Scholar Publishing Group
International Journal of Neural Network
https://doi.org/10.38007/NN.2025.040108
ISSN 2790-2692 Vol. 4, Issue 1: 65-72

Exploration of the Application of Malicious Code Detection in Network Security

Yihong Zou

Amazon Data Services, Inc, Intent Driven Network, Cupertino, California, 95014, United States

Keywords: Malicious code detection; Network security; Artificial intelligence; Threat intelligence; Comprehensive detection framework

Abstract: With the continuous advancement of technology, malicious code has become a major challenge in the field of network security, seriously threatening data confidentiality, normal system operation, and personal privacy protection. Existing detection technologies face problems such as insufficient identification of obfuscated code, adversarial sample interference, and excessive reliance on feature libraries leading to false negatives when dealing with complex and diverse malicious code, making it difficult to effectively prevent increasingly complex network threats. This article starts with the harm of malicious code and elaborates on its negative effects in areas such as information leakage and property damage. It also compares the advantages and disadvantages of static and dynamic detection techniques and explores their limitations in application. On this basis, an optimization path is proposed, including building a framework that combines static and dynamic analysis, using artificial intelligence to achieve intelligent detection, strengthening threat intelligence sharing and collaboration, and enhancing behavior analysis and real-time monitoring of unknown malicious code, opening up new directions for enhancing the detection efficiency of network security.

1. Introduction

In network attacks, malicious code is a critical tool that seriously violates data confidentiality, system reliability, and the economic benefits of enterprises. With the rapid progress of Internet technology, the types of malicious code and attack methods have become more diversified and difficult to detect. However, traditional detection methods have shown their limitations in countering unknown attacks and samples. Therefore, improving the accuracy and efficiency of malicious code detection technology is the core of improving the network security protection system. Exploring cutting-edge detection strategies and optimizing usage methods is crucial for effectively responding to malicious code attacks and ensuring the security of cyberspace.

2. The harm of malicious code

2.1 Data Leakage

Data security is an important issue in the network environment, but under complex attack methods, the leakage of sensitive user information has become an undeniable problem. Attackers use various technical methods, such as phishing, malicious code injection, and illegal exploitation of system defects, to steal users' private information, involving sensitive data such as login credentials, bank account details, and health records. Once these data are illegally stolen, it is highly likely to cause identity theft, fraudulent transactions, or the spread of harmful information, directly infringing on the legitimate rights of users. In the business field, if customer information, trade secrets, and financial data are leaked, it will cause serious damage to the company's image and may lead to chain reactions such as reduced customer numbers and weakened market competitiveness. More seriously, when government agencies encounter data breaches, the disclosure of sensitive information may pose unpredictable threats to national security, social stability, and economic stability.

2.2 Economic losses

The economic losses caused by malicious programs exhibit diverse characteristics and have serious consequences. For ordinary users, once they are invaded by malicious programs, they may directly face problems such as bank account fund theft, credit card information leakage, and fraudulent transactions, resulting in property losses. The theft of personal identity information may also lead to long-term economic and legal disputes. For enterprises, the economic losses caused by malicious code attacks are even more severe, such as the need to pay huge ransom after being locked by ransomware, the decrease in revenue caused by business suspension, and the costs required for system recovery and data rescue. Enterprises also need to face legal lawsuits and fines caused by data breaches, as well as customer trust crises caused by brand image damage, which are potential losses that cannot be simply measured.

3. The current application status of malicious code detection in network security

3.1 Insufficient recognition of obfuscation and encrypted code in static analysis

Static analysis is a method of detecting potential malicious behavior by directly parsing the structure and features of code, which has the characteristics of high efficiency and wide coverage. But when faced with obfuscated or encrypted code, the effectiveness of this method will be limited. Confusion techniques can increase the difficulty of analyzing malicious code by transforming code organization, embedding invalid code fragments, and simplifying control processes, thereby making static analysis tools encounter obstacles in capturing key information. The encrypted code encrypts the data or the entire file by applying encryption algorithms, obscuring the actual content and preventing static analysis tools from directly interpreting it, thereby losing detection efficiency. Static analysis often relies on feature signature databases to identify known threats, and it is often powerless to detect new types of malicious code that have not been entered into the database, which can easily lead to missed detections. The detection process for feature matching can be represented by the following formula:

$$D = \sum_{i=1}^{n} 1(f(C_i, C_j)) \tag{1}$$

Among them, D is the detection result, Ci represents the features of the code sample, Cj represents the feature signature library, and 1 is the matching indicator function. When features cannot be matched due to confusion or encrypted code, the detection accuracy will significantly

decrease. This indicates that static analysis needs to be combined with other technologies to achieve more comprehensive protection when dealing with obfuscation and encrypted code.

3.2 Resource consumption and detection efficiency limitations of dynamic analysis

By observing the execution behavior of code in an independent space, dynamic analysis can effectively reveal potential threats, especially for dealing with malicious code that is obfuscated and encrypted. However, its high resource consumption and efficiency limitations clearly constrain its practical application effectiveness. Dynamic analysis relies on virtual machines or sandbox operating environments, which require a significant amount of computing resources and storage space, especially when dealing with large-scale samples or high complexity behaviors, resulting in significantly extended analysis time. Dynamic analysis relies on the active activation of malicious code, and some malicious code conceals its malicious nature by designing delayed execution, multi-path selection, or conditional branching, allowing certain behavioral patterns to escape detection. The difficulty of dynamic analysis lies in extracting behavioral features from the execution process, which requires efficient algorithms to handle the massive amount of data generated during runtime, thereby exacerbating the computational load. The dynamic behavior extraction process can be represented by the following formula:

$$B = \int_{t_o}^{t_n} g(S_t)dt \tag{2}$$

Among them, B represents the behavior feature result, St is the system state feature at time t, and g is the behavior extraction function. The high complexity of behavior and resource consumption issues limit the efficiency and application scope of dynamic analysis in high-frequency and large-scale detection scenarios.

3.3 Feature library dependency leads to unknown threat omission

In traditional malicious code detection methods, feature libraries play a core role, relying on the comparison of specific signatures of malicious code to discover security vulnerabilities. However, this highly dependent feature library strategy exposes obvious shortcomings, especially when facing unknown threats, it is prone to false negatives. The establishment of a feature library must be based on known malicious samples, which limits its detection capability to only the recorded feature set. It is often powerless against new or modified malicious code. Some malicious code bypasses existing feature library detection by using code mutation techniques, such as generating random variants, or using obfuscation and encryption methods to mask its own features. The update pace of feature libraries often cannot keep up with the rapid mutation of malicious code, and the analysis and input of new sample features require a lot of time. During this time gap, unknown threats may have already posed a threat to system security.

3.4 Decreased detection accuracy caused by adversarial samples

Adversarial samples are an attack method that misleads detection algorithms by adding subtle but malicious interference information, allowing malicious code to maintain its functionality. This type of sample seriously weakens the accuracy of the detection system, especially in the field of malware identification that relies on machine intelligence. Adversarial samples cleverly exploit the weaknesses of algorithms by creating seemingly normal code variants to evade monitoring by detection systems. Hackers are able to use these samples to evade the feature analysis step, thereby undermining the discriminative power of the detection engine. The production of adversarial

samples often comes with uncertainty and variability, which makes it difficult for defense systems to form stable protection strategies. The current methods used to identify malicious code are likely to result in false positives or false negatives when encountering adversarial samples, thereby affecting the credibility of the system. Adversarial samples may also interfere with the training process of the detection model, contaminate the training dataset, and further affect the detection performance.

4. The application path of malicious code detection in network security

4.1 Comprehensive detection framework combining static and dynamic analysis

Building a comprehensive detection framework that combines static and dynamic analysis is an effective approach to address the limitations of static and dynamic analysis. Static analysis can quickly comb through file architecture and extract key attributes, while dynamic analysis can accurately capture malicious actions during obfuscation of code and program execution, such as file tampering, abnormal network connections, or diffusion behavior. For example, during a security inspection of a company's intranet, static testing revealed that a commonly used third-party compression tool contains numerous abnormal code modules, especially frequent calls to network communication APIs. When conducting in-depth dynamic detection, it was observed that the software intentionally sent specific files in the user directory to an unknown foreign server after the decompression action was completed. Through the combination of these two detection methods, the security team confirmed that the software contained malicious code, quickly cut off its network channel, and warned employees not to continue using it. This detection framework can also combine two detection results based on registry changes and network activity characteristics to make judgments, and use feature comparison and behavior tracking to comprehensively evaluate the threat of samples. The detection process can be summarized as follows:

$$R = \alpha \cdot \sum_{i=1}^{n} M(F_s(C_i), T_j) + \beta \cdot \int_{t_o}^{t_n} g(S_t)\ dt \tag{3}$$

Among them, R is the comprehensive detection result, Fs(Ci) represents the static feature extraction function, Tj is the feature library, g (St) represents the dynamic behavior extraction function, and α and β are the weight parameters for static and dynamic analysis. Through the complementarity of static and dynamic analysis, this framework significantly reduces the risk of false positives and false negatives while improving detection efficiency.

4.2 Intelligent Detection Technology for Malicious Code Based on Artificial Intelligence

The application of artificial intelligence effectively supplements the shortcomings of traditional detection methods, greatly enhancing the intelligence and accuracy of detecting malicious code through advanced technologies such as deep learning and feature extraction. In the face of the limitations of confusion and encrypted code in static analysis, intelligent detection technology can use deep learning algorithms to automatically extract features from code samples. Through the learning of feature vectors, unique attributes of complex code can be effectively extracted without interference from confusion or encryption methods. The introduction of intelligent optimization strategies can significantly improve the efficiency of capturing dynamic behavior, while ensuring detection quality and optimizing resource allocation, in response to the challenges of resource consumption in dynamic analysis. For example, when analyzing a specific Trojan program, deep learning algorithms can automatically mine the core characteristics of network behavior and

identify its hidden remote control components. In practical application scenarios, it also involves using adversarial samples to reinforce the training of the model, generating adversarial samples through Generative Adversarial Networks (GANs) to enhance the robustness of the detection model. Combining a large number of malicious code samples and normal samples during model training helps balance the performance of the model in classification. The specific implementation of intelligent detection can be described by the following formula:

$$p = \mathrm{f}(W \cdot X + \mathrm{b}) + \gamma \cdot \int_{t_o}^{t_n} \mathrm{h}(S_t)\mathrm{d}t \tag{4}$$

Among them, P represents the final detection result, W and b are the feature weight matrix and bias, X is the input sample feature vector, h (St) is the dynamic behavior extraction function, and γ is the adjustment parameter for dynamic analysis. Through artificial intelligence technology, detection systems can achieve comprehensive analysis and adaptive adjustment of static and dynamic features to efficiently respond to the variability and complexity of malicious programs.

4.3 Collaboration of Threat Intelligence Sharing and Detection Systems

Threat intelligence sharing provides critical support for malicious code detection by gathering threat information from multiple sources. Its core lies in the organic combination of information collection, data sharing, and system collaboration. We need to establish an efficient intelligence gathering system that extracts potential threat information from network data streams, system logs, and security incident reports, and then classifies, associates, and filters this information to generate valuable intelligence data. These intelligence materials are then uploaded to a shared platform and stored in a central database to facilitate information exchange between different organizations, while also balancing data confidentiality and access control management. The detection system needs to integrate these intelligence data, combined with local databases and behavioral analysis tools, to make accurate judgments on unknown samples. Collaboration between systems also includes building an instant response mechanism, which can quickly notify other members and update protective measures once a new threat is detected, thus creating a comprehensive defense system. For example, in an attack incident targeting an e-commerce platform, the threat intelligence system identified malicious traffic appearing on multiple user devices with the same source and associated with malicious files disguised as normal advertisement push. Through the exchange and sharing of intelligence, it was confirmed that the file contained malicious code that stole payment information, and detailed information on the source and transmission path of the file was provided, which helped the platform quickly identify the problem and cut off the malicious transmission chain. Figure 1 shows the complete process of threat intelligence collection, sharing, and collaboration with detection systems, providing an efficient collaborative mechanism for malicious code detection.

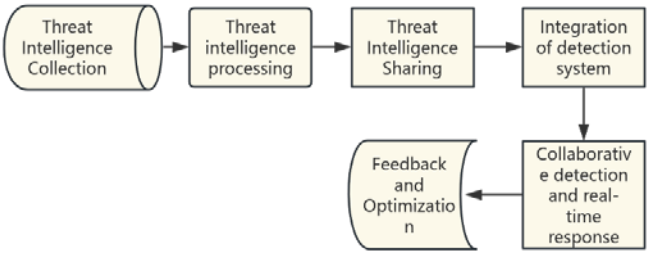

Figure 1. Collaborative process diagram of threat intelligence sharing and detection system

4.4 Behavior analysis and real-time monitoring of unknown malicious code

Behavior analysis and real-time monitoring of unknown malicious code can be achieved by establishing standard behavior templates, analyzing dynamic activity characteristics, and improving detection strategies. It is necessary to collect various data of the system under normal conditions, such as processor utilization, network data transmission mode, and file operation frequency, in order to create a standard behavior template for identifying abnormal behavior. Install real-time monitoring devices at critical locations to detect abnormal behavior, such as illegal file operations, abnormal increases in network traffic, or illegal requests for sensitive permissions. The dynamic rule making system will develop detection standards in real-time based on abnormal data. Once behavior exceeds the template range, an alert will be triggered and relevant information will be recorded. At the same time, machine learning techniques are used to classify and extract attributes from the collected data, iteratively update the threat model, and enhance the recognition effect of unknown threats. For example, during the operation of a financial institution's server, the monitoring system detects abnormal database query behavior that occurs outside of working hours and observes data being transmitted to external IP addresses. The system detected that this behavior does not match the standard template and immediately triggered an alert and formulated corresponding rules. After analysis, it has been confirmed that this behavior is caused by a new type of malicious code resulting in privacy data leakage. Financial institutions have optimized their models using the collected data, successfully preventing further spread of risks. Table 1 provides a detailed list of the key steps for behavior analysis and real-time monitoring of unknown malicious code, providing a clear path reference for system design and optimization.

Table 1. Analysis of Unknown Malicious Code Behavior and Real time Monitoring Path

step	Specific operation	target	Key points
Construction of behavioral baseline	Collect normal operational data, including network communication, file access, etc	Establish baseline behavioral characteristics for the system	Accurately define the normal operating range
Abnormal behavior monitoring	Capture dynamic operational data and identify abnormal behavior	Identify potential abnormal operations	Deviation point of positioning behavior
Real time monitoring deployment	Deploy monitoring tools at critical nodes to obtain real-time data	Dynamically capture operational status	Covering critical areas of the system
Application of dynamic rule engine	Analyze behavior deviation and generate detection rules	Implement dynamic anomaly detection	Continuously optimizing the rule library
Alarm triggering and log generation	Abnormal exceeding threshold triggers alarm and logs	Provide follow-up analysis basis	Ensure record integrity
Threat Mode Analysis	Classify abnormal behavior and extract threat features	Formulate targeted analysis results	Associate historical data
Threat Model Update	Integrating new features into model training data	Improve detection capability	Updating models in a timely and effective manner

5. Conclusion

With the continuous evolution of malicious code, network security is facing unprecedented challenges. Its advanced disguise techniques and complex mutation mechanisms make conventional defense methods inadequate. By building a comprehensive detection framework that combines

artificial intelligence technology and threat intelligence sharing mechanisms to address the limitations of both static and dynamic analysis, detection efficiency and accuracy can be effectively improved. The introduction of real-time monitoring and behavior analysis provides technical support for the timely detection of unknown threats, while the application of dynamic rules significantly enhances the flexibility of the detection system. In the future, malicious code detection technology needs to be deeply optimized in model training and real-time response to reduce resource consumption and latency issues. It is crucial to promote the exchange of threat intelligence between different industries and build a multi-party cooperative network security defense line. Continuous innovation and technological advancements will make the network security environment more robust, providing a solid foundation for digital development in various fields.

Reference:

[1] *Geetha C, Ramakrishnan M. Replica Node Attack Detection Approaches in Static WSNs. Përparimet në shkencat kompjuterike dhe teknologjinë, 2023, 16(2):141-150.*

[2] *Marc Chalé, Cox B, Weir J, et al. Optimizim i kufizuar bazuar në gjenerimin e shembullit kundërshtar për sulmet e transferimit në sistemet e zbulimit të intrusioneve në rrjet. Optimization Letters, 2024, 18(9):2169-2188.*

[3] *Gjetjet nga Universiteti i Teksasit San Antonio Update Knowledge of Information Security (Image-based Malware Representation Approach with Efficientnet Convolutional Neural Networks for Effective Malware Classification). Network Daily News, 2022(26):31-32.*

[4] *Hulumtuesit e Universitetit Nagoya përshkruajnë kërkimet në teknologjinë e rrjetit (Malware Detection by Control-Flow Graph Level Representation Learning with Graph Isomorphism Network). Network Daily News, 2022(14):34-34.*

[5] *Landauer M, Skopik F, Frank M, et al. Dataset loge të mirëmbajtura për vlerësimin e sistemeve të zbulimit të intrusioneve. IEEE transactions on dependable and secure computing, 2023(4):20.*

[6] *Zhu, P. (2025). The Role and Mechanism of Deep Statistical Machine Learning In Biological Target Screening and Immune Microenvironment Regulation of Asthma. arXiv preprint arXiv:2511. 05904.*

[7] *Liu, B. (2025). Design and Implementation of Data Acquisition and Analysis System for Programming Debugging Process Based On VS Code Plug-In. arXiv preprint arXiv:2511. 05825.*

[8] *Chang, Chen-Wei. "Compiling Declarative Privacy Policies into Runtime Enforcement for Cloud and Web Infrastructure." (2025).*

[9] *Sun, Jiahe. "Research on Sentiment Analysis Based on Multi-source Data Fusion and Pre-trained Model Optimization in Quantitative Finance." (2025).*

[10] *F. Liu, "Architecture and Algorithm Optimization of Realtime User Behavior Analysis System for Ecommerce Based on Distributed Stream Computing," 2025 International Conference on Intelligent Communication Networks and Computational Techniques (ICICNCT), Bidar, India, 2025, pp. 1-8.*

[11] *F. Liu, "Transformer XL Long Range Dependency Modeling and Dynamic Growth Prediction Algorithm for E-Commerce User Behavior Sequence," 2025 2nd International Conference on Intelligent Algorithms for Computational Intelligence Systems (IACIS), Hassan, India, 2025, pp. 1-6.*

[12] *K. Zhang, "Optimization and Performance Analysis of Personalized Sequence Recommendation Algorithm Based on Knowledge Graph and Long Short Term Memory*

Network," 2025 2nd International Conference on Intelligent Algorithms for Computational Intelligence Systems (IACIS), Hassan, India, 2025, pp. 1-6.

[13] Su H, Luo W, Mehdad Y, et al. Llm-friendly knowledge representation for customer support[C]//Proceedings of the 31st International Conference on Computational Linguistics: Industry Track. 2025: 496-504.

[14] Q. Hu, "Research on Dynamic Identification and Prediction Model of Tax Fraud Based on Deep Learning," 2025 2nd International Conference on Intelligent Algorithms for Computational Intelligence Systems (IACIS), Hassan, India, 2025, pp. 1-6.

[15] D. Shen, "Complex Pattern Recognition and Clinical Application of Artificial Intelligence in Medical Imaging Diagnosis, " 2025 International Conference on Intelligent Communication Networks and Computational Techniques (ICICNCT), Bidar, India, 2025, pp. 1-8.

[16] Chen M. Research on Automated Risk Detection Methods in Machine Learning Integrating Privacy Computing. 2025.

[17] Ding, J. (2025). Intelligent Sensor and System Integration Optimization of Auto Drive System. International Journal of Engineering Advances, 2(3), 124-130.

[18] Mingjie Chen. (2025). Exploration of the Application of the LINDDUN Model in Privacy Protection for Electric Vehicle Users. Engineering Advances, 5(4), 160-165.

[19] Liu, X. (2025). Research on Real-Time User Feedback Acceleration Mechanism Based on Genai Chatbot. International Journal of Engineering Advances, 2(3), 109-116.

[20] Zhang, M. (2025). Research on Collaborative Development Mode of C# and Python in Medical Device Software Development. Journal of Computer, Signal, and System Research, 2(7), 25-32.