WEP Warwick Evans Publishing

International Journal of Computer Science and Information Technology

ISSN: 3005-9682 (Print), ISSN: 3005-7140 (Online) | Volume 2, Number 2, Year 2024
DOI: https://doi.org/10.62051/ijcsit.v2n2.10
Journal homepage: https://wepub.org/index.php/IJCSIT/index

Classification of Malicious Document Detection Based on Artificial Intelligence

Xingyu Wang

Southwest Jiaotong University, Chengdu, 610031, China

ABSTRACT

Malicious document attacks are one of the severe threats in the current field of cybersecurity. This paper compares traditional and artificial intelligence methods in detecting malicious documents and proposes an application and technical framework of artificial intelligence in malicious document detection (AI-DDNet). First, it introduces various methods of malicious file attacks, including malicious code attacks, object embedding attacks, document vulnerability attacks, and remote link attacks. Then, it systematically elaborates on traditional static and dynamic analysis methods, as well as static and dynamic analysis techniques integrated with artificial intelligence, including machine learning, deep learning, and reinforcement learning. Lastly, it summarizes the challenges and unresolved issues of current technology and looks forward to the future development direction of artificial intelligence in malicious document detection. Through this study, it promotes the innovation and development of cybersecurity technology.

KEYWORDS

Cybersecurity, Document Detection, Artificial Intelligence, Machine Learning

1. INTRODUCTION

Malicious document attacks are a covert and destructive cybersecurity threat, essentially using document files as carriers to achieve the attack's purpose through methods like embedding malicious code, object embedding, exploiting vulnerabilities, or remote links. This attack method has a long history, tracing back to the 1990s. In 1993, Microsoft introduced the Visual Basic for Applications (VBA) language for Office document development, soon leading to the emergence of macro viruses, marking the beginning of document attacks. The Melissa macro virus incident in 1999 created a new attack vector, sparking widespread attention to malicious document attacks.

Currently, malicious document attacks have become a highly covert method widely used in cybercriminal activities such as APT (Advanced Persistent Threat) attacks. Attackers embed malicious code into documents using vulnerabilities or social engineering tactics, then spread them through emails, phishing sites, etc. Once these documents are opened by users, it may lead to system infection, sensitive data leakage, system paralysis, and other severe consequences. This highly covert attack method poses a great threat and risk to personal, corporate, and even national security.

Malicious document attacks are a highly covert and continuously evolving cybersecurity threat, facing multiple unresolved key issues.

Researching malicious document attacks is of significant practical importance. Firstly, malicious document attacks have become a serious threat in the field of cybersecurity, causing severe impacts on personal, corporate, and national security. Secondly, traditional antivirus technologies often struggle to timely and effectively detect and prevent malicious document attacks. Therefore, it is

necessary to deeply study the characteristics, techniques, and defense methods of malicious document attacks to enhance the level of cybersecurity and better protect the safety of the cyber environment.

This paper will explore the challenges of malicious document attacks through an in-depth study of malicious file attack methods and technology frameworks, providing a theoretical foundation and technical support for future research and defense.

2. CLASSIFICATION OF ARTIFICIAL INTELLIGENCE-BASED MALICIOUS DOCUMENT DETECTION.

2.1. Technical Approaches for Traditional Detection Methods

2.1.1. Traditional Static Detection

Traditional static detection methods mainly include statistical analysis, which regards the document as a continuous sequence of bytes and extracts continuous bytes as features for analysis. For example, Schultz et al., in 2001, extracted byte sequence features from malicious payloads and used a Bayesian classifier to categorize documents' maliciousness [1]. Another common approach utilizes n-grams as analysis features and calculates the Markov distance of n-grams as the basis for determining malicious documents. Literature further applied the 5-gram model for malicious document detection, considering the impact of different sectors of DOC documents on detection results, assigning different weights to different sectors [2]. Literature calculated the Markov chain entropy rate of documents using 5-grams and used a Bloom filter for document judgment. However, these methods might produce false positives in certain cases, such as detecting PDF and Office documents, because these documents support object embedding [3]. Literature proposed new solutions to the limitations of statistical analysis models, focusing more on the extraction and model training of features with greater information gain [4].

Detection methods based on structural features often analyze the internal structure of documents more deeply. For instance, Smutz and others extracted metadata and document structure features from PDF documents to establish a more accurate classifier, effectively countering mimicry attacks [4]. Šrndic and others proposed a feature representation method based on the structural paths of PDF documents, using decision trees and SVM for malicious classification of documents [5]. When Maiorca et al. extracted document structure and metadata features, they focused more on the frequency of features and the scenarios of embedding malicious files [6]. Additionally, malicious code detection methods based on active learning have gradually gained attention, such as the ALDOCX framework for Word document detection proposed by Nissim et al. [7]. However, it's important to note that analysis based solely on structural features might be vulnerable to mimicry attacks, affecting the effectiveness of detection.

2.1.2. Traditional Dynamic Detection

Traditional dynamic detection methods primarily detect malicious code by simulating the document's execution in a virtual environment. Early research, such as the method proposed by Toth et al., focused on monitoring changes in file operations, network communications, and other behaviors [8]. The CWSandbox model determines maliciousness by recording runtime behavior features, effectively bypassing the difficulties of static analysis of malicious code, but at a higher cost. Additionally, research has tried strategies like changing the order of DLL loading, observing registry changes, etc., to determine malicious documents, but these methods have high false positive rates due to incomplete dynamic features. Subsequent research enhanced detection by modifying some byte values in the document's data part and monitoring the system's reaction. Although it could resist some attacks, it was ineffective against macro viruses. Dynamic detection is robust but resource-intensive and susceptible to anti-virtual machine techniques, facing challenges in triggering the execution of malicious code.

2.2. Integration of Artificial Intelligence in Static Analysis

2.2.1. Static Detection Methods for Machine Learning:

The content of a document can serve as a key factor in detecting its maliciousness. The method proposed by Supu Rui and others is mainly based on vector space calculations, determining the maliciousness of a document by comparing suspected Shellcode fragments with known Shellcode. However, this method has limitations in locating Shellcode and difficulties in detecting new types of attacks [9]. In 2011, Laskov and others proposed PJScan, a model for static detection of malicious PDF documents based on JavaScript [10]. It uses the SpiderMonkey engine's intermediate language to describe the features of JavaScript code. However, it still has limitations in handling JavaScript code obfuscation. Maiorca and others proposed a detection method based on document content and keywords in 2012, selecting PDF metadata as features and using classification algorithms for classification. This method avoids the limitations of targeting Shellcode and JavaScript but lacks in selecting metadata features. Research in 2013 proposed classification based on three features of document content, including functional keywords, habitual words, and constant segments, and used TF-IDF to normalize the feature vectors, but it still had limited handling of obfuscated documents.

Another static method integrating machine learning is based on multi-layer abstraction for malicious document detection, aiming to research a unified expression form for different types of documents to achieve unified detection of various documents. This method first classifies and parses documents. Then, the document structure is abstracted into a tree structure and vectorized for calculation. Moreover, scripting languages are also abstracted, essentially a unified expression of document content features. Then, the extracted data are vectorized for quantitative calculation. Although the unified detection method based on abstraction avoids analyzing embedded code in feature selection, improving universality and detection accuracy, it mostly selects superficial features and does not touch the essence of malicious documents, thus lacking in detecting mimic attacks.

2.2.2. Static Detection Methods for Deep Learning

In the field of research where deep learning combined with static analysis is used for malicious document detection, specifically for PDF documents, there are some specific research methods and results. A prominent example is the malicious PDF document detection scheme based on hybrid features, which uses static analysis techniques to extract conventional information (such as version number, size, etc.) and structural information from PDF documents, combined with dynamic analysis techniques to extract API call information during document execution [11].This information, after filtering and clustering, forms the feature vector for maliciousness determination, ultimately validated by a classifier built using the random forest algorithm. This method significantly improves detection rates and reduces false positives compared to existing technologies and can effectively counter feature-addition mimic attacks.

On the other hand, research on malicious PDF document detection also covers in-depth analysis of the PDF document structure, including detailed introductions to the file header, body, cross-reference table, and tail. These structural features are crucial for understanding the construction of PDF documents and potential security risks. The development of malicious document detection technology, especially methods combining static and dynamic analysis, is continuously advancing, aiming to improve the detection accuracy of malware and attacks.

2.2.3. Static Detection Methods for Reinforcement Learning

Machine learning typically requires a large amount of labeled data to train models, while deep learning uses deep neural networks to process high-level features of data. In contrast, reinforcement learning does not rely on pre-labeled datasets but optimizes the decision-making process through exploration and exploitation strategies. The goal of reinforcement learning is to learn a policy to maximize cumulative rewards in a certain environment. This method makes reinforcement learning

particularly powerful in dealing with problems that require continuous action decisions, as it can consider future rewards in a long-term sequence of decisions.

Reinforcement learning can learn without explicit labels. Unlike traditional machine learning and deep learning methods, reinforcement learning learns by interacting with the environment, thereby finding the strategy to obtain the maximum reward in a given task. This way of learning makes reinforcement learning very suitable for solving decision-making process problems. Moreover, reinforcement learning can continuously adapt to changes in the environment, making the generated strategies more robust and adaptable.

In research combining reinforcement learning and static analysis for malicious document detection, various obfuscation strategies are designed, and the agent (i.e., the reinforcement learning model) interacts with the detection engine in a series of interactions, enabling the reinforcement learning model to learn how to bypass the static detection engine. This method is mainly aimed at the detection of malicious software in static executable files (such as PE files), where the techniques used include, but are not limited to, adding functions to unused import address tables, manipulating existing section names, creating new (unused) sections, appending bytes to extra space at the end of sections, creating a new entry point, etc. These actions aim to change the PE file format without altering the normal execution function of the PE file, thereby bypassing machine learning-based static malware detection models.

For example, one article showed how to use reinforcement learning to automatically bypass static file detection engines, improving the success rate of malicious software bypassing static detection, and the number of mutation attempts required for bypassing using reinforcement learning is less than the success rate of random mutations [12]. Another related study further improved the escape rate of malicious samples generated using RL by adding more actions and improving the reinforcement learning model [13].

Reinforcement learning has broad application potential in the field of malicious document detection, especially in improving the robustness of malware detection models. Adversarial samples generated through reinforcement learning can not only bypass existing detection models but also be used to enhance the performance of detection models, thereby improving the detection capability for unknown malicious software samples. These advancements provide new ideas and methods for the development of malicious document detection technology.

2.3. Dynamics of Integrating Artificial Intelligence

2.3.1. Machine Learning-Oriented Dynamic Detection Methods

For the detection of malicious documents, especially PDF files, combining machine learning with dynamic detection techniques provides an effective solution. This method extracts the basic and structural information of PDF documents through static analysis techniques, while utilizing dynamic analysis techniques to obtain the API call information when the document is executed. By this mixed feature approach, it is possible to more accurately identify and classify malicious documents, improving the detection rate and reducing the false positive rate. Using the K-means clustering algorithm to cluster features and the random forest algorithm to build classifiers, this method has proven its effectiveness in experiments, especially in terms of performance against feature addition-based mimicry attacks [14].

On the other hand, research on malicious sample behavior detection based on deep learning adopts Convolutional Neural Network (CNN) algorithms to process samples' dynamic behavior information [15]. This method does not require manual extraction of feature vectors, as the algorithm can learn features based on the sample's dynamic behavior autonomously. Dynamic behavior reports are obtained by running samples in a sandbox, and these reports are converted into text format for processing. Using CNN for text classification, dynamic behaviors are converted into sequences of

token IDs from the vocabulary, which are then transformed into a two-dimensional matrix for training. This method is capable of effectively performing binary and multi-classification of malicious samples and achieves high accuracy rates on both training and testing sets.

2.3.2. Deep Learning-Oriented Dynamic Detection Methods

Deep learning combined with dynamic detection has shown significant advantages in the identification of malicious software and documents. By utilizing a combination of Convolutional Neural Networks (CNN) and Long Short-Term Memory networks (LSTM), researchers can extract key features from complex API call sequences and learn the behavior patterns of malicious samples. This method does not require manual extraction of feature vectors, as deep learning models can directly learn features from samples' dynamic behavior information.

In the data preprocessing phase, by running samples in a sandbox environment and obtaining dynamic behavior reports, researchers can convert the original dynamic behavior reports into text format, simplifying subsequent processing. Each sample's behavior is represented through a text document, containing the type, name, and parameters of API calls. However, to improve algorithm efficiency, parameter information is often ignored, and deduplication is performed for adjacent, repeated API calls.

In the specific implementation of deep learning algorithms, it's necessary first to build a vocabulary from the dynamic behavior text and convert each dynamic behavior into a unique identifier. Then, the CNN model is used to train the samples, including using convolutional kernels of different lengths to simulate the N-Gram method for extracting features of adjacent dynamic behaviors. Finally, softmax is used for multi-classification to identify whether a sample is malicious and its category of malice.

Research has demonstrated the effectiveness of deep learning models in malware detection, especially in handling large volumes of complex data and identifying nuanced malicious behavior patterns. Through extensive experiments and data analysis, this method not only surpasses traditional machine learning solutions but also provides valuable insights into how feature and model architecture design can improve generalization performance [16].

3. CONCLUSION

With the rapid development of information technology, the digital ecosystem faces unprecedented threats, especially the surge in malicious file attacks, posing a serious challenge to the data security of individuals and organizations. Traditional malicious file detection methods, such as static and dynamic analysis, provide solutions to some extent but show clear limitations in dealing with technological advancements and the evolution of attack strategies. These methods often rely on statistical analysis and detection of structural features, lacking depth and adaptability, making it difficult to effectively cope with complex and varied attack methods.

Machine learning technology, despite showing certain advantages in processing limited data sets and adopting statistical methods, reducing the need for manual intervention, has relatively shallow feature extraction and lacks in-depth process analysis, which to some extent limits its efficiency and accuracy in malicious file detection.

Deep learning technology, by using more training samples and neural network training methods, can further obtain document features and represent them from a higher-dimensional perspective. This method has shown significant advantages in handling complex data and identifying covert attack methods, providing a more powerful and precise framework for malicious file detection.

Reinforcement learning, despite facing challenges of limited sample numbers and the need for high efficiency, can achieve results comparable to deep learning by setting reward functions and adopting reward-punishment mechanisms. This method is particularly suited for malicious file detection in

dynamic environments, capable of continuously adapting to new attack patterns, thereby improving the flexibility and effectiveness of defense strategies.

Although artificial intelligence technologies such as machine learning, deep learning, and reinforcement learning offer unprecedented possibilities for malicious file detection, in the struggle against malicious attackers, there is no absolute security. The tech community must remain vigilant, continuously learn, research, and innovate to ensure our digital ecosystem can withstand increasingly complex attack methods, maintaining security and health.

REFERENCES

[1] Schultz, M.G., Eskin, E., Zadok, F., et al. (2001). Data Mining Methods for Detection of New Malicious Executables. In: 2001 IEEE Symposium on Security and Privacy. pp. 38-49.

[2] Li, W.J., Stolfo, S., Stavrou, A., et al. (2007). A Study of Malcode-Bearing Documents. In: Detection of Intrusions and Malware, and Vulnerability Assessment. Springer, Berlin, Heidelberg. pp. 231-250.

[3] Gao, Y.X., Qi, D.Y. (2011). Analyze and Detect Malicious Code for Compound Document Binary Storage Format. In: 2011 International Conference on Machine Learning and Cybernetics. pp. 593-596.

[4] Smutz, C., Stavrou, A. (2012). Malicious PDF Detection Using Metadata and Structural Features. In: The 28th Annual Computer Security Applications Conference on - ACSAC '12. pp. 239-248.

[5] Srndic, N., Laskov, P. (2013). Detection of Malicious PDF Files Based on Hierarchical Document Structure. In: The 20th Annual Network & Distributed System Security Symposium. pp. 1-16.

[6] Maiorca, D., Ariu, D., Corona, I., et al. (2015). A Structural and Content-Based Approach for a Precise and Robust Detection of Malicious PDF Files. In: 2015 International Conference on Information Systems Security and Privacy. pp. 27-36.

[7] Nissim, N., Cohen, A., Elovici, Y. (2017). ALDOCX: Detection of Unknown Malicious Microsoft Office Documents Using Designated Active Learning Methods Based on New Structural Feature Extraction Methodology. IEEE Transactions on Information Forensics and Security, 12(3), 631-646.

[8] Toth, T., Kruegel, C. (2002). Accurate Buffer Overflow Detection via Abstract Payload Execution. In: International Workshop on Recent Advances in Intrusion Detection. Springer, Berlin, Heidelberg. pp. 274-291.

[9] Li, W., Su, P.R., Shi, Y.F. (2010). A Technique for Detecting Malicious Documents Based on Calculation of Vector Spaces. Journal of the Graduate School of the Chinese Academy of Sciences, (2), 267-274.

[10] Laskov, P., Šrndić, N. (2011). Static Detection of Malicious JavaScript-Bearing PDF Documents. In: The 27th Annual Computer Security Applications Conference on - ACSAC '11. pp. 373-382.

[11] Du, X. (2019). Malicious PDF Document Detection Based on Mixed Feature. Journal on Communications, 40(2), 118-128.

[12] Anderson, H.S., et al. (2018). Learning to Evade Static PE Machine Learning Malware Models via Reinforcement Learning. Retrieved from https://arxiv.org/pdf/1801.08917.pdf.

[13] Wu, C., et al. (2018). Enhancing Machine Learning Based Malware Detection Model by Reinforcement Learning. In: ICCNS '18: Proceedings of the 8th International Conference on Communication and Network Security. ACM. pp. 74–78.

[14] Pirscoveanu, R.S., Hansen, S.S., Larsen, T.M.T., Stevanovic, M., Pedersen, J.M., Czech, A. (2015). Analysis of Malware Behavior: Type Classification Using Machine Learning. In: 2015 International Conference on Cyber Situational Awareness, Data Analytics and Assessment (CyberSA). London, UK. pp. 1-7.

[15] Firdausi, I., et al. (2010). Analysis of Machine Learning Techniques Used in Behavior-Based Malware Detection. In: Proceedings of the 2010 International Conference on Advances in Computing, Control, and Telecommunication Technologies.

[16] Zhang, Z., Qi, P., Wang, W. (2020). Dynamic Malware Analysis with Feature Engineering and Feature Learning. arXiv, version 5, arXiv:1907.07352v5.