Proceedings of The 6th International Conference on Signal Processing and Machine Learning
DOI: 10.54254/2755-2721/2026.31419

Research on Automatic Detection Technology for Firmware Vulnerabilities of IoT Devices

Ertong Yin

School of Cyber Security, Xi'an Polytechnic University, Xi'an, China
2051375738@qq.com

Abstract. IoT devices are used more and more widely, so firmware security (the safety of the core software in devices) becomes a key part of online safety. Firmware vulnerabilities (weak spots in the software) can easily cause big safety problems, like bad people taking control of devices or stealing data. The old way of checking vulnerabilities by hand is slow and can't cover all problems, and automatic detection technology (machines check the weak spots by themselves) is the best way to solve this problem. This paper focuses on one main goal: make the checking of IoT firmware vulnerabilities more accurate and faster. We study the structure and core rules of automatic detection technology carefully, and we also focus on how to make key steps better, such as firmware analysis (reading and understanding the device's core software), vulnerability feature extraction (finding the special signs of weak spots), automatic verification (machines test if the weak spots are real). By adjusting the technology and changing the work steps, we can quickly find and exactly locate weak spots in different kinds of IoT device firmware. The results of this research help build a strong technology base for IoT firmware safety, they also cut down the cost of checking vulnerabilities a lot, and make the whole IoT system much safer.

Keywords: IoT devices, Firmware vulnerabilities, Automatic detection, Vulnerability detection technology

1. Introduction

With the wide use of IoT technology, smart devices have been widely used in key fields such as industrial control, smart home and public security. Firmware is the core support for device operation, and its safety is directly related to the overall protection level of the IoT system. At present, security incidents such as device hijacking and data theft caused by firmware vulnerabilities often happen. The traditional manual detection method is limited by low efficiency and narrow coverage, so it is difficult to meet the detection needs of a large number of different IoT devices. Under this background, automatic detection technology for firmware vulnerabilities has become a key breakthrough to ensure IoT security. This paper focuses on the research of automatic detection technology for firmware vulnerabilities of IoT devices. It aims to break through the existing technical bottlenecks and provide an efficient and feasible technical path for the security protection of IoT devices. The following will explain the relevant technical system and research practice in detail.

DOI: 10.54254/2755-2721/2026.31419

2. Background and security situation of firmware vulnerability detection technology for IoT devices

2.1. Widespread penetration of IoT devices and highlighted core status of firmware

IoT technology has been deeply applied in multiple fields such as industrial manufacturing, smart home, intelligent transportation and medical health. The scale of various terminal devices is steadily rising, building a heterogeneous device cluster network. As the core underlying software for the operation of IoT devices, firmware is responsible for key tasks such as hardware driver adaptation, functional logic implementation and data transmission regulation. Its security is directly related to the stable and reliable operation of devices, and further affects the data security and operation foundation of the entire IoT system. With the expansion of device application scenarios to key infrastructure, the impact of firmware security has gone beyond a single device, rising to the core dimension of public security and industrial security. The demand for R&D and implementation of firmware vulnerability detection technology is becoming increasingly urgent.

2.2. Diverse manifestations of firmware vulnerabilities in IoT devices and security risk transmission

At this stage, firmware vulnerabilities of IoT devices have shown a diversified trend. The core causes include design omissions and non-standard coding in the firmware development process, as well as legacy vulnerabilities associated with the reuse of third-party open-source components. To improve R&D efficiency, some device manufacturers simplify the firmware security verification process, leading to the repeated occurrence of typical vulnerabilities such as buffer overflow, permission bypass and command injection [1]. If such vulnerabilities are exploited, they can easily trigger security risks such as illegal device control, leakage of sensitive data and device malfunction. These risks are prone to spread within the device network, interfere with the normal operation of associated devices and systems, and pose severe challenges to the stable operation of relevant fields.

2.3. Development context and practical demands of firmware vulnerability detection technology for IoT devices

In the early stage, IoT devices were single in type and simple in function, and firmware vulnerability detection mainly relied on traditional methods such as manual auditing. With the increasing variety of devices and the gradual improvement of firmware complexity, the traditional detection mode can neither match the firmware differences of heterogeneous devices nor meet the detection demands of massive devices. In addition, device manufacturers and regulatory authorities are paying increasing attention to firmware security, and relevant security standards and specifications are continuously improved, driving the transformation of vulnerability detection technology towards automation and precision. Against this background, sorting out the development context and current security situation of firmware vulnerability detection technology is the key foundation for the subsequent promotion of technology optimization and scheme design.

DOI: 10.54254/2755-2721/2026.31419

3. Analysis of existing core problems in automatic detection of IoT device firmware vulnerabilities

3.1. Insufficient adaptation to heterogeneous device firmware leading to limited detection coverage

IoT devices involve a variety of hardware architectures and operating system kernels, and firmware formats show a fragmented distribution. There are significant differences in firmware packaging standards and file structures among different manufacturers. Most existing automatic detection tools are built for specific architectures or formats, lacking a universal firmware parsing and adaptation mechanism, and thus cannot achieve full coverage of heterogeneous device firmware. For some devices with niche architectures or customized firmware, due to the lack of matching parsing modules, it is difficult to complete firmware extraction and vulnerability scanning, resulting in obvious blind spots in the detection process. This cannot meet the detection needs of large-scale IoT devices and limits the application scope of automatic detection technology.

3.2. Lag in vulnerability feature extraction and weak ability to identify unknown vulnerabilities

At present, most automatic detection technologies rely on vulnerability feature databases for matching detection. However, IoT firmware vulnerabilities are evolving rapidly, and new types of vulnerabilities continue to emerge. The update of feature databases cannot keep pace with the evolution of vulnerabilities, leading to a lag in the identification of new vulnerabilities. In addition, there are few detection methods for unknown vulnerabilities. Most detection tools can only identify known vulnerabilities, and it is difficult to discover potential unknown vulnerabilities through methods such as behavior analysis and logical reasoning [2]. Furthermore, protection measures such as firmware encryption and code obfuscation further increase the difficulty of vulnerability feature extraction and reduce the accuracy of detection.

3.3. Insufficient collaboration in detection processes and lack of dynamic verification mechanisms

At this stage, the automatic detection process generally has the phenomenon of disconnection between static analysis and dynamic testing. The static analysis stage cannot accurately determine the actual exploitability of vulnerabilities, while the dynamic testing stage lacks effective linkage with the results of static analysis, resulting in low detection efficiency. Many detection tools only complete the preliminary screening of vulnerabilities, lacking a complete dynamic verification mechanism, so it is difficult to reproduce the vulnerability triggering process in real attack scenarios and define the actual hazard level of vulnerabilities. In addition, the insufficient accuracy of simulating the firmware operating environment during detection is prone to false positives and false negatives, which reduces the credibility of detection results.

DOI: 10.54254/2755-2721/2026.31419

4. Automatic detection technology scheme for IoT firmware vulnerabilities targeting core problems

4.1. Construction of a universal firmware parsing system with multi-architecture adaptability

To address the problem of adapting to heterogeneous device firmware, a universal firmware parsing system with multi-hardware architecture compatibility and multi-format parsing capabilities is built. This system adopts a flexible and scalable modular architecture design concept, integrates dedicated parsing units corresponding to mainstream hardware architectures, and covers common architecture types such as reduced instruction set and complex instruction set. It optimizes the parsing logic according to the instruction set characteristics of different architectures to ensure the stability and completeness of the parsing process. The system embeds an adaptive format recognition engine synchronously. To cope with the differences in firmware packaging of different manufacturers, it realizes automatic recognition and accurate parsing conversion of customized firmware, compressed package format firmware and encrypted firmware through multi-dimensional technical methods such as firmware file header feature matching, segment structure in-depth analysis and data offset verification. For encrypted firmware, it integrates a general decryption algorithm resource library and a dynamic key extraction unit, completes decryption processing by simulating the key call link in the firmware running environment, and restores the original running data of the firmware. To solve the problem of adapting to firmware of niche architectures, an extensible parsing plug-in platform is built, which supports the import, iterative update and flexible deployment of custom parsing rules, reducing the threshold for adapting to niche architectures [3]. After parsing, it automatically extracts key information such as code segments, configuration files, dependent components and system call interfaces in the firmware. After standardization with unified data format and field specifications, the information is accurately sent to the subsequent detection link, providing consistent and reliable data support for the full-process automatic detection.

4.2. Development of an intelligent feature extraction and unknown vulnerability mining mechanism

To tackle the pain points of lagging vulnerability feature extraction and weak unknown vulnerability identification, an intelligent detection model integrating static analysis and machine learning technology is built. In the static analysis stage, it accurately captures the code features and logical features associated with vulnerabilities through technical methods such as code slicing, control flow graph construction and semantic analysis, and builds a dynamically iterative vulnerability feature resource library based on these features to ensure the rapid matching and identification of discovered vulnerabilities. Deep learning algorithms are introduced to train massive historical vulnerability sample data, generating an intelligent vulnerability feature identification model to achieve autonomous mining and real-time update of new vulnerability features. For unknown vulnerabilities, a mining system based on behavioral anomaly detection is built. By constructing a firmware simulation running environment, it monitors behavioral data such as memory operations, system calls, network interactions and resource occupancy of the firmware under normal input and abnormal input scenarios, and establishes a behavioral baseline for the normal operation of the firmware. If the monitored behavioral data deviates from the baseline threshold, it is marked as a potential vulnerability and the complete behavioral trajectory is recorded. At the same time, symbolic execution technology is introduced to conduct full traversal analysis of key code paths, accurately locating the code fragments that may cause vulnerabilities.

DOI: 10.54254/2755-2721/2026.31419

4.3. Construction of a full-process detection framework with static-dynamic collaboration

To address the pain points of insufficient collaboration in detection processes and lack of dynamic verification mechanisms, a full-process automatic detection framework with deep collaboration between static analysis and dynamic testing is built. This framework sets three core links: static pre-detection, dynamic verification and result calibration, and all links achieve seamless connection through data interfaces. In the static pre-detection link, based on the aforementioned intelligent feature extraction model, full-scale detection is carried out on the standardized firmware data, and a list of suspected vulnerabilities, as well as information such as corresponding code locations and preliminary determination of vulnerability types, are output. In the dynamic verification link, a high-fidelity firmware running environment is built based on virtualization technology, customized test cases are generated according to the types of suspected vulnerabilities, the vulnerability triggering conditions in real attack scenarios are reproduced, the actual exploitability of vulnerabilities is verified, and the system state change data during the vulnerability triggering process is recorded synchronously. In the result calibration link, the code feature data from static analysis and the behavior state data from dynamic testing are integrated, the suspected vulnerabilities are re-verified by means of a rule engine, false positives caused by factors such as environmental differences and code redundancy are eliminated, and a high-precision detection report including vulnerability location, hazard level, triggering conditions and repair suggestions is generated [4]. The framework embeds a process scheduling module to realize automatic triggering of each link, automatic data flow and real-time feedback of detection results, which significantly improves detection efficiency and result credibility. See Figure 1.

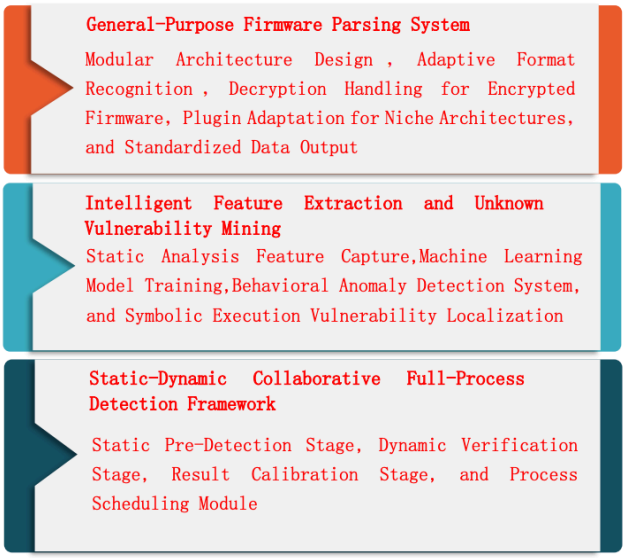

Figure 1. Automated vulnerability detection technology framework for IoT firmware

5. Evaluation of implementation effects of automatic detection technology scheme for IoT firmware vulnerabilities

5.1. Evaluation of multi-scenario detection coverage capability

Based on the constructed universal firmware parsing system, heterogeneous device firmwares from core IoT application fields such as industrial control, smart home and medical equipment are

DOI: 10.54254/2755-2721/2026.31419

specially selected for multi-scenario coverage capability testing. The test samples not only fully cover mainstream hardware architecture types, but also intentionally include typical heterogeneous samples such as customized firmwares from different manufacturers, encrypted firmwares and firmwares of niche architectures. At the same time, they take into account device firmware types at different technical iteration stages, fully conforming to the diversified and differentiated distribution characteristics of IoT devices in practical applications, so as to ensure the comprehensiveness and representativeness of test scenarios. The test results show that through modular compatibility design and adaptive format recognition mechanism, the technical scheme can realize complete parsing of firmwares of various mainstream architectures. The decryption adaptation for encrypted firmwares and compatibility for customized firmwares have both achieved significant improvement, which can effectively cover the detection needs of firmwares of niche architectures and successfully fill the gap of traditional detection schemes in heterogeneous device adaptation. After multiple rounds of cross-validation of batch firmware samples in multiple application scenarios, the scheme can stably support the automatic detection of IoT device firmwares across fields. Compared with traditional schemes, the detection coverage is greatly expanded, which can fully adapt to the detection application needs of large-scale IoT devices and meet the device security detection requirements of different scenarios such as industrial production, home life and medical services.

5.2. Evaluation of vulnerability detection accuracy and effectiveness

A set of firmware samples covering different application scenarios and complexities is selected for verification. The samples include known vulnerability types and potential unknown vulnerability situations, focusing on verifying the detection accuracy and practical application effectiveness of the intelligent feature extraction and unknown vulnerability mining mechanism. In the known vulnerability detection link, the scheme focuses on the accurate capture of code features and logical associations through static analysis, and simulates the vulnerability triggering scenarios in real operating environments combined with dynamic verification. Under the joint action of the two, false positives caused by factors such as code redundancy and environmental differences are successfully eliminated, and the accuracy of vulnerability identification is significantly optimized compared with the traditional feature matching mode. In the unknown vulnerability mining link, real-time monitoring of behavioral data fluctuations during firmware operation is carried out relying on behavioral anomaly detection technology. Combined with the in-depth analysis capability of symbolic execution technology, potential vulnerabilities that have not been publicly disclosed are accurately located, and the behavioral trajectory of vulnerability triggering and code association path are completely retained, providing direct technical support for vulnerability repair [5]. After multiple rounds of systematic test verification, the scheme maintains stable performance in unknown vulnerability mining, effectively solves the core problem of traditional schemes' lag in identifying new and unknown vulnerabilities, and fully demonstrates the effectiveness of technical application.

5.3. Evaluation of detection efficiency and resource occupation rationality

A set of firmware samples covering low, medium and high complexity gradients is selected to conduct a special evaluation on the operation efficiency and resource occupation rationality of the full-process detection framework with static-dynamic collaboration. In terms of detection efficiency, relying on the automated flow design and real-time data sharing mechanism of parsing, detection and verification links, the scheme realizes seamless connection of all processes, avoids process interruptions and repeated operations caused by manual intervention, and greatly shortens the

DOI: 10.54254/2755-2721/2026.31419

detection cycle of a single firmware. Compared with the traditional manual auditing mode and single static detection scheme, this framework significantly improves detection throughput through process closed-loop optimization, can stably support the rapid batch detection of massive firmware samples, and fully meets the needs of large-scale detection scenarios. In terms of resource occupation, the framework adopts a modular resource scheduling strategy, dynamically allocates computing resources according to the complexity of detection tasks, and optimizes the resource allocation of non-core detection links. On the premise of ensuring undiminished detection accuracy, it effectively controls memory consumption and computing resource overhead, and can flexibly adapt to different hardware deployment environments such as regular servers and embedded detection devices. In addition, the detection reports generated by the scheme are standardized in structure and comprehensive in information, covering key contents such as vulnerability location, hazard level, triggering conditions and repair suggestions. No additional manual sorting and supplementation are required, which further improves the conversion efficiency of detection results and lays a solid foundation for the rapid response and decision-making implementation of firmware security protection.

6. Conclusion

This paper focuses on the systematic research of automatic detection technology for IoT device firmware vulnerabilities. Combined with the security situation of in-depth popularization of devices, it accurately judges the core problems of existing technologies in heterogeneous adaptation, feature extraction, process collaboration and other dimensions. On this basis, a universal parsing system with multi-architecture adaptation, an intelligent vulnerability mining mechanism and a full-process detection framework with static-dynamic collaboration are constructed to form a complete technical scheme. After multi-dimensional verification, the scheme effectively expands detection coverage, improves the accuracy and efficiency of vulnerability detection, controls resource costs, and lays a solid technical foundation for firmware security protection. In the future, we can deepen technical adaptation capabilities, refine detection algorithms and scheduling mechanisms, promote the deep integration of technology and the full-life-cycle security management of devices, strengthen the systematicness and forward-lookingness of vulnerability prevention and control, and help improve the IoT security protection system.

References

[1] Peng, Y. (2025). Research and implementation of dynamic vulnerability detection techniques for intelligent Internet of Things device firmware (Doctoral dissertation, Beijing University of Posts and Telecommunications).
[2] Zhang, H., & Xie, Y. (2024). A review of firmware vulnerability detection technologies for the Internet of Things. Software Guide, 23(5), 205–211.
[3] Chen, S., & Huang, X. (2022). Research on vulnerability detection technologies for Internet of Things devices. Information & Computer (Theory Edition), 34(21), 219–221.
[4] Situ, L. (2020). Research on software vulnerability detection technologies for Internet of Things devices (Doctoral dissertation, Nanjing University).
[5] Li, T., Tian, Y., Ge, Y., et al. (2022). A review of security detection technologies for Internet of Things firmware vulnerabilities. Journal of Information Security Research, 8(12), 1146–1155.