Sensitive Information Detection and Governance in Binary Data Processing

Jiahong Wang, Yibo Chang*
College of Electronic Engineering, National University of Defense Technology, Hefei, Anhui, China
**Corresponding Author*

Abstract: In the current era of rapid digital development, binary data, as the fundamental storage and transmission form in computer systems, faces increasingly prominent security issues. Binary data exists in various non-text forms like computer programs and encrypted data. Its unstructured nature and poor readability make it difficult to detect sensitive information using traditional text-based methods. This paper starts with the characteristics of binary data and deeply explores the definition, classification of information sensitivity, and its significant importance in binary data processing. Information sensitivity is crucial for measuring the potential harm of data leakage or abuse. It includes personal privacy data, business-critical assets, and national-security-related information. It studies the identification technologies based on data feature analysis and machine learning. In the information age, binary data serves as the core carrier of digital systems, playing a crucial role. It is widely applied in various fields such as file storage, network communication, and program execution. Binary data may contain a large amount of sensitive information. If it is leaked or misused, it is highly likely to lead to serious consequences. Given that binary data has the characteristics of being unstructured and having poor readability, traditional text-based sensitive information detection strategies cannot be directly implemented. How to efficiently identify the sensitive content within binary data and formulate scientific management plans has become a major issue in the field of network security. In this paper, in combination with computer technology, a systematic exploration is carried out on the methods for identifying and managing the information sensitivity within binary data.

Keywords: Binary Data; Information Sensitivity; Privacy Protection; Data Encryption; Access Control

1. Introduction

1.1 The Definition and Characteristics of Binary Data

Binary data is a form of digital information composed of 0s and 1s as basic units. It is commonly found in non-textual carriers such as computer programs, image files, and encrypted data. This type of data does not possess the structural characteristics of human language. Its content has no fixed pattern, and there are no obvious semantic breaks. It exhibits the characteristics of a bit sequence with both discrete and continuous arrangements. In terms of storage efficiency, performing efficient compression processing can significantly reduce its volume. Compared with text formats, it can save more storage space. Its content expression mode is different from that of natural language and cannot be directly interpreted by humans or general software. It is necessary to use a dedicated decoder or analysis program to convert it into understandable information. Based on these characteristics, the privacy information or confidential content that may be contained in binary data is often more concealed. When conventional detection tools do not have the corresponding analysis capabilities, it is not easy to directly identify the potential sensitive content within it.

1.2 The Meaning and Classification of Information Sensitivity

Information sensitivity measures the potential harm level of data in the face of leakage or abuse in unauthorized scenarios [1]. According to the nature of the data and the scope of its influence, the core types include three dimensions: privacy data related to personal

identification or physiological characteristics, such as ID card numbers and fingerprint information; commercial assets that play a role in an enterprise's core competitiveness, such as documents related to core technologies or the database corresponding to an enterprise's customers; and confidential intelligence involving national strategic security, such as strategic deployment plans or the security keys of key facilities [2]. When such information exists in binary format, it may be hidden in the metadata area of a file, in encrypted data fragments, or in the execution process of software. It is necessary to use a comprehensive analysis of structural parsing and semantic association to accurately locate it [3].

1.3 The Importance of Sensitive Data in Binary Processing

From the perspective of the operation rules of computer systems, the high-efficiency characteristics of binary data processing support the operational efficiency of the digital world. And the security control measures implemented for sensitive information within it determine the defensive capability of the system. Malicious programs may take advantage of vulnerabilities in the binary format to implant harmful instructions. For example, they may disguise themselves as the structure of normal files and embed destructive code into them. Attackers frequently target binary files such as database backups, attempting to use reverse engineering to dig out hidden confidential content. The inherent complexity of this data form makes the identification of sensitive information, permission control, and flow tracking of binary data the core pillars of the network security protection system. Improving the accuracy of identifying the sensitivity of binary data and establishing a comprehensive management system can not only effectively block network attack methods but also serve as the core key to meeting data security regulations and industry compliance requirements.

2. Information Sensitivity Identification Technology

2.1 Data Feature Analysis and Sensitivity Identification

The core of detecting the sensitivity of binary data focuses on feature mining and marking strategies. The identification operation generally starts from the file structure, such as analyzing the information in the beginning part of an executable file or the fixed items in the program loading format [4]. At the same time, a quantitative evaluation of data randomness is carried out in combination. For example, entropy calculation is used to determine whether a data segment has been obfuscated or encrypted [5]. Detecting the regularity of specific byte sequences can identify the typical features of known encryption algorithms or malicious codes. When the data shows highly randomized characteristics, it generally indicates that there may be encrypted data here [6]. At this moment, decryption methods should be used for subsequent verification. The supplementary information of the file also has considerable reference value. For instance, the timestamp of file generation or the signature content of the developer. These metadata can add an additional verification dimension to the classification of sensitive data, thereby improving the reliability level of the detection results.

2.2 Application of Machine Learning in Sensitive Information Identification

In the process of intelligent analysis of binary data, machine learning builds a sensitive information identification system by deeply mining the potential laws of byte sequences. According to the methods of natural language processing, the binary stream is converted into a symbol sequence with contextual relationships [7]. Relying on the time-series model, it captures the special patterns caused by instruction jumps or data encryption. For example, the recurrent neural network can identify the abnormal performance of the function call chain across byte segments. Technologies for spatial feature extraction, by reconstructing the byte arrangement structure, convert binary files into analyzable two-dimensional images, enabling the convolutional neural network to locate sensitive segments such as the key storage area and the malicious code implantation area from the pixel matrix. In the actual engineering application process, it is often necessary to combine preprocessing methods such as byte frequency statistics and offset alignment to optimize the data representation. At the same

time, generative adversarial networks are used to expand rare samples, thereby alleviating the situation of data imbalance. Although these methods break through the inherent limitations of the traditional rule base, their effects are limited by the quality and scale of the labeled samples. Moreover, the internal decision-making logic of the model cannot be explained, which may lead to misjudgments in key scenarios. Most of the time, it is necessary to integrate static feature analysis and dynamic behavior verification to establish a dual verification mechanism.

2.3 The Advantages and Limitations of Existing Identification Technologies are Discussed

In the evolution process of binary data identification technology, the existing methods can already use a static rule base to quickly match sensitive features. For example, predefined byte signatures or structural templates are used to quickly locate known malicious code segments. For binary files with a fixed format, traditional technologies can accurately identify encrypted sections or resource hiding behaviors by means of file header features, entropy value analysis, etc. However, when dealing with network protocol payloads generated in a dynamic form, due to the complexity of data fragmentation transmission and real-time parsing, it is often impossible to completely restore the context-related features [8], resulting in a significant increase in the missed detection rate of zero-day attack payloads or custom encrypted streams. Attackers use binary obfuscation techniques to perform operations such as instruction equivalent replacement of code segments and filling of invalid bytes, which can easily disrupt the detection logic based on fixed offsets or byte frequencies. Even if the mechanisms of streaming feature extraction and sliding windows are introduced, due to the limitations of computing resources, it may still be impossible to cover long-distance dependencies. At present, some solutions attempt to combine dynamic taint tracking and heterogeneous model integration to mitigate adversarial sample attacks, but such methods often bring additional performance losses and face implementation challenges in embedded devices or high-concurrency scenarios, which reflects that the binary sensitive information identification technology still needs to find a better balance among real-time performance, generalization ability, and resource efficiency.

3. Information Sensitivity Management Strategy

3.1 Storage and Access Control of Sensitive Data

Within the scope of the security governance system for binary sensitive information, the storage stage should implement multi-level protection according to the characteristics of the data form. Physical isolation and dynamic encryption technologies should be applied to the binary keys resident in memory to ensure that even in the event of a memory dump attack, it is difficult to restore the original data. The access control mechanism should conform to the underlying interaction characteristics of binary files. It should rely on kernel-level drivers to intercept API call requests from untrusted processes. At the same time, hardware security modules should be used to solidify the core verification logic to prevent malicious exploitation of privilege escalation vulnerabilities [9]. In the context of binary auditing, an instruction-level behavior analysis system can be deployed to capture irregular operations on sensitive registers or interrupt vectors during process runtime. Such fine-grained monitoring is more effective in identifying covert injection attacks than traditional file logs. During actual deployment, attention should be paid to balancing security and system performance. For example, when conducting packet-by-packet verification of encrypted binary frames in real-time audio and video streams, streaming verification algorithms can be developed to reduce computational latency and prevent interference with the normal business interaction process.

3.2 Data Encryption and Data Masking Technology

In the technical scope of binary data protection, the choice of encryption algorithm should be closely aligned with the data flow scenarios. Take structured binary files such as firmware upgrade packages as an example. A symmetric encryption scheme based on hardware acceleration can be adopted. A dedicated instruction set can be used to improve the

operational efficiency of the AES algorithm in processing multi-threaded byte blocks. At the same time, combined with memory encryption technology, it can prevent the key from being maliciously intercepted during operation. For binary instruction streams that are frequently replaced in dynamic interaction scenarios, a lightweight streaming encryption algorithm can be employed. While maintaining real-time performance, perform byte-by-byte obfuscation of register operation codes. When applying the data masking technology, the integrity constraints of the binary format need to be taken into account. For example, for the memory address information in the debugging log, a bit mask can be used to retain the characteristics of the address range while blurring the specific offset. In the test environment, the protocol message payload can be selectively replaced at the byte granularity, which not only hides the actual parameters but also does not affect the parsing process of the protocol stack. It should be noted that due to the strong format correlation of binary data, traditional masking technology may damage the checksum or hash value. Therefore, generally, it is necessary to cooperate with a dynamic recalculation module to update the relevant fields synchronously. For adversarial attacks such as side-channel analysis, it is necessary to introduce random noise perturbations during the encryption masking process. By blurring the characteristics of the power supply ripple or the timing differences in instruction execution, the difficulty for attackers to reverse and restore the data is increased [10].

3.3 Compliance Management and Legal Requirements

Within the framework of global data governance, the compliant management of binary sensitive information needs to break through the traditional regulatory paradigm for textual data. For example, for binary logs generated by embedded devices, a feature obfuscation engine should be incorporated at the data collection stage, and entropy perturbation should be performed on device unique identifiers such as hardware fingerprints. This not only meets the data minimization requirements stipulated by the GDPR but also avoids the difficulties in fault diagnosis caused by excessive information desensitization. In scenarios related to cross-border transmission, the encryption strength of binary firmware should dynamically conform to the legal thresholds of the target regions. When it comes to export control issues regarding cryptographic algorithms, a hybrid solution combining sharded encryption with local key escrow can be employed to ensure that the core logic code segments meet the regulatory requirements of the importing countries. When implementing the privacy by design principle during the development stage, attention should be paid to the risks of covert channels in binary instruction sets. A toolchain for code obfuscation and metadata stripping should be integrated at the compiler level to eliminate potential information leakage sources such as debugging symbol tables and stack pointer offsets from the machine code generation stage. In addition to verifying the integrity of encryption protocols during regular compliance audits, binary difference analysis techniques should also be used to compare the hash fingerprints of execution files in the production environment with those of the compliant baseline version, enabling rapid identification of policy drift issues caused by hot update operations. Especially during the firmware iteration process of IoT devices, this byte-level verification mechanism can effectively prevent the implantation of unauthorized sensitive function calls.

4. Information Leakage Risk in Binary Data Processing

Throughout all stages of the entire life cycle of binary data, the risk of information leakage permeates the storage, transmission, and processing links. The risks of static data are manifested as the theft of unencrypted files or the recovery of residual data from disks. Dynamic risks are caused by the hijacking of privileges resulting from the leakage of sensitive information in memory or buffer overflow vulnerabilities. Logical vulnerabilities often occur because debugging symbols are not stripped, allowing reverse engineering to restore the business logic. In the situation where internal and external threats coexist, situations such as developers mistakenly sending test packages containing sensitive information and malicious code being implanted in third-party libraries keep

emerging. It is necessary to use data flow diagrams to track the flow paths of binary data, identify key nodes, and use automated scanning and taint tracking technologies to monitor abnormal memory accesses. The risk management and control system should integrate static analysis to identify hard-coded keys, use a dynamic overwriting mechanism to eliminate memory residues, and incorporate disassembly comparison during the firmware signature verification stage to block supply chain attacks. Ultimately, a closed-loop defense structure covering threat modeling, priority evaluation, and real-time protection is formed [11].

5. Future Development Trends and Challenges

When facing the challenges in the security of binary data in the future, the combination of technological innovation and regulatory evolution will establish a new protection paradigm. Due to the threat of quantum computing, lattice-based resistant encryption algorithms have been developed. With the help of homomorphic encryption technology, it has become feasible to process ciphertext in the cloud. Edge computing enables lightweight detection models to conduct real-time sensitive information screening on terminal devices. In the face of the wave of global data sovereignty legislation, binary files need to embed machine-readable metadata tags to support automated compliance review work. The in-depth implementation of technologies such as differential privacy and federated learning will unleash the value of data while ensuring individual information. In the face of increasingly complex attack methods, the adaptive security architecture uses AI to dynamically optimize encryption strategies, and the threat intelligence sharing mechanism promotes the industry-wide synchronization of the attack feature library. Hardware-level protection builds a physically isolated secure enclave relying on a trusted execution environment. Cultivating cross-disciplinary talents has become the core link. Talents are required not only to have a good command of the frontier breakthroughs in quantum cryptography but also to master the engineering implementation of privacy enhancement technologies, so as to resolve the disconnection dilemma between security theories and actual deployments and endow the risk management of the entire life cycle of binary data with the ability to continuously progress and evolve.

6. Conclusion

Binary data serves as the core carrier of the digital world. Managing its information sensitivity not only poses technical challenges but also represents a governance proposition. This paper has constructed a systematic full-cycle management framework ranging from identification to protection. From a technical perspective, by means of parsing data features and utilizing machine learning models, the barriers of binary data's unstructured nature are broken down, enabling precise location and search for sensitive information. From the strategic dimension, by integrating dynamic encryption, access control, and compliance auditing, a deep defense system is formed, effectively addressing potential leakage risks caused by static storage, dynamic memory issues, and logical vulnerabilities. Facing the dual trends of the rise of quantum computing and the continuous deepening of data sovereignty, the future protection system needs to make continuous breakthroughs in areas such as quantum-resistant encryption, edge intelligent detection, and adaptive security architectures. At the same time, it is necessary to enhance the dynamic adaptation ability of privacy enhancement technologies to international regulations. The research reveals the important contradiction in binary data security—the competition between the demand for efficient processing and risk prevention and control. Only through the coordinated progress of technological innovation and institutional design, and by building an interpretable, verifiable, and sustainable governance framework, can we achieve a symbiotic balance between data value mining and security protection. This requires the academic and industrial circles to form a collaborative force in algorithm optimization, hardware reinforcement, and cross-domain talent cultivation, endowing the security of binary data in the digital age with continuous vitality.

References

[1] Simon D Duque Anton, Hans Dieter Schotten. Intrusion Detection in Binary Process Data: Introducing the

Hamming-distance to Matrix Profiles. 2020,

[2] Hioki Hirohisa. Data Hiding for Text and Binary Files. Computational Linguistics, 2024.

[3] Langner, Paula R. CAa, Juarez-Colunga, et al. Efficiency loss with binary pre-processing of continuous monitoring data. Statistics in Biosciences, 2025.

[4] Or Ordentlich, Yury Polyanskiy. Strong Data Processing Constant Is Achieved by Binary Inputs. IEEE Transactions on Information Theory, 2022, 68(3): 1480-1481.

[5] Manh Khoi Duong, Stefan Conrad. Towards Fairness and Privacy: A Novel Data Pre-processing Optimization Framework for Non-binary Protected Attributes. 2024,

[6] Koeck, Philip J.B. philip.Koeck@csb.ki.se. Missing data in image and signal processing: The case of binary objects. Optik-International Journal for Light and Electron Optics, 2004, 115(10): 459-472.

[7] Mueller, bernd, gladigau, et al. Method and device for processing binary code Data: WO2017EP6310. 2017-12-28.

[8] Gomer H. Redmond, Dennis E. Mulvihill. The use of a binary computer for data processing//IRE-AIEE-ACM '60 (Eastern): Papers presented at the December 13-15, 1960, eastern joint IRE-AIEE-ACM computer conference. 1960.

[9] Wilkinson, james h. Processing binary DATA: CA374878A. 1983-08-09.

[10] Ive, John G.S., Ive John G S. Processing Binary Data: CA376263A. 1983-08-23.

[11] Jin Guo, Xuebin Wang, Yanling Zhang. System identification with binary-valued observations under both denial-of-service attacks and data tampering attacks: the optimality of attack strategy. Control Theory and Technology, 2022, 20(1): 127-138.

