

Potential Threats and Prevention Strategies of IoT Robots to Network Information Security

Wanrong Bai,^{1,a,*} Zuohu Chen,^{2,b} Lei Zhang,^{1,c} Di Wang^{1,d} and Rangle Wu^{2,e}

¹Electric Power Research Institute of State Grid Gansu Electric Power Company, 730070, Gansu, China

²Gansu Tongxing Intelligent Technology Development Co., LTD., 730050, Gansu, China

This paper presents a range of holistic prevention strategies comprising hardware attack mitigations, firmware vulnerabilities management, vulnerability patches of the operating system, malware protections, encryption solutions for communicating information, strong authentications, access permission management, automatic systems updating mechanisms, vulnerability patches management, and the monitoring of networking and detection of anomalies. Through practical application, the effectiveness of the protection mechanisms for IoT robot security and performance has been gradually integrated and evaluated. In addition, performance monitoring tools such as PerfMon and top have been used to log system response time and resource consumption of an IoT robot under varying system loads after security measures had been deployed. The system response time did increase slightly, but it was still within an acceptable range, and did not have a significant impact on the performance of the system itself. Overall, the proposed holistic prevention strategies demonstrated its ability to provide a level of security protection to the IoT robot.

Keywords: Internet of Things robots; network information security; potential threats; prevention strategies; vulnerability detection

1. INTRODUCTION

With the fast growth of Internet of Things (IoT) technology, IoT robots are being used in an increasing number of areas. However, the security threats presented by these robots are becoming more and more visible, and cannot be neglected [1–2]. The current traditional methods are inadequate for the protection of network information security for IoT robots. For instance, the security research is lagging, as it does not allow for a timely response to newly identified security threats, and the security design does not include systematic protective elements. In addition, there are no established standards and regulations that constrain systems to offer effective security protection measures. Finally, and perhaps most importantly, insufficient updates and maintenance leave

the system unable to respond to future attacks. Overall and ultimately, these issues adversely affect the security and reliability of IoT robot systems, and there is a need for new methods and strategies to improve their protection [3].

This study explores in detail the potential threats to network information security pertaining to IoT robots, and offers specific prevention methods to improve their overall level of protection. The study deploys the following specific techniques and methods. First, a systematic threat analysis technique is used to determine threats that IoT robots face in terms of security: specifically, remote attack, data/IO breaches, malware attacks, and DDoS (Distributed Denial of Service) attacks. Subsequently, the advanced encryption standard (AES) algorithm encryption technique is used to protect the confidentiality and integrity of data during transmission and/or storage. Also, to ensure that access permissions for each component and/or user of the system adhere to its security policies and prevent breach of access permissions, this study employs a dynamic access

*Corresponding author. ^aEmail: baiwr0824@163.com, ^bChengzHU198909@163.com, ^czhanglei2964@163.com, ^dwangd49936@163.com, ^eWurle1989@163.com

control policy and method defined by user roles, the current operational context, and the minimum privileges, for users and an even stricter set of methods for components. Lastly, a real-time monitoring system and log analysis, and anomaly detection methodology is employed to monitor all activities of the system continuously and provide feedback to administrators and authorities, enabling them to respond to appropriately to developing security threats. Furthermore, this study emphasizes the accessibility and space of updates and vulnerability fixes of IoT robot systems in response to the dynamic threat landscape related to security. Fully utilizing these specific technical methods, this study demonstrates not only a comprehensive and adaptable security protection premise, in a way that advances the science, but also provides a valuable reference for future IoT security research and implementation. Through empirical research and performance assessment, this study demonstrates that the proposed methods can enhance system security and reliability, ultimately providing a solid basis for the secure operation of IoT robots in a complex networked environment.

2. LITERATURE REVIEW

In current research, scholars have explored from multiple perspectives the potential threats posed by IoT robots to network information security. Previous research has focused mainly on describing the types of attacks and threat sources that IoT robots may face, such as remote intrusion, malware infection, etc. [4–5]. Szykiewicz proposed a behavior recognition method based on deep learning, which achieved automatic detection and alarm of abnormal behavior of IoT robots. By using deep neural network models for real-time monitoring and analysis of robot operation behavior, abnormal behavior could be detected in a timely manner, improving the system's security response capability. However, this method still faced challenges of high computational resource consumption and further improvement in model generalization ability when dealing with large-scale data and complex network environments [6]. Malsa investigated the use of blockchain technology in securely authenticating IoT robots. In order to securely verify identities and provide traceability of data, blockchain's distributed ledger and smart contract technology were utilized for these objectives. Malsa noted that the practical use of blockchain technology for large-scale IoT applications was restricted by performance bottlenecks and high energy consumption [7]. Kabir used physical feature-based authentication methods, including biometric identification recognition and hardware security modules, for user or device verification via unique biometric or device hardware identifiers to mitigate risks of identity forgery and device tampering. However, this method may not always effectively protect the privacy of biological data, and interoperability between devices may be an issue [8]. Khan introduced a security isolation approach via virtualization technology, which isolated critical components and data flows within IoT robot systems by deploying virtualization layers. The authors reduced an attack surface and data breaches; however, virtualization technology can cause performance impact and reduced resource utilization

efficiency, thus highlighting the trade-off between security and performance [9]. These examples demonstrate both the significant efforts and advancements made in the field of contemporary research on security threats to IoT robots, but they also highlight the existing technological challenges and implementation obstacles that IoT robots will have to overcome before any meaningful or applicable conclusions are established. A systematic and comprehensive methodology is needed, along with more in-depth empirical research and appropriate response plans, in order to substantially improve the protection mechanisms of IoT robots in network information security in time.

To tackle the different challenges in current studies, several researchers have applied different methods to improve the security protection of IoT robot systems: threat analysis, real-time monitoring system, encryption technology, and dynamic access control. Ali proposed a threat analysis framework, based on deep learning, which monitored and detected abnormal behavior eyes on IoT robot systems in real-time by training models. Yet, it proves challenging to precisely detect and respond to complex and evolving threat scenarios during real-world implementation and deployment of IoT robot systems [10]. Zhu designed a real-time monitoring system that applies distributed sensor network and real-time data processing technology to enable prompt responses to security-sensitive events in IoT robot systems. However, higher resource investment and performance optimization were required in terms of data storage and processing capabilities [11]. Varma adopted the AES based on quantum key distribution (QKD), which effectively protected the security of IoT robots' communication data, although the long-term security of encryption algorithms and the real-time performance of the system are still the focus of attention in today's development of quantum computing capabilities [12]. Pal proposed a security management strategy involving role-based access control (RBAC), which allowed permissions to be adjusted according to the real-time status of users and devices. However, permission management and policy coordination still faced challenges in large-scale deployment and complex network environments [13]. Therefore, this present study applies threat analysis, encryption technology, access control, and real-time monitoring methods to propose a comprehensive and adaptive security protection strategy for various security threats that IoT robots may face, in order to fill the gaps in existing research. The application of these strategies is expected to strengthen the overall protection level of IoT robot systems in terms of network information security, providing valuable references for future IoT security research and practice.

3. CONSTRUCTION OF SECURITY SYSTEM

3.1 Construction of Threat Identification Module at The Device Level

(1) Hardware attack protection

To prevent physical tampering and side-channel attacks, this article deploys various hardware security technolo-

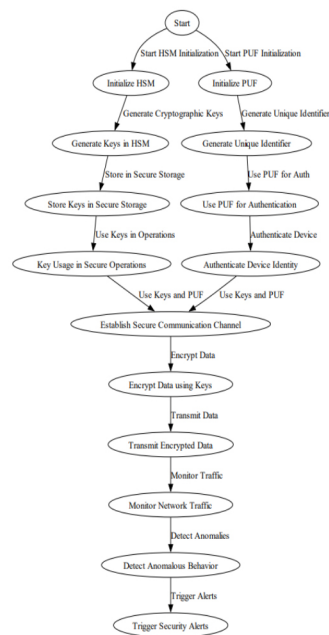


Figure 1 Working principle and implementation flowchart of the HSM and the PUF technology.

gies. The first of the specifications is the hardware security module (HSM), whose key advantage is that it provides encryption and decryption functions for sensitive data, augmenting the device's physical safeguard capability [14–15]. HSM uses purpose-built security chips inside the device to ensure that even if an attacker has physical access to the device, the data will be difficult to crack. The second of the specifications is the physical unclonable function (PUF) technology. PUF generates unique and non-elastic keys from the device's inherent physical characteristics, further strengthening hardware safety. The operating principle and implementation steps of the HSM and physical unclonable function PUF technology process are illustrated in Figure 1:

Moreover, this paper incorporates tamper-resistant sealing technology to offer physical protection and minimize unauthorized disassembly of and tampering with the device. Hardware components are regularly inspected and maintained so that timely identification and repair can be carried out.

(2) Firmware vulnerability management

Firmware vulnerability management is a key step in ensuring device security. This article employs automated firmware analysis tools such as Binwalk and Firmware Analysis Toolkit (FAT) to perform routine scanning and analysis of device firmware in order to detect potential security vulnerabilities in a timely manner [16]. The scan is performed with the following steps:

To begin, it is necessary to run network scanning tools that can thoroughly scan an IoT robot and associated devices, and keep a record of all connected devices. After all devices have been scanned, scanning parameters can be defined based on the device type, operating system, and configuration to account for all potential vulnerability points.

Vulnerability scanning tools are used in a proactive manner to automatically identify known vulnerabilities in operating systems, applications, and firmware. Once scanning is

completed, results are evaluated and categorized. The steps include: exporting reports from the scanning tool(s), including relevant details such as vulnerability description, scope of impact, common vulnerabilities and exposures (CVE) number, etc.; using the common vulnerability scoring system (CVSS) scoring criteria to rate each vulnerability, a score of high, medium, or low; classifying vulnerabilities according to severity and scope of impact for remediation categories.

To resolve identified vulnerabilities in a timely manner, we execute auto-updates of device firmware remotely using Over-The-Air (OTA) technology, and develop associated repairs for a classified vulnerabilities category. High-risk vulnerabilities are prioritized for repair using mostly patch updates or configuration adjustments or access control. Temporary remediation measures must be taken for high-risk vulnerabilities that cannot be repaired immediately, including isolation using network segmentation or temporary access control. Medium-risk vulnerabilities are repaired following high-risk vulnerabilities and are addressed mainly through patch management or configuration optimizations. Low-risk vulnerabilities are repaired after high-risk vulnerabilities and medium-risk vulnerabilities have been repaired, and can be managed Biswal through routine updates and regular maintenance.

After the repair is completed, it is necessary to conduct a vulnerability scan again to verify the effectiveness of the repair. The specific steps are: use the same vulnerability scanning tools to rescan the repaired system and confirm that the repaired vulnerabilities no longer exist; compare the rescan results with the initial scan results to ensure that high-risk and medium-risk vulnerabilities are effectively repaired and that low-risk vulnerabilities are reduced or eliminated; finally, record and archive the repair process and results, and generate a vulnerability repair report to provide a basis for subsequent audits and security assessments.

The vulnerability repair rate is calculated with Formula (1):

$$\text{Vulnerability repair rate} = \frac{\text{Number of vulnerabilities repaired}}{\text{Total number of vulnerabilities discovered}} \times 100\% \quad (1)$$

By regularly scanning and repairing vulnerabilities and monitoring changes in vulnerability repair rates, the continuous improvement of system security can be ensured.

3.2 Construction of Threat Protection Module at The Software Level

(1) Operating system vulnerability patching

To ensure the security of the operating system, this study uses an automatic patch management system to regularly check and install the latest security patches. By configuring system update policies and setting up scheduled tasks, this ensures that the operating system will be able to obtain and install patches in a timely manner. Furthermore, the security benchmark of the Center for Internet Security (CIS) was taken into account in order to audit the operating system to verify that its configuration aligns with best practice security protocols, and make adjustments according to the outcomes of the audit to mitigate the exposure vulnerabilities [17].

(2) Malware protection

Antivirus software and a signature-based and behavior-based firewall are employed to protect the system from malicious software. Antivirus software works by either removing malicious software that has been identified, or real-time monitoring or beyond installing software regular scanning. Firewalls classify the network access control policy to prevent suspicious traffic entering the system. If an outer layer of machine learning-based intrusion detection system (IDS) and intrusion prevention system (IPS) is connected, together with the long short-term memory (LSTM) deep learning system, then this would promptly detect attacks based on intrusion data [18–19]. The steps are as follows:

Data collection and preprocessing:

In the data collection and pre-processing stage, first, the network traffic collection tools are used to collect all traffic data in the IoT robot network environment. Subsequently, the collected traffic data is processed to eliminate redundant information and noise whilst retaining important attributes including source IP, destination IP, port, protocol type, and packet size. Then the feature extraction tools are used to extract features from traffic data and create feature vectors for machine learning mode application.

Model training and optimization:

In the model training and optimization phase, the collected traffic data is first annotated to distinguish between normal traffic and attack traffic, generating training and testing sets. Next, the LSTM model is trained using annotated data, and the model performance is optimized by adjusting hyperparameters. During the model validation process, the trained model is validated using a testing set to evaluate its accuracy, recall, precision, and F1-score. The formulas for accuracy, recall, precision, and F1-score are shown in Formulas (2), (3), (4), and (5), respectively:

$$\text{Accuracy} = \frac{TP + TN}{TP + TN + FP + FN} \quad (2)$$

$$\text{Recall} = \frac{TP}{TP + FN} \quad (3)$$

$$\text{Precision} = \frac{TP}{TP + FP} \quad (4)$$

$$\text{F1-score} = 2 \times \frac{\text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}} \quad (5)$$

Among them, TP is the number of correctly detected attacks; TN is the number of correctly detected normal traffic; FP is the number of normal traffic that is falsely reported as an attack; FN is the number of actual attacks missed.

Real-time monitoring and detection:

During the real-time monitoring and detection phase, the trained intrusion detection model is deployed to the IoT robot network environment to analyze network traffic in real time. When suspicious behavior or attack patterns are detected, the system generates alarm information and records detailed attack characteristics and behavior logs.

The calculation formula for attack detection rate is shown in Formula (6):

$$\text{Attack detection rate} = \frac{\text{Actual number of attacks detected}}{\text{Actual total number of attacks}} \times 100\% \quad (6)$$

The above is the attack detection section. The positions and working principles of the antivirus software and the firewall in network traffic, including the transmission process of data packets and detailed information of each node, are shown in Figure 2:

3.3 Construction of Data Transmission Security Measures Module

(1) Encrypted communication

To protect the confidentiality and integrity of data being transmitted, this study adopts advanced encryption technology. In terms of specific implementation, the AES symmetric encryption algorithm is used to encrypt the data stream, and AES-256-bit encryption provides extremely high security. Meanwhile, the Rivest-Shamir-Adleman (RSA) asymmetric encryption algorithm is used to encrypt the symmetric key, ensuring the security of key transmission. This study also combines the Diffie-Hellman key exchange protocol to securely exchange encryption keys between communicating parties, preventing the keys from being intercepted during transmission [20]. The AES encryption algorithm flow and RAS encryption algorithm flow are shown in Figure 3(a) and Figure 3(b), respectively:

(2) Security protocol application

To prevent data interception and tampering, secure communication protocols such as Transport Layer Security/Secure Sockets Layer (TLS/SSL) are widely used [21–22]. These protocols, by encrypting the data transmission channel, ensure that data is not stolen or tampered with during transmission. Moreover, this study configures the mandatory HyperText Transfer Protocol Secure (HTTPS)

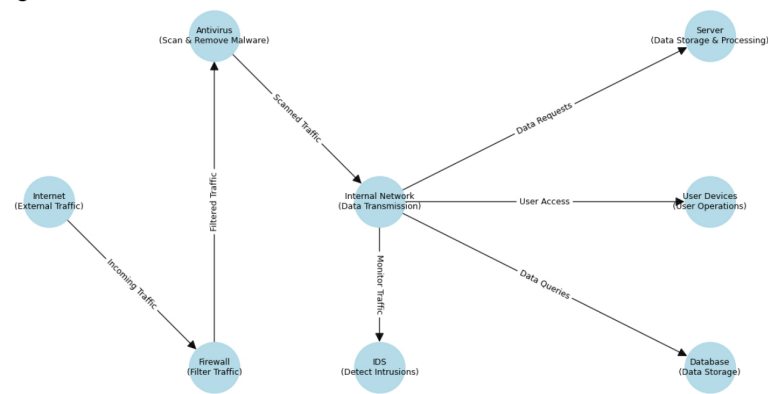


Figure 2 Positions and workflow diagram of the antivirus software and the firewall.

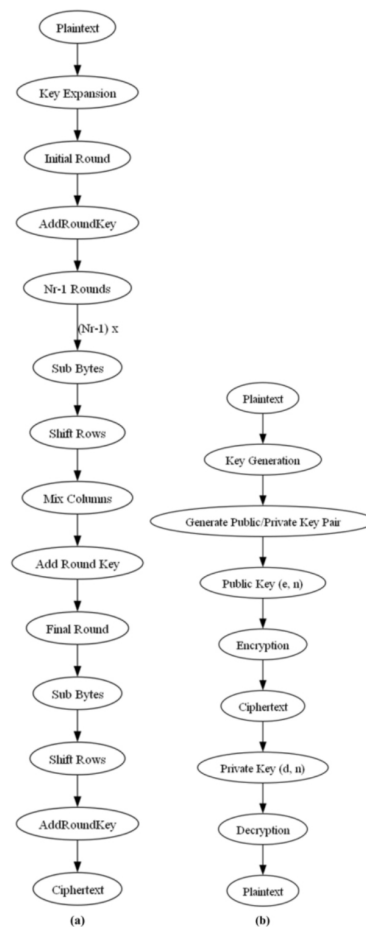


Figure 3 Flowchart of AES and RAS encryption algorithms.

to ensure that all data is transmitted through encrypted channels. The identity of all the communicating parties are verified through a digital certificate written by the Certificate Authority (CA) that prevents man-in-the-middle attacks.

3.4 Construction of Device Authentication and Access Control Module

(1) Strong authentication

In order to prevent unauthorized access, we implemented multi-factor authentication (MFA). MFA is an authentication

mechanism that uses more than one credential, such as matching passwords with a hardware token, or requiring a biometric feature such as a fingerprint or facial recognition. Specifically, the MFA mechanism operates through the OAuth 2.0 protocol, where the user's access to critical system resources necessitates that they must go through multiple authentication steps so that we provide increased security [23]. The workflow of MFA is shown in Figure 4:

(2) Access permission management

This study provides the foundation for strict access control policies, role-based access control, and minimum privilege

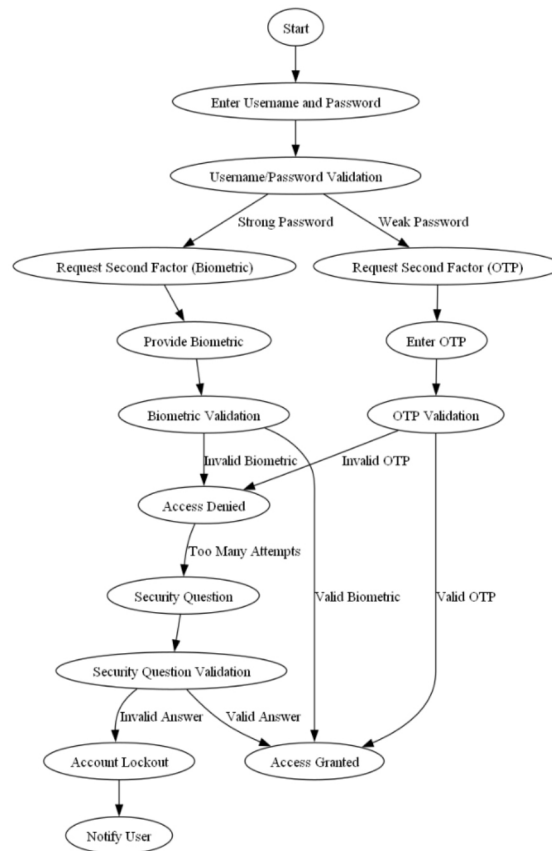


Figure 4 Workflow diagram of MFA.

principles, which allow only authorized users and devices to access sensitive data and functionality. ACL and permission management systems allow dynamic adjustments to user permissions ensuring that each user has only necessary minimum permissions to accomplish his or her tasks and to minimize unauthorized access [24].

3.5 Construction of Device Update and Maintenance Module

(1) Automatic update mechanism

To ensure the devices can receive the latest security patches in a timely fashion, an automatic updating mechanism is put into place. By using the Over-The-Air (OTA) method, devices can automatically download and install a package of updates [25]. In this study, Mender, a software update management system, runs the entire updating process and safeguards the security and integrity of the update process.

(2) Vulnerability patching management

This study establishes a vulnerability management process, using the vulnerability scanning tool, Nessus, to routinely scan the system and identify security vulnerabilities in a timely manner. A vulnerability management system to track the progress of vulnerability fixes is configured to ensure all known vulnerabilities are patched promptly. The vulnerability management process is reviewed and updated on a regular basis to enable it to respond to emerging security

threats and challenges [26–27]. The vulnerability patching process is illustrated in Figure 5:

3.6 Construction of Network Monitoring and Anomaly Detection Module

(1) Real-time network monitoring

To monitor the network activity of devices in real time, this study uses Nagios as the network monitoring system. These systems can detect and respond to abnormal network behavior in real time by configuring monitoring rules and alarm policies [28]. By monitoring network traffic, device connectivity status, and network performance indicators, potential security threats can be identified in a timely manner.

(2) Abnormal behavior detection

This study conducts in-depth analysis of system logs and identifies potential threats through log analysis and an anomaly detection tool: Splunk. Machine learning algorithms are used to automatically identify and handle abnormal behaviors. Log analysis involves collecting and analyzing device operation logs to identify abnormal patterns and behaviors. Anomaly detection tools train and detect abnormal activities through machine learning models, and trigger alerts [29–30].

The architecture process and abnormal behavior detection process of the real-time network monitoring system are shown in Figure 6(a) and Figure 6(b) respectively:

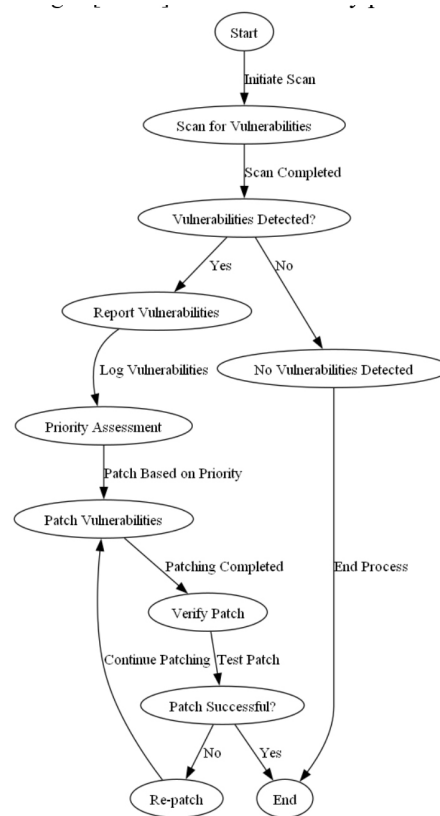


Figure 5 Vulnerability patching process diagram.

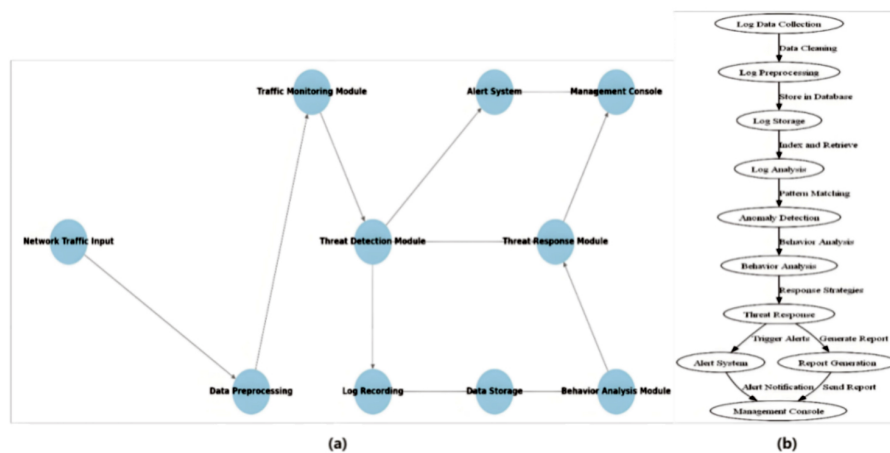


Figure 6 Architecture of monitoring system and flowchart of abnormal behavior detection.

4. EVALUATION OF THE EFFECTIVENESS OF SECURITY SYSTEMS

In this study, a detailed evaluation of the security of IoT robots is measured by two indicators: vulnerability repair rate and attack detection rate. The following are the specific implementation steps and methods used.

4.1 Security Indicators

(1) Vulnerability repair rate

Vulnerability scanning and classification:

This study uses FAT as a scanning tool to conduct a comprehensive vulnerability scan of IoT robots. After scanning multiple robots used for testing, the scanning results are obtained, shown in Table 1:

Vulnerability repair and verification:

The detected vulnerabilities are repaired and retested after the repair is completed. The detection results are shown in Table 2. Only the number of vulnerabilities is displayed here.

According to the data analysis in Table 2, the system shows significant results after vulnerability repair: all high-risk vulnerabilities are effectively repaired; the repair rate

Table 1 Vulnerability detection data.

Device ID	Vulnerability Description	Impact Scope	CVE ID	Number of Devices	High-risk Vulnerabilities	Medium-risk Vulnerabilities	Low-risk Vulnerabilities
Device 1	Remote Code Execution	All Versions of Operating System	CVE-2023-12345	5	1	1	0
Device 2	Privilege Escalation	Application Permission Management Module	CVE-2023-23456	5	0	1	1
Device 3	Denial of Service Attack	Network Communication Module	CVE-2023-34567	4	1	0	2
Device 4	Information Disclosure	Data Transmission Encryption Module	CVE-2023-45678	4	0	2	0
Device 5	Buffer Overflow	System Kernel	CVE-2023-56789	6	2	1	1
Device 6	SQL Injection	Database Management Module	CVE-2023-67890	6	1	1	2
Device 7	Cross-Site Scripting	User Input Handling Module	CVE-2023-78901	7	1	2	0
Device 8	Directory Traversal	File System Access Module	CVE-2023-89012	7	2	1	1
Device 9	Security Authentication Bypass	User Authentication Management Module	CVE-2023-90123	6	1	1	1
Device 10	Data Tampering	Data Integrity Protection Module	CVE-2023-01234	6	1	2	0

Table 2 Data after vulnerability repair.

Device	High-risk Vulnerabilities	Medium-risk Vulnerabilities	Low-risk Vulnerabilities	Device	High-risk Vulnerabilities	Medium-risk Vulnerabilities	Low-risk Vulnerabilities
Device 1	0	0	0	Device 6	0	0	1
Device 2	0	0	0	Device 7	0	0	0
Device 3	0	0	1	Device 8	0	0	0
Device 4	0	0	0	Device 9	0	1	1
Device 5	0	0	0	Device 10	0	0	0

Table 3 Attack detection data.

Traffic sample number	Traffic data volume (GB)	Normal traffic number	Number of attack traffic	Total amount of annotated data	Number of attacks detected by the model	Number of false positives
Data sample 1	20	10000	1000	11000	800	20
Data sample 2	15	8000	800	8800	650	15
Data sample 3	18	9000	900	9900	750	18
Data sample 4	25	12000	1200	13200	1000	25
Data sample 5	22	11000	1100	12100	900	20
Data sample 6	30	15000	1500	16500	1250	30
Data sample 7	20	10000	1000	11000	850	20
Data sample 8	28	14000	1400	15400	1150	28
Data sample 9	24	12000	1200	13200	1000	25
Data sample 10	18	9000	900	9900	780	20

of medium-risk vulnerabilities is also extremely high, with only one case remaining unresolved; there are still three low-risk vulnerabilities. The result calculated by the vulnerability repair rate formula is about 86.7%, which strongly proves that the system has efficient vulnerability detection and repair capabilities and can handle most of the detected security vulnerabilities. However, it is worth noting that the repair rate for low-risk vulnerabilities is significantly lower compared to high-risk and medium-risk vulnerabilities. This may be attributed to the relatively low priority setting for repairing low-risk vulnerabilities, resulting in a lack of equal attention and handling in practical operations.

(2) Attack detection rate

This study deploys a machine learning-based intrusion detection system (IDS) and a deep learning model LSTM in an IoT robot system to monitor and identify potential network attack behaviors. The data obtained from the testing is shown in Table 3:

The performance evaluation indicators of the system for detecting attacks, such as accuracy, recall, F1-score, and attack detection rate, are obtained by combining the data in the table with Formulas (2)–(6), as shown in Table 4:

Based on the experimental results of the 10 different-sized datasets listed in Table 4, the system demonstrates stable performance. Its accuracy remains within the range of 93% to 96%, indicating that in the vast majority of cases, the system can effectively distinguish between normal traffic and attack traffic, achieving a lower false alarm rate. In terms of recall rate, the system achieves a level of 88% to 92%, indicating that it can capture most of the attack behaviors, although there is still room for improvement to achieve the capture of all potential threats. This situation could lead to optimization of detection algorithms and improve forecasting extraction strategies to further reduce missed detection.

As observed, the F1-score remains steady between 91% and 93.5%, which is a high-level indicator that represents the system's excellent balance for accuracy and recall, and it shows that it can accurately detect and respond to attack behaviors in complicated network environments. The attack detection rate falls within the 80% to 87% range, which means that the system is able to cover most attacks. In order to ensure that future protection standards are higher, optimization of algorithm performance, and application of

more diverse detection feature sets are suggested in order to decrease false alarm rates and improve protection effectiveness.

In short, the performance of the IoT robot security system is excellent overall in relation to detection of attacks with high accuracy and a good F1 score. That said, as new network threats continue to emerge, the system must still continue to optimize recall and attack detection rates to provide a more complete and efficient level of security protection.

4.2 Performance Indicators

(1) System response time

The changes in response time after constructing a security system are measured by recording the average response time of the system under different loads and comparing the differences before and after building the security system. The system response time is continuously monitored through the performance monitoring tool PerfMon, and the impact of the prevention system on overall system performance is evaluated. After multiple tests, the average response time of the system under different load conditions before and after building the prevention system was obtained, as shown in Figure 7:

In Figure 7, the blue bar represents the response time before building the security system, and the orange bar represents the response time after building the security system. After the building the security system, the average response time and fluctuation of the system both increase, especially at the maximum load, where the average response time changes by 120ms. Despite the fact that adding the security system causes the response time to change and the magnitude of deviation to increase, the total amount of change remains within predictable bounds, indicating that the introduction of the security system has a limited effect on performance. Such data is helpful for conducting a more in-dept analysis of the performance of security preventive systems, providing assurance that while the security advantage is being provided, there is no significant deterioration in performance.

(2) Resource consumption

The impact of security systems on device resources such as CPU and memory usage is assessed using these steps:

Table 4 Data table for performance evaluation of detection system.

Traffic sample number	Detection accuracy	Detection recall rate	F1-Score	Attack detection rate	Traffic sample number	Detection accuracy	Detection recall rate	F1-Score	Attack detection rate
Data sample 1	95%	90%	92%	80%	Data sample 6	94%	90%	92%	83%
Data sample 2	94%	88%	91%	81%	Data sample 7	95%	92%	93.5%	85%
Data sample 3	96%	91%	93%	83%	Data sample 8	93%	89%	91%	82%
Data sample 4	95%	92%	93.5%	83%	Data sample 9	94%	90%	92%	83%
Data sample 5	93%	89%	91%	82%	Data sample 10	96%	91%	93%	87%

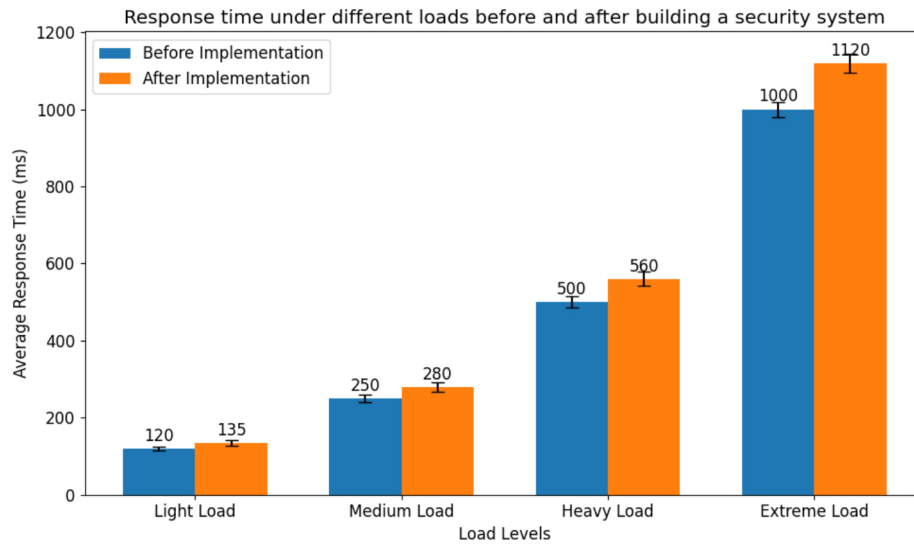


Figure 7 System response time under different load conditions before and after building a security system.

first, the performance monitoring tool, top, is used to record the resource usage before building the security system to obtain benchmark data; subsequently gradually the various security measures are integrated, including hardware attack protection, firmware vulnerability management, operating system vulnerability patching, malware protection, encrypted communication, security protocol application, strong authentication, access permission management, automatic update mechanism, vulnerability patching management, real-time network monitoring, and abnormal behavior detection. After integrating each security measure, the resource usage of the device is retested, and the new CPU and memory usage are recorded for the purpose of comparative analysis. By analyzing resource consumption data, it is ensured that the security system can provide effective security protection without significantly affecting system performance. The experimental data obtained from the testing are shown in Table 5:

The data in Table 5 shows that without any security measures, the system's resource consumption is relatively low, and CPU and memory usages gradually increase with increasing load. Subsequently, 12 security protection measures are gradually integrated. As hardware attack protection is integrated into the security system along with firmware vulnerability management, the increase in CPU and memory usages after integration is greater than that when one protection measure is integrated. After the security measures have been integrated, the CPU and memory usage rates gradually increase, as follows:

Idle state: The CPU usage increases from 5% to 17%, and the memory usage increases from 10% to 22%.

Low-load state: The CPU usage increases from 20% to 44%, and the memory usage increases from 30% to 54%.

Medium-load state: The CPU usage increases from 50% to 74%, and the memory usage increases from 60% to 84%.

High-load state: The CPU usage increases from 80% to 104%, and the memory usage increases from 90% to 114%.

Hence, it can be seen that after integrating security protective measures, the resource consumption of the system increases, but it does not have a significant impact on system performance. Overall, these security measures not only enhance the network information security of IoT robots, but also have relatively little impact on system performance.

5. CONCLUSION

This paper provides a thorough examination of the possible threats against the network information security of IoT robots, and proposes measures for comprehensive prevention including: hardware attack protection, firmware vulnerability management, OS vulnerability patching, malware protection, encrypted communication, strong authentication, management of access rights, automatic update mechanism, vulnerability patching management and network monitoring and anomaly detection. The experimental results show that the set of prevention strategies can perform effectively in terms of vulnerability repair rate and attack detection rate with a vulnerability repair rate of 86.7% and an attack detection rate where the rate is between 80% and 87%, ultimately improving the overall level of security protection of the IoT robots. Despite the application of security mechanisms increased overall system resource consumption and response time, and after extensive monitoring and analyzing system performance, this study showed that the overall impact is still within an acceptable level. Future research directions should focus on optimizing detection algorithms and proposals for feature extraction mechanisms in order to better the effectiveness of security protection of the IoT robots and continue to enable efficient operation and information security of the system. These findings provide valuable empirical data and guidance for research the security of IoT robots and related fields, which can help promote the development and application of IoT security technology.

Table 5 Gradual performance changes of integrated systems under different loads.

	No security system		Integrating hardware attack protection and firmware vulnerability management		Integrating operating system vulnerability patching		Integrating malware protection	
	CPU usage (%)	Memory usage (%)	CPU usage (%)	Memory usage (%)	CPU usage (%)	Memory usage (%)	CPU usage (%)	Memory usage (%)
Idle	5	10	7	12	8	13	9	14
Low load	20	30	24	34	26	36	28	38
Medium load	50	60	54	64	56	66	58	68
High load	80	90	84	94	86	96	88	98

	Integrating encrypted communication		Integrating security protocol application		Integrating strong authentication		Integrating access permission management	
	CPU usage (%)	Memory usage (%)	CPU usage (%)	Memory usage (%)	CPU usage (%)	Memory usage (%)	CPU usage (%)	Memory usage (%)
Idle	10	15	11	16	12	17	13	18
Low load	30	40	32	42	34	44	36	46
Medium load	60	70	62	72	64	74	66	76
High load	90	100	92	102	94	104	96	106

	Integrating automatic update mechanism		Integrating vulnerability patching management		Integrating real-time network monitoring		Integrating abnormal behavior detection	
	CPU usage (%)	Memory usage (%)	CPU usage (%)	Memory usage (%)	CPU usage (%)	Memory usage (%)	CPU usage (%)	Memory usage (%)
Idle	14	19	15	20	16	21	17	22
Low load	38	48	40	50	42	52	44	54
Medium load	68	78	70	80	72	82	74	84
High load	98	108	100	110	102	112	104	114

REFERENCES

1. Sayeed, A., Verma, C., Kumar, N., Koul, N., & Illés, Z. (2022). Approaches and challenges in Internet of robotic things. *Future Internet*, 14(9), 265-295. <https://doi.org/10.3390/fi14090265>
2. Yaacoub, J. P. A., Noura, H. N., Salman, O., & Chehab, A. (2022). Robotics cyber security: Vulnerabilities, attacks, countermeasures, and recommendations. *International Journal of Information Security*, 21(1), 115-158. <https://doi.org/10.1007/s10207-021-00545-8>
3. Rahmani, A. M., Bayramov, S., & Kiani Kalejahi, B. (2022). Internet of things applications: opportunities and threats. *Wireless Personal Communications*, 122(1), 451-476. <https://doi.org/10.1007/s11277-021-08907-0>
4. Zaman, S., Alhazmi, K., Aseeri, M. A., Ahmed, M. R., Khan, R. T., Kaiser, M. S., & Mahmud, M. (2021). Security threats and artificial intelligence based countermeasures for internet of things networks: a comprehensive survey. *IEEE Access*, 9, 94668-94690. <https://doi.org/10.1109/ACCESS.2021.3089681>
5. Altulaihan, E., Almaiah, M. A., & Aljughaiman, A. (2022). Cybersecurity threats, countermeasures and mitigation techniques on the IoT: Future research directions. *Electronics*, 11(20), 3330-3371. <https://doi.org/10.3390/electronics11203330>
6. Szynekiewicz, W., Niewiadomska-Szynekiewicz, E., & Lis, K. (2023). Deep Learning of Sensor Data in Cybersecurity of Robotic Systems: Overview and Case Study Results. *Electronics*, 12(19), 4146-4169. <https://doi.org/10.3390/electronics12194146>
7. Malsa, N., Vyas, V., Gautam, J., Shaw, R. N., & Ghosh, A. (2021). Framework and smart contract for blockchain enabled certificate verification system using robotics. *Machine Learning for Robotics Applications*, 125-138. https://doi.org/10.1007/978-981-16-0598-7_10
8. Kabir, H., Tham, M. L., & Chang, Y. C. (2023). Internet of robotic things for mobile robots: concepts, technologies, challenges, applications, and future directions. *Digital Communications and Networks*, 9(6), 1265-1290. <https://doi.org/10.1016/j.dcan.2023.05.006>
9. Khan, A. A., Laghari, A. A., Shaikh, Z. A., Dacko-Pikiewicz, Z., & Kot, S. (2022). Internet of Things (IoT) security with blockchain technology: A state-of-the-art review. *IEEE Access*, 10, 122679-122695. <https://doi.org/10.1109/ACCESS.2022.3223370>

10. Ali, M. H., Jaber, M. M., Abd, S. K., Rehman, A., Awan, M. J., Damaševičius, R., & Bahaj, S. A. (2022). Threat analysis and distributed denial of service (DDoS) attack recognition in the internet of things (IoT). *Electronics*, 11(3), 494-512. <https://doi.org/10.3390/electronics11030494>
11. Zhu, Z., Lin, Z., Huang, J., Zheng, L., & He, B. (2023). A digital twin-based machining motion simulation and visualization monitoring system for milling robot. *The International Journal of Advanced Manufacturing Technology*, 127(9), 4387-4399. <https://doi.org/10.1007/s00170-023-11827-x>
12. Varma, R., Melville, C., Pinello, C., & Sahai, T. (2021). Post quantum secure command and control of mobile agents inserting quantum-resistant encryption schemes in the secure robot operating system. *International Journal of Semantic Computing*, 15(03), 359-379. <https://doi.org/10.1142/S1793351X21400092>
13. Pal, S., & Jadidi, Z. (2021). Protocol-based and hybrid access control for the IoT: Approaches and research opportunities. *Sensors*, 21(20), 6832-6858. <https://doi.org/10.3390/s21206832>
14. Cabrera-Gutiérrez, A. J., Castillo, E., Escobar-Molero, A., Álvarez-Bermejo, J. A., Morales, D. P., & Parrilla, L. (2022). Integration of hardware security modules and permissioned blockchain in industrial IoT networks. *IEEE Access*, 10, 114331-114345. <https://doi.org/10.1109/ACCESS.2022.3217815>
15. Dutta, V., & Zielińska, T. (2021). Cybersecurity of robotic systems: Leading challenges and robotic system design methodology. *Electronics*, 10(22), 2850-2872. <https://doi.org/10.3390/electronics10222850>
16. Pu, H., He, L., Cheng, P., Sun, M., & Chen, J. (2022). Security of industrial robots: Vulnerabilities, attacks, and mitigations. *IEEE Network*, 37(1), 111-117. <https://doi.org/10.1109/MNET.116.2200034>
17. Botta, A., Rotbei, S., Zinno, S., & Ventre, G. (2023). Cyber security of robots: A comprehensive survey. *Intelligent Systems with Applications*, 18, 200237-200253. <https://doi.org/10.1016/j.iswa.2023.200237>
18. Yadav, C. S., Singh, J., Yadav, A., Pattanayak, H. S., Kumar, R., Khan, A. A. & Alharby, S. (2022). Malware analysis in IoT & android systems with defensive mechanism. *Electronics*, 11(15), 2354-2372. <https://doi.org/10.3390/electronics11152354>
19. Riaz, S., Latif, S., Usman, S. M., Ullah, S. S., Algarni, A. D., Yasin, A., ... & Hussain, S. (2022). Malware detection in internet of things (IoT) devices using deep learning. *Sensors*, 22(23), 9305-9327. <https://doi.org/10.3390/s22239305>
20. Liang, W., Ning, Z., Xie, S., Hu, Y., Lu, S., & Zhang, D. (2021). Secure fusion approach for the internet of things in smart autonomous multi-robot systems. *Information Sciences*, 579(1), 468-482. <https://doi.org/10.1016/j.ins.2021.08.035>
21. Fernández-Jiménez, F. J., & Martínez-de Dios, J. R. (2021). A Robot-Sensor Network Security Architecture for Monitoring Applications. *IEEE Internet of Things Journal*, 9(8), 6288-6304. <https://doi.org/10.1109/IIOT.2021.3113756>
22. Xun, T., Liu, Z., & Jiang, B. (2026). Adaptive fuzzy security control of uncertain semi-Markov jump cyber-physical systems with stochastic perturbation. *Cyber-Physical Systems*, 12(1), 31-50.
23. Oruma, S. O., & Petrovic, S. (2023). Security threats to 5G networks for social robots in public spaces: a survey. *IEEE Access*, 11, 63205-63237. <https://doi.org/10.1109/ACCESS.2023.3288338>
24. Golightly, L., Modesti, P., Garcia, R., & Chang, V. (2023). Securing distributed systems: A survey on access control techniques for cloud, blockchain, IoT and SDN. *Cyber Security and Applications*, 1(7), 100015-100035. <https://doi.org/10.1016/j.csa.2023.100015>
25. Sadhu, P. K., Yanambaka, V. P., & Abdelgawad, A. (2022). Internet of things: Security and solutions survey. *Sensors*, 22(19), 7433-7484. <https://doi.org/10.3390/s22197433>
26. Lesi, V., Jakovljevic, Z., & Pajic, M. (2021). Security analysis for distributed IoT-based industrial automation. *IEEE Transactions on Automation Science and Engineering*, 19(4), 3093-3108. <https://doi.org/10.1109/TASE.2021.3106335>
27. Sotiropoulos, P., Mathas, C. M., Vassilakis, C., & Kolokotronis, N. (2023). A Software Vulnerability Management Framework for the Minimization of System Attack Surface and Risk. *Electronics*, 12(10), 2278-2299. <https://doi.org/10.3390/electronics12102278>
28. Rawat, R., Rajawat, A. S., Mahor, V., Shaw, R. N., & Ghosh, A. (2021). Surveillance robot in cyber intelligence for vulnerability detection. *Machine Learning for Robotics Applications*, 107-123. https://doi.org/10.1007/978-981-16-0598-7_9
29. Haji, S. H., & Ameen, S. Y. (2021). Attack and anomaly detection in iot networks using machine learning techniques: A review. *Asian Journal of Research in Computer Science*, 9(2), 30-46. <https://doi.org/10.9734/ajrcos/2021/v9i230218>
30. Xu, H., Sun, Z., Cao, Y., & Bilal, H. (2023). A data-driven approach for intrusion and anomaly detection using automated machine learning for the Internet of Things. *Soft Computing*, 27(19), 14469-14481. <https://doi.org/10.1007/s00500-023-09037-4>
31. Ying, Y., Zhou, S. & Shen, B. (2026). Autonomous Decision-Making and Control Algorithm for Live Working Robots Based on Artificial Intelligence. *International Journal of Engineering Intelligent Systems*, 34(2), 195-209.
32. Wang, B. & Ye, Y. (2025). Construction of Internet of Things Resource-Sharing Platform Based on Intelligent Information Processing from the Perspective of Network Security. *International Journal of Engineering Intelligent Systems*, 33(3), 239-248.