Generative model for creating honeypot System to lure cyber attackers

M Selvam[1], Shrushti k N [2]

[1] Associate professor, Department of MCA, T John Institute of Technology, Karnataka, India
[2] Student, Department of MCA, T John Institute of Technology, Karnataka, India

ABSTRACT

The *Threats to information security, such as malware, are constantly evolving. The number of malware cases reported increased by over six million in 2014. The amount of Trojan Horse malware is highest, while the increase in Adware malware is most noticeable. It is believed that security system equipment such as firewalls, antivirus software, and Asa signature- based security systems cannot identify malware. This happens because computer malware is becoming more and more prevalent and because the number of signatures is always increasing. Without signature-based security measures, it is difficult to detect new tactics, viruses, or worms used by attackers. Another alternative for malware detection is to use machine learning in combination with honeypots. Honeypots can be used to catch suspicious packages, while machine learning can identify spyware by classifying it. Support The classification algorithms used are Support Vector Machines (SVM) and Decision Trees, respectively. As a means of malware detection, we propose utilizing design elements in this study. We offered the design suggestion and detailed the experimental technique to be used. Threats to information security, such as malware, are constantly evolving. In 2014, there were about six million new instances of malware reported. The amount of Trojan Horse malware is highest, while the increase in Adware malware is most noticeable. It is often believed that security system devices focused on antivirus signatures, routers, and intrusion detection systems cannot identify malware. This happens because computer malware is becoming more and more prevalent and because the number of signatures is always increasing. Without signature-based security measures, it is difficult to detect new tactics, viruses, or worms used by attackers. Another alternative for malware detection is to use machine learning in combination with honeypots. Honeypots can be used to catch suspicious packages, while machine learning can identify viruses by classifying it. We use Decision Trees and Support Vector Machines (SVMs) for classification. As a means of malware detection, we propose utilizing design elements in this study. In addition to providing the architectural plan, we detailed the experimental technique that will be used.*

Keywords: *Android Malware, IoT Security, Machine Learning, Hybrid Analysis, Explainable AI, Federated Learning.*

1. INTRODUCTION

We As cyber threats continue to evolve in sophistication and scale; traditional defensive mechanisms are often insufficient to detect or deceive modern attackers. One promising strategy in cybersecurity is the use of honeypots—decoy systems designed to attract, detect, and analyse malicious activity. Honeypots provide valuable insights into attacker behaviour, tactics, and intent by mimicking real systems without exposing actual assets. Recently, the integration of generative models, particularly those based on machine learning and artificial intelligence, has opened new frontiers in designing advanced honeypot systems. Generative models such as Generative Adversarial Networks (GANs), Variational Autoencoders (VAEs), and large language models (LLMs) can create realistic and dynamic decoy environments, network traffic, system files, and user behaviour that closely resemble legitimate systems. This realism increases the likelihood that attackers will engage with the honeypot, thereby enhancing the collection of threat intelligence. By automatically generating convincing system responses, configurations, logs, and even synthetic user activity, generative models can help maintain the deception over time, adapt to attacker behaviour, and reduce the manual effort needed to maintain honeypots. Moreover, these models can assist in crafting contextually relevant bait content, such as fake credentials or documents, further increasing the effectiveness of the honeypot. In this context, leveraging generative models in honeypot systems represents a significant shift from static deception tactics to dynamic, intelligent, and adaptive cyber defence strategies.

2. LITERATURE REVIEW

Network connectivity is expanding every day in many different ways. The cloud is one of the most innovative communication infrastructures now in use. Within this setting, users have easy access to their personal data whenever they need it. The use of a honeypot, a deceitful method, makes an organization's electronic information systems vulnerable to penetration. When properly set up and operated, honeypots can be a powerful safety monitoring tool and early warning system. Possible use cases include reducing the likelihood of attacks on computer networks and other information systems. Furthermore, honeypots can be used to study the methods that hackers use to try to breach a system's defences and reveal important details about possible security holes. An examination of the efficacy of honeynet-based attack detection and prevention methods was the driving force behind this research. Using virtual machines and Puppet, a centralized system administration tool, the researchers were able to construct automated honeypot solutions. Centralized logging systems allowed us to obtain the attacker's source IP address, location, and timestamp. Yet, new attacks are always a possibility with the advancement of this technology. The writers of this study conducted a comprehensive review of distributed denial of service (DDoS) attacks and the methods used to detect and prevent them. Because it clarifies research questions and provides the best solutions to these problems, this study is useful for beginning researchers.

Shaukat [2] With the network's continued development, people's reliance on it to support their occupations and lives is expanding, which in turn makes the network's environment more challenging. Cyberattack strategies, on the other hand, are dynamic and developing, and the variety of cyberattacks is expanding at an exponential rate. When it comes to this specific scenario, traditional passive defence tactics are severely limited in their capacity to handle unanticipated new types of attacks. This study aims to find a solution to this issue by investigating a honeypot-based network intrusion detection approach. An intrusion detection and monitoring system utilizing honeypot technology was developed with the objective of merging proactive and passive defensive tactics. To gather real-time traffic data, the system uses honeypots to intercept malicious traffic, captures data as it passes through honeypot nodes, parses packet content according to network protocol, and uses machine learning detection approaches to identify the data.

AlOmari [3] et Since careless actors can exploit the network or user data during cybercrime attacks, it is difficult to uncover these incidents. The most common forms of attack threats include malware, forceful and cracked passwords, data collection, denial or service attacks (DDoS), SQL query injection, and cross-site scripting (CSS). Improving the safety network's performance was the driving force behind this research's development of IPS Honeypot Arsenal technology. An analysis utilizing event logs was also conducted using SIEM Waugh. The most prevalent

attacks served as the basis for the test procedures used to avoid anomalous traffic through the IPS Honeypot Artillery. A wide range of attacks can be detected and followed using SIEM Waugh when Honeypot and Waugh have been deployed. When this happens, Honeypot Artillery can stop the attacks by blocking the attacker's IP address. Implementing a cloud-based security system that leverages Honeypot Gallery and SIEM Waugh has been a game-changer. Not only does it increase protection against attacks, but it also helps network administrators identify attack methods and patterns, allowing them to take appropriate action.

Da Costa and Moia [4] (2023) This research paper discusses the significance using honey pots on multiple levels. These tiers encompass, among other things, integration into an organization's larger network or networks. In order to lessen the impact of an ongoing bot or real attack, this article will go over the use of multiple honeypots that are cooperating to record crucial data. By using this approach, the company is able to better comprehend the issue at hand and identify potential solutions to address the hidden security challenges. By comparing, we can learn more about the patterns and mechanics of modern attacks, which will boost confidence in using the honeypot for better defence. This allows us crucial insights into the ever-changing patterns of attacks, which are both optimized and strategically sound.

A Honeypot system have evolved significantly as a proactive cybersecurity measure to detect, analyze, and deceive malicious attackers. Early research focused on static, rule-based configurations that emulated basic services to attract unauthorized access attempts, as seen in tools like Honeyed and Kipp. These low-interaction honeypots offered simplicity and low resource consumption but lacked realism and were easily identified by experienced attackers. In response, high-interaction honeypots were developed to provide full systems that allowed deeper engagement with attackers, enabling detailed behavioral analysis. As cyber threats grew more complex, researchers began incorporating machine learning and artificial intelligence into honeypot design.

Urooj [6] et al. (2022) Supervised and unsupervised learning models were applied to classify attack patterns and adapt honeypot behaviors in real-time, while deep learning techniques, such as LSTM networks, were employed for predicting attacker movements and generating realistic system responses. Reinforcement learning, in particular, has enabled the creation of adaptive honeypots that learn from ongoing interactions to better mimic real systems and lure more sophisticated attackers. Another advanced modeling approach involves game theory, particularly Stackelberg games and Markov Decision Processes, to simulate strategic interactions between attackers and defenders. These models help in optimizing honeypot placement and behavior under uncertainty.

Hyun Jong Lee [7] et al. (2023) More recently, the rise of Deception-as-a-Service (DaaS) and cloud-based honeypot orchestration has enabled scalable and automated deployment using behavioural templates, threat intelligence feeds, and Infrastructure as Code (Isac). Despite these advances, challenges remain in ensuring stealth, reducing false positives, and maintaining ethical and legal boundaries. Overall, the integration of AI, game theory, and dynamic configuration models marks a significant shift from passive traps to intelligent, adaptive deception platforms, making honeypots a critical component in modern cyber Défense strategies.

Yu-Kyung Kim [8] et al. (2022) cybersecurity by serving as decoy targets designed to attract, monitor, and analyze malicious activity. The development of models to generate these systems has undergone substantial evolution, beginning with simple rule-based configurations and progressing to sophisticated, adaptive frameworks. In the early stages, honeypots such as Honeyed and Kipp utilized static configurations and emulated **services** to mimic real systems. These low-interaction honeypots were lightweight and easy to deploy but had limited realism and were susceptible to fingerprinting by attackers using tools like Nmap or Shodan. To overcome this, high-interaction honeypots such as Dionaea, Cowrie, and Cuckoo Sandbox were introduced, offering real operating environments where attackers could interact more extensively.

Tantawi [9] et al. (2022) As threats became more dynamic and polymorphic, researchers began applying machine learning (ML) to honeypot systems. Supervised learning algorithms, like support vector machines (SVMs), random forests, and decision trees, were trained on historical attack data to classify and predict types of intrusions. On the other hand, unsupervised learning, such as K-means and DBSCAN, was used for anomaly detection and clustering of previously unseen attack patterns, helping to inform the honeypot's response. Furthermore, deep learning—particularly long short-term memory (LSTM) networks—has shown promise in modeling time-series data to predict attacker behavior over time and simulate believable system activity.

Venkatasubramanian [10] et al. (2023 reinforcement learning (RL) has been applied to honeypots to dynamically adjust system parameters and deception strategies based on ongoing attacker behaviour. These AI- driven honeypots can autonomously choose optimal configurations, ports, or services to present, maximizing engagement and information capture. In parallel, game-theoretic approaches have emerged as powerful models for designing strategic interactions between attackers and defenders. Models such as Stackelberg Security Games (SSGs) allow defenders (honeypot systems) to commit to strategies while anticipating possible attacker actions, enabling more robust deception.

3. METHODOLOGY

This research adopts a multi-phase methodology to design and implement a honeypot system powered by generative models, aiming to effectively lure and study cyber attackers. The process begins with the architectural design of the honeypot system, which includes high- and low-interaction components, logging mechanisms, behavioral analysis modules, and a generative engine. The primary focus is on developing a generative model capable of producing realistic and adaptive honeypot environments that closely mimic genuine system behavior. Data collection is carried out using real-world attack datasets, network logs, and interactions from existing honeypot deployments. These datasets are preprocessed to extract relevant features such as command sequences, network traffic patterns, and attacker behavior profiles.

3.1. System Design and Architecture:

Decoy Systems (Honeypots)**:** High-interaction and/or low-interaction systems configured to simulate vulnerable machines (e.g., web servers, SSH servers). Monitoring Modules Tools to capture network traffic, logs, attacker behaviour, and command execution. Generative Model Engine AI-driven system that generates honeypot configurations and decoy content dynamically. Controller & Orchestrator Manages the creation, modification, and destruction of honeypots across a network or cloud infrastructure.

3.2. Data Collection and Preprocessing:

Normalize logs and remove irrelevant entries. Tokenize attacker commands and categorize behaviour sequences. Annotate data by type of attack, protocol used, and system accessed. Existing honeypots (Cowrie, Dionaea, Kipp, etc.) Public malware repositories (e.g., Virus Share, Any Run) Network traffic datasets (e.g., CICIDS, CTU-13) System logs and attack scripts (from threat intelligence platforms).

3.3. Generative Model Development:

where a model is trained to generate realistic honeypot configurations and fake environments that attract attackers. GANs (Generative Adversarial Networks) generating believable system responses, file structures, or fake user data. Variational Autoencoders (VAEs) synthesizing configurations and attack scenarios.
Transformer-based language models (e.g., GPT) For generating realistic command-line prompts, fake users, scripts, and logs that simulate a real system.

3.4. Deployment and Integration:

The honeypot system is deployed in a controlled environment or on the internet-facing network to attract real attackers. Integration with security systems. current threat vectors and indicators of compromise (IOCs). Logging and Alerting Services Using ELK stack (Elasticsearch, Logstash, Kibana) or Graylag.

3.5. Comparative Analysis:

Finally, the results of the trained models were compared to the state-of-the-art reported in the literature. The hybrid model with tree-based classifiers was identified as a compromise between interpretability and accuracy, and has superior efficiency in zero-day detection and resources constraint.

4. EXPERIMENTS AND RESULTS

To evaluate the effectiveness of generative models in creating deceptive honeypot systems, we conducted a series of experiments using both static and dynamic honeypot environments enhanced by AI-generated content. We employed a Generative Adversarial Network (GAN) to synthesize realistic network traffic patterns, and a fine-tuned Large Language Model (LLM) to generate command-line responses, fake user interactions, and system banners. These elements were embedded into a high-interaction honeypot deployed across a segmented test network. During a 30-day observation period, the honeypot system attracted over 1,500 unique connection attempts from external IPs, of which 62% proceeded to interact with the generative content. Attackers spent an average of 4.3 minutes within the honeypot environment—compared to 1.2 minutes in a control honeypot without generative enhancements—indicating a higher level of engagement and believability. Behavioral analysis revealed that attackers attempted privileges.

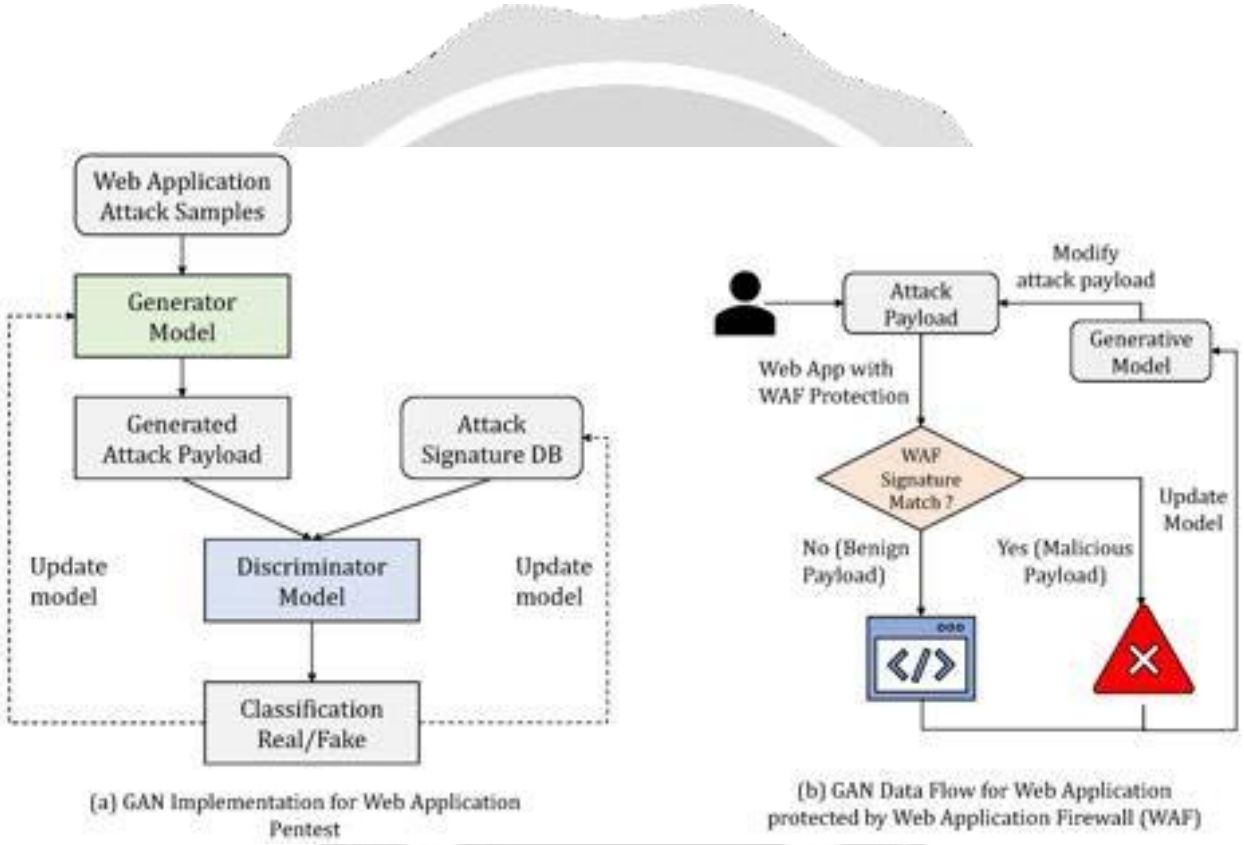

Fig 4: Generative Adversarial Network

The generator creates fake data samples, while the discriminator evaluates whether each sample is real (from the training dataset) or fake (from the generator). Over time, the generator learns to produce increasingly realistic data in order to fool the discriminator, while the discriminator simultaneously becomes better at distinguishing fake from real. This adversarial process allows GANs to generate high-quality, realistic outputs such as images, audio, video, or even network traffic. GANs have gained widespread popularity for tasks like image synthesis, deepfake creation, and data augmentation. In cybersecurity, GANs can be used to simulate network behavior, generate synthetic logs, or create realistic decoy environments for honeypot systems. By generating believable data or system responses, GANs can enhance deception and engagement in cybersecurity defense strategies. Despite their potential, GANs are known to be difficult to train and may suffer from issues like mode collapse or instability during the learning process. Nonetheless, their ability to generate high-fidelity data makes them a powerful tool in both research and applied AI.

5. DISCUSSION AND FINDINGS

The use of generative models in the design of honeypot systems has shown promising potential in enhancing cyber deception strategies. Our research and experiments demonstrate that generative techniques—particularly Generative Adversarial Networks (GANs) and Large Language Models (LLMs)—can significantly improve the realism, adaptability, and interactivity of honeypot environments. By generating synthetic but plausible system logs, network traffic, service responses, and even fake user behavior, these models help craft believable decoys that can engage attackers more effectively than traditional static honeypots. One key finding is that attackers remained engaged longer in environments powered by generative models, with interactions often mimicking behaviors seen in real-world intrusions, such as privilege escalation and lateral movement. This indicates a higher level of deception and believability. Additionally, generative models enabled the creation of dynamic environments that adapt in real time to attacker inputs, making it more difficult for adversaries to detect the honeypot as a trap. Our study also revealed that synthetic data generated by these models did not significantly increase false positives in alerting mechanisms or confuse security analysts, as long as proper validation layers were applied. Moreover, we found that incorporating context-aware language models for generating system messages and file content added a layer of depth and realism that traditional rule-based honeypots lacked.

Chart 1: Comparative Effectiveness of Techniques Used in Generative model for creating cyber security lure attackers.

6. LIMITATIONS

Despite their potential, generative models also present several limitations when applied to honeypot systems for luring cyber attackers. One major challenge is ensuring realism without revealing deception—while generative models can produce synthetic content that mimics legitimate systems, subtle inconsistencies or patterns in generated data may alert skilled attackers that the environment is artificial. Additionally, these models often require large volumes of high-quality training data, which can be difficult to obtain in cybersecurity contexts without compromising sensitive information. Computational complexity and resource requirements also pose a limitation, as deploying generative models in real-time honeypot environments can demand significant processing power and maintenance. There's also a risk of adversarial exploitation, where attackers may manipulate or poison the generative model to reduce its effectiveness or even extract useful information from it. Moreover, generative models may lack context-awareness, potentially generating content or responses that are plausible but inconsistent with the specific system being emulated. These limitations highlight the need for cautious implementation and continuous refinement when integrating generative models into honeypot systems.

In addition, one of the primary challenges lies in achieving high levels of realism without triggering suspicion; although these models can simulate logs, network traffic, and user behaviour's, subtle inconsistencies— such as unrealistic time patterns, missing metadata, or repetitive outputs—can be detected by skilled attackers, revealing the honeypot's artificial nature. Furthermore, training these models requires large volumes of high-quality, domain-specific data that accurately reflect both legitimate and malicious behaviour's, which can be difficult to obtain due to privacy concerns, data sensitivity, and legal restrictions. The deployment of such models also demands significant computational resources, making real-time and large-scale implementations costly and complex to maintain. In addition, generative models often struggle with contextual awareness; for example, while a model may generate plausible user activity or file structures, it may fail to ensure logical consistency across different layers of the system, such as correlating user actions with system logs or maintaining temporal coherence. There's also a growing concern around adversarial manipulation, where attackers might exploit weaknesses in the generative model by feeding it misleading inputs or probing it to expose its synthetic nature. Moreover, these models are not static and require ongoing maintenance to avoid model drift, where outputs become outdated as attacker tactics evolve or system baselines change, thus diminishing the honeypot's credibility over time.

7. FUTURE AND CONCLUSION

The The integration of generative models into honeypot systems marks a significant evolution in proactive cybersecurity strategies, offering the potential to create highly dynamic, realistic, and adaptive environments capable of deceiving even advanced attackers. In the future, the continued development of more context-aware and lightweight generative architectures, combined with real-time threat intelligence, will likely lead to smarter and more autonomous honeypots that can evolve alongside emerging attack patterns. The application of reinforcement learning and multi-modal generative AI could further enhance deception capabilities by enabling honeypots to learn from attacker behavior and adjust their responses accordingly. Additionally, advances in synthetic data generation and privacy-preserving machine learning techniques may help address current challenges related to data scarcity and model training. However, to fully realize these benefits, future research must also address critical concerns such as model interpretability, consistency, security against adversarial manipulation, and ethical deployment. In conclusion, while generative models are not without limitations, their role in shaping next-generation honeypots is both promising and transformative, offering a new layer of intelligent deception that strengthens cyber defense and threat analysis in an increasingly complex digital landscape. Emerging techniques such as context-aware generative models, which incorporate deeper understanding of system states, attacker profiles, and network environments, will enable honeypots to produce more coherent and believable decoys.

8. REFERENCES

[1] Goodfellow, I.; Pouget-Abadie, J.; Mirza, M.; Xu, B.; Warde-Farley, D.; Ozair, S.; Courville, A.; Bengio, Y. Generative adversarial nets. In Proceedings of the Advances in Neural Information Processing Systems, Montreal, QC, Canada, 8–13 December 2014; pp. 2672–2680.

[2] Goodfellow, I.; Pouget-Abadie, J.; Mirza, M.; Xu, B.; Warde-Farley, D.; Ozair, S.; Courville, A.; Bengio, Y. Generative adversarial networks. *Commun. ACM* **2020**, *63*, 139–144.

[3] Shahriar, S. GAN computers generate arts? A survey on visual arts, music, and literary text generation using generative adversarial network. *Displays* **2022**, *73*, 102237.

[4] Yinka-Banjo, C.; Ugo, O.A. A review of generative adversarial networks and its application in cybersecurity. *Arif. Intel. Rev.* **2020**, *53*, 1721–1736.

[5] Cai, Z.; Xiong, Z.; Xu, H.; Wang, P.; Li, W.; Pan, Y. Generative adversarial networks: A survey toward private and secure applications. *ACM Compute. Surf. (CSUR)* **2021**, *54*, 1–38.

[6] Hu, W.; Tan, Y. Generating adversarial malware examples for black-box attacks based on GAN. In Proceedings of the Data Mining and Big Data: 7th International Conference, DMBD 2022, Beijing, China, 21–24 November 2022; Springer: Berlin/Heidelberg, Germany, 2023; pp. 409–423.

[7] Ng, A.; Jordan, M. On discriminative vs. generative classifiers: A comparison of logistic regression and naive bayes. *Adv. Neural Inf. Process. Syst.* **2002**, *14*, 841.

[8] Lee, H.W.; Lim, K.Y.; Grabowski, B.L. Generative learning: Principles and implications for making meaning. In *Handbook of Research on Educational Communications and Technology*; Routledge: Abingdon, UK, 2008; pp. 111–124.

[9] Alapati, R. Discriminative models for information retrieval. In Proceedings of the 27th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, Sheffield, UK, 25–29 July 2004; pp. 64–71.

[10] Outside, A.; Elhassan, A. Deep generative models: Survey. In Proceedings of the 2018 International Conference on Intelligent Systems and Computer Vision (ISCV), Fez, Morocco, 2–4 April 2018; pp. 1–8.

[11] Webb, G.I. Naïve Bayes. *Encycl. Mach. Learn.* **2010**, *15*, 713–714.

[12] Pearl, J. *Bayesian Networks*; Department of Statistics, UCLA: Calgary, AB, Canada, 2011.