Scalable Computing: Practice and Experience, ISSN 1895-1767, http://www.scpe.org
© 2024 SCPE. Volume 25, Issues 1, pp. 35–44, DOI 10.12694/scpe.v25i1.2172

NETWORK SECURITY WITH VIRTUAL REALITY BASED ANTIVIRUS PROTECTION AND REDUCED DETECTION DELAYS

CHUNNA SONG[*] JINFANG CHENG[†] AND GUOQIU ZHANG[‡]

Abstract. Addressing the persistent delay problem in traditional network security antivirus protection systems, this paper introduces an innovative approach utilizing virtual reality (VR) technology. The primary objective is to significantly reduce detection delays and enhance the efficiency of network security measures. An enhanced decision-making algorithm is proposed to identify relevant features associated with network security. These features are then weighted and optimized to improve the overall detection process. An injection list is generated through web crawling techniques to strengthen security measures. A virtual protection block is also developed to serve as a barrier against potential threats. The proposed method claims a detection delay of only 75.33 milliseconds, significantly outperforming two traditional methods that recorded 290.11 milliseconds and 337.30 milliseconds, respectively. This substantial decrease in detection delay emphasizes the effectiveness of automatic detection within the context of VR technology. Practical implementation and empirical evidence further validate the success of this approach. The automatic detection of network security vulnerabilities within the VR technology framework is efficient and exhibits considerable progress. As such, this research offers a promising solution to the delay problem in network security antivirus protection. Embracing VR technology achieves shorter detection delays, ultimately improving the security posture of network systems.

Key words: Network Security, Antivirus Protection, Virtual Reality Technology, Detection Delay, Automatic Detection

AMS subject classifications. 15A15, 15A09, 15A23

1. Introduction.

Virtual reality technology has widespread applications across diverse domains, including entertainment, education, and healthcare. In computer network security, leveraging virtual reality technology for vulnerability detection represents a novel and emerging approach. Computer network systems are naturally vulnerable to security threats, which put system components and data integrity at serious risk. Malicious viruses use these gaps, frequently relying on shortcomings in security rules, protocols, hardware, and software implementations, to obtain access to the system without authorization. For instance, vulnerabilities such as flaws in the Network File System (NFS) protocol's authentication method, logic errors in microchips, or misconfigurations by Unix system administrators setting up anonymous File Transfer Protocol (FTP) services can all be exploited as security vulnerabilities within a system [11].

The Morris worm virus, which emerged during the early stages of computer networks, occupies a pivotal position in the records of computer viruses. Its profound impact was a catalyst for the subsequent proliferation of computer viruses. In the wake of the Morris worm, the computer virus has experienced an explosive expansion, characterized by a sheer increase in numbers and a marked elevation in sophistication. Notably, this propagation has been accompanied by a distinct shift in the motivations driving the creation and dissemination of viruses. Initially conceived as experiments, viruses have metamorphosed into instruments wielded for profit-driven endeavours, including cybercrime and data theft. This evolutionary shift has been accompanied by a substantial augmentation of viruses' capabilities, enabling them to escape detection and inflict significant harm upon computer systems and networks. Consequently, computer security has had to adapt continuously, developing strategies and technologies to counter these ever-evolving threats and safeguard digital environments from an expanding array of viruses and malware [4, 15].

The escalating frequency of attacks and threats to computer networks can be attributed to multiple factors. On the one hand, diverse security vulnerabilities and their growth rates continue to rise, often without

[*] Information Engineering College, Shijiazhuang Vocational College of Finance & Economics, Shijiazhuang, 050000, China

[†] Department of Student Affairs, Shijiazhuang Preschool Teachers College, China, 050228 (Corresponding author: jinfangcheng9@126.com)

[‡] Department of Aeronautical Engineering, Shijiazhuang Engineering Vocational College, China, 050061

proportionate attention. On the other hand, the increasing complexity of network systems inherently escalates security risks, making it imperative to explore innovative methods, such as those grounded in virtual reality technology, for robust vulnerability detection and mitigation [13].

Currently, computer network vulnerabilities often appear in the following dimensions [17, 3]:

- Inherent System Vulnerabilities: Present-day operating systems, including Unix and Windows, exhibit various security risks. Virtually every operating system contains known security vulnerabilities that have been identified, addressed, and remain potential threats.
- Unauthorized User Access: Unauthorized users often manage to breach network security measures, gaining entry to networks to which they should not have access.
- Unauthorized Escalation of User Privileges: In some instances, legitimate users may exploit vulnerabilities to increase their access privileges without proper authorization, potentially compromising system security.
- Multi-Vector Attack Vulnerabilities: Network systems are susceptible to attacks from multiple angles and approaches, leaving them vulnerable to various security threats.

These typical vulnerabilities can be traced back to various underlying factors, encompassing security weaknesses within network protocols, vulnerabilities inherent to operating systems, and flaws within different applications and software components.

2. Literature Review. Computer network security frequently encounters challenges, with a particular emphasis on addressing numerous vulnerabilities. These vulnerabilities may result from poor usage, design defects in software or hardware components, or both.

The vulnerabilities in computer network security are predominantly evident in three key dimensions [10]:

(a) Operating System Vulnerabilities: These vulnerabilities inherent to operating systems are the foundation for network infrastructure. Security weaknesses within operating systems can provide entry points for attacks and unauthorized access.

(b) Computer Software Vulnerabilities: Vulnerabilities in various software applications, utilities, and programs utilized within the network environment present another significant risk. These software vulnerabilities can be exploited to compromise network security.

(c) Network Hardware Facility Weaknesses: Hardware components within the network infrastructure may also contain vulnerabilities. These weaknesses can expose critical network assets to threats, making them susceptible to attacks and breaches.

Addressing these vulnerabilities is dominant to strengthening computer network security and mitigating the risks associated with cyberattacks and unauthorized access. Failure to address vulnerabilities within the operating system can have severe repercussions on computer network security. The operating system is a critical component of computer networks, and any vulnerabilities can pose a significant threat. The most substantial security risk within the network environment lies in remote attacks that exploit operating system vulnerabilities. Consequently, ensuring a secure operational environment for computer network operating systems is imperative. Most operating systems exhibit various security vulnerabilities, which may only become apparent during routine usage. Some security risks remain covered within these system vulnerabilities, making them particularly challenging to identify and mitigate [18].

Computer software vulnerabilities constitute a substantial impediment to preserving computer network security. Allowing such vulnerabilities to persist may ultimately result in these flaws evolving into software defects susceptible to external attacks. In instances where software security vulnerabilities occur frequently, particularly if they predominantly consist of high-risk vulnerabilities, they pose a grave threat to the overall security of computer networks. Failing to rectify these vulnerabilities promptly leaves networks susceptible to exploitation and attacks by malicious hackers. It’s crucial to acknowledge that software vulnerabilities are the underlying source of security incidents. This is especially significant when hackers gain access to sensitive information, as it can lead to severe cases of fraudulent activities and other serious security breaches [8].

Numerous security vulnerabilities within computer networks can be attributed to weaknesses in network hardware facilities. For instance, using removable storage media, such as USB flash drives, can potentially lead to the inadvertent disclosure of sensitive information when these drives are borrowed or shared. In conventional methods of detecting computer network security vulnerabilities, inspectors typically must interact

Table 2.1: Comparison of network security vulnerability and detection approaches

Technology	Advantages	Disadvantages
Operating System Vulnerabilities	Critical for network infrastructure security. Understanding and addressing these vulnerabilities are essential.	Difficult to identify and mitigate. Vulnerabilities may not be apparent in routine usage.
Computer Software Vulnerabilities	Significant for overall network security. Prompt remediation is crucial.	Can evolve into software defects susceptible to attacks High-risk vulnerabilities pose a serious threat
Network Hardware Facility Weaknesses	Addresses potential weaknesses in network hardware. Recognizes risks of removable storage media.	Limited by conventional detection methods. Challenges with accuracy, speed, and cost.
Network Vulnerability Detection Methods	Involves simulation experiments and dataset modelling. Offers automated updates in vulnerability detection.	Complexity in acquiring network flow data tables Reduced detection efficiency
Automatic Detection Method	Optimizes weaker variables efficiently. Creates injection point lists and simulates attacks. Utilizes virtual reality technology.	Mainly applied in controlled simulated experiments. Uncertain real-world applicability Requires further empirical testing.

with computer terminals and rely on various vulnerability detection tools. However, these tools often exhibit significant limitations in practical use, including issues related to accuracy, speed, and the associated high costs of vulnerability detection [2].

Network vulnerability detection methods typically involve modelling a sample dataset and extracting network variables for subsequent simulation experiments. For instance, some literature employs the hidden Markov model to represent the sample dataset of network information, aiming to achieve automated updates in network security vulnerability detection. However, this approach introduces complexity in acquiring network flow data tables, reducing detection efficiency and efficiency-related issues [1].

The automatic detection method involves extracting dynamic numerical variables and analyzing static attack processes. It's crucial to highlight that this approach has mainly been employed in controlled simulated experiments, often utilizing open-source software. Although these experiments have provided valuable insights, they come with certain limitations. One notable limitation pertains to the constrained scope and specificity of the experimental outcomes. Since these simulations are artificial and controlled, they may not have adequately explored the full range of real-world scenarios and vulnerabilities. Consequently, this method's practical applicability and real-world effectiveness remain uncertain, as its performance may vary when challenged with the intricacies and diverse landscapes of actual network environments. Further research and empirical testing are warranted to establish the method's reliability and suitability for addressing the intricate and evolving challenges of network security vulnerability detection [14].

An optimization strategy is employed to fine-tune the weighting of weaker variables. This optimization process involves the identification and analysis of vulnerability behaviours by deploying data entry points, as well as the creation of injection point lists. Additionally, virtual reality technology is harnessed to simulate malicious attacks effectively. Through the comprehensive documentation and analysis of fundamental attributes related to virtual protection, research endeavours to implement automatic network security detection within virtual reality technology have been completed [12]. Table 2.1 provides a comparison of network security and vulnerability detection techniques.

3. Research on Automated Network Security Vulnerability Detection using VR Technology. The network security and vulnerability detection technology involves performing source code analysis of the relevant program to identify and address vulnerabilities. In the context of virtual reality technology, automatic network security vulnerability detection relies on simulating and capturing malicious attacks using virtual reality

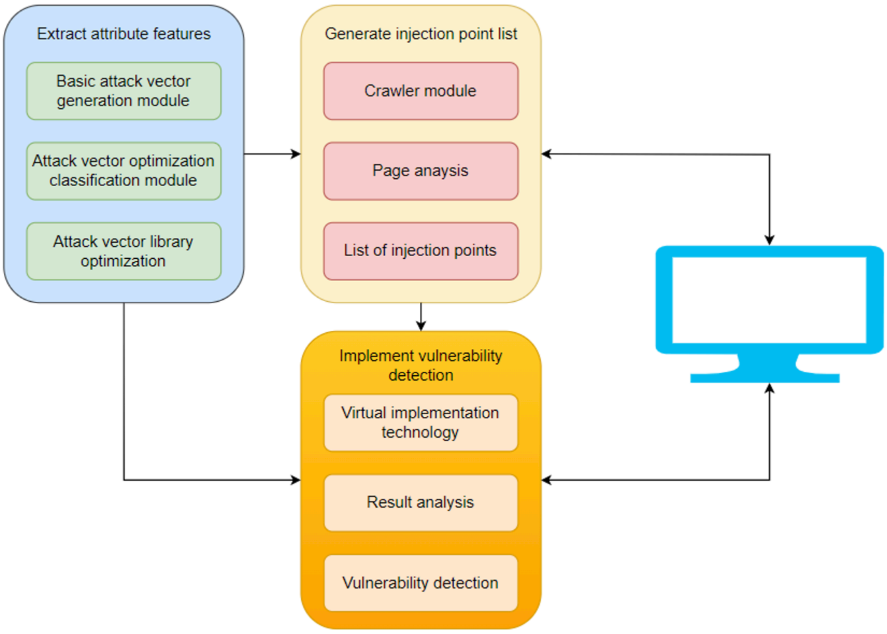
Extract attribute features
Basic attack vector generation module
Attack vector optimization classification module
Attack vector library optimization
Generate injection point list
Crawler module
Page anaysis
List of injection points
Implement vulnerability detection
Virtual implementation technology
Result analysis
Vulnerability detection

Fig. 3.1: Proposed network security and vulnerability detection process

technology. This process involves the automated generation of attack graphs through detection mechanisms. The comprehensive architectural depiction of this method is illustrated in Figure 3.1.

During the development of the attack graph, the primary focus centres on the automated construction of this graph. It involves the creation of an attack vector repository through optimization calculations. Additionally, analysis of attack injection points aims to identify potential entry points within the network. It is achieved by employing web crawlers to retrieve web pages and utilizing crawler tools for web page analysis. The attack and analysis process entails simulating the attack sequence using virtual reality technology. Subsequently, the outcomes generated by this simulation are analyzed to arrive at a final assessment of vulnerabilities.

3.1. Extracting Network Security Vulnerability Characteristic Attributes. Leveraging virtual reality technology for vulnerability detection enables the identification of diverse vulnerabilities within a virtual network environment created for this purpose. Using VR technology, vulnerability testers can interact with vulnerabilities within virtual environments, engaging in activities such as analyzing network traffic, conducting penetration tests on operating systems, and evaluating firewall defences. These operations closely mimic real-world attack scenarios, enhancing vulnerability detection accuracy.

In the automated network security and vulnerability detection process, the network system treats vulnerabilities on nodes as if they were part of its codebase. Employing optimization calculations, the system adjusts the weights assigned to various attributes associated with these vulnerabilities, generating an attack graph. Attackers simulate network attacks through different vulnerabilities over time, with the disparities in weights between vulnerabilities reflected in the attack graph through the aggregation of vulnerability groups [6]. At time t, a cluster with m vulnerabilities is calculated. Suppose there are k individual genes in each vulnerability, the vulnerability gene studied by the author adopts the difficulty gene, exposure gene, repair gene, exposure gene and repair gene, therefore, the value of h: is 3, and the location of a vulnerability i can be expressed as $q_i = (q_{i1}, q_{i2}, q_{i3})$, where $i = 1, 2, \cdots, m$. In the process of generating the attack graph, the starting position of the vulnerability can be regarded as a point of the spatial coordinates, so the position of vulnerability i can be digitized to represent it during coordinate iteration. Set the loophole movement speed, $v_i = (v_{i1}, v_{i2}, v_{i3}), v_{i1}, v_{i2}, v_{i3}$ as the loophole movement speed in the three coordinate axes, and its attributes can be expressed as:

$$V_{i+1} = \omega\vartheta_{ir} + \varepsilon_1 \left(z_{ir} - p_{ir}\right) + \varepsilon_2 \left(z_{ir} - p_{ir}\right) \tag{3.1}$$

In the above Equation, ε_1 is the learning coefficient, ε_2 is the network learning coefficient, ω is the inertia coefficient, ϑ_{ir} is the initial speed of the vulnerability, z_{ir} is the optimal location of the vulnerability, and p_{ir} is the probability of the vulnerability being applied in the attack, so p_{ir} is given by the Equation

$$\sum_{i=1}^{m} p_{ir} = 1, 0 < p_{ir} < 1 \tag{3.2}$$

When ω is between 0 and 1, the vulnerabilities in motion can converge; therefore, when the ω value is determined, the number of malicious attacks can be predicted. When the vulnerability exposure in the network is large, the degree of repair will also increase, and the vulnerability will be difficult to apply. Therefore, the three attributes of vulnerabilities will increase with time, so p_{ir} will decrease accordingly, and the extraction of vulnerability characteristic attributes is completed [5].

3.2. Generate Injection Point List. Based on the extracted vulnerability feature attributes, an analysis of injection points is conducted to produce a comprehensive list of these points. This process relies on a web crawler's assistance, an automated program that captures the vulnerability feature attributes. The crawler's function involves evaluating these attributes to establish the prioritization order for the URLs to be captured while preserving the essential attributes necessary for the analysis.

During the web crawler search process, a significant portion of the URL queue may contain duplicate entries, impeding retrieval speed. We employ the BloomFilter algorithm to mitigate this issue for duplicate removal, vulnerability detection, adjustment, and data storage. As the URL queue is obtained through crawling, each entry undergoes assessment. Entries not already present in the collection are added to it. When identified with an ID of Y, it signifies inclusion in the collection. Thus, they are placed into the URL set. This operation effectively eliminates duplicates, resulting in the compilation of a list of injection points.

The system utilizes the data obtained through crawling attack injection points to perform vulnerability analysis and record the results. This process yields an attribute injection point list to conduct a comprehensive vulnerability analysis based on these records. Subsequently, the system automatically selects the most suitable method to detect the identified vulnerabilities. When implemented through virtual reality technology, vulnerability detection significantly enhances detection speed. Traditional vulnerability detection methods often necessitate manual intervention or specialized vulnerability detection tools, resulting in comparatively slower detection processes. In contrast, vulnerability detection leveraging virtual reality technology can be carried out programmatically, thereby substantially improving the speed and efficiency of the detection process.

3.3. Implementation of Vulnerability Detection under Virtual Reality Technology. Virtual reality technology employs a sense of telepresence to recreate a lifelike perception of the simulated environment. Applying this technology to network security vulnerability detection enables a comprehensive three-dimensional analysis of vulnerabilities within virtual settings. During virtual attack testing, the information in the generated injection point list is a guide. Virtual reality technology is leveraged to simulate attack behaviours and facilitate interaction with the server in line with the identified vulnerabilities [7].

Within this process, since the attacker's identity is entirely virtual, the attributes of network security vulnerabilities serve as the fundamental genetic elements for constructing virtual attacks. Information embedded within network security vulnerabilities is systematically extracted through the established virtual reality environment and virtual attack mechanisms. Security vulnerabilities are classified based on categorizing vulnerability genes, exposure genes, and repair genes inherent in attack information. This constitutes the operational structure during the virtual attack vulnerability detection process.

In the detection phase, artificial virtual vulnerability attacks are executed following the selection and cross-compilation of virtual attack genes. The outcomes of these attacks are then factored into calculating a fitness function alongside other vulnerability-related data. Ultimately, this process culminates in the successful detection of network security vulnerabilities. With this, the research into automatic detection methods for network security vulnerabilities within the virtual reality technology framework is concluded [9].

3.4. Protection Technology for Computer Network Security Vulnerabilities. The diverse types and their high complexity characterize computer network vulnerabilities widespread in many computers and significantly threaten network security. Their presence renders the networks susceptible to external attacks, disrupting regular operations. To overcome these risks, conventional practices involve the installation of vulnerability patches and continuously updating operating systems. Additionally, specialized Trojan-killing software is deployed to eliminate existing Trojan horses, and this software is frequently updated to ensure thorough system checks and eradication. Real-time monitoring functionality is enabled to safeguard against external virus intrusions and system damage proactively [16]. Computer operating systems and software frequently contain numerous vulnerabilities that hackers can exploit to compromise system security. Consequently, keeping the operating system and software up to date by promptly installing the latest patches and security updates is imperative. This practice effectively mitigates the risk of attackers exploiting known vulnerabilities and compromising the system's integrity.

(1) Security configuration switch

One commonly employed method to minimize the forwarding of unicast broadcast traffic on specific ports is the application of multicast or unicast broadcast blocking attributes. By reducing traffic volume on a per-port basis, several advantages are realized. This approach enhances network security by limiting traffic and prevents network devices from processing unneeded, directionless packets. Port security allows or denies traffic based on the host Media Access Control (MAC) address. Depending on the switch model, there may be varying maximum MAC address allowances. This functionality specifies the permissible number of hosts per port, which can then be configured to meet network requirements.

(2) Security configuration router

Routers can be configured to establish a secure perimeter and defend against external attacks within a defined range. Typically, access lists are employed to restrict the source and destination addresses of packets traversing through the router. Additionally, some routers implement the Reverse Path Forwarding (RPF) check to enhance security further. Furthermore, it's possible to configure the "no IP directed broadcast" setting on all routers potentially linked to the target subnet. To improve security measures, the router can turn source route options off using the command "no IP source route". This precautionary step serves as an effective deterrent against source route attacks.

(3) Set up the computer.

To increase security, it's essential to avoid guest accounts, conceal IP addresses, exercise caution with unfamiliar emails, install and regularly update effective security software, turn off "printing and file sharing" when not needed, promptly close unused ports, regularly update administrator information, prevent empty connections, and address program logic vulnerabilities by updating to corrected software versions. When hackers misuse legitimate program functionalities, gaining insight into their tactics is crucial. To defend against attacks, users should avoid hacker-exploited program steps and employ methods to bypass their attack vectors, thus enhancing overall cybersecurity.

(4) Securely configure the Windows server.

In managing Windows servers, it is imperative to prohibit using Remote Registry, Messenger, and Telnet. Additionally, safeguarding important files and directories can be achieved by modifying the registry to hide them. Strengthening defence involves setting and managing accounts with complex passwords. To ensure the security of all network connections within the local system, it is crucial to protect them using Windows Firewall promptly. This requires configuring the appropriate Group Policy parameters in the system environment of Windows Server 2008. Trusted access is essential for verifying user requests and establishing trust when connecting to cloud computing resources, ultimately ensuring cloud security.

Moreover, the cloud security management and defence system, consisting of a cloud security detection platform, a cloud security response platform, and a cloud security recovery strategy, plays a pivotal role in eliminating security vulnerabilities and upholding the security of hosts, networks, and application layers. Regularly updating and maintaining Windows Firewall rules is recommended to safeguard network security. The core of the cloud security management and defence system is the trusted link platform system, which constructs a comprehensive framework through user identity authentication, data encryption, and authorization management, ensuring operational security.

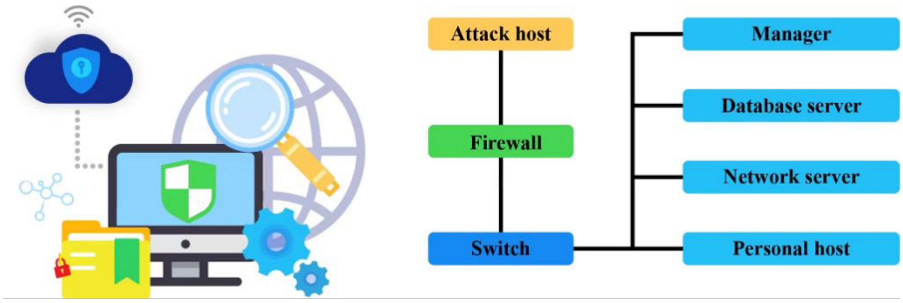

Fig. 4.1: Experimental network topology

Table 4.1: Intranet node vulnerability settings

Node	Vulnerability Number	Server or software	Permission promotion	Ease of use 1%
Network server B	Me10-062	WWW	U-A	70
Network server B	Me11-002	WWW	U-A	100
Database server C	Me11-012	Oracle	O-A	90
Database server C	Me10-065	Oracle	O-U	85
Manager D	Mc10-045	Windows	O-A	100
Personal host E	Mc10-006	Office	O-A	70

4. Simulation Results. To assess the network security vulnerabilities within the VR technology framework established previously, simulation experiments are employed to validate the reliability of the proposed automated vulnerability detection method. These experiments involve comparing the detection outcomes with those obtained using two conventional vulnerability detection technologies, thereby confirming the advantages of the proposed method.

4.1. Experimental Environment and Vulnerability Parameter Settings. In network security and vulnerability detection research, establishing a well-structured experimental environment and carefully configuring the experimental network structure is essential, as shown in Figure 4.1.

Generally, the topology consists of five nodes: A, B, C, D, and E. Node A represents external network host access. The other four internal network nodes serve distinct roles: network services, database services, server protection management, and important data storage. Communication within or between internal network nodes does not necessitate firewall traversal. However, when external network nodes seek to access internal network nodes, they must pass through the firewall, which grants access solely to network servers. However, certain inherent vulnerabilities within the intranet expose internal network nodes to real risks. Table 4.1 outlines these weaknesses in the intranet nodes.

4.2. Experimental Results and Analysis. After constructing the experimental environment, the traditional modelling, numerical, and automatic detection methods of network security vulnerabilities under the virtual reality technology designed by the author are used for experiments.

The key step to reduce the detection time is to improve the network interaction frequency during detection. By optimizing the structure of the neural network, the number of network layers and parameters can be reduced, and the computational efficiency and response speed of the network can be improved. Reducing input data size can reduce the network's processing time and computational complexity, thereby improving the network's response speed. To obtain better vulnerability detection results, 70 detection samples are set to view the change in interaction frequency of the two methods in the detection process. Record the time spent by each method to view requirements, as shown in Figure 4.2.

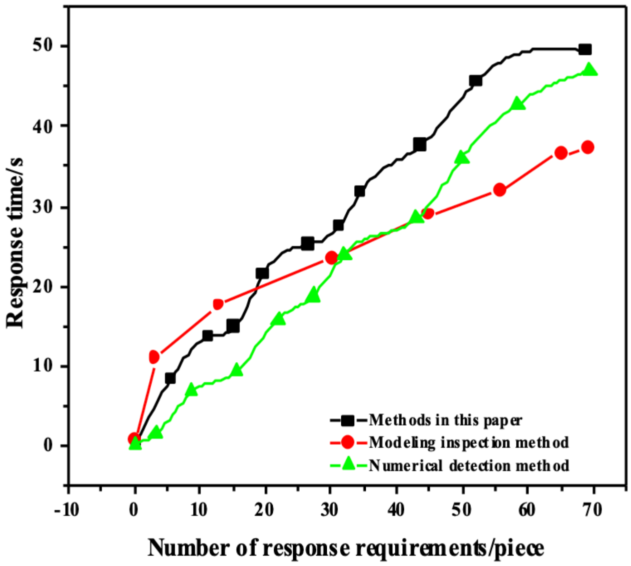

Fig. 4.2: Comparison of interaction frequency detection

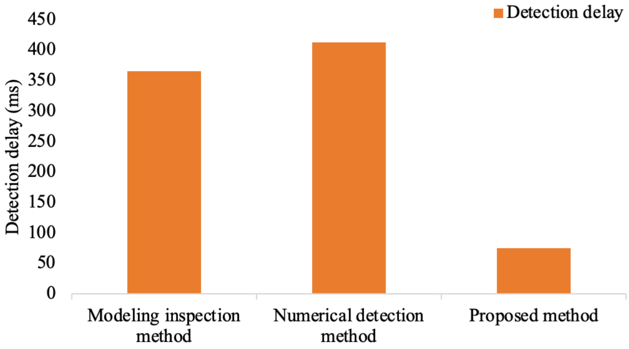

Fig. 4.3: Test and detection delay of the three methods

It can be seen from the above results that with the increase in the number of test samples, the number of methods proposed by the authors to respond to requests is increasing. The first 20 seconds of detection has strong advantages because the modelling detection method covers part of the sample information. When the test time reaches 20 seconds, the method proposed by the author is compared with the traditional two methods. The number of response requirements simultaneously is higher than that of the other two methods, the interaction frequency is higher, and the number of vulnerabilities detected is more, which has obvious advantages.

The simulation network vulnerability detection experiment is carried out for the four target nodes B, C, D and E. Record the time delay of the two methods to detect vulnerabilities and compare the detection results, as shown in Table 4.2 and Figure 4.3.

From the simulation results, the proposed detection method exhibits a detection delay of 75.33ms. Compared to the two traditional methods, this represents a reduction of 290.11ms and 337.30ms. This significant reduction in detection delay serves as compelling evidence for the efficiency of the automatic network security vulnerability detection method within the framework of virtual reality technology.

Table 4.2: Comparison of delay detection for the target nodes

Method	Target Node	Detected vulnerability number	Detection delay (ms)
Modelling inspection method	B	Me10-062	365.44
	C	Mc10-065	
	D	Me10-045	
	E	-	
Numerical detection method	B	Mc10-062 Me11-002	412.63
	C	-	
	D	Me10-045	
	E	Mc10-006	
Proposed method	B	Mc10-062 Mc11-002	75.33
	C	Mc11-012 Mc10-065	
	D	Me10-045	
	E	Mc10-006	

5. Conclusion. Traditional network security antivirus protection suffers from prolonged detection delays. An automatic network security and antivirus protection system based on virtual reality technology is proposed. The system employs optimization techniques to fine-tune vulnerability weights, extracts network security vulnerability features, utilizes web crawlers to capture and identify vulnerability characteristics and employs virtual reality technology for testing malicious attacks. Through extensive experimentation, this novel approach is compared with two traditional methods, resulting in significant reductions in detection delays of 290.11ms and 337.30ms, respectively, thereby confirming the efficiency of automatic network security detection within the virtual reality technology framework. Computer network security and vulnerability detection within virtual reality technology is an emerging and promising field with several potential avenues for future development. These include enhancing interactivity to create more realistic detection environments, integrating deep learning for improved accuracy and speed, and ensuring multi-platform support to provide various operating systems.

REFERENCES

[1] A. Ahmadian Ramaki, A. Rasoolzadegan, and A. Javan Jafari, *A systematic review on intrusion detection based on the hidden markov model*, Statistical Analysis and Data Mining: The ASA Data Science Journal, 11 (2018), pp. 111–134.

[2] R. Amankwah, P. K. Kudjo, and S. Y. Antwi, *Evaluation of software vulnerability detection methods and tools: a review*, International Journal of Computer Applications, 169 (2017), pp. 22–27.

[3] Z. Chen, X. Zuo, N. Dong, and B. Hou, *Application of network security penetration technology in power internet of things security vulnerability detection*, Transactions on Emerging Telecommunications Technologies, 33 (2022), p. e3859.

[4] R. J. Cole, *Computer worms, detection, and defense*, in Encyclopedia of Information Ethics and Security, IGI Global, 2007, pp. 89–95.

[5] X. He, *Analysis of network intrusion detection technology based on computer information security technology*, Journal of Physics: Conference Series, 1744 (2021), p. 042038.

[6] J. Hu, J. Chen, L. Zhang, Y. Liu, Q. Bao, H. Ackah-Arthur, and C. Zhang, *A memory-related vulnerability detection approach based on vulnerability features*, Tsinghua Science and Technology, 25 (2020), pp. 604–613.

[7] ———, *A memory-related vulnerability detection approach based on vulnerability features*, Tsinghua Science and Technology, 25 (2020), pp. 604–613.

[8] W. Huang, X. Li, and Z. Huo, *XSS vuinerability detection technology based on EBNF and twice crawling strategy*, Application Research of Computers, 36 (2019), pp. 2458–2463.

[9] X. Jia, *Research on college sports training based on computer virtual reality technology*, Journal of Physics: Conference Series, 1648 (2020), p. 032132.

[10] J. M. Kizza and J. M. Kizza, *Introduction to computer network vulnerabilities*, Guide to Computer Network Security, (2017), pp. 87–103.

[11] J. Li, P. Cao, and J. Yang, *Research on noc static vulnerability detection system based on big data technology*, Modern Electronics Technique, 42 (2019), pp. 77–81.

[12] R. Liu, *A computer network intrusion detection technology based on improved neural network algorithm*, Telecommunications and Radio Engineering, 79 (2020).

[13] G. Luo, *Research on network security vulnerability detection method based on artificial intelligence*, Journal of Physics: Conference Series, 1651 (2020), p. 012005.

[14] I. Medeiros, N. F. Neves, and M. Correia, *Automatic detection and correction of web application vulnerabilities using data mining to predict false positives*, in Proceedings of the 23rd International Conference on World Wide Web, Republic of Korea, 2014, pp. 63–74.

[15] H. Orman, *The morris worm: A fifteen-year perspective*, IEEE Security & Privacy, 1 (2003), pp. 35–43.

[16] C. Wang, T. Ren, Q. Li, X. Wang, G. Guo, and J. Dong, *Network computer security hidden dangers and vulnerability mining technology*, IOP Conference Series: Materials Science and Engineering, 750 (2020), p. 012155.

[17] M. Yi, X. Xu, and L. Xu, *An intelligent communication warning vulnerability detection algorithm based on iot technology*, IEEE Access, 7 (2019), pp. 164803–164814.

[18] D. Zhaokun, L. Yuliang, H. Zhao, H. Hui, and Z. Kailong, *Network program vulnerability detection technology based on program modeling*, Journal of Beijing University of Aeronautics and Astronautics, 45 (2019), pp. 796–803.

Edited by: Venkatesan C
Special issue on: Next Generation Pervasive Reconfigurable Computing for High Performance Real Time Applications
Received: Mar 21, 2023
Accepted: Oct 8, 2023