieving 96.4% accuracy in challenging channel conditions [20]. They identify denial-of-service vulnerabilities susceptible to LLM

Table 8. Representative LLM-based agent cyberttacks on infrastructure-free networks.

Ref.	Agent Architecture	Network Type	Attack Goal	Blue-team Impact
[50, 100, 183]	Multi-agent CoT & ReAct planner	Social Networks	Disrupt decision-making via misinformation flooding	Trust scoring, identity verification, and anomaly detection required
[119, 152, 181]	Prompt-driven traffic shaping with adaptive evasion	Content Delivery Networks	Saturate edge caches and degrade cache-hit ratio	Real-time provenance validation and adaptive rate-limiting needed
[17, 101, 199]	Code-aware retrieval & static analysis loops	Blockchain	Inject malicious smart contracts and poison consensus models	Fine-grained auditing, anomaly scoring, and peer reputation
[26, 109, 220]	KG memory & reflexive telemetry generation	Digital Twin	Inject deceptive sensor data and modify PLC state safely	Requires runtime certification and reasoning-path explainability
[34, 82, 205]	Multimodal RAG & ReInteract dialogue engine	Immersive XR/VR	Personalized social engineering through affect-aware overlays	Adaptive behavior detection and multimodal trust feedback needed
[50, 100, 146, 191]	Swarm RL with self-reflective memory	Agent Networks	Spread prompt-level misinformation and reduce task success	Memory isolation, prompt sanitization, and agent provenance tracking

exploitation, particularly regarding propagation delays and authentication weaknesses. Jocil *et al.* in [99] demonstrate ChatGPT applications in security toolkit development, while Adam *et al.* in [2] address dataset limitations and advocate for generative model applications in cryptographic testing.

5.7 Lessons Learned for Blue Teams

(1) **Edge-native Security:** For IoT environments, security controls should be pushed to edge devices like gateways and MEC servers. This includes implementing anomaly detection systems for LLM-based cyberattack agents at network entry points to catch coordinated attacks from LLM-orchestrated threats.

(2) **Multi-Layer Defense Strategy:** Mobile networks need multiple layers of protection to handle cyber threats from LLM-based agents. For example, in MANETs, this means combining radio monitoring, packet inspection, and host-based protection to quickly catch evolving attack tactics. Similarly, in vehicle networks, critical systems should be segregated with rigorous security checks between layers.

6 Cyberattack Capabilities of LLMs-based Agents on Infrastructure-free Networks

Table 8 outlines representative LLM-agent attack strategies across infrastructure-free networks, highlighting their architectures, network targets, attack goals, and implications for blue-team defense.

6.1 Social Networks

In social networks, LLM-based agents can create and manage fake personas at scale, which can flood social platforms with propaganda, phishing, or manipulative content [201]. For instance, CheatAgent shows that by impersonating recommender-system users, an LLM can steer ranking outcomes and exfiltrate private preference data without tripping anomaly detectors [137]. Earlier work on social-network honeypots demonstrated large-scale, automated creation and curation of fake identities to lure threat actors [143]. When combined with generative text models, such bots now produce

spear-phishing content that is statistically indistinguishable from human prose [63]. This may include analyzing behavior over time for human-like inconsistencies, using graph analysis to spot botnets.

6.2 Content-Delivery Networks

Content-delivery networks (CDNs) and information-centric overlays are vulnerable to several types of attacks [133], including cache saturation (Partition DoS), cache-miss amplification, content poisoning, and forwarding loop creation. Since LLM-based agents can generate large volumes of fake or poisoned content to store in caches, implement content verification where possible. Takashima *et al.* in [181] show that LLM-based agents coordinating many low-rate clients can bypass traditional volumetric, DoS thresholds and still saturate edge caches (partition DoS). Liu *et al.* in [119] highlight how intelligent request shaping maximises cache-miss penalties, pushing excessive origin traffic. Models for availability assessment[152] predict that a mere 3-5% decrease in cache-hit ratio can trigger SLA violations network-wide. When detecting the activities of LLM-based agents, CDNs can activate defenses such as serving stale content to suspected nodes, challenging them with CAPTCHAs, or temporarily isolating those requests.

6.3 Blockchain Networks

LLMs can rapidly identify and exploit vulnerabilities in smart contracts. Xiao *et al.* in [199] demonstrate an autonomous agent that locates re-entrancy and integer-overflow patterns, then patches malicious logic stubs into otherwise legitimate Solidity code, producing “smart-contract malware" with nearly zero human effort. A complementary survey work [17] catalogues GPT-powered phishing kits that fabricate token-airdrop sites and wallet-connect dialogs en masse. To compound the risk, collaborative-learning approaches for blockchain anomaly detection [101] make themselves be poisoned through subtle gradient perturbations introduced by a malicious LLM peer, causing selective blindness to the attacker’s transactions. These observations suggest that defending against LLM-based threats requires not only traditional vulnerability patching but also a deep understanding of agents’ capabilities in reasoning, tool orchestration, and stealthy adaptation.

6.4 Digital Twin Networks

Digital Twins rely on accurate data mirroring physical systems. Therefore, LLM-based cyberattack agents can inject deceptive telemetry or alter the twin’s state to mislead operators. Zheng *et al.* in [220] highlight how injecting deceptive telemetry via an LLM-based agent can mislead predictive-maintenance models, triggering premature or unsafe actuator commands. High-fidelity industrial twins are equally vulnerable: Balta *et al.* in [26] report that a twin-resident agent, when compromised, manipulated PLC set-points while maintaining plausible sensor traces. Aviation studies confirm that prompt-level attacks on twin-embedded copilots bypass traditional air-gap assumptions [109]. Krishnaveni *et al.* in [108] propose an intelligent defense framework that deploys counter-agent honeypots and trust scoring, but stresses the need for runtime certification of LLM reasoning paths.

6.5 Immersive Networks

Augmented/virtual reality (AR/VR) platforms present new attack vectors, such as malicious 3D content or overlay attacks [172]. In particular, LLM-based agents amplify these risks by autonomously generating dynamic, personalized attacks. Happa *et al.* [82] are the first to map extended reality (XR)-specific threats; recent work shows LLM-driven avatars dynamically adapt dialogue tone and visual cues to victims’ affective states [205]. Kilger *et al.* in [102] demonstrate detection of camera spoofing in Mixed Reality, yet admit failure against sophisticated, AI-generated overlays. Malicious

VR cues can mislead disabled users into hazardous movements [204]. The authors in [34] systematically investigate how multimodal LLMs paired with AR devices can be weaponized for next-generation social engineering, introducing the SEAR framework. The SEAR framework pairs multimodal LLMs with AR devices for social engineering by fusing visual and audio context, retrieving the target's digital footprint, and driving an agent through conversational stages.

6.6 Autonomous Agent Networks

Finally, attacks in autonomous agent networks include knowledge poisoning, prompt injection, backdoored system prompts, adaptive jailbreaks, and misinformation flooding. LLM agents execute attacks by crafting malicious prompts, corrupting memory, and amplifying errors through collaboration. Debar *et al.* in [50] outline threats when nodes can explain, plan, and act. Tete *et al.* in [183] provide a taxonomy for agent applications, focusing on backdoored prompts. Ju *et al.* in [100] show misinformation can flood multi-agent communities within minutes, reducing task success by 42%. Pasquini *et al.* in [146] reveal benign prompt-injection can defend against LLM hacking, while Wang *et al.* in [191] use reinforcement learning for adaptive jailbreaks. Agent-native networks are both attacker and defender domains, requiring formal verification and memory isolation. Countering these threats requires hardening reasoning integrity, controlling memory updates, and ensuring prompt sanitization.

6.7 Lessons Learned for Blue Teams

(1) **Trust and Reputation Mechanisms:** In infrastructure-free environments, LLM-based agents can create fake identities to conduct Sybil attacks and manipulate consensus. Blue teams must implement trust mechanisms like cryptographic attestations and behavioral scoring to ensure network accountability.

(2) **Resilience Through Redundancy and Decentralized Recovery:** LLM-based agents can target weak points in peer-to-peer networks to disrupt communication. Blue teams should design networks with redundancy in routing, storage, and decisions, and incorporate decentralized recovery protocols can help maintain function under compromise.

7 Future Research Directions

1) Governance/Guardrails for LLM-based Agents: Developing effective governance for LLM-based agents is critical. Unlike traditional tools, these agents can reason and escalate attacks independently. To mitigate risks, agent architectures must embed safety constraints. Research should implement ethical enforcement, compliance checking, and intervention mechanisms. Standardized audit frameworks would ensure transparency and accountability. International policies must regulate agents while preserving innovation.

2) Human-in-the-Loop Alignment for LLM-based Cyberattack Agents: As LLM-based agents acquire increasing autonomy, integrating human oversight becomes a fundamental challenge [140]. Systems should ensure human review at critical decision points during high-risk operations. Research must balance autonomy and human intervention while maintaining effectiveness. Dynamic human-in-the-loop systems and reinforcement learning from feedback can support this goal. Agents should seek human guidance when encountering ethical ambiguities, creating a symbiotic relationship between human expertise and machine operation.

3) Sustainable Red-teaming: Red-teaming uses simulated adversaries to test vulnerabilities while accounting for environmental impact [65]. Research should develop efficient methodologies that minimize energy use while maintaining vulnerability coverage. Techniques like scenario sampling, model distillation, and RL-based exploration can improve resource efficiency. Sustainable red-teaming practices will enhance both AI safety and environmental responsibility.

4) Privacy-preservation during Multi-Agent Collaboration: Federated learning enables collaborative improvement without centralized data collection [41]. Future research should explore protocols for agents to share threat insights while protecting organizational data. Key challenges include secure aggregation, poisoning resistance, and non-IID data robustness. Real-time federated updates could help defensive agents quickly adapt to new attack patterns.

5) Defense Against LLM-based Agent Swarms: As single-agent threats evolve into coordinated multi-agent attacks, future defenses must prepare for the possibility of intelligent agent swarms executing synchronized cyber operations [161]. Future research should focus on developing detection and mitigation techniques specifically tailored to the behavioral signatures of swarm-based attacks. Distributed anomaly detection, decentralized defense architectures, and deception-based countermeasures capable of confusing or fragmenting swarm coordination will be vital. Defensive swarms composed of autonomous security agents could also be explored as a countermeasure, creating dynamic, self-organizing barriers against distributed attacks at machine speed.

6) LLM-based Agent Honeypots: Deception remains a powerful tool in cybersecurity, and the emergence of LLM-based agents unlocks new possibilities for intelligent, adaptive honeypots [134, 139]. Future honeypots could leverage LLM capabilities to engage attackers in realistic dialogues, simulate system behaviors dynamically, and capture detailed telemetry of attack tactics. Developing efficient, scalable LLM-based honeypots could transform cyber defense from a reactive model into a proactive intelligence-gathering operation.

7) Agent-to-Agent Deception: Cyber conflict now includes autonomous adversarial agents [214]. Deception between LLM-based agents is a crucial research frontier. Defensive strategies could deploy decoys and misinformation to mislead attacker agents. Researchers must also defend against malicious agents manipulating defensive AI. Agent-to-agent cyber deception will require interdisciplinary insights from game theory, adversarial machine learning, and multi-agent systems to craft effective tactics and countermeasures.

8 Conclusion

This survey highlights a fundamental shift in the cybersecurity landscape, driven by the rise of autonomous LLM-based cyberattack agents. These agents make sophisticated cyber threats more scalable, more accessible, and more difficult to defend against. As attack costs fall and operational complexity increases, traditional defenses are struggling to keep pace. The spread of coordinated multi-agent systems further amplifies the challenge. To respond, the cybersecurity community must adopt forward-looking strategies that prioritize adaptability, intelligent defense, and proactive threat engagement. Ultimately, understanding the strategic implications of LLM-enabled threats is essential to safeguarding the future of digital infrastructure.

References

[1] Talor Abramovich, Meet Udeshi, Minghao Shao, Kilian Lieret, Haoran Xi, Kimberly Milner, Sofija Jancheska, John Yang, Carlos E Jimenez, Farshad Khorrami, et al. 2024. EnIGMA: Enhanced Interactive Generative Model Agent for CTF Challenges. *arXiv preprint arXiv:2409.16165* (2024).

[2] Nadir Adam, Mansoor Ali, Faisal Naeem, Abdallah S Ghazy, and Georges Kaddoum. 2024. State-of-the-art security schemes for the Internet of Underwater Things: A holistic survey. *IEEE Open Journal of the Communications Society* (2024).

[3] Santosh Reddy Addula, Udit Mamodiya, Weiwei Jiang, and Mohammed Amin Almaiah. 2025. Generative AI-Enhanced Intrusion Detection Framework for Secure Healthcare Networks in MANETs. *SHIFRA* 2025 (2025), 62–68.

[4] Khalifa Afane, Wenqi Wei, Ying Mao, Junaid Farooq, and Juntao Chen. 2024. Next-Generation Phishing: How LLM Agents Empower Cyber Attackers. In *2024 IEEE International Conference on Big Data (BigData)*. IEEE, 2558–2567.

[5] Dennis Agnew, Ashlee Rice-Bladykas, and Janise Mcnair. 2024. Detection of Zero-Day Attacks in a Software-Defined LEO Constellation Network Using Enhanced Network Metric Predictions. *IEEE Open Journal of the Communications Society* (2024).

[6] Chuadhry Mujeeb Ahmed. 2025. AttackLLM: LLM-based Attack Pattern Generation for an Industrial Control System. *arXiv preprint arXiv:2504.04187* (2025).

[7] Dalia Shihab Ahmed, Abbas Abdulazeez Abdulhameed, and Methaq T Gaata. 2024. A Systematic Literature Review on Cyber Attack Detection in Software-Define Networking (SDN). *Mesopotamian Journal of CyberSecurity* 4, 3 (2024), 86–135.

[8] Lin Ai, Tharindu Kumarage, Amrita Bhattacharjee, Zizhou Liu, Zheng Hui, Michael Davinroy, James Cook, Laura Cassani, Kirill Trapeznikov, Matthias Kirchner, et al. 2024. Defending against social engineering attacks in the age of llms. *arXiv preprint arXiv:2406.12263* (2024).

[9] Soby T Ajimon and Sachil Kumar. 2025. Applications of LLMs in Quantum-Aware Cybersecurity Leveraging LLMs for Real-Time Anomaly Detection and Threat Intelligence. In *Leveraging Large Language Models for Quantum-Aware Cybersecurity*. IGI Global Scientific Publishing, 201–246.

[10] Vishwanath Akuthota, Raghunandan Kasula, Sabiha Tasnim Sumona, Masud Mohiuddin, Md Tanzim Reza, and Md Mizanur Rahman. 2023. Vulnerability detection and monitoring using llm. In *2023 IEEE 9th International Women in Engineering (WIE) Conference on Electrical and Computer Engineering (WIECON-ECE)*. IEEE, 309–314.

[11] Jamal Al-Karaki, Muhammad Al-Zafar Khan, and Marwan Omar. 2024. Exploring llms for malware detection: Review, framework design, and countermeasure approaches. *arXiv preprint arXiv:2409.07587* (2024).

[12] Rasha Hameed Khudhur Al-Rubaye and Ayça Kurnaz Türkben. 2024. Using artificial intelligence to evaluating detection of cybersecurity threats in ad hoc networks. *Babylonian Journal of Networking* 2024 (2024), 45–56.

[13] Haitham S Al-Sinani and Chris J Mitchell. 2025. PenTest++: Elevating Ethical Hacking with AI and Automation. *arXiv preprint arXiv:2502.09484* (2025).

[14] Md Tanvirul Alam, Dipkamal Bhusal, Le Nguyen, and Nidhi Rastogi. 2024. Ctibench: A benchmark for evaluating llms in cyber threat intelligence. *arXiv preprint arXiv:2406.07599* (2024).

[15] Theyazn HH Aldhyani and Hasan Alkahtani. 2022. Attacks to automatous vehicles: A deep learning algorithm for cybersecurity. *Sensors* 22, 1 (2022), 360.

[16] Ahmed AlEroud and Izzat Alsmadi. 2017. Identifying cyber-attacks on software defined networks: An inference-based intrusion detection approach. *Journal of Network and Computer Applications* 80 (2017), 152–164.

[17] Bandar Alotaibi. 2025. Cybersecurity Attacks and Detection Methods in Web 3.0 Technology: A Review. *Sensors* 25, 2 (2025), 342.

[18] Lara Alotaibi, Sumayyah Seher, and Nazeeruddin Mohammad. 2024. Cyberattacks using chatgpt: Exploring malicious content generation through prompt engineering. In *2024 ASU international conference in emerging technologies for sustainability and intelligent systems (ICETSIS)*. IEEE, 1304–1311.

[19] Ibrahim Alshehri, Adnan Alshehri, Abdulrahman Almalki, Majed Bamardouf, and Alaqsa Akbar. 2024. Breachseek: A multi-agent automated penetration tester. *arXiv preprint arXiv:2409.03789* (2024).

[20] Atyaf Ismaeel Altameemi, Sahar Jasim Mohammed, Zainab Qahtan Mohammed, Qusay Kanaan Kadhim, and Shaymaa Taha Ahmed. 2024. Enhanced SVM and RNN Classifier for Cyberattacks Detection in Underwater Wireless Sensor Networks. *International Journal of Safety & Security Engineering* 14, 5 (2024).

[21] Martin Andreoni, Willian T Lunardi, George Lawton, and Shreekant Thakkar. 2024. Enhancing autonomous system security and resilience with generative AI: A comprehensive survey. *IEEE Access* (2024).

[22] Maksym Andriushchenko, Alexandra Souly, Mateusz Dziemian, Derek Duenas, Maxwell Lin, Justin Wang, Dan Hendrycks, Andy Zou, Zico Kolter, Matt Fredrikson, et al. 2024. Agentharm: A benchmark for measuring harmfulness of llm agents. *arXiv preprint arXiv:2410.09024* (2024).

[23] Anthropic. 2025. *Progress from our Frontier Red Team*. https://www.anthropic.com/news/strategic-warning-for-ai-risk-progress-and-insights-from-our-frontier-red-team Accessed: 2025-05-02.

[24] Artificial Analysis. 2025. Artificial Analysis: AI Model Evaluation and Insights. https://artificialanalysis.ai/. Accessed: 2025-05-03.

[25] Daniel Ayzenshteyn, Roy Weiss, and Yisroel Mirsky. 2024. The Best Defense is a Good Offense: Countering LLM-Powered Cyberattacks. *arXiv preprint arXiv:2410.15396* (2024).

[26] Efe C Balta, Michael Pease, James Moyne, Kira Barton, and Dawn M Tilbury. 2023. Digital twin-based cyber-attack detection framework for cyber-physical manufacturing systems. *IEEE Transactions on Automation Science and Engineering* 21, 2 (2023), 1695–1712.

[27] Enna Basic and Alberto Giaretta. 2024. Large Language Models and Code Security: A Systematic Literature Review. *arXiv preprint arXiv:2412.15004* (2024).

[28] Mika Beckerich, Laura Plein, and Sergio Coronado. 2023. Ratgpt: Turning online llms into proxies for malware attacks. *arXiv preprint arXiv:2308.09183* (2023).

[29] Nils Begou, Jérémy Vinoy, Andrzej Duda, and Maciej Korczyński. 2023. Exploring the dark side of ai: Advanced phishing attack design and deployment using chatgpt. In *2023 IEEE Conference on Communications and Network Security (CNS)*. IEEE, 1–6.

[30] Mubeena Begum, Gunasekaran Raja, and Mohsen Guizani. 2023. Ai-based sensor attack detection and classification for autonomous vehicles in 6g-v2x environment. *IEEE Transactions on Vehicular Technology* 73, 4 (2023), 5054–5063.

[31] Maciej Besta, Nils Blach, Ales Kubicek, Robert Gerstenberger, Michal Podstawski, Lukas Gianinazzi, Joanna Gajda, Tomasz Lehmann, Hubert Niewiadomski, Piotr Nyczyk, et al. 2024. Graph of thoughts: Solving elaborate problems with large language models. In *Proceedings of the AAAI Conference on Artificial Intelligence*, Vol. 38. 17682–17690.

[32] Manish Bhatt, Sahana Chennabasappa, Yue Li, Cyrus Nikolaidis, Daniel Song, Shengye Wan, Faizan Ahmad, Cornelius Aschermann, Yaohui Chen, Dhaval Kapil, et al. 2024. Cyberseceval 2: A wide-ranging cybersecurity evaluation suite for large language models. *arXiv preprint arXiv:2404.13161* (2024).

[33] Manish Bhatt, Sahana Chennabasappa, Cyrus Nikolaidis, Shengye Wan, Ivan Evtimov, Dominik Gabi, Daniel Song, Faizan Ahmad, Cornelius Aschermann, Lorenzo Fontana, et al. 2023. Purple llama cyberseceval: A secure coding benchmark for language models. *arXiv preprint arXiv:2312.04724* (2023).

[34] Ting Bi, Chenghang Ye, Zheyu Yang, Ziyi Zhou, Cui Tang, Jun Zhang, Zui Tao, Kailong Wang, Liting Zhou, Yang Yang, et al. 2025. On the Feasibility of Using MultiModal LLMs to Execute AR Social Engineering Attacks. *arXiv preprint arXiv:2504.13209* (2025).

[35] Stanislas G Bianou and Rodrigue G Batogna. 2024. PENTEST-AI, an LLM-Powered multi-agents framework for penetration testing automation leveraging mitre attack. In *2024 IEEE International Conference on Cyber Security and Resilience (CSR)*. IEEE, 763–770.

[36] Sarah Binhulayyil, Shancang Li, and Neetesh Saxena. 2024. IoT Vulnerability Detection using Featureless LLM CyBert Model. In *2024 IEEE 23rd International Conference on Trust, Security and Privacy in Computing and Communications (TrustCom)*. IEEE, 2474–2480.

[37] Emilie Bout, Valeria Loscri, and Antoine Gallais. 2021. How machine learning changes the nature of cyberattacks on IoT networks: A survey. *IEEE Communications Surveys & Tutorials* 24, 1 (2021), 248–279.

[38] William N Caballero and Phillip R Jenkins. 2025. On large language models in national security applications. *Stat* 14, 2 (2025), e70057.

[39] Tri Cao, Chengyu Huang, Yuexin Li, Wang Huilin, Amy He, Nay Oo, and Bryan Hooi. 2025. Phishagent: a robust multimodal agent for phishing webpage detection. In *Proceedings of the AAAI Conference on Artificial Intelligence*, Vol. 39. 27869–27877.

[40] PV Charan, Hrushikesh Chunduri, P Mohan Anand, and Sandeep K Shukla. 2023. From text to mitre techniques: Exploring the malicious use of large language models for generating cyber attack payloads. *arXiv preprint arXiv:2305.15336* (2023).

[41] Chaochao Chen, Xiaohua Feng, Yuyuan Li, Lingjuan Lyu, Jun Zhou, Xiaolin Zheng, and Jianwei Yin. 2024. Integration of large language models and federated learning. *Patterns* 5, 12 (2024).

[42] Fengchao Chen, Tingmin Wu, Van Nguyen, Shuo Wang, Hongsheng Hu, Alsharif Abuadbba, and Carsten Rudolph. 2024. Adapting to Cyber Threats: A Phishing Evolution Network (PEN) Framework for Phishing Generation and Analyzing Evolution Patterns using Large Language Models. *arXiv preprint arXiv:2411.11389* (2024).

[43] Hongbo Chen, Yifan Zhang, Xing Han, Huanyao Rong, Yuheng Zhang, Tianhao Mao, Hang Zhang, XiaoFeng Wang, Luyi Xing, and Xun Chen. 2024. WitheredLeaf: Finding Entity-Inconsistency Bugs with LLMs. *arXiv preprint arXiv:2405.01668* (2024).

[44] Yutong Cheng, Osama Bajaber, Saimon Amanuel Tsegai, Dawn Song, and Peng Gao. 2024. CTINEXUS: Leveraging Optimized LLM In-Context Learning for Constructing Cybersecurity Knowledge Graphs Under Data Scarcity. *arXiv preprint arXiv:2410.21060* (2024).

[45] Vanessa Clairoux-Trepanier, Isa-May Beauchamp, Estelle Ruellan, Masarah Paquet-Clouston, Serge-Olivier Paquette, and Eric Clay. 2024. The use of large language models (llm) for cyber threat intelligence (cti) in cybercrime forums. *arXiv preprint arXiv:2408.03354* (2024).

[46] Mustafa Cosar. 2022. Cyber attacks on unmanned aerial vehicles and cyber security measures. *The Eurasia Proceedings of Science Technology Engineering and Mathematics* 21 (2022), 258–265.

[47] Garrett Crumrine, Izzat Alsmadi, Jesus Guerrero, and Yuvaraj Munian. 2024. Transforming computer security and public trust through the exploration of fine-tuning large language models. *arXiv preprint arXiv:2406.00628* (2024).

[48] Susheela Dahiya and Manik Garg. 2019. Unmanned aerial vehicles: Vulnerability to cyber attacks. In *International Conference on Unmanned Aerial System in Geomatics*. Springer, 201–211.

[49] Seyed Shayan Daneshvar, Yu Nong, Xu Yang, Shaowei Wang, and Haipeng Cai. 2024. Exploring RAG-based Vulnerability Augmentation with LLMs. *arXiv preprint arXiv:2408.04125* (2024).

[50] Herve Debar, Sven Dietrich, Pavel Laskov, Emil C Lupu, and Eirini Ntoutsi. 2024. Emerging Security Challenges of Large Language Models. *arXiv preprint arXiv:2412.17614* (2024).

[51] Pritam Deka, Sampath Rajapaksha, Ruby Rani, Amirah Almutairi, and Erisa Karafili. 2024. Attacker: towards enhancing cyber-attack attribution with a named entity recognition dataset. In *International Conference on Web Information Systems Engineering*. Springer, 255–270.

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

[53] Alaeddine Diaf, Abdelaziz Amara Korba, Nour Elislem Karabadji, and Yacine Ghamri-Doudane. 2024. Beyond detection: Leveraging large language models for cyber attack prediction in iot networks. In *2024 20th International Conference on Distributed Computing in Smart Systems and the Internet of Things (DCOSS-IoT)*. IEEE, 117–123.

[54] Alaeddine Diaf, Abdelaziz Amara Korba, Nour Elislem Karabadji, and Yacine Ghamri-Doudane. 2025. BARTPredict: Empowering IoT Security with LLM-Driven Cyber Threat Prediction. *arXiv preprint arXiv:2501.01664* (2025).

[55] Ye Dong, Yan Lin Aung, Sudipta Chattopadhyay, and Jianying Zhou. 2025. ChatIoT: Large Language Model-based Security Assistant for Internet of Things with Retrieval-Augmented Generation. *arXiv preprint arXiv:2502.09896* (2025).

[56] Dan Du, Xingmao Guan, Yuling Liu, Bo Jiang, Song Liu, Huamin Feng, and Junrong Liu. 2024. MAD-LLM: A Novel Approach for Alert-Based Multi-stage Attack Detection via LLM. In *2024 IEEE International Symposium on Parallel and Distributed Processing with Applications (ISPA)*. IEEE, 2046–2053.

[57] Xueying Du, Geng Zheng, Kaixin Wang, Jiayi Feng, Wentai Deng, Mingwei Liu, Bihuan Chen, Xin Peng, Tao Ma, and Yiling Lou. 2024. Vul-rag: Enhancing llm-based vulnerability detection via knowledge-level rag. *arXiv preprint arXiv:2406.11147* (2024).

[58] Wenli Duo, MengChu Zhou, and Abdullah Abusorrah. 2022. A survey of cyber attacks on cyber physical systems: Recent advances and challenges. *IEEE/CAA Journal of Automatica Sinica* 9, 5 (2022), 784–800.

[59] Joshua Dwight. 2024. Building Cyber Attack Trees with the Help of My LLM? A Mixed Method Study. In *Proceedings of the 2024 12th International Conference on Computer and Communications Management*. 132–138.

[60] Wenjun Fan, Zichen Yang, Yuanzhen Liu, Lang Qin, and Jia Liu. 2024. HoneyLLM: A Large Language Model-Powered Medium-Interaction Honeypot. In *International Conference on Information and Communications Security*. Springer, 253–272.

[61] Richard Fang, Rohan Bindu, Akul Gupta, and Daniel Kang. 2024. Llm agents can autonomously exploit one-day vulnerabilities. *arXiv preprint arXiv:2404.08144* 13 (2024), 14.

[62] Richard Fang, Rohan Bindu, Akul Gupta, Qiusi Zhan, and Daniel Kang. 2024. Llm agents can autonomously hack websites. *arXiv preprint arXiv:2402.06664* (2024).

[63] Bo Feng, Qiang Li, Yuede Ji, Dong Guo, and Xiangyu Meng. 2019. Stopping the cyberattack in the early stage: assessing the security risks of social network users. *Security and Communication Networks* 2019, 1 (2019), 3053418.

[64] Sidong Feng and Chunyang Chen. 2024. Prompting is all you need: Automated android bug replay with large language models. In *Proceedings of the 46th IEEE/ACM International Conference on Software Engineering*. 1–13.

[65] Mohamed Amine Ferrag, Fatima Alwahedi, Ammar Battah, Bilel Cherif, Abdechakour Mechri, and Norbert Tihanyi. 2024. Generative ai and large language models for cyber security: All insights you need. *Available at SSRN 4853709* (2024).

[66] Mohamed Amine Ferrag, Ammar Battah, Norbert Tihanyi, Ridhi Jain, Diana Maimut, Fatima Alwahedi, Thierry Lestable, Narinderjit Singh Thandi, Abdechakour Mechri, Merouane Debbah, et al. 2023. SecureFalcon: Are we there yet in automated software vulnerability detection with LLMs? *arXiv preprint arXiv:2307.06616* (2023).

[67] Mohamed Amine Ferrag, Mthandazo Ndhlovu, Norbert Tihanyi, Lucas C Cordeiro, Merouane Debbah, and Thierry Lestable. 2023. Revolutionizing cyber threat detection with large language models. *arXiv preprint arXiv:2306.14263* (2023), 195–202.

[68] Romy Fieblinger, Md Tanvirul Alam, and Nidhi Rastogi. 2024. Actionable cyber threat intelligence using knowledge graphs and large language models. In *2024 IEEE European Symposium on Security and Privacy Workshops (EuroS&PW)*. IEEE, 100–111.

[69] João Figueiredo, Afonso Carvalho, Daniel Castro, Daniel Gonçalves, and Nuno Santos. 2024. On the Feasibility of Fully AI-automated Vishing Attacks. *arXiv preprint arXiv:2409.13793* (2024).

[70] Mohamed Fazil Mohamed Firdhous, Walid Elbreiki, Ibrahim Abdullahi, BH Sudantha, and Rahmat Budiarto. 2023. Wormgpt: a large language model chatbot for criminals. In *2023 24th International Arab Conference on Information Technology (ACIT)*. IEEE, 1–6.

[71] Jerson Francia, Derek Hansen, Ben Schooley, Matthew Taylor, Shydra Murray, and Greg Snow. 2024. Assessing AI vs human-authored spear phishing sms attacks: An empirical study using the trapd method. *arXiv preprint arXiv:2406.13049* (2024).

[72] Yunfan Gao, Yun Xiong, Xinyu Gao, Kangxiang Jia, Jinliu Pan, Yuxi Bi, Yixin Dai, Jiawei Sun, Haofen Wang, and Haofen Wang. 2023. Retrieval-augmented generation for large language models: A survey. *arXiv preprint arXiv:2312.10997* 2 (2023), 1.

[73] Rikhiya Ghosh, Oladimeji Farri, Hans-Martin von Stockhausen, Martin Schmitt, and George Marica Vasile. 2024. CVE-LLM: Automatic vulnerability evaluation in medical device industry using large language models. *arXiv preprint arXiv:2407.14640* (2024).

[74] Luca Gioacchini, Marco Mellia, Idilio Drago, Alexander Delsanto, Giuseppe Siracusano, and Roberto Bifulco. 2024. AutoPenBench: Benchmarking Generative Agents for Penetration Testing. *arXiv preprint arXiv:2410.03225* (2024).

[75] Sergei Glazunov and Mark Brand. 2024. *Project Naptime: Evaluating Offensive Security Capabilities of Large Language Models*. https://googleprojectzero.blogspot.com/2024/06/project-naptime.html Accessed: 2025-05-02.

[76] Dhruva Goyal, Sitaraman Subramanian, and Aditya Peela. 2024. Hacking, the lazy way: LLM augmented pentesting. *arXiv preprint arXiv:2409.09493* (2024).

[77] Jonathan Gregory and Qi Liao. 2024. Autonomous Cyberattack with Security-Augmented Generative Artificial Intelligence. In *2024 IEEE International Conference on Cyber Security and Resilience (CSR)*. IEEE, 270–275.

[78] Chengquan Guo, Xun Liu, Chulin Xie, Andy Zhou, Yi Zeng, Zinan Lin, Dawn Song, and Bo Li. 2024. Redcode: Risky code execution and generation benchmark for code agents. *Advances in Neural Information Processing Systems* 37 (2024), 106190–106236.

[79] Chengquan Guo, Chulin Xie, Yu Yang, Zinan Lin, and Bo Li. [n. d.]. RedCodeAgent: Automatic Red-teaming Agent against Code Agents. ([n. d.]).

[80] Wenbo Guo, Yujin Potter, Tianneng Shi, Zhun Wang, Andy Zhang, and Dawn Song. 2025. SoK: Frontier AI's Impact on the Cybersecurity Landscape. *arXiv preprint arXiv:2504.05408* (2025).

[81] Achref Haddaji, Samiha Ayed, and Lamia Chaari Fourati. 2022. Artificial Intelligence techniques to mitigate cyber-attacks within vehicular networks: Survey. *Computers and Electrical Engineering* 104 (2022), 108460.

[82] Jassim Happa, Mashhuda Glencross, and Anthony Steed. 2019. Cyber security threats and challenges in collaborative mixed-reality. *Frontiers in ICT* 6 (2019), 5.

[83] Andreas Happe and Jürgen Cito. 2023. 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*. 2082–2086.

[84] Andreas Happe and Jürgen Cito. 2025. Can LLMs Hack Enterprise Networks? Autonomous Assumed Breach Penetration-Testing Active Directory Networks. *arXiv preprint arXiv:2502.04227* (2025).

[85] Mohammed Hassanin, Marwa Keshk, Sara Salim, Majid Alsubaie, and Dharmendra Sharma. 2025. Pllm-cs: Pre-trained large language model (llm) for cyber threat detection in satellite networks. *Ad Hoc Networks* 166 (2025), 103645.

[86] Junda He, Christoph Treude, and David Lo. 2024. LLM-Based Multi-Agent Systems for Software Engineering: Literature Review, Vision and the Road Ahead. *ACM Transactions on Software Engineering and Methodology* (2024).

[87] Md Imran Hossen, Jianyi Zhang, Yinzhi Cao, and Xiali Hei. 2024. Assessing cybersecurity vulnerabilities in code large language models. *arXiv preprint arXiv:2404.18567* (2024).

[88] Edward J Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, Weizhu Chen, et al. 2022. Lora: Low-rank adaptation of large language models. *ICLR* 1, 2 (2022), 3.

[89] Junjie Huang and Quanyan Zhu. [n. d.]. Penhealnet: An Agent-Based Llm Framework for Automated Pentesting and Optimal Remediation. *Available at SSRN 4941478* ([n. d.]).

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

[91] Liangyi Huang and Xusheng Xiao. 2024. CTIKG: LLM-Powered Knowledge Graph Construction from Cyber Threat Intelligence. In *First Conference on Language Modeling*.

[92] Sian-Yao Huang, Cheng-Lin Yang, Che-Yu Lin, and Chun-Ying Huang. 2024. CmdCaliper: A Semantic-Aware Command-Line Embedding Model and Dataset for Security Research. *arXiv preprint arXiv:2411.01176* (2024).

[93] Nourhan Ibrahim and Rasha Kashef. 2025. Exploring the emerging role of large language models in smart grid cybersecurity: a survey of attacks, detection mechanisms, and mitigation strategies. *Frontiers in Energy Research* 13 (2025), 1531655.

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

[95] Hamed Jelodar, Samita Bai, Parisa Hamedi, Hesamodin Mohammadian, Roozbeh Razavi-Far, and Ali Ghorbani. 2025. Large Language Model (LLM) for Software Security: Code Analysis, Malware Analysis, Reverse Engineering. *arXiv preprint arXiv:2504.07137* (2025).

[96] Hangyuan Ji, Jian Yang, Linzheng Chai, Chaoren Wei, Liqun Yang, Yunlong Duan, Yunli Wang, Tianzhen Sun, Hongcheng Guo, Tongliang Li, et al. 2024. Sevenllm: Benchmarking, eliciting, and enhancing abilities of large language models in cyber threat intelligence. *arXiv preprint arXiv:2405.03446* (2024).

[97] Haolin Jin, Linghan Huang, Haipeng Cai, Jun Yan, Bo Li, and Huaming Chen. 2024. From llms to llm-based agents for software engineering: A survey of current, challenges and future. *arXiv preprint arXiv:2408.02479* (2024).

[98] Jiandong Jin, Bowen Tang, Mingxuan Ma, Xiao Liu, Yunfei Wang, Qingnan Lai, Jia Yang, and Changling Zhou. 2024. Crimson: Empowering strategic reasoning in cybersecurity through large language models. *arXiv preprint arXiv:2403.00878* (2024).

[99] D Jocil and R Vadivel. 2024. Network Security Risks and Solutions Through Automated Toolkits in Underwater Sensor Network: A Survey. In *Leveraging Artificial Intelligence (AI) Competencies for Next-Generation Cybersecurity Solutions*. Apple Academic Press, 1–37.

[100] Tianjie Ju, Yiting Wang, Xinbei Ma, Pengzhou Cheng, Haodong Zhao, Yulong Wang, Lifeng Liu, Jian Xie, Zhuosheng Zhang, and Gongshen Liu. 2024. Flooding spread of manipulated knowledge in llm-based multi-agent communities. *arXiv preprint arXiv:2407.07791* (2024).

[101] Tran Viet Khoa, Do Hai Son, Dinh Thai Hoang, Nguyen Linh Trung, Tran Thi Thuy Quynh, Diep N Nguyen, Nguyen Viet Ha, and Eryk Dutkiewicz. 2024. Collaborative learning for cyberattack detection in blockchain networks. *IEEE Transactions on Systems, Man, and Cybernetics: Systems* (2024).

[102] Fabian Kilger, Alexandre Kabil, Volker Tippmann, Gudrun Klinker, and Marc-Oliver Pahl. 2021. Detecting and preventing faked mixed reality. In *2021 IEEE 4th International Conference on Multimedia Information Processing and Retrieval (MIPR)*. IEEE, 399–405.

[103] Hanna Kim, Minkyoo Song, Seung Ho Na, Seungwon Shin, and Kimin Lee. 2024. When LLMs Go Online: The Emerging Threat of Web-Enabled LLMs. *arXiv preprint arXiv:2410.14569* (2024).

[104] Masaya Kobayashi, Masane Fuchi, Amar Zanashir, Tomonori Yoneda, and Tomohiro Takagi. 2025. Construction and Evaluation of LLM-based agents for Semi-Autonomous penetration testing. *arXiv preprint arXiv:2502.15506* (2025).

[105] He Kong, Die Hu, Jingguo Ge, Liangxiong Li, Tong Li, and Bingzhen Wu. 2025. VulnBot: Autonomous Penetration Testing for A Multi-Agent Collaborative Framework. *arXiv preprint arXiv:2501.13411* (2025).

[106] Peng-Yong Kong. 2021. A survey of cyberattack countermeasures for unmanned aerial vehicles. *IEEE Access* 9 (2021), 148244–148263.

[107] Antreas Konstantinou, Dimitrios Kasimatis, William J Buchanan, Sana Ullah Jan, Jawad Ahmad, Ilias Politis, and Nikolaos Pitropakis. 2025. Leveraging LLMs for Non-Security Experts in Threat Hunting: Detecting Living off the Land Techniques. *Machine Learning and Knowledge Extraction* 7, 2 (2025), 31.

[108] S Krishnaveni, Thomas M Chen, Mithileysh Sathiyanarayanan, and B Amutha. 2024. CyberDefender: an integrated intelligent defense framework for digital-twin-based industrial cyber-physical systems. *Cluster Computing* 27, 6 (2024), 7273–7306.

[109] Yury A Kuleshov, Kabir Nagpal, Korel Ucpinar, Alisha Gadaginmath, Sanjana Gadaginmath, Katie O'Daniel, Dalbert Sun, Lucas Tan, Nathan Veatch, and Hridhay Monangi. 2024. Cyber attacks on avionics networks in digital twin environment: detection and defense. In *AIAA SCITECH 2024 Forum*. 0277.

[110] Tharindu Kumarage, Cameron Johnson, Jadie Adams, Lin Ai, Matthias Kirchner, Anthony Hoogs, Joshua Garland, Julia Hirschberg, Arslan Basharat, and Huan Liu. 2025. Personalized Attacks of Social Engineering in Multi-turn Conversations–LLM Agents for Simulation and Detection. *arXiv preprint arXiv:2503.15552* (2025).

[111] Mehmet Necip Kurt, Oyetunji Ogundijo, Chong Li, and Xiaodong Wang. 2018. Online cyber-attack detection in smart grid: A reinforcement learning approach. *IEEE Transactions on Smart Grid* 10, 5 (2018), 5174–5185.

[112] Tan Duy Le, Adnan Anwar, Seng W Loke, Razvan Beuran, and Yasuo Tan. 2020. Gridattacksim: A cyber attack simulation framework for smart grids. *Electronics* 9, 8 (2020), 1218.

[113] Tan Duy Le, Mengmeng Ge, Adnan Anwar, Seng W Loke, Razvan Beuran, Robin Doss, and Yasuo Tan. 2022. Gridattackanalyzer: A cyber attack analysis framework for smart grids. *Sensors* 22, 13 (2022), 4795.

[114] Leonid Legashev and Arthur Zhigalov. 2025. Investigating cybersecurity incidents using large language models in latest-generation wireless networks. *arXiv preprint arXiv:2504.13196* (2025).

[115] Matan Levi, Yair Allouche, Daniel Ohayon, and Anton Puzanov. 2025. Cyberpal. ai: Empowering llms with expert-driven cybersecurity instructions. In *Proceedings of the AAAI Conference on Artificial Intelligence*, Vol. 39. 24402–24412.

[116] Jiangnan Li, Yingyuan Yang, and Jinyuan Sun. 2024. Risks of practicing large language models in smart grid: Threat modeling and validation. *arXiv preprint arXiv:2405.06237* (2024).

[117] Xu Li, Xiaohui Liang, Rongxing Lu, Xuemin Shen, Xiaodong Lin, and Haojin Zhu. 2012. Securing smart grid: cyber attacks, countermeasures, and challenges. *IEEE Communications Magazine* 50, 8 (2012), 38–45.

[118] Zilong Lin, Jian Cui, Xiaojing Liao, and XiaoFeng Wang. 2024. Malla: Demystifying real-world large language model integrated malicious services. In *33rd USENIX Security Symposium (USENIX Security 24)*. 4693–4710.

[119] Jiaqi Liu and Noriaki Kamiyama. 2024. Investigating Impact of DDoS Attack and CPA Targeting CDN Caches. In *NOMS 2024-2024 IEEE Network Operations and Management Symposium*. IEEE, 1–6.

[120] Junwei Liu, Kaixin Wang, Yixuan Chen, Xin Peng, Zhenpeng Chen, Lingming Zhang, and Yiling Lou. 2024. Large language model-based agents for software engineering: A survey. *arXiv preprint arXiv:2409.02977* (2024).

[121] Zefang Liu. 2023. Secqa: A concise question-answering dataset for evaluating large language models in computer security. *arXiv preprint arXiv:2312.15838* (2023).

[122] Guilong Lu, Xiaolin Ju, Xiang Chen, Wenlong Pei, and Zhilong Cai. 2024. GRACE: Empowering LLM-based software vulnerability detection with graph structure and in-context learning. *Journal of Systems and Software* 212 (2024), 112031.

[123] Junyu Luo, Weizhi Zhang, Ye Yuan, Yusheng Zhao, Junwei Yang, Yiyang Gu, Bohan Wu, Binqi Chen, Ziyue Qiao, Qingqing Long, et al. 2025. Large Language Model Agent: A Survey on Methodology, Applications and Challenges. *arXiv preprint arXiv:2503.21460* (2025).

[124] Minzhao Lyu, Hassan Habibi Gharakheili, and Vijay Sivaraman. 2024. A survey on enterprise network security: Asset behavioral monitoring and distributed attack detection. *IEEE Access* (2024).

[125] Sathiya Kumaran Mani, Yajie Zhou, Kevin Hsieh, Santiago Segarra, Trevor Eberl, Eliran Azulai, Ido Frizler, Ranveer Chandra, and Srikanth Kandula. 2023. Enhancing network management using code generated by large language models. In *Proceedings of the 22nd ACM Workshop on Hot Topics in Networks*. 196–204.

[126] Mantas Mazeika, Long Phan, Xuwang Yin, Andy Zou, Zifan Wang, Norman Mu, Elham Sakhaee, Nathaniel Li, Steven Basart, Bo Li, et al. 2024. Harmbench: A standardized evaluation framework for automated red teaming and robust refusal. *arXiv preprint arXiv:2402.04249* (2024).

[127] Microsoft. 2025. What Is the Cyber Kill Chain? https://www.microsoft.com/en-us/security/business/security-101/what-is-cyber-kill-chain Accessed: 2025-05-06.

[128] Shaswata Mitra, Subash Neupane, Trisha Chakraborty, Sudip Mittal, Aritran Piplai, Manas Gaur, and Shahram Rahimi. 2024. Localintel: Generating organizational threat intelligence from global and local cyber knowledge. *arXiv preprint arXiv:2401.10036* (2024).

[129] R Mohandas, Karthik Kumar Vaigandla, N Sivapriya, and K Kirubasankar. 2024. Detection and Evaluation of Cybersecurity Threats in MANET Based on AI. In *2024 4th International Conference on Ubiquitous Computing and Intelligent Information Systems (ICUIS)*. IEEE, 1486–1492.

[130] Stephen Moskal, Sam Laney, Erik Hemberg, and Una-May O'Reilly. 2023. Llms killed the script kiddie: How agents supported by large language models change the landscape of network threat testing. *arXiv preprint arXiv:2310.06936* (2023).

[131] Lajos Muzsai, David Imolai, and András Lukács. 2024. HackSynth: LLM Agent and Evaluation Framework for Autonomous Penetration Testing. *arXiv preprint arXiv:2412.01778* (2024).

[132] Sho Nakatani. 2025. RapidPen: Fully Automated IP-to-Shell Penetration Testing with LLM-based Agents. *arXiv preprint arXiv:2502.16730* (2025).

[133] Carlos Natalino, Aysegul Yayimli, Lena Wosinska, and Marija Furdek. 2019. Infrastructure upgrade framework for content delivery networks robust to targeted attacks. *Optical Switching and Networking* 31 (2019), 202–210.

[134] Lewis Newsham, Ryan Hyland, and Daniel Prince. 2025. Inducing Personality in LLM-Based Honeypot Agents: Measuring the Effect on Human-Like Agenda Generation. *arXiv preprint arXiv:2503.19752* (2025).

[135] Tri Nguyen, Huong Nguyen, Ahmad Ijaz, Saeid Sheikhi, Athanasios V Vasilakos, and Panos Kostakos. 2024. Large language models in 6g security: challenges and opportunities. *arXiv preprint arXiv:2403.12239* (2024).

[136] Tomas Nieponice, Veronica Valeros, and Sebastian Garcia. 2025. ARACNE: An LLM-Based Autonomous Shell Pentesting Agent. *arXiv preprint arXiv:2502.18528* (2025).

[137] Liang-bo Ning, Shijie Wang, Wenqi Fan, Qing Li, Xin Xu, Hao Chen, and Feiran Huang. 2024. Cheatagent: Attacking llm-empowered recommender systems via llm agent. In *Proceedings of the 30th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*. 2284–2295.

[138] Temitayo O Olowu, Shamini Dharmasena, Alexandar Hernandez, and Arif Sarwat. 2021. Impact analysis of cyber attacks on smart grid: A review and case study. *New Research Directions in Solar Energy Technologies* (2021), 31–51.

[139] Hakan T Otal and M Abdullah Canbaz. 2024. LLM Honeypot: Leveraging Large Language Models as Advanced Interactive Honeypot Systems. In *2024 IEEE Conference on Communications and Network Security (CNS)*. IEEE, 1–6.

[140] Long Ouyang, Jeffrey Wu, Xu Jiang, Diogo Almeida, Carroll Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, et al. 2022. Training language models to follow instructions with human feedback. *Advances in neural information processing systems* 35

(2022), 27730–27744.

[141] Shirui Pan, Linhao Luo, Yufei Wang, Chen Chen, Jiapu Wang, and Xindong Wu. 2024. Unifying large language models and knowledge graphs: A roadmap. *IEEE Transactions on Knowledge and Data Engineering* 36, 7 (2024), 3580–3599.

[142] Francesco Panebianco, Andrea Isgro, Stefano Longari, Stefano Zanero, Michele Carminati, et al. 2025. Guessing as a service: Large language models are not yet ready for vulnerability detection. In *Guessing As A Service: Large Language Models Are Not Yet Ready For Vulnerability Detection*. N–A.

[143] Abigail Paradise, Asaf Shabtai, Rami Puzis, Aviad Elyashar, Yuval Elovici, Mehran Roshandel, and Christoph Peylo. 2017. Creation and management of social network honeypots for detecting targeted cyber attacks. *IEEE transactions on computational social systems* 4, 3 (2017), 65–79.

[144] Cherilyn E Pascoe. 2023. Public Draft: The NIST Cybersecurity Framework 2.0. *National Institute of Standards and Technology* (2023).

[145] Samuele Pasini, Jinhan Kim, Tommaso Aiello, Rocio Cabrera Lozoya, Antonino Sabetta, and Paolo Tonella. 2024. Evaluating and Improving the Robustness of Security Attack Detectors Generated by LLMs. *arXiv preprint arXiv:2411.18216* (2024).

[146] Dario Pasquini, Evgenios M Kornaropoulos, and Giuseppe Ateniese. 2024. Hacking Back the AI-Hacker: Prompt Injection as a Defense Against LLM-driven Cyberattacks. *arXiv preprint arXiv:2410.20911* (2024).

[147] Kapil Patil and Bhavin Desai. 2024. Leveraging llm for zero-day exploit detection in cloud networks. *Asian American Research Letters Journal* 1, 4 (2024).

[148] Constantinos Patsakis, Fran Casino, and Nikolaos Lykousas. 2024. Assessing LLMs in malicious code deobfuscation of real-world malware campaigns. *Expert Systems with Applications* 256 (2024), 124912.

[149] Shuva Paul, Farhad Alemi, and Richard Macwan. 2025. LLM-Assisted Proactive Threat Intelligence for Automated Reasoning. *arXiv preprint arXiv:2504.00428* (2025).

[150] Mary Phuong, Matthew Aitchison, Elliot Catt, Sarah Cogan, Alexandre Kaskasoli, Victoria Krakovna, David Lindner, Matthew Rahtz, Yannis Assael, Sarah Hodkinson, et al. 2024. Evaluating frontier models for dangerous capabilities. *arXiv preprint arXiv:2403.13793* (2024).

[151] Brett Piggott, Siddhant Patil, Guohuan Feng, Ibrahim Odat, Rajdeep Mukherjee, Balakrishnan Dharmalingam, and Anyi Liu. 2023. Net-GPT: A LLM-empowered man-in-the-middle chatbot for unmanned aerial vehicle. In *Proceedings of the Eighth ACM/IEEE Symposium on Edge Computing*. 287–293.

[152] Yuriy Ponochovnyi, Oleg Ivanchenko, Vyacheslav Kharchenko, Iryna Udovyk, and Eduard Baiev. 2022. Models for Cloud System Availability Assessment Considering Attacks on CDN and ML Based Parametrization.. In *COLINS*. 1149–1159.

[153] Derry Pratama, Naufal Suryanto, Andro Aprila Adiputra, Thi-Thu-Huong Le, Ahmada Yusril Kadiptya, Muhammad Iqbal, and Howon Kim. 2024. Cipher: Cybersecurity intelligent penetration-testing helper for ethical researcher. *Sensors* 24, 21 (2024), 6878.

[154] Yujia Qin, Shihao Liang, Yining Ye, Kunlun Zhu, Lan Yan, Yaxi Lu, Yankai Lin, Xin Cong, Xiangru Tang, Bill Qian, et al. 2023. Toolllm: Facilitating large language models to master 16000+ real-world apis. *arXiv preprint arXiv:2307.16789* (2023).

[155] Jianing Qiu, Lin Li, Jiankai Sun, Hao Wei, Zhe Xu, Kyle Lam, and Wu Yuan. 2025. Emerging Cyber Attack Risks of Medical AI Agents. *arXiv preprint arXiv:2504.03759* (2025).

[156] Sampath Rajapaksha, Harsha Kalutarage, M Omar Al-Kadri, Andrei Petrovski, Garikayi Madzudzo, and Madeline Cheah. 2023. Ai-based intrusion detection systems for in-vehicle networks: A survey. *Comput. Surveys* 55, 11 (2023), 1–40.

[157] Hooman Razavi and Mohammad Reza Jamali. 2024. Large Language Models (LLM) for Estimating the Cost of Cyber-attacks. In *2024 11th International Symposium on Telecommunications (IST)*. IEEE, 403–409.

[158] Daniel Reti, Norman Becker, Tillmann Angeli, Anasuya Chattopadhyay, Daniel Schneider, Sebastian Vollmer, and Hans D Schotten. 2024. Act as a honeytoken generator! an investigation into honeytoken generation with large language models. In *Proceedings of the 11th ACM Workshop on Adaptive and Autonomous Cyber Defense*. 1–12.

[159] Maria Rigaki, Carlos Catania, and Sebastian Garcia. 2024. Hackphyr: A Local Fine-Tuned LLM Agent for Network Security Environments. *arXiv preprint arXiv:2409.11276* (2024).

[160] Dan Ristea, Vasilios Mavroudis, and Chris Hicks. 2024. AI Cyber Risk Benchmark: Automated Exploitation Capabilities. *arXiv preprint arXiv:2410.21939* (2024).

[161] Mikel Rodriguez, Raluca Ada Popa, Four Flynn, Lihao Liang, Allan Dafoe, and Anna Wang. 2025. A Framework for Evaluating Emerging Cyberattack Capabilities of AI. *arXiv preprint arXiv:2503.11917* (2025).

[162] Christian Rondanini, Barbara Carminati, Elena Ferrari, Ashish Kundu, and Akshay Jajoo. 2024. Large Language Models to Enhance Malware Detection in Edge Computing. In *2024 IEEE 6th International Conference on Trust, Privacy and Security in Intelligent Systems, and Applications (TPS-ISA)*. IEEE, 1–10.

[163] Sayak Saha Roy, Poojitha Thota, Krishna Vamsi Naragam, and Shirin Nilizadeh. 2023. From Chatbots to PhishBots?–Preventing Phishing scams created using ChatGPT, Google Bard and Claude. *arXiv preprint arXiv:2310.19181* (2023).

[164] Yuval Schwartz, Lavi Benshimol, Dudu Mimran, Yuval Elovici, and Asaf Shabtai. 2024. Llmcloudhunter: Harnessing llms for automated extraction of detection rules from cloud-based cti. *arXiv preprint arXiv:2407.05194* (2024).

[165] Hichem Sedjelmaci, Sidi Mohammed Senouci, and Nirwan Ansari. 2017. A hierarchical detection and response system to enhance security against lethal cyber-attacks in UAV networks. *IEEE Transactions on Systems, Man, and Cybernetics: Systems* 48, 9 (2017), 1594–1606.

[166] Hichem Sedjelmaci, Sidi Mohammed Senouci, and Mohamed-Ayoub Messous. 2016. How to detect cyber-attacks in unmanned aerial vehicles network?. In *2016 IEEE global communications conference (GLOBECOM)*. IEEE, 1–6.

[167] Samaneh Shafee, Alysson Bessani, and Pedro M Ferreira. 2024. Evaluation of LLM chatbots for OSINT-based cyber threat awareness. *arXiv preprint arXiv:2401.15127* (2024).

[168] Rahman Shafique, Furqan Rustam, Gyu Sang Choi, and Anca Delia Jurcut. 2024. Enhancing in-vehicle network security against ai-generated cyberattacks using machine learning. In *2024 IEEE Wireless Communications and Networking Conference (WCNC)*. IEEE, 1–6.

[169] Weijie Shan, Teng Long, and Zhangbing Zhou. 2024. Adversarial Attacks on IoT Systems Leveraging Large Language Models. In *2024 International Conference on Identification, Information and Knowledge in the Internet of Things (IIKI)*. IEEE, 154–159.

[170] Xinyue Shen, Zeyuan Chen, Michael Backes, Yun Shen, and Yang Zhang. 2024. "Do anything now": Characterizing and evaluating in-the-wild jailbreak prompts on large language models. In *Proceedings of the 2024 on ACM SIGSAC Conference on Computer and Communications Security*. 1671–1685.

[171] Xiangmin Shen, Lingzhi Wang, Zhenyuan Li, Yan Chen, Wencheng Zhao, Dawei Sun, Jiashui Wang, and Wei Ruan. 2024. PentestAgent: Incorporating LLM Agents to Automated Penetration Testing. *arXiv preprint arXiv:2411.05185* (2024).

[172] Xuemin Sherman Shen, Xinyu Huang, Jianzhe Xue, Conghao Zhou, Xiufang Shi, and Weihua Zhuang. 2025. Revolutionizing QoE-Driven Network Management with Digital Agents in 6G. *IEEE Communications Magazine* (2025).

[173] Ze Sheng, Fenghua Wu, Xiangwu Zuo, Chao Li, Yuxin Qiao, and Lei Hang. 2024. Lprotector: An llm-driven vulnerability detection system. *arXiv preprint arXiv:2411.06493* (2024).

[174] Alexey Shestov, Rodion Levichev, Ravil Mussabayev, Evgeny Maslov, Anton Cheshkov, and Pavel Zadorozhny. 2024. Finetuning large language models for vulnerability detection. *arXiv preprint arXiv:2401.17010* (2024).

[175] Brian Singer, Keane Lucas, Lakshmi Adiga, Meghna Jain, Lujo Bauer, and Vyas Sekar. 2025. On the Feasibility of Using LLMs to Execute Multistage Network Attacks. *arXiv preprint arXiv:2501.16466* (2025).

[176] Muris Sladić, Veronica Valeros, Carlos Catania, and Sebastian Garcia. 2024. Llm in the shell: Generative honeypots. In *2024 IEEE European Symposium on Security and Privacy Workshops (EuroS&PW)*. IEEE, 430–435.

[177] Chengyu Song, Linru Ma, Jianming Zheng, Jinzhi Liao, Hongyu Kuang, and Lin Yang. 2024. Audit-LLM: Multi-Agent Collaboration for Log-based Insider Threat Detection. *arXiv preprint arXiv:2408.08902* (2024).

[178] Felix Specht, Jens Otto, and Jens Eickmeyer. 2022. Cyberattack impact reduction using software-defined networking for cyber-physical production systems. In *2022 IEEE 20th International Conference on Industrial Informatics (INDIN)*. IEEE, 188–194.

[179] Yuan Sun and Jorge Ortiz. 2024. GenAI-Driven Cyberattack Detection in V2X Networks for Enhanced Road Safety and Autonomous Vehicle Defense. *International Journal of Advance in Applied Science Research* 3 (2024), 67–75.

[180] Mohammed N Swileh and Shengli Zhang. 2024. Unseen Attack Detection in Software-Defined Networking Using a BERT-Based Large Language Model. *arXiv preprint arXiv:2412.06239* (2024).

[181] Kazuki Takashima, Daisuke Kotani, and Yasuo Okabe. 2024. DDoS Attack Information Sharing Among CDNs Interconnected Through CDNI. In *2024 IEEE 48th Annual Computers, Software, and Applications Conference (COMPSAC)*. IEEE, 2209–2214.

[182] Wesley Tann, Yuancheng Liu, Jun Heng Sim, Choon Meng Seah, and Ee-Chien Chang. 2023. Using large language models for cybersecurity capture-the-flag challenges and certification questions. *arXiv preprint arXiv:2308.10443* (2023).

[183] Stephen Burabari Tete. 2024. Threat modelling and risk analysis for large language model (llm)-powered applications. *arXiv preprint arXiv:2406.11007* (2024).

[184] PeiYu Tseng, ZihDwo Yeh, Xushu Dai, and Peng Liu. 2024. Using llms to automate threat intelligence analysis workflows in security operation centers. *arXiv preprint arXiv:2407.13093* (2024).

[185] Rustem Turtayev, Artem Petrov, Dmitrii Volkov, and Denis Volk. 2024. Hacking CTFs with Plain Agents. *arXiv preprint arXiv:2412.02776* (2024).

[186] Yusuf Usman, Prashnna K Gyawali, Sohan Gyawali, and Robin Chataut. 2024. The Dark Side of AI: Large Language Models as Tools for Cyber Attacks on Vehicle Systems. In *2024 IEEE 15th Annual Ubiquitous Computing, Electronics & Mobile Communication Conference (UEMCON)*. IEEE, 169–175.

[187] Christoforos Vasilatos, Dunia J Mahboobeh, Hithem Lamri, Manaar Alam, and Michail Maniatakos. 2024. Llmpot: Automated llm-based industrial protocol and physical process emulation for ics honeypots. *arXiv preprint arXiv:2405.05999* (2024).

[188] Dmitrii Volkov et al. 2024. LLM Agent Honeypot: Monitoring AI Hacking Agents in the Wild. *arXiv preprint arXiv:2410.13919* (2024).

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

[190] Lingzhi Wang, Jiahui Wang, Kyle Jung, Kedar Thiagarajan, Emily Wei, Xiangmin Shen, Yan Chen, and Zhenyuan Li. 2024. From sands to mansions: Enabling automatic full-life-cycle cyberattack construction with llm. *arXiv preprint arXiv:2407.16928* (2024).

[191] Xiangwen Wang, Jie Peng, Kaidi Xu, Huaxiu Yao, and Tianlong Chen. 2024. Reinforcement learning-driven llm agent for automated attacks on llms. In *Proceedings of the Fifth Workshop on Privacy in Natural Language Processing*. 170–177.

[192] Yunfei Wang, Shixuan Liu, Wenhao Wang, Changling Zhou, Chao Zhang, Jiandong Jin, and Cheng Zhu. 2025. A Unified Modeling Framework for Automated Penetration Testing. *arXiv preprint arXiv:2502.11588* (2025).

[193] Yuntao Wang, Zhou Su, Ning Zhang, Rui Xing, Dongxiao Liu, Tom H Luan, and Xuemin Shen. 2022. A survey on metaverse: Fundamentals, security, and privacy. *IEEE communications surveys & tutorials* 25, 1 (2022), 319–352.

[194] Yifei Wang, Dizhan Xue, Shengjie Zhang, and Shengsheng Qian. 2024. Badagent: Inserting and activating backdoor attacks in llm agents. *arXiv preprint arXiv:2406.03007* (2024).

[195] Braden K Webb, Sumit Purohit, and Rounak Meyur. 2024. Cyber knowledge completion using large language models. *arXiv preprint arXiv:2409.16176* (2024).

[196] Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Fei Xia, Ed Chi, Quoc V Le, Denny Zhou, et al. 2022. Chain-of-thought prompting elicits reasoning in large language models. *Advances in neural information processing systems* 35 (2022), 24824–24837.

[197] Benlong Wu, Guoqiang Chen, Kejiang Chen, Xiuwei Shang, Jiapeng Han, Yanru He, Weiming Zhang, and Nenghai Yu. 2024. Autopt: How far are we from the end2end automated web penetration testing? *arXiv preprint arXiv:2411.01236* (2024).

[198] Zhiheng Xi, Wenxiang Chen, Xin Guo, Wei He, Yiwen Ding, Boyang Hong, Ming Zhang, Junzhe Wang, Senjie Jin, Enyu Zhou, et al. 2025. The rise and potential of large language model based agents: A survey. *Science China Information Sciences* 68, 2 (2025), 121101.

[199] ZeKe Xiao, Qin Wang, Hammond Pearce, and Shiping Chen. 2025. Logic meets magic: Llms cracking smart contract vulnerabilities. *arXiv preprint arXiv:2501.07058* (2025).

[200] John Yang, Akshara Prabhakar, Shunyu Yao, Kexin Pei, and Karthik R Narasimhan. 2023. Language agents as hackers: Evaluating cybersecurity skills with capture the flag. In *Multi-Agent Security Workshop@ NeurIPS'23*.

[201] Kai-Cheng Yang and Filippo Menczer. 2023. Anatomy of an AI-powered malicious social botnet. *arXiv preprint arXiv:2307.16336* (2023).

[202] Shunyu Yao, Dian Yu, Jeffrey Zhao, Izhak Shafran, Tom Griffiths, Yuan Cao, and Karthik Narasimhan. 2023. Tree of thoughts: Deliberate problem solving with large language models. *Advances in neural information processing systems* 36 (2023), 11809–11822.

[203] Shunyu Yao, Jeffrey Zhao, Dian Yu, Nan Du, Izhak Shafran, Karthik Narasimhan, and Yuan Cao. 2023. React: Synergizing reasoning and acting in language models. In *International Conference on Learning Representations (ICLR)*.

[204] Abel Yeboah-Ofori and Aden Hawsh. 2023. Effects of cyberattacks on virtual reality and augmented reality technologies for people with disabilities. In *2023 IEEE international smart cities conference (ISC2)*. IEEE, 1–7.

[205] Roop Kumar Yekollu, Tejal Bhimraj Ghuge, Sammip Sunil Biradar, Shivkumar V Haldikar, and Omer Farook Mohideen Abdul Kader. 2024. Securing the Virtual Realm: Strategies for Cybersecurity in Augmented Reality (AR) and Virtual Reality (VR) Applications. In *2024 8th International Conference on I-SMAC (IoT in Social, Mobile, Analytics and Cloud)(I-SMAC)*. IEEE, 520–526.

[206] Yagmur Yigit, Mohamed Amine Ferrag, Iqbal H Sarker, Leandros A Maglaras, Christos Chrysoulas, Naghmeh Moradpoor, and Helge Janicke. 2024. Critical infrastructure protection: Generative AI, challenges, and opportunities. *arXiv preprint arXiv:2405.04874* (2024).

[207] Jingru Yu, Yi Yu, Xuhong Wang, Yilun Lin, Manzhi Yang, Yu Qiao, and Fei-Yue Wang. 2024. The Shadow of Fraud: The Emerging Danger of AI-powered Social Engineering and its Possible Cure. *arXiv preprint arXiv:2407.15912* (2024).

[208] Yao-Ching Yu, Tsun-Han Chiang, Cheng-Wei Tsai, Chien-Ming Huang, and Wen-Kwang Tsao. 2025. Primus: A Pioneering Collection of Open-Source Datasets for Cybersecurity LLM Training. *arXiv preprint arXiv:2502.11191* (2025).

[209] Zhengmin Yu, Jiutian Zeng, Siyi Chen, Wenhan Xu, Dandan Xu, Xiangyu Liu, Zonghao Ying, Nan Wang, Yuan Zhang, and Min Yang. 2024. CS-Eval: A Comprehensive Large Language Model Benchmark for CyberSecurity. *arXiv preprint arXiv:2411.16239* (2024).

[210] Tongxin Yuan, Zhiwei He, Lingzhong Dong, Yiming Wang, Ruijie Zhao, Tian Xia, Lizhen Xu, Binglin Zhou, Fangqi Li, Zhuosheng Zhang, et al. 2024. R-judge: Benchmarking safety risk awareness for llm agents. *arXiv preprint arXiv:2401.10019* (2024).

[211] Aydin Zaboli, Seong Lok Choi, Tai-Jin Song, and Junho Hong. 2024. Chatgpt and other large language models for cybersecurity of smart grid applications. In *2024 IEEE Power & Energy Society General Meeting (PESGM)*. IEEE, 1–5.

[212] Hanrong Zhang, Jingyuan Huang, Kai Mei, Yifei Yao, Zhenting Wang, Chenlu Zhan, Hongwei Wang, and Yongfeng Zhang. 2024. Agent security bench (asb): Formalizing and benchmarking attacks and defenses in llm-based agents. *arXiv preprint arXiv:2410.02644* (2024).

[213] Han Zhang, Akram Bin Sediq, Ali Afana, and Melike Erol-Kantarci. 2024. Large language models in wireless application design: In-context learning-enhanced automatic network intrusion detection. *arXiv preprint arXiv:2405.11002* (2024).

[214] Jie Zhang, Haoyu Bu, Hui Wen, Yongji Liu, Haiqiang Fei, Rongrong Xi, Lun Li, Yun Yang, Hongsong Zhu, and Dan Meng. 2025. When llms meet cybersecurity: A systematic literature review. *Cybersecurity* 8, 1 (2025), 1–41.

[215] Yongheng Zhang, Tingwen Du, Yunshan Ma, Xiang Wang, Yi Xie, Guozheng Yang, Yuliang Lu, and Ee-Chien Chang. 2024. AttacKG+: Boosting attack knowledge graph construction with large language models. *arXiv preprint arXiv:2405.04753* (2024).

[216] Ying Zhang, Xiaoyan Zhou, Hui Wen, Wenjia Niu, Jiqiang Liu, Haining Wang, and Qiang Li. 2024. Tactics, Techniques, and Procedures (TTPs) in Interpreted Malware: A Zero-Shot Generation with Large Language Models. *arXiv preprint arXiv:2407.08532* (2024).

[217] Zhenyong Zhang, Mengxiang Liu, Mingyang Sun, Ruilong Deng, Peng Cheng, Dusit Niyato, Mo-Yuen Chow, and Jiming Chen. 2024. Vulnerability of machine learning approaches applied in iot-based smart grid: A review. *IEEE Internet of Things Journal* 11, 11 (2024), 18951–18975.

[218] Wenxiang Zhao, Juntao Wu, and Zhaoyi Meng. 2025. Apppoet: Large language model based android malware detection via multi-view prompt engineering. *Expert Systems with Applications* 262 (2025), 125546.

[219] Lianmin Zheng, Wei-Lin Chiang, Ying Sheng, Siyuan Zhuang, Zhanghao Wu, Yonghao Zhuang, Zi Lin, Zhuohan Li, Dacheng Li, Eric Xing, et al. 2023. Judging llm-as-a-judge with mt-bench and chatbot arena. *Advances in Neural Information Processing Systems* 36 (2023), 46595–46623.

[220] Tianming Zheng, Ming Liu, Deepak Puthal, Ping Yi, Yue Wu, and Xiangjian He. 2022. Smart grid: Cyber attacks, critical defense approaches, and digital twin. *arXiv preprint arXiv:2205.11783* (2022).

[221] Xin Zhou, Sicong Cao, Xiaobing Sun, and David Lo. 2024. Large language model for vulnerability detection and repair: Literature review and the road ahead. *ACM Transactions on Software Engineering and Methodology* (2024).

[222] Yuxuan Zhu, Antony Kellermann, Akul Gupta, Philip Li, Richard Fang, Rohan Bindu, and Daniel Kang. 2024. Teams of llm agents can exploit zero-day vulnerabilities. *arXiv preprint arXiv:2406.01637* (2024).