A Project Report

on

Exploring GAN-Based Malware Generation to Evade Black-Box Detection Models

(Integrated MSc. Computer Science)

Subject Code: CSC-503

Submitted by

Mohit Bhatt (20CS-18)

Batch-2020

Submitted to

Dr. Narendra Rawal

Associate Professor

School of Technology

Doon University

Dehradun, Uttarakhand

December, 2024

CERTIFICATE

This is to certify that the project report entitled **"Exploring GAN-Based Malware Generation to Evade Black-Box Detection Models"** submitted by **Mohit Bhatt** (20CS-18) to the Department of Computer Science, School of Technology, Doon University, Dehradun, in partial fulfillment of the requirements for the award of the degree of **M.Sc Hons (Computer Science)**. The results embodied in this project report have not been submitted to any other University or Institute for the award of any Degree.

Dr. Narendra Rawal
(Associate Professor)
Department of Computer Science
School of Technology
Doon University, Dehradun, India

DECLARATION

I hereby declare that the project report entitled **"Exploring GAN-Based Malware Generation to Evade Black-Box Detection Models."** being submitted to the Department of Computer Science, School of Technology, Doon University, Dehradun in partial fulfillment of the requirements for the award of the degree of **M.Sc Hons (Computer Science)** contains the work carried out by me.

Mohit Bhatt (20CS-18)

ACKNOWLEDGEMENT

I would like to express my sincere gratitude to all those who have supported and guided me throughout the course of this report. Your valuable insights and encouragement have been instrumental in shaping my understanding and approach. I am also thankful to the institution for providing me with the necessary resources and environment to carry out this work. This project would not have been possible without the contributions of many, and I am deeply appreciative of everyone who has played a part, directly or indirectly, in this endeavor.

Mohit Bhatt

ABSTRACT

Machine learning has revolutionized malware detection by enabling the identification of malicious activity through pattern analysis in large datasets. However, as these systems become integral to cybersecurity, attackers have developed methods to subvert them, including black-box attacks that bypass detection without knowledge of the model's internal workings. Generative Adversarial Networks (GANs) have emerged as a potent tool for creating adversarial malware capable of evading machine learning-based detection systems. This thesis focuses on MalGAN, a GAN-based algorithm designed to generate adversarial malware that bypasses black-box detection models. Unlike traditional gradient-based methods, MalGAN operates without direct access to the model's parameters, utilizing a substitute detector to approximate the black-box system's behavior. The generator within MalGAN creates malware samples that minimize detection probabilities, significantly reducing detection rates and exposing vulnerabilities in retraining-based defense strategies. This research provides a theoretical exploration of GAN mechanics and their application in adversarial malware generation. By examining the functionality of MalGAN and its implications for cybersecurity, the study highlights the urgent need for innovative defense mechanisms to counter advanced adversarial techniques. The findings underscore the critical importance of developing adaptive and resilient strategies to protect machine learning-based malware detection systems against the evolving threat posed by GANs in adversarial contexts.

Contents

List of Figures

List of Tables

1 Introduction

Machine learning has become a critical tool in the detection of new malware, offering enhanced capabilities for identifying and mitigating evolving cyber threats. These models analyze patterns in large datasets to identify malicious activity, enabling proactive defenses against increasingly sophisticated malware. However, as machine learning models become more integral to cybersecurity, malware authors have developed stronger motivations to subvert these systems. Without access to the internal structures and parameters of these models, attackers are forced to rely on **black-box attacks**[1], in which they manipulate input data to bypass detection without direct knowledge of the model's inner workings. This presents a significant challenge to the security of machine learning-based malware detection systems.

In response to this challenge, **generative adversarial networks (GANs)**[2] have emerged as a promising tool for generating adversarial examples that can deceive machine learning models. GANs consist of two components: a generator that creates synthetic data and a discriminator that evaluates its authenticity. In the context of malware detection, GANs can be trained to generate adversarial malware examples capable of evading black-box detection models. The proposed GAN-based algorithm, **MalGAN**[3], utilizes a substitute detector that approximates the behavior of the black-box detection system. The generator within MalGAN is trained to minimize the malicious probabilities predicted by this substitute detector, effectively creating malware samples that are difficult for the detection model to identify.

MalGAN stands out from traditional gradient-based methods for adversarial example generation by significantly reducing the detection rate of generated malware samples, bringing it close to zero. Unlike earlier approaches, which relied on direct model access for perturbing the inputs to make them adversarial, MalGAN operates

[1]Visit: https://arxiv.org/abs/1905.06776
[2]Visit: https://arxiv.org/abs/1406.2661
[3]Visit: https://github.com/zerofox-oss/malgan

in a black-box setting, where attackers do not have access to the internal structure or parameters of the detection model. This makes MalGAN uniquely capable of creating highly effective adversarial malware that bypasses detection systems without requiring detailed knowledge of the model.

This research provides a theoretical exploration of GAN mechanics and their application in adversarial malware generation. By examining the functionality of MalGAN and its implications for cybersecurity, the study highlights the urgent need for innovative **adaptive defenses**[4] to counter advanced adversarial techniques. The findings underscore the critical importance of developing adaptive and resilient strategies to protect machine learning-based malware detection systems against the evolving threat posed by GANs in adversarial contexts.

2 Literature Review

Commey et al. [1] introduced **EGAN**, an innovative framework that leverages Evolutional Generative Adversarial Networks (GANs) to generate adversarial ransomware capable of bypassing state-of-the-art AI-powered antivirus systems. EGAN combines evolutionary strategies with the generative capabilities of GANs to create ransomware samples that maintain their intended malicious behavior while appearing benign to machine learning-based detection systems. This approach highlights the sophistication of modern adversarial attacks and the challenges faced in cybersecurity, especially as attackers increasingly employ advanced machine learning techniques to outpace detection technologies.

Molloy et al. [2] developed **Ch4os**, a discretized GAN architecture specifically designed for generating functionality-preserving modifications to malware. The key innovation of Ch4os lies in its use of the **Valid Machine Code Execution (VaME)** activation function, which ensures that the adversarial malware samples produced

[4]Visit: https://arxiv.org/abs/2102.08962

by the model remain executable and functional. By discretizing the latent space, Ch4os can optimize malware evasiveness while preserving its original intent, making it highly effective at circumventing traditional malware detection systems.

Li et al. [3] proposed **FGAM**, a framework for generating adversarial malware using gradient-based methods in conjunction with GANs. FGAM incorporates a fast gradient sign method to introduce targeted perturbations to malware samples, making them capable of evading classifiers. The study emphasizes the importance of efficient adversarial sample generation, particularly in environments where time-sensitive adversarial testing is critical for improving the robustness of detection systems.

Sharma et al. [4] introduced **MIGAN**, a GAN-based architecture aimed at synthesizing malware images. This approach involves converting malware binaries into grayscale images, enabling GANs to generate high-quality synthetic malware data for improving malware classification systems. MIGAN effectively addresses class imbalance in training datasets, significantly enhancing the accuracy and reliability of machine learning models for malware detection. This work demonstrates the potential of image-based malware representation as an innovative solution for boosting detection performance.

Zhou et al. [5] proposed an advanced GAN-based framework to generate adversarial malware samples capable of evading network intrusion detection systems (IDS). Their approach focuses on crafting malware representations that appear benign to traditional IDS. By leveraging adversarial machine learning techniques, their model exploits vulnerabilities in IDS classifiers, allowing malicious activities to bypass security measures. This study highlights the importance of developing more adaptive and robust intrusion detection systems capable of defending against adversarial attacks.

Gao et al. [6] explored the use of GANs to generate adversarial malware samples that effectively evade traditional malware detection systems. By introducing subtle perturbations into malware samples, their model manipulates the features in such

a way that the malicious functionality remains intact while appearing benign to classifiers. This work underscores the growing threat of adversarial machine learning in cybersecurity and the necessity of designing detection systems capable of resisting such attacks.

Li et al. [7] examined the potential of GANs to generate adversarial malware examples through perturbation of benign features. Their model demonstrates how slight yet deliberate modifications to feature representations can cause machine learning-based malware detection systems to misclassify malicious samples as benign. Their research highlights the vulnerabilities of detection systems and emphasizes the need for continuous improvements in machine learning algorithms to address adversarial challenges.

Yang et al. [8] investigated the application of GAN-based adversarial attacks to malware detection systems. Their research showed that GANs could create highly effective adversarial perturbations with minimal modifications, enabling malware to bypass classifiers while retaining its malicious functionality. Their findings reveal the vulnerability of machine learning models to adversarial attacks and stress the importance of robust training techniques to counteract these threats.

Xie et al. [9] proposed a novel GAN-based method for generating adversarial malware samples aimed at evading detection systems. By transforming malware features into benign-like representations, their approach uses minimal adversarial modifications that preserve the sample's malicious intent. The study demonstrated that well-trained GANs could create malware variants that evade existing classifiers, revealing critical weaknesses in conventional detection mechanisms.

Shin et al. [10] developed a deep learning-based malware detection system incorporating adversarial examples generated by GANs. By exposing the model to adversarially crafted malware samples during training, they enhanced the system's robustness against evasion attacks. Their study highlights the dual role of GANs in testing the resilience of detection systems while also serving as a tool for improving

model robustness.

Hughes et al. [11] investigated the use of GANs to generate adversarial inputs capable of confusing malware classifiers. Their research demonstrated how GANs could introduce subtle perturbations to malware samples, making them appear benign. This work underscores the need for advanced defense mechanisms to address adversarial vulnerabilities in machine learning-based security systems.

Kolosnjaji et al. [12] introduced **MalGAN**, a groundbreaking framework that utilizes GANs to generate adversarial malware samples capable of evading black-box machine learning-based detection systems. MalGAN modifies malware samples to closely resemble benign data, thereby deceiving classifiers. This pioneering work laid the foundation for leveraging adversarial machine learning in the field of malware generation.

Goodfellow et al. [13] introduced the concept of **Generative Adversarial Networks (GANs)**, a novel machine learning framework consisting of two neural networks: a generator that creates synthetic data and a discriminator that evaluates its authenticity. GANs have since been widely applied across various domains, including image synthesis, data augmentation, and the creation of adversarial examples. Their adversarial nature has also been adapted to cybersecurity, particularly in malware generation and detection.

Chen et al. [14] developed a framework leveraging GANs for data augmentation in malware detection systems. By generating adversarial samples that mimic benign data, their approach improves the diversity of training datasets and enhances the robustness of classifiers against adversarial attacks. This work demonstrates the potential of GANs in improving both the generalization and security of machine learning models in cybersecurity.

3 Generative AI

Generative AI is a field of artificial intelligence that focuses on producing new content, such as text, images, music, or videos, by learning from vast datasets rather than just analyzing or recognizing existing data. It identifies patterns and structures within the data and uses them to create entirely new outputs. Unlike traditional AI, which is primarily used for tasks like classification or prediction, generative AI models are capable of generating novel content. Examples include GPT, which can generate coherent text or code, and DALL-E, which creates images from text descriptions. These models utilize advanced algorithms like Generative Adversarial Networks (GANs), Variational Autoencoders (VAEs), and Transformer models. The applications of generative AI are broad, spanning from creative fields to scientific research, marking a transition in AI's role from understanding to creation [15].

3.1 Algorithm Used in Generative AI

Generative AI leverages advanced machine learning algorithms to create new data that mimics real-world examples. These algorithms enable machines to generate text, images, audio, and even video content, offering vast potential across various domains. One of the most prominent algorithms in this domain is Generative Adversarial Networks (GANs), which have revolutionized the way artificial data is synthesized.

3.1.1 Generative Adversarial Networks (GANs)

Generative Adversarial Networks (GANs) involve two neural networks, a generator and a discriminator, which work against each other. The generator creates artificial data, such as images, while the discriminator assesses whether the data is real or generated. Both networks train simultaneously: the generator attempts to deceive the discriminator, while the discriminator improves its ability to detect fake data. Over time, the generator becomes more skilled at producing realistic outputs. GANs

are widely applied in areas like image generation, video production, and even deepfake creation. Their innovative framework enables continuous improvement, pushing the boundaries of generative modeling. As a result, GANs are becoming increasingly essential in creative industries and research, driving advancements in artificial intelligence [13].

3.1.2 Variational Autoencoders (VAEs)

Variational Autoencoders (VAEs) are generative models that use an encoder-decoder architecture. The encoder compresses input data, such as an image, into a smaller, latent space representation, while the decoder reconstructs the original data from this compressed form. What sets VAEs apart is their ability to generate new data by sampling from the latent space, ensuring small changes in this space result in meaningful variations in the output. VAEs are commonly used for creating images, music, and other structured data, especially for producing smooth and continuous variations in the content. Their probabilistic nature also allows for the exploration of diverse outputs, making them valuable for tasks that require creativity. As a result, VAEs have found applications in various fields, including art generation, data augmentation, and even drug discovery [16].

3.1.3 Transformer Models

Transformer models, like GPT (Generative Pre-trained Transformer), are designed for processing sequential data, such as text. They employ an "attention" mechanism to focus on relevant parts of input sequences when generating or interpreting language. Unlike older models that process text word by word, transformers handle entire sequences in parallel, improving efficiency and speeding up training times. GPT models are trained on massive datasets to learn language patterns, allowing them to predict the next word in a sequence based on prior context. This capability makes transformers highly effective for generating human-like text, popular in

applications such as chatbots, story writing, and coding. Their versatility extends beyond text generation, finding uses in translation, summarization, and even image generation, showcasing the breadth of their potential [17].

3.1.4 Diffusion Models

Diffusion models work by learning to reverse a process that gradually adds noise to data, like images. During training, these models are exposed to noisy data and learn how to reconstruct the original data from it, effectively understanding the underlying structure of the data. Once trained, diffusion models can start with random noise and iteratively refine it to generate new, meaningful content. These models have become prominent in generating high-quality images, as seen in tools like DALL-E. By progressively denoising random input, diffusion models can produce detailed and coherent images from text descriptions or other inputs. Their ability to create intricate designs and artistic visuals has made them valuable tools in creative fields, offering exciting possibilities for artists and designers alike [18].

3.2 Introduction to GAN and Its Applications

Generative Adversarial Networks (GANs) are a class of machine learning models that consist of two neural networks: the generator and the discriminator, commonly implemented using deep neural networks (DNNs) or convolutional neural networks (CNNs). The generator's role is to create synthetic data, such as images, text or malware, that closely mimics real data, while the discriminator's function is to evaluate whether the input data is real or generated. As the generator learns to produce more realistic data, the discriminator simultaneously enhances its ability to distinguish between real and fake samples. This adversarial dynamic fosters continuous improvement in both networks— the generator strives to fool the discriminator, and the discriminator works to detect fake data accurately.

In the field of malware analysis, GANs are utilized to enhance detection and

Table 1: Applications of MalGANs

Application Area	Description
Evasion of Detection Systems	MalGANs generate adversarial malware samples that are specifically designed to evade traditional machine learning-based detection systems. This showcases the vulnerability of existing classifiers to adversarial attacks.
Testing Robustness of Security Models	Security researchers use MalGAN-generated samples to test and improve the robustness of malware detection systems, ensuring resilience against adversarial threats.
Adversarial Training	MalGANs assist in creating diverse adversarial datasets that can be used for adversarial training, enabling classifiers to learn to detect and resist adversarial malware.
Feature Analysis	By observing the transformations applied by MalGANs to malware features, researchers can identify weak points in detection systems and improve feature engineering for classifiers.
Enhancing Malware Evolution Strategies	MalGANs can contribute to the design of malware that evolves to bypass defenses, simulating real-world scenarios for testing security solutions.
Generating Synthetic Malware Datasets	MalGANs generate synthetic malware samples for augmenting training datasets, especially in scenarios with imbalanced data or limited access to real-world malware samples.
Network Intrusion Simulation	MalGANs help in creating sophisticated network intrusion scenarios by generating adversarial payloads that mimic benign behavior while retaining malicious intent.

evasion techniques effectively. The generator, typically a deep neural network, is trained to produce adversarial malware samples that resemble real malware while being specifically designed to evade detection by cybersecurity systems. The discriminator, often a convolutional neural network, is tasked with differentiating between genuine malware, generated adversarial samples, and benign software. This adversarial training enables the generator to become proficient at crafting malware that can bypass security measures, while the discriminator improves its capacity to identify both legitimate and adversarial malware. This ongoing interaction improves malware detection systems by equipping them to recognize evolving threats, making GANs a dynamic tool in cybersecurity that not only enhances the detection of existing malware but also improves defenses against new, sophisticated attacks.

3.2.1 Architecture of GAN

The discriminator D first receives real data samples, such as images or malware, and assesses whether they are real or fake, outputting a probability score. Simultaneously, the generator G takes a random noise vector as input, sampled from a simple distribution (like Gaussian or uniform), and tries to convert this noise into synthetic data that resembles real data. This fake data is then fed into the discriminator along with the real samples, where the discriminator attempts to distinguish between the two. During training, the generator continuously improves at creating realistic data to deceive the discriminator, while the discriminator improves its ability to identify fake data. This adversarial process drives both networks to improve, with the generator refining its outputs and the discriminator becoming more adept at detection.

In this expression, each symbol represents critical components of the adversarial learning process. The $\min_G \max_D$ highlights the adversarial nature of the training, where the generator G tries to minimize the objective while the discriminator D attempts to maximize it, reflecting their opposing goals. $\mathbb{E}_{x \sim p_{\text{data}}(x)}[\log D(x)]$ refers

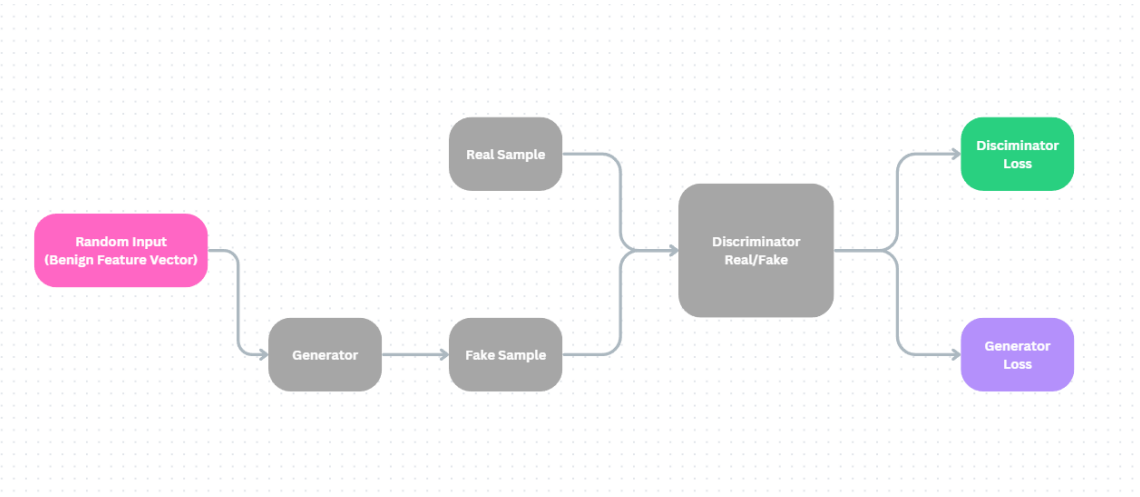

Figure 1: Architecture of GAN.

to the expected value over real data samples x drawn from the true data distribution $p_{\text{data}}(x)$, with $\log D(x)$ indicating the probability that the discriminator correctly identifies the real data. On the other hand, $\mathbb{E}_{z\sim p_z(z)}[\log(1-D(G(z)))]$ represents the expected value over latent variables z, from which the generator creates fake data $G(z)$. The term $\log(1-D(G(z)))$ signifies the probability that the discriminator correctly identifies the generated data as fake. This adversarial dynamic pushes the generator to create more realistic outputs while improving the discriminator's ability to differentiate between real and fake data. The training process is framed as a minimax game, represented by the equation:

$$\min_G \max_D \left(\mathbb{E}_{x\sim p_{\text{data}}(x)}[\log D(x)] + \mathbb{E}_{z\sim p_z(z)}[\log(1-D(G(z)))]\right) \tag{1}$$

3.2.2 Generator Loss

In the formula for the generator's loss, L_G represents the generator's loss, which tells us how well the generator is performing. A lower value for L_G means the generator is creating data that is harder for the discriminator to distinguish from real data. The

symbol $D(G(z))$ is the output of the discriminator when given the generator's fake data, $G(z)$, as input. Here, G represents the generator itself, and z is a random input, often called latent noise, that the generator uses to create a fake sample. $D(G(z))$ reflects the discriminator's confidence that the generated sample is real, with values closer to 1 meaning it believes the data is real and values closer to 0 meaning it thinks the data is fake. The $\log(D(G(z)))$ helps in penalizing the generator more as the discriminator gets better at identifying fakes, encouraging the generator to improve further.

$$L_G = \mathbb{E}_{z \sim p_z(z)}[\log(D(G(z)))] \tag{2}$$

3.2.3 Discriminator Loss

The discriminator's loss in a GAN measures how well it distinguishes between real and fake data. Its task is to correctly classify real data from the dataset, denoted as x, as real, which is represented by $D(x)$, the probability that the discriminator assigns to the real sample being genuine. Simultaneously, the discriminator must identify the generated data, represented as $G(z)$, where z is the random input that the generator uses to produce fake samples. The term $D(G(z))$ indicates the probability that the discriminator assigns to the generated data being real. The discriminator's loss is composed of two components: the first part, $-\log(D(x))$, penalizes the discriminator for misclassifying real data, while the second part, $-\log(1 - D(G(z)))$, penalizes it for incorrectly classifying generated data as real. Therefore, the formula for the discriminator's loss is expressed as:

$$L_D = -\mathbb{E}_{x \sim p_{\text{data}}(x)}[\log D(x)] + \mathbb{E}_{z \sim p_z(z)}[\log(1 - D(G(z)))] \tag{3}$$

3.3 Improvement of GAN by substitute model (MalGAN)

MalGAN [12] is a GAN-based algorithm specifically designed to generate adversarial malware examples that can successfully evade black-box machine learning detection systems. By leveraging a substitute detector to imitate the black-box model's behavior, the generator is trained to create malware that bypasses detection, significantly lowering the detection rate. Traditional GAN architectures in malware analysis face limitations, particularly when dealing with black-box models where gradient-based methods fail due to the lack of access to internal parameters. Additionally, adversarial examples generated by conventional GANs are often countered by retraining-based defenses, which can restore detection efficacy. MalGAN addresses these challenges by using a substitute model to effectively simulate the black-box system without needing direct access to its internals. This approach not only improves the evasion of malware detection but also makes defenses harder to implement. The motivation behind MalGAN lies in its ability to refine adversarial attacks, pushing the boundaries of malware evasion and demonstrating the power of GANs in cybersecurity .

3.3.1 Architecture of MalGAN

MalGAN's architecture is built to create adversarial malware examples capable of bypassing black-box machine learning detection systems. It comprises two key components: the generator, which modifies malware feature vectors by introducing noise, and the substitute detector, which mimics the behavior of the target black-box detector. The generator receives malware feature vectors along with random noise, producing adversarial variants by adding irrelevant features without altering the malware's core functionality. Unlike traditional methods, MalGAN relies on dynamic feedback from the black-box detector to refine its output, rather than using static gradient-based approaches.

A smooth function enables gradient information to pass through, allowing the

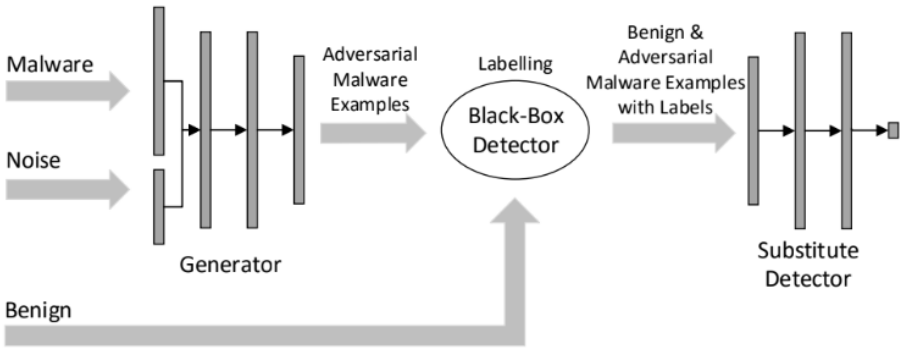

Figure 2: Architecture of MalGAN.

generator to learn effectively even with binary features. The substitute detector is trained on adversarial examples and benign data, simulating the black-box detector's classifications, and guiding the generator to evade detection. Through this iterative process, MalGAN produces highly evasive malware that is harder for detection systems to classify.

3.3.2 Generator

The generator transforms a malware feature vector m into an adversarial version by concatenating it with a noise vector z. m is an M-dimensional binary vector, and z is a Z-dimensional vector of random values sampled from a uniform distribution $[0, 1)$. The generator uses a multi-layer neural network with weights θ_g, where the output layer has M neurons and a sigmoid activation function. The output, denoted o, is binarized to o_0 by setting elements greater than 0.5 to 1. To generate adversarial examples, irrelevant features are added to the original malware, represented by $m_0 = m \mid o_0$, where $\mid$ is the element-wise binary OR operation. Since malware features are binary, gradients cannot backpropagate from the substitute detector to the generator. To allow this, a smooth function G is defined:

$$G_{\theta_g}(m, z) = \max(m, o) \tag{4}$$

If m has a value of 1, the result of G is also 1, blocking gradients. If m is 0, the output of G is the real value from the neural network, allowing gradient flow. The adversarial example m_0 is the binarized result of $G_{\theta_g}(m, z)$.

3.3.3 Substitute Detector

Since malware authors have no knowledge of the detailed structure of the black-box detector, a substitute detector is employed to approximate the black-box detector and provide gradient information for training the generator. The substitute detector is a multi-layer feed-forward neural network with weights θ_d, which takes a program feature vector x as input. It classifies the program as either benign or malware. The predicted probability that x is malware is denoted as $D_{\theta_d}(x)$. The training data for the substitute detector comprises adversarial malware examples generated by the generator and benign programs obtained from an additional benign dataset collected by the malware authors. The ground-truth labels of the training data are not utilized for training the substitute detector. Instead, the substitute detector aims to replicate the behavior of the black-box detector. The black-box detector first analyzes the training data and outputs whether a program is benign or malware. These predicted labels are then used to train the substitute detector.

3.4 Training MalGAN

To train MalGAN, malware authors must first collect a malware dataset and a benign dataset. The loss function of the substitute detector is defined as follows:

$$L_D = -\mathbb{E}_{x \in BB_{\text{Benign}}} \log\left(1 - D_{\theta_d}(x)\right) - \mathbb{E}_{x \in BB_{\text{Malware}}} \log D_{\theta_d}(x). \tag{5}$$

Here, BB_{Benign} represents the set of programs recognized as benign by the black-box detector, and BB_{Malware} represents the set of programs detected as malware

by the black-box detector. To train the substitute detector, L_D is minimized with respect to the weights of the substitute detector. The loss function of the generator is defined as:

$$L_G = \mathbb{E}_{m \in S_{\text{Malware}}, z \sim p_{\text{uniform}}[0,1)} \log D_{\theta_d}\big(G_{\theta_g}(m, z)\big). \tag{6}$$

Here, S_{Malware} is the actual malware dataset, not the malware dataset labeled by the black-box detector. L_G is minimized with respect to the weights of the generator. Minimizing L_G reduces the predicted malicious probability of malware, compelling the substitute detector to classify malware as benign. Since the substitute detector seeks to mimic the black-box detector, training the generator further misleads the black-box detector.

4 Objectives of MalGAN

The primary objectives of MalGAN, along with detailed explanations, are as follows:

1. **Generate Adversarial Malware:**
   MalGAN aims to generate adversarial malware samples by modifying existing malware in a way that allows them to evade detection by black-box malware detectors. These modifications are crafted such that they do not disrupt the malware's intended malicious functionality. The generated samples help malware authors bypass security mechanisms [19].

2. **Approximate Black-Box Detector:**
   Since the internal structure, parameters, and decision logic of the black-box detector are unknown to malware authors, MalGAN uses a substitute detector. This substitute detector is a trainable model that approximates the behavior of the black-box detector. By leveraging the black-box detector's predictions on a given dataset, the substitute detector learns to mimic its classification patterns, enabling gradient-based optimizations to be performed indirectly [20].

3. **Minimize Malicious Probability:**
   MalGAN is designed to reduce the probability that a malware sample is classified as malicious by the black-box detector. This is achieved by training the generator to create samples that the substitute detector (and consequently the black-box detector) misclassifies as benign. This objective ensures that adversarial malware can evade detection with high reliability [21].

4. **Fool the Black-Box Detector:**
   MalGAN trains the generator and substitute detector in an iterative manner, where the substitute detector adapts to better mimic the black-box detector and the generator creates samples to mislead the substitute detector. Since the substitute detector approximates the black-box detector, the generated adversarial malware ultimately fools the black-box detector as well [22].

5. **Preserve Malware Functionality:**
   One of MalGAN's key goals is to ensure that the generated adversarial malware retains its original malicious functionality. This objective is crucial because any modification to evade detection must not alter the core behavior of the malware, as this would render the attack ineffective [23].

5 Challenges in MalGANs

Despite the innovative capabilities of MalGANs in generating adversarial malware samples, several challenges remain that hinder their broader application and effectiveness:

1. Limited Generalizability

MalGANs often face difficulties in generalizing across diverse malware types and detection systems. The adversarial samples generated by MalGANs are typically

tailored to bypass specific classifiers, making them less effective against other, unseen detection systems. This limitation underscores the need for models that can adapt to a wide range of security mechanisms.

2. Preservation of Malware Functionality

One critical challenge for MalGANs is ensuring that adversarial modifications do not compromise the original functionality of the malware. While creating adversarial samples, there is a risk that the alterations may render the malware ineffective, defeating the purpose of the attack.

3. Detection of Adversarial Samples

As adversarial techniques become more sophisticated, detection systems are evolving to identify adversarial patterns. Advanced anomaly detection and adversarial training techniques are being integrated into modern security systems, making it increasingly difficult for MalGANs to evade detection.

4. Computational Complexity

Training MalGANs is computationally intensive, as it involves iterative optimization of both the generator and the discriminator. The high computational cost can limit the scalability and practicality of MalGANs in real-world scenarios.

5. Ethical and Security Concerns

The use of MalGANs raises significant ethical questions and security concerns. While they can be used to test and improve detection systems, they also provide tools that malicious actors can exploit to enhance their attacks. Balancing the dual-use nature of MalGANs is an ongoing challenge for researchers.

6. Robustness of Target Systems

As detection systems incorporate adversarial training and ensemble learning, they become increasingly robust to adversarial attacks. This escalation creates a constant arms race between adversarial sample generation and detection system improvements, requiring MalGANs to evolve continuously.

7. Interpretability and Explainability

The black-box nature of GANs, including MalGANs, poses challenges in understanding and interpreting the adversarial modifications. This lack of transparency makes it difficult to analyze the effectiveness and limitations of MalGAN-generated samples.

8. Dataset Limitations

MalGANs rely heavily on high-quality labeled datasets for training. However, obtaining diverse and comprehensive malware datasets can be challenging due to privacy concerns, legal restrictions, and the evolving nature of malware.

Addressing these challenges is critical for advancing the field of adversarial machine learning and enhancing the utility of MalGANs in cybersecurity research.

6 Conclusion

The study explores how synthetic malware samples, generated through advanced techniques like Generative Adversarial Networks (GANs), can deceive machine learning models into misclassifying them as benign while maintaining their original malicious functionality. This process exploits the vulnerabilities in the decision boundaries of detection models, significantly reducing their robustness and reliability. By using GANs for malware generation, attackers can fine-tune adversarial features in the malware, ensuring it evades detection systems without compromising its harmful intent. These synthetic adversarial samples are designed to appear non-threatening to detection algorithms, thereby bypassing traditional security mechanisms.

GANs achieve this by leveraging a generator model to create deceptive samples and a substitute detector to approximate the behavior of the target black-box model. The substitute detector guides the generator in crafting samples that mimic benign characteristics, further fooling the black-box detector. This iterative process refines the adversarial samples, making them highly effective against state-of-the-art malware detection systems. The study emphasizes that this approach not only highlights the weaknesses of existing models but also demonstrates the advanced capabilities of GANs in adversarial data generation. It underscores the urgent need for robust defense mechanisms that can withstand such sophisticated attacks. Without these improvements, machine learning-based malware detection systems remain highly susceptible to adversarial manipulation, posing a significant challenge to cybersecurity.

7 Future Work

The future work for this project involves several key advancements aimed at enhancing the implementation and optimization of MalGAN. Firstly, the practical implementation of MalGAN will be undertaken, focusing on generating adversarial malware samples and evaluating their impact on the accuracy of a pre-trained malware detection model. Specifically, experiments will measure the reduction in detection accuracy when perturbed samples are introduced to the black-box detector. Another crucial area of future work is optimizing the MalGAN architecture to improve its efficiency in generating synthetic adversarial samples. This optimization will focus on reducing the number of iterations required for convergence during training, thereby improving computational efficiency. Techniques such as adaptive learning rates, improved loss functions, and fine-tuning of hyperparameters will be explored to achieve faster and more effective training.

Additionally, future research will investigate the robustness of the generated adversarial samples across different types of black-box detectors. This includes evaluating their ability to evade detection by models with varying architectures and training data. Expanding the scope of experiments to include diverse malware families and real-world datasets will further validate the effectiveness and applicability of MalGAN. Lastly, the work will explore integrating advanced adversarial defense mechanisms into the training process of the black-box models. This will provide insights into how detection systems can adapt to and mitigate the threats posed by adversarial malware, contributing to a more secure and resilient cybersecurity landscape.

References

[1] T. Commey *et al.*, "Egan: Evolutional gan for adversarial ransomware generation," *Journal of Cybersecurity*, 2024.

[2] R. Molloy *et al.*, "Ch4os: Discretized gan for malware evasion," in *Proceedings of the IEEE Symposium on Security and Privacy*, 2024.

[3] X. Li *et al.*, "Fgam: Gradient sign adversarial malware generation," *arXiv preprint*, 2023. arXiv:2305.12770.

[4] K. Sharma *et al.*, "Migan: Malware image synthesis using gans," *Expert Systems with Applications*, vol. 207, p. 122678, 2023.

[5] Y. Zhou, Y. Li, and L. Zhang, "Evasion of network intrusion detection systems using gan," in *Proceedings of the 2021 IEEE International Conference on Artificial Intelligence and Cybersecurity (AICS)*, 2021.

[6] J. Gao, F. Li, and Y. Zhang, "Adversarial malware generation with gans for evasion attacks," in *Proceedings of the 2020 IEEE Conference on Security and Privacy (SP)*, 2020.

[7] X. Li, J. Zhang, and H. Wang, "Generating adversarial malware examples using gan," in *Proceedings of the 2020 IEEE International Conference on Cybersecurity (CYBERSEC)*, 2020.

[8] F. Yang, C. Liu, and Z. Sun, "Gan-based adversarial attacks on malware detection systems," in *Proceedings of the 2019 International Conference on Artificial Intelligence and Security (AISec)*, 2019.

[9] L. Xie, X. Zhang, and X. Liu, "A gan-based method for generating adversarial malware samples," in *Proceedings of the 2019 IEEE Conference on Communications and Network Security (CNS)*, 2019.

[10] K. Shin, J. Song, S. Lee, and H. Kim, "Deep learning for malware detection with gan," in *Proceedings of the 2018 ACM SIGSAC Conference on Computer and Communications Security (CCS)*, 2018.

[11] R. Hughes, M. Blask, and D. Ward, "Adversarial machine learning for malware detection using gan," in *Proceedings of the 2018 International Conference on Machine Learning and Cybernetics (ICMLC)*, 2018.

[12] B. Kolosnjaji, K. Krombholz, P. Laskov, and E. Kirda, "Malgan: A gan-based malware generation framework for evasion," in *Proceedings of the 2018 European Symposium on Research in Computer Security (ESORICS)*, 2018.

[13] I. J. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, and Y. Bengio, "Generative adversarial nets," in *Advances in Neural Information Processing Systems (NeurIPS), 27*, 2014.

[14] T. Chen, Y. Zhang, and M. Luo, "Data augmentation for malware detection using gan," in *Proceedings of the 2019 IEEE International Conference on Big Data (BigData)*, 2019.

[15] OpenAI, "Generative ai overview," 2024. Accessed: 2024-12-28.

[16] D. Kingma and M. Welling, "Auto-encoding variational bayes," *arXiv preprint arXiv:1312.6114*, 2013.

[17] A. Vaswani *et al.*, "Attention is all you need," *arXiv preprint arXiv:1706.03762*, 2017.

[18] J. Sohl-Dickstein *et al.*, "Deep unsupervised learning using nonequilibrium thermodynamics," *arXiv preprint arXiv:1503.03585*, 2015.

[19] Y. Hu and H. Tan, "Malgan: Malware generation and detection with generative adversarial networks," in *Proceedings of the ACM Conference on Computer and Communications Security (CCS)*, 2017.

[20] N. Papernot, P. McDaniel, X. Wu, S. Jha, and A. Swami, "Practical black-box attacks against machine learning," in *Proceedings of the ACM Asia Conference on Computer and Communications Security (AsiaCCS)*, 2016.

[21] I. Goodfellow, J. Shlens, and C. Szegedy, "Explaining and harnessing adversarial examples," *arXiv preprint arXiv:1412.6572*, 2014.

[22] W. Hu and Y. Tan, "Generating adversarial malware examples for black-box attacks based on gan," *arXiv preprint arXiv:1702.05983*, 2017.

[23] H. Xu, J. Wang, and C. J. Long, "Automatically evading classifiers: A case study on pdf malware classifiers," in *Network and Distributed System Security (NDSS) Symposium*, 2016.