Hindawi
Advances in Multimedia
Volume 2022, Article ID 6039690, 9 pages
https://doi.org/10.1155/2022/6039690

Research Article

Computer Deep Learning Network Security Vulnerability Detection Based on Virtual Reality Technology

Xiaokun Zheng

Yantai Gold College, Yantai, Shandong 265401, China

Correspondence should be addressed to Xiaokun Zheng; 20163075@ayit.edu.cn

Received 18 February 2022; Revised 3 April 2022; Accepted 15 April 2022; Published 5 May 2022

Academic Editor: Qiangyi Li

In order to detect the computer network security technology vulnerabilities due to various factors, the normal operation of the computer network must be ensured, the user's confidential information must be protected, and it is proposed that the analysis and research on security vulnerability detection must be strengthened. This study introduces the working principle of the network security monitoring system, analyzes the key technologies involved in the system development process and network programming technology, gives the overall architecture of the system, and designs the processing flow of the monitoring function. The test and analysis of the system show that the design of the system has achieved the expected design goal. The design of the system has achieved the expected design goal. The five test points can meet the standard time specified in the demand analysis process. The time difference of all module test points in the test is less than 3 s. The system can realize the remote acquisition and real-time monitoring of the network access, file system operation, system operation status, and other information of the controlled computer. Through the test and analysis of the system, it is shown that the system has achieved the expected design goal, the working state problem, and can meet the functional requirements of internal network security monitoring. It can be applied to enterprises, institutions, and departments that have higher requirements for intranet information security.

1. Introduction

At present, "network security" has not yet had a fixed definition, and its specific definition category is also different from different starting angles. In essence, network security refers to information security issues, that is, through multiparty management, to effectively ensure the security of software and hardware data, and to prevent information damage, leakage, and other accidents. From a macroperspective, network security research includes not only network management but also related technologies to ensure the authenticity and integrity of information. Through the dual support of technology and management methods, the protection of information can be effectively achieved. Network security vulnerability detection is one of the key means to eliminate network security risks and ensure the stable operation of the system. As shown in Figure 1, for the increasingly complex network security situation, this study proposes an automatic network security vulnerability detection method based on virtual reality technology, analyzes the implementation process of network security vulnerability detection automation, and provides a theoretical reference for the development of related research. The basic requirements of network security include reliability, availability, confidentiality, and integrity: ① reliability, that is, under certain conditions and within a time range, the network information system can maintain its due functional characteristics; ② availability, that is, through the network setting, authorized users can access and use related network information; ③ confidentiality, that is, to ensure that illegal users cannot access and use network information, and prevent information from being tampered and leaked; and ④ integrity, that is, the security of network information transmission or storage process, to prevent illegal deletion, tampering, and insertion. The integration of network technology and information technology in modern production and life is deepening, information interaction and transmission activities are frequently carried out, and the

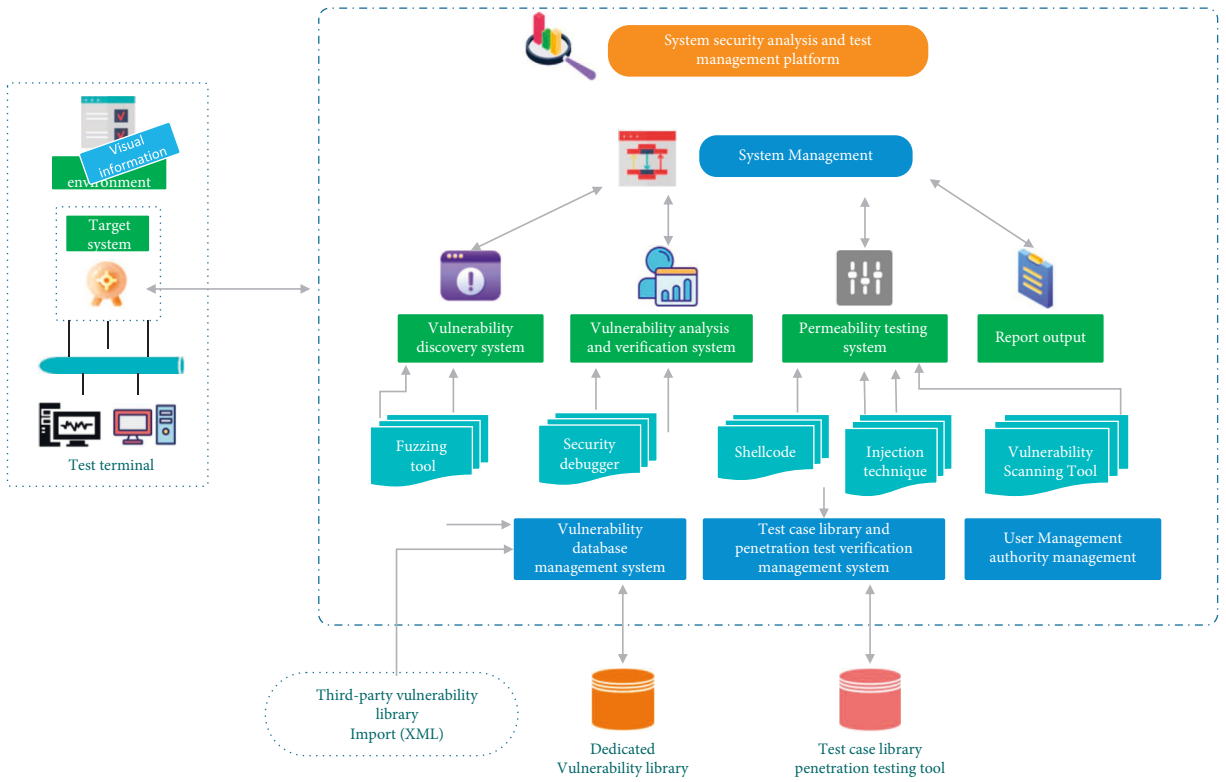

Figure 1: Flowchart of network security vulnerability detection.

influence of the openness and freedom of the network itself makes the network security problems more and more complex. Once there are security problems such as system intrusion, information loss, and leakage, it will bring huge benefit losses to users. At the same time, it affects the operation order of the network environment [1]. The development of virtual reality technology brings new ideas to the treatment of network security problems. It is necessary to analyze and summarize its automatic detection methods [2].

2. Literature Review

In recent years, with the rapid development of computer application technology and network technology, especially the continuous promotion and application of the internet, Shirakabe et al. found that network security has increasingly become one of the hotspots of attention [3]. Li et al. found that due to the openness of the internet and the convenience of access through the network, all kinds of network attacks can cross space constraints and remotely destroy the normal operation of enterprise and government computer information systems [4]. At the same time, Sun et al. found that people in the internal system visit illegal internet sites, spread illegal information, abuse internal information resources, and even illegally steal internal trade secrets [5]. Zagane et al. put forward that network security monitoring technology is specifically aimed at internal network security problems. Its basic idea is to timely find users' violations in the process of using internal network resources through real-time monitoring of host behavior or internal network data and force users to abide by the internal network security management system through technical means such as remote control, so as to help managers effectively manage and maintain the information resources in the internal network [6]. Li et al. found that the research on internal network security detection technology started early. In the early 1980s, the detection method of finding exceptions by analyzing the system log information inside the host was proposed for the first time [7]. Under the guidance of this idea, Satan, the first network security detection software, was released in 1995. The system requires a high technical level of users and is difficult for ordinary users to master. Subsequently, there are many tools with similar functions, whose main working principle is based on the detection and audit of user behavior, such as the security configuration editor SCE in wondows environment and the audit tool of the UHX system. The limitation of these tools is that their functions mainly focus on analysis. Through the processing of collected data, the user can obtain the state when using the system but does not provide or only provides limited management functions. At present, Lin et al. proposed that network security detection system has become an important part in the field of network security [8]. In terms of research, academia mainly focuses on the analysis model of the safety detection system, such as machine learning. Guo et al. aimed at artificial intelligence, data mining, and other technologies to realize the in-depth analysis of user behavior or system log data. The analysis content not only includes traditional host-based user behavior and local

system call but also gradually transitions to user network access behavior and network data [9]. Al Abassi a. et al. found that in terms of application, more and more security manufacturers have also launched their own network security monitoring systems. Manufacturers mainly focus on the support of security audit standards and national laws. Therefore, they rarely independently release internal network detection software with management function. In China, Hanbang Minimally Invasive released the "comprehensive strong audit monitoring system for information security" in 2000, which can provide illegal internal and external online supervision, user network behavior audit, mobile storage management, host authorization management, database operation audit, intelligent report system, etc. [10]. Yen et al. found that information security companies such as Qiming Xingchen and Tianrongxin also launched corresponding software products, and these professional products have strong functions [11]. Chen et al. proposed that it is mainly aimed at the general security management requirements, and the network security detection system in a special environment is mainly developed and deployed in the way of software customization [12]. For very important data, it must be remembered to encrypt it. When choosing a computer system, it must be tried to choose a computer with high availability. The advantage of this type of computer is that when there is a problem with the system or hardware device of the computer equipment itself, the computer can quickly switch commands to ensure uninterrupted operation and reduce losses. The important data are regularly backed up, which can effectively avoid the problem of important data loss due to equipment damage.

3. Method

The overall functional structure of the system is divided into six parts: user identity management module, real-time monitoring module, hardware object management module, software object management module, network object management module, and file object management module, as shown in Figure 2.

The form of the sigmoid function is shown as follows:

$$\sigma(x) = \frac{1}{1 + \exp(-x)}. \tag{1}$$

The form of the ReLU function is shown as follows:

$$\begin{aligned} \sigma(x) &= \max(0, x), \\ &= \begin{cases} x, & (x \geq 0) \\ 0, & (x < 0) \end{cases}. \end{aligned} \tag{2}$$

From the ReLU function form, it can be concluded that the gradient is 1 for $x \geq 0$, and otherwise, 0. That is, when $x \geq 0$, the gradient saturation effect of the sigmoid function is completely eliminated, and at the same time, it helps to increase the convergence speed of the stochastic gradient descent method. The input layer of the auto-encoder is a k-dimensional input vector x, and the mapping function is shown in formula (3):

$$\begin{aligned} y &= f\theta(x), \\ &= \sigma(W \cdot x + b), \end{aligned} \tag{3}$$

where x is the input vector, b is the offset, W is the weight, and $\sigma(x)$ is the activation function. The function of the activation function is that when the neuron receives the input value and compares it with the neuron's threshold, the activation function is processed to generate the neuron's output. The main function of the activation function is to transform the linear mapping in the network into a nonlinear mapping, so as to autonomously perform feature learning; otherwise, there is no substantial difference with the single-layer neural network. In the decoding process, the hidden layer y reversely reconstructs z to the decoding layer, and the reverse reconstruction function is shown as follows:

$$\begin{aligned} z &= f\theta'(y), \\ &= \sigma(W' \cdot y + b'), \end{aligned} \tag{4}$$

where $w\prime$ is the weight, and b' is the offset. The whole process of auto-encoding adjusts the weight and offsets by minimizing the loss function $L(x, z)$ between the network output and the real value, so that the low-dimensional code layer data can be used as the expression of high-dimensional data, so that the low-dimensional data of the last layer of the hidden layer can be used as the expression of input samples to better characterize normal samples and malicious samples. To achieve the optimization, see formula (5):

$$\theta, \theta' = \min_{\theta, \theta'} \sum_{i=1}^{n} L\left(x_j, Z_j\right). \tag{5}$$

Among them, θ and θ' represent the input vector and the network output vector, respectively, and L is the loss function.

The speed can have a better fitting ability, and the ReLU function is used as the activation function, that is, $f(x) = \max(0, x)$. The convolutional neural network has the feature of weight sharing, which can better express higher-level features. The r must be used to represent the area where the weights are shared and b to represent the bias. The formula for the convolution process is shown as follows:

$$y^{j(r)} = \max\left(0, b^{j(r)} + \sum_{i}^{ij(r) \times xi(r)}\right). \tag{6}$$

The feature map obtained after the input data is convolved is expressed as

$$S \in R^{(m-p+1) \times (n-q+1)}. \tag{7}$$

Aiming at the problem of too large data dimension and overfitting, the pooling layer is connected after the convolution layer to find the local optimal features. The fully connected layer is composed of a second pooling layer and a pooled convolutional layer, which can not only learn the local optimal features of the sample but also learn the global features, which can effectively reduce the loss of possible important features. This is very important for the expression of features. The formula for this layer is as follows:

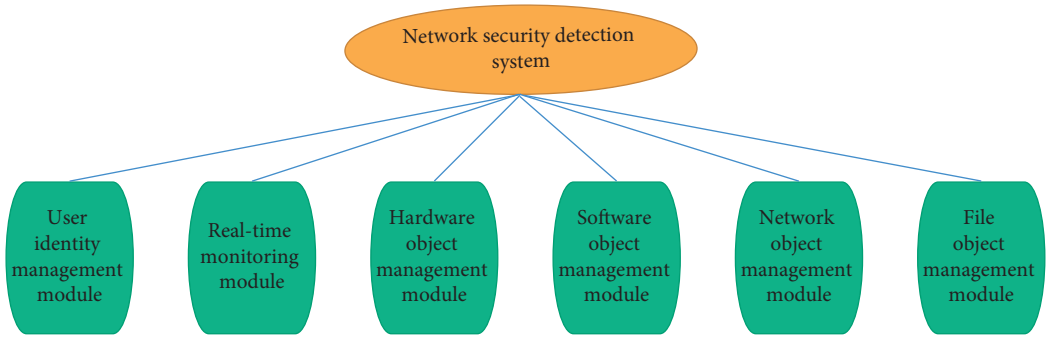

Figure 2: Overall functional structure of the system.

$$yj = \max\left(0, \sum_i x_i^o \times w_{i,j}^o + \sum_i x_i^1 \times w_{i,j}^1 + b_j\right). \quad (8)$$

The output layer of the model is the softmax layer, which outputs two nodes, so as to achieve the purpose of Android malicious application detection. The expression formula of this layer is shown as follows:

$$yi = \frac{\exp(y_i')}{\sum_{j=1}^{2} \exp(y_j')}. \quad (9)$$

User identity mainly completes the functions of adding and deleting system users. The system adopts a role-based access control model to realize user identity management. The main operations include creating and deleting roles [13]. After you select a role, you can modify it (modify role name and role description) and control rule settings. The control rules are set by selecting the policy, the new policy corresponding to the role is written to the database, and the security detection agent applies the new policy. In the process of creating roles, first the administrator permissions must be judged. If you have permission, duplicate name detection on the role name must be performed. If there is no duplicate name, a new role in the database should be created and the role name and policy options should be filled in. If the insertion is successful, the processing result is returned. Otherwise, the prompt of insertion failure is returned. Real-time monitoring mainly includes viewing and recording the process list, real-time desktop, and file directory of the online controlled host. The design idea is to use UDP to send monitoring instructions to the detection agent deployed on the controlled host. After receiving the monitoring instructions, the detection agent will read the local process list, real-time desktop screenshot, file directory information, and other operations according to the instruction type obtained after parsing, and return the corresponding information to the server. The logical processing flow of file directory acquisition and desktop snapshot in real-time monitoring is similar to process monitoring. Both detection agents read the corresponding information and return the corresponding query results to the server. Finally, the security detection server displays the output on the administrator's user interface [14]. The main function of software object management is that the security detection server sends the software black-and-white list strategy to the controlled host and the security detection agent execution strategy to prohibit the software on the blacklist from executing on the controlled host. The processing flow of the server is shown in Figure 3.

After the software black-and-white list is set, the software black-and-white list update notification will be sent to the detection agent, and the software blacklist definition function belongs to the software object management function module. The administrator can add and delete the software blacklist, including the start and end time. After the administrator's setting is completed, the software blacklist will be written to the database server. After enabling the software security management policy, the security detection agent will read the software blacklist in the database and check whether the software is on the blacklist when the controlled host starts the software. If it belongs to the blacklisted software, the security detection agent will end the corresponding process, and run and send alarm information to the security detection server. After receiving the notification, the detection agent will read the new software black-and-white list in the database and execute the software object control strategy according to the new list [15].

(1) The software operation control process of the security detection agent is

(2) Obtain the black-and-white list of software

(3) Get the first process of the system

(4) Judge whether the process is on the software blacklist. If yes, kill the process; otherwise, enter the next step

(5) Judge whether the process has been obtained. If yes, go to the next step; otherwise, obtain the next system process

(6) Wait for the system to create a process. If the system creates a new process, judge whether the process is on the software blacklist. If it is, block the process creation; otherwise, continue to wait

The hardware object management mainly completes the acquisition of the hardware list of the controlled host, the early warning of the change of hardware equipment, and the information acquisition of the use status of peripherals. The processing flow of the security detection server for obtaining

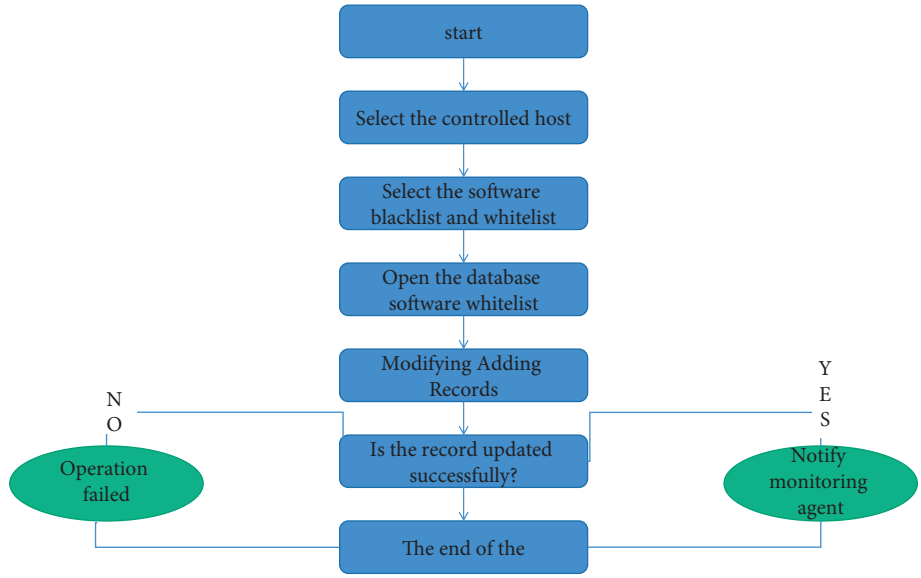

Figure 3: Software black-and-white list policy-setting process.

the hardware list of the controlled host is to first select the specified controlled host, query the hardware list information of the controlled host from the database, and display it after analysis [4]. In order to ensure that the security detection server can read the correct hardware list of the controlled host, the security detection agent must regularly check the hardware list of the host and synchronize it to the database. Its processing flow is as follows:

(1) The security detection agent obtains the current hardware object list of the machine stored in the database server every time it starts

(2) The security detection agent compares the obtained hardware object list in the database server with the previous hardware object list locally cached

(3) If the comparison results are the same, no operation will be carried out

(4) If the comparison results are different, the local hardware object list should be updated, the list should be updated to the database, and the local log should be recorded at the same time

Peripheral use management can enable or disable some peripheral interfaces (floppy disk, optical drive, and modem dial-up internet access) of the host at the controlled end. When receiving the instruction to enable or disable peripheral interfaces, the security detection agent will allow or prohibit the controlled host user to perform application operations based on these peripheral interfaces [16, 17]. The processing flow is as follows:

(1) When the security detection agent receives a new peripheral policy change message, the policy should be taken out and cached for preprocessing, that is, the command of the security detection server should be formatted into a standard command format, and the corresponding module for processing should be called according to the command classification

(2) The security detection agent obtains its corresponding role from the policy, polls all policies corresponding to the role, and determines whether the policy exists or whether the policy has changed

(3) If the policy is not changed, the cache must be cleared and directly exit; otherwise, the new policy should be executed

(4) The policy should be updated, and the cache should be deleted

(5) The update time should be recorded as a system log

Network object management is mainly to control network access and audit network traffic on the controlled machine [18, 19]. The processing process is that the security detection server formulates the black-and-white list of website access, and after distribution, the security detection agent executes the access control strategy on the controlled machine; the traffic audit is carried out by the security detection agent for data statistics and recording, and the server can query the traffic statistics of the specified controlled host through the database. The black-and-white list processing process of the security detection agent's website is as follows:

(1) The black-and-white list of websites should be analyzed: the websites that are not allowed or allowed to be accessed from the list should be extracted.

(2) URL filtering: each URL requested should be matched by the controlled host with the URL in the list. If the requested URL is legal, it will be normally accessed. If it is illegal, it will be blocked, as shown in Figure 4.

The flow audit process is as follows:

(1) The security detection agent records the local traffic: all network packets of the controlled host in unit time should be captured and the real-time traffic should be calculated.

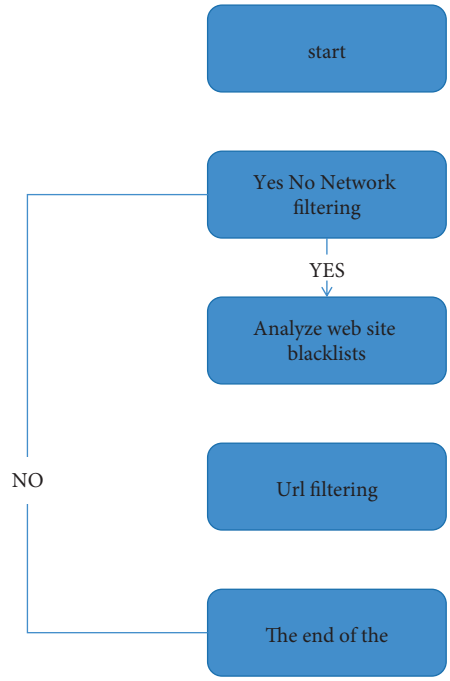

Figure 4: Process flow of URL black-and-white list control.

(2) The security detection agent analyzes the local traffic: the real-time traffic should be compared with the specified legal peak. If the local traffic exceeds the legal peak for a period of time (the length of this period can be specified), the local traffic is abnormal.

(3) Security detection agent notifies the server: if the local traffic is abnormal, the server must be alarmed, as shown in Figure 5.

In the network security detection system, the security detection agent is transparent to users, and the security detection server provides management.

Graphical interface for operator operation includes the following interfaces.

(1) Login interface: the administrator enters the main interface of the security detection server console through this interface

(2) Console main interface: the administrator can perform relevant operations through this interface

(3) Controlled host list interface: through this interface, the administrator selects the controlled host to monitor the operation status in the real time

(4) Controlled host system information viewing interface: the administrator can view the system information of a controlled host through this interface

(5) Screen monitoring interface of the controlled host: through this interface, the administrator can view the current screenshot of a controlled host and lock/unlock the mouse and keyboard operation of the controlled host

(6) Controlled host process information viewing interface: the administrator can view the process information of a controlled host through this interface and terminate a process of the controlled host

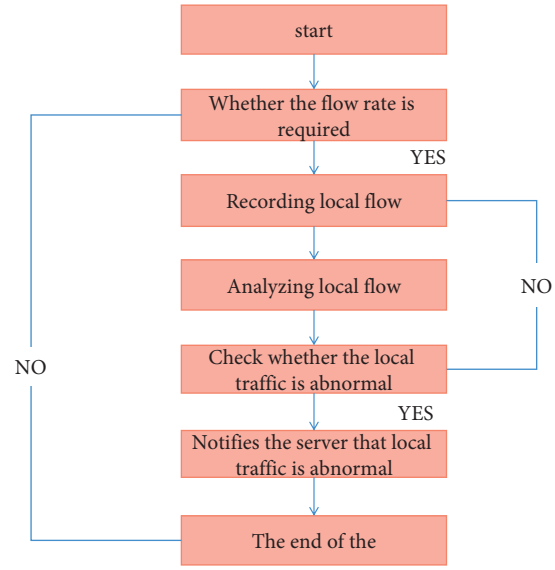

Figure 5: Flowchart of controlled host traffic audit.

(7) Controlled host file information viewing interface: the administrator can view the file information in a directory of a controlled host through this interface

(8) Controlled host hardware list viewing interface: the administrator can view the hardware list of a controlled host through this interface and authorize the host to update the hardware list in its database

(9) Log audit interface: the administrator queries the database through this interface to view and analyze the security event information of all controlled hosts

(10) Health audit interface: through this interface, the administrator queries the database to view and analyze the health forensic information of all controlled hosts, including screenshots, system information, and process information

(11) Configure event security-level interface: the administrator sets the security level for each security event through this interface

(12) Entity view and management interface: the administrator can browse and set the entity through this interface

(13) Role viewing and management interface: the administrator can browse and set roles through this interface

(14) Interface for viewing and managing entities to be assigned: administrators can browse and set new entities to be assigned through this interface

(15) Policy configuration interface: the administrator sets policies through this interface

(16) Exit interface: the administrator exits the console through this interface

Table 1: Add equipment test cases.

Test case name	New authorized equipment
Test process	The user enters the interface of new authorized equipment and fills in the MAC address, IP address, transmission power, and other information of the equipment according to the interface prompt and submits it after completion
Expected results	Equipment information filled in by new users in the legal access point information table of the system
Actual results	Consistent with expected results

Table 2: Retrieve test cases.

Test case name	Detection of intrusion feature knowledge base retrieval
Preconditions	The user has system permission to log in to the system
Test process	Log in to the system and enter the search interface of intrusion feature knowledge base Submit after filling in the query statement
Expected results	The system returns intrusion features consistent with user query information
Actual results	Consistent with expected results

4. Results and Analysis

The function test is completed by using the automatic test framework selenium [6, 8]. We can use the browser to complete the selenium test and simulate users. Mozilla Firefox, Mozilla Suite, and other browsers support selenium testing.

Taking adding legal access device information as an example, the test for adding access devices is shown in Table 1.

The system administrator enters the device addition interface through the add device option. The user inputs the basic information of the equipment, and the system reviews the basic information of the equipment. If the approval is passed, the basic information will be added to the database. If the approval is not passed, the user will be prompted to fill in again [5, 7]. The administrator enters the equipment addition interface and fills in the equipment type, equipment name, equipment status, coordinates, and user name for managing the equipment as required. After filling in, the equipment information should be submitted and the addition of the equipment information should be completed.

The test of the intrusion feature retrieval function is shown in Table 2.

The system uses LoadRunner software to simulate 100 users to log in to the wireless network, and then, the administrator carries out relevant operations to record the response time of the system when users operate. During this test, five parts are selected for testing, which are called test points [20, 21]. In the whole test process, each test point needs to be operated on many times. In the test in this study, 10 operations are selected, and the response time of each operation is recorded. From the recorded time of each operation, the average consumption time of the selected test point in the user's use process can be calculated. With the average time consumption, it can be compared with the specified time in the demand analysis. If the time difference exceeds 3 s, it does not meet the requirements, and within 3 s, it meets the requirements. The concurrent test system uses LoadRunner software to simulate 100 users logging into the wireless network, and then, the administrator carries out relevant operations to record the response time of the system during user operation. In this test process, five parts are selected for the test, which are called test points. In the whole test process, each test point needs to be operated on many times. In the test in this study, 10 operations are selected, and the response time of each operation is recorded. From the recorded time of each operation, the average consumption time of the selected test point in the user's use process can be calculated. With the average time consumption, it can be compared with the specified time in the demand analysis. If the time difference exceeds 3 s, it does not meet the requirements, and within 3 s, it meets the requirements. The corresponding time results of the concurrent test are shown in Figure 6.

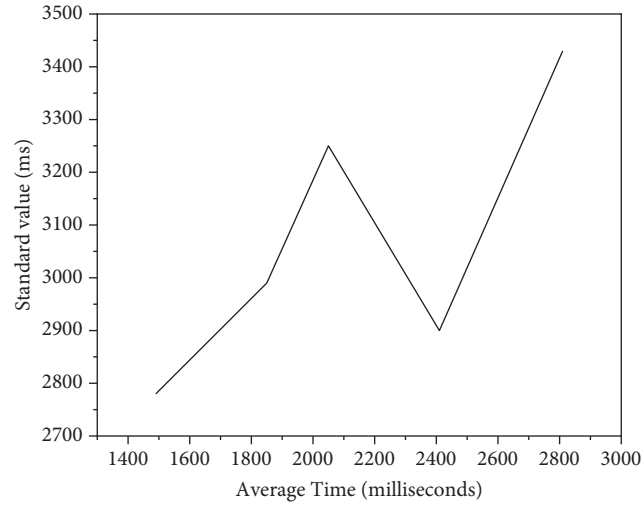

Figure 6: Concurrent test results.

From the analysis and description of test case results in Figure 6, it can be seen that the five test points can meet the standard time specified in the demand analysis process, and the time difference of all module test points in the test is less than 3 s, which meets the requirements set by the system. Therefore, the results of the concurrent test meet the system settings. Through the performance test of the system, it is

found that the performance of the wireless network security detection system has reached the expected goal of the system. Through the introduction of the design process of the network security detection system, the design of the system's functional structure and functional processing logic is given, and the core data structure, user interface, database structure, and security design process used in the system are discussed, described, and obtained satisfactory results.

5. Conclusions

Through the research and development work involved in this study, we have a more in-depth understanding of the working principle and key technologies of network security monitoring system. On this basis, the overall architecture and specific monitoring function of the system are designed. The completed system can realize the remote acquisition and real-time monitoring of the network access, file system operation, system operation status, and other information of the controlled computer. The test and analysis of the system show that the system has achieved the expected design objectives. The five test points can meet the standard time specified in the demand analysis process. The time difference of all module test points in the test is less than 3 s, which meets the requirements set by the system and can meet the functional requirements of internal network security monitoring, and it can be used in enterprises, institutions, and departments that have higher requirements for intranet information security. With the development and popularization of computer network technology, the informatization of management not only improves the work efficiency but also introduces new security risks and management problems. A network security monitoring system is an important technical means for an internal network security audit. It can be widely used to ensure internal information security and strengthen internal security management. The currently implemented system can basically complete the function of security detection, but there are still deficiencies in the following aspects that need to be improved. The security detection scope of the controlled host should be more comprehensive. For example, the supervision of instant messaging software and e-mail software can only be controlled through the software blacklist, and the function of a content audit is not realized. The detection granularity is relatively coarse. The above problems will be focused on and solved in the next step.

Data Availability

The data used to support the findings of this study are available from the corresponding author upon request.

Conflicts of Interest

The authors declare no conflicts of interest.

References

[1] S. Chander, V. Padmanabha, and J. Mani, "Jaya spider monkey optimization-driven deep convolutional lstm for the prediction of covid'19," *Bio-Algorithms and Med-Systems*, vol. 16, no. 4, pp. 145–151, 2020.

[2] Y. Bao, Z. Tang, H. Li, and Y. Zhang, "Computer vision and deep learning-based data anomaly detection method for structural health monitoring," *Structural Health Monitoring*, vol. 18, no. 2, pp. 401–421, 2019.

[3] S. Shirakabe, H. Kataoka, K. Iwata, and Y. Satoh, "Robust person detection in a semi-shielding environment based on the camera array and deep learning," *Journal of the Japan Society for Precision Engineering*, vol. 82, no. 12, pp. 1067–1071, 2016.

[4] Z. Li, D. Zou, J. Tang, Z. Zhang, and H. Jin, "A comparative study of deep learning-based vulnerability detection system," *Journals & Magazines*, vol. 7, no. 99, 2019.

[5] H. Sun, L. Cui, L. Li, Z. Ding, and Z. Hao, J. Cui and P. Liu, "VDSimilar: vulnerability detection based on code similarity of vulnerabilities and patches," *Computers & Security*, vol. 110, Article ID 102417, 2021.

[6] M. Zagane, M. K. Abdi, and M. Alenez, "Deep learning for software vulnerabilities detection using code metrics," *IEEE Access*, no. 99, 1 page, 2020.

[7] Z. Li, D. Zou, S. Xu, H. Jin, Y. Zhu, and Z. Chen, "SySeVR: a framework for using deep learning to detect software vulnerabilities," *IEEE Transactions on Dependable and Secure Computing*, vol. 8, no. 99, 1 page, 2021.

[8] G. Lin, S. Wen, Q. L. Han, J. Zhang, and Y. Xiang, "Software vulnerability detection using deep neural networks： a survey," *Proceedings of the IEEE*, vol. 108, no. 99, pp. 1–24, 2020.

[9] H. Guo, S. Huang, C. Huang, M. Zhang, and X. Wang, "A lightweight cross-version binary code similarity detection based on similarity and correlation coefficient features," *IEEE Access*, vol. 8, no. 99, 1 page, 2020.

[10] A. Al-Abassi, H. Karimipour, A. Dehghantanha, and R. M. Parizi, "An ensemble deep learning-based cyber-attack detection in industrial control system," *IEEE Access*, vol. 8, no. 99, 1 page, 2020.

[11] Y.-S. Yen and H.-M. Sun, "An android mutation malware detection based on deep learning using visualization of importance from codes," *Microelectronics Reliability*, vol. 93, pp. 109–114, 2019.

[12] F.-C. Chen and M. R. Jahanshahi, "NB-CNN: deep learning-based crack detection using convolutional neural network and naïve bayes data fusion," *IEEE Transactions on Industrial Electronics*, vol. 65, no. 5, pp. 4392–4400, 2018.

[13] F. Xie, M. Shi, Z. Shi, J. Yin, and D. Zhao, "Multilevel cloud detection in remote sensing images based on deep learning," *Ieee Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, vol. 10, no. 8, pp. 3631–3640, 2017.

[14] H. C. Ke, H. Wang, H. W. Zhao, and W. J. Sun, "Deep reinforcement learning-based computation offloading and resource allocation in security-aware mobile edge computing," *Wireless Networks*, vol. 27, no. 5, pp. 3357–3373, 2021.

[15] N. Lv, C. Chen, T. Qiu, A. K. Sangaiah, and Qiu, "Deep learning and superpixel feature extraction based on contractive autoencoder for change detection in sar images," *IEEE Transactions on Industrial Informatics*, vol. 14, no. 12, pp. 5530–5538, 2018.

[16] S. Parkinson, A. Crampton, and R. Hill, "Guide to vulnerability analysis for computer networks and systems," *Computer Communications and Networks*, pp. 211–234, 2018.

[17] X. Ma, N. Kittikunakorn, B. Sorman et al., "Deep learning convolutional neural networks for pharmaceutical tablet defect detection," *Microscopy and Microanalysis*, vol. 26, no. 2, pp. 1606–1609, 2020.

[18] L. Li, L. Gang, L. Zhang, and Q. Li, "Deep learning-based sensor fault detection using s-long short term memory networks," *Structural Monitoring and Maintenance*, vol. 5, no. 1, pp. 51–65, 2018.
[19] Z. Yan, R. Molva, W. Mazurczyk, and R. Kantola, "Network and system security," *Lecture Notes in Computer Science*, pp. 169–183, 2017.
[20] F. Catani, "Landslide detection by deep learning of non-nadiral and crowdsourced optical images," *Landslides*, vol. 18, no. 3, pp. 1025–1044, 2021.
[21] P. Zhou, J. Liu, X. Liu, Z. Yang, and J. Grundy, "Is deep learning better than traditional approaches in tag recommendation for software information sites?" *Information and Software Technology*, vol. 109, pp. 1–13, 2019.