ISSN: 2754-6659

Journal of Artificial Intelligence & Cloud Computing

SCIENTIFIC Research and Community

Review Article

Open Access

Generative AI for Vulnerability Management: A Blueprint

Varadharaj Varadhan Krishnan

Independent Researcher, Washington, USA

ABSTRACT

Cybersecurity threats continue to evolve in complexity and scale every year. The need for more effective vulnerability management has become more important now than ever. Traditional vulnerability management methods rely heavily on human resources for the time-consuming research aspect. Relying on manual processes and having multiple human touchpoints affects the ability to scale and the velocity of the team. In this paper, a design blueprint for integrating Generative Artificial Intelligence (AI), particularly large language models (LLMs) into the vulnerability management lifecycle are discussed. By leveraging the advanced capabilities of Generative AI, organizations can improve efficiency across various stages of vulnerability management, from intelligence gathering, risk analysis, and prioritization to remediation. Specifically, the paper provides a comprehensive blueprint for applying Generative AI to vulnerability intelligence, research, and remediation. The paper illustrates a logical design for each of these areas and discusses the design, key components, and functions. The paper also captures the challenges of adopting Generative AI-based solutions and guidance to overcome them. Lastly, the paper outlines future research directions aimed at overcoming challenges and further enhancing the role of AI in vulnerability management.

***Corresponding author**

Varadharaj Varadhan Krishnan, Independent Researcher, Washington, USA.

Received: August 19, 2024; **Accepted:** August 23, 2024; **Published:** September 02, 2024

Keywords: Generative Artificial Intelligence, Large Language Model, Vulnerability Management, Vulnerability Research, Vulnerability Intelligence

Introduction

Today, cybersecurity threats are increasingly sophisticated and frequent [1]. To protect an organization from such a persistent onslaught of attacks, a robust security posture management program is a must. One of the key areas of security posture management is vulnerability management [2]. More than ever, robust vulnerability management is now critical. Vulnerability Management includes various processes involved in identifying, assessing, reporting, and remediating security vulnerabilities in software systems [3,4]. In the last decade, various industries have gone through digital transformation, which has led to an exponential increase in the number of software applications and consequently the number of vulnerabilities. Traditional approaches to vulnerability management are heavily reliant on human expertise and manual processes with some automation [5]. Though automation helps with improving the efficiency of a vulnerability management team, a significant part of the analysis and risk calculation is done manually, making them time-consuming and room for oversight. The recent advancements in Generative Artificial Intelligence (AI) offer solutions to some of the challenges with traditional vulnerability management. Generative AI, particularly large language models (LLMs), have demonstrated great success in processing and generating human-like text, making them well-suited to augment various stages of the vulnerability management lifecycle. It can be applied from enhancing vulnerability intelligence to automating remediation processes [6]. This paper explores the intersection of Generative AI and vulnerability management and presents a design blueprint for integrating AI-driven solutions into existing vulnerability management practices. By leveraging the strengths of Generative AI, organizations can significantly reduce the time and effort required for vulnerability management and strengthen their defenses against ever-evolving cyber threats.

Background: Vulnerability Management

Vulnerability management is the process of identifying, assessing, reporting, and remediating security vulnerabilities in software systems [7]. Vulnerability management is one of the vital functions of the cybersecurity team of an organization aimed to reduce possible threats to the organization and shrink the attack surface. Vulnerabilities are technically a weakness in any software system that allows attackers to compromise the system and existing security controls built in and around the software system. The traditional vulnerability management process involves using a scanner, or an agent more formally known as an endpoint security agent, to discover all assets on a computer network and then perform assessments on them to discover known vulnerabilities [8]. After the vulnerability assessment, each identified vulnerability is then analyzed for risk. A prioritization logic is applied to prioritize the most important vulnerability occurrence that needs to be addressed first. The prioritization logic varies from organization to organization depending on the IT environment and system that are core to running their organization. On a high level, the vulnerability management lifecycle involves four main steps.

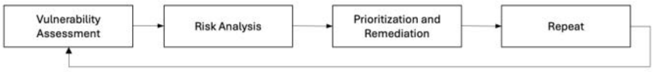

Figure 1: Vulnerability Management Lifecycle

The risk analysis process involves answering questions like

- Is the vulnerability finding a true positive or false positive?
- Can it be remotely exploited from the internet or from outside the organization?
- What is the difficulty in exploiting the vulnerability?
- Are there any readily available exploits available to the public?
- What would be the size of the impact if the vulnerable system is compromised?
- Are there any mitigative controls in place to prevent the exploitation of the vulnerability?

By answering these questions, one can determine the level of the risk posed by the vulnerability. The next step in the process is remediation. Remediation can be done in multiple ways. A direct way is to address the vulnerability by applying a patch. In cases where patches are not available, the risk posed by the vulnerability can be mitigated by performing actions that reduce the likelihood of exploitation of the vulnerability. These changes can be done within the software system, which has the vulnerability, or outside of it, too. The last option would be accepting the risk associated with the vulnerability. Generally, low-risk vulnerabilities are dealt with this type of remediation. Though there are various tools and automation are available to efficiently and effectively perform actions for every stage of the vulnerability management process, this process involves significant human interaction for analysis, critical thinking, and identifying alternate mitigations for vulnerabilities. With recent developments in the Generative AI domain, the technology can be adopted to improve and optimize the entire vulnerability management lifecycle. In this paper, we discuss the opportunities where Generative AI can be used in the vulnerability management lifecycle, and a solution blueprint is also discussed.

Background: Generative AI

Generative AI is a subdomain of the Artificial Intelligence subject area focused on building systems to generate content artificially, where the content generation happens based on the learnings from the large data set used during the training [9]. Generative AI is a subset of Deep Learning and is more closely associated with the term GPT today. GPT stands for Generative Pre-Trained Transformer; these GPTs are a type of Neural Network good at identifying relationships within a sequence, like words in a sentence [6]. Transformer-based machine learning models have been around for many years; the critical turning point in transformer architecture came with the Self-Attention technique introduced in the paper "Attention is All You Need" by Vaswani and team in 2017 [6,10,11]. The paper describes a technique that allows the model to be effective for tasks involving sequential data, like text, image, audio, or video. The transformer architecture described in the paper also paved the way for faster processing and horizontal scaling, resulting in faster neural network training on very large data sets [12]. The machine learning models produced by these techniques are generally referred to as the Large Language Model (LLM); in the rest of the paper, various applications of vulnerability management are discussed.

Vulnerability Intelligence

Vulnerability intelligence is a form of threat intelligence focused on the aggregation of information about software system vulnerabilities. With the exponential growth in software and software usage in about every aspect of modern-day life, the number of vulnerabilities in these software systems is growing exponentially, too. Most organizations use multitudes of software, and as such, it is imperative for organizations to have comprehensive vulnerability intelligence to better protect themselves from threat actors. Today, vulnerability disclosures are published through social media, public web forums, the dark web, independent researcher blogs, product vendor advisories, and more. There is no single source for all types of vulnerabilities. Keeping track of all these channels, analyzing content from these channels, and correlating them is a challenge. Adding to the difficulty is the volume of these vulnerabilities coming through different channels. Generative AI can be applied here to make the aggregation, summarization, and even analysis faster.

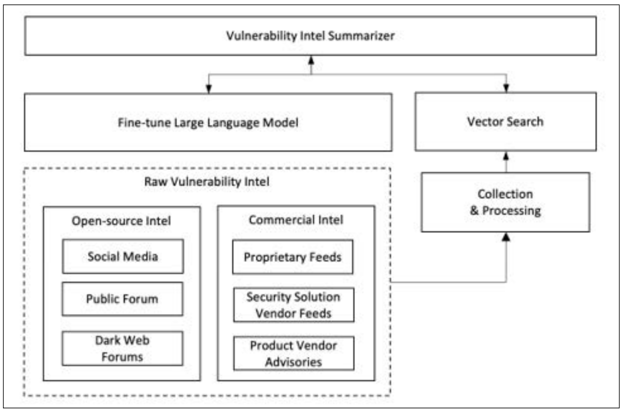

Figure 2: System Design AI-Based Vulnerability Intelligence System

An intelligent agent implementation using LLM is one of the powerful use cases for applying Generative AI in cybersecurity. As discussed earlier, vulnerability management teams deal with a lot of structured data from feeds and unstructured data from web forums, social media sites, and more [13]. The design in Figure 2 is for a virtual assistant-type system that can summarize data from various sources and support it with analysis and remediations. The main point of interaction would be an interface where users can input queries or data regarding potential security concerns or emerging vulnerability disclosures. At the core of this system is a fine-tuned Large Language Model, which is tailored to interpret and analyze the specific terminology and nuances of vulnerabilities. The LLM sifts through vast amounts of data, extracting relevant information and generating insights that are essential for identifying and understanding vulnerabilities. A fine-tuned LLM with Retrieval Augmented Generation is the foundation of the solution. RAG (Retrieval-Augmented Generation) is an AI framework that combines the strengths of traditional information retrieval systems (such as databases) with the capabilities of generative large language models (LLMs). By combining this extra knowledge with its own language skills, the AI can write text that is more accurate, up-to-date, and relevant. In the design illustrated in Figure 2, Open-source intel and proprietary intel are parsed and maintained in a vector DB. User queries are then enriched with the data stored in the RAG database before it is sent to the LLM model. This technique allows for the retrieval of information that is not just keyword-matched but semantically linked to the user's query, enabling more accurate and relevant results. An automatic intelligent summarizer can reduce the time taken for the vulnerability management team to act on new situations emerging in the wild.

Vulnerability Research

Vulnerability research is another important domain of cybersecurity, focusing on identifying and analyzing potential vulnerabilities that malicious actors could exploit. Generative AI can be applied here to enhance the discovery and analysis of security weaknesses in a given software or system [14]. Figure 3 shows a logical design of a system using LLM for vulnerability research. The fundamental approach here is to use Generative AI for the mutator function to generate mutated seed data for fuzzing [15,16]. Seed Data: The process starts with seed data, which typically includes code samples, system configurations, or application binaries known to be stable. This data acts as a baseline for testing and analysis. Mutator: The seed data is then processed by a mutator, an algorithm that systematically alters the data or code to create numerous permutations. This process is designed to simulate potential attack vectors or uncover unexpected behavior in the software that could lead to vulnerabilities. Fine-tuned Large Language Model: Parallel to mutation, a fine-tuned Large Language Model (LLM) refines the mutation process by analyzing the code and suggesting areas where vulnerabilities are likely to occur. This LLM has been specifically adjusted to understand and predict software vulnerability patterns [17,18]. Generated Test Cases: The mutator and the LLM work together to produce a comprehensive set of test cases. These test cases represent a variety of mutated data scenarios derived from the seed data, each potentially exposing a different vulnerability in the system. Runtime/Application Testing: The generated test cases are then applied to the runtime environment or application under scrutiny. The application is monitored for unexpected behavior, crashes, or other indicators that could signify a bug or vulnerability. Crash Detector: As test cases are executed, a crash detector monitors the system for failures. These failures, often indicative of underlying vulnerabilities, are flagged for further investigation.

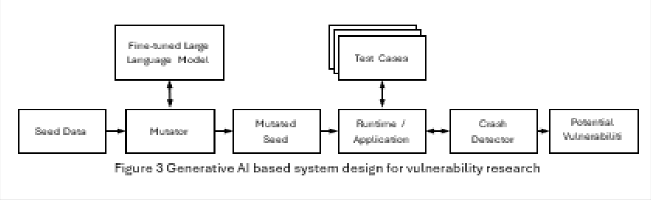
Fine-tuned Large Language Model
Test Cases
Seed Data
Mutator
Mutated Seed
Runtime / Application
Crash Detector
Potential Vulnerabiliti
Figure 3 Generative AI based system design for vulnerability research

Figure 3: Generative AI Based System Design for Vulnerability Research

Vulnerability Remediation

In the vulnerability management lifecycle, the remediation step is another important stage. At this step, the risks posed by the vulnerabilities are addressed or mitigated. It primarily involves applying security patches and software updates, implementing new security controls, and finally verifying the effectiveness of security measures [6,19]. The goal of the vulnerability remediation process is to reduce the risk; the focus should be addressing high-risk vulnerabilities first. The process of determining which vulnerability gets fixed first is called vulnerability prioritization. There are various strategies for prioritization; risk-based prioritization is the most adopted method. Risk can be calculated as the product of impact and likelihood [20]. To calculate, security professionals need to have an understanding of the vulnerability, the asset, the environment, and the business service details. Performing such analysis is time-consuming. Generative AI can be applied here to solve the challenges and improve the velocity. An intelligent virtual assistant with knowledge of the vulnerabilities, assets, and business services can assist with analysis. Figure 3 illustrates a design of a Large Language Model-d chat interface to assist with vulnerability remediations.

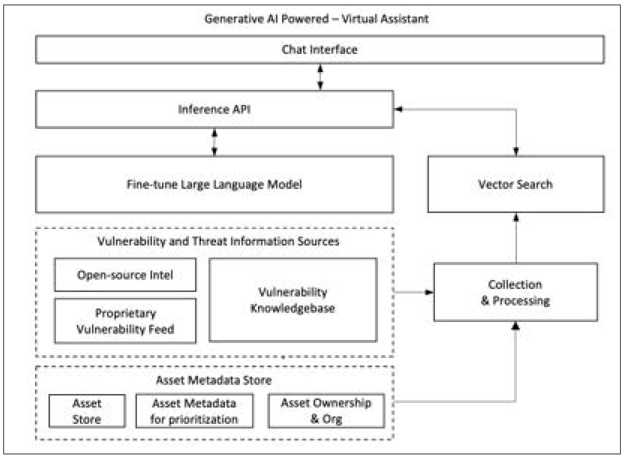
Generative AI Powered – Virtual Assistant
Chat Interface
Inference API
Fine-tune Large Language Model
Vector Search
Vulnerability and Threat Information Sources
Open-source Intel
Proprietary Vulnerability Feed
Vulnerability Knowledgebase
Collection & Processing
Asset Metadata Store
Asset Store
Asset Metadata for prioritization
Asset Ownership & Org

Figure 4: Generative AI-based System Design for Vulnerability Remediation

The core component of the solution is the knowledge base consisting of the vulnerability details, asset inventory, asset ownership information, and business service metadata. A Retrieval Augmented Generation database with this knowledge is created. The Inference API layers perform the orchestration of querying the RAG database and use the results to enrich the original ask by the virtual assistant user. The agent can be used to answer a wider variety of questions related to the vulnerability and the vulnerability occurrence. The analyst could have the agent summarize the inherent risk associated with the vulnerability and further calculate the likelihood. The impact assessment is also made easy with the virtual assistant, which has knowledge about the asset and the business service. The virtual assistant can be used to assist the asset owner's remediation of the vulnerability. The agent can be designed to summarize the patching or software update that can fix the problem. The essence of the idea is to build an intelligent agent who understands the environment and can assist analysts by answering their questions in plain text.

Challenges and Future Work

While the potential of applying Generative AI in vulnerability management to boost efficiency is immense, there are several challenges that must be addressed to completely realize its benefits. Generative AI models, particularly large language models (LLMs), require large amounts of data for training. In the cybersecurity-related use case, the data used for training could be sensitive, and it poses additional challenges in securing that and preventing data leakage. Out of all the challenges, the main is the accuracy and reliability of AI-generated insights. Although LLMs have shown remarkable proficiency in processing and generating content, they are not 100% accurate all the time. Hallucinations are common, and sometimes LLM would give outright wrong answers [10,14,21]. The potential for generating false positives or overlooking critical vulnerabilities remains a concern while using them for vulnerability management use cases. This can be addressed with a balance of AI and human expertise instead of totally removing the human element in executing critical tasks. Model bias and fairness are additional concerns. The data used to train the base generative models can unintentionally introduce biases, resulting in skewed assessments of risk or the prioritization of vulnerabilities. Addressing these biases would require continuous monitoring and fine-tuning of the AI models to keep them unbiased. Lastly, scalability and integration complexity also pose significant challenges. Organizations must ensure that solutions with AI systems can scale effectively to handle the volume and CPU-intensive tasks. Large Language Models still would require significant computing, and a system that will be used by many would require expensive scaling.

Looking forward, there are several areas for future work that will further enhance the use of Generative AI in vulnerability management. Research into Collaborative AI systems, where multiple AI models work together to validate findings and recommendations. This approach can enhance accuracy and reduce the risk of false positives. Secondly, applying generative AI to proactively threat hunting and vulnerability research can help identify potential vulnerabilities before they are found by malicious actors and exploited.

Conclusion

This paper presented a blueprint for applying Generative AI technology for vulnerability management. This design blueprint aims to aid organizations and security product engineering teams in how to leverage the powerful Generative AI capabilities to improve vulnerability management operations efficiency and velocity. The architecture and the system design presented serve as a starting point for organizations to further develop their solutions and can be further expanded to create development roadmaps. Successful implementation of Generative AI for the vulnerability management lifecycle has its own challenges, including data privacy, accuracy of the output, and inbuilt bias, which should be considered before deploying solutions that are mission-critical and have low fault tolerance. With the advances in AI tools and the application of AI advancement, it is essential for cyber security professionals to equip themselves with the knowledge and skills to effectively use them.

In conclusion, Generative AI presents an excellent transformative opportunity for vulnerability management; it offers the potential to significantly improve and transform how organizations protect their digital information technology footprint. By embracing these technologies, cybersecurity teams can stay ahead of the curve and be equipped and prepared to meet the challenges of ever-evolving cyber threats [22-24].

References

1. (2022) Enhancing cybersecurity with AI: The new frontier. Microsoft Security https://www.microsoft.com/security/blog/2022/03/07/enhancing-cybersecurity-with-ai-the-new-frontier/.
2. (2021) Artificial intelligence for cybersecurity. Cybersecurity and Infrastructure Security Agency (CISA) https://www.cisa.gov/uscert/ncas/current-activity/2021/11/04/artificial-intelligence-cybersecurity.
3. The role of artificial intelligence (AI) in security automation. Palo Alto Networks https://www.paloaltonetworks.com/cyberpedia/role-of-artificial-intelligence-ai-in-security-automation.
4. (2022) The transformative role of AI in cybersecurity: Understanding current applications and benefits. R Street https://www.rstreet.org/commentary/the-transformative-role-of-ai-in-cybersecurity-understanding-current-applications-and-benefits/.
5. Ferrag MA, Ndhlovu M, Tihanyi N, Cordeiro LC, Debbah M, et al. (2024) Revolutionizing cyber threat detection with large language models: A privacy-preserving BERT-based lightweight model for IoT/IIoT devices. IEEE Access https://ieeexplore.ieee.org/stamp/stamp.jsp?arnumber=10423646.
6. (2024) Applying generative AI for CVE analysis at an enterprise scale. NVIDIA Developer https://developer.nvidia.com/blog/applying-generative-ai-for-cve-analysis-at-an-enterprise-scale/.
7. Gillette B, Karvelas K (2024) AI-based vulnerability management: Current challenges and opportunities. arXiv https://arxiv.org/pdf/2405.02435.
8. Artificial intelligence in cybersecurity. Fortinet https://www.fortinet.com/resources/cyberglossary/artificial-intelligence-in-cybersecurity.
9. Myers D, Mohawesh R, Chellaboina VI, Anantha Lakshmi Sathvik, Praveen Venkatesh, et al. (2024) Foundation and large language models: Fundamentals, challenges, opportunities, and social impacts. Cluster Computing 27: 1-26.
10. Liu Z (2024) A review of advancements and applications of pre-trained language models in cybersecurity. 12th International Symposium on Digital Forensics and Security (ISDFS) 1-10.
11. (2023) Leveraging AI-informed cybersecurity to measure, communicate, and eliminate cyber risk. Qualys https://blog.qualys.com/qualys-insights/qualys-security-conference/2023/11/09/leveraging-ai-informed-cybersecurity-to-measure-communicate-and-eliminate-cyber-risk.
12. Vaswani N Shazeer, N Parmar, J Uszkoreit, L Jones, AN Gomez, et al. (2017) Attention is all you need. Advances in Neural Information Processing Systems 30.
13. Li Z, Zou D, Xu S, Ou X, Jin H, et al. (2018) VulDeePecker: A deep learning-based system for vulnerability detection. arXiv preprint arXiv:1801.01681.
14. Y Liu, G Deng, Y Li, K Wang, T Zhang, et al. (2023) Prompt injection attack against LLM-integrated applications. arXiv preprint arXiv:2306.05499.
15. Lu S, Guo D, Ren S, Huang J, Svyatkovskiy A, et al. (2021) CodeXGLUE: A Machine Learning Benchmark Dataset for Code Understanding and Generation. arXiv preprint arXiv:2102.04664.
16. (2023) Is AI-based vulnerability management really that efficient? AIthority https://aithority.com/machine-learning/is-ai-based-vulnerability-management-really-that-efficient/.
17. (2023) 5 ways AI-driven patch management is driving the future of cybersecurity. VentureBeat https://venturebeat.com/security/5-ways-ai-driven-patch-management-is-driving-the-future-of-cybersecurity/.
18. Zhou Y, Liu S, Siow J, Du X, Liu Y (2019) Devign: Effective Vulnerability Identification by Learning Comprehensive Program Semantics via Graph Neural Networks. arXiv e-prints arXiv:1909.03496.
19. Thapa C, Jang SI, Ahmed ME, Camtepe S, Pieprzyk J, et al. (2022) Transformer-based language models for software vulnerability detection. Proceedings of the 38th Annual Computer Security Applications Conference 481-496.
20. Liu Y, Ott M, Goyal N, Du J, Joshi M, et al. (2019) RoBERTa: A Robustly Optimized BERT Pretraining Approach. arXiv preprint arXiv:1907.11692.
21. (2023) AI-powered vulnerability management. IBM Security https://securityintelligence.com/posts/ai-powered-vulnerability-management/.
22. Yan B, Li K, Xu M, Dong Y, Zhang Y, et al. (2024) On protecting the data privacy of large language models (LLMs): A survey. arXiv preprint arXiv:2403.05156.
23. Ebert C, Beck M (2023) Artificial intelligence for cybersecurity. IEEE Software 40: 27-34.
24. Fang R, Bindu R, Gupta A, Kang D (2024) LLM agents can autonomously exploit one-day vulnerabilities. arXiv preprint arXiv:2404.08144.