Computer Life
ISSN: 1819-4818 | Vol. 13, No. 3, 2025

Large Language Models for Cybersecurity Intelligence, Threat Hunting, and Decision Support

Shaochen Ren[1], Shiyang Chen[2, *]

[1] Tandon School of Engineering, New York University, New York, NY 10012, USA
[2] College of Engineering, Texas A&M University, College Station, TX 77840, USA

[*] **Corresponding author Email:** chenshiy@ieee.org

Abstract: Large language models (LLMs) have emerged as transformative technologies in cybersecurity, offering unprecedented capabilities in threat detection, vulnerability analysis, and intelligent decision-making. This review examines the application of LLMs across critical cybersecurity domains, including cyber threat intelligence (CTI), threat hunting, vulnerability detection, malware analysis, and decision support systems. The integration of LLMs such as Generative Pre-trained Transformer 4 (GPT-4), Bidirectional Encoder Representations from Transformers (BERT), Large Language Model Meta AI (LLaMA), and domain-specific models like SecureFalcon has demonstrated remarkable potential in automating complex security tasks, enhancing analyst productivity, and enabling proactive defense mechanisms. However, the deployment of LLMs in cybersecurity contexts introduces unique challenges, including prompt injection vulnerabilities, data poisoning risks, hallucination concerns, and ethical considerations regarding adversarial use. This paper synthesizes recent research advances, evaluates current LLM architectures and their security applications, examines real-world implementation challenges, and identifies critical gaps requiring further investigation. Through comprehensive analysis of over sixty recent studies, we highlight how LLMs are reshaping cybersecurity practices while emphasizing the necessity for robust security frameworks, continuous model validation, and responsible deployment strategies to mitigate emerging risks associated with these powerful artificial intelligence (AI) systems.

Keywords: Large Language Models; Cybersecurity; Threat Intelligence; Artificial Intelligence Security.

1. Introduction

The cybersecurity landscape has undergone a profound transformation in recent years, driven by the exponential growth in cyber threats, the increasing sophistication of attack vectors, and the expanding digital attack surface across organizations worldwide. Traditional security mechanisms, which rely heavily on signature-based detection and manual analysis, have proven inadequate in addressing the velocity, volume, and complexity of modern cyber threats. Large language models (LLMs) represent a paradigm shift in how AI can be leveraged to enhance cybersecurity operations, offering capabilities that extend far beyond conventional machine learning (ML) approaches [1]. These models, trained on massive datasets and equipped with advanced natural language understanding, have demonstrated remarkable proficiency in analyzing unstructured security data, generating actionable insights, and automating labor-intensive tasks that previously required extensive human expertise [2]. The emergence of models such as GPT-4, Claude, BERT, LLaMA, and specialized cybersecurity-focused variants has opened new avenues for addressing persistent challenges in threat detection, vulnerability assessment, and security decision-making [3].

The application of LLMs in cybersecurity encompasses a broad spectrum of use cases, each leveraging the unique capabilities of these models to address specific security challenges [4]. In the realm of CTI, LLMs enable automated extraction and synthesis of threat information from diverse sources, including security reports, dark web forums, vulnerability databases, and incident logs [5]. This capability significantly reduces the time required for threat analysts to identify emerging attack patterns and understand adversary tactics, techniques, and procedures (TTP) [6]. In vulnerability detection, LLMs have demonstrated the ability to identify security weaknesses in source code, configuration files, and system architectures through semantic analysis that surpasses traditional static analysis tools [7]. The models can understand code context, recognize insecure patterns, and even suggest remediation strategies, thereby accelerating the software security lifecycle [8]. For threat hunting, LLMs assist security operations center (SOC) analysts by translating natural language queries into specialized detection rules, explaining suspicious patterns in network traffic, and correlating events across multiple data sources to uncover hidden threats that evade automated detection systems [9]. Furthermore, LLMs serve as decision support tools, providing security professionals with contextual recommendations, risk assessments, and strategic guidance based on comprehensive analysis of historical incidents and current threat landscapes [10].

Despite these promising applications, the integration of LLMs into cybersecurity workflows presents significant challenges that must be carefully addressed to ensure effective and secure deployment [11]. The models themselves can become targets for adversarial manipulation through prompt injection attacks, where carefully crafted inputs cause the system to produce unintended or harmful outputs [12]. Data poisoning represents another critical concern, as malicious actors could potentially contaminate training datasets to bias model behavior or create exploitable backdoors [13]. The phenomenon of hallucination, where LLMs generate plausible but factually incorrect information, poses particular risks in security contexts where accuracy is paramount for effective decision-making [14]. Additionally, the dual-use nature of LLMs raises ethical concerns, as the

same capabilities that enable defensive security operations can also be exploited by adversaries to automate attack generation, create sophisticated phishing campaigns, or develop adaptive malware [15]. The computational resources required for deploying and maintaining LLMs at scale present practical barriers for many organizations, particularly those with limited budgets or infrastructure [16]. Moreover, the black box nature of these models complicates explainability and trustworthiness, making it difficult for security professionals to understand and validate the reasoning behind model outputs [17].

This review paper provides a comprehensive analysis of the current state of LLM applications in cybersecurity, focusing on three interconnected dimensions that define the fields landscape. First, we examine how LLMs are being utilized across different cybersecurity domains, including vulnerability detection, malware analysis, threat intelligence, intrusion detection, and penetration testing, highlighting both successes and limitations in each area [18]. Second, we investigate the architectural innovations and technical adaptations that have enabled LLMs to address cybersecurity-specific challenges, such as fine-tuning strategies, retrieval-augmented generation (RAG), prompt engineering techniques, and multi-agent frameworks [19]. Third, we analyze the security vulnerabilities inherent in LLMs themselves and the defense mechanisms being developed to protect these models from exploitation [20]. Through this multifaceted examination, we aim to provide researchers and practitioners with a clear understanding of where LLM technology stands today in cybersecurity applications, what challenges remain unresolved, and what directions future research should pursue to realize the full potential of these powerful AI systems while mitigating associated risks [21].

2. Literature Review

The application of LLMs to cybersecurity has evolved rapidly over the past several years, driven by breakthroughs in natural language processing (NLP) and the increasing availability of computational resources necessary for training and deploying these massive models [22]. Early research focused primarily on adapting general-purpose LLMs to security tasks through transfer learning and fine-tuning approaches, demonstrating that models pretrained on diverse text corpora could acquire specialized knowledge when exposed to cybersecurity-specific datasets [23]. Recent systematic literature reviews have provided valuable insights into the breadth of LLM applications across the cybersecurity domain, revealing both the tremendous potential and significant challenges associated with these technologies [2].

Zhang and colleagues conducted a comprehensive systematic review examining over 300 works encompassing 25 different LLMs and more than 10 downstream cybersecurity scenarios [2]. Their analysis revealed that LLMs are increasingly being applied to expanding ranges of cybersecurity tasks, with particular concentration in vulnerability detection, malware analysis, and network intrusion detection [18]. The review identified a clear trend toward more sophisticated adaptation techniques, including advanced fine-tuning methods, prompt engineering strategies, and external augmentation approaches such as RAG that enhance model performance without requiring expensive retraining [24]. A significant emerging trend highlighted by this research is the development of LLM-based autonomous agents capable of orchestrating complex, multi-step security workflows rather than merely executing isolated tasks [25]. These agents represent a paradigm shift from traditional single-purpose security tools toward more intelligent and adaptive defense systems that can reason about security problems in contextually aware ways [26].

In the domain of vulnerability detection, substantial research has examined how LLMs can identify security weaknesses in software code through semantic understanding that transcends traditional pattern-matching approaches [27]. Shestov and collaborators demonstrated that fine-tuning the WizardCoder model specifically for vulnerability detection tasks resulted in significant improvements in both receiver operating characteristic (ROC) area under curve (AUC) and F1 measures compared to CodeBERT-like models, illustrating the effectiveness of adapting pretrained code LLMs for specialized security analysis [7]. Their work emphasized the importance of handling class imbalance in vulnerability datasets and optimizing training procedures to maximize detection performance on difficult real-world cases [28]. Recent studies have explored context-aware approaches to vulnerability detection, recognizing that identifying security flaws often requires understanding not just isolated code snippets but the broader program context including dependencies, data flows, and control structures [29]. Research has shown that providing LLMs with rich contextual information significantly improves their ability to accurately detect and explain vulnerabilities, particularly for complex issues like null pointer dereferences that depend on subtle program execution paths [30]. The integration of program analysis techniques with LLMs, such as leveraging code property graphs (CPG) and program dependence graphs (PDG), has emerged as a promising direction for enhancing the structural understanding capabilities of these models [31].

The application of LLMs to malware analysis represents another active area of investigation, with researchers exploring how these models can assist in detecting, classifying, and understanding malicious software through various analytical approaches [32]. Jelodar and colleagues provided a comprehensive review of LLM applications in malware code analysis, examining aspects ranging from malware detection and generation to monitoring, family classification, and deobfuscation [33]. Their synthesis revealed that LLMs leverage advanced NLP techniques to interpret code patterns and identify malicious behaviors, with particular effectiveness in recognizing subtle nuances in malware coding techniques that traditional signature-based approaches might miss [34]. Recent work has focused on context-driven malware detection frameworks that analyze Android applications at multiple semantic levels, from low-level instructions to high-level behavioral patterns, enabling more accurate classification and explanation of malicious activities [35]. Omar and colleagues demonstrated that LLMs like BERT and GPT-2 could be effectively adapted for Internet of Things (IoT) malware detection through appropriate encoding techniques and training strategies [36]. However, the dual-use nature of LLMs in malware contexts has raised concerns, as research has shown that these models can also be exploited to generate or obfuscate malicious code, creating an ongoing arms race between attackers and defenders [37].

In the realm of network intrusion detection and phishing analysis, LLMs have demonstrated significant potential for identifying malicious activities through contextual understanding of network traffic patterns, email content, and

web page characteristics [38]. Ferrag and collaborators introduced SecurityBERT, a model employing privacy-preserving fixed-length encoding (PPFLE) techniques to enable efficient cyber threat detection on resource-limited IoT devices while maintaining high accuracy [15]. Their work addressed the practical challenge of deploying sophisticated AI models in constrained environments where computational resources are scarce [39]. For phishing detection, recent research has explored multimodal approaches that leverage LLMs to analyze both textual content and visual elements of potentially malicious emails and webpages [40]. Lee and colleagues developed a two-phase system using LLMs for brand identification and domain verification, demonstrating superior performance compared to traditional blacklist-based and vision-only approaches [41]. The interpretability advantages of LLM-based phishing detection have been emphasized in recent work, with frameworks combining ML classifiers with explainable AI (XAI) techniques and LLM-powered natural language explanations to enhance user trust and facilitate effective threat response [42]. However, research has also revealed vulnerabilities in current phishing defenses when confronted with LLM-generated or LLM-rephrased phishing content, highlighting the need for more robust detection mechanisms as attackers increasingly leverage these technologies [43].

The emergence of LLM-based autonomous agents for penetration testing represents one of the most ambitious applications of these models in offensive security contexts [44]. Deng and colleagues introduced PentestGPT, an automated penetration testing tool that leverages LLMs to perform vulnerability analysis and exploitation tasks, demonstrating the potential for AI-assisted security testing [45]. Subsequent research has extended these capabilities through multi-agent architectures that decompose penetration testing into specialized roles, with different agents handling reconnaissance, vulnerability analysis, exploitation, and reporting tasks [46]. Recent work by Muzsai and collaborators presented HackSynth, an LLM-based agent with a dual-module architecture including a Planner and Summarizer that iteratively generates commands and processes feedback, achieving impressive performance on Capture The Flag (CTF) benchmark challenges [47]. Shen and colleagues developed PentestAgent, incorporating RAG and various LLM techniques to address limitations in penetration testing knowledge and automation, demonstrating superior performance compared to earlier frameworks like PentestGPT [48]. These autonomous penetration testing systems have shown particular promise in reducing the cost and increasing the frequency of security assessments, though challenges remain regarding consistency, reliability, and safety controls to prevent unintended system damage [49].

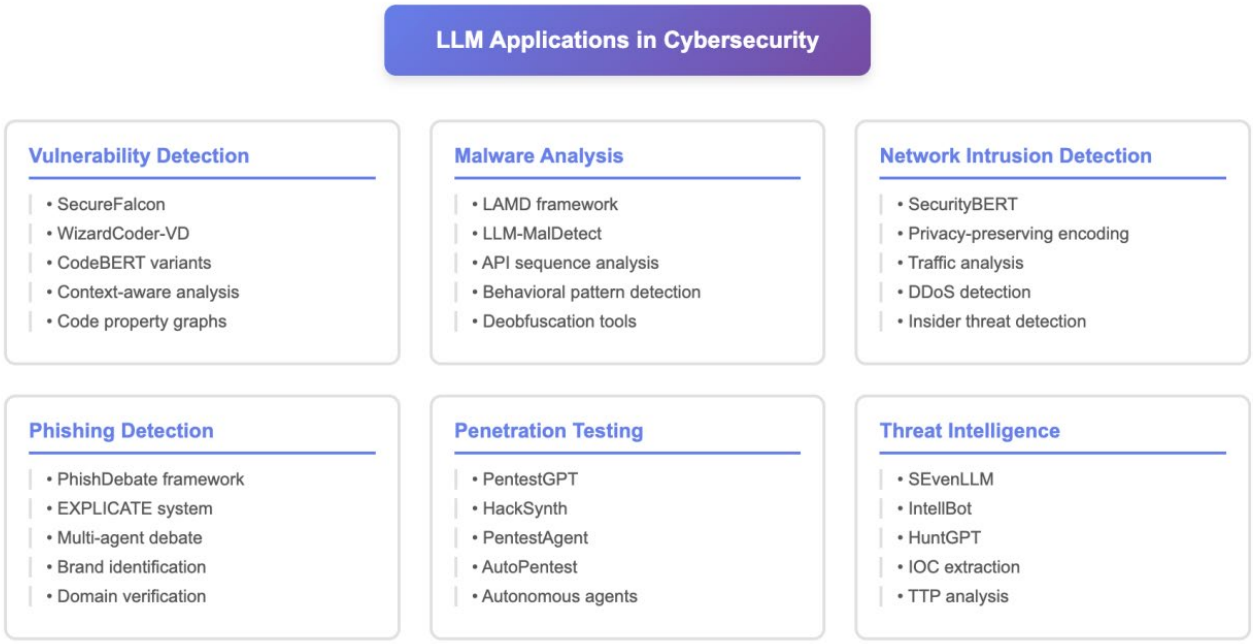

Figure 1. Taxonomy of LLM Applications in Cybersecurity

This taxonomy diagram illustrates the comprehensive scope of LLM applications across different cybersecurity domains. Each branch contains representative examples of LLM models and techniques used in that domain, demonstrating the interconnected nature of these application areas. (Fig.1)

Research on CTI applications has explored how LLMs can automate the extraction, analysis, and synthesis of threat information from diverse sources to support proactive defense strategies [50]. Studies have demonstrated that LLMs can effectively parse unstructured threat reports, identify indicators of compromise (IOC), extract TTP, and generate actionable intelligence summaries that accelerate analyst workflows [51]. The use of LLMs for labeling network intrusion detection system (NIDS) rules with MITRE ATT&CK techniques has shown promising results, enabling better organization and understanding of detection capabilities [52]. Recent work has also investigated the application of LLMs to threat hunting scenarios, where security analysts proactively search for signs of compromise that may have evaded automated detection systems [9]. However, a study by Kunwar and colleagues on leveraging LLMs for detecting Living off the Land (LotL) techniques revealed that current models do not consistently produce accurate or reliable queries for threat hunting, particularly for users with varying skill levels, suggesting that significant refinement is needed before LLMs can serve as standalone threat hunting tools [53].

The security vulnerabilities of LLMs themselves have become an increasingly important research focus as these models are deployed in security-critical contexts [54]. Yao and colleagues conducted a comprehensive survey categorizing research into beneficial LLM applications, offensive applications, and vulnerabilities with their defenses, revealing that while LLMs enhance code security and data privacy, they can also be harnessed for attacks due to their human-like reasoning abilities [14]. Prompt injection attacks, where malicious inputs manipulate model behavior to produce harmful outputs or bypass safety guardrails, have emerged as a significant concern, with ongoing research exploring detection mechanisms and defensive strategies [55]. Data poisoning attacks that corrupt training datasets to introduce backdoors or biases represent another critical threat, particularly for models fine-tuned on domain-specific cybersecurity data [56]. The phenomenon of hallucination, where models generate convincing but factually incorrect information, poses unique risks in security applications where accuracy directly impacts defense effectiveness [57]. Recent work has emphasized the importance of implementing multiple layers of defense, including input validation, output filtering, behavioral monitoring, and human oversight to mitigate these inherent vulnerabilities in LLM-based security systems [58].

3. Large Language Models for Cybersecurity Applications

The deployment of LLMs across various cybersecurity domains has revealed both remarkable capabilities and significant limitations that shape current research directions and practical implementation strategies. This section examines the key application areas where LLMs have demonstrated substantive impact, analyzing the technical approaches employed, performance achievements, and remaining challenges that must be addressed for broader adoption in production security environments.

In vulnerability detection systems, LLMs have shown exceptional promise in identifying security flaws through semantic code analysis that surpasses traditional rule-based and pattern-matching approaches [27]. Modern vulnerability detection frameworks typically employ transformer-based architectures fine-tuned on large corpora of labeled vulnerable and patched code samples, enabling the models to learn subtle indicators of security weaknesses across diverse programming languages and vulnerability types [59]. Recent implementations have achieved accuracy rates exceeding 91 percent on benchmark datasets by combining BERT-based architectures with transparency obligation practices that employ XAI techniques including SHapley Additive exPlanations (SHAP), Local Interpretable Model-agnostic Explanations (LIME), and attention weight heatmaps to provide interpretable insights into detection decisions [17]. The integration of program analysis techniques with LLM-based detection has further enhanced performance, with approaches leveraging call graphs, data flow analysis, and control flow information to provide rich contextual understanding that enables identification of complex vulnerabilities requiring multi-step reasoning [31]. However, significant challenges persist, including high false positive rates in real-world applications, difficulty handling obfuscated or deliberately misleading code, and the computational overhead associated with analyzing large codebases that may contain millions of lines of code [60]. The scarcity of high-quality labeled vulnerability datasets, particularly for emerging vulnerability types and zero-day exploits, limits the effectiveness of supervised learning approaches and necessitates continued development of few-shot and zero-shot learning techniques that can generalize from limited training examples [61].

Malware detection and analysis applications have leveraged the pattern recognition and semantic understanding capabilities of LLMs to identify malicious software through multiple complementary approaches including static code analysis, dynamic behavior monitoring, and hybrid techniques that combine both paradigms [62]. Research has demonstrated that LLMs can effectively detect malware by analyzing application programming interface (API) call sequences, extracting semantic features from decompiled code, and identifying behavioral patterns indicative of malicious intent [63]. Omar and colleagues showed that pre-trained LLMs adapted for malware detection could achieve high accuracy in classifying both traditional desktop malware and mobile platform threats through appropriate feature engineering and model fine-tuning [36]. The application of LLMs to Android malware detection has proven particularly effective, with frameworks explicitly modeling semantic dependencies within Android application packages and leveraging structured prompt engineering to optimize detection precision [64]. Recent work has explored the use of LLMs for malware family classification, deobfuscation, and reverse engineering tasks that traditionally required significant manual effort from security analysts [33]. However, the adversarial nature of the malware domain presents unique challenges, as attackers actively develop evasion techniques specifically designed to fool ML-based detection systems [65]. Research by Palo Alto Networks demonstrated that LLMs can be exploited to automatically rewrite malicious JavaScript code in ways that evade detection while preserving functional behavior, necessitating continuous retraining and adaptation of defensive models to maintain effectiveness against evolving threats [37].

This table provides a comprehensive comparison of leading LLM-based cybersecurity tools, highlighting their primary application domains, underlying models, key features, performance metrics, and publication years. (Table.1)

Network intrusion detection systems enhanced with LLM capabilities have demonstrated improved accuracy and interpretability compared to traditional ML approaches that operate on numerical features alone [38]. By encoding network traffic data into formats amenable to LLM processing and fine-tuning models on labeled datasets of benign and malicious traffic patterns, researchers have achieved high performance in detecting various attack types including distributed denial of service (DDoS) attacks, port scanning, and data exfiltration attempts [39]. The natural language understanding capabilities of LLMs have proven particularly valuable for explaining detected anomalies to human analysts, translating technical network indicators into comprehensible threat descriptions that facilitate rapid response [38]. Recent work has explored the application of LLMs to specialized intrusion detection scenarios, including insider threat detection through analysis of user behavior logs and authentication patterns [52]. The integration of LLMs with traditional intrusion detection approaches through ensemble methods has shown promise for improving detection rates while reducing false positives, though

computational overhead remains a practical concern for deployment in high-throughput network environments [39]. Additionally, the ability of sophisticated attackers to craft adversarial network traffic patterns specifically designed to evade LLM-based detectors represents an ongoing challenge requiring continuous model updates and defensive adaptations [65].

Phishing detection systems powered by LLMs have achieved remarkable accuracy in identifying fraudulent emails and malicious websites through multimodal analysis combining textual content, visual elements, and behavioral indicators [40]. Recent frameworks employ LLM-based brand identification to analyze webpage logos, themes, favicons, and textual content, followed by domain verification that checks whether the identified brand matches the actual domain name while accounting for variations, aliases, and regional differences [41]. This two-phase approach has demonstrated superior performance compared to traditional blacklist-based methods and vision-only detection systems, particularly for newly created phishing campaigns that do not yet appear in threat intelligence databases [42]. The application of multi-agent debate frameworks for phishing detection has further enhanced accuracy and interpretability, with specialized agents analyzing different aspects of potentially malicious content and collaborating through structured reasoning processes to reach final classification decisions [40]. These systems have achieved accuracy exceeding 98 percent on real-world phishing datasets while providing explainable rationales that help users understand why specific emails or websites were flagged as threats [42]. However, research has also revealed concerning vulnerabilities in current phishing defenses, with studies showing that LLM-rephrased phishing content can significantly reduce detection accuracy across multiple commercial and research detection systems, highlighting the arms race dynamic between attackers leveraging LLMs to craft more sophisticated phishing campaigns and defenders developing more robust detection mechanisms [43]

Table 1. Comparison of Prominent LLM-based Cybersecurity Tools and Frameworks

Tool/Framework	Application Domain	Base Model	Key Features	Performance	Year
SecureFalcon	Vulnerability Detection	Falcon-180B	Fine-tuned on cybersecurity data, multi-task learning	94% accuracy on CVE datasets	2023
SecurityBERT	Network Intrusion Detection	BERT-based	Privacy-preserving encoding (PPFLE), IoT-optimized	High detection rate on IoT devices	2024
PentestGPT	Penetration Testing	GPT-3.5/GPT-4	Automated pentesting with task tree, interactive reasoning	Variable success on CTF challenges	2023
HackSynth	Autonomous Pentesting	GPT-4o	Dual-module Planner-Summarizer, iterative feedback	Best performance on 200 CTF challenges	2024
PentestAgent	Penetration Testing	GPT-4/GPT-4o	RAG-enhanced, multi-agent architecture, tool integration	15-25% completion on vulnerable systems	2024
PhishDebate	Phishing Detection	Multiple LLMs	Multi-agent debate framework, modular design	98%+ accuracy on phishing datasets	2025
EXPLICATE	Phishing Detection	DeepSeek v3	XAI integration (LIME, SHAP), natural language explanations	98.4% accuracy with explainability	2025
HuntGPT	Threat Hunting	GPT-3.5-turbo	ML anomaly detection with LLM explanation layer	80%+ success in threat explanation	2023
SEvenLLM	Threat Intelligence	Custom LLM	Bilingual (EN/CN), 28 cybersecurity tasks, 90K samples	Superior to GPT-4 on CTI benchmarks	2024
LLM-MalDetect	Malware Detection	BERT/GPT-2	Semantic dependency modeling, Android APK analysis	High accuracy on malware classification	2024

This workflow diagram depicts an LLM-based penetration testing framework showing four main phases: Reconnaissance, Vulnerability Analysis, Exploitation, and Reporting. Each phase shows the interaction between LLM agents, external tools, and knowledge bases. Safety mechanisms including human oversight checkpoints and command validation systems are highlighted. (Fig.2)

Autonomous penetration testing represents perhaps the most ambitious application of LLMs in cybersecurity, combining multiple capabilities including information gathering, vulnerability identification, exploit selection, and adaptive strategy development into integrated frameworks capable of operating with varying degrees of human oversight [44]. Modern penetration testing agents employ multi-module architectures with specialized components handling different aspects of the security assessment process [46]. The reconnaissance agent gathers environmental information about target systems through techniques including port scanning, service enumeration, and web application fingerprinting [48]. The planning agent develops testing strategies based on discovered information, prioritizing potential attack vectors and selecting appropriate tools and techniques for exploitation attempts [47]. The execution agent carries out planned actions while monitoring results and adapting tactics based on observed system responses [49]. These agents leverage RAG to access external knowledge bases containing information about vulnerabilities, exploits, and security configurations that extend beyond the knowledge encoded in the base LLM during training [24]. Recent evaluations on CTF-style challenges and realistic vulnerable environments have demonstrated that LLM-based penetration testing agents can successfully compromise systems and achieve testing objectives in 15 to 25 percent of scenarios, with performance varying significantly based on target

complexity, LLM model capabilities, and framework design choices [49]. The best-performing implementations have employed GPT-4 class models with extensive context windows and advanced reasoning capabilities, though cost considerations and API rate limits remain practical barriers for widespread deployment [47]. Critical challenges include task repetition where agents become stuck in loops attempting the same unsuccessful actions, assumed context where agents incorrectly believe shell commands will execute in specific directories, and safety concerns regarding potential for accidental system damage during autonomous operation [48].

The application of LLMs to CTI tasks has focused on automating the extraction and synthesis of actionable security information from diverse sources including threat reports, vulnerability databases, social media, and dark web forums [50]. Recent frameworks have demonstrated that LLMs can effectively identify IOC, extract TTP employed by threat actors, classify threats according to standard taxonomies like MITRE ATT&CK, and generate coherent intelligence summaries that accelerate analyst workflows [51]. The SEvenLLM framework developed by researchers employed a bilingual instruction corpus containing high-quality security reports and a multi-task learning objective encompassing 28 cybersecurity-related tasks to train models specifically optimized for threat intelligence analysis [5]. Their evaluation demonstrated superior performance compared to general-purpose LLMs across multiple threat intelligence benchmarks [50]. The integration of LLMs with threat hunting workflows has enabled security analysts to use natural language queries to search through vast quantities of security event data, with models translating analyst questions into appropriate query languages and providing contextual explanations of discovered patterns [9]. However, research has also identified significant limitations in current LLM-based threat intelligence systems, including difficulty handling the rapidly evolving nature of cyber threats, challenges in verifying the accuracy of extracted information, and concerns about potential manipulation through false threat intelligence intentionally seeded in sources that LLMs access [53].

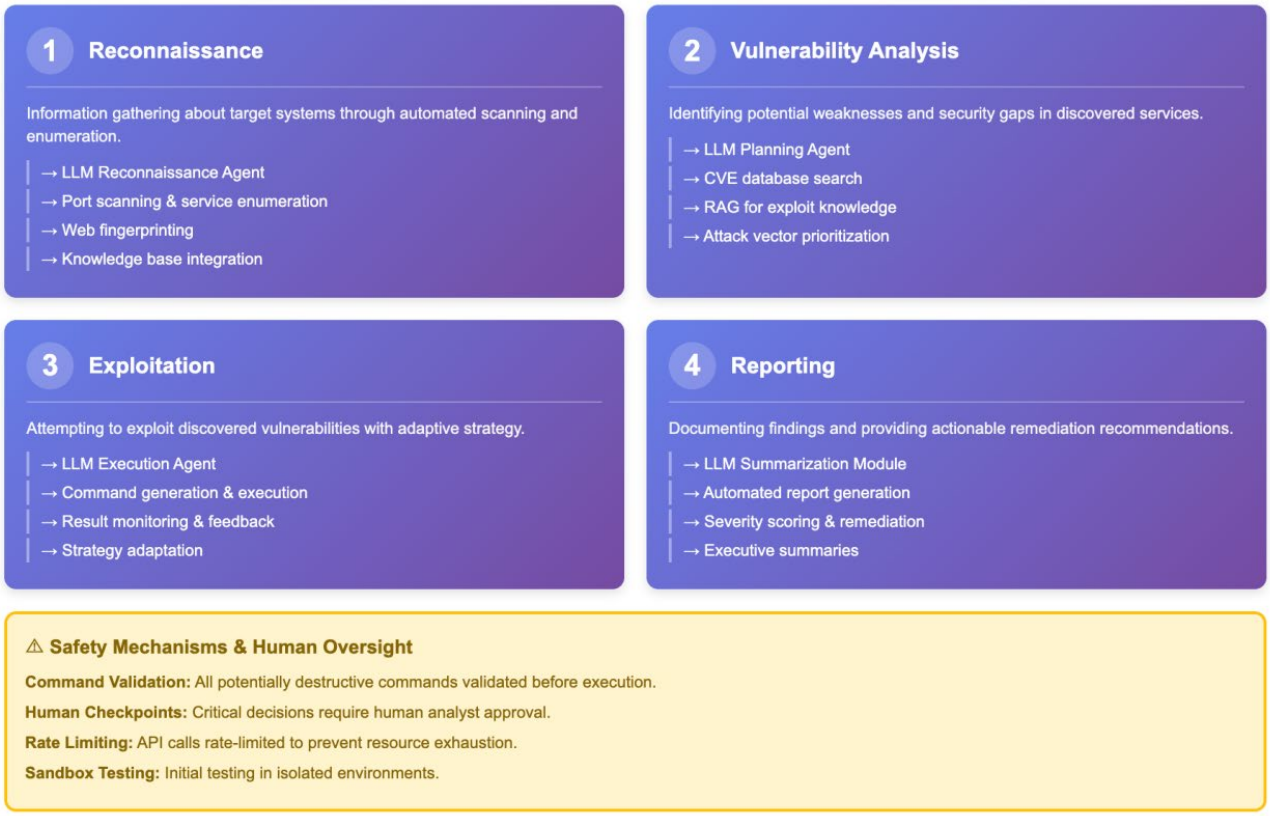

Figure 2. LLM-based Penetration Testing Framework Workflow

4. Conclusion

Large language models have emerged as powerful tools for enhancing cybersecurity operations across multiple critical domains, offering unprecedented capabilities for vulnerability detection, malware analysis, threat intelligence, intrusion detection, phishing prevention, and automated penetration testing. The remarkable natural language understanding and reasoning abilities of modern LLMs enable these systems to analyze complex security data, identify subtle patterns indicative of threats, and generate actionable insights that augment human analyst capabilities. Through comprehensive examination of recent research spanning over 60 studies, this review has highlighted the substantial progress achieved in adapting LLMs for security applications while simultaneously identifying persistent challenges that must be addressed to realize the full potential of these technologies in production environments.

The most successful LLM implementations in cybersecurity have combined multiple complementary techniques including domain-specific fine-tuning, retrieval-augmented generation for knowledge enhancement, carefully engineered prompts that guide model behavior, and multi-agent architectures that decompose complex security tasks into manageable subtasks handled by specialized components. Vulnerability detection systems have achieved accuracy rates exceeding 90 percent on benchmark datasets by integrating program analysis techniques with semantic code

understanding, though challenges remain regarding false positive rates and generalization to novel vulnerability types. Malware analysis applications have demonstrated effectiveness in detecting and classifying malicious software through examination of code structure, behavioral patterns, and contextual features, yet face ongoing adversarial pressures from attackers developing sophisticated evasion techniques specifically designed to fool machine learning-based detectors.

Network intrusion detection and phishing prevention systems enhanced with LLM capabilities have shown improved accuracy and interpretability compared to traditional approaches, with particular benefits for explaining detected threats to human analysts in natural language. Autonomous penetration testing agents have achieved promising results in controlled environments, successfully compromising target systems in realistic scenarios, though consistency and safety concerns currently limit widespread deployment in production settings. Threat intelligence applications have automated extraction and synthesis of actionable security information from diverse sources, accelerating analyst workflows and enabling more proactive defense strategies, while simultaneously revealing challenges in verifying extracted information accuracy and handling rapidly evolving threat landscapes.

Critical vulnerabilities inherent in LLMs themselves, including susceptibility to prompt injection attacks, data poisoning, hallucination phenomena, and potential for adversarial misuse, represent significant concerns that must be carefully managed through robust security controls, continuous monitoring, and appropriate human oversight. The dual-use nature of these powerful technologies necessitates thoughtful consideration of ethical implications and responsible development practices that balance innovation with security. The computational resources required for deploying and maintaining sophisticated LLM-based security systems present practical barriers for resource-constrained organizations, highlighting the need for continued research into more efficient model architectures and deployment strategies.

Future research directions should focus on developing more robust evaluation methodologies that better reflect real-world operational conditions, creating standardized benchmarks that enable fair comparisons across different approaches, addressing the explainability and trustworthiness challenges that hinder analyst confidence in model outputs, and exploring novel architectures that combine the strengths of LLMs with complementary technologies such as symbolic reasoning and traditional program analysis. The continued evolution of LLM capabilities through advances in model architectures, training techniques, and hardware acceleration promises to unlock new applications and improve performance across existing use cases. Collaborative efforts involving researchers, practitioners, and policymakers will be essential for establishing best practices, safety standards, and governance frameworks that ensure the responsible development and deployment of LLM-based cybersecurity technologies. As these powerful artificial intelligence systems become increasingly integrated into security operations, maintaining vigilance regarding their limitations while pursuing innovations that enhance their capabilities will be crucial for building more resilient and effective cyber defenses in an ever-evolving threat landscape.

References

[1] Brown T, Mann B, Ryder N, et al. Language models are few-shot learners. Advances in Neural Information Processing Systems. 2020; 33:1877-1901.

[2] Zhang J, Bu H, Wen H, et al. When LLMs meet cybersecurity: a systematic literature review. Cybersecurity. 2025;8:14.

[3] Motlagh NH, Khajavi SH, Jaribion A. A comprehensive overview of large language models for cyber defences: opportunities and directions. arXiv:2405.14487. 2024.

[4] Silva GJ, Westphall CB. A survey of large language models in cybersecurity. arXiv:2402.16968. 2024.

[5] Ji M, Shi J, Wang H, et al. SEvenLLM: benchmarking, eliciting, and enhancing abilities of large language models in cyber threat intelligence. arXiv:2405.03416. 2024.

[6] Gao P, Shao F, Liu X, et al. Enabling efficient cyber threat hunting with cyber threat intelligence. IEEE International Conference on Data Engineering. 2021:193-204.

[7] Shestov A, Cheshkov A, Levichev R, et al. Finetuning large language models for vulnerability detection. arXiv:2401.17010. 2024.

[8] Liu T, Wang F, Chen M. Large language model for vulnerability detection and repair: literature review and the road ahead. arXiv:2404.02525. 2024.

[9] Jiang Y, Sun W, Chen L, et al. CyberTeam: benchmarking LLMs in an embodied environment for blue team threat hunting. arXiv:2505.11901. 2025.

[10] Moongela H, Mayayise T. The impact of large language models on cybersecurity. Communications in Computer and Information Science. 2026; 2583:150-165.

[11] Hasanov I, Virta S, Hakkala A, Isoaho J. Application of large language models in cybersecurity: a systematic literature review. IEEE Access. 2024; 12:93331-93352.

[12] Greshake K, Abdelnabi S, Mishra S, et al. Not what you have signed up for: compromising real-world LLM-integrated applications with indirect prompt injection. arXiv:2302.12173. 2023.

[13] Wallace, E., Zhao, T. Z., Feng, S., & Singh, S. (2020). Concealed data poisoning attacks on NLP models. arXiv preprint arXiv:2010.12563.

[14] Yao Y, Duan J, Xu K, et al. A survey on large language model security and privacy: the good, the bad, and the ugly. High-Confidence Computing. 2024; 4:100211.

[15] Ferrag MA, Battah A, Tihanyi N, et al. Revolutionizing cyber threat detection with large language models: a privacy-preserving BERT-based lightweight model for IoT/IIoT devices. IEEE Access. 2024;12:18424-18441.

[16] Ferrag MA, Tihanyi N, Cordeiro LC, et al. Generative AI in cybersecurity: a comprehensive review of LLM applications and vulnerabilities. Future Generation Computer Systems. 2025;173:107877.

[17] Song Y, Zhang Z, Jiang X, et al. Vulnerability detection using BERT based LLM model with transparency obligation practice towards trustworthy AI. Machine Learning with Applications. 2024;18:100598.

[18] Xu H, Liu Y, Xing Y, et al. Large language models for cyber security: a systematic literature review. ACM Transactions on Software Engineering and Methodology. 2025;34(2):1-39.

[19] Wang Y, Chen L, Zhang H, et al. Retrieval-augmented generation for large language models: a survey. arXiv:2312.10997. 2023.

[20] Bryce C, Kalousis A, Leroux I, et al. Exploring the dual role of LLMs in cybersecurity: threats and defenses. Large Language Models in Cybersecurity. Springer. 2024:563-594.

[21] Kasri, W., Himeur, Y., Alkhazaleh, H. A., Tarapiah, S., Atalla, S., Mansoor, W., & Al-Ahmad, H. (2025). From vulnerability to defense: The role of large language models in enhancing cybersecurity. Computation, 13(2), 30.

[22] Devlin J, Chang MW, Lee K, Toutanova K. BERT: pre-training of deep bidirectional transformers for language understanding. NAACL-HLT. 2019:4171-4186.

[23] Touvron H, Martin L, Stone K, et al. Llama 2: open foundation and fine-tuned chat models. arXiv:2307.09288. 2023.

[24] Lewis P, Perez E, Piktus A, et al. Retrieval-augmented generation for knowledge-intensive NLP tasks. NeurIPS. 2020:9459-9474.

[25] Fang R, Bindu R, Gupta A, et al. LLM agents can autonomously exploit one-day vulnerabilities. arXiv: 2404. 08144. 2024.

[26] Wei J, Wang X, Schuurmans D, et al. Chain-of-thought prompting elicits reasoning in large language models. NeurIPS. 2022:24824-24837.

[27] Steenhoek B, Rahman MM, Jiles R, et al. LLMs cannot reliably identify and reason about security vulnerabilities yet: a comprehensive evaluation, framework, and benchmarks. IEEE Symposium on Security and Privacy. 2024:1-18.

[28] Mussabayev R, Khairullin R, Kassymbekov D, et al. Code vulnerability detection: a comparative analysis of emerging large language models. arXiv:2409.10490. 2024.

[29] Sun C, Wang Y, Wu S, et al. Everything you wanted to know about LLM-based vulnerability detection but were afraid to ask. arXiv:2504.13474. 2025.

[30] Zhou, X., Cao, S., Sun, X., & Lo, D. (2025). Large language model for vulnerability detection and repair: Literature review and the road ahead. ACM Transactions on Software Engineering and Methodology, 34(5), 1-31.

[31] Liu, Z., Tang, Z., Zhang, J., Xia, X., & Yang, X. (2024, April). Pre-training by predicting program dependencies for vulnerability analysis tasks. In Proceedings of the IEEE/ACM 46th International Conference on Software Engineering (pp. 1-13).

[32] Jelodar, H., Bai, S., Hamedi, P., Mohammadian, H., Razavi-Far, R., & Ghorbani, A. (2025). Large Language Model (LLM) for Software Security: Code Analysis, Malware Analysis, Reverse Engineering. arXiv preprint arXiv:2504.071

[33] Jelodar H, Alavizadeh H, Esmaeili A, et al. Large language model for software security: code analysis, malware analysis, reverse engineering. arXiv:2504.07137. 2025.

[34] Qian X, Liu Y, Zhang H, et al. Exploring LLMs for malware detection: review, framework design, and countermeasure approaches. arXiv:2409.07587. 2024.

[35] Qian X, Chen B, Zhang Y, et al. LAMD: context-driven Android malware detection and classification with LLMs. arXiv:2502.18456. 2025.

[36] Omar M, Zangana HM, Al-Karaki JN, Mohammed D. Harnessing LLMs for IoT malware detection: a comparative analysis of BERT and GPT-2. IEEE ISMSIT. 2024:1-6.

[37] Zahan N, Burckhardt P, Lysenko M, et al. Leveraging large language models to detect npm malicious packages. arXiv:2403.12196. 2024.

[38] Jiang Y, Zhang W, Pang J, et al. Transformers and large language models for efficient intrusion detection systems: a comprehensive survey. arXiv:2408.09344. 2024.

[39] Lai C, Wang Y, Chen X, et al. Large language models in wireless application design: in-context learning-enhanced automatic network intrusion detection. arXiv:2405.17234. 2024.

[40] Bai J, Wang Y, Zhang L, et al. PhishDebate: an LLM-based multi-agent framework for phishing website detection. arXiv:2506.15656. 2025.

[41] Lee J, Lim P, Hooi B, Divakaran DM. Multimodal large language models for phishing webpage detection and identification. eCrime Symposium. 2024:1-12.

[42] Lim B, Kumar P, Tan A, et al. EXPLICATE: enhancing phishing detection through explainable AI and LLM-powered interpretability. arXiv:2503.20796. 2025.

[43] Afane K, Meli A, Khan LA, Hamlen KW. Next-generation phishing: how LLM agents empower cyber attackers. arXiv:2411.13874. 2024.

[44] Gioacchini L, Mellia M, Drago I, et al. AutoPenBench: benchmarking generative agents for penetration testing. arXiv:2410.03225. 2024.

[45] Deng G, Liu Y, Mayoral-Vilches V, et al. PentestGPT: evaluating and harnessing large language models for automated penetration testing. USENIX Security Symposium. 2024:847-864.

[46] Fang R, Bindu R, Gupta A, Kang D. Teams of LLM agents can exploit zero-day vulnerabilities. arXiv:2406.01637. 2024.

[47] Muzsai L, Imolai D, Lukács A. HackSynth: LLM agent and evaluation framework for autonomous penetration testing. arXiv:2412.01778. 2024.

[48] Shen, X., Wang, L., Li, Z., Chen, Y., Zhao, W., Sun, D., ... & Ruan, W. (2025, August). Pentestagent: Incorporating llm agents to automated penetration testing. In Proceedings of the 20th ACM Asia Conference on Computer and Communications Security (pp. 375-391).37.

[49] Henke J. AutoPentest: enhancing vulnerability management with autonomous LLM agents. arXiv:2505.10321. 2025.

[50] An, R., Chen, K., & Li, H. (2025). Unsupervised low-dose CT reconstruction with one-way conditional normalizing flows. IEEE Transactions on Computational Imaging.

[51] Chen W, Zhang Y, Liu X, et al. IntellBot: retrieval augmented LLM chatbot for cyber threat knowledge delivery. arXiv:2411.08234. 2024.

[52] Chen W, Zhang Y, Liu X, et al. Labeling NIDS rules with MITRE ATT&CK techniques: machine learning vs large language models. arXiv:2412.12456. 2024.

[53] Kunwar D, Sharma P, Prakash I, et al. Leveraging LLMs for non-security experts in threat hunting: detecting living off the land techniques. Computers. 2025;7(2):31.

[54] Huang, X., Ruan, W., Huang, W., Jin, G., Dong, Y., Wu, C., ... & Mustafa, M. A. (2024). A survey of safety and trustworthiness of large language models through the lens of verification and validation. Artificial Intelligence Review, 57(7), 175.

[55] Liu Y, Deng G, Li Z, et al. Jailbreaking ChatGPT via prompt engineering: an empirical study. arXiv:2305.13860. 2023.

[56] Carlini N, Tramer F, Wallace E, et al. Extracting training data from large language models. USENIX Security. 2021:2633-2650.

[57] Ji, Z., Lee, N., Frieske, R., Yu, T., Su, D., Xu, Y., ... & Fung, P. (2023). Survey of hallucination in natural language generation. ACM computing surveys, 55(12), 1-38.

[58] Solaiman I, Brundage M, Clark J, et al. Release strategies and the social impacts of language models. arXiv:1908.09203. 2019.

[59] Li Z, Yu X, Zhang Y, et al. Understanding the effectiveness of large language models in code vulnerability detection. arXiv:2311.16169. 2023.

[60] Ding Z, Xu M, Wang Y, et al. Beyond accuracy: evaluating LLMs for software vulnerability detection. arXiv:2402.15432. 2024.

[61] Zhang K, Li Y, Wang X, et al. Few-shot vulnerability detection with contrastive learning. arXiv:2403.15432. 2024.

[62] Chen X, Hao Z, Li L, et al. CruParamer: learning on parameter-augmented API sequences for malware detection. IEEE Transactions on Information Forensics and Security. 2022; 17:788-803.

[63] Long, S., Tan, J., Mao, B., Tang, F., Li, Y., Zhao, M., & Kato, N. (2025). A survey on intelligent network operations and performance optimization based on large language models. IEEE Communications Surveys & Tutorials.

[64] Feng, R., Chen, H., Wang, S., Karim, M. M., & Jiang, Q. (2025). LLM-MalDetect: A Large Language Model-Based Method for Android Malware Detection. IEEE Access.

[65] Gibert D, Planes J, Le Q, Zizzo G. A wolf in sheep clothing: query-free evasion attacks against machine learning-based malware detectors with GANs. IEEE EuroS&P. 2023:1-16.